別添資料１　位置パッケージの開発

1. 位置特定パッケージ実装設計

1.1. モジュール設計

位置特定パッケージの詳細設計をもとに、モジュール設計として、次に示す位置特定パッケージのクラスのメッソッド等を検討した。

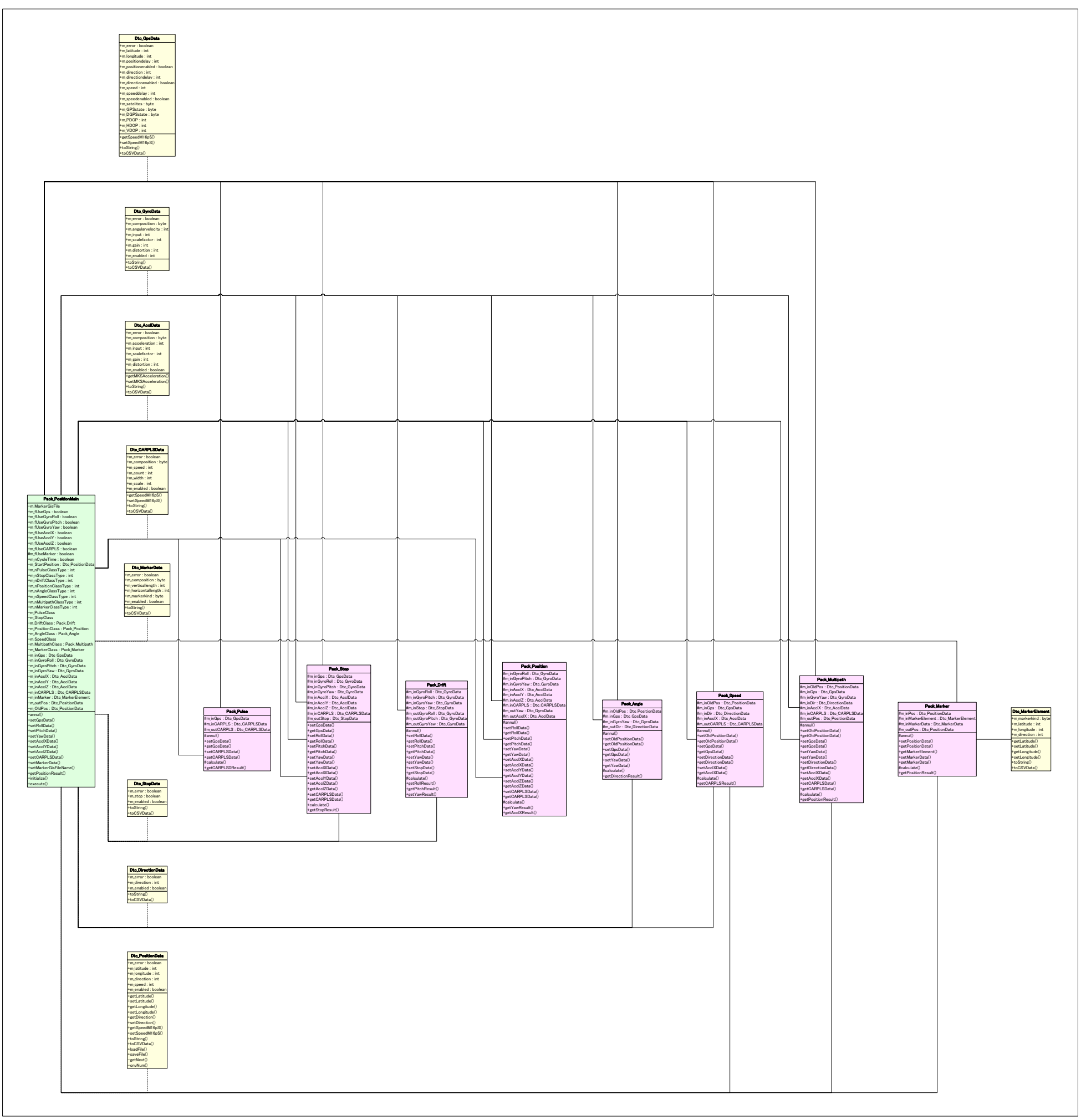


図 1-1　位置特定パッケージ クラス図

(1) データクラス

a） Dtc_GpsData クラス

Dtc_GpsData クラスは車速データを管理するクラスで、フィールドを表 2-59、各メソッドを表 2-60 から表 2-61 にそれぞれ示す。

表 1-1：Dtc_GpsData クラス フィールド

No.	型	変数名	概要	備考
1	boolean	m_error	GPS エラー情報	
2	int	m_latitude	GPS 緯度	
3	int	m_longitude	GPS 経度	
4	int	m_positiondelay	GPS 位置遅延時間	
5	boolean	m_positionenabled	GPS 位置有効情報	
6	int	m_direction	GPS 方位	
7	int	m_directiondelay	GPS 方位遅延時間	
8	boolean	m_directionenabled	GPS 方位有効情報	
9	int	m_speed	GPS 速度	
10	int	m_speeddelay	GPS 速度遅延情報	
11	boolean	m_speedenabled	GPS 速度有効情報	
12	byte	m_GPSstate	GPS 測位状態	
13	byte	m_DGPSstate	ディファレンシャル測位状態	
14	byte	m_satelites	GPS 測位衛星数	
15	int	m_PDOP	PDOP 値	
16	int	m_VDOP	VDOP 値	
17	int	m_HDOP	HDOP 値	

表 1-2：getSpeedM16pS メソッド

メソッド名	getSpeedM16pS	クラス名	Dtc_GpsData

入力

なし

概要

2^{-16} [m/sec] 単位の速度を取得。

出力

2^{-16} [m/sec] 単位の速度。

表 1-3：setSpeedM16pS メソッド

メソッド名	setSpeedM16pS	クラス名	Dtc_GpsData

入力

2^{-16} [m/sec] 単位の速度。

概要

2^{-16} [m/sec] 単位の速度を設定。

出力

なし。

表 1-4：toString メソッド

メソッド名	toString	クラス名	Dtc_GpsData

入力

なし。

概要

フィールドの値を文字列に変換。

出力

フィールドの文字列。

表 1-5：toCSVData メソッド

メソッド名	toCSVData	クラス名	Dtc_GpsData

入力

なし。

概要

CSV ファイル出力用の文字列を作成。

出力

CSV 出力文字列。

b） Dtc_GyroData クラス

Dtc_GyroData クラスは GPS データを管理するクラスで、フィールドを表 2-64、各メソッドを表 2-65 から表 2-66 にそれぞれ示す。

表 1-6：Dtc_GyroData クラス フィールド

No.	型	変数名	概要	備考
1	boolean	m_error	ジャイロエラー情報	
2	byte	m_composition	ジャイロ構成	
3	int	m_angularvelocity	角速度	
4	int	m_input	センサ入力値	
5	int	m_scalefactor	スケールファクタ	
6	int	m_distortion	歪み	
7	int	m_gain	ゲイン	
8	boolean	m_enabled	ジャイロ有効情報	

表 1-7：toString メソッド

メソッド名	toString	クラス名	Dtc_GyroData

入力

なし。

概要

フィールドの値を文字列に変換。

出力

フィールドの文字列。

表 1-8：toCSVData メソッド

メソッド名	toCSVData	クラス名	Dtc_GyroData

入力

なし。

概要

CSV ファイル出力用の文字列を作成。

出力

CSV 出力文字列。

c） Dtc_AcclData クラス

Dtc_AcclData クラスは加速度データを管理するクラスで、フィールドを表 2-67、各メソッドを表 2-68 から表 2-71 にそれぞれ示す。

表 1-9：Dtc_AcclData クラス フィールド

No.	型	変数名	概要	備考
1	boolean	m_error	加速度計エラー情報	
2	byte	m_composition	加速度計構成	
3	int	m_acceleration	加速度	
4	int	m_input	センサ入力値	
5	int	m_scalefactor	スケールファクタ	
6	int	m_distortion	歪み	
7	int	m_gain	ゲイン	
8	boolean	m_enabled	加速度計有効情報	
9	double	G2MKS	G/MKS 変換パラメータ	

表 1-10：getMKSAcceleration メソッド

メソッド名	getMKSAcceleration	クラス名	Dtc_AcclData

入力

なし

概要

0.1×2^{-16} [m/sec^2] 単位の加速度を取得。

出力

0.1×2^{-16} [m/sec^2] 単位の加速度

表 1-11：setMKSAcceleration メソッド

メソッド名	setMKSAcceleration	クラス名	Dtc_AcclData

入力

0.1×2^{-16} [m/sec^2] 単位の加速度

概要

0.1×2^{-16} [m/sec^2] 単位の加速度を設定。

出力

なし

表 1-12：toString メソッド

メソッド名	toString	クラス名	Dtc_AcclData

入力

なし

概要

フィールドの値を文字列に変換。

出力

フィールドの文字列。

表 1-13：toCSVData メソッド

メソッド名	toCSVData	クラス名	Dtc_AcclData

入力

なし

概要

CSV ファイル出力用の文字列を作成。

出力

CSV 出力文字列

d） Dtc_CARPLSData クラス

Dtc_CARPLSData クラスは車速データを管理するクラスで、フィールドを表 2-72、各メソッドを表 2-73 から表 2-76 にそれぞれ示す。

表 1-14：Dtc_CARPLSData クラス フィールド

No.	型	変数名	概要	備考
1	boolean	m_error	車速センサエラー情報	
2	byte	m_composition	車速センサ構成	
3	int	m_speed	車速パルス速度	
4	int	m_count	パルス数	
5	int	m_width	パルス幅	
6	int	m_scale	パルススケール	
7	boolean	m_enabled	車速センサ有効情報	

表 1-15：getSpeedM16pS メソッド

メソッド名	getSpeedM16pS	クラス名	Dtc_CARPLSData

入力

なし。

概要

2^{-16} [m/sec] 単位の速度を取得。

出力

2^{-16} [m/sec] 単位の速度。

表 1-16：setSpeedM16pS メソッド

メソッド名	setSpeedM16pS	クラス名	Dtc_CARPLSData

入力

2^{-16} [m/sec] 単位の速度。

概要

2^{-16} [m/sec] 単位の速度を設定。

出力

なし。

表 1-17：toString メソッド

メソッド名	toString	クラス名	Dtc_CARPLSData

入力

なし。

概要

フィールドの値を文字列に変換。

出力

フィールドの文字列。

表 1-18：toCSVData メソッド

メソッド名	toCSVData	クラス名	Dtc_CARPLSData

入力

なし。

概要

CSV ファイル出力用の文字列を作成。

出力

CSV 出力文字列。

e） Dtc_MarkerData クラス

Dtc_MarkerData クラスはレーンマーカデータを管理するクラスで、フィールドを表 2-77、各メソッドを表 2-78 から表 2-79 にそれぞれ示す。

表 1-19：Dtc_MarkerData クラス フィールド

No.	型	変数名	概要	備考
1	boolean	m_error	マーカセンサエラー情報	
2	byte	m_composition	マーカセンサ構成	
3	int	m_horizontallength	マーカ横距離	
4	int	m_verticallength	マーカ縦距離	
5	byte	m_markerkind	マーカ種別	
6	boolean	m_enabled	マーカセンサ有効情報	

表 1-20：toString メソッド

メソッド名	toString	クラス名	Dtc_MarkerData

入力

なし。

概要

フィールドの値を文字列に変換。

出力

フィールドの文字列。

表 1-21：toCSVData メソッド

メソッド名	toCSVData	クラス名	Dtc_MarkerData

入力

なし。

概要

CSV ファイル出力用の文字列を作成。

出力

CSV 出力文字列。

f） Dtc_StopData クラス

Dtc_StopData クラスは停止状態データを管理するクラスで、フィールドを表 2-80、各メソッドを表 2-81 から表 2-82 にそれぞれ示す。

表 1-22：Dtc_StopData クラス フィールド

No.	型	変数名	概要	備考
1	boolean	m_error	エラー情報	
2	boolean	m_stop	停止情報	
3	boolean	m_enabled	有効情報	

表 1-23：toString メソッド

メソッド名	toString	クラス名	Dtc_StopData

入力

なし。

概要

フィールドの値を文字列に変換。

出力

フィールドの文字列。

表 1-24：toCSVData メソッド

メソッド名	toCSVData	クラス名	Dtc_StopData

入力

なし。

概要

CSV ファイル出力用の文字列を作成。

出力

CSV 出力文字列。

g） Dtc_DirectionData クラス

Dtc_DirectionData クラスは方位データを管理するクラスで、フィールドを表 2-83、各メソッドを表 2-84 から表 2-85 にそれぞれ示す。

表 1-25：Dtc_DirectionData クラス フィールド

No.	型	変数名	概要	備考
1	boolean	m_error	エラー情報	
2	int	m_direction	方位角	
3	boolean	m_enabled	有効情報	

表 1-26：toString メソッド

メソッド名	toString	クラス名	Dtc_DirectionData

入力

なし。

概要

フィールドの値を文字列に変換。

出力

フィールドの文字列。

表 1-27：toCSVData メソッド

メソッド名	toCSVData	クラス名	Dtc_DirectionData

入力

なし。

概要

CSV ファイル出力用の文字列を作成。

出力

CSV 出力文字列。

h） Dtc_PositionData クラス

Dtc_PositionData クラスは現在位置データを管理するクラスで、フィールドを表 2-86、各メソッドを表 2-87 から表 2-90 にそれぞれ示す。

表 1-28：Dtc_PositionData クラス フィールド

No.	型	変数名	概要	備考
1	boolean	m_error	エラー情報	
2	int	m_latitude	緯度	
3	int	m_longitude	経度	
4	int	m_direction	方位	
5	int	m_speed	速度	
6	boolean	m_enabled	有効情報	

表 1-29：getLatitude メソッド

メソッド名	getLatitude	クラス名	Dtc_PositionData

入力

なし。

概要

[deg] 単位の緯度を取得。

出力

[deg] 単位の緯度。

表 1-30：setLatitude メソッド

メソッド名	setLatitude	クラス名	Dtc_PositionData

入力

[deg] 単位の緯度。

概要

[deg] 単位の緯度を設定。

出力

なし。

表 1-31：getLongitude メソッド

メソッド名	getLongitude	クラス名	Dtc_PositionData

入力

なし。

概要

[deg] 単位の経度を取得。

出力

[deg] 単位の経度。

表 1-32：setLongitude メソッド

メソッド名	setLongitude	クラス名	Dtc_PositionData

入力

[deg] 単位の経度。

概要

[deg] 単位の経度を設定。

出力

なし。

表 1-33：getDirection メソッド

メソッド名	getDirection	クラス名	Dtc_PositionData

入力

なし。

概要

[deg] 単位の方位を取得。

出力

[deg] 単位の方位。

表 1-34：setDirection メソッド

メソッド名	setDirection	クラス名	Dtc_PositionData

入力

[deg] 単位の方位。

概要

[deg] 単位の方位を設定。

出力

なし。

表 1-35：getSpeedM16pS メソッド

メソッド名	getSpeedM16pS	クラス名	Dtc_PositionData

入力

なし。

概要

2^{-16} [m/sec] 単位の速度を取得。

出力

2^{-16} [m/sec] 単位の速度。

表 1-36：setSpeedM16pS メソッド

メソッド名	setSpeedM16pS	クラス名	Dtc_PositionData

入力

2^{-16} [m/sec] 単位の速度。

概要

2^{-16} [m/sec] 単位の速度を設定。

出力

なし。

表 1-37：toString メソッド

メソッド名	toString	クラス名	Dtc_PositionData

入力

なし。

概要

フィールドの値を文字列に変換。

出力

フィールドの文字列。

表 1-38：toCSVData メソッド

メソッド名	toCSVData	クラス名	Dtc_PositionData

入力

なし。

概要

CSV ファイル出力用の文字列を保存。

出力

CSV 出力文字列。

表 1-39：saveFile メソッド

メソッド名	saveFile	クラス名	Dtc_PositionData

入力

CSV ファイル名。

概要

データを CSV ファイルへ書込。

出力

保存結果。

表 1-40：loadFile メソッド

メソッド名	loadFile	クラス名	Dtc_PositionData

入力

CSV ファイル名。

概要

データを CSV ファイルから読込。

出力

現在位置データ。

i） Dtc_MarkerElement クラス

Dtc_MarkerElement クラスは検索したレーンマーカマーカデータを管理するクラスで、フィールドを表 2-99、各メソッドを表 2-100 から表 2-105 にそれぞれ示す。

表 1-41　Dtc_MarkerElement クラス フィールド

No.	型	変数名	概要	備考
1	byte	m_markerkind	マーカ種別	
2	int	m_latitude	緯度	
3	int	m_longitude	経度	
4	int	m_direction	方位	

表 1-42　getLatitude メソッド

メソッド名	getLatitude	クラス名	Dtc_MarkerElement

入力

なし。

概要

[deg] 単位の緯度を取得。

出力

[deg] 単位の緯度。

表 1-43　setLatitude メソッド

メソッド名	setLatitude	クラス名	Dtc_MarkerElement

入力

[deg] 単位の緯度。

概要

[deg] 単位の緯度を設定。

出力

なし。

表 1-44　getLongitude メソッド

メソッド名	getLongitude	クラス名	Dtc_MarkerElement

入力

なし。

概要

[deg] 単位の経度を取得。

出力

[deg] 単位の経度。

表 1-45　setLongitude メソッド

メソッド名	setLongitude	クラス名	Dtc_MarkerElement

入力

[deg] 単位の経度。

概要

[deg] 単位の経度を設定。

出力

なし。

表 1-46　toString メソッド

メソッド名	toString	クラス名	Dtc_MarkerElement

入力

なし。

概要

フィールドの値を文字列に変換。

出力

フィールドの文字列。

表 1-47　toCSVData メソッド

メソッド名	toCSVData	クラス名	Dtc_MarkerElement

入力

なし。

概要

CSV ファイル出力用の文字列を作成。

出力

CSV 出力文字列。

j） Dtc_MeshElement クラス

メッシュデータの定義クラスである Dtc_MeshElement クラスのメソッドについて、以下に示す。

表 1-48　Dtc_MeshElement クラス フィールド

No.			型	フィールド	概要
1	public		int	m_meshID	メッシュデータ ID
2	public		int	m_laneNO	メッシュデータ車線番号
3	public		Dtc_PointData	m_pointA	メッシュデータ地点 A の緯度・経度
4	public		Dtc_PointData	m_pointB	メッシュデータ地点 B の緯度・経度
5	public		Dtc_PointData	m_pointC	メッシュデータ地点 C の緯度・経度
6	public		Dtc_PointData	m_pointD	メッシュデータ地点 D の緯度・経度
7	public		byte	m_type	メッシュデータ逸脱可フラグ
8	public		Dtc_PointData	m_Center	メッシュデータ中心緯度・経度
9	public		int	m_Direction	メッシュデータ方位
10	public		int	m_Index	メッシュデータ行番号
11	public	static	Dtc_PointData	point_1	メッシュデータ地点 A-B 間の中間点 1
12	public	static	Dtc_PointData	point_3	メッシュデータ地点 C-D 間の中間点 3
13	public	static	double	dist_1	任意点とメッシュデータ中間点 1 間の距離
14	public	static	double	dist_3	任意点とメッシュデータ中間点 3 間の距離
15	public	static	Dtc_PointData	point_2	任意点をメッシュデータ中心線上に移動した点 2
16	public	static	double	dist_2	任意点とメッシュデータ中間点 2 間の距離

表 1-49：getCenter メソッド

メソッド名	getCenter	クラス名	Dtc_MeshElement

入力

Dtc_PointData pA ⇒ メッシュデータ地点 A 緯度・経度
Dtc_PointData pB ⇒ メッシュデータ地点 B 緯度・経度

概要

メッシュデータの中心点を算出。

出力

メッシュデータの中心点。

メソッド名	getDirection	クラス名	Dtc_MeshElement

入力

Dtc_PointData pA ⇒ メッシュデータ地点 A 緯度・経度
Dtc_PointData pB ⇒ メッシュデータ地点 B 緯度・経度

概要

メッシュデータの方位を算出。

出力

メッシュデータの方位。

表 1-51：getDistance メソッド

メソッド名	getDistance	クラス名	Dtc_MeshElement

入力

Dtc_PointData pA ⇒ 地点 1 緯度・経度
Dtc_PointData pB ⇒ 地点 2 緯度・経度

概要

地点 1 と地点 2 間の距離を算出。

出力

距離。

表 1-52：getSideSlidePoint2 メソッド

メソッド名	getSideSlidePoint2	クラス名	Dtc_MeshElement

入力

Dtc_MeshElement mesh⇒ メッシュデータ
Dtc_PositionData in⇒ 任意の緯度・経度

概要

任意点をメッシュデータの中心線上へ移動した点 2 を算出。

出力

なし。

表 1-53：getPoint2 メソッド

メソッド名	getPoint2	クラス名	Dtc_MeshElement

入力

なし。

概要

メッシュデータの中心線上へ移動した点 2 を返却。

出力

メッシュデータの中心線上へ移動した点 2。

表 1-54：getDist2 メソッド

メソッド名	getDist2	クラス名	Dtc_MeshElement

入力

なし。

概要

任意点とメッシュデータの中心線上へ移動した点 2 間の距離を返却。

出力

任意点とメッシュデータの中心線上へ移動した点 2 間の距離。

表 1-55：getSideSlidePoint13 メソッド

メソッド名	getSideSlidePoint13	クラス名	Dtc_MeshElement

入力

Dtc_MeshElement mesh⇒ メッシュデータ
Dtc_PositionData in⇒ 任意の緯度・経度

概要

任意点とメッシュデータの境界へ移動した点 1 および点 3 間の距離を算出。

出力

任意点とメッシュデータの境界へ移動した点 1 および点 3 間の距離。

表 1-56：getPoint1 メソッド

メソッド名	getPoint1	クラス名	Dtc_MeshElement

入力

なし。

概要

メッシュデータの境界 A-B へ移動した点 1 を返却。

出力

メッシュデータの境界 A-B へ移動した点 1。

メソッド名	getPoint3	クラス名	Dtc_MeshElement

入力

なし。

概要

メッシュデータの境界 C-D へ移動した点 3 を返却。

出力

メッシュデータの境界 C-D へ移動した点 3。

表 1-58：getDist1 メソッド

メソッド名	getDist1	クラス名	Dtc_MeshElement

入力

なし。

概要

任意点とメッシュデータの境界 A-B へ移動した点 1 間の距離を返却。

出力

任意点とメッシュデータの境界 A-B へ移動した点 1 間の距離。

メソッド名	getDist3	クラス名	Dtc_MeshElement

入力

なし。

概要

任意点とメッシュデータの境界 C-D へ移動した点 3 間の距離を返却。

出力

任意点とメッシュデータの境界 C-D へ移動した点 3 間の距離。

(2) 補正クラス

a） Pack_Pulse クラス

Pack_Pulse クラスのメソッドについて、以下に示す。

Pack_Pulse クラスはジャイロのドリフトを除去するクラスで、フィールドを表 2-118、各メソッドを表 2-119 から表 2-125 にそれぞれ示す。

表 1-60　Pack_Pulse クラス フィールド

protected Dtc_CARPLSData	**m_inCARPLS**　入力車速パルスデータ
protected Dtc_GpsData	**m_inGps**　入力GPSデータ
protected Dtc_CARPLSData	**m_outCARPLS**　出力車速パルスデータ

表 1-61　annul メソッド

メソッド名	annul	クラス名	Pack_Pulse

入力

なし。

概要

算出結果を無効化。

出力

なし。

表 1-62　setGpsData メソッド

メソッド名	setGpsData	クラス名	Pack_Pulse

入力

GPS データ。

概要

GPS データを設定。

出力

なし。

表 1-63　getGpsData メソッド

メソッド名	getGpsData	クラス名	Pack_Pulse

入力

なし。

概要

GPS データを取得。

出力

GPS データ。

表 1-64　setCARPLSData メソッド

メソッド名	setCARPLSData	クラス名	Pack_Pulse

入力

車速データ。

概要

車速データを設定。

出力

なし。

表 1-65　getCARPLSData メソッド

メソッド名	getCARPLSData	クラス名	Pack_Pulse

入力

なし。

概要

車速データを取得。

出力

車速データ。

表 1-66　calculate メソッド

メソッド名	calculate	クラス名	Pack_Pulse

入力

なし。

概要

パルススケールを補正し、車速パルス速度を算出。

出力

なし。

表 1-67　getCARPLSDResult メソッド

メソッド名	getCARPLSDResult	クラス名	Pack_Pulse

入力

なし。

概要

車速データを取得。

出力

車速データ。

b） Pack_DllPulse クラス

Pack_DllPulse クラスは Pack_Pulse クラスの継承であるため、メソッド概要については Pack_Pulse クラスを参照のこと。

c） Pack_Stop クラス

Pack_Stop クラスは車両の停止状態を判定するクラスで、フィールドを、各メソッドをからにそれぞれ示す。

表 1-68　Pack_Drift クラス フィールド

protected Dtc_AcclData	**m_inAcclX**　入力前後加速度データ
protected Dtc_AcclData	**m_inAcclY**　入力左右加速度データ
protected Dtc_AcclData	**m_inAcclZ**　入力上下加速度データ
protected Dtc_CARPLSData	**m_inCARPLS**　入力車速パルスデータ
protected Dtc_GpsData	**m_inGps**　入力GPSデータ
protected Dtc_GyroData	**m_inGyroPitch**　入力ピッチデータ
protected Dtc_GyroData	**m_inGyroRoll**　入力ロールデータ
protected Dtc_GyroData	**m_inGyroYaw**　入力方位データ
protected Dtc_StopData	**m_outStop**　出力停止データ

表 1-69　annul メソッド

メソッド名	annul	クラス名	Pack_Stop

入力

なし。

概要

算出結果を無効化。

出力

なし。

表 1-70　setGpsData メソッド

メソッド名	setGpsData	クラス名	Pack_Stop

入力

GPS データ。

概要

GPS データを設定。

出力

なし。

表 1-71　getGpsData メソッド

メソッド名	getGpsData	クラス名	Pack_Stop

入力

なし。

概要

GPS データを取得。

出力

GPS データ。

表 1-72　setRollData メソッド

メソッド名	setRollData	クラス名	Pack_Stop

入力

ロール角速度データ。

概要

ロール角速度データを設定。

出力

なし。

表 1-73　getRollData メソッド

メソッド名	getRollData	クラス名	Pack_Stop

入力

なし。

概要

ロール角速度データを取得。

出力

ロール角速度データ。

表 1-74　setPitchData メソッド

メソッド名	setPitchData	クラス名	Pack_Stop

入力

ピッチ角速度データ。

概要

ピッチ角速度データを設定。

出力

なし。

表 1-75　getPitchData メソッド

メソッド名	getPitchData	クラス名	Pack_Stop

入力

なし。

概要

ピッチ角速度データを取得。

出力

ピッチ角速度データ。

表 1-76　setYawData メソッド

メソッド名	setYawData	クラス名	Pack_Stop

入力

方位角速度データ。

概要

方位角速度データを設定。

出力

なし。

表 1-77　getYawData メソッド

メソッド名	getYawData	クラス名	Pack_Stop

入力

なし。

概要

入力方位データ取得

出力

方位角データ。

表 1-78　setAcclXData メソッド

メソッド名	setAcclXData	クラス名	Pack_Stop

入力

前後加速度データ。

概要

入力前後加速度データ設定。

出力

なし。

表 1-79　getAcclXData メソッド

メソッド名	getAcclXData	クラス名	Pack_Stop

入力

なし。

概要

入力前後加速度データ取得。

出力

前後加速度データ。

表 1-80　setAcclYData メソッド

メソッド名	setAcclYData	クラス名	Pack_Stop

入力

左右加速度データ。

概要

入力左右加速度データ設定。

出力

なし。

表 1-81　getAcclYData メソッド

メソッド名	getAcclYData	クラス名	Pack_Stop

入力

なし。

概要

入力左右加速度データ取得。

出力

左右加速度データ。

表 1-82　setAcclZData メソッド

メソッド名	setAcclZData	クラス名	Pack_Stop

入力

上下加速度データ。

概要

入力上下加速度データ設定。

出力

なし。

表 1-83　getAcclZData メソッド

メソッド名	getAcclZData	クラス名	Pack_Stop

入力

なし。

概要

入力上下加速度データ取得。

出力

上下加速度データ。

表 1-84　setCARPLSData メソッド

メソッド名	setCARPLSData	クラス名	Pack_Stop

入力

車速パルスデータ。

概要

入力車速パルスデータ設定。

出力

なし。

表 1-85　getCARPLSData メソッド

メソッド名	getCARPLSData	クラス名	Pack_Stop

入力

なし。

概要

入力車速パルスデータ取得。

出力

車速パルスデータ。

表 1-86　calculate メソッド

メソッド名	calculate	クラス名	Pack_Stop

入力

なし。

概要

停止判定計算処理。

出力

なし。

表 1-87　getStopResult メソッド

メソッド名	getStopResult	クラス名	Pack_Stop

入力

なし。

概要

結果停止データ取得。

出力

停止データ。

d） Pack_DllStop クラス

Pack_DllStop クラスは Pack_Stop クラスの継承であるため、メソッド概要については Pack_Stop クラスを参照のこと。

e） Pack_Drift クラス

Pack_Drift クラスはジャイロのドリフトを除去するクラスで、フィールドを表 2-146、各メソッドを表 2-147 から表 2-159 にそれぞれ示す。

表 1-88　Pack_Drift クラス フィールド

protected Dtc_GyroData	**m_inGyroPitch**　入力ピッチデータ
protected Dtc_GyroData	**m_inGyroRoll**　入力ロールデータ
protected Dtc_GyroData	**m_inGyroYaw**　入力方位データ
protected Dtc_StopData	**m_inStop**　入力停止情報データ
protected Dtc_GyroData	**m_outGyroPitch**　出力ピッチデータ
protected Dtc_GyroData	**m_outGyroRoll**　出力ロールデータ
protected Dtc_GyroData	**m_outGyroYaw**　出力方位データ

表 1-89　annul メソッド

メソッド名	annul	クラス名	Pack_Drift

入力

なし。

概要

算出結果を無効化。

出力

なし。

表 1-90　setRollData メソッド

メソッド名	setRollData	クラス名	Pack_Drift

入力

ロール角速度データ。

概要

ロール角速度データを設定。

出力

なし。

表 1-91　getRollData メソッド

メソッド名	getRollData	クラス名	Pack_Drift

入力

なし。

概要

ロール角速度データを取得。

出力

ロール角速度データ。

表 1-92　setPitchData メソッド

メソッド名	setPitchData	クラス名	Pack_Drift

入力

ピッチ角速度データ。

概要

ピッチ角速度データを設定。

出力

なし。

表 1-93　getPitchData メソッド

メソッド名	getPitchData	クラス名	Pack_Drift

入力

なし。

概要

ピッチ角速度データを取得。

出力

ピッチ角速度データ。

表 1-94　setYawData メソッド

メソッド名	setYawData	クラス名	Pack_Drift

入力

方位角速度データ。

概要

方位角速度データを設定。

出力

なし。

表 1-95　getYawData メソッド

メソッド名	getYawData	クラス名	Pack_Drift

入力

なし。

概要

方位角速度データを取得。

出力

方位角速度データ。

表 1-96　setStopData メソッド

メソッド名	setStopData	クラス名	Pack_Drift

入力

停止状態データ。

概要

停止状態データを設定。

出力

なし。

表 1-97　getStopData メソッド

メソッド名	getStopData	クラス名	Pack_Drift

入力

なし。

概要

停止状態データを取得。

出力

停止状態データ。

表 1-98　calculate メソッド

メソッド名	calculate	クラス名	Pack_Drift

入力

なし。

概要

ドリフトを算出し、角速度を補正。

出力

なし。

表 1-99　getRollResult メソッド

メソッド名	getRollResult	クラス名	Pack_Drift

入力

なし。

概要

ロール角速度データを取得。

出力

ロール角速度データ。

表 1-100　getPitchResult メソッド

メソッド名	getPitchResult	クラス名	Pack_Drift

入力

なし。

概要

ピッチ角速度データを取得。

出力

ピッチ角速度データ。

表 1-101　getYawResult メソッド

メソッド名	getYawResult	クラス名	Pack_Drift

入力

なし。

概要

方位角速度データを取得。

出力

方位角速度データ。

f） Pack_DllDrift クラス

Pack_ DllDrift クラスは Pack_Drift クラスの継承であるため、メソッド概要については Pack_Drift クラスを参照のこと。

g） Pack_Angle クラス

Pack_Angle クラスは方位を算出するクラスで、フィールドを表 2-160、各メソッドを表 2-161 から表 2-169 にそれぞれ示す。

表 1-102　Pack_Angle クラス フィールド

protected Dtc_GpsData	**m_inGps**　入力GPSデータ
protected Dtc_GyroData	**m_inGyroYaw**　入力方位データ
protected static Dtc_PositionData	**m_inOldPos**　入力前回位置データ
protected Dtc_DirectionData	**m_outDir**　出力方位データ

表 1-103　annul メソッド

メソッド名	annul	クラス名	Pack_Angle

入力

なし。

概要

算出結果を無効化。

出力

なし。

表 1-104　setOldPositionData メソッド

メソッド名	setOldPositionData	クラス名	Pack_Angle

入力

前回位置データ。

概要

前回位置データを設定。

出力

なし。

表 1-105　getOldPositionData メソッド

メソッド名	getOldPositionData	クラス名	Pack_Angle

入力

なし。

概要

前回位置データを取得。

出力

前回位置データ。

表 1-106　setGpsData メソッド

メソッド名	setGpsData	クラス名	Pack_Angle
入力			
GPS データ設定			
概要			
GPS データを設定。			
出力			
なし。			

表 1-107　getGpsData メソッド

メソッド名	getGpsData	クラス名	Pack_Angle

入力

なし。

概要

GPS データを取得。

出力

GPS データ。

表 1-108　setYawData メソッド

メソッド名	setYawData	クラス名	Pack_Angle

入力

方位角速度データ。

概要

方位角速度データを設定。

出力

なし。

表 1-109　getYawData メソッド

メソッド名	getYawData	クラス名	Pack_Angle

入力

なし。

概要

方位角速度データを取得。

出力

方位角速度データ。

表 1-110　calculate メソッド

メソッド名	calculate	クラス名	Pack_Angle

入力

なし。

概要

方位を算出。

出力

なし。

表 1-111　getDirectionResult メソッド

メソッド名	getDirectionResult	クラス名	Pack_Angle

入力

なし。

概要

算出した方位データを取得。

出力

方位データ。

h） Pack_DllAngle クラス

Pack_DllAngle クラスは Pack_Angle クラスの継承であるため、メソッド概要については Pack_Angle クラスを参照のこと。

i） Pack_Speed クラス

Pack_Speed クラスのメソッドについて、以下に示す。

表 1-112　Pack_Speed　クラス フィールド

protected	Dtc_AcclData	**m_inAcclX**　入力前後加速度データ
protected	Dtc_DirectionData	**m_inDir**　入力方位データ
protected	Dtc_GpsData	**m_inGps**　入力GPSデータ
protected static	Dtc_PositionData	**m_inOldPos**　入力前回位置データ
protected	Dtc_CARPLSData	**m_outCARPLS**　出力車速パルスデータ

表 1-113　annul メソッド

メソッド名	annul	クラス名	Pack_Speed

入力

なし。

概要

結果無効化処理。

出力

なし。

表 1-114　setOldPositionData メソッド

メソッド名	setOldPositionData	クラス名	Pack_Speed

入力

前回位置データ。

概要

前回位置データ設定。

出力

なし。

表 1-115　getOldPositionData メソッド

メソッド名	getOldPositionData	クラス名	Pack_Speed

入力

なし。

概要

前回位置データ取得。

出力

前回位置データ。

表 1-116　setGpsData メソッド

メソッド名	setGpsData	クラス名	Pack_Speed

入力

GPS データ。

概要

GPS データ設定。

出力

なし。

表 1-117　getGpsData メソッド

メソッド名	getGpsData	クラス名	Pack_Speed

入力

なし。

概要

GPS データ取得。

出力

GPS データ。

表 1-118　setDirectionData メソッド

メソッド名	setDirectionData	クラス名	Pack_Speed
入力			
方位角データ。			
概要			
入力方位データ設定。			
出力			
なし。			

表 1-119　getDirectionData メソッド

メソッド名	getDirectionData	クラス名	Pack_Speed

入力

なし。

概要

入力方位データ取得。

出力

方位角データ。

表 1-120　setAcclXData メソッド

メソッド名	setAcclXData	クラス名	Pack_Speed

入力

前後加速度データ。

概要

入力前後加速度データ設定。

出力

なし。

表 1-121　getAcclXData メソッド

メソッド名	getAcclXData	クラス名	Pack_Speed

入力

なし。

概要

入力前後加速度データ取得。

出力

前後加速度データ。

表 1-122　calculate メソッド

メソッド名	calculate	クラス名	Pack_Speed

入力

なし。

概要

車速算出処理。

出力

なし。

表 1-123　getCARPLSResult メソッド

メソッド名	getCARPLSResult	クラス名	Pack_Speed

入力

なし。

概要

結果車速データ取得。

出力

車速センサデータ。

j） Pack_DllSpeed クラス

Pack_DllSpeed クラスは Pack_Speed クラスの継承であるため、メソッド概要については Pack_Speed　クラスを参照のこと。

k） Pack_Multipath クラス

Pack_Multipath クラスのメソッドについて、以下に示す。

表 1-124　Pack_Multipath クラス フィールド

protected	Dtc_AcclData	**m_inAcclX** 入力前後加速度データ
protected	Dtc_CARPLSData	**m_inCARPLS** 入力車速パルスデータ
protected	Dtc_DirectionData	**m_inDir** 入力方位データ
protected	Dtc_GpsData	**m_inGps** 入力GPSデータ
protected	Dtc_GyroData	**m_inGyroYaw** 入力ヨーデータ
protected static	Dtc_PositionData	**m_inOldPos** 入力前回位置データ
protected	Dtc_PositionData	**m_outPos** 出力位置データ

表 1-125　annul メソッド

メソッド名	annul	クラス名	Pack_Multipath

入力

なし。

概要

結果無効化処理。

出力

なし。

表 1-126　setOldPositionData メソッド

メソッド名	setOldPositionData	クラス名	Pack_Multipath

入力

前回位置データ。

概要

前回位置データ設定。

出力

なし。

表 1-127　getOldPositionData メソッド

メソッド名	getOldPositionData	クラス名	Pack_Multipath

入力

なし。

概要

前回位置データ取得。

出力

前回位置データ。

表 1-128　setGpsData メソッド

メソッド名	setGpsData	クラス名	Pack_Multipath

入力

GPS データ。

概要

GPS データ設定。

出力

なし。

表 1-129　getGpsData メソッド

メソッド名	getGpsData	クラス名	Pack_Multipath

入力

なし。

概要

GPS データ取得。

出力

GPS データ。

表 1-130　setYawData メソッド

メソッド名	setYawData	クラス名	Pack_Multipath

入力

方位角速度データ。

概要

入力ヨーデータ設定。

出力

なし。

表 1-131　getYawData メソッド

メソッド名	getYawData	クラス名	Pack_Multipath

入力

なし。

概要

入力ヨーデータ取得。

出力

方位角速度データ。

表 1-132　setDirectionData メソッド

メソッド名	setDirectionData	クラス名	Pack_Multipath

入力

方位角データ。

概要

入力方位データ設定。

出力

なし。

表 1-133　getDirectionData メソッド

メソッド名	getDirectionData	クラス名	Pack_Multipath

入力

なし。

概要

入力方位データ取得。

出力

方位角データ。

表 1-134　setAcclXData メソッド

メソッド名	setAcclXData	クラス名	Pack_Multipath

入力

前後加速度データ。

概要

入力前後加速度データ設定。

出力

なし。

表 1-135　getAcclXData メソッド

メソッド名	getAcclXData	クラス名	Pack_Multipath

入力

なし。

概要

入力方位データ取得。

出力

方位角データ。

表 1-136　setCARPLSData メソッド

メソッド名	setCARPLSData	クラス名	Pack_Multipath

入力

車速センサデータ。

概要

入力車速パルスデータ設定。

出力

なし。

表 1-137　getCARPLSData メソッド

メソッド名	getCARPLSData	クラス名	Pack_Multipath

入力

なし。

概要

入力車速パルスデータ取得。

出力

車速センサデータ。

表 1-138　calculate メソッド

メソッド名	calculate	クラス名	Pack_Multipath

入力

なし。

概要

マルチパス補正処理。

出力

なし。

表 1-139　getPositionResult メソッド

メソッド名	getPositionResult	クラス名	Pack_Multipath

入力

なし。

概要

結果位置データ取得。

出力

位置データ。

l） Pack_DllMultipath クラス

Pack_DllMultipath クラスは Pack_Multipath クラスの継承であるため、メソッド概要については Pack_Multipath クラスを参照のこと。

m） Pack_Position クラス

Pack_Position クラスのメソッドについて、以下に示す。

表 1-140　Pack_Position クラス フィールド

protected Dtc_AcclData	**m_inAcclX**	入力前後加速度データ
protected Dtc_AcclData	**m_inAcclY**	入力左右加速度データ
protected Dtc_AcclData	**m_inAcclZ**	入力上下加速度データ
protected Dtc_CARPLSData	**m_inCARPLS**	入力車速パルスデータ
protected Dtc_GyroData	**m_inGyroPitch**	入力ピッチデータ
protected Dtc_GyroData	**m_inGyroRoll**	入力ロールデータ
protected Dtc_GyroData	**m_inGyroYaw**	入力方位データ
protected Dtc_AcclData	**m_outAcclX**	出力前後加速度データ
protected Dtc_GyroData	**m_outYaw**	出力方位データ

表 1-141　annul メソッド

メソッド名	annul	クラス名	Pack_Position

入力

なし。

概要

結果無効化処理。

出力

なし。

表 1-142　setRollData メソッド

メソッド名	setRollData	クラス名	Pack_Position

入力

ロール角速度データ。

概要

入力ロールデータ設定。

出力

なし。

表 1-143　getRollData メソッド

メソッド名	getRollData	クラス名	Pack_Position

入力

なし。

概要

入力ロールデータ取得。

出力

ロール角速度データ。

表 1-144　setPitchData メソッド

メソッド名	setPitchData	クラス名	Pack_Position

入力

ピッチ角速度データ。

概要

入力ピッチデータ設定。

出力

なし。

表 1-145　getPitchData メソッド

メソッド名	getPitchData	クラス名	Pack_Position

入力

なし。

概要

入力ピッチデータ取得。

出力

ピッチ角速度データ。

表 1-146　setYawData メソッド

メソッド名	setYawData	クラス名	Pack_Position

入力

方位角速度データ。

概要

入力方位データ設定。

出力

なし。

表 1-147　getYawData メソッド

メソッド名	getYawData	クラス名	Pack_Position

入力

なし。

概要

入力方位データ取得。

出力

方位角速度データ。

表 1-148　setAcclXData メソッド

メソッド名	setAcclXData	クラス名	Pack_Position

入力

前後加速度データ。

概要

入力前後加速度データ設定。

出力

なし。

表 1-149　getAcclXData メソッド

メソッド名	getAcclXData	クラス名	Pack_Position

入力

なし。

概要

結果前後加速度データ取得。

出力

前後加速度データ。

表 1-150　setAcclYData メソッド

メソッド名	setAcclYData	クラス名	Pack_Position

入力

左右加速度データ。

概要

入力左右加速度データ設定。

出力

なし。

表 1-151　getAcclZData メソッド

メソッド名	getAcclZData	クラス名	Pack_Position

入力

なし。

概要

入力上下加速度データ取得。

出力

上下加速度データ。

表 1-152　setCARPLSData メソッド

メソッド名	setCARPLSData	クラス名	Pack_Position

入力

車速パルスデータ。

概要

入力車速パルスデータ設定。

出力

なし。

表 1-153　getCARPLSData メソッド

メソッド名	getCARPLSData	クラス名	Pack_Position

入力

なし。

概要

入力車速パルスデータ取得。

出力

車速センサデータ。

表 1-154　calculate メソッド

メソッド名	calculate	クラス名	Pack_Position

入力

なし。

概要

姿勢補正計算処理。

出力

なし。

表 1-155　getYawResult メソッド

メソッド名	getYawResult	クラス名	Pack_Position

入力

なし。

概要

結果方位データ取得。

出力

方位角データ。

表 1-156　getAcclXResult メソッド

メソッド名	getAcclXResult	クラス名	Pack_Position

入力

なし。

概要

結果前後加速度データ取得。

出力

前後加速度データ。

n） Pack_Marker クラス

Pack_Marker クラスのメソッドについて、以下に示す。

Pack_Marker クラスはレーンマーカにより補正するクラスで、フィールドを表 2-215、各メソッドをからにそれぞれ示す。

表 1-157　Pack_Marker クラス フィールド

protected Dtc_MarkerData	**m_inMarkerData**　入力マーカセンサデータ
protected Dtc_MarkerElement	**m_inMarkerElement**　マーカ素子データ
protected Dtc_PositionData	**m_inPos**　入力位置データ
protected Dtc_PositionData	**m_outPos**　出力位置データ

表 1-158　annul メソッド

メソッド名	annul	クラス名	Pack_Marker

入力

なし。

概要

算出結果を無効化。

出力

なし。

表 1-159　setMarkerData メソッド

メソッド名	setMarkerData	クラス名	Pack_Marker

入力

レーンマーカデータ。

概要

レーンマーカデータを設定。

出力

なし。

表 1-160　getMarkerData メソッド

メソッド名	getMarkerData	クラス名	Pack_Marker

入力

なし。

概要

レーンマーカデータを取得。

出力

レーンマーカデータ。

表 1-161　calculate メソッド

メソッド名	calculate	クラス名	Pack_Marker

入力

なし。

概要

マーカにより位置を補正。

出力

なし。

表 1-162　getPositionResult メソッド

メソッド名	getPositionResult	クラス名	Pack_Marker

入力

なし。

概要

現在位置データを取得。

出力

現在位置データ。

o） Pack_DllMarker クラス

Pack_DllMarker クラスは Pack_Marker クラスの継承であるため、メソッド概要については Pack_Marker クラスを参照のこと。

p） Pack_PositionMain クラス

Pack_PositionMain クラスのメソッドについて、以下に示す。

表 1-163　Pack_PositionMain クラス フィールド

boolean	**m_fUseAcclX**	前後加速度センサ使用フラッグ
boolean	**m_fUseAcclY**	左右加速度センサ使用フラッグ
boolean	**m_fUseAcclZ**	上下加速度センサ使用フラッグ
boolean	**m_fUseCARPLS**	車速パルス使用フラッグ
boolean	**m_fUseGps**	GPS使用フラッグ
boolean	**m_fUseGyroPitch**	ピッチジャイロ使用フラッグ
boolean	**m_fUseGyroRoll**	ロールジャイロ使用フラッグ
boolean	**m_fUseGyroYaw**	方位ジャイロ使用フラッグ
boolean	**m_fUseMarker**	マーカー使用フラッグ
int	**m_nAngleClassType**	使用方位算出クラスタイプ
int	**m_nCycleTime**	サイクルタイム
int	**m_nDriftClassType**	使用ドリフト補正クラスタイプ
int	**m_nMarkerClassType**	使用マーカー補正クラスタイプ
int	**m_nMultipathClassType**	使用マルチパス補正クラスタイプ
int	**m_nPositionClassType**	使用姿勢補正クラスタイプ
int	**m_nPulseClassType**	使用パルススケール補正クラスタイプ
int	**m_nSpeedClassType**	使用速度算出クラスタイプ
int	**m_nStopClassType**	使用停止判定クラスタイプ
protected Dtc_PositionData	**m_OldPos**	前回位置データ
protected Dtc_PositionData	**m_outPos**	出力位置データ

表 1-164　setMarkerGisFileName メソッド

メソッド名	setMarkerGisFileName	クラス名	Pack_PositionMain

入力

マーカ GIS ファイル名。

概要

マーカ GIS ファイル名設定。

出力

なし。

表 1-165　setGpsData メソッド

メソッド名	setGpsData	クラス名	Pack_PositionMain

入力

GPS データ。

概要

GPS データ設定。

出力

なし。

表 1-166　setRollData メソッド

メソッド名	setRollData	クラス名	Pack_PositionMain

入力

ロール角速度データ。

概要

入力ロールデータ設定。

出力

なし。

表 1-167　setPitchData メソッド

メソッド名	setPitchData	クラス名	Pack_PositionMain

入力

ピッチ角速度データ。

概要

入力ピッチデータ設定。

出力

なし。

表 1-168　setYawData メソッド

メソッド名	setYawData	クラス名	Pack_PositionMain

入力

入力方位角速度データ。

概要

入力方位データ設定。

出力

なし。

表 1-169　setAcclXData メソッド

メソッド名	setAcclXData	クラス名	Pack_PositionMain

入力

前後加速度データ。

概要

入力前後加速度データ設定。

出力

なし。

表 1-170　setAcclYData メソッド

メソッド名	setAcclYData	クラス名	Pack_PositionMain
入力			
左右加速度データ。			
概要			
入力左右加速度データ設定。			
出力			
なし。			

表 1-171　setAcclZData メソッド

メソッド名	setAcclZData	クラス名	Pack_PositionMain

入力

上下加速度データ。

概要

入力上下加速度データ設定。

出力

なし。

表 1-172　setCARPLSData メソッド

メソッド名	setCARPLSData	クラス名	Pack_PositionMain

入力

車速パルスデータ。

概要

入力車速パルスデータ設定。

出力

なし。

表 1-173　setMarkerData メソッド

メソッド名	setMarkerData	クラス名	Pack_PositionMain

入力

マーカセンサデータ。

概要

入力マーカデータ設定。

出力

なし。

表 1-174　getPositionResult メソッド

メソッド名	getPositionResult	クラス名	Pack_PositionMain

入力

なし。

概要

結果位置データ取得。

出力

位置データ。

表 1-175　initialize メソッド

メソッド名	initialize	クラス名	Pack_PositionMain

入力

初期位置データ。

概要

初期化処理
センサ使用情報を設定後にこのメソッドを呼び出す必要がある。

出力

なし。

表 1-176　execute メソッド

メソッド名	execute	クラス名	Pack_PositionMain

入力

なし。

概要

位置特定処理実行。

出力

なし。

q） PackageDllIf クラス

PackageDllIf クラスのメソッドについて、以下に示す。

表 1-177　PackageDllIf　クラス フィールド

static Dtc_AcclData	**m_inAcclX**	入力前後加速度データ
static Dtc_AcclData	**m_inAcclY**	入力左右加速度データ
static Dtc_AcclData	**m_inAcclZ**	入力上下加速度データ
static Dtc_CARPLSData	**m_inCARPLS**	入力車速パルスデータ
static Dtc_GpsData	**m_inGps**	入力GPSデータ
static Dtc_GyroData	**m_inGyroPitch**	入力ピッチデータ
static Dtc_GyroData	**m_inGyroRoll**	入力ロールデータ
static Dtc_GyroData	**m_inGyroYaw**	入力方位データ
static Dtc_MarkerData	**m_inMarker**	マーカーセンサデータ
static Dtc_PositionData	**m_inOldPos**	入力前回位置データ

表 1-178　ExecPackageCore メソッド

メソッド名	ExecPackageCore	クラス名	PackageDllIf

入力

なし。

概要

位置特定パッケージ実行処理。

出力

なし。

表 1-179　getPositionResult メソッド

メソッド名	getPositionResult	クラス名	PackageDllIf

入力

なし。

概要

結果位置データ取得。

出力

位置データ。

表 1-180　InitPackageCore メソッド

メソッド名	InitPackageCore	クラス名	PackageDllIf

入力

GPS 使用フラグ、ロール角速度使用フラグ、ピッチ角速度使用フラグ、方位角速度使用フラグ、
前後加速度使用フラグ、左右加速度使用フラグ、上下加速度使用フラグ、
車速パルス使用フラグ、レーンマーカセンサ使用フラグ、
演算周期、初期緯度、初期経度、初期方位、初期データ有効フラグ

概要

位置特定パッケージ初期化処理。

出力

なし。

表 1-181　MainPackageCore メッド

メソッド名	MainPackageCore	クラス名	PackageDllIf

入力

GPS 緯度、GPS 経度、GPS 位置遅延時間、　GPS 位置データ情報、
GPS 速度、GPS 速度遅延時間 、GPS 速度データ情報、
GPS 方位、GPS 方位遅延時間 、GPS 方位データ情報 、
測位情報、 ディファレンシャル情報 、GPS エラー情報、
ロール角速度、ロール角速度データ情報 、ロール角速度エラー情報、
ピッチ角速度、ピッチ角速度データ情報 、ピッチ角速度エラー情報、
方位角速度、方位角速度データ情報、方位角速度エラー情報、
前後加速度、前後加速度データ情報　 前後加速度エラー情報、
左右加速度、左右加速度データ情報　 左右加速度エラー情報、
上下加速度、上下加速度データ情報、上下加速度エラー情報、
車速センサ構成、車速パルス数、パルススケール車速パルス速度、
車速センサデータ情報、車速センサエラー情報、
前回緯度、前回経度、前回方位、前回位置エラー情報、

概要

位置特定パッケージメインモジュール。

出力

なし。

表 1-182　GetPackageCoreLati メソッド

メソッド名	GetPackageCoreLati	クラス名	PackageDllIf

入力

なし。

概要

緯度情報取得。

出力

緯度データ。

表 1-183　GetPackageCoreLongi メソッド

メソッド名	GetPackageCoreLongi	クラス名	PackageDllIf

入力

なし。

概要

経度情報取得。

出力

なし。

表 1-184　GetPackageCorePosBeing メソッド

メソッド名	GetPackageCorePosBeing	クラス名	PackageDllIf

入力

なし。

概要

位置データ情報取得。

出力

位置データ情報有無。

表 1-185　GetPackageCorePosError メソッド

メソッド名	GetPackageCorePosError	クラス名	PackageDllIf

入力

なし。

概要

位置エラー情報取得。

出力

位置エラー情報。

表 1-186　GetPackageCoreAngle メソッド

メソッド名	GetPackageCoreAngle	クラス名	PackageDllIf

入力

なし。

概要

方位情報取得。

出力

方位角データ。

表 1-187　GetPackageCoreAngleBeing メソッド

メソッド名	GetPackageCoreAngleBeing	クラス名	PackageDllIf

入力

なし。

概要

方位データ情報。

出力

方位情報の有無。

表 1-188　GetPackageCoreAngleError メソッド

メソッド名	GetPackageCoreAngleError	クラス名	PackageDllIf

入力

なし。

概要

方位エラー情報取得。

出力

方位エラー情報。

表 1-189　GetPackageCoreSpeed メソッド

メソッド名	GetPackageCoreSpeed	クラス名	PackageDllIf

入力

なし。

概要

速度情報取得。

出力

速度データ。

表 1-190　GetPackageCoreSpeedBeing メソッド

メソッド名	GetPackageCoreSpeedBeing	クラス名	PackageDllIf

入力

なし。

概要

速度データ情報取得。

出力

速度データの有無。

表 1-191　GetPackageCoreSpeedError メソッド

メソッド名	GetPackageCoreSpeedError	クラス名	PackageDllIf

入力

なし。

概要

速度エラー情報取得。

出力

速度エラー情報。

r） MarkerDllIf クラス

MarkerDllIf クラスはマーカ補正 DLL を管理するクラスで、各メソッドを表 2-250 から表 2-252 にそれぞれ示す。

表 1-192　ExecMarkerCore メソッド

メソッド名	ExecMarkerCore	クラス名	MarkerDllIf

入力

現在位置データ、レーンマーカデータ、検索レーンマーカデータ。

概要

マーカ補正を実行。

出力

現在位置データ。

表 1-193　GetMarkerCoreLatiDiff メソッド

メソッド名	GetMarkerCoreLatiDiff	クラス名	MarkerDllIf

入力

なし。

概要

マーカと自車両の緯度差を取得。

出力

マーカと自車両の緯度差。

表 1-194　GetMarkerCoreLongiDiff メソッド

メソッド名	GetMarkerCoreLongiDiff	クラス名	MarkerDllIf

入力

なし。

概要

マーカと自車両の経度差を取得。

出力

マーカと自車両の経度差。

s） Pack_GISMarkerData クラス

Pack_GISMarkerData クラスは Pack_Marker クラスの継承であるため、メソッド概要については Pack_Marker クラスを参照のこと。

t） Pack_GISMeshData クラス

メッシュデータの定義クラスである Pack_GISMeshData クラスのメソッドについて、以下に示す。

表 1-195：Pack_GISMeshData クラス フィールド

No.			型	フィールド	概要
1	protected	static	Vector	m_ElementsM	メッシュデータ情報リスト
2	public	static	double	m_MeshDistance	メッシュデータとの距離

表 1-196：loadFile メソッド

メソッド名	loadFile	クラス名	Pack_GISMeshData

入力

String fname メッシュテーブルファイル名

概要

メッシュテーブルを読み込む。

出力

なし。

表 1-197：count メソッド

メソッド名	count	クラス名	Pack_GISMeshData

入力

なし

概要

メッシュテーブルのデータ数を返却。

出力

メッシュテーブルのデータ数。

表 1-198：getNearMeshWithAll メソッド

メソッド名	getNearMeshWithAll	クラス名	Pack_GISMeshData

入力

Dtc_PositionData in_Pos 入力位置

概要

入力位置に最も近いメッシュデータをメッシュテーブルから検索。

出力

入力位置に最も近いメッシュデータ。

表 1-199：getNearMeshWithLane メソッド

メソッド名	getNearMeshWithLane	クラス名	Pack_GISMeshData

入力

Dtc_PositionData in_Pos 入力位置
int laneNo 車線番号
int index メッシュデータ ID

概要

入力位置に最も近いメッシュデータをメッシュテーブルから下記条件で検索。
○ 同じ車線上に存在する
○ メッシュ ID に近いものから 20 個

出力

入力位置に最も近いメッシュデータ。

u） Pack_Mesh クラス

マップマッチング演算クラスである Pack_Mesh クラスのメソッドについて、以下に示す。

表 1-200：Pack_Mesh クラス フィールド

No.			型	フィールド	概要
1	protected		Dtc_GyroData	m_yawGyro	入力方位角速度データ
2	protected		Dtc_MarkerData	m_inMarkerData	入力マーカデータ
3	protected		Dtc_PositionData	m_inPos	入力位置データ
4	protected		Dtc_MeshElement	m_MeshElement	メッシュデータ
5	protected		Dtc_PositionData	m_outPos	出力位置データ
6	protected		int	m_Count	
7	protected		int	m_Config	マッチング状態
8	protected		int	m_LaneNo	車線番号
9	protected		int	m_Index	メッシュデータ行番号
10	protected		int	m_diffLati	緯度差
11	protected		int	m_diffLongi	経度差
12	protected		int	m_diffDirect	方位差
13	protected		int	m_LaneChange	車線変更検出フラグ
14	protected		int	m_LaneCount	車線変更検出カウンタ
15	protected		int	m_LaneThre	車線変更閾値
16	protected		int	m_Interval	メッシュデータ検索周期

表 1-201：annul メソッド

メソッド名	annul	クラス名	Pack_Mesh

入力

なし

概要

結果を無効化する処理

出力

なし

表 1-202：setYawData メソッド

メソッド名	setYawData	クラス名	Pack_Mesh

入力

Dtc_GyroData data 方位データ

概要

入力方位データを保存する。

出力

なし。

表 1-203：setYawData メソッド

メソッド名	setYawData	クラス名	Pack_Mesh

入力

Dtc_GyroData data 方位データ

概要

入力マーカデータを保存する。

出力

なし。

表 1-204：setPositionData メソッド

メソッド名	setPositionData	クラス名	Pack_Mesh

入力

Dtc_PositionData data 入力位置データ

概要

入力位置データを保存する。

出力

なし。

表 1-205：getPositionData メソッド

メソッド名	getPositionData	クラス名	Pack_Mesh

入力

なし

概要

入力位置データを返却する。

出力

入力位置データ。

表 1-206：getMeshElement メソッド

メソッド名	getMeshElement	クラス名	Pack_Mesh

入力

なし

概要

入力メッシュデータを返却する。

出力

入力メッシュデータ。

表 1-207：setMeshInterval メソッド

メソッド名	setMeshInterval	クラス名	Pack_Mesh

入力

int interval メッシュデータ検索周期

概要

メッシュデータ検索周期を設定する。

出力

なし

表 1-208：setMeshThreshold メソッド

メソッド名	setMeshThreshold	クラス名	Pack_Mesh

入力

int threshold 車線変更閾値

概要

車線変更閾値を設定する。

出力

なし

表 1-209：getPositionResult メソッド

メソッド名	getPositionResult	クラス名	Pack_Mesh

入力

なし

概要

出力位置データを返却する。

出力

出力位置データ。

v） Pack_MeshA クラス

マップマッチングを何もしない Pack_MeshA クラスは、マップマッチング演算クラスである Pack_Mesh クラスを継承する。このクラスのメソッドについて、以下に示す。

表 1-210：calculate メソッド

メソッド名	calculate	クラス名	Pack_MeshA

入力

なし。

概要

入力位置データをそのまま出力位置データに格納する。

出力

なし。

w） Pack_MeshB クラス

メッシュデータから外れた場合にメッシュデータ内に収めるマップマッチングを行う Pack_MeshB クラスは、マップマッチング演算クラスである Pack_Mesh クラスを継承する。このクラスのメソッドについて、以下に示す。

表 1-211：calculate メソッド

メソッド名	calculate	クラス名	Pack_MeshB

入力

なし。

概要

入力位置データがメッシュデータの外にある場合は、メッシュデータの境界点を出力位置データとして格納する。
メッシュデータの内にある場合は、入力位置データをそのまま出力位置データに格納する。

出力

なし。

x） Pack_MeshC クラス

車線の中央に収めるマップマッチングを行う Pack_MeshC クラスは、マップマッチング演算クラスである Pack_Mesh クラスを継承する。このクラスのメソッドについて、以下に示す。

表 1-212：calculate メソッド

メソッド名	calculate	クラス名	Pack_MeshC

入力

なし。

概要

入力位置データを車線の中央に移動した点を出力位置データとして格納する。

出力

なし。

2. 位置特定パッケージのプロトタイプ製作およびテスト

位置特定パッケージの実装設計をもとに、プロトタイプを製作しテストを行った。テスト結果を以下に示す。

表 2-1 位置特定パッケージ単体テスト（1/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
1.1	Dtc_GpsData (GPSクラス)	簡易デバッガを作成し、 入力に応じた出力を確認する。		getSpeedM16pS()メソッドを呼び出すと、 速度が2-16[m/sec]単位で取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				setSpeedM16pS()メソッドを呼び出すと、 速度が2-16[m/sec]単位で設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
		実験システムをデバッガとして、 右記の項目を実行する。		ケーブル切断などの通信エラーが発生すると、 エラーフラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			位置	更新タイミングで受信データがある場合、 緯度・経度が360×2-32[deg]単位でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				更新タイミングで受信データがない場合、 位置有効フラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			方位	更新タイミングで受信データがある場合、 方位が360×2-32[deg]単位でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				更新タイミングで受信データがない場合、 方位有効フラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			速度	更新タイミングで受信データがある場合、 速度が0.01[km/h]単位でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				更新タイミングで受信データがない場合、 速度有効フラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
1.2	Dtc_GyroData (ジャイロクラス)	実験システムをデバッガとして、 右記の項目を実行する。	全体	ケーブル切断などの通信エラーが発生すると、 ロール・ピッチ・方位の各エラーフラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			ロール角速度	更新タイミングで受信データがある場合、 ロール角速度が360×2-32[deg/sec]単位でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				更新タイミングで受信データがない場合、 ロール有効フラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			ピッチ角速度	更新タイミングで受信データがある場合、 ピッチ角速度が360×2-32[deg/sec]単位でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				更新タイミングで受信データがない場合、 ピッチ有効フラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			方位角速度	更新タイミングで受信データがある場合、 方位角速度が360×2-33[deg/sec]単位でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				更新タイミングで受信データがない場合、 方位有効フラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	

表 2-2　位置特定パッケージ単体テスト（2/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
1.3	Dtc_AcclData （加速度計クラス）	簡易デバッガを作成し、 入力に応じた出力を確認する。		getMKSAcceleration()メソッドを呼び出すと、 加速度が0.1×2-16[m/sec2]単位で取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				setMKSAcceleration()メソッドを呼び出すと、 加速度が0.1×2-16[m/sec2]単位で設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
		実験システムをデバッガとして、 右記の項目を実行する。	全体	ケーブル切断などの通信エラーが発生すると、 前後・左右・上下の各エラーフラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			前後	更新タイミングで受信データがある場合、 前後加速度が2-16[G]単位でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				更新タイミングで受信データがない場合、 前後有効フラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			左右	更新タイミングで受信データがある場合、 左右加速度が2-16[G]単位でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				更新タイミングで受信データがない場合、 左右有効フラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			上下	更新タイミングで受信データがある場合、 上下加速度が2-16[G]単位でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				更新タイミングで受信データがない場合、 上下有効フラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
1.4	Dtc_CARPLSData （車速クラス）	実験システムをデバッガとして、 右記の項目を実行する。		ケーブル切断などの通信エラーが発生すると、 エラーフラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				更新タイミングで受信データがある場合、 パルス数が2-16[cnt]でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	
1.5	Dtc_MarkerData （マーカクラス）	実験システムをデバッガとして、 右記の項目を実行する。		更新タイミングで受信データがある場合、 種別および横距離、縦距離が0.001[m]単位でセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	

表 2-3　位置特定パッケージ単体テスト（3/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
1.6	Dtc_PositionData (位置クラス)	簡易デバッガを作成し、 入力に応じた出力を確認する。	緯度	getLatitude()メソッドを呼び出すと、 緯度が1[deg]単位で取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				setLatitude()メソッドを呼び出すと、 緯度が1[deg]単位で設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			経度	getLongitude()メソッドを呼び出すと、 経度が1[deg]単位で取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				setLongitude()メソッドを呼び出すと、 経度が1[deg]単位で設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			方位	getDirection()メソッドを呼び出すと、 方位が1[deg]単位で取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				setDirection()メソッドを呼び出すと、 方位が1[deg]単位で設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			速度	getSpeedM16pS()メソッドを呼び出すと、 速度が2-16[m/sec]単位で取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				setSpeedM16pS()メソッドを呼び出すと、 速度が2-16[m/sec]単位で設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
		実験システムをデバッガとして、 右記の項目を実行する。		GPS、ジャイロ、加速度計、車速で通信エラーが発生すると、 エラーフラグにfalseがセットされることを、 出力データで確認する。	合格	2003/3/27	藤澤	

表 2-4　位置特定パッケージ単体テスト（4/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
2.1	Pack_Pulse （パルススケール補正クラス）	実験システムをデバッガとして、 右記の項目を実行する。	GPS	setGpsData()メソッドを呼び出すと、 最新のGPSデータが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			車速	setCARPLSData()メソッドを呼び出すと、 最新の車速データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			車速	getCARPLSDResult()メソッドを呼び出すと、 パルススケール補正後の車速データが取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
2.2	Pack_DllPulse （パルススケール補正サブクラスA）			2.1項と同様の結果が得られる。	合格	2003/3/27	藤澤	
2.3	Pack_NilimPulse （パルススケール補正サブクラスB）			2.1項と同様の結果が得られる。	合格	2003/3/27	藤澤	

表 2-5　位置特定パッケージ単体テスト（5/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
3.1	Pack_Stop (停止判定 クラス)	実験システムをデバッガとして、 右記の項目を実行する。	GPS	setGpsData()メソッドを呼び出すと、 最新のGPSデータが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			ロール角速度	setRollData()メソッドを呼び出すと、 最新のロール角速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			ピッチ角速度	setPitchData()メソッドを呼び出すと、 最新のピッチ角速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			方位角速度	setYawData()メソッドを呼び出すと、 最新の方位角速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			前後	setAcclXData()メソッドを呼び出すと、 最新の前後加速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			左右	setAcclYData()メソッドを呼び出すと、 最新の左右加速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			上下	setAcclZData()メソッドを呼び出すと、 最新の上下加速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			車速	setCARPLSData()メソッドを呼び出すと、 最新の車速データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			停止	getStopResult()メソッドを呼び出すと、 停止判定データが取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
3.2	Pack_DllStop (停止判定 サブクラスA)			3.1項と同様の結果が得られる。	合格	2003/3/27	藤澤	
3.3	Pack_NilimStop (停止判定 サブクラスB)			3.1項と同様の結果が得られる。	合格	2003/3/27	藤澤	

表 2-6　位置特定パッケージ単体テスト（6/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
4.1	Pack_Drift (ドリフト補正 クラス)	実験システムをデバッガとして、 右記の項目を実行する。	停止	setStopData()メソッドを呼び出すと、 最新の停止判定データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			ロール角速度	setRollData()メソッドを呼び出すと、 最新のロール角速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				getRollResult()メソッドを呼び出すと、 ドリフト補正後のロール角速度データが取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			ピッチ角速度	setPitchData()メソッドを呼び出すと、 最新のピッチ角速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				getPitchResult()メソッドを呼び出すと、 ドリフト補正後のピッチ角速度データが取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			方位角速度	setYawData()メソッドを呼び出すと、 最新の方位角速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				getYawResult()メソッドを呼び出すと、 ドリフト補正後の方位角速度データが取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
4.2	Pack_DllDrift (ドリフト補正 サブクラスA)			4.1項と同様の結果が得られる。	合格	2003/3/27	藤澤	
4.3	Pack_NilimDrift (ドリフト補正 サブクラスB)			4.1項と同様の結果が得られる。	合格	2003/3/27	藤澤	

表 2-7　位置特定パッケージ単体テスト（7/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
5.1	Pack_Position (姿勢補正 クラス)	実験システムをデバッガとして、 右記の項目を実行する。	ロール角速度	setRollData()メソッドを呼び出すと、 最新のロール角速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			ピッチ角速度	setPitchData()メソッドを呼び出すと、 最新のピッチ角速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			方位角速度	setYawData()メソッドを呼び出すと、 最新の方位角速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				getYawResult()メソッドを呼び出すと、 姿勢補正後の方位角速度データが取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			前後	setAcclXData()メソッドを呼び出すと、 最新の前後加速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
				getAcclXResult()メソッドを呼び出すと、 姿勢補正後の前後加速度データが取得されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			左右	setAcclYData()メソッドを呼び出すと、 最新の左右加速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			上下	setAcclZData()メソッドを呼び出すと、 最新の上下加速度データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
			車速	setCARPLSData()メソッドを呼び出すと、 最新の車速データが設定されることを、 出力データで確認する。	合格	2003/3/27	藤澤	
5.2	Pack_DllPosition (姿勢補正 サブクラスA)			5.1項と同様の結果が得られる。	合格	2003/3/27	藤澤	
5.3	Pack_NilimPosition (姿勢補正 サブクラスB)			5.1項と同様の結果が得られる。	合格	2003/3/27	藤澤	

表 2-8　位置特定パッケージ単体テスト（8/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
6.1	Pack_Angle (方位算出 クラス)		GPS	setGpsData()メソッドを呼び出すと、 最新のGPSデータが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			方位角速度	setYawData()メソッドを呼び出すと、 最新の方位角速度データが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			現在方位	getDirectionResult()メソッドを呼び出すと、 方位算出後の現在方位データが取得されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
6.2	Pack_DllAngle (方位算出 サブクラスA)			6.1項と同様の結果が得られる。	合格	2003/3/24	藤澤	
6.3	Pack_NilimAngle (方位算出 サブクラスB)			6.1項と同様の結果が得られる。	合格	2003/3/24	藤澤	

表 2-9　位置特定パッケージ単体テスト（9/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
7.1	Pack_Speed （速度算出 クラス）	実験システムをデバッガとして、 右記の項目を実行する。	GPS	setGpsData()メソッドを呼び出すと、 最新のGPSデータが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			前後加速度	setAcclXData()メソッドを呼び出すと、 最新の前後加速度データが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			現在方位	setDirectionData()メソッドを呼び出すと、 最新の現在方位データが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			現在速度	getCARPLSResult()メソッドを呼び出すと、 速度算出後の現在速度データが取得されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
7.2	Pack_DllSpeed （速度算出 サブクラスA）			7.1項と同様の結果が得られる。	合格	2003/3/24	藤澤	
7.3	Pack_NilimSpeed （速度算出 サブクラスB）			7.1項と同様の結果が得られる。	合格	2003/3/24	藤澤	

表 2-10　位置特定パッケージ単体テスト（10/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
8.1	Pack_Multipath （マルチパス補正 クラス）	実験システムをデバッガとして、 右記の項目を実行する。	GPS	setGpsData()メソッドを呼び出すと、 最新のGPSデータが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			方位角速度	setYawData()メソッドを呼び出すと、 最新の方位角速度データが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			前後加速度	setAcclXData()メソッドを呼び出すと、 最新の前後加速度データが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			現在方位	setDirectionData()メソッドを呼び出すと、 最新の現在方位データが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			現在速度	setCARPLSData()メソッドを呼び出すと、 最新の現在方位データが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			現在位置	getPositionResult()メソッドを呼び出すと、 マルチパス補正後の現在位置データが取得されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
8.2	Pack_DllMultipath （マルチパス補正 サブクラスA）			8.1項と同様の結果が得られる。	合格	2003/3/24	藤澤	
8.3	Pack_NilimMultipath （マルチパス補正 サブクラスB）			8.1項と同様の結果が得られる。	合格	2003/3/24	藤澤	

表 2-11　位置特定パッケージ単体テスト（11/11）

番号	クラス名	試験方法		試験項目	合否	実施日	実施者	備考
9.1	Pack_Marker (マーカ補正 クラス)		マーカ	setMarkerData()メソッドを呼び出すと、 最新のマーカデータが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
			現在位置	setPositionData()メソッドを呼び出すと、 最新の現在位置データが設定されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
				getPositionResult()メソッドを呼び出すと、 マーカ補正後の現在位置データが取得されることを、 出力データで確認する。	合格	2003/3/24	藤澤	
9.2	Pack_DllMarker (マーカ補正 サブクラスA)			9.1項と同様の結果が得られる。	合格	2003/3/24	藤澤	
9.3	Pack_NilimMarker (マーカ補正 サブクラスB)			9.1項と同様の結果が得られる。	合格	2003/3/24	藤澤	

別添資料２　凍結防止剤散布作業支援システムの開発

1. 基礎調査

1.1. 凍結防止剤散布作業の実態およびシステム要望に関する調査

(1) 調査概要

○ 日　時：平成 15 年 9 月 17 日

- 10:30-12:00　散布経路の現地確認（車中からの確認）
- 13:00-15:00　散布作業の実態に関するヒアリング
- 15:00-17:00　散布作業支援システムに関するヒアリング

○ 場　所：北陸地方整備局　長岡国道事務所

○ 出席者：

- 国土技術政策総合研究所　：関本研究官，都鳥研究官
- 長岡国道事務所長岡国道維持出張所　：池田係長
- 福田道路株式会社（除雪作業委託業者）：新保主任
- 株式会社　長大　：桂木，石濱

(2) ヒアリング結果

■ 散布作業の実態に関するヒアリング

- 宮本工区で実施している散布作業については、大きく次の 3 つについて実施している。

○ 凍結防止のための散布

○ 降雪中の散布（新雪融解や凍結融解。降雪量 2cm で出動。5cm 以上ではグレーダが出動するため、散布しない）

○ グレーダ除雪後の散布（わだち部の雪を溶かす）

- 散布作業の出動状況については、次に示すとおり。

○ 昼間の散布については、ほとんど実施していない。

○ 待機判断の目安については、-2℃以下が予想される場合である。（2 月は毎日待機している）

○ 散布の回数は基本的に 1 日 2 回（19:00 と 6:00 頃）

○ 気温の低い場合や少量の降雪時（グレーダが出動しない場合）には、3:00 頃散布を実施する場合もある。

○ 気温が急激に下がった場合には、夕方散布する場合もある。

- 散布量，散布箇所については、次に示すとおり。

○ 散布量については、気温によっても変えている。概ね、次のとおり。

・ -2℃～-4℃：20g

・ -5℃以下　：30-40g

・ 排水性舗装：40g

○ 散布箇所については、基本的には予め定められた区間（橋梁，カーブ，勾配部，交差点等）と排水性舗装区間である。（後日、散布箇所に関する資料をいただく）

○ 路面状態が乾燥の場合、散布をしないが、排水性舗装区間や橋梁部については、状況に応じて散布を実施している。

○ 排水性舗装区間における散布効果については、密粒区間と比較して、孔に薬剤がたまることから、はじめは効きが良い。しかし、孔の中の水の塩分が薄まると、散布しても効果が出にくくなる場合もある。また、排水性舗装区間が圧雪になると、圧雪が排水性舗装の孔に入り込んで凍結した水分くっついていしまい、グレーダ除雪で圧雪がはがし難くくなってしまう。

○ 橋梁部については、路面状態が乾燥であっても、霧による凍結の可能性があるので、薄く散布する場合がある。

○ 特に重点的に散布する箇所については、長岡大橋である。初回出動では、工区全域の定められた区間について、一通り散布を行う。その後、警察等から長岡大橋への散布要望のあった場合や気温が低下した場合には、長岡大橋を散布するために出動する。

○ その際における長岡大橋以外の散布箇所については、路面状況および積載量等を考慮しながら、オペレータと助手が相談しながら、散布するかしないかを判断している。

- 散布作業パターンについては、次に示すとおり。

○ 散布作業のパターン化については、凍結防止の場合の散布が挙げられる。降雪中の散布については、グレーダ除雪との関係もあり、臨機応変に対応しており、パターン化が難

しいと考えられる。

- 散布操作，経路等については、次に示すとおり。

○ 散布車の速度については、50-60km／ｈ程度である。

○ 除雪 ST より柏崎方面（片側 1 車線区間）については、往路で対向車線へも同時散布を行っている。復路では、1 車線分を薄目に撒いている。

○ 片側 1 車線散布を行う場合の散布幅については、3-4m である。

○ 片側 2 車線同時散布を行う場合、走行車線を走行し、散布幅 5-6m で散布している。（追い越し車線を走行しながら散布することはない。また、散布方向については、右側散布で固定で、左側散布に切り換えることはない。）

○ 片側 2 車線で右折レーンがあり 3 車線になった場合は、散布幅 6-7m で散布している。

○ 片側 4 車線での散布については、堺交差点付近（上下ともに片側 4 車）がある。このような箇所については、2 回に分けて散布している。（1 回目：一番左側の車線を走行しながら、走行している車線と右隣の車線を同時散布。2 回目：右から 2 番目の車線を走行しながら、走行している車線と右隣の車線を同時散布。）

○ 片側 2 車線の走行車線より左にある左折専用レーンについては、基本的に散布しない。

○ 喜多 IC 橋ランプ部については、消雪パイプの稼動状況を見て、散布の必要性判断をしている。

○ この部分の走行経路については、2 回走行しており、基本的に、1 回目の走行では、本線の高架橋を通らずにランプ部を降りてから、交差路を直進，ランプ部通過，本線と合流，途中でUターンしてもどり、2 回目の走行で本線の高架部を走行している。

○ しかし、消雪パイプが正常に作動している場合には、ランプ部の路面が凍結していないので、本線を走行した後にランプ部を走行する、あるいは、ランプ部を走行せずに本線のみを走行する場合もある（本線からランプ部の路面の様子もある程度見える）。

○ 散布中、駐停車車両等の障害物があった場合、特に散布操作を変更することはない。（基本的には、事故等の発生を除き、散布操作を変更するようなことはない）

○ このような障害物のある箇所については、散布されないこととなるが、付近の散布剤が通行車両により、散布されていない箇所へも散布剤が引きずられるため、障害物がなくなった後、その場所に戻り、散布するということはしていない。

- その他

○ 巡回箇所（センサ設置箇所であり、7 箇所）での作業については、走行しながら、車載気温計のデータ，目視による路面状態判別，渋滞状況について、助手が記述している。

○ 円盤回転開始タイミングについては、路上に出る前に、構内で作動させている。円盤停止タイミングについては、除雪 ST に戻る最後の区間の散布が終了次第行っている。

○ 消雪パイプの制御については、温度と降雪状況に応じて自動的に制御している。

■ 散布作業支援システムの画面に関するヒアリング

- 操作支援システム（車両側システム）については、次に示すとおり。

○ 構築するシステムについては、オペレータの負担を最小限にすることが望まれる。

○ 詳細地図の表示範囲については、自由に設定できるが、デフォルト値としては、車線が

確認できる程度で、散布車から 200m 程度先まで表示ができれば良いのではないか。

○ 提案した地図の表示地物の過不足については、次に示すとおり。
  - 待避所，バス停については不要である。
  - 交差点部と橋梁については、名称を表示してもらいたい。
  - 急勾配箇所やカーブ箇所を表示してもらいたい。

○ 道路周辺の背景地図については、住宅地図が表示されると良い。

○ 提案した表示項目以外に要望する情報については、気温や路温が挙げられる。（本年度対象外）

○ 散布箇所の音声案内については、次に示すとおり。
  - 散布操作変更予告の音声案内については、自動散布なので不要ではないか。
  - 音声案内の音については、ビープ音等で良いと考えられる。（人の声は不要では）

○ 散布計画箇所のデータについては、出動時に選択・編集したものを用いて行うが、出動後に散布計画の内容を変更できると良い。

- 計画支援システム（ステーション側システム）については、次に示すとおり。

○ 毎回、PC カードで散布履歴データを登録することは、現場の負荷がかかり、無線 LAN 等による方法で自動的にデータを吸い上げる仕組みがあると良い。

○ 散布計画データの作成については、車両側で選択するようにした方が良いのでは。

○ 地図に表示する地物の考え方，道路周辺の背景地図については、車両側システムと同様である。

○ 散布履歴の必要性については、事故発生時の警察対応で必要である。

○ 散布実施後の経過時間表示については、過去の作業履歴の表示から分かるので、経過時間表示の必要性は低いと思われる。

○ 塩分濃度を考慮した凍結時間の推定ができると良い。（高田国道事務所で実施している）

- その他

○ 助手の必要性について、目視による路面状況や道路周辺状況の確認，U ターン時等の誘導，事故車対応では、助手が必要であるが、散布車のワンマン化を考える際には、散布作業と付帯的な作業を分けて考える必要がある。

○ 散布剤の持続時間については、3-4 時間程度と考えられる。

1.2. 散布車とのデータ交換調整

範多機械製の自動散布制御装置とのデータ交換等に関する調整を行った結果を次に示す。

○ 国総研 PC と範多自動散布装置との通信は、両方向とする。

- 国総研 PC→範多自動散布装置の通信内容：**散布制御情報**

  ※散布制御情報：散布動作（散布停止／散布待機／散布），散布密度，散布幅，散布方向，湿式割合，散布薬剤種別

  ※「散布動作」について、現状の範多自動散布制御装置では「散布 ON-OFF」信号により、ベルトコンベア作動／停止とゲート開閉の制御のみ可能であった。(円盤制御はサブコントローラのスティックを手動操作）しかし、今回、国総研 PC からベルトコンベア，ゲート開閉に加えて、円盤の作動／停止も信号で制御できるようにする。なお、「散布停止」，「散布待機」，「散布」の定義については、以下に示すとおり。

  —「散布停止」：ゲート閉，円盤停止，コンベア停止

  —「散布待機」：ゲート閉，円盤作動，コンベア停止

  —「散布」とは、ゲート開，円盤作動，コンベア作動

- 範多自動散布装置→国総研 PC の通信内容：**散布結果情報，切替スイッチ状態，エラー情報**

  ※散布結果情報：散布動作（散布停止／散布待機／散布），散布密度，散布幅，散布方向，湿式割合，散布薬剤種別

○ 通信タイミングについては、以下に示すとおり。

- 国総研 PC→範多自動散布装置：任意（ただし、1 秒以上の間隔を空ける）
- 範多自動散布装置→国総研 PC：散布装置（円盤，ベルトコンベア，ゲート）の制御内容変更時（ただし、1 秒以上の間隔を空ける）

○ 範多機械から提供された詳細な仕様については、次に示すとおり。

＜散布制御装置（範多機械製）データ交換仕様＞

１．制御信号（ＲＳ２３２Ｃ）の送受信について

１－１　切り替えスイッチを「ＰＣ側入り」に選択されている時

１－１－１　ＰＣ側より送信されて来るデータを､範多自動散布装置が規定したバイト数を受信し終った時点から受信したデータを検証し約 0．1 秒後にＰＣ側へ送信します。但し、通信エラーまたはＢＣＣエラーが発生した場合は、エラー情報（1bit ）を付加して送信します。

※エラーが発生した場合は、範多自動散布装置内の散布諸元は更新されません。

１－１－２　ＰＣ側より送信されて来るデータの更新には､1 秒以上の間隔を開けて下さい。

１－２　切り替えスイッチを「ＰＣ側切り」に選択されている時

１－２－１　範多自動散布装置の散布諸元が、変更された時のみＰＣ側へ、そのデータを送信します。

１－２－２　範多自動散布装置から送信しますデータの更新は､1 秒の間隔を空けます。

１－３　切り替えスイッチを「ＰＣ側切り」から「ＰＣ側入り」に選択した時

１－３－１　範多自動散布装置の現在設定されている散布諸元を､1 秒以内にＰＣ側へ送信します。

１－３－２　現在設定されている散布諸元は､ＰＣ側からのデータ更新がない限り変わりません。

１－４　切り替えスイッチを「ＰＣ側入り」から「ＰＣ側切り」に選択した時

１－４－１　範多自動散布装置の現在設定されている散布諸元を､1 秒以内にＰＣ側へ送信します。

１－４－２　散布幅補正、散布薬剤、散布方向、散布/停止の各諸元は、範多自動散布装置の操作パネル面のスイッチ及びモノレバーの設定位置により決定されます。

１－５　範多自動散布装置からＰＣ側への送信時（１－２，１－３，１－４）にはデータ構造の「散布０Ｎ－０ＦＦ」の２bit=１にします。

１－６　切り替えスイッチについて、切り替えスイッチは、サブコントローラの操作パネル面に配置し,トグルスイッチを上側に倒すと「ＰＣ側入り」に、下側に倒すと「ＰＣ側切り」になります。

２．ＰＣ側から範多自動散布装置への送信モード（散布０Ｎ－０ＦＦ）について

２－１　散布停止（ゲート閉、散布円盤停止、コンベアー停止)。
データ構造の「散布０Ｎ－０ＦＦ」を　1 bit=０　０bit=０　にします。

２－２　散布（ゲート開、散布円盤作動、コンベアー作動)。データ構造の「散布０Ｎ－０ＦＦ」を　1 bit=１　０bit=０　にします。

２－３　散布待機（ゲート開、散布円盤作動、コンベアー停止)。データ構造の「散布０Ｎ－０ＦＦ」を　1 bit=０　０bit=１　にします。

３．フォーマットの追加変更について

３－１　「散布０Ｎ－０ＦＦ」

・散布待機を１bit めに追加しました。

・機械側動作エラーを、上位４bit に、今後の為にスペースのみを設けました今回は、とりあえずＰＣ側からの受信エラーが発生したとき４bit=１にします。

・１－５の項目追加しました。

３－２　伝送諸元のパリティビットを‘無し’から‘偶数’に変更しました。

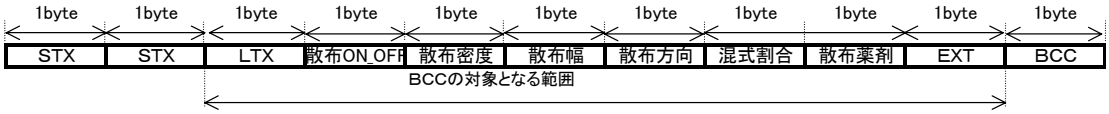
３－３　詳しくは、データ送受信フオーマットを参照して下さい。

４．ＰＣ側と範多自動散布装置を接続する通信ケーブルについて

範多自動散布装置の正面より見て左側面のコネクターから１ｍのケーブルの先端に、９ピン、Ｄｓｕｂ(メス)を取り付けておきます。

データ送受信フォーマット

1 ﾃﾞｰﾀ構造

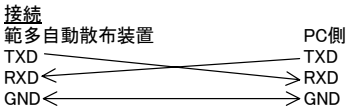


名称	内容(各1byte)
STX	先頭ﾃｷｽﾄﾃﾞｰﾀ (0xE2)
LTX	通信するﾃﾞｰﾀの長さ (0x06)
散布ON_OFF	別紙参照(データ詳細)
散布密度	(例)散布密度50g/m^2(0x32)
散布幅	(例)散布幅3.0m(0x1E)、散布幅7.0m(0x46)
散布方向	右方向(0×01)、左方向(0×00)
混式割合	今回は常に(0x00)
散布薬剤	フレーク(0x01)、粒(0x02)、粒大(0×03)、原塩(0×04)
EXT	終端ﾃｷｽﾄﾃﾞｰﾀ (0xE3)
BCC	LTXからETXまでのXOR結果

2 伝送諸元及びキャラクター構成

①同期方式　調歩同期
②通信速度　9600bps
③ｷｬﾗｸﾀ長　8ﾋﾞｯﾄ
④ｽﾄｯﾌﾟﾋﾞｯﾄ　1ﾋﾞｯﾄ
⑤ﾊﾟﾘﾃｨﾋﾞｯﾄ　偶数

3 接続



4 自動散布装置側のコネクター仕様

・DSUB(9ピン)メス。
・RXD(2番ピン)、TXD(3番ピン)、GND(5番ピン)、その他のピンは全てNC。

→「散布ON－OFF」の詳細

D7	D6	D5	D4	D3	D2	D1	D0

Bit 1	Bit 0	動作
0	0	散布停止
0	1	散布待機
1	0	散布

Bit 2	PC側への送信
0	「PC側入り」
1	「PC側切り」

Bit 7	Bit 6	Bit 5	Bit 4	エラー情報
×	×	×	0	通信、BCCエラーなし
×	×	×	1	通信、BCCエラーあり

1.3. 現地測位状況に関する調査

本業務を遂行するにあたり、実道実験の指定コースである長岡宮本工区における GPS 測位状況や FM 多重による DGPS 受信状況について事前調査を行った結果を報告する。

(1) 現地調査の概要

表 2-1 に示す概要の通りに現地調査を実施した。

表 1-1：長岡宮本工区の測位状況調査概要

No.	項目	概要	備考
1	場所	一般国道 8 号線 58.5 kp～75.7kp（上下走行車線）、 凍結防止剤散布作業時の迂回路および宮本除雪ステーション	
2	日時	2003 年 9 月 3 日(水)　AM11:20～PM3:20	
3	使用装置	DGPS 用 FM 多重レシーバ M51M タブレット PC	

a） 現地調査の結果

長岡宮本工区における測位衛星数・FM 多重受信・DOP の調査結果について、からに示す。さらに、KP 間の各受信状況をまとめた結果を、からに示す。

b） 測位衛星数の調査結果

調査結果から算出した KP 間の測位衛星数を表 2-2 示す。表から、長岡宮本工区の都市部・平野部・山間部において GPS の測位衛星数はほとんど変化しないことがわかる。

表 1-2：測位衛星数の調査結果

コース	都市部(～60.0KP)	平野部(60.0～70.0KP)	山間部(70.0KP～)	平均
下り	7.48	7.69	6.75	7.35
上り	7.32	7.38	6.83	7.19

c） DGPS(FM 多重受信)の調査結果

DGPS ステータス(ディファレンシャル情報)の値は、2=FM 多重受信、1=GPS 単独測位を表している。調査結果から算出した KP 間の FM 多重受信状況を表 2-3 に示す。表から、山間部では FM 多重受信の可能性が低く、都市部でも一部 GPS 単独測位になることがわかる。

表 1-3：FM 多重受信の調査結果

コース	都市部(～60.0KP)	平野部(60.0～70.0KP)	山間部(70.0KP～)	平均
下り	1.82	1.99	1.38	1.77
上り	1.64	1.95	1.28	1.69

d） DOP の調査結果

調査結果から算出した KP 間の DOP 値を表 2-4 に示す。表から、山間部では FM 多重受信の可能性が低いため、GPS の信頼度が低下していることがわかる。

表 1-4：DOP の調査結果

コース	都市部(～60.0KP)	平野部(60.0～70.0KP)	山間部(70.0KP～)	平均
下り	0.61	0.33	1.86	0.87
上り	0.98	0.43	1.98	1.00

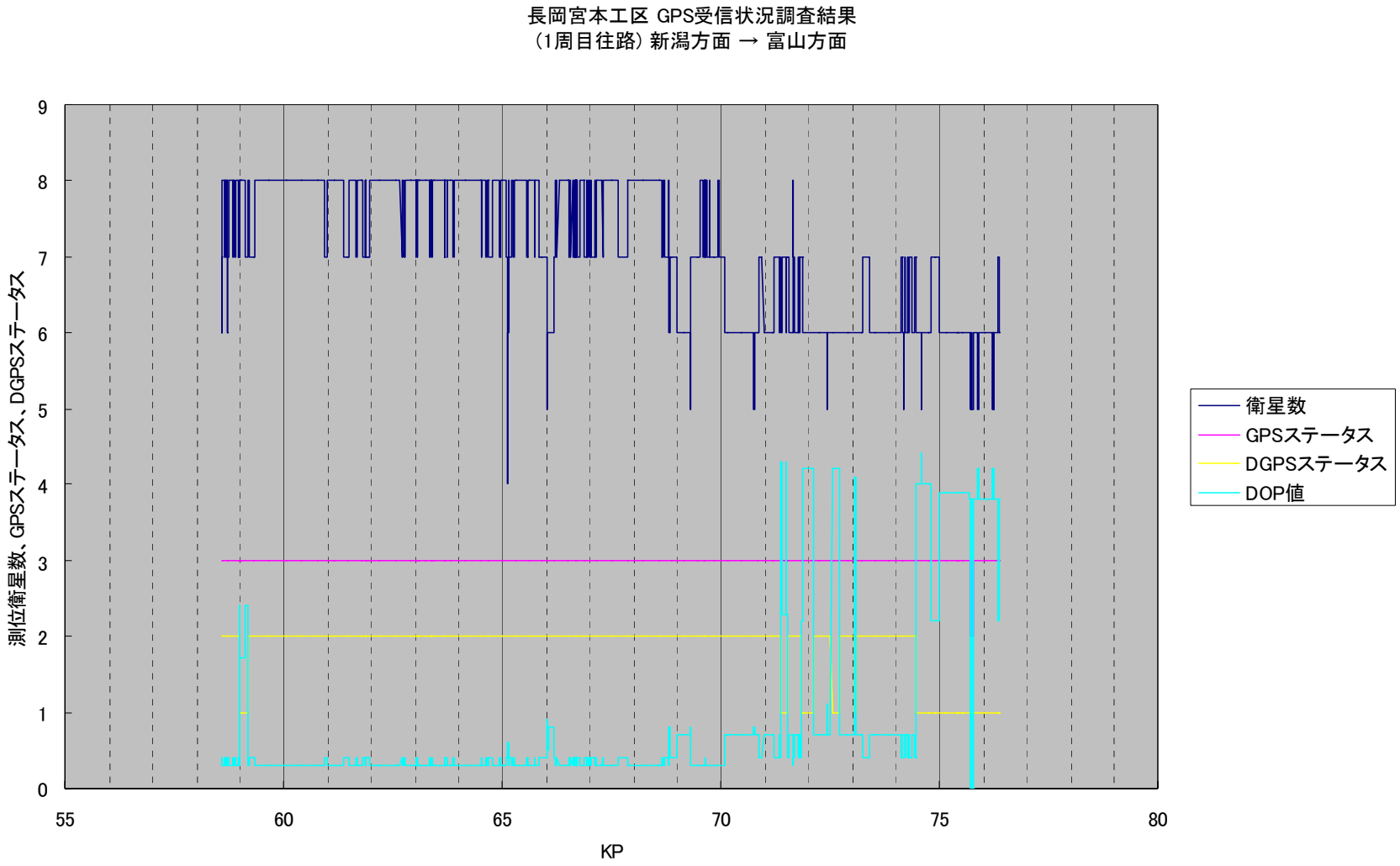


図 1-1：長岡宮本工区 GPS 受信状況調査結果 (1 周目往路)

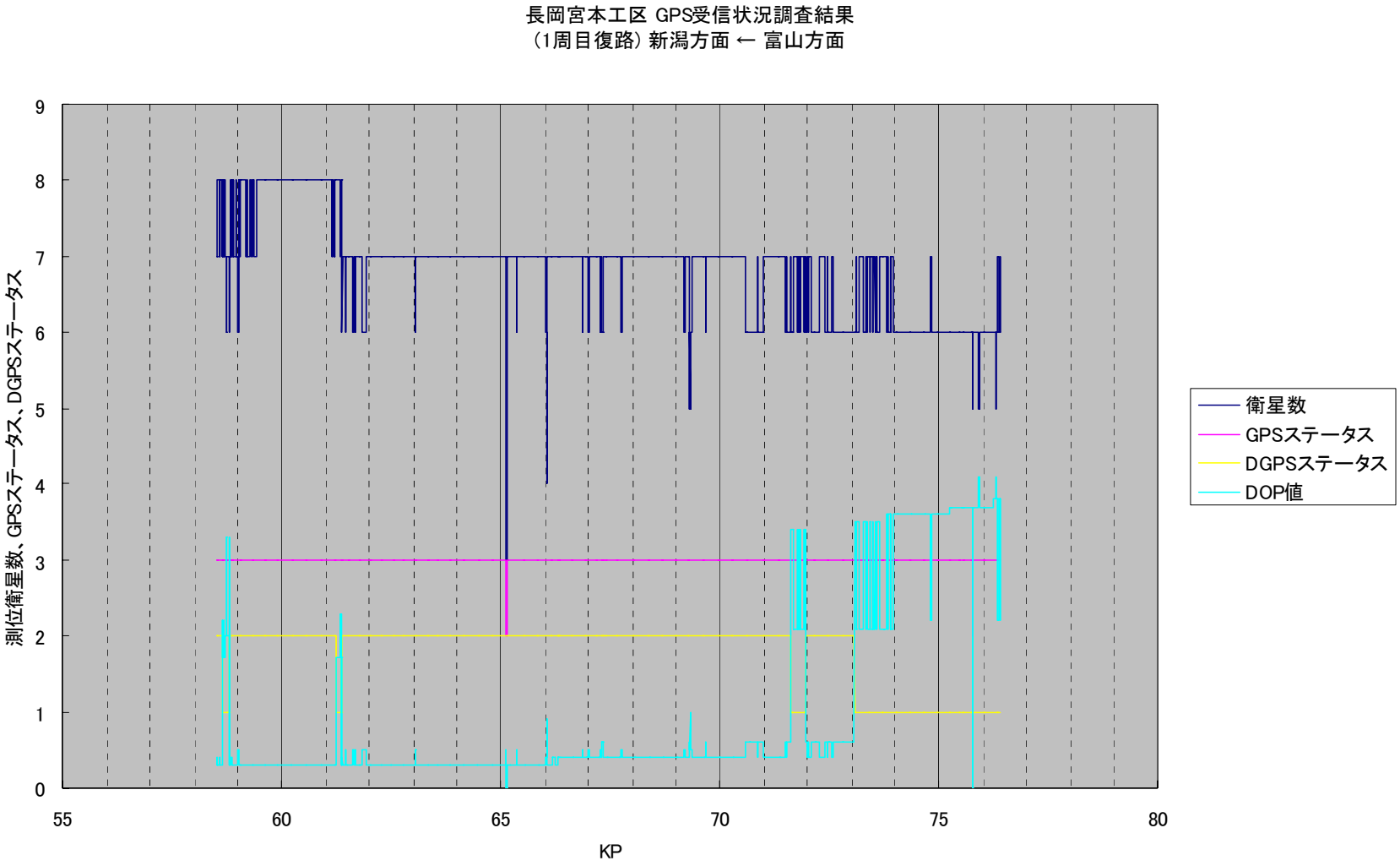


図 1-2：長岡宮本工区 GPS 受信状況調査結果 (1 周目復路)

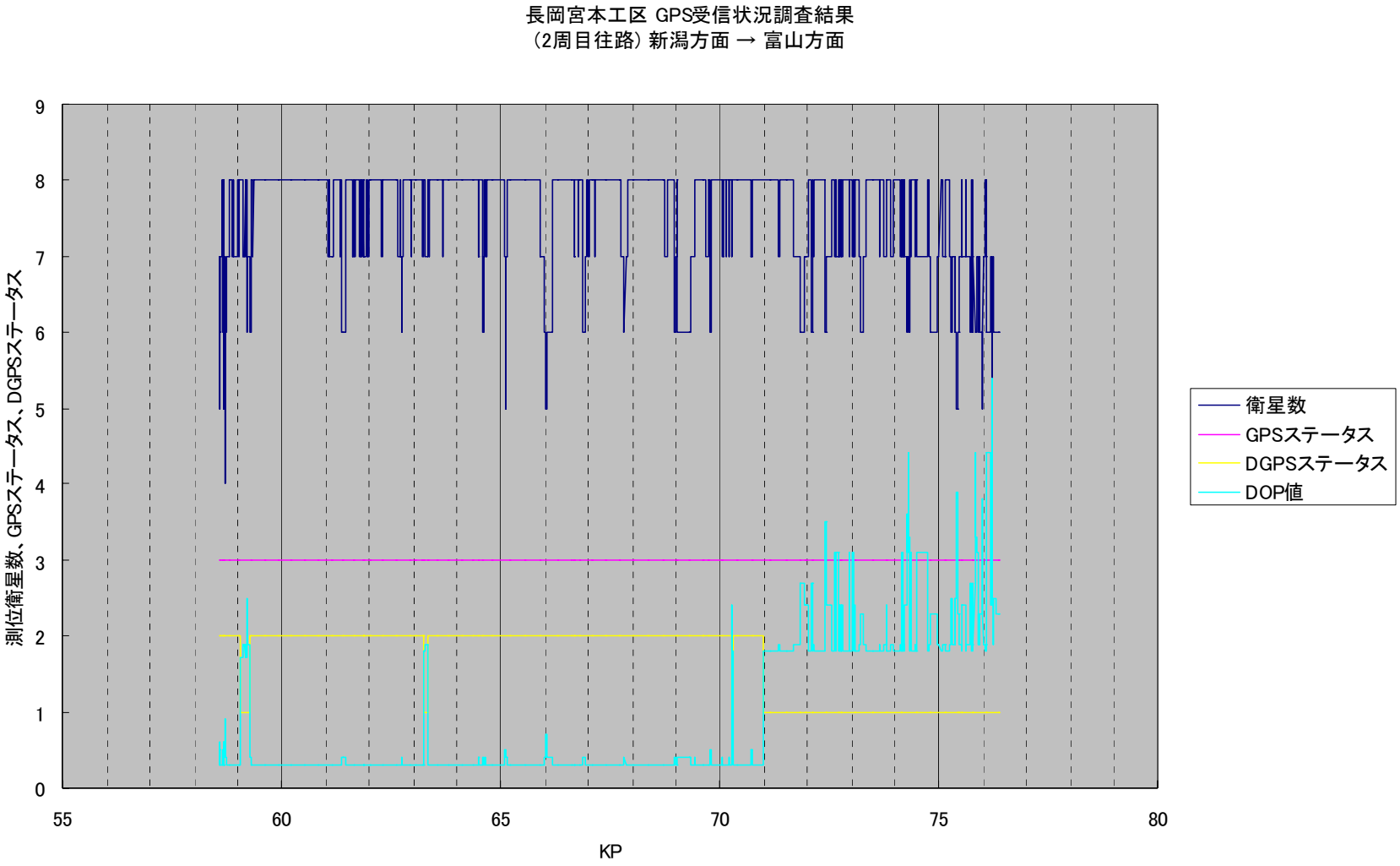


図 1-3：長岡宮本工区 GPS 受信状況調査結果 (2 周目往路)

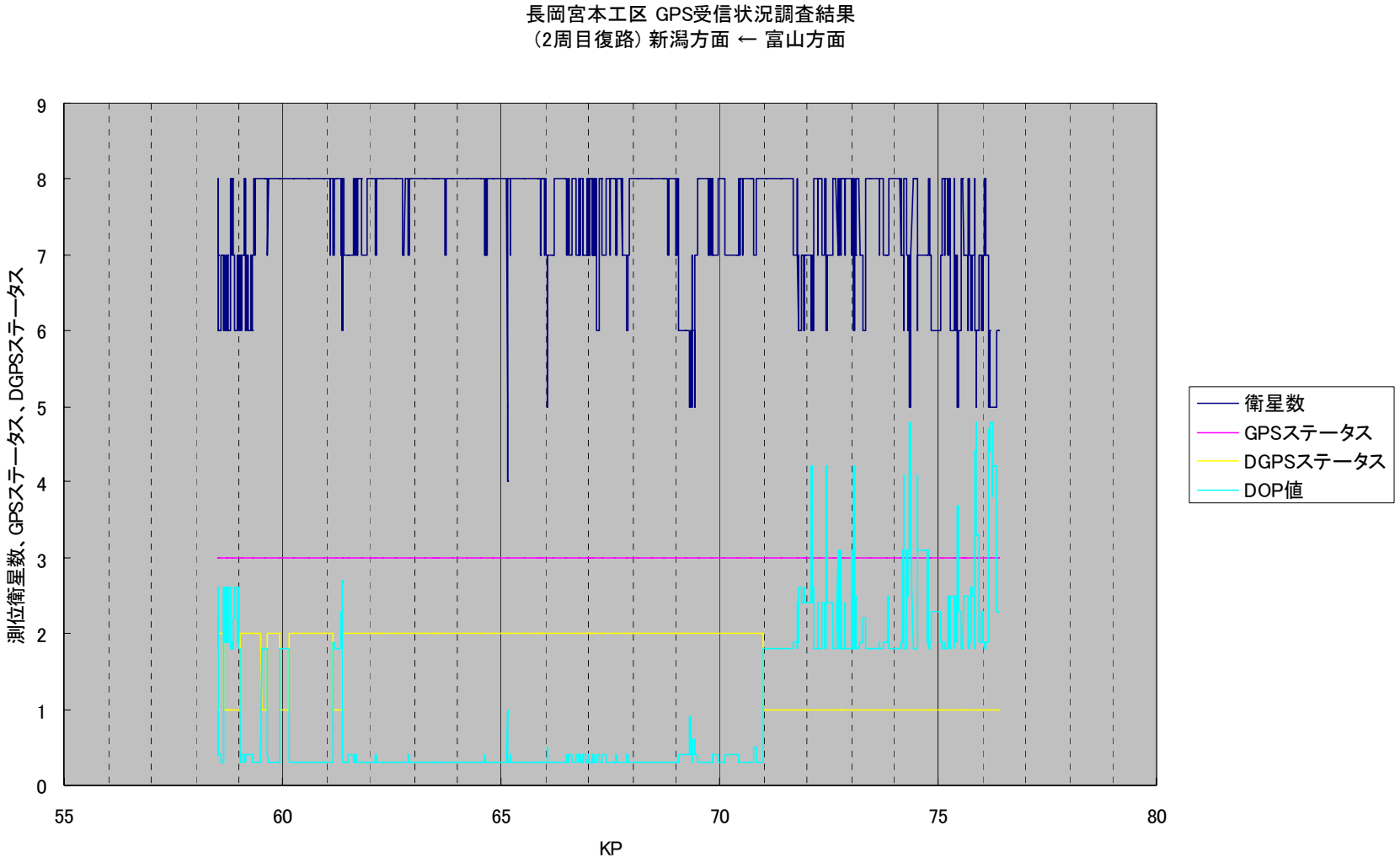


図 1-4：長岡宮本工区 GPS 受信状況調査結果 (2 周目復路)

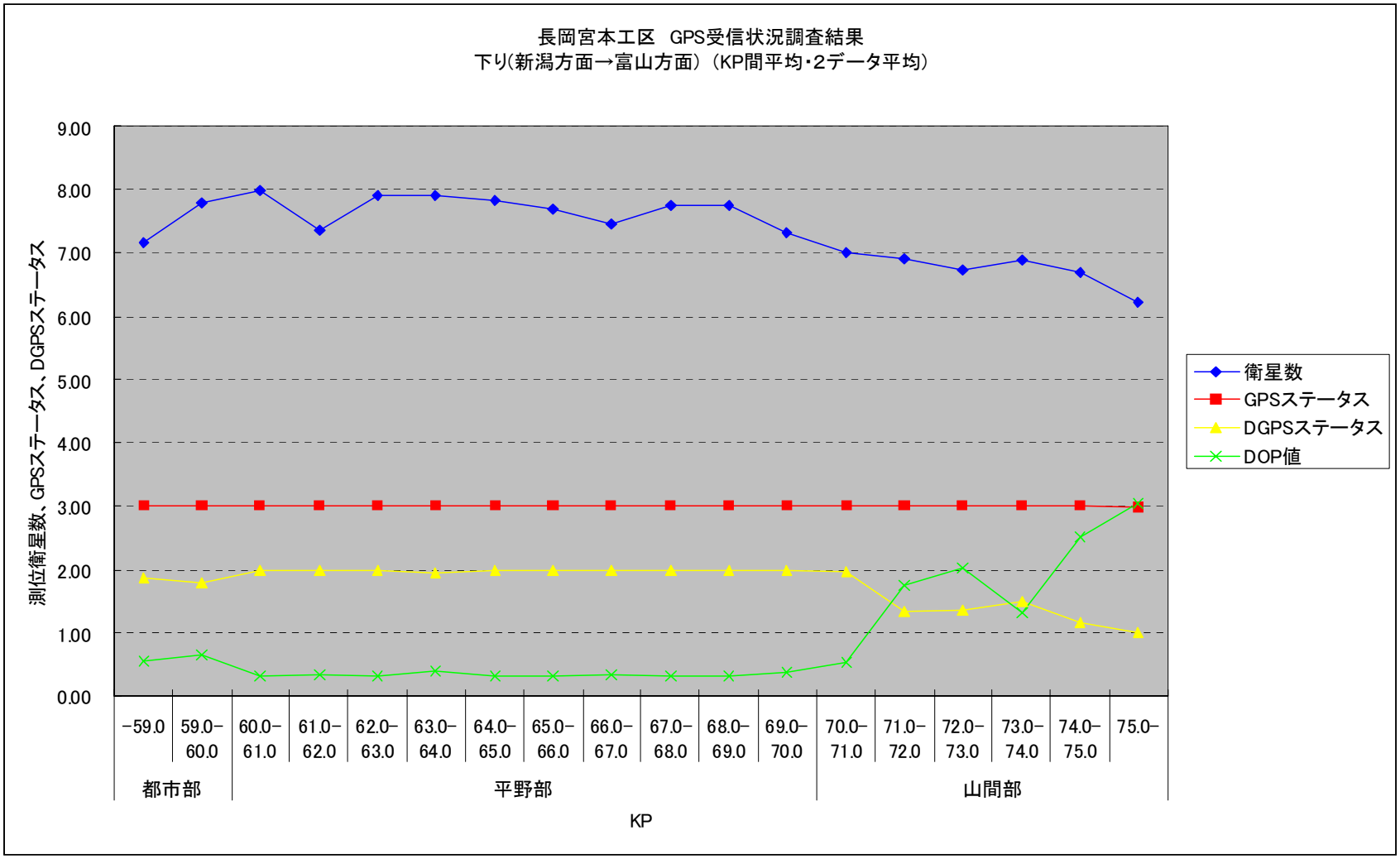


図 1-5：長岡宮本工区 GPS 受信状況調査結果　(下り･KP 間平均･２データ平均)

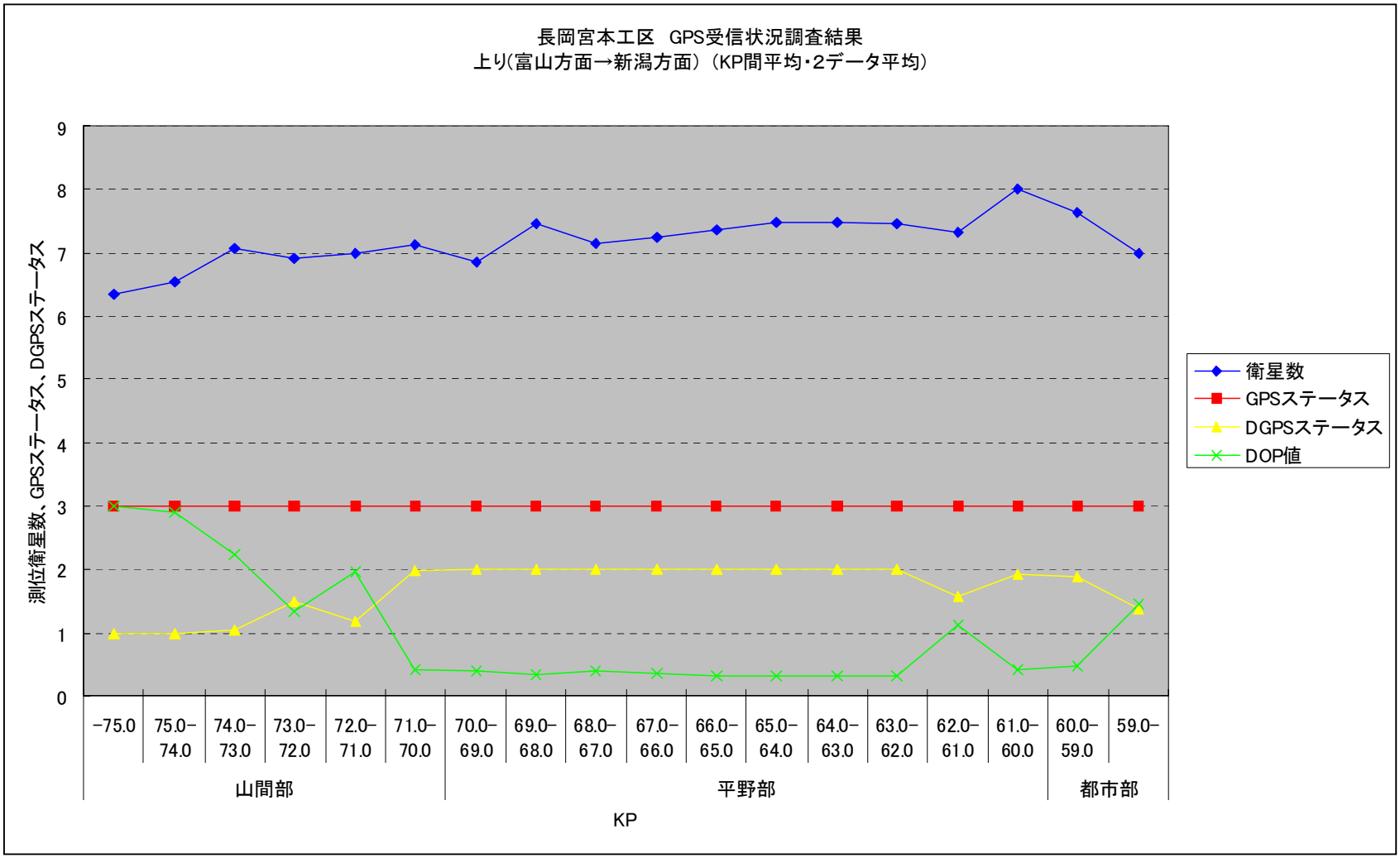


図 1-6：長岡宮本工区 GPS 受信状況調査結果　(上り･KP 間平均･２データ平均)

2. 操作支援システムの製作

2.1. 操作支援システムの基本設計

(1) 適用範囲

本節は、散布操作支援システムにおける基本設計についてのみ記載するものとする。

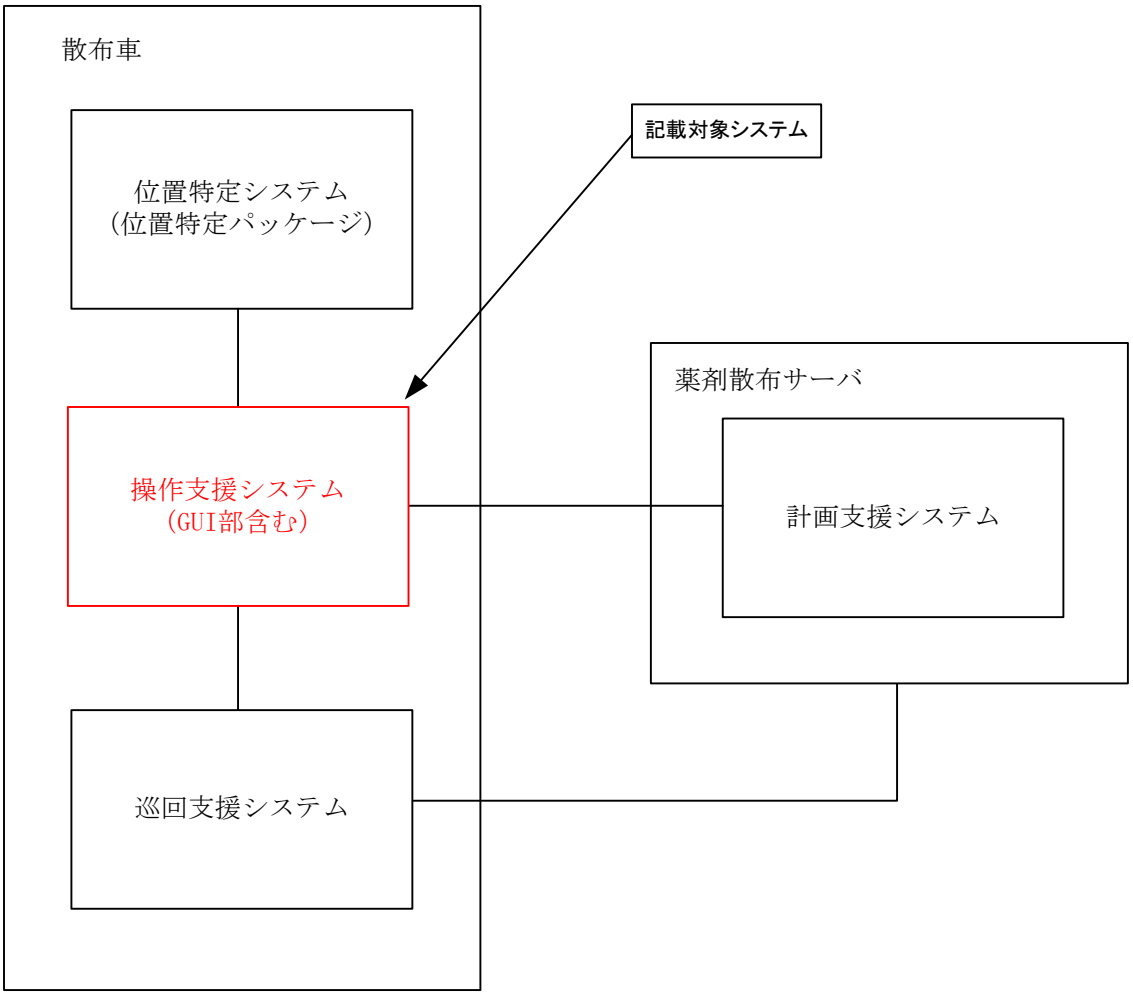


図 2-1　記載範囲

(2) 要件定義の確認

本システム全体における要件を以下のユースケース図に示す。なお、本文に記載する要件については、赤色で示されるユースケース部分である。

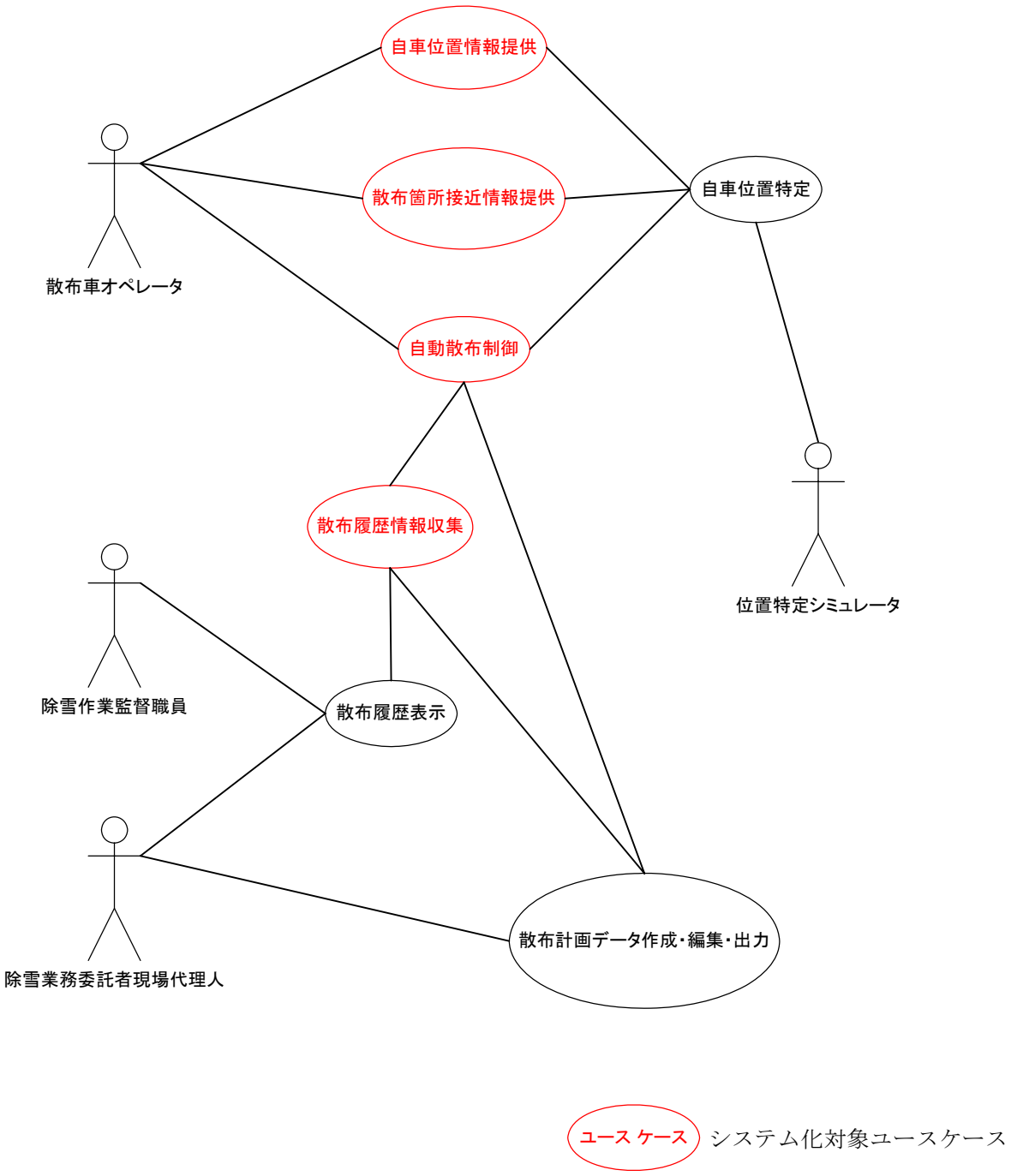


図 2-2　システム全体ユースケース図

(3)　システム全体構成

システムの全体構成および処理の流れを以下に示す。

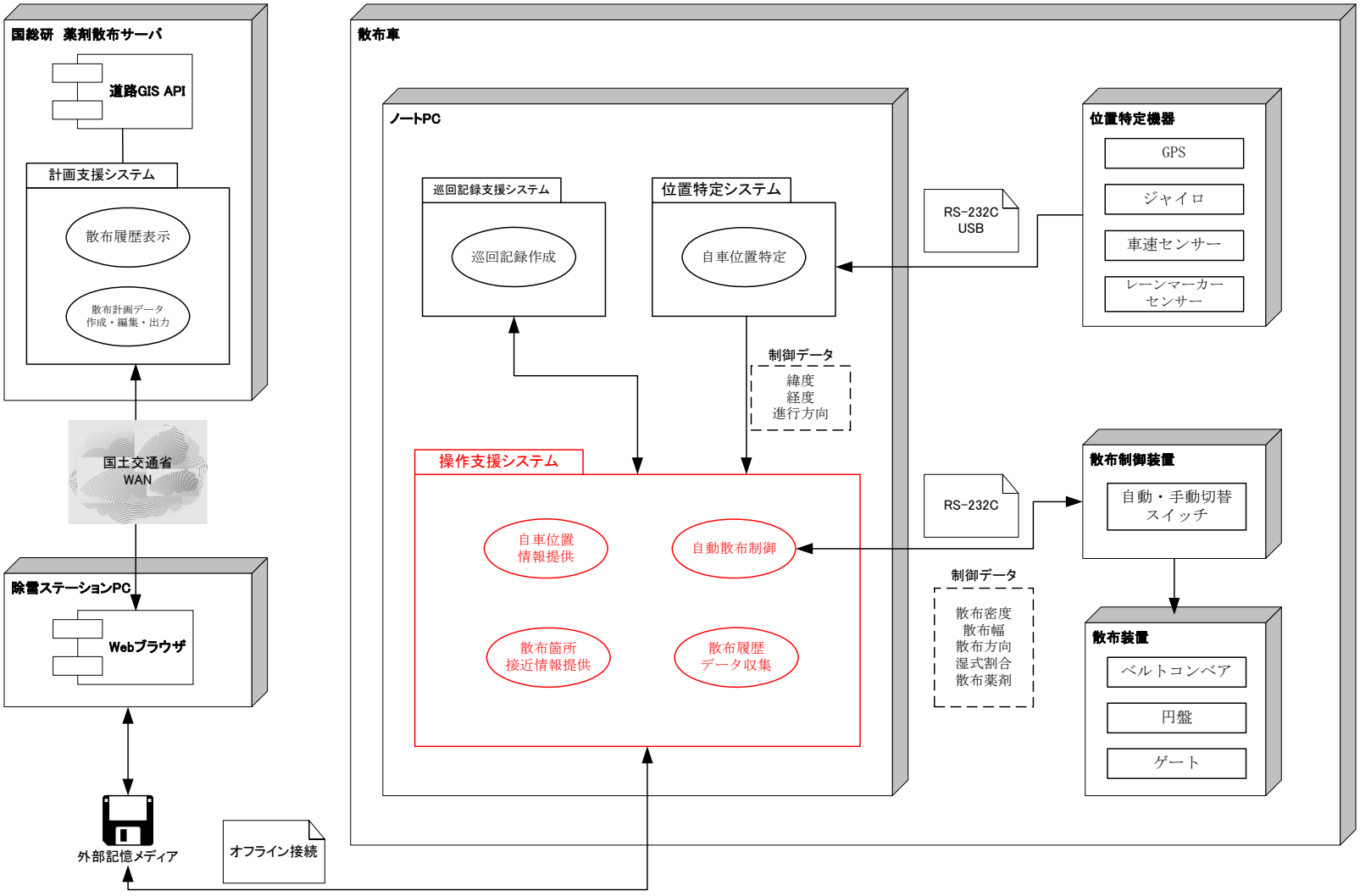


図 2-3　システム全体構成図

また、システム全体における処理の流れを以下のシーケンス図に示す。なお、図中の赤色の部分が本基本設計書に該当する箇所である。

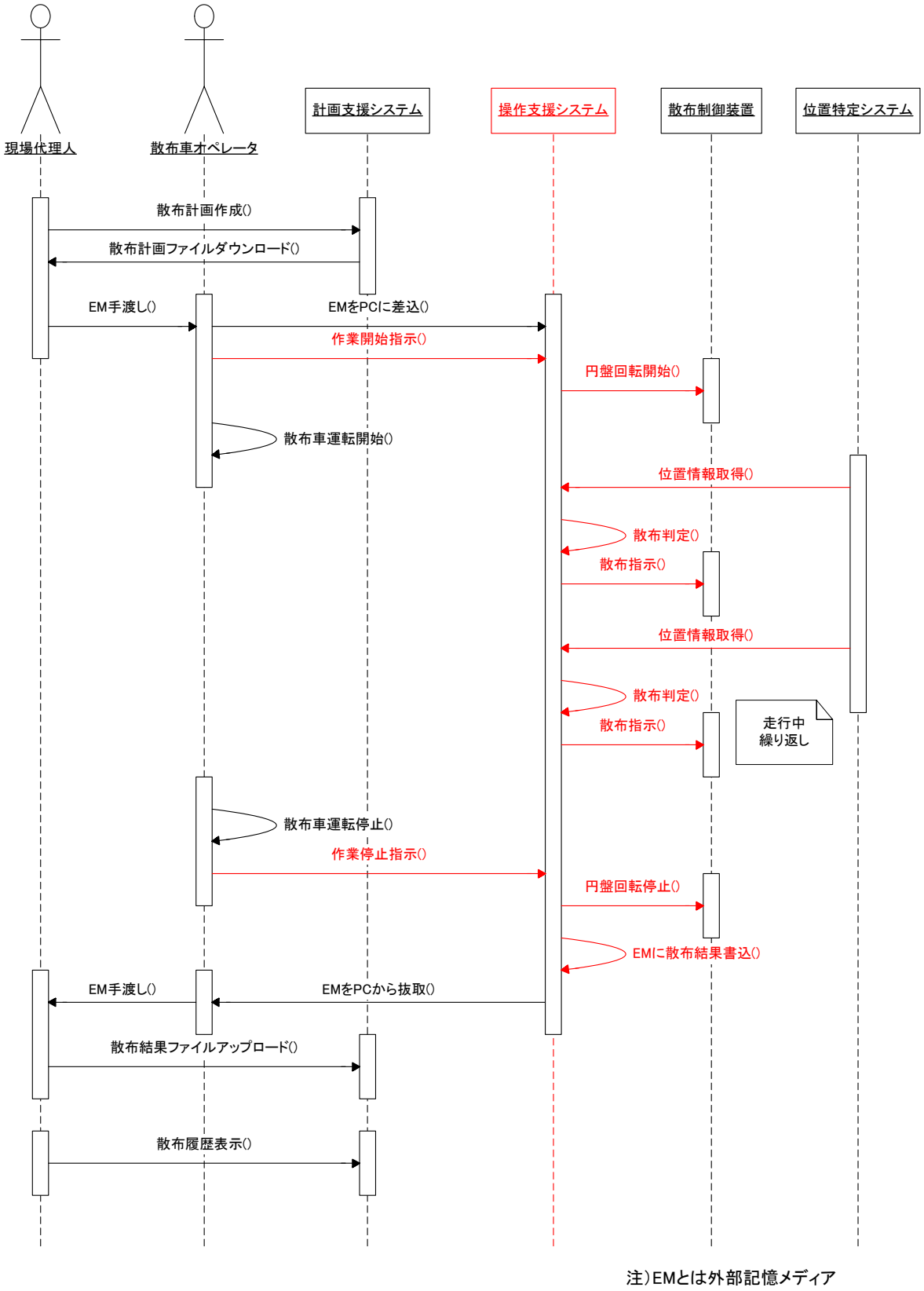


図 2-4　システム全体シーケンス図

(4) 機能設計

■ ユースケースの詳細化

システムとして必要な機能を詳細化するために、各ユースケースをさらに詳細化する。詳細化した機能を表 2-5 に示す。

表 2-1 システムに必要な機能

機能		詳細機能	機能概要
散布操作支援	自車位置情報提供	道路地物検出機能	自車位置周辺の道路地物等を検出する機能。
		自車位置情報提供機能	自車位置周辺の道路地物等の位置情報をオペレータに提供する機能。
	散布箇所接近情報提供	散布制御変更箇所接近検出機能	自車両と接近した散布制御変更箇所を検出する機能。
		散布制御変更箇所の位置情報提供機能	散布制御変更箇所の存在をオペレータに注意喚起する機能。
	自動散布制御	散布制御変更箇所検出機能	自車両が散布制御変更箇所に存在するかを検出する機能。
		自動散布機能	散布計画データに基づいて散布装置を自動的に制御する機能。
	散布履歴情報収集	散布履歴情報収集機能	散布履歴情報を収集する機能。

■ 自車位置情報提供機能

自車位置情報提供機能は、予め登録しておいた道路地物を検索し、画面地図上にその位置を表示するとともに、その地物までの距離を画面上に表示する機能（道路地物検出機能）およびカーナビゲーションのように自車の現在位置を画面地図上に表示（自車位置情報提供機能）するものである。機能の制約としては、以下とする。

検索対象地物の変更が容易なこと

エリア広域を簡易表示する全体地図と、現在位置を中心として周辺を詳細に表示する２つの地図画面を有すること

自車位置を表示する詳細地図はヘディングアップ機能を持たせること

詳細地図には道路 GIS データを用いることとし、他の市販地図にも対応可能なこと

詳細地図は、拡大・縮小が可能なこと

詳細地図に表示する地物は表示切替えが可能なこと

なお、道路地物検索機能において、実現する検索機能を図 2-12 に示す。また、そのアルゴリズムの概要を図 2-13 に示す。また、ヘディングアップ機能には、位置特定システムから得られる自車の進行方向をもとに地図を回転させることによって実現する。このため、背景地図として利用できる地図データの形式はベクターデータ型の地図の使用に限定する。

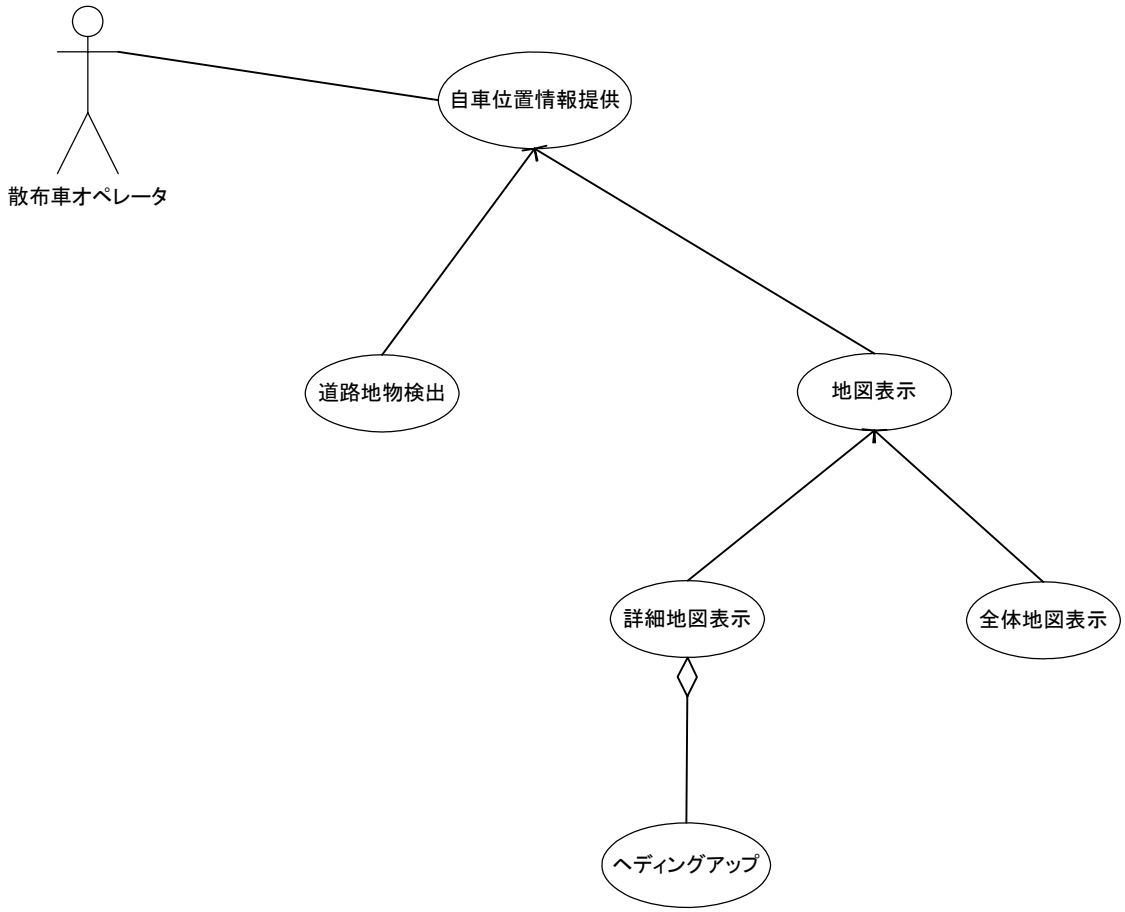


図 2-5　自車位置情報提供　詳細ユースケース図

表 2-2　自車位置情報提供　ユースケースシナリオ

ユースケース	自車位置情報提供	主アクタ	散布車オペレーター

事前条件

操作支援システムが起動され、背景地図、道路 GIS および表示設定ファイルが読み込まれていること。

散布車の発進箇所は、除雪ステーションとする。

メインフロー

散布車オペレーターは、画面上の「開始」ボタンを押す。

散布車オペレーターは、散布車を発進する。

本システムは、位置特定パッケージより、緯度、経度、進行方向方位角を取得する。

本システムは、緯度、経度より地図の表示位置を算出し、進行方向方位角に合わせて地図を回転し、表示する。

設定ファイルに予め設定された検索地物情報に従い地物を検索し、現在走行中の車線上で最も近い地物の名称とその地物までの距離を表示する。また、地図上に地物の箇所を点滅して表示する。

設定ファイルに予め設定された表示周期で、地図を表示する。

２から４の処理を繰り返す

散布車オペレーターは、地図の表示範囲を変更するため、「拡大」または「縮小」ボタンを押す：例外処理１

散布車オペレーターは、表示する地物を変更するため、「表示変更」ボタンを押す：例外処理２

散布車オペレーターは、散布車を停止する。

散布車オペレーターは、画面上の「停止」ボタンを押して、本システムを終了する。

事後条件

散布車の停止箇所は、除雪ステーションとする。

例外フロー

例外処理１：本システムは、設定ファイルに予め設定された拡大・縮小率で背景地図を拡大・縮小して表示する。

例外処理２：本システムは、設定ファイルに予め設定された表示地物設定の地物表示情報設定ウィンドウを表示し、散布車オペレーターが変更した設定情報に従って地物の表示・非表示を切り替え地図を表示する。

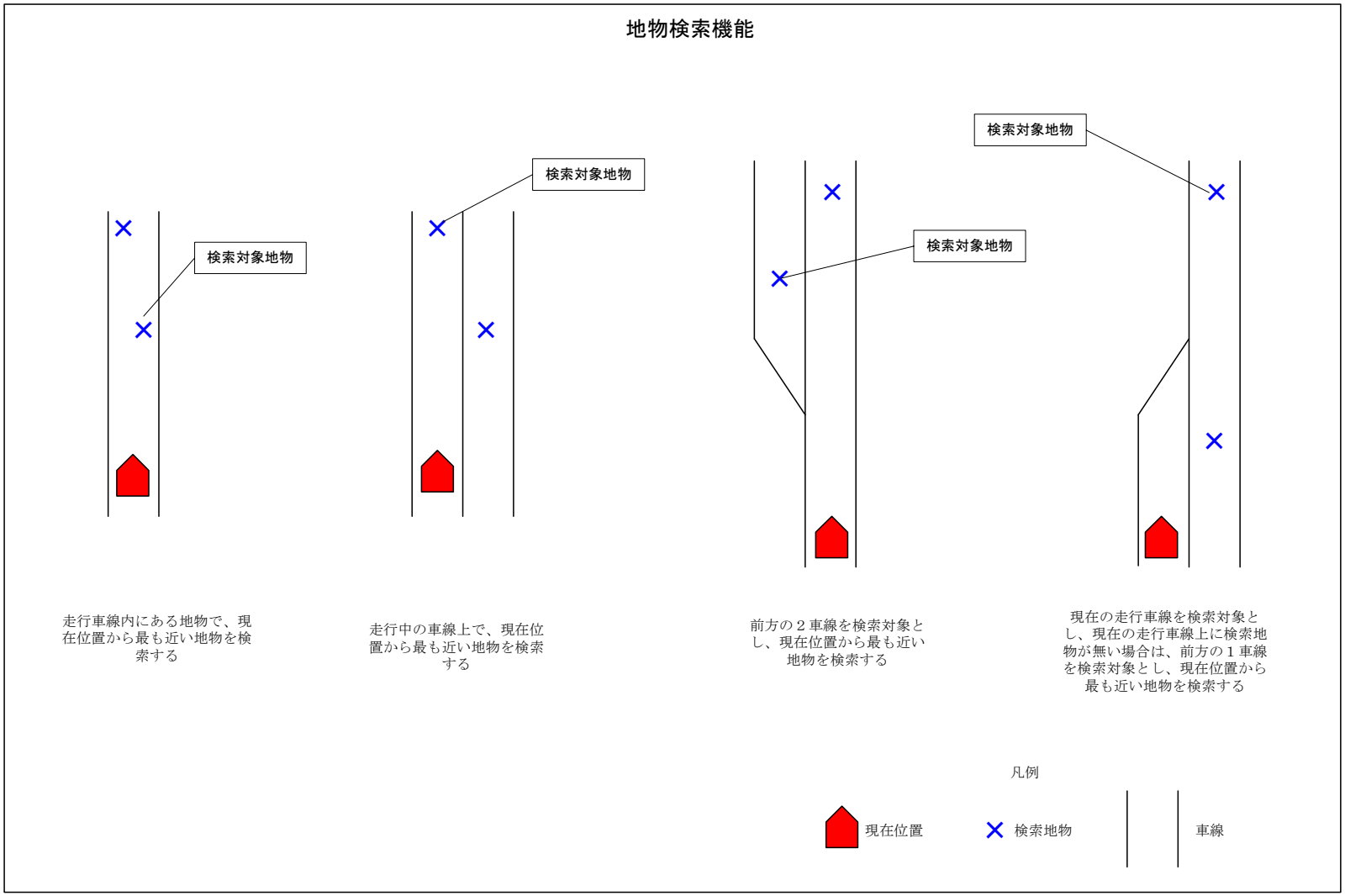

地物検索機能
検索対象地物
走行車線内にある地物で、現在位置から最も近い地物を検索する
検索対象地物
走行中の車線上で、現在位置から最も近い地物を検索する
検索対象地物
前方の２車線を検索対象とし、現在位置から最も近い地物を検索する
検索対象地物
現在の走行車線を検索対象とし、現在の走行車線上に検索地物が無い場合は、前方の１車線を検索対象とし、現在位置から最も近い地物を検索する
凡例
現在位置
検索地物
車線

図 2-6　地物検索機能

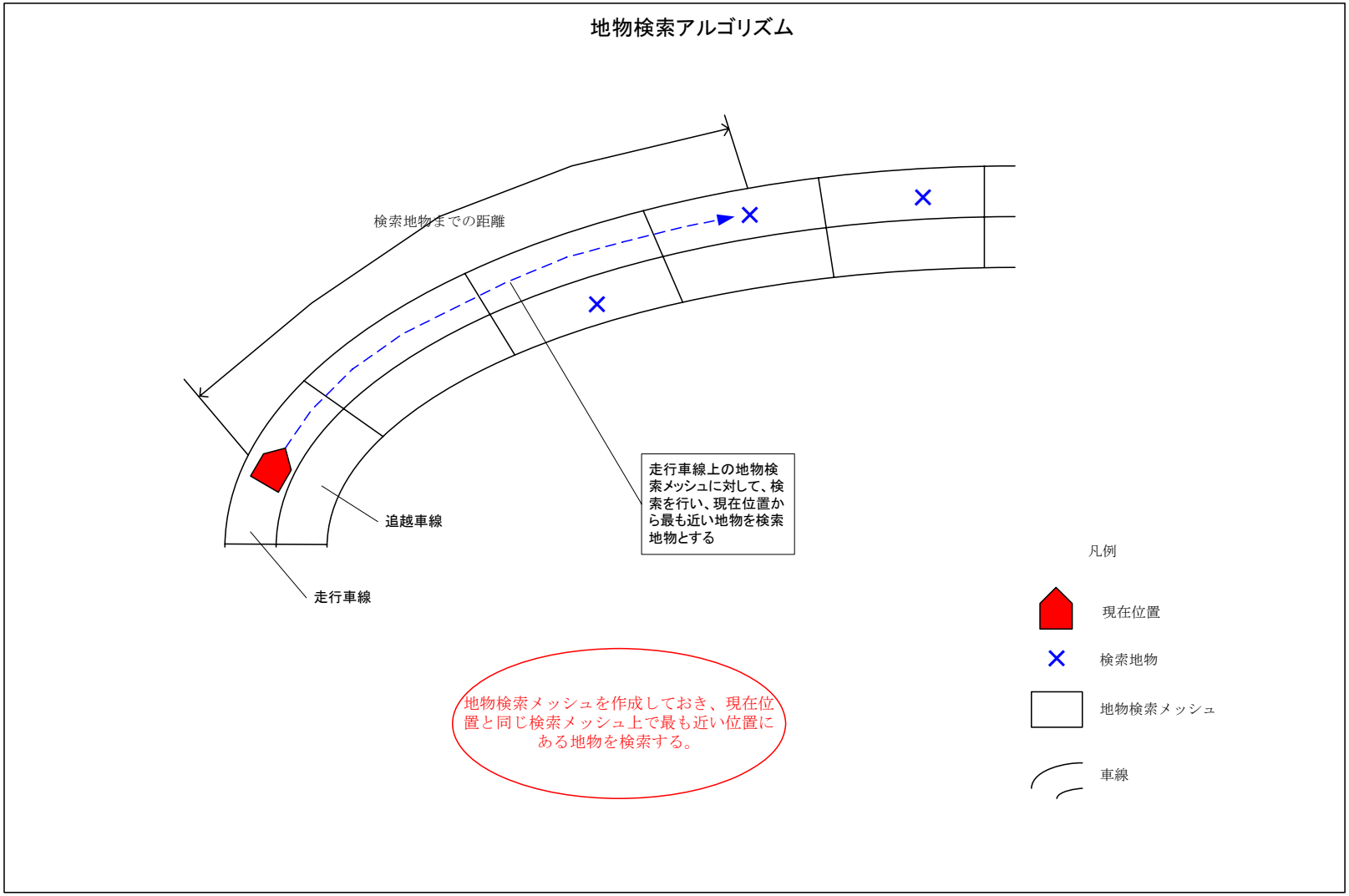


図 2-7　地物検索アルゴリズム

■ **散布箇所接近情報提供機能**

散布箇所接近情報提供機能は、散布開始位置や散布停止位置、散布量や散布幅などの散布情報が変わる地点などの散布条件が変更する箇所の検索を行うものである。機能の制約としては、以下とする。

散布変更箇所までの距離を表示する

散布情報変更時には、音声（ビープ音）で知らせる

散布予定箇所を地図画面上に表示する

散布済箇所を地図上に表示する

散布変更位置検索アルゴリズムを図 2-15 に、音声案内イメージを図 2-16 に示す。

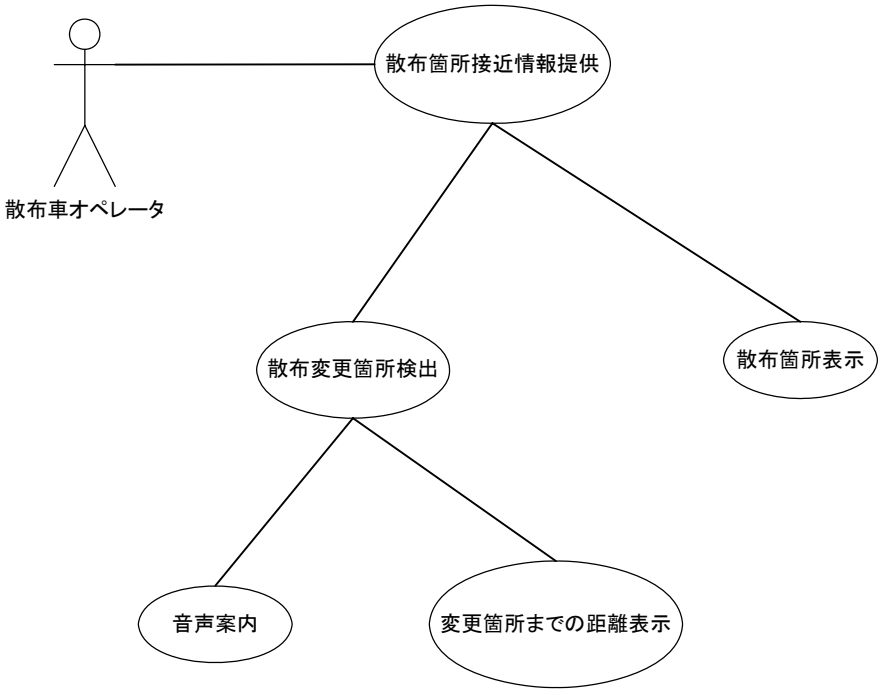


図 2-8　散布箇所接近情報提供　詳細ユースケース図

表 2-3　散布箇所接近情報提供　ユースケースシナリオ

ユースケース	散布箇所接近情報提供	主アクタ	散布車オペレータ

事前条件

散布制御用のデータが作成され、PC 内あるいは、外部記憶装置に記録されていること。
散布車オペレーターの名前が、本システムに登録されていること。
散布車の発進箇所は、除雪ステーションとする。

メインフロー

散布車オペレーターは、システムを起動する。
散布車オペレーターは、作業者名選択画面画面が表示されるので、自分の名前を選択する。
散布車オペレーターは、散布計画ファイル選択画面が表示されるので、これから散布制御する散布計画ファイルを選択する（例外処理１）。
散布車オペレーターは、画面上の「開始」ボタンを押す。
散布車オペレーターは、散布車を発進する。
本システムは、現在位置より、散布箇所の検索を行い、現在の散布状況により、「散布開始まで」「散布変更まで」「散布停止まで」の表示を切り換え、その地点までの距離を表示する（例外処理２）。
本システムは、散布の制御情報が開始または停止、変更になった時点で、音声（ビープ音）のより散布車オペレータに散布制御変更を知らせる。
散布車オペレーターは、散布車を停止する。
散布車オペレーターは、画面上の「停止」ボタンを押して、本システムを終了する。

事後条件

散布車の停止箇所は、除雪ステーションとする。

例外フロー

例外処理１：設定ファイルに予め設定された散布計画ファイル格納ディスク（コンパクトフラッシュ等の外部記憶装置）が存在しない場合は、格納ディスクの挿入を促すメッセージを表示して、格納ディスクの挿入を待つ。
例外処理２：走行中の車線上に、散布箇所が存在しないまたは全て散布済みの場合には、何も表示しない。

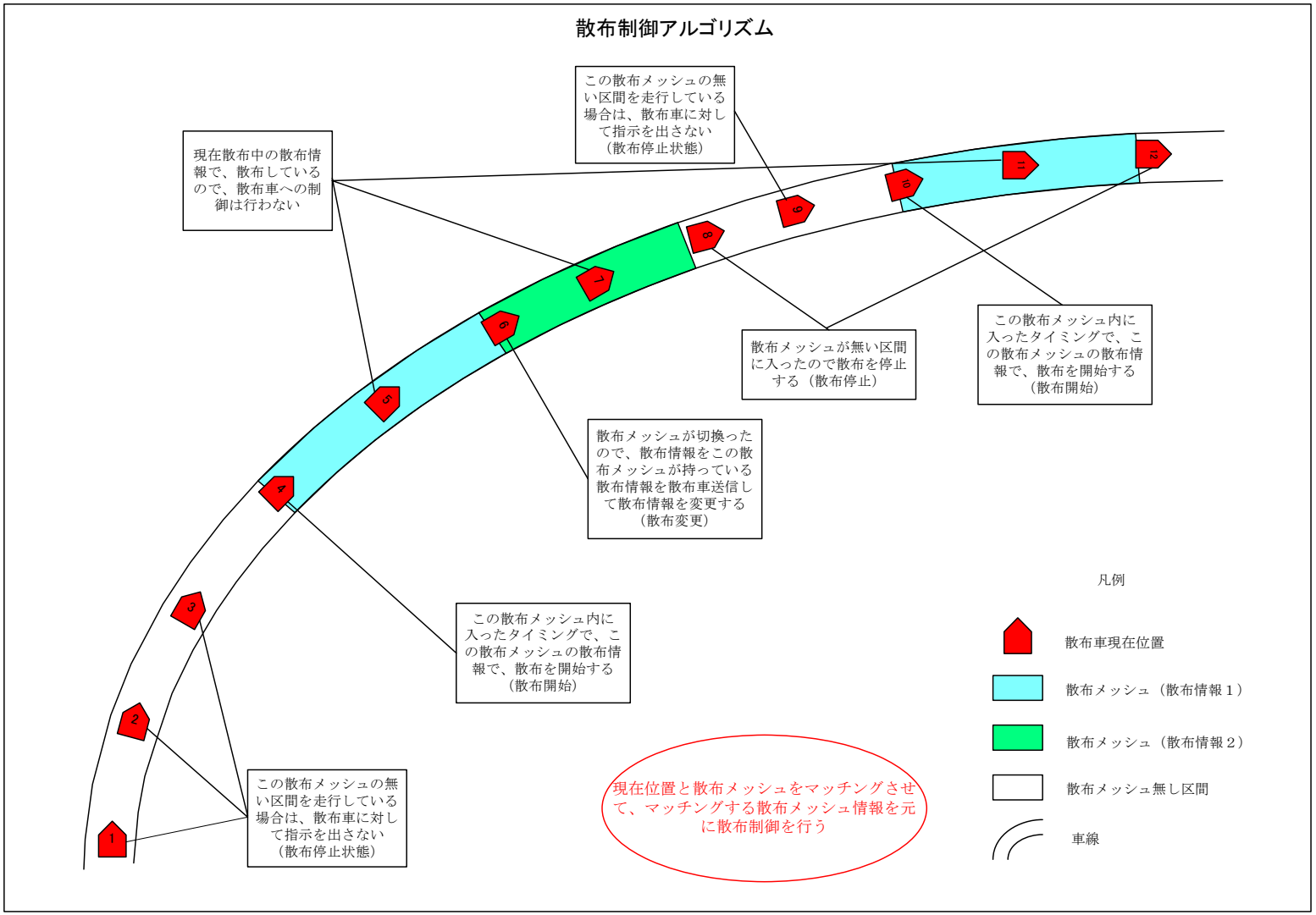


図 2-9　散布変更位置アルゴリズム

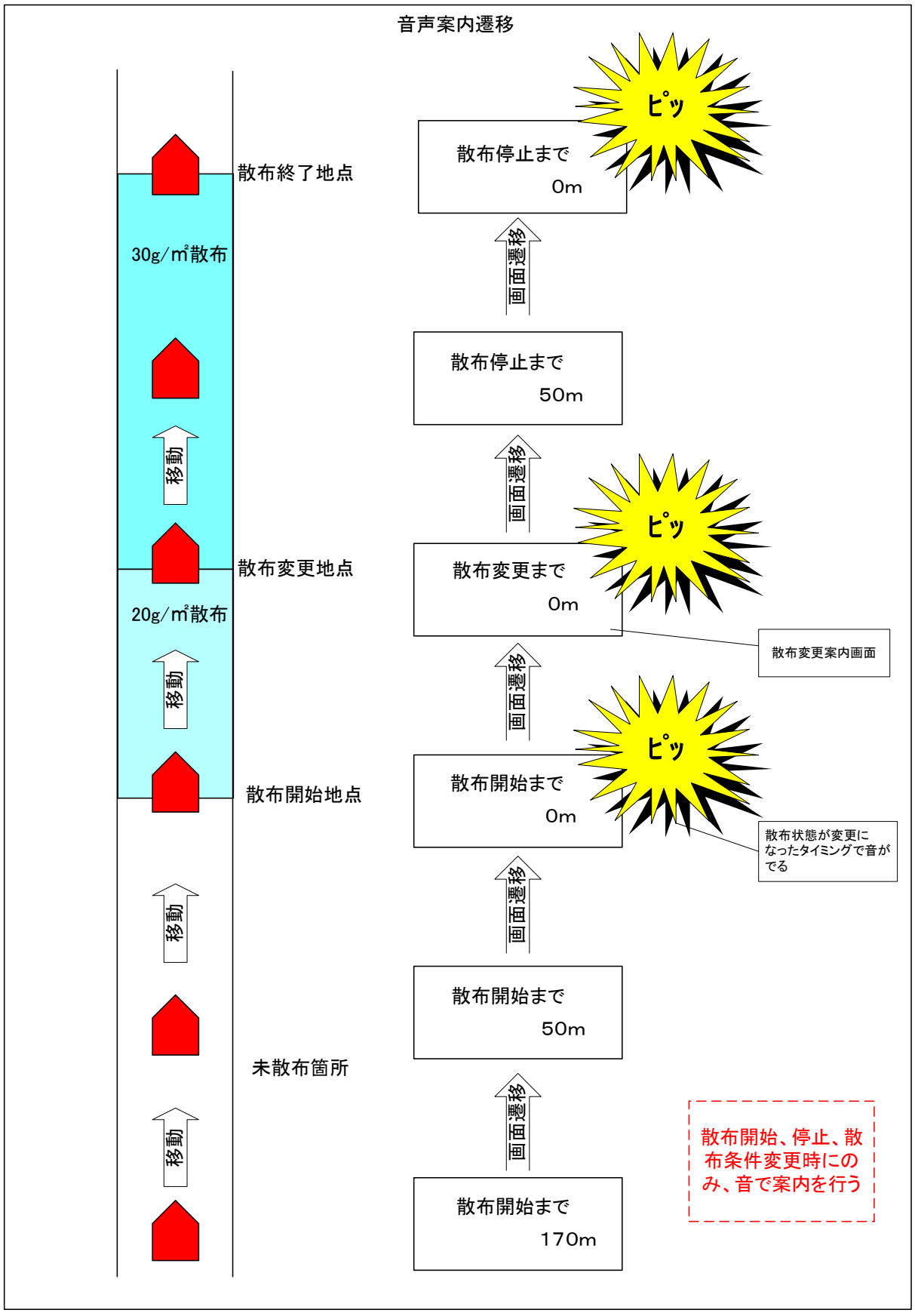


図 2-10　音声案内イメージ

■ **自動散布制御機能**

自動散布制御機能は、散布計画支援システムによって作成された作業計画ファイルを利用して、散布装置の自動制御を行うものである。機能の制約としては、以下とする。

散布制御ファイルに従って、散布の ON、OFF を自動的に切り替える

散布制御ファイルに従って、散布量や散布幅などの散布情報を自動的に切り替える

使用する散布制御ファイルを使い分けることにより、散布制御を変更することができる

散布制御ファイルを散布途中で切り替える

散布制御アルゴリズムを図 2-18 に、散布制御対応パターンを図２－１３～図２－１６に、散布条件切替パターンを図 2-19 に、未対応機能を図２－１８に示す。

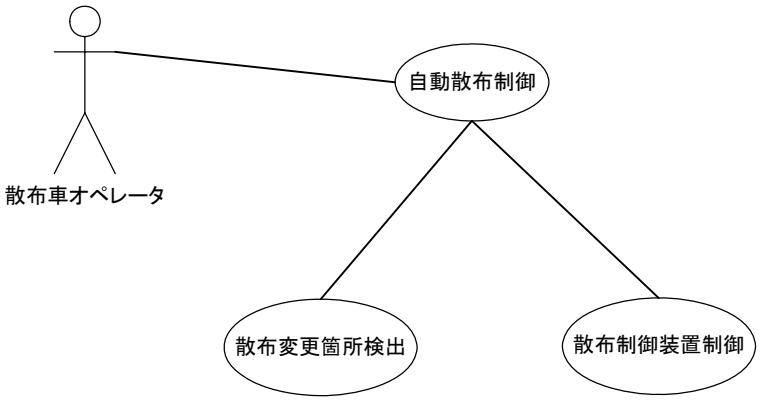


図 2-11　自動散布制御　詳細ユースケース図

表 2-4　自動散布制御機能　ユースケースシナリオ

ユースケース	自動散布制御機能	主アクタ	散布車オペレーター

事前条件

操作支援システムが起動され、散布計画ファイル選択画面が表示されていること。
散布制御用のデータが作成され、PC 内あるいは、外部記憶装置に記録されていること。
散布車の発進箇所は、除雪ステーションとする。

メインフロー

散布車オペレーターは、散布計画ファイル選択画面が表示されるので、これから散布制御する散布計画ファイルを選択する（例外処理１）。
散布車オペレーターは、画面上の「開始」ボタンを押す。
散布車オペレーターは、散布車を発進する。
本ステムは、現在位置と散布計画ファイルによる散布箇所により、散布箇所箇所かどうかの判断を行い、散布開始、停止、変更の処理を行う。
本システムは、散布中の情報（自動・手動の区別、散布量、散布幅）を画面上に表示する。
散布車オペレーターは、走行途中で、最初に選択した散布計画を変更する必要が生じたので、散布計画ファイルを変更する（例外処理２）。
散布車オペレーターは、散布車を停止する。
散布車オペレーターは、画面上の「停止」ボタンを押して、本システムを終了する。

事後条件

散布車の停止箇所は、除雪ステーションとする。

例外フロー

例外処理１：設定ファイルに予め設定された散布計画ファイル格納ディスク（コンパクトフラッシュ等の外部記憶装置）が存在しない場合は、格納ディスクの挿入を促すメッセージを表示して、格納ディスクの挿入を待つ。
例外処理２：散布車オペレーターは、「停止」ボタンを押し、散布計画ファイルを再度選択してから「開始」ボタンを押す。本システムは、新たな散布制御ファイルに従って散布制御を行う。

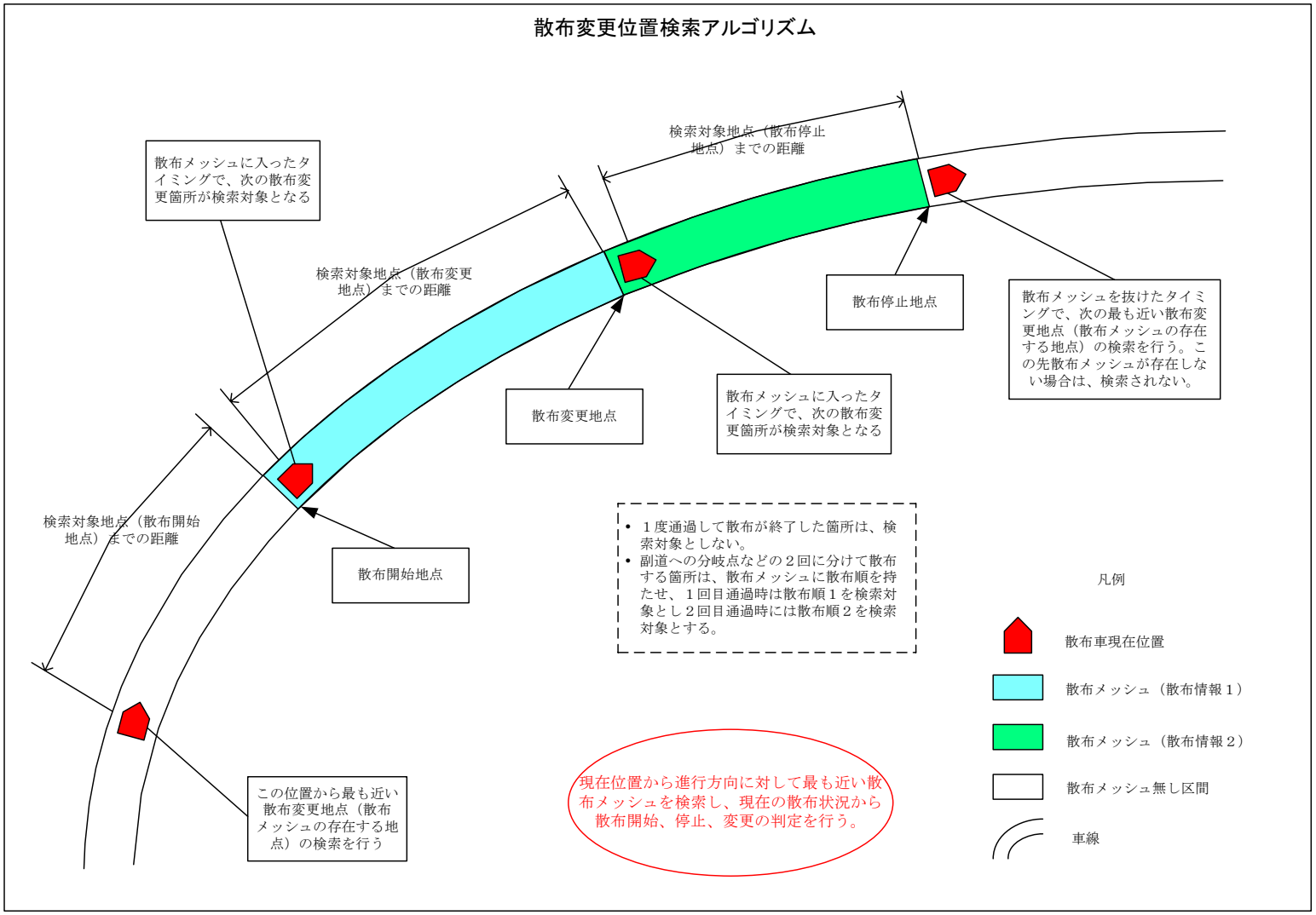


図 2-12　散布制御アルゴリズム

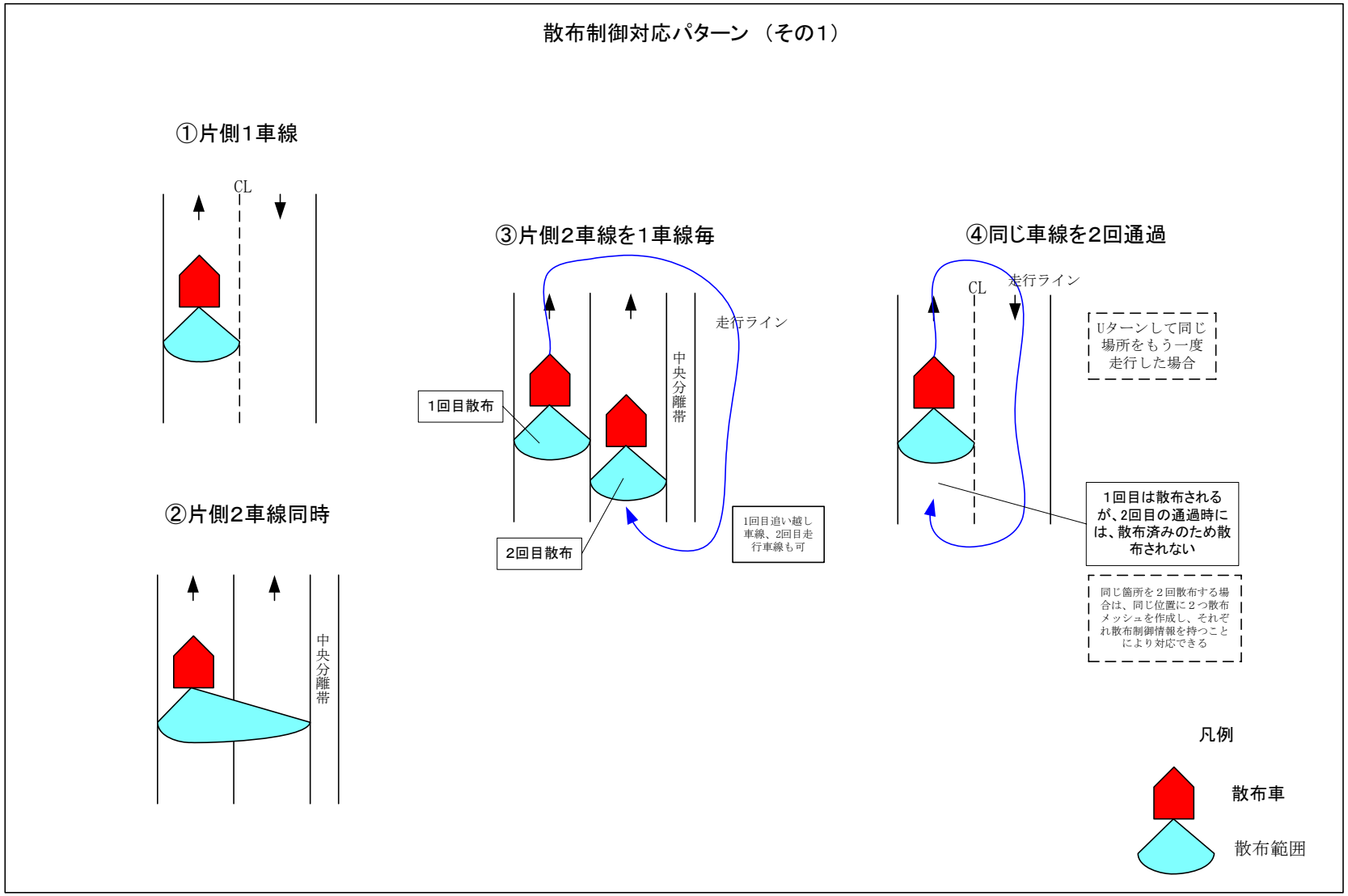


図 2-13　散布制御対応パターン（その１）

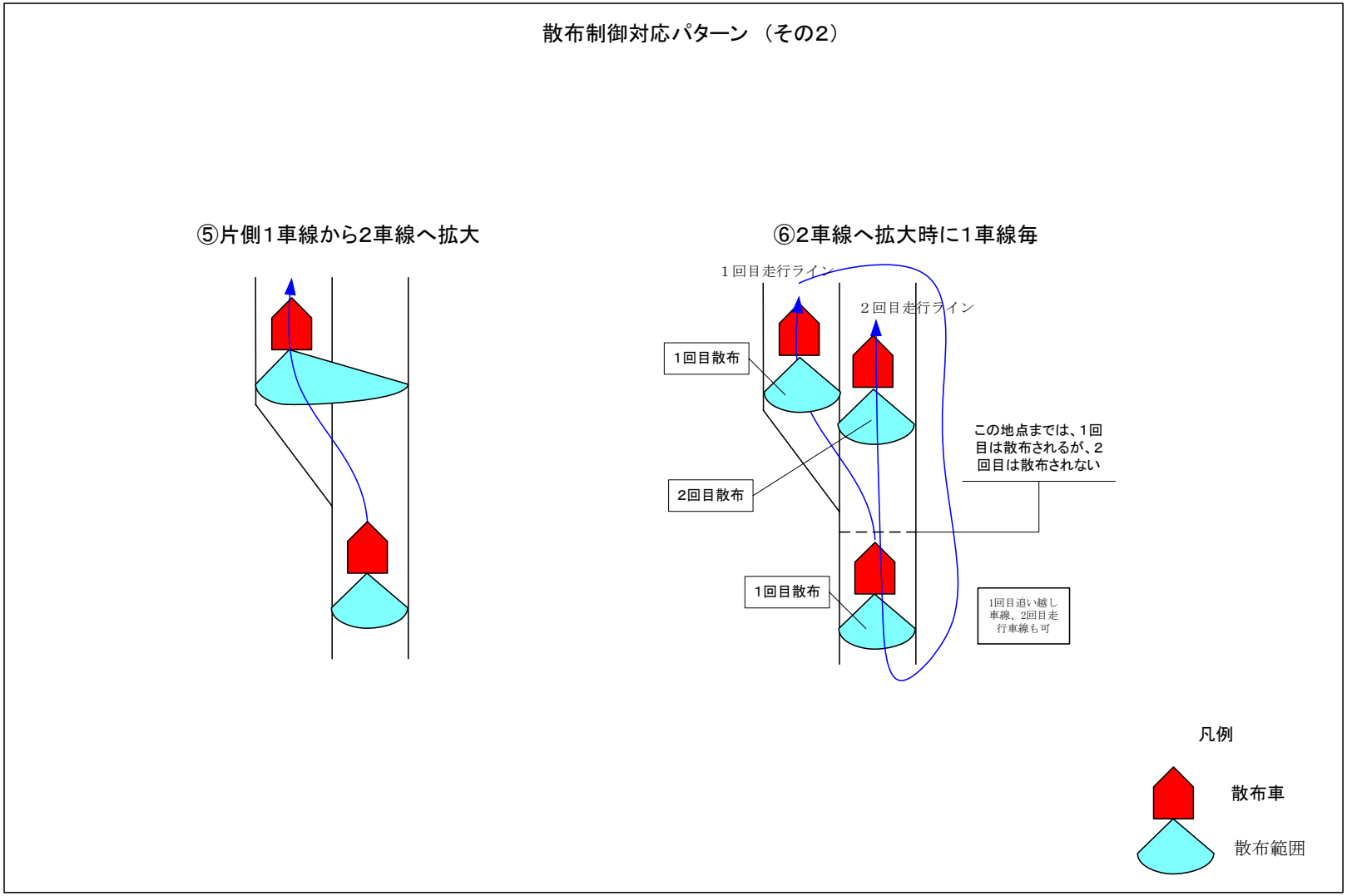


図 2-14　散布制御対応パターン（その２）

散布制御対応パターン（その3）

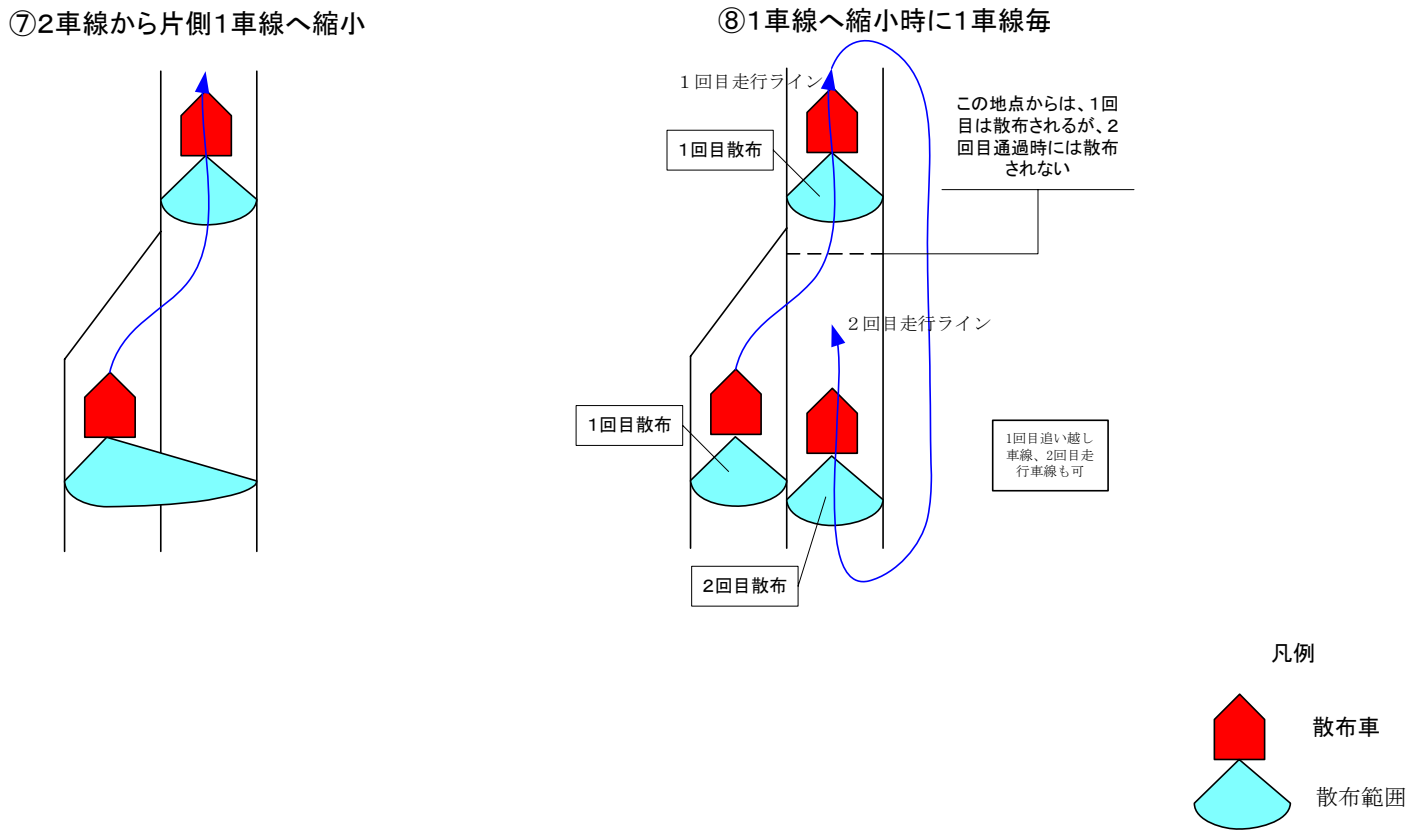


図 2-15　散布制御対応パターン（その３）

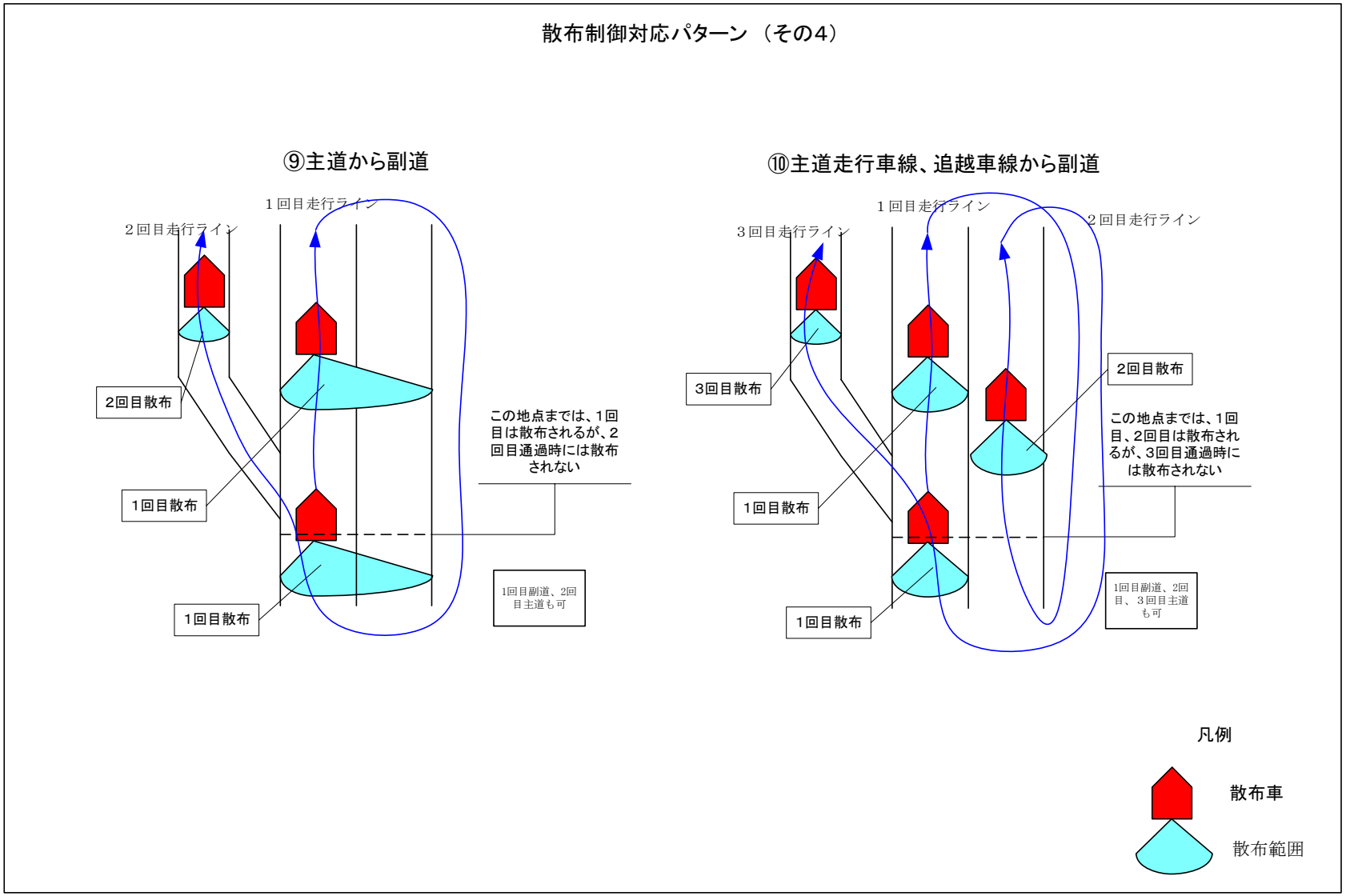


図 2-16　散布制御対応パターン（その４）

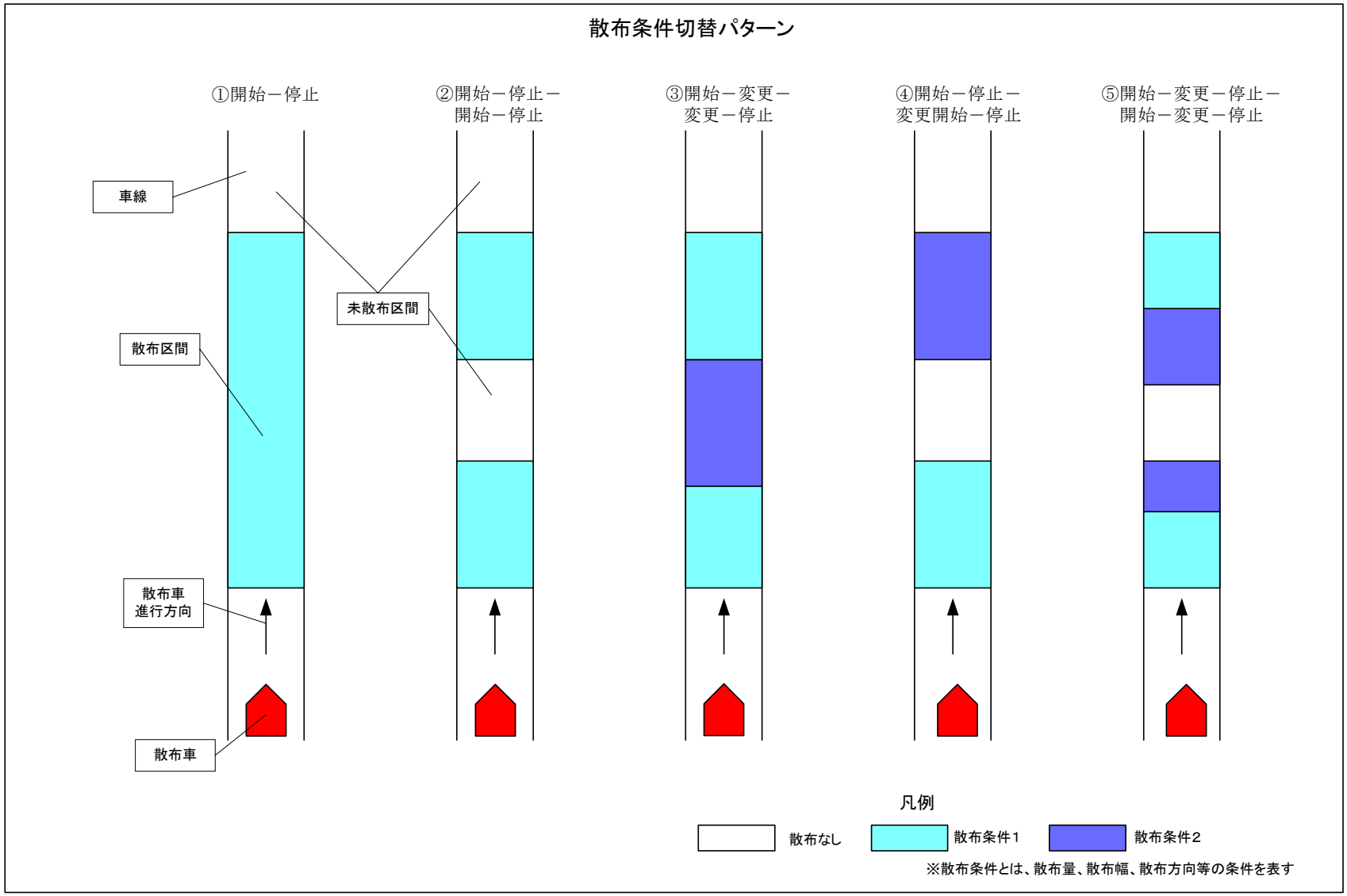


図 2-17　散布条件切替パターン

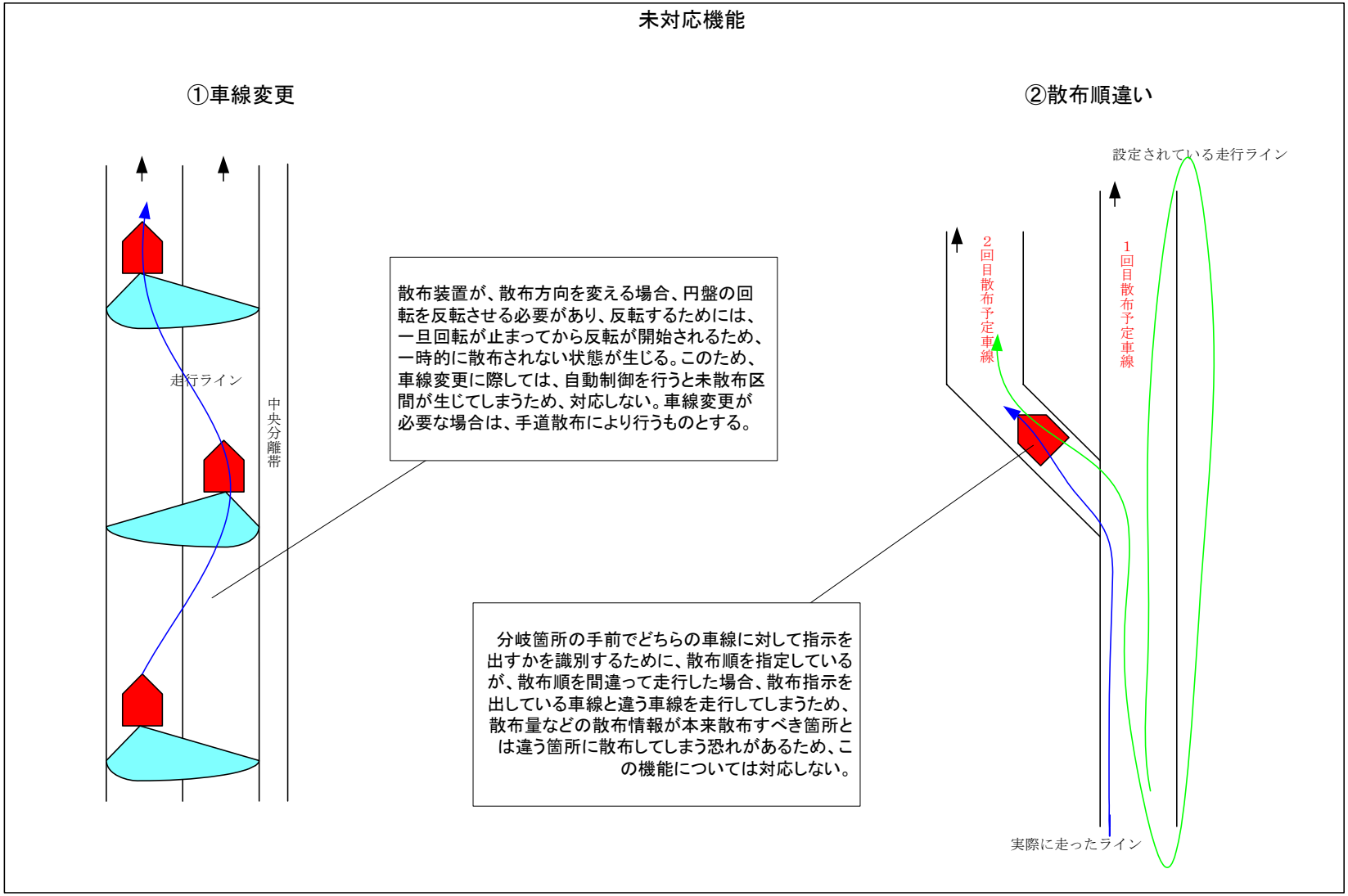


図 2-18　散布制御未対応パターン

■ **散布履歴情報収集機能**

散布履歴情報収集機能は、実際に散布された箇所や散布時間、散布量や散布幅などの散布情報を記録し、計画支援システムにデータを登録することによって、散布履歴の表示を行えるようにするためのものである。機能の制約としては、以下とする。

作業記録として、作業者名、作業計画ファイル名を記録する

作業記録として、散布時刻を記録する

作業記録として、散布位置を KP 単位で記録する

作業記録として、散布量、散布幅、散布方向など散布車の制御項目を記録する

作業記録として、散布状態（自動散布または手動散布）を記録する

記録するデータは、散布が切換ったタイミングで記録する

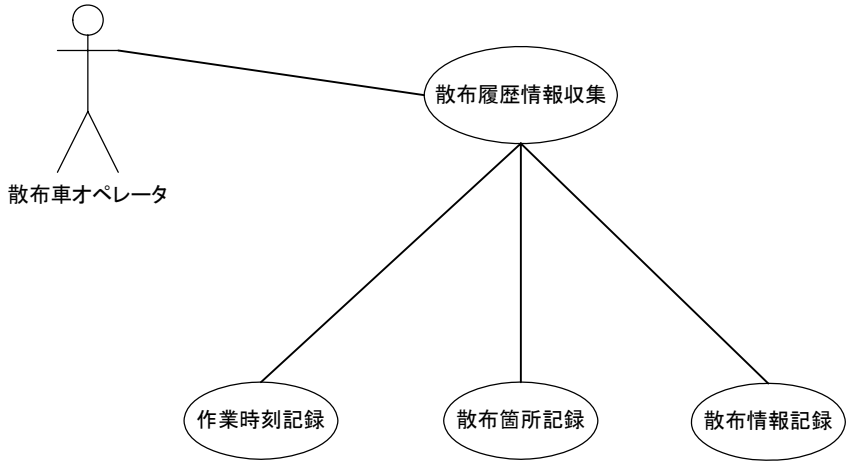


図 2-19　散布履歴情報収集　詳細ユースケース図

表 2-5　散布履歴情報収集　ユースケースシナリオ

ユースケース	散布履歴情報収集	主アクタ	散布車オペレーター

事前条件

操作支援システムが起動され、作業員名、散布計画ファイルが選択されていること。
散布車の発進箇所は、除雪ステーションとする。

メインフロー

散布車オペレーターは、画面上の「開始」ボタンを押す。
散布車オペレーターは、散布車を発進する。
本システムは、散布計画ファイルに従って散布制御装置に散布制御情報を渡し散布を制御する。
本システムは、散布制御装置から返される散布制御結果をシステム内に記録する。
散布車オペレーターは、散布車を停止する。
散布車オペレーターは、画面上の「停止」ボタンを押して、本システムを終了する。
本システムは、設定ファイルに予め設定された散布履歴格納ディスクに散布履歴情報を格納する（例外処理１）。

事後条件

散布車の停止箇所は、除雪ステーションとする。

例外フロー

例外処理１：設定ファイルに予め設定された散布履歴ファイル格納ディスク（外部記憶装置）が存在しない場合は、格納ディスクの挿入を促すメッセージを表示して、格納ディスクの挿入を待つ。

■ その他システム上必要な機能

システムの目的を達するための機能の他に、運用する上で必要な機能として以下の機能を実現するものとする。

システムハングアップ時に、再開できる機能

表 2-6　システムハングアップ時　ユースケースシナリオ

ユースケース	システムハングアップ	主アクタ	散布車オペレーター

事前条件

システムは、既に起動済みであること。

メインフロー

本システムは、電源切断等による外的要因により、停止。

散布オペレーターは、安全地帯に移動し、散布車を停止する。

散布車オペレーターは、位置特定機器および PC を再起動する。

散布車オペレーターは、本システムを起動する。

散布車オペレーターは、画面上の「開始」ボタンを押す。

本システムは、ハングアップ時のログ情報をもとに散布制御を再開する。

散布車オペレーターは、散布車を停止する。

散布車オペレーターは、画面上の「停止」ボタンを押して、本システムを終了する。

事後条件

散布車の停止箇所は、除雪ステーションとする。

例外フロー

(5) システムモデリング

本システムを概念的にモデリングした概念クラス図および、シーケンス図を以下に示す。

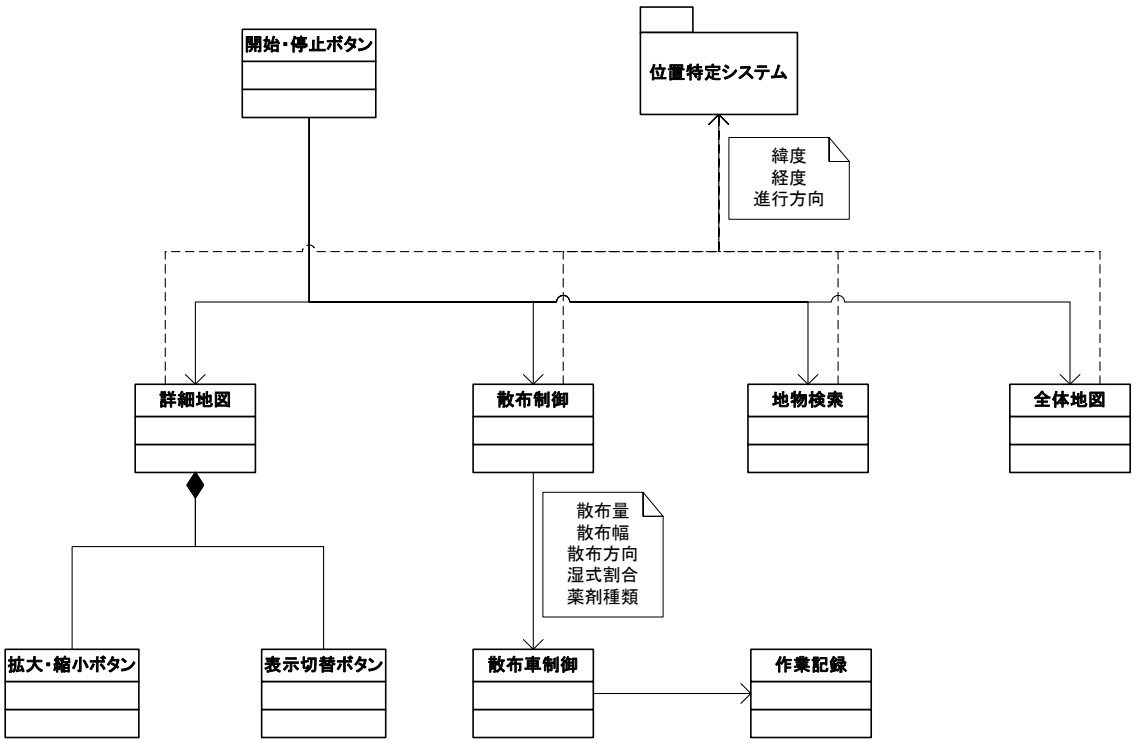


図 2-20　概念クラス図

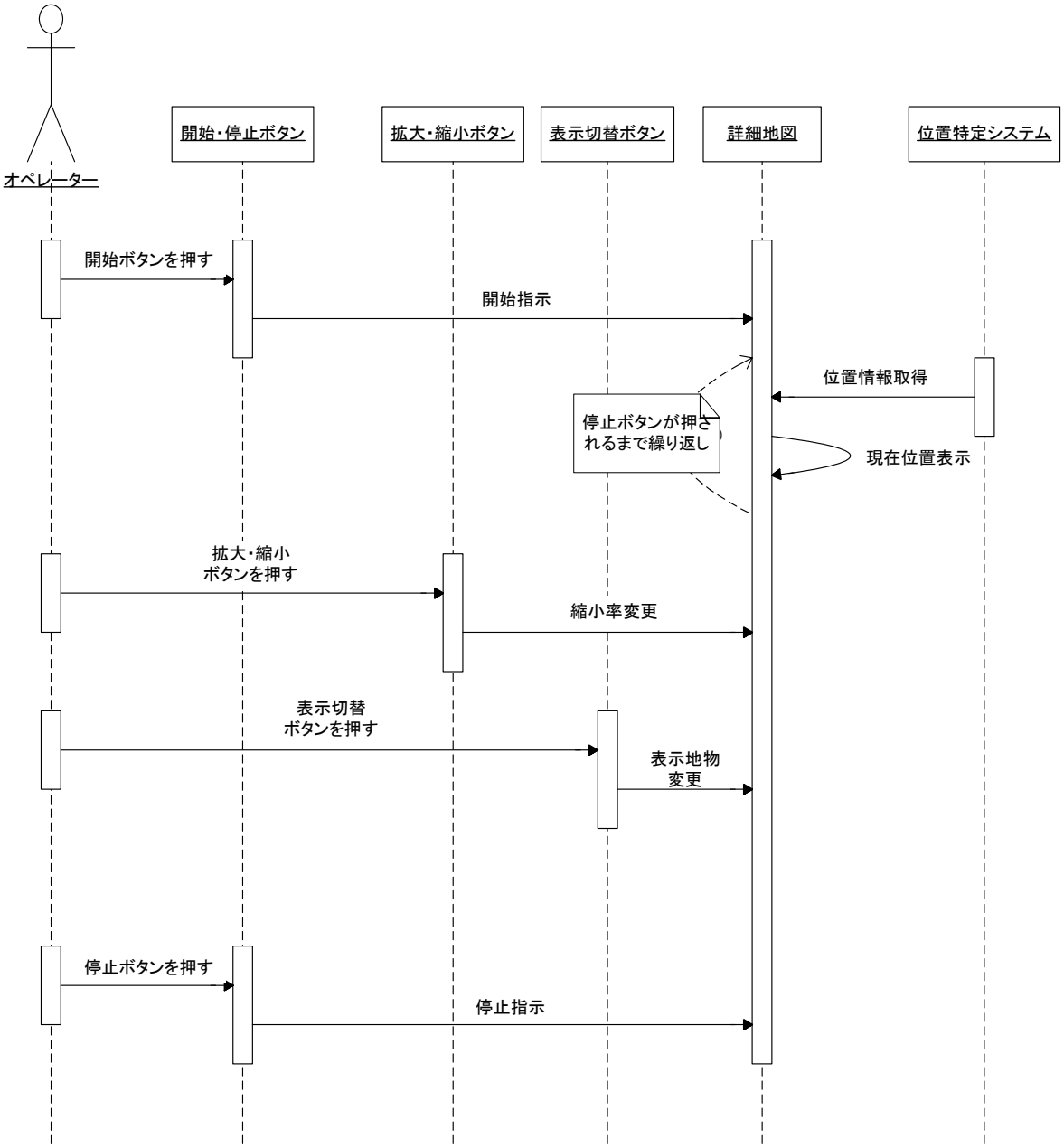


図 2-21　詳細地図シーケンス図

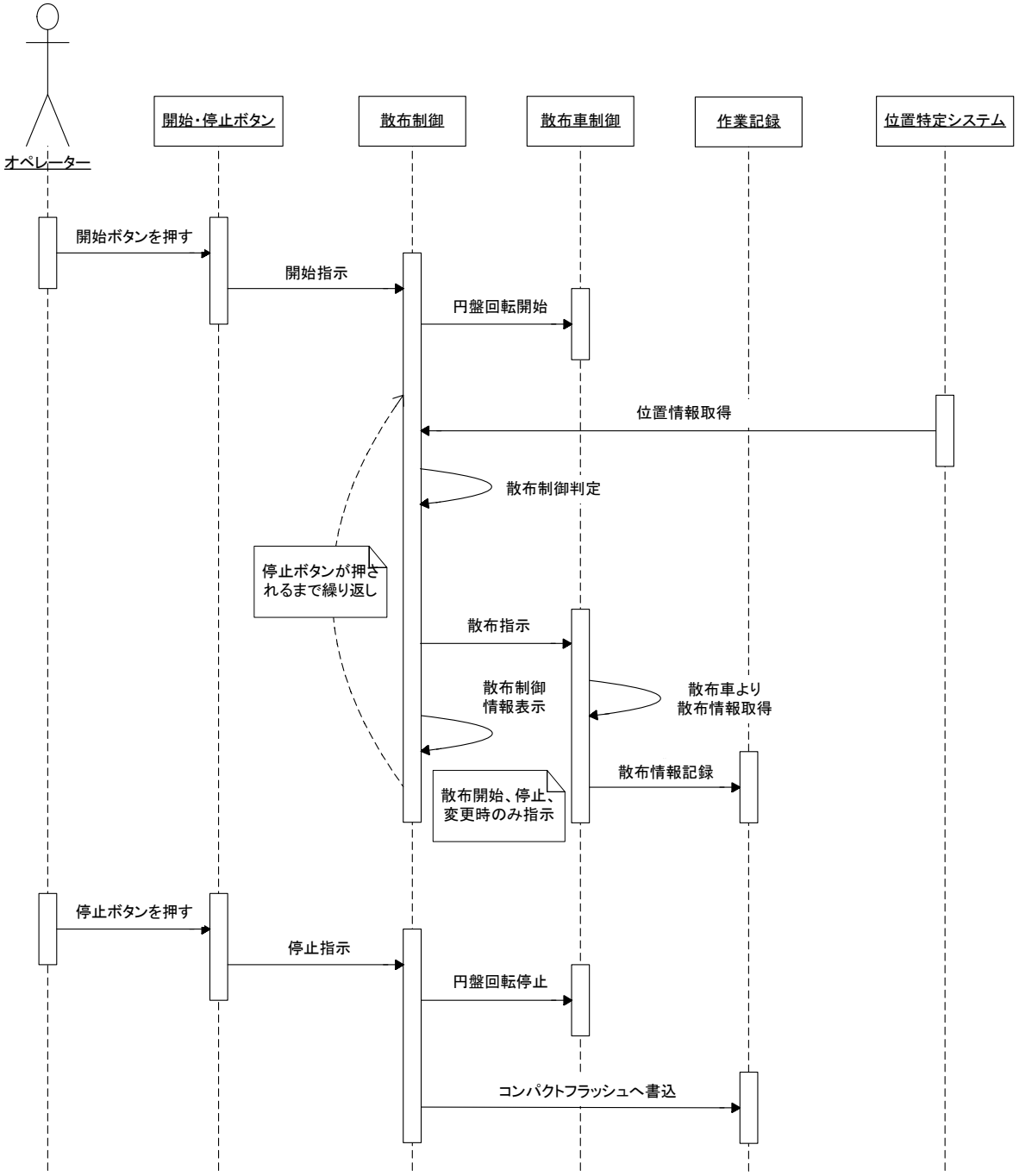


図 2-22　散布制御シーケンス図

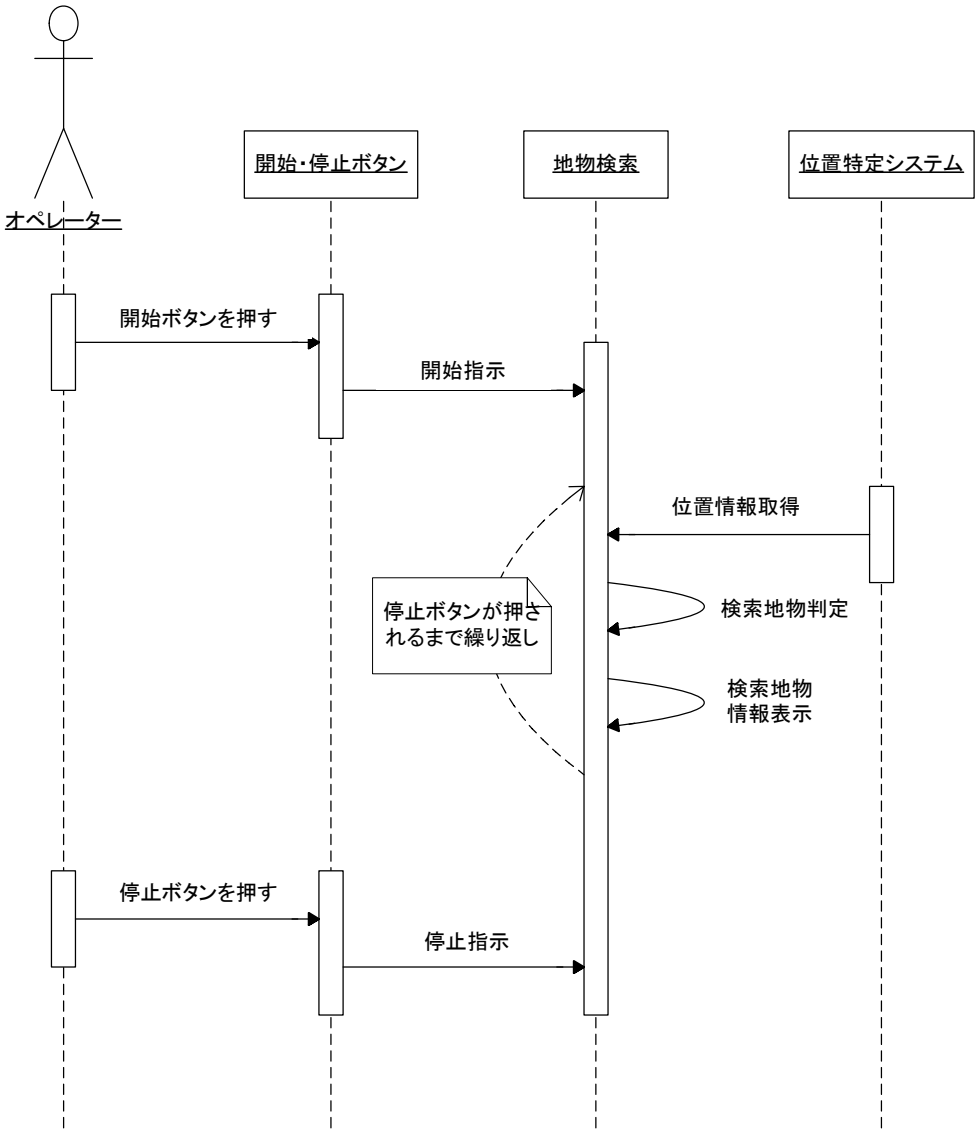


図 2-23　地物検索シーケンス図

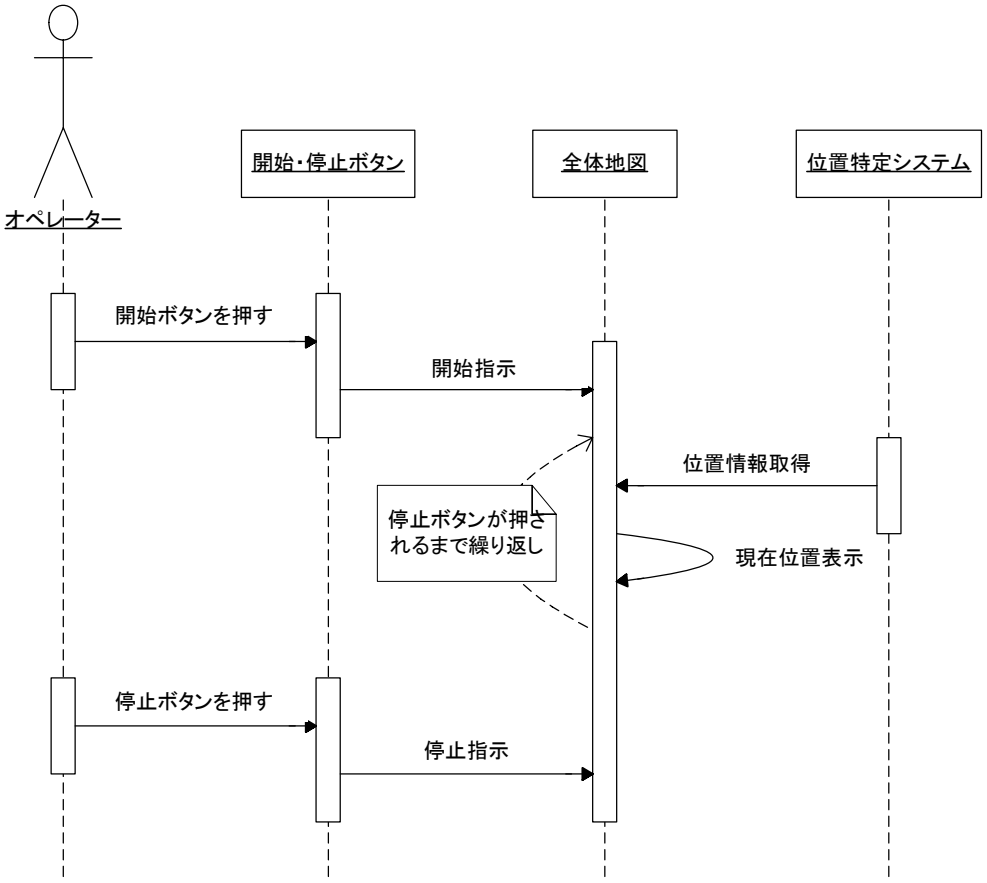


図 2-24　全体地図シーケンス図

(6) 画面設計

本システムにおける画面および画面内の表示の遷移図を以下に示す。

散布自動制御システム
全体地図
自車位置のKP表示
散布量表示
散布変更地点までの距離
指定した地物(交差点、橋梁等)までの距離
詳細地図
(ヘディングアップ機能により、進行方向が上になります)
散布予定箇所
自車位置
散布済箇所
下り 61.25KP
通過中
自動散布中
散布量：20g/㎡
散布幅：5.0m
散布停止まで
147m
85m先
蓮潟交差点
停止

図 2-25　画面イメージ

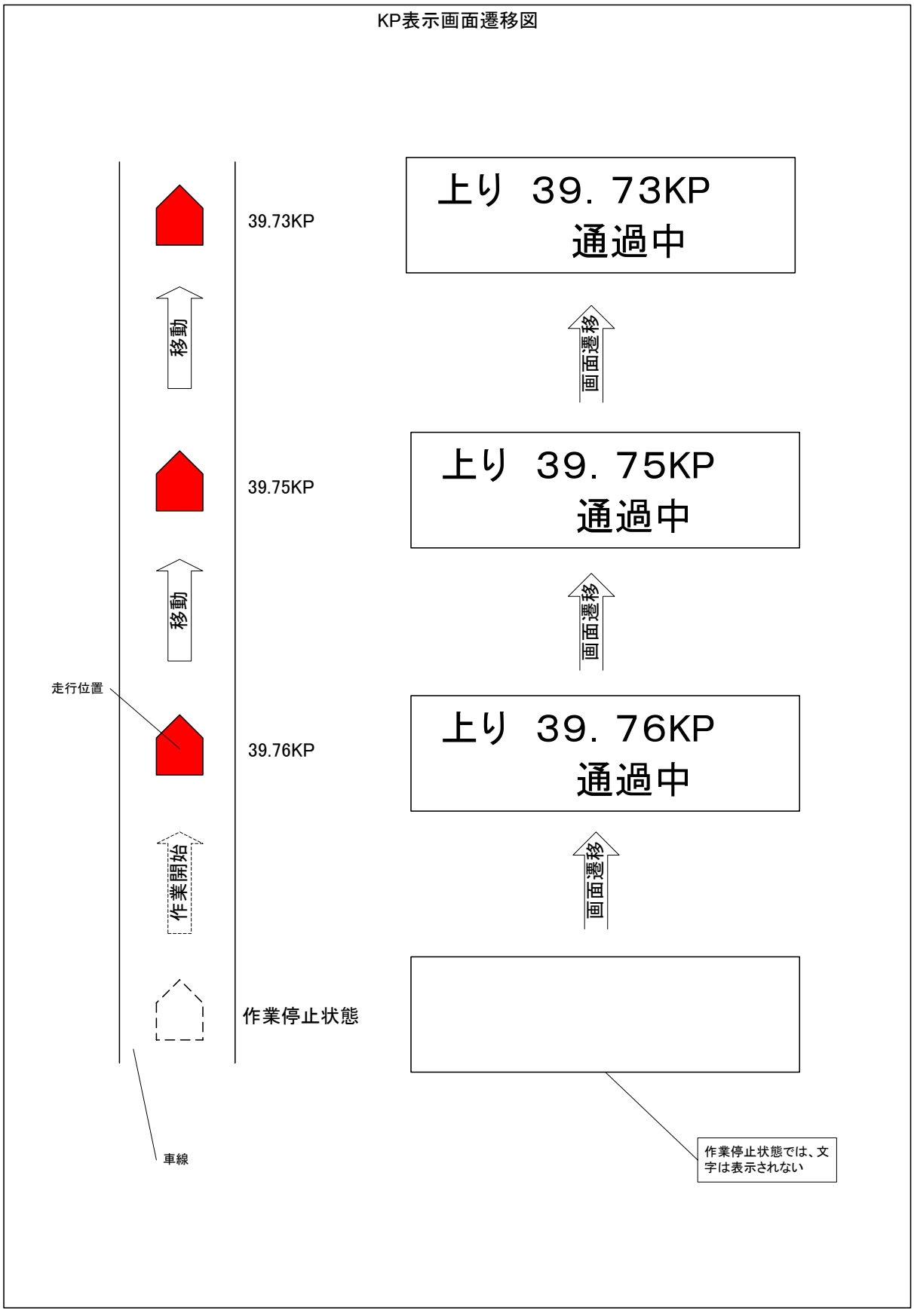


図 2-26　ＫＰ表示画面遷移

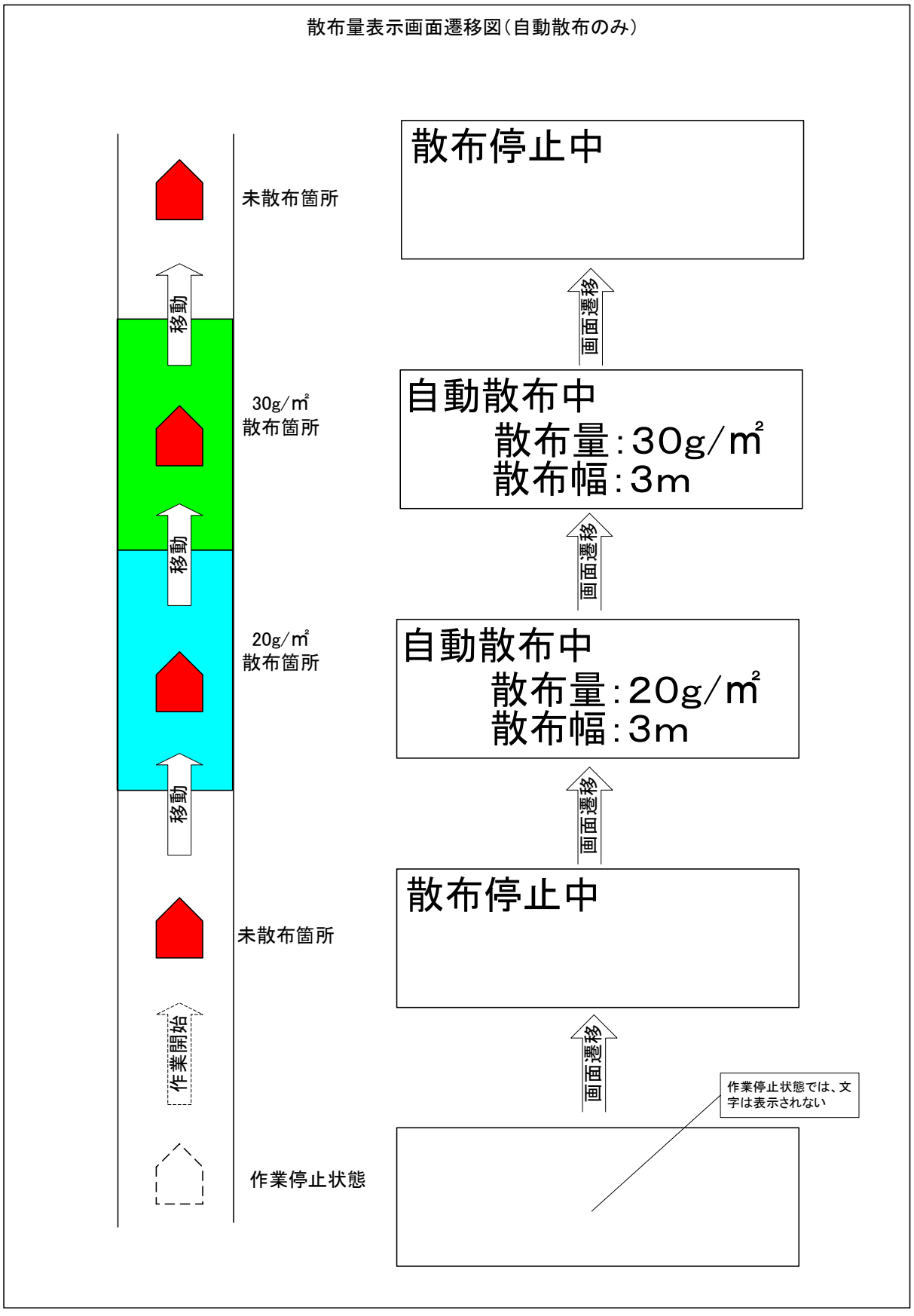


図 2-27　散布量表示画面遷移（自動散布のみ）

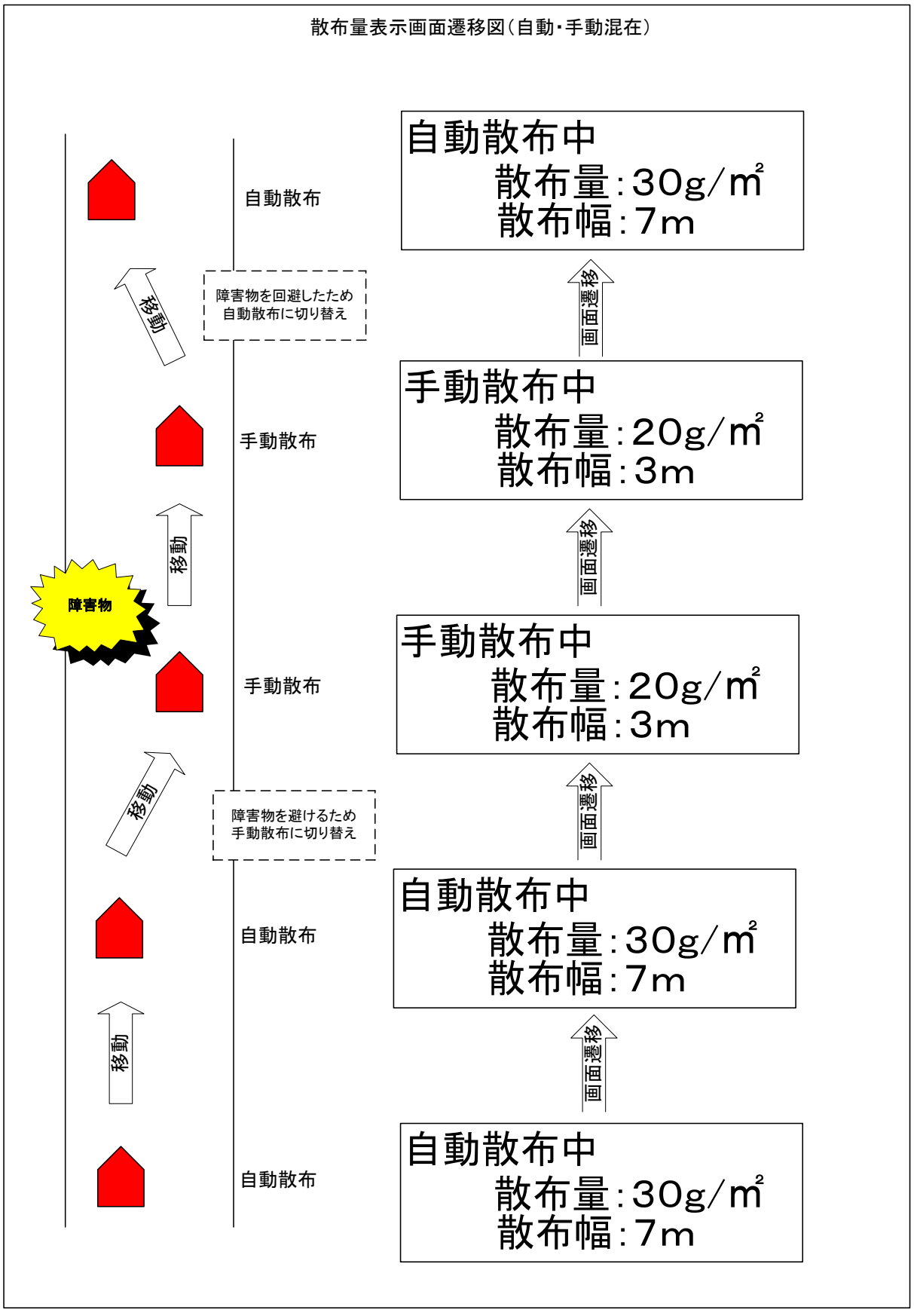


図 2-28　散布量表示画面遷移（自動・手動混在）

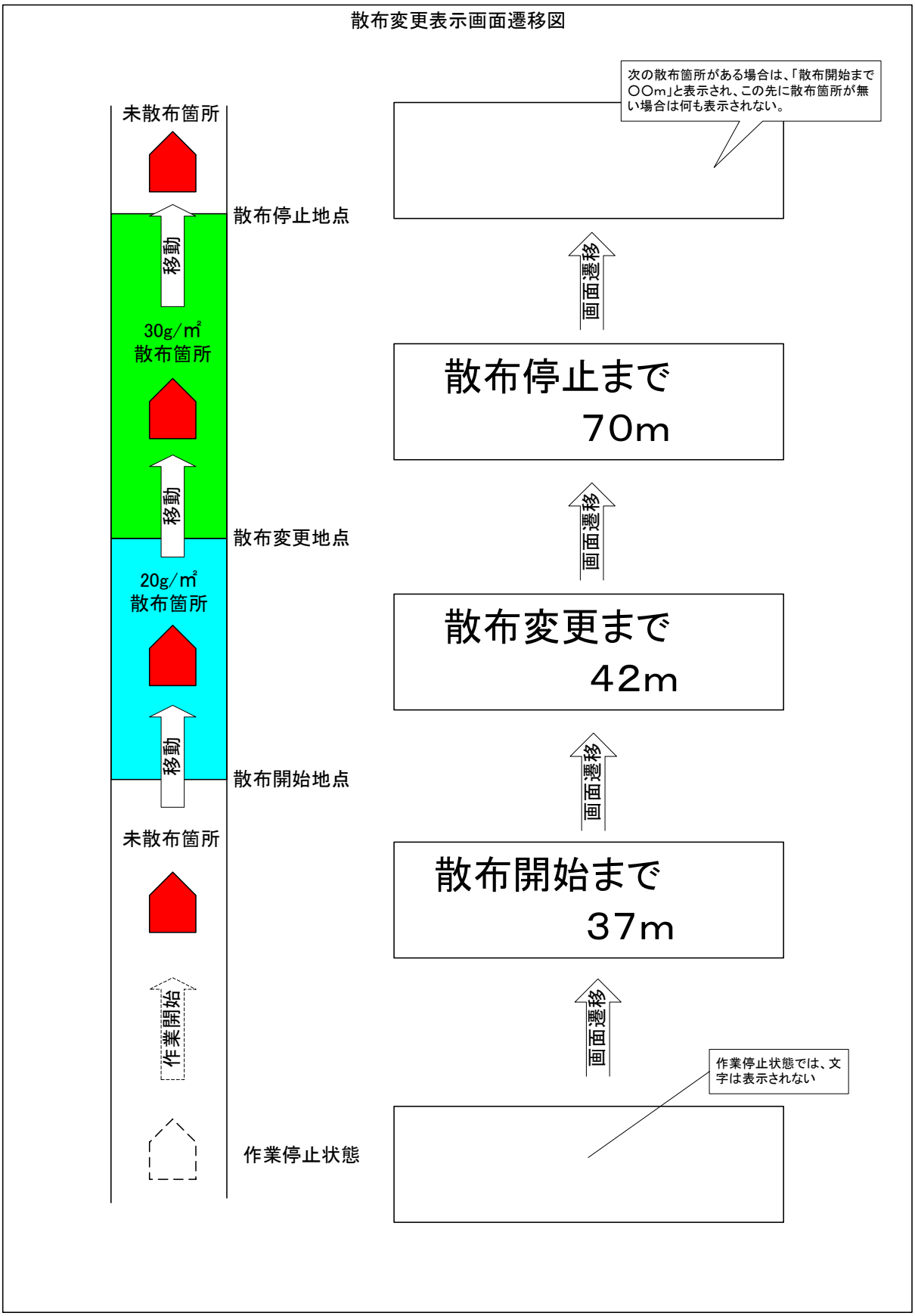


図 2-29　散布変更表示画面遷移

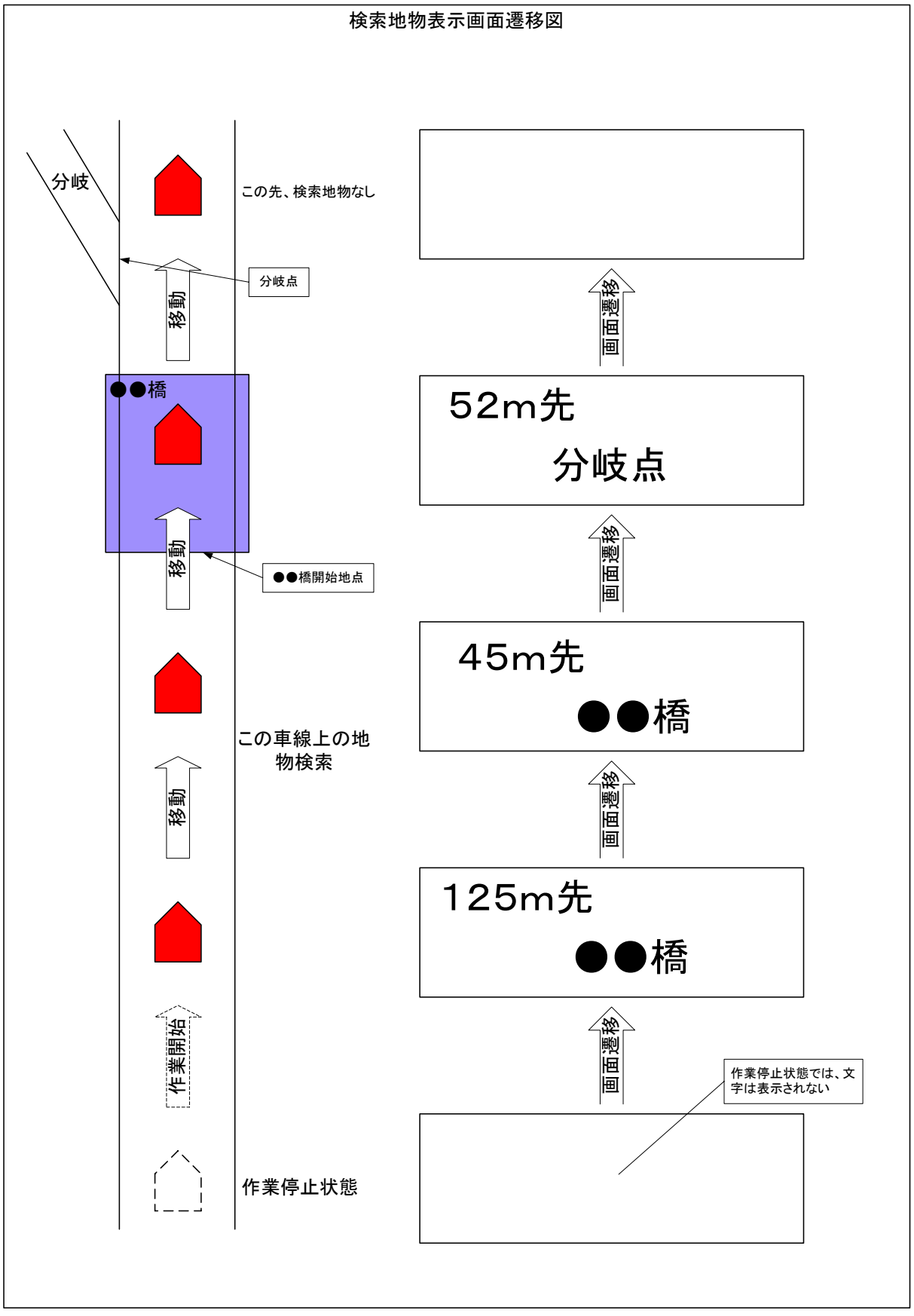


図 2-30　地物検索表示遷移

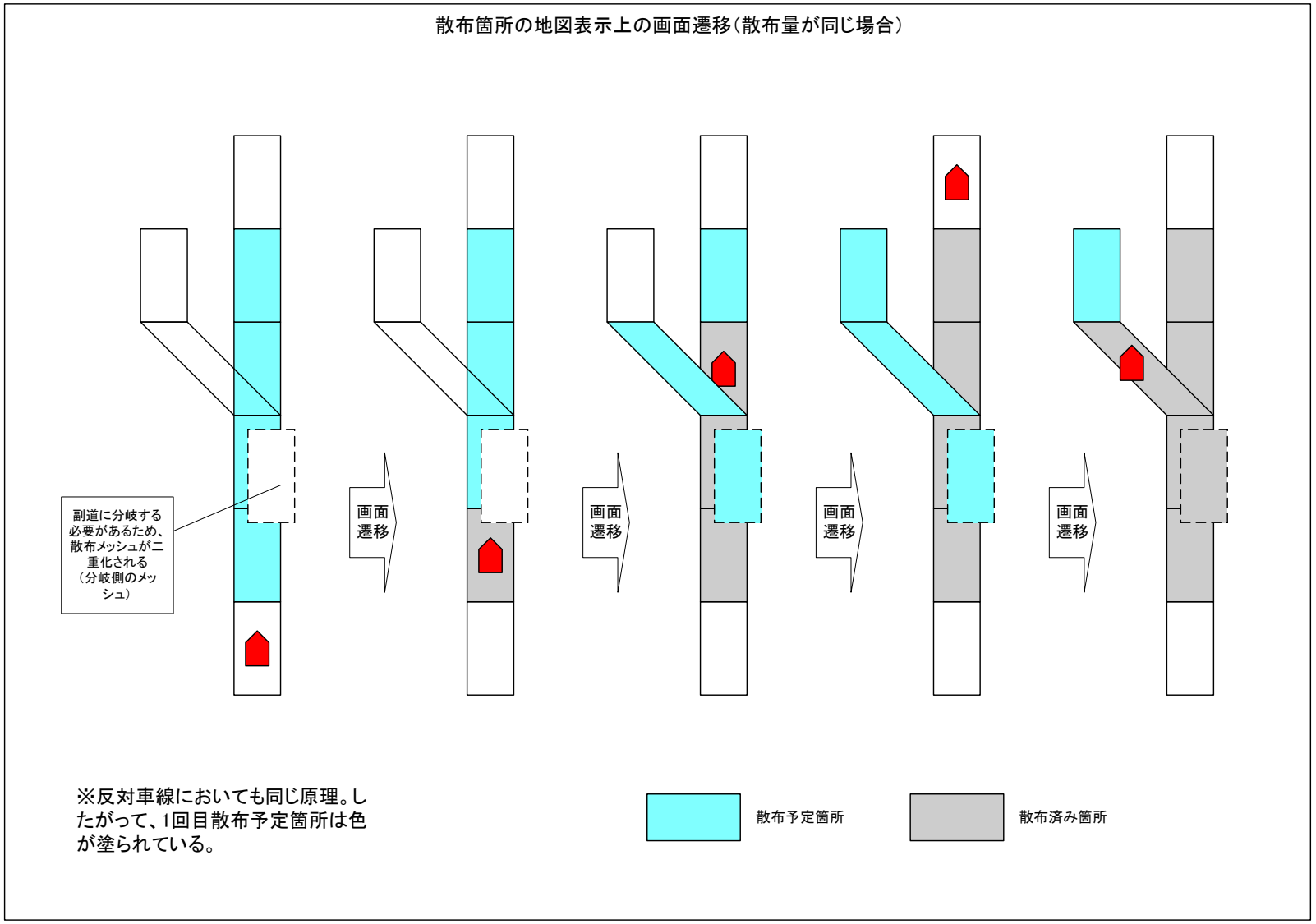


図 2-31　散布箇所表示遷移（散布量が同じ場合）

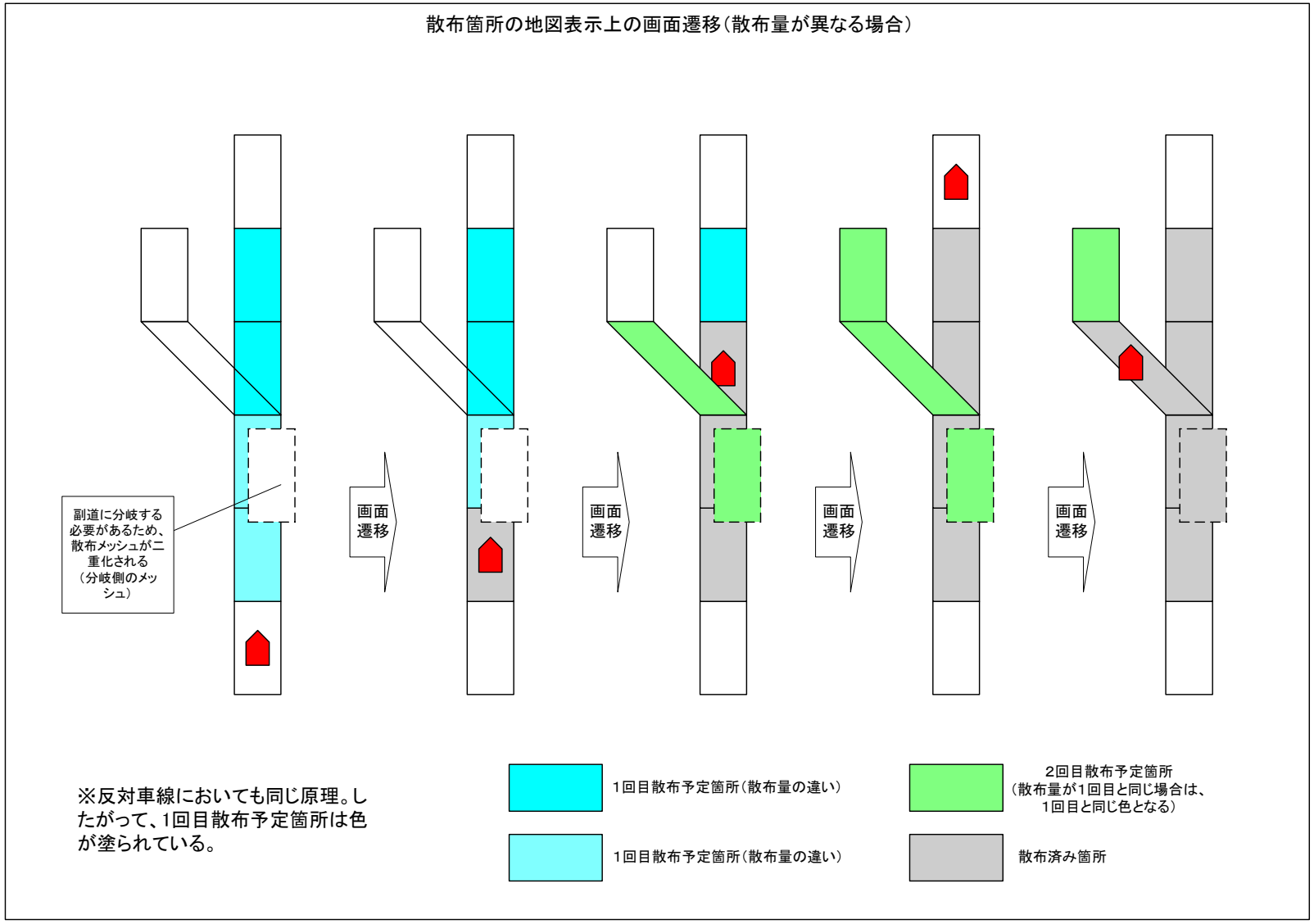


図 2-32　散布箇所表示遷移（散布量が異なる場合）

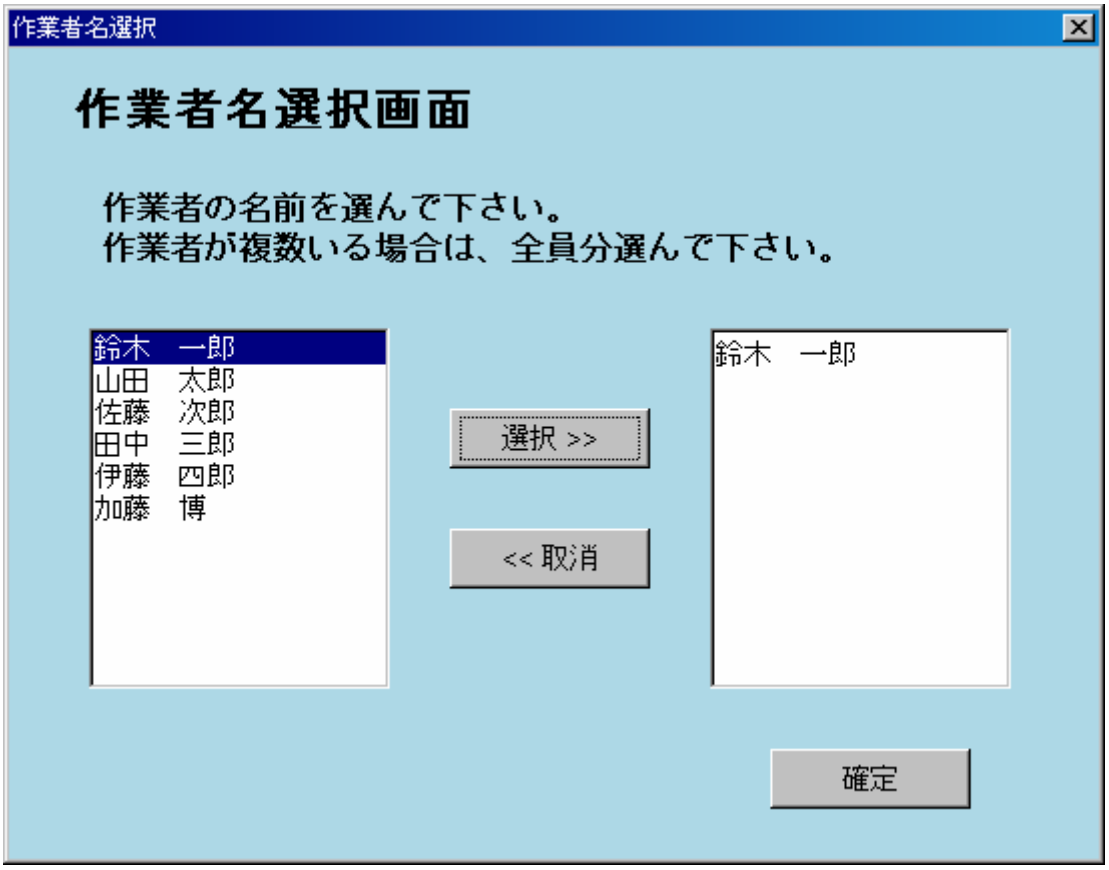


図 2-33　作業者名選択画面

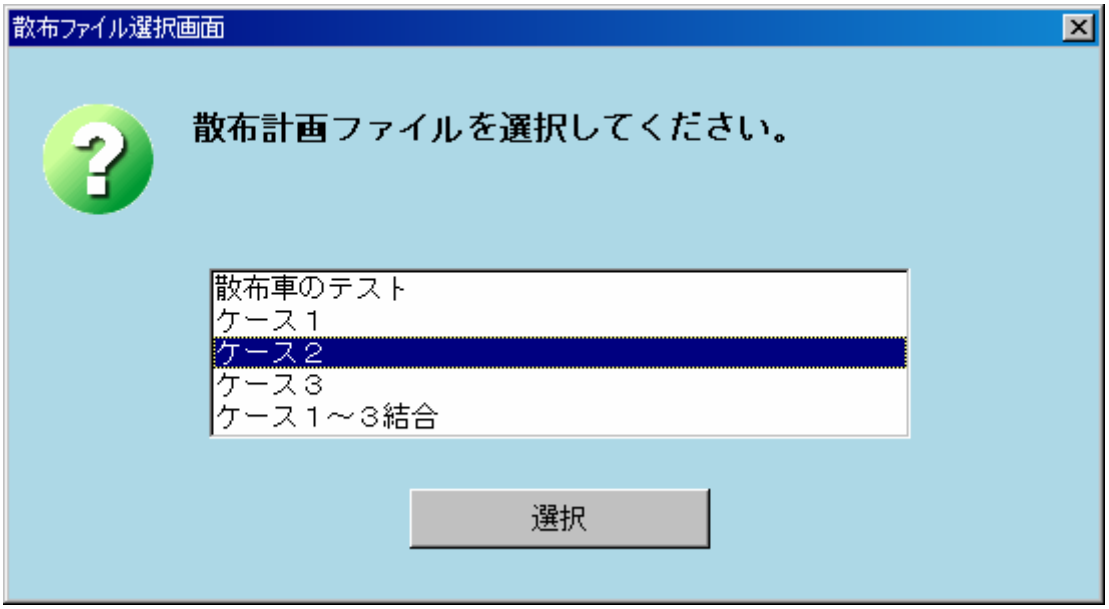


図 2-34　散布計画ファイル選択画面

2.2. 操作支援システムの詳細設計

(1) 適用範囲

本節は、散布操作支援システムにおける詳細設計についてのみ記載するものとする。

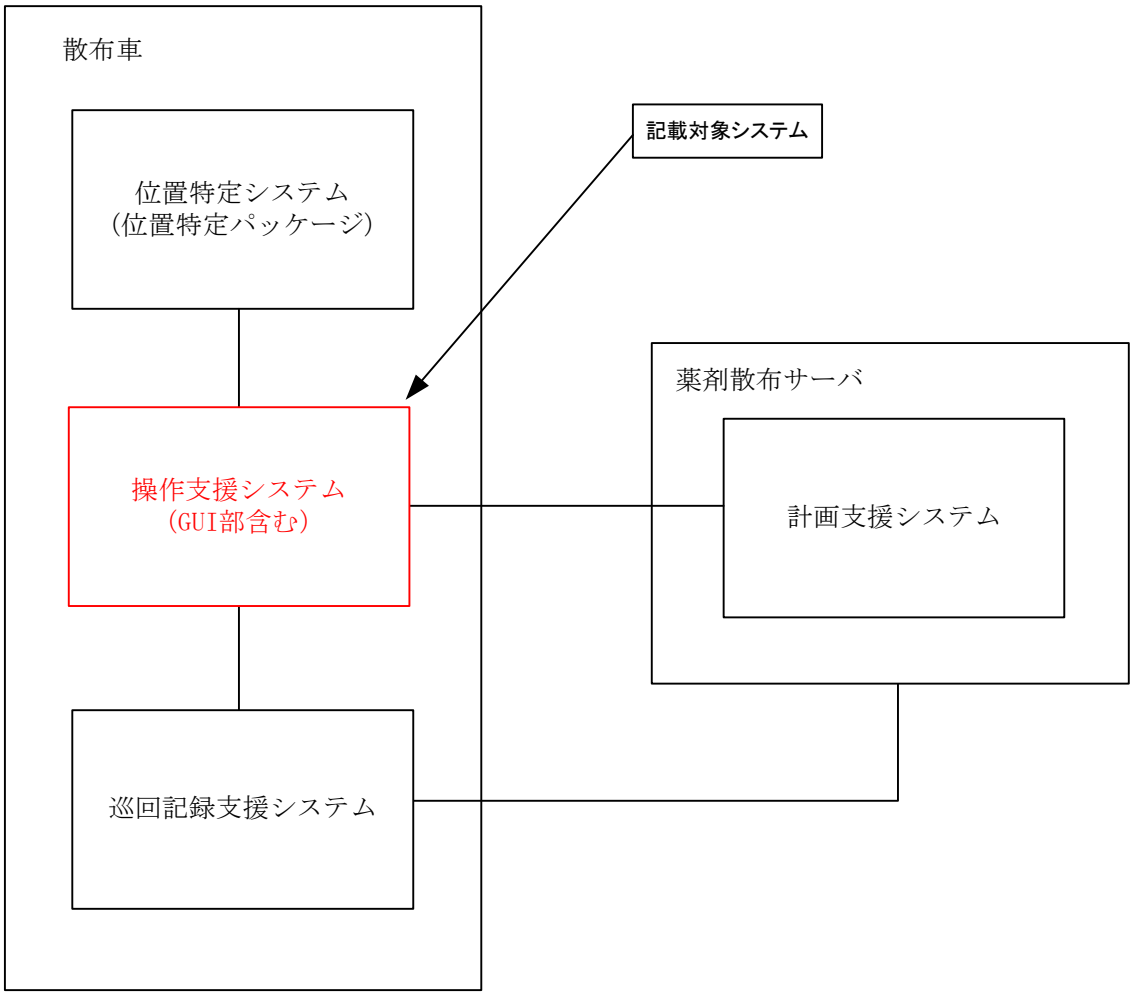


図 2-35　本設計書の記載範囲

(2) ソフトウェア構成

本システムにおけるソフトウェアのパッケージ図、クラス図、シーケンス図を以下に示す。

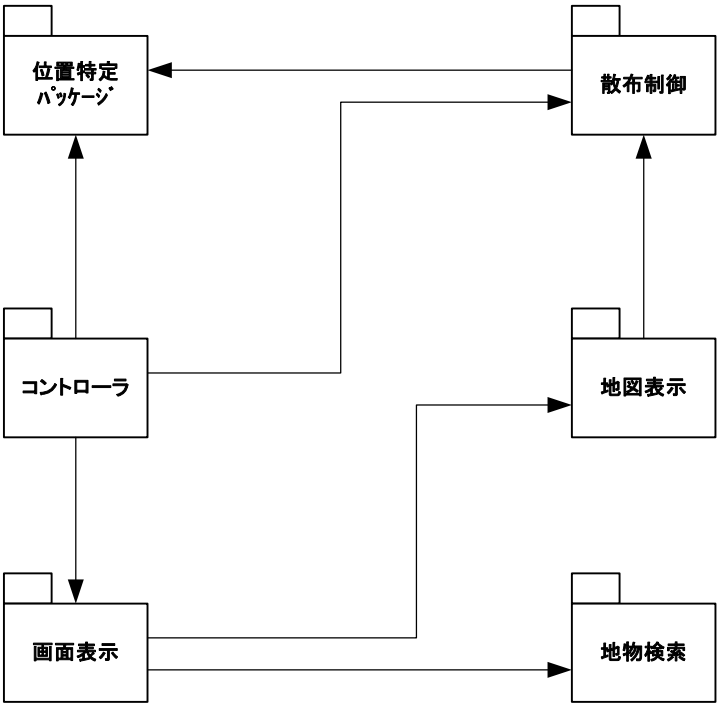


図 2-36　パッケージ図

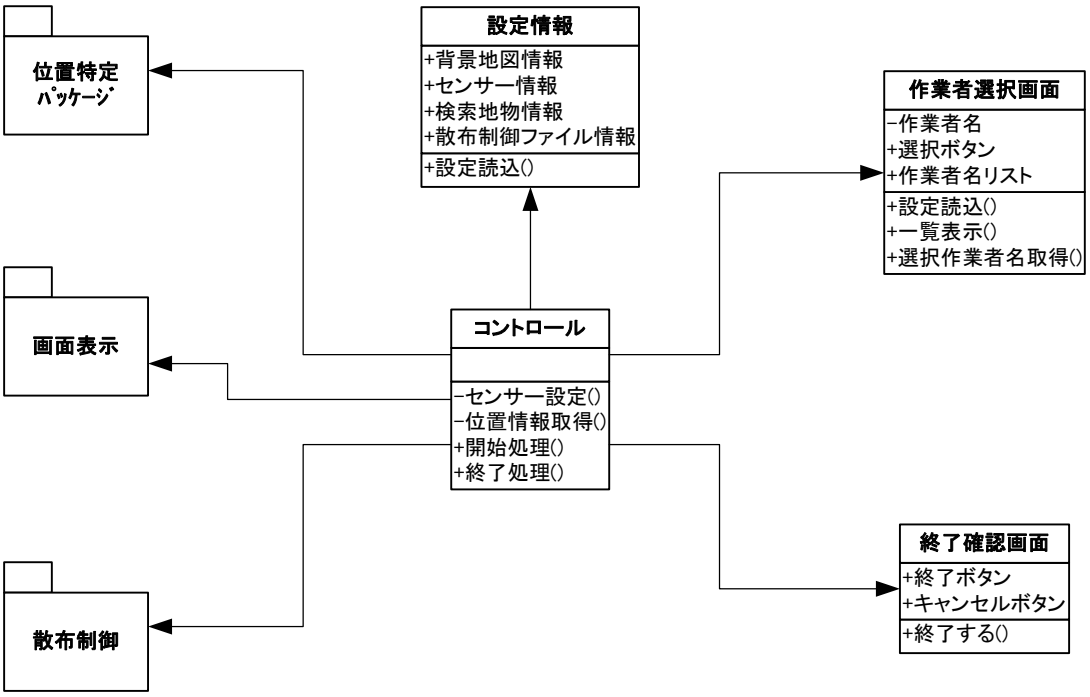


図 2-37　コントローラクラス図

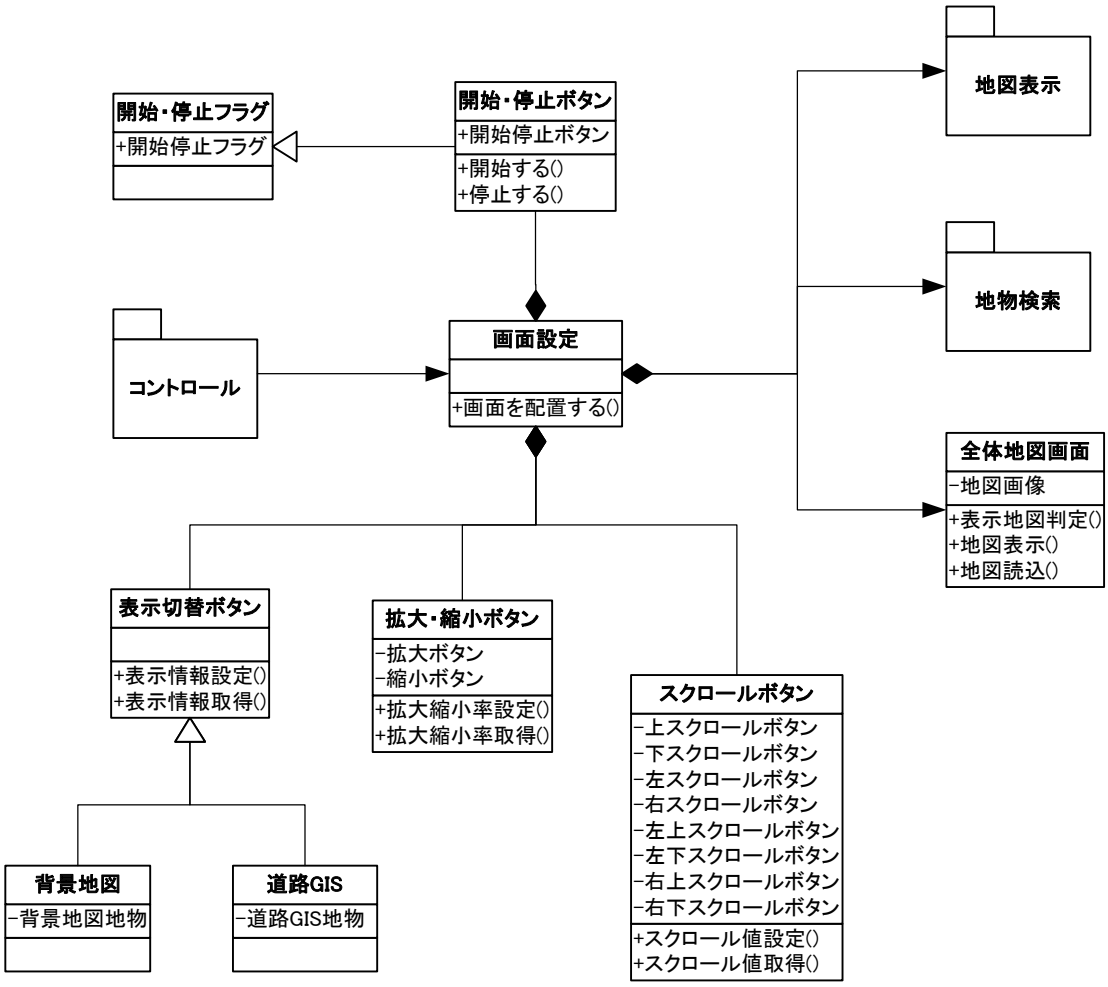


図 2-38　画面表示クラス図

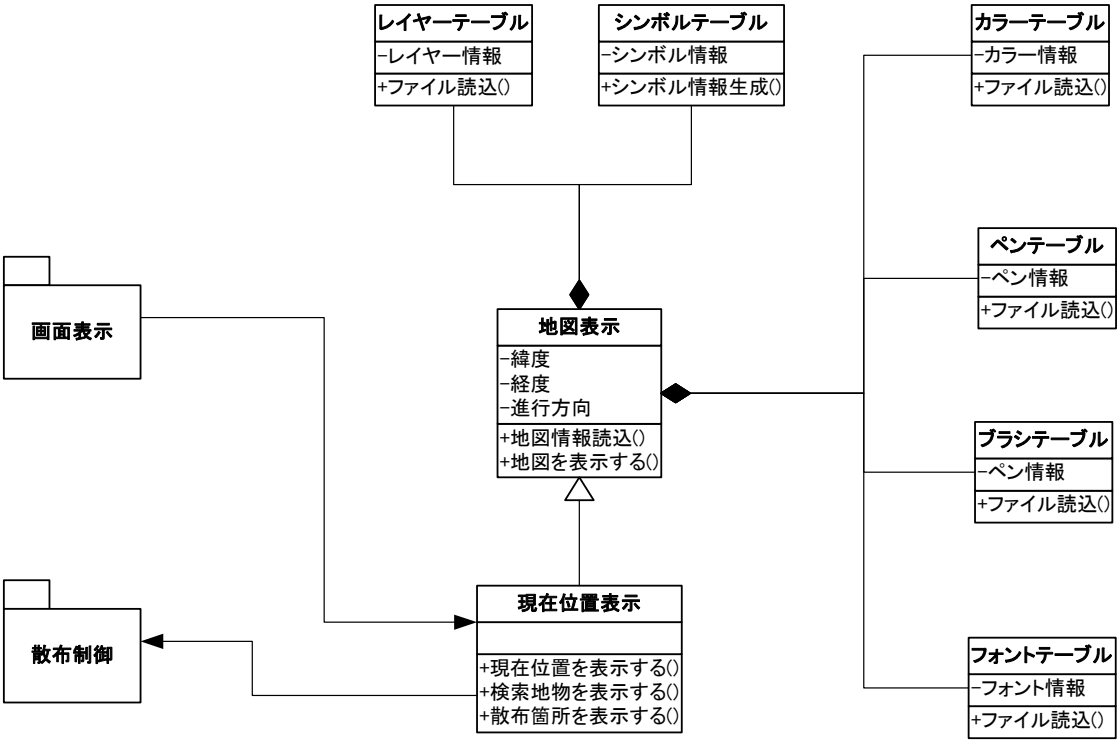


図 2-39　地図表示クラス図

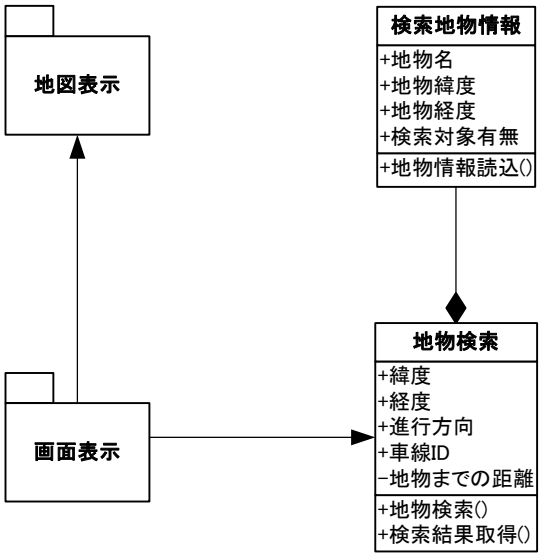


図 2-40　地物検索クラス図

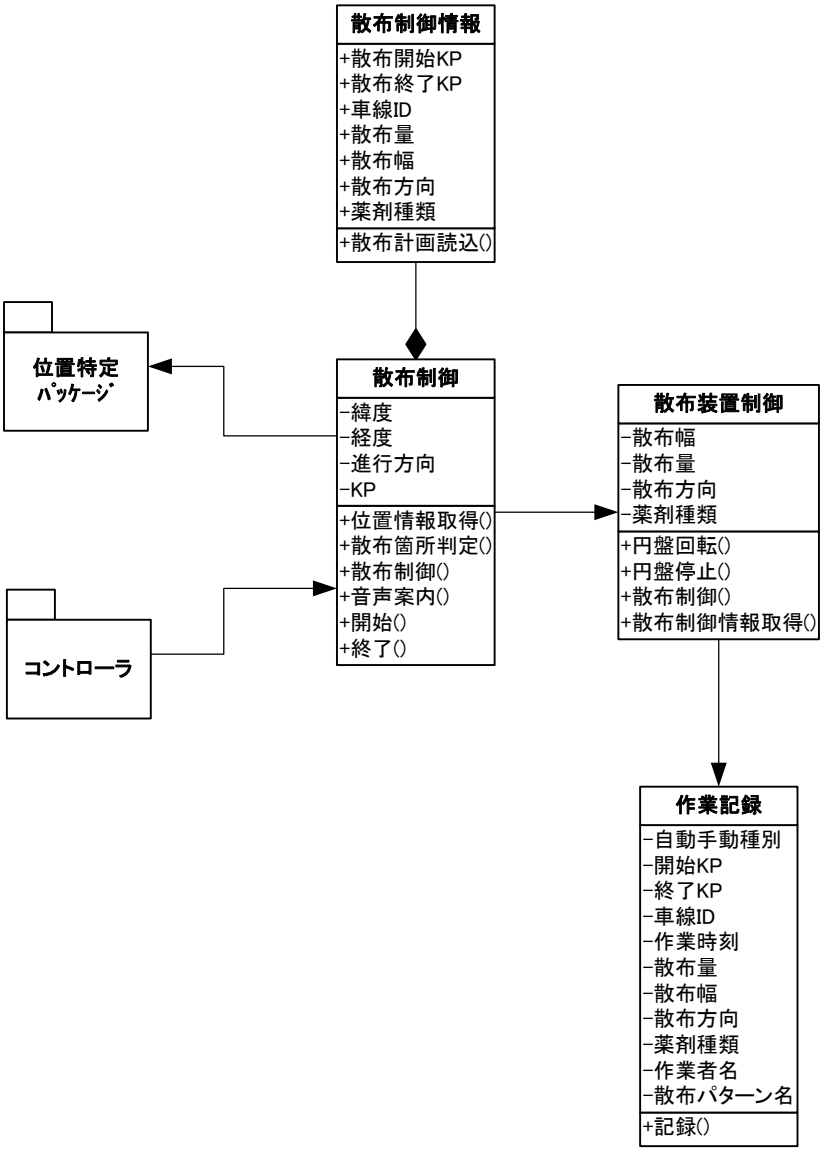


図 2-41　散布制御クラス図

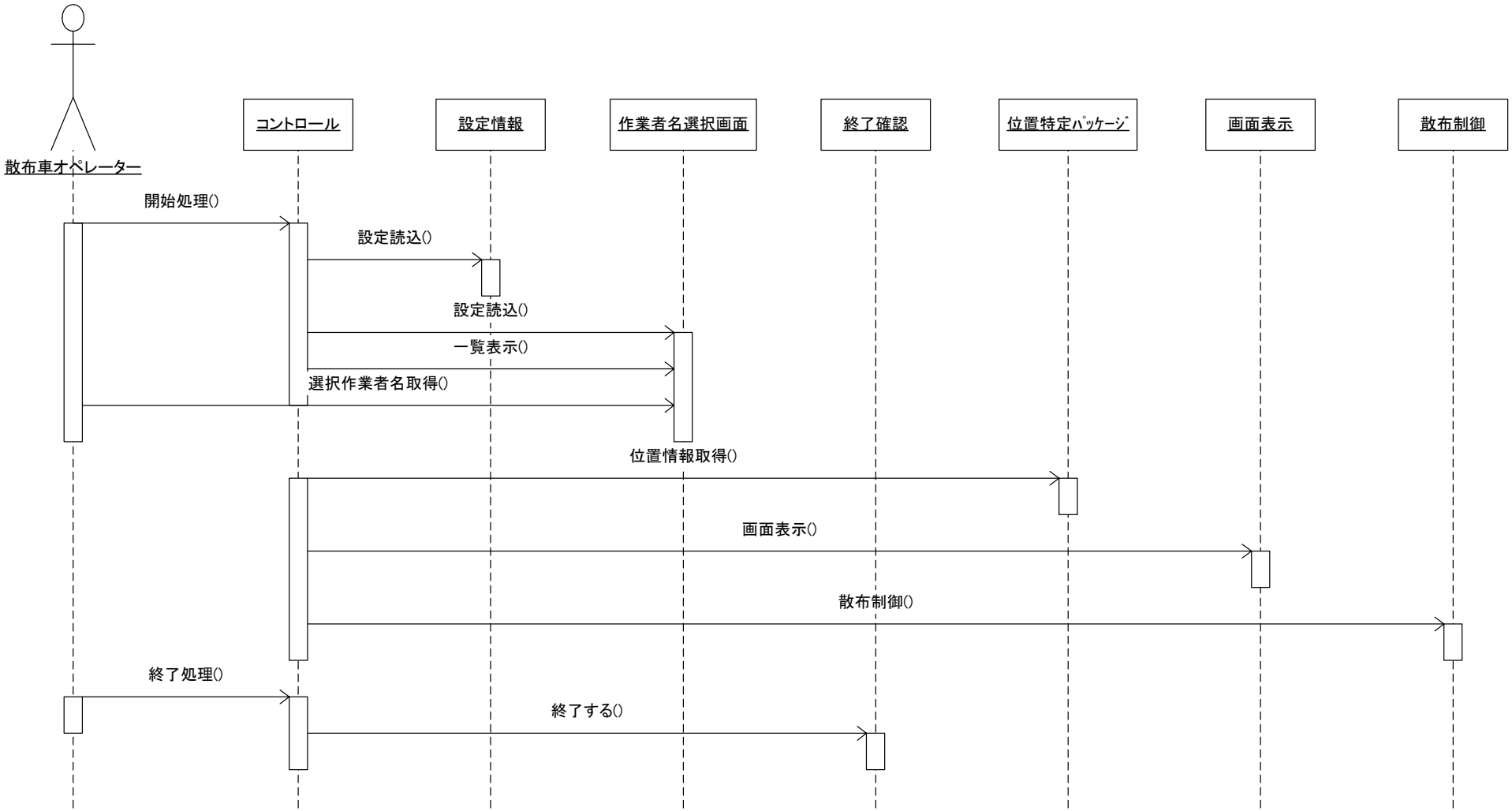


図 2-42　コントローラシーケンス図

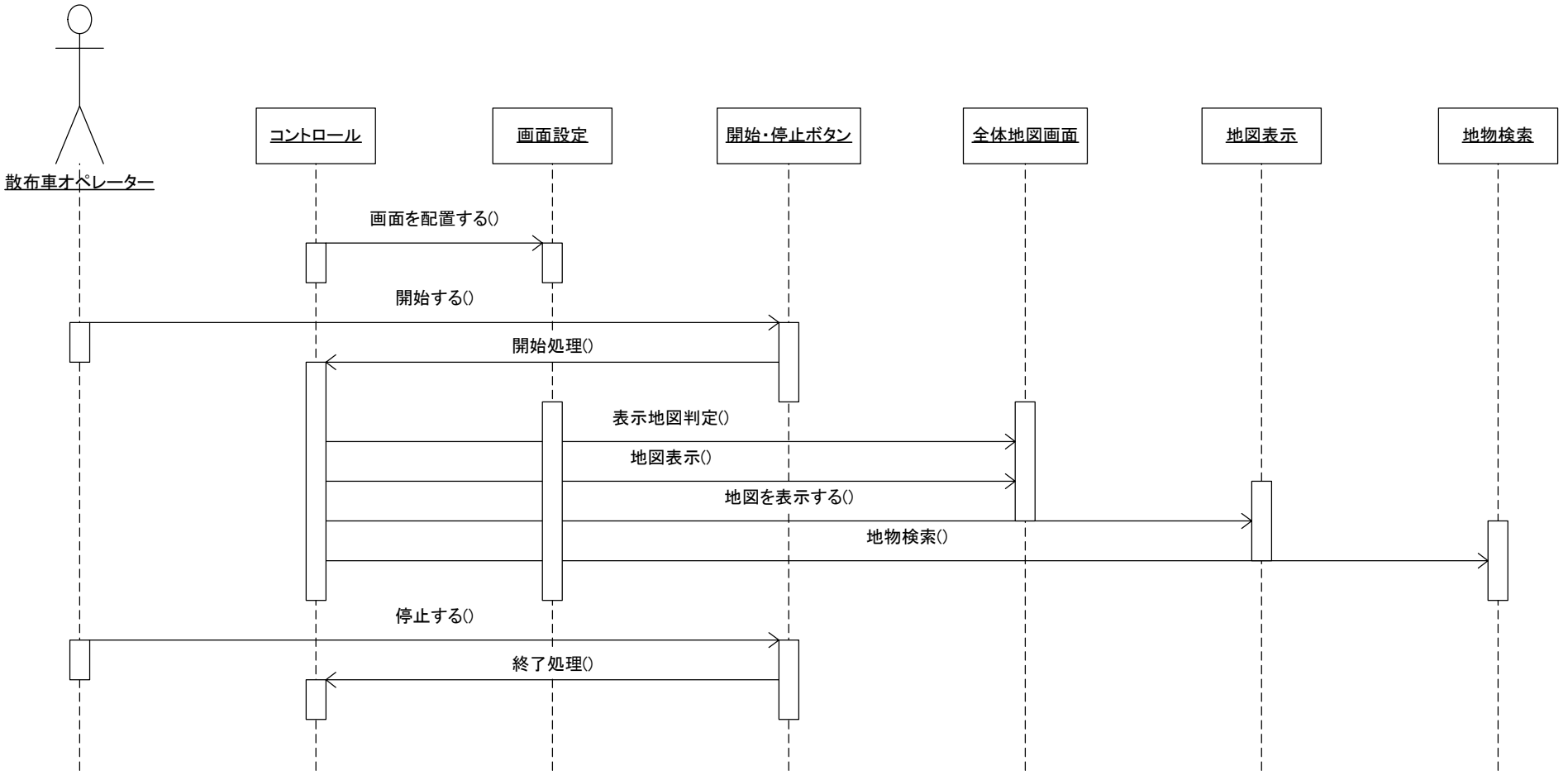


図 2-43　画面表示シーケンス図（その１）

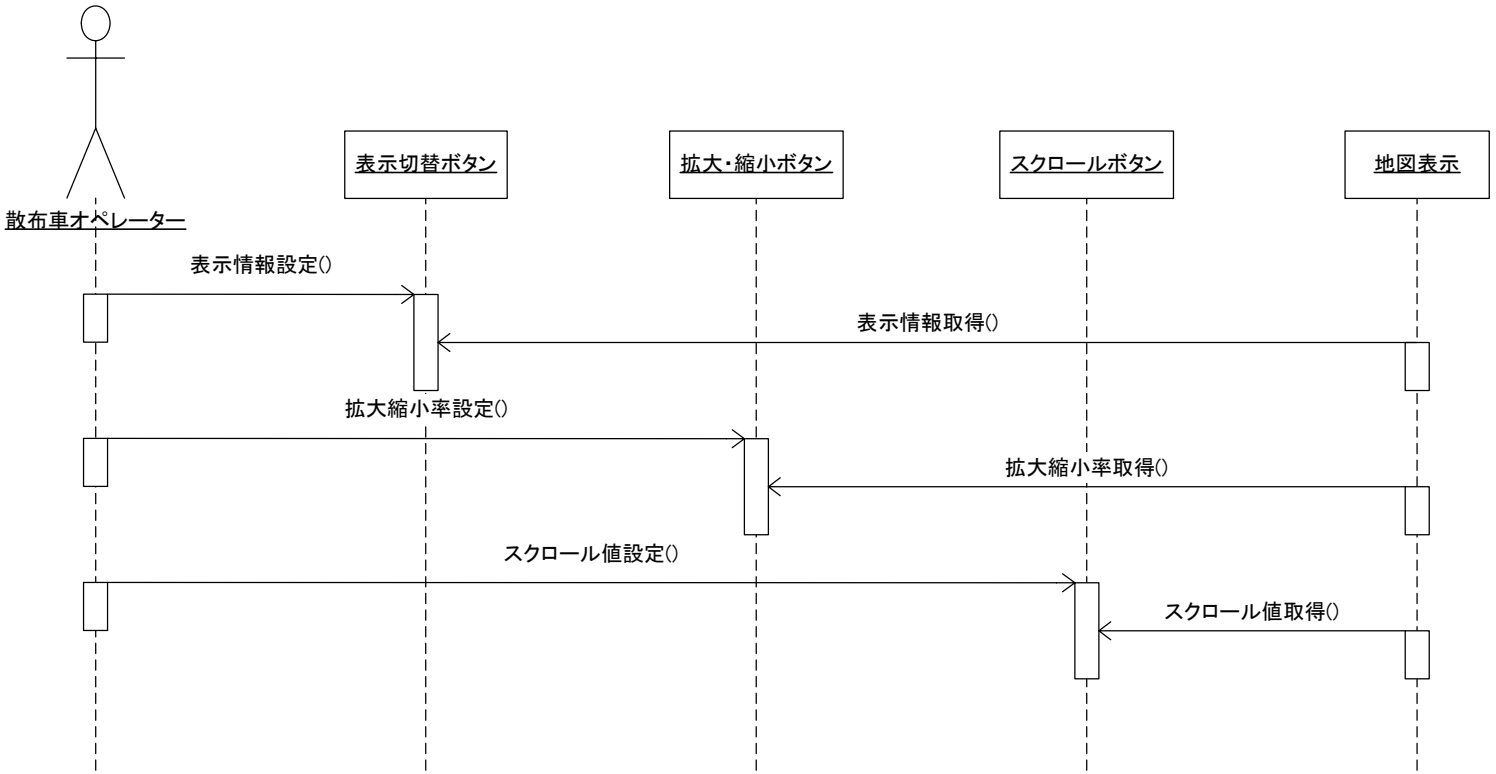


図 2-44　画面表示シーケンス図（その２）

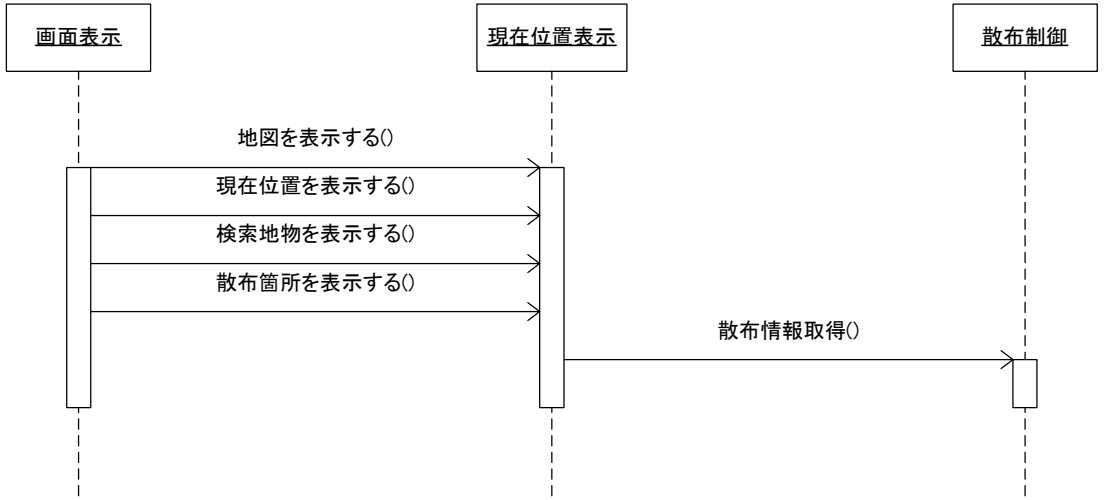


図 2-45　地図表示シーケンス図

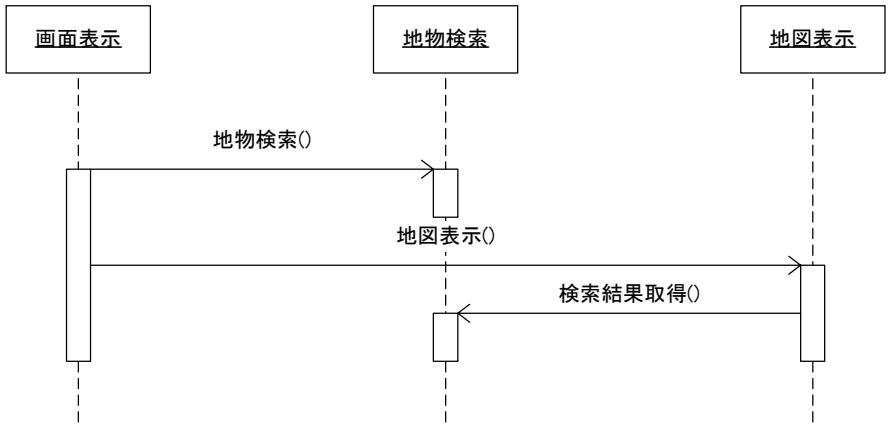


図 2-46　地物検索シーケンス図

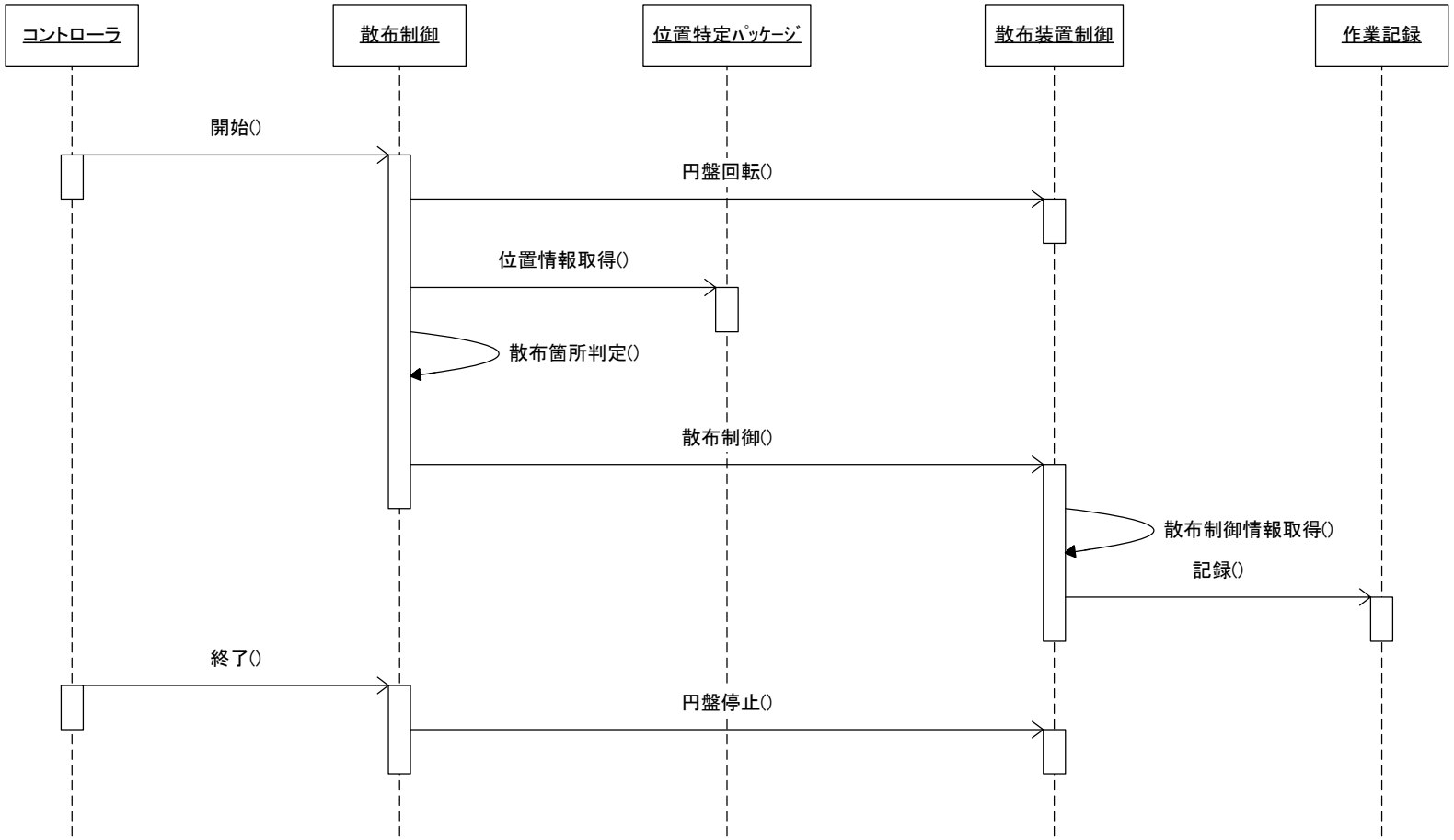


図 2-47　散布制御シーケンス図

表 2-7　コントロールクラスの機能

クラス名	コントロール

入力データ

なし

機能概要

本クラスは、全体コントロールを行うものであり、本システムのメインメソッドを含むものである。

(1) 設定読込コントロール

設定ファイルが指定されている場合は､設定情報クラスを呼び出して設定情報を読み込む｡設定ファイルが指定されていない場合は、ダイアログを表示してファイルや各種設定等を行うダイアログを表示する。

また、作業者選択画面クラスを呼び出し、作業者名選択ダイアログを表示して、作業者名の選択を行う。

(2)各クラスのインスタンス化

位置特定ﾊﾟｯｹｰｼﾞクラス、画面設定クラス、散布制御クラスをインスタンス化して、本システムに必要なオブジェクトを生成する。

(3) タイマー処理

設定ファイルに設定されているインターバル時間毎に、位置情報の取得、地図画面の表示のコントロールを行う。

(4)処理開始終了処理

出力データ

なし

表 2-8　設定情報クラスの機能

クラス名	設定情報

入力データ

○地図表示設定、センサー設定ファイル

機能概要

本クラスは、本システムの制御ファイル名等の記載された、基本設定情報を読み込み、保持する。

(1) 設定読込

地図表示設定、センサー設定ファイルを読み込んで、地図表示、地物検索、散布制御に必要な情報をセットする。

出力データ

なし

表 2-9　作業者選択画面クラスの機能

クラス名	作業者選択画面

入力データ

○作業者名一覧ファイル

○散布車オペレーターのマウス操作

機能概要

本クラスは、本システムを使用する作業者名を選択し、保持する。システム起動時に画面に表示される。

(1) 作業者名一覧ファイル読込、一覧表示

作業者名一覧ファイルを読み込んで、作業者の一覧を選択ダイアログボックスに表示する。

(2) 作業者選択

散布車オペレーターからのマウス入力を受け取り、システム上に作業者名をセットする。

出力データ

なし

表 2-10　終了確認クラスの機能

クラス名	終了確認

入力データ

○散布車オペレータのマウス操作

機能概要

本クラスは、終了確認のダイアログを表示して、終了するかどうかの判断を行う。

(1) ダイアログ表示

「終了」「一旦終了」「終了しない」のボタンを表示する。

(2) 終了処理判定

押されたボタンにより処理を制御する。「終了」が押された場合は、作業記録等のクローズ処理を行ってから終了。「一旦終了」が押された場合は、現在の散布制御状態を維持したまま終了。「終了しない」が押された場合は、終了せず画面および散布制御状態を維持する。

出力データ

なし

表 2-11　画面設定クラスの機能

クラス名	画面設定

入力データ

なし

機能概要

本クラスは、画面をレイアウトし、各クラスを配置する。

(1) 画面配置

画面に表示するクラスをインスタンス化し、レイアウトを行う。レイアウトするクラスは、以下の通りである。

- 開始・停止ボタン
- 拡大・縮小ボタン
- スクロールボタン
- 表示切替ボタン
- 全体地図画面
- 地物検索画面
- 散布制御画面

出力データ

なし

表 2-12　開始・停止フラグクラスの機能

クラス名	開始・停止フラグ

入力データ

なし

機能概要

本クラスは、ナビゲーション及び散布制御の ON、OFF 状態を保持する。

開始・停止ボタンの押下により、開始か停止の状態を保持する。開始・停止ボタンクラスはこのクラスを継承する。

出力データ

なし

表 2-13　開始・停止ボタンクラスの機能

クラス名	開始・停止ボタン

入力データ

○散布車オペレーターのマウス操作

機能概要

本クラスは、開始・停止ボタンの表示を切り換え、開始・停止フラグの状態を変更する。

(1) ボタン表示

「開始」と表示して、ボタンを表示する。

(2) イベント処理

ボタンが押された場合に処理を行う。表示が「開始」の時にボタンが押された場合は、表示を「停止」にし、開始・停止フラグを「開始」にする。表示が「停止」の時にボタンが押された場合は、終了確認画面を表示する。終了確認画面において「終了しない」が押された場合は、表示を「開始」にし、開始・停止フラグを「停止」にする。

出力データ

なし

表 2-14　表示切替ボタンクラスの機能

クラス名	表示切替ボタン

入力データ

○　散布車オペレーターのマウス操作

○　レイヤー設定ファイル

機能概要

本クラスは、地図画面上の表示地物を切り替えるボタン（チェックボックス）の表示と設定を行う。背景地図クラス、道路 GIS クラスは、このクラスを継承して表示地物を実装する。

(1) 表示情報取得

レイヤー地図設定ファイルを読み込んで、地図表示設定を行う

(2) イベント処理

表示・非表示のボタン（チェックボックス）が押された場合、地図上の地物の表示の ON、OFF の切り替えを行う。地図表示クラスが本クラスの状態を読み込んで、地図上の地物の表示・非表示を切り替える。

出力データ

なし

表 2-15　背景地図クラスクラスの機能

クラス名	背景地図クラス

入力データ

○ゼンリン住宅地図レイヤー設定ファイル

機能概要

本クラスは、ゼンリン住宅地図の地物表示情報を保持する。

ゼンリン住宅地図の地物の名前と表示 ON、OFF の状態を保持する。表示が ON でボタン（チェックボックス）が押された時は、OFF に、OFF でボタン（チェックボックス）が押された時は ON に切り替える。

表示切替ボタンクラスを継承する。

出力データ

なし

表 2-16　道路 GIS クラスの機能

クラス名	道路 GIS
入力データ	
○道路 GIS レイヤー設定ファイル	
機能概要	
本クラスは、道路 GIS の地物表示情報を保持する。 道路 GIS の地物の名前と表示 ON、OFF の状態を保持する。表示が ON でボタン（チェックボックス）が押された時は、OFF に、OFF でボタン（チェックボックス）が押された時は ON に切り替える。 表示切替ボタンクラスを継承する。	
出力データ	
なし	

表 2-17　拡大・縮小ボタンクラスの機能

クラス名	拡大・縮小ボタン

入力データ

○散布車オペレーターのマウス操作

機能概要

本クラスは、拡大ボタン、縮小ボタンの表示と、ボタンが押された時のイベント処理を行う。拡大縮小率を保持する。

(1) イベント処理

拡大ボタンが押された時は、拡大縮小率を大きくする。

縮小ボタンが押された時は拡大縮小率を小さくする。

地図表示クラスは、拡大縮小率を読み込んで、地図の表示する大きさを設定する。

出力データ

なし

表 2-18　スクロールボタンクラスの機能

クラス名	スクロールボタン

入力データ

○散布車オペレーターのマウス操作

機能概要

本クラスは、スクロールボタン（上、下、左、右、左上、左下、右上、右下）の表示と、ボタンが押された時のイベント処理を行う。上下方向、左右方向のスクロール値を保持する。

(1) イベント処理

上ボタンが押された時は、上下方向スクロール値を大きくする。

下ボタンが押された時は、上下方向スクロール値を小さくする。

左ボタンが押された時は、左右方向スクロール値を小さくする。

右ボタンが押された時は、左右方向スクロール値を大きくする。

左上ボタンが押された時は、上下方向スクロール値を大きくし、左右方向スクロール値を小さくする。

左下ボタンが押された時は、上下方向スクロール値を小さくし、左右方向スクロール値を小さくする。

右上ボタンが押された時は、上下方向スクロール値を大きくし、左右方向スクロール値を大きくする。

右下ボタンが押された時は、上下方向スクロール値を小さくし、左右方向スクロール値を大きくする。

地図表示クラスは、上下方向スクロール値、左右方向スクロール値を読み込んで、地図

出力データ

なし

表 2-19　クラスの機能

クラス名	全体地図画面

入力データ

○全体地図

機能概要

本クラスは、全体地図画面の表示、切り替えを行う。

(1) 地図読込

全体地図画面の画像ファイルと画像ファイルの範囲情報を読み込む。

(2) 表示地図判定

現在の緯度、経度と画像ファイルの範囲情報より、表示する画像を判定し、表示画面に該当する地図画像ファイルを貼り付ける。

(3) 地図表示

現在の緯度、経度に該当する地図画像を表示する。また、現在位置を赤丸で描画する。

出力データ

なし

表 2-20　地図表示クラスの機能

クラス名	地図表示

入力データ

なし

機能概要

本クラスは、背景地図（ゼンリン住宅地図、道路 GIS）の描画を行う。GeoSQL クラスを継承する。また、現在位置表示クラスのスーパークラス。

(1) 地図情報読込

背景地図（ゼンリン住宅地図、道路 GIS）の PQ ファイルを読み込む

(2) 地図表示

拡大・縮小ボタンクラスに設定されている拡大縮小率、スクロールボタンクラスに設定されている上下スクロール値、左右スクロール値および、緯度、経度、進行方向角により、地図を表示する。地図は、緯度、経度が中心（設定値により上下にスクロール）とし、進行方向角により回転させて表示する。

出力データ

なし

表 2-21　現在位置表示クラスの機能

クラス名	現在位置表示

入力データ

なし

機能概要

本クラスは、背景地図上に、現在位置（自車マーク）、検索対象地物、散布制御メッシュの描画を行う。地図表示クラスを継承する。

(1) 現在位置を表示する

背景地図上の現在の緯度、経度位置に自車マークを表示する。

(2) 検索地物を表示する

背景地図上の対象検索地物の緯度、経度位置に検索地物マークを表示する。検索地物マークは、画面表示切替毎に大きさを変化させ、点滅しているように描画する。

(3) 散布箇所を表示する

背景地図上に散布箇所を表示する。散布箇所は散布量に応じて色を変える。また、散布済み箇所についても色を変えて描画する。

出力データ

なし

表 2-22　レイヤーテーブルクラスの機能

クラス名	レイヤーテーブル

入力データ

○ ゼンリン住宅地図レイヤー設定ファイル

○ 道路 GIS レイヤー設定ファイル

○ 散布箇所、マークレイヤー設定ファイル

機能概要

本クラスは、背景地図の描画カラーを設定するレイヤー情報を保持する。地図表示クラス内で使用される。

(1) ファイル読込

ゼンリン住宅地図レイヤー設定ファイル、道路 GIS レイヤー設定ファイル、散布箇所、マークレイヤー設定ファイルの各設定ファイルを読み込んで、その情報を保持する。

出力データ

なし

表 2-23　シンボルテーブルクラスの機能

クラス名	シンボルテーブル

入力データ

なし

機能概要

本クラスは、背景地図で使用するシンボルを生成する。ただし、本システムにおいては使用されないが、GeoSQL をインスタンス化する際に必要となるので、生成する必要がある。

(1) シンボル情報生成

○、□、◇、△、▽のシンボルを生成する。

出力データ

なし

表 2-24　カラーテーブルクラスの機能

クラス名	カラーテーブル

入力データ

○カラー設定ファイル

機能概要

本クラスは、背景地図用のカラーテーブル情報を保持する。地図表示クラス内で使用される。

(1) ファイル読込

カラー設定ファイルを読み込んで、その情報を保持する。

出力データ

なし

表 2-25　ペンテーブルクラスの機能

クラス名	ペンテーブル

入力データ

○ペン設定ファイル

機能概要

本クラスは、背景地図用のペンテーブル情報を保持する。地図表示クラス内で使用される。

(1) ファイル読込

ペン設定ファイルを読み込んで、その情報を保持する。

出力データ

なし

表 2-26　ブラシテーブルクラスの機能

クラス名	ブラシテーブル

入力データ

○ブラシ設定ファイル

機能概要

本クラスは、背景地図用のブラシテーブル情報を保持する。地図表示クラス内で使用される。

(1) ファイル読込

ブラシ設定ファイルを読み込んで、その情報を保持する。

出力データ

なし

表 2-27　フォントテーブルクラスの機能

クラス名	フォントテーブル

入力データ

○フォント設定ファイル

機能概要

本クラスは、背景地図用のフォントテーブル情報を保持する。地図表示クラス内で使用される。

(1) ファイル読込

フォント設定ファイルを読み込んで、その情報を保持する。

出力データ

なし

表 2-28　地物検索クラスの機能

クラス名	地物検索

入力データ

なし

機能概要

本クラスは、接近地物の検索を行い、地物情報を画面に表示する。

(1) 地物検索

現在の緯度、経度、進行方向から車線を検出し、その車線上で最も近い地物の名称とその地物までの距離を画面に表示する。また、最も近い地物の緯度、経度を保持する。

(2) 検索結果取得

現在位置表示クラスから呼ばれ、最も近い地物の緯度、経度を返す。現在位置表示クラスはこの値により、検索地物を描画する。

出力データ

なし

表 2-29　検索地物情報クラスの機能

クラス名	検索地物情報

入力データ

○検索地物設定ファイル

機能概要

本クラスは、検索地物設定ファイルを読み込み、検索地物の名称、検索対象の有無の情報を保持する。検索地物クラスから呼び出される。

(1) 地物情報読込

検索地物設定ファイルを読み込んで、検索地物の名称、検索対象の有無の情報を保持する。この情報を基に、地物検索クラスは地物の検索を行い、名称の設定を行う。

出力データ

なし

表 2-30　散布制御クラスの機能

クラス名	散布制御

入力データ

なし

機能概要

本クラスは、散布制御の全体コントロールを行う。

(1) 位置情報取得、散布箇所判定

位置特定パッケージより現在の緯度、経度、進行方向を取得し、現在いる箇所が散布箇所かどうかの判定を行う。

(2) 散布制御

現在の散布の状態から散布装置制御クラスに制御を指示する。散布 OFF 状態で散布箇所に入った場合は散布指示、散布 ON 状態で未散布箇所に入った場合は散布 OFF、散布 ON 状態で散布制御情報が変更になった場合は散布変更の指示を散布装置制御クラスへ指示する。

(3) 音声案内

散布が ON から OFF、OFF から ON、散布変更時に音声（ビープ音）を鳴らす。

出力データ

なし

表 2-31　散布制御情報クラスの機能

クラス名	散布制御情報

入力データ

○　散布計画ファイル

○　メッシュファイル

機能概要

本クラスは、散布計画ファイル、メッシュファイルを読み込み、散布制御情報を生成し、保持する。

(1) 散布計画読込

散布計画ファイルとメッシュファイルを読み込んで、散布制御に必要な各メッシュにおける散布量、散布幅、散布方向、薬剤種類をセットする。

散布制御クラスから呼び出される。

出力データ

なし

表 2-32　散布装置制御クラスの機能

クラス名	散布装置制御

入力データ

○散布制御装置からの散布指示データ

機能概要

本クラスは、散布車に搭載されている散布装置の制御を行う。また、散布装置からの実際の散布情報を取得し、作業記録としてファイルに記録する。

(1) 円盤回転、円盤停止

散布制御クラスの指示により、散布車の円盤の回転、停止の指示を行う。

(2) 散布制御

散布制御クラスからの指示により、散布装置に散布量、散布幅、散布方向、薬剤種類の散布情報を送り、散布の指示を行う。

(3) 散布制御情報取得

散布装置から実際の散布情報（散布 ON・OFF、手動・自動の別、散布量、散布幅、散布方向、薬剤種類）を受け取り、作業記録クラスに記録指示を出す。

出力データ

○散布制御装置への散布指示データ

表 2-33　作業記録クラスの機能

クラス名	作業記録

入力データ

○散布装置からの散布データ

機能概要

本クラスは、散布装置からの実際の作業記録をファイルに保存する。

(1) 記録

散布装置制御クラスからの記録指示毎に、散布データをワークファイルに保存する。システム終了時に、ワークファイルの内容を編集して、作業記録ファイルとして、設定されている外部記憶装置またはディスク内に保存する。

出力データ

○作業記録ファイル

(3) ファイル設計

本システムで使用するファイルは、以下のとおりである。

表 2-34　ファイル一覧（その１）

ファイル名	形式	I/O	説　　明
ゼンリン住宅地図	PQ	I	背景地図に表示するゼンリンの住宅地図データ。GeoSQL 用。GeoSQL のツールを用いて変換を行っておく。
道路 GIS	PQ	I	背景地図に表示する道路 GIS データ。GeoSQL 用。GeoSQL のツールを用いて変換を行っておく。
カラー設定	CSV	I	PQ ファイルを用いて表示する背景地図のカラーとカラー番号の設定ファイル。
フォント設定	CSV	I	PQ ファイルを用いて表示する背景地図のフォントとフォント番号の設定ファイル。
ペン設定	CSV	I	PQ ファイルを用いて表示する背景地図のペンとペン番号の設定ファイル。
ブラシ設定	CSV	I	PQ ファイルを用いて表示する背景地図のブラシとブラシ番号の設定ファイル。
ゼンリン住宅地図 レイヤー設定	CSV	I	PQ ファイルを用いて表示するゼンリン住宅地図のレイヤーとレイヤー番号の設定ファイル。
道路 GIS レイヤー設定	CSV	I	PQ ファイルを用いて表示する道路 GIS のレイヤーとレイヤー番号の設定ファイル。
散布箇所、マーク レイヤー設定	CSV	I	背景地図上に表示する散布箇所やマーク等のレイヤーとレイヤー番号の設定ファイル。
作業員名一覧	TXT	I	システム起動時に設定する作業員名の一覧を表示するための、作業名リストを記述したファイル。
地図表示設定	TXT	I	背景地図のデータ範囲や初期表示位置、設定ファイル名の情報等を記述したファイル。
センサー設定	TXT	I	位置特定から位置データを取得する際のセンサーの設定および散布制御ファイルを記述したファイル。

（次ページへ続く）

表 2-35　ファイル一覧（その２）

ファイル名	形式	I/O	説　　明
メッシュ	CSV	I	地物検索および散布制御に用いる道路車線のメッシュファイル
地物検索	CSV	I	地物検索に用いる地物情報のファイル
散布制御	CSV	I	散布制御に用いるファイル
散布結果作業用	TXT	I/O	散布結果を記録するための作業用のファイル
散布結果	CSV	O	散布結果を保存したファイル
全体地図	GIF JPEG	I	全体地図を表示するための画像ファイル

表 2-36　ファイルフォーマット

ファイル名	ゼンリン住宅地図	形式	PQ
説明	背景地図に表示するゼンリンの住宅地図データ。GeoSQL 用。GeoSQL のツールを用いて変換を行っておく。		

フォーマット

GeoSQL 用のため、仕様未公開。

「BaseMapConvertWizard」を用いて、ゼンリン住宅地図データを直接 PQ ファイルに変換して生成する。

表 2-37　ファイルフォーマット

ファイル名	道路 GIS	形式	PQ
説明	背景地図に表示する道路 GIS データ。GeoSQL 用。GeoSQL のツールを用いて変換を行っておく。		

フォーマット

GeoSQL 用のため、仕様未公開。

「道路 GIS　変換ツール」「GB32->GbText　変換ツール」「GbText->PQ　変換ツール」を用いて、道路 GIS データを PQ ファイルに変換して生成する。

表 2-38　ファイルフォーマット

ファイル名	カラー設定	形式	CSV
説明	PQ ファイルを用いて表示する背景地図のカラーとカラー番号の設定ファイル。		

フォーマット

1 行 1 カラーとし、以下の項目をカンマで区切った CSV 形式とする。設定カラー数分の行数が必要。

- カラー番号
- 赤色濃度（0～255）
- 緑色濃度（0～255）
- 青色濃度（0～255）
- 透明度（0～255）
- コメント

※第 1 カラムが “;”（半角）の場合はコメント、空行は無視

表 2-39　ファイルフォーマット

ファイル名	フォント設定	形式	CSV
説明	PQ ファイルを用いて表示する背景地図のフォントとフォント番号の設定ファイル。		

フォーマット

1 行 1 フォントとし、以下の項目をカンマで区切った CSV 形式とする。設定フォント数分の行数が必要。

- フォント番号
- カラー番号
- フォント
- スタイル
- サイズ
- コメント

カラー番号は、カラー設定ファイルの番号

※第 1 カラムが “;”（半角）の場合はコメント、空行は無視

（現システムでは、設定情報を使用していないので、設定を変更しても変化なし。GeoSQL 設定時に必要なため、作成してある）

表 2-40　ファイルフォーマット

ファイル名	ペン設定	形式	CSV
説明	PQ ファイルを用いて表示する背景地図のペンとペン番号の設定ファイル。 ペンとは、線を描画する時に使用する設定のこと。		

フォーマット

1 行 1 ペンとし、以下の項目をカンマで区切った CSV 形式とする。設定ペン数分の行数が必要。

- ペン番号
- カラー番号
- カラー設定タイプ（0:太さ固定　1:太さ指定値）
- 線の太さ(カラー設定タイプが 1 の時有効)
- コメント

カラー番号は、カラー設定ファイルの番号

※第 1 カラムが “;”（半角）の場合はコメント、空行は無視

表 2-41　ファイルフォーマット

ファイル名	ブラシ設定	形式	CSV
説明	PQ ファイルを用いて表示する背景地図のブラシとブラシ番号の設定ファイル。 ブラシとは、面情報を持つデータの塗りつぶしの設定のこと。		

フォーマット

1 行 1 ブラシとし、以下の項目をカンマで区切った CSV 形式とする。設定ブラシ数分の行数が必要。

- ブラシ番号
- カラー番号
- コメント

カラー番号は、カラー設定ファイルの番号

※第 1 カラムが “;”（半角）の場合はコメント、空行は無視

表 2-42　ファイルフォーマット

ファイル名	ゼンリン住宅地図レイヤー設定	形式	CSV
説明	PQ ファイルを用いて表示するゼンリン住宅地図のレイヤーとレイヤー番号の設定ファイル。		

フォーマット

1 行 1 レイヤーとし、以下の項目をカンマで区切った CSV 形式とする。設定レイヤー数分の行数が必要。

- レイヤ番号
- 表示フラグ(0：非表示 1：表示)
- レイヤードフラグ(現在 GeoSQL の仕様では未使用)
- ペン番号
- ブラシ番号
- フォント番号
- シンボル番号
- レイヤ名称

ペン番号は、ペン設定ファイルの番号

ブラシ番号は、ブラシ設定ファイルの番号

フォント番号は、フォント設定ファイルの番号（現システムでは未使用）

シンボル番号は、プログラム内部で生成しているシンボルの番号。以下、実装番号

0： 描画無し

1： ○

2： □

3： ◇

4： △

5： ▽

※第 1 カラムが “;”（半角）の場合はコメント、空行は無視

表 2-43　ファイルフォーマット

ファイル名	道路 GIS レイヤー設定	形式	CSV
説明	PQ ファイルを用いて表示する道路 GIS のレイヤーとレイヤー番号の設定ファイル。		

フォーマット

1 行 1 レイヤーとし、以下の項目をカンマで区切った CSV 形式とする。設定レイヤー数分の行数が必要。

- レイヤ番号
- 表示フラグ(0：非表示 1：表示)
- レイヤードフラグ(現在 GeoSQL の仕様では未使用)
- ペン番号
- ブラシ番号
- フォント番号
- シンボル番号
- レイヤ名称

ペン番号は、ペン設定ファイルの番号

ブラシ番号は、ブラシ設定ファイルの番号

フォント番号は、フォント設定ファイルの番号（現システムでは未使用）

シンボル番号は、プログラム内部で生成しているシンボルの番号。以下、実装番号

0： 描画無し

1： ○

2： □

3： ◇

4： △

5： ▽

※第 1 カラムが “;”（半角）の場合はコメント、空行は無視

表 2-44　ファイルフォーマット

ファイル名	散布箇所、マークレイヤー設定	形式	CSV
説明	背景地図上に表示する散布箇所やマーク等のレイヤーとレイヤー番号の設定ファイル。		

フォーマット

1行1レイヤーとし、以下の項目をカンマで区切った CSV 形式とする。設定レイヤー数分の行数が必要。

- レイヤ番号
- 表示フラグ(0：非表示 1：表示)
- レイヤードフラグ(現在 GeoSQL の仕様では未使用)
- ペン番号
- ブラシ番号
- フォント番号
- シンボル番号
- レイヤ名称

ペン番号は、ペン設定ファイルの番号

ブラシ番号は、ブラシ設定ファイルの番号

フォント番号は、フォント設定ファイルの番号（現システムでは未使用）

シンボル番号は、プログラム内部で生成しているシンボルの番号。以下、実装番号

0： 描画無し

1： ○

2： □

3： ◇

4： △

5： ▽

※第1カラムが “;”（半角）の場合はコメント、空行は無視

表 2-45　ファイルフォーマット

ファイル名	作業員名一覧	形式	TXT
説明	システム起動時に設定する作業員名の一覧を表示するための、作業名リストを記述したファイル。		

フォーマット

作業員の名前を一行 1 人で記述する

例）

山田　太郎

鈴木　一郎

田中　次郎

※第 1 カラムが “;”（半角）の場合はコメント、空行は無視

表 2-46　ファイルフォーマット

ファイル名	地図表示設定（その１）	形式	TXT
説明	背景地図のデータ範囲や初期表示位置、設定ファイル名の情報等を記述したファイル。		

フォーマット

以下の項目を順番に、書式に従って記述する。項目の抜けや順番違いの場合エラーとなるので注意。

- 地図データ PQ ファイル配置ディレクトリ
  PQ ファイルの格納されているディレクトリを指定する
- 地物データ PQ ファイル配置ディレクトリ
  PQ ファイルの格納されているディレクトリを指定する
- カラー設定ファイル
- ブラシ設定ファイル
- ペン設定ファイル
- フォント設定ファイル
- レイヤー設定ファイル（ゼンリン住宅地図用）
- レイヤー設定ファイル（道路 GIS 用）
- レイヤー設定ファイル（散布メッシュ、マーク等）
- 実地図領域における原点 X,Y,終点 X,Y（ゼンリン住宅地図用）
  原点Ｘ　，　原点Ｙ　，　終点Ｘ　，　終点Ｙ　の４つの実座標（平面直角座標）をカンマ（,）で区切って入力
- 実地図領域における原点 X,Y,終点 X,Y（道路 GIS 用）
  原点Ｘ　，　原点Ｙ　，　終点Ｘ　，　終点Ｙ　の４つの実座標（平面直角座標）をカンマ（,）で区切って入力
- PQ ファイル領域における原点 X,Y,終点 X,Y（ゼンリン住宅地図用）
  原点Ｘ　，　原点Ｙ　，　終点Ｘ　，　終点Ｙ　の４つの PQ ファイル座標をカンマ（,）で区切って入力
- PQ ファイル領域における原点 X,Y,終点 X,Y（道路 GIS 用）
  原点Ｘ　，　原点Ｙ　，　終点Ｘ　，　終点Ｙ　の４つの PQ ファイル座標をカンマ（,）で区切って入力
- メッシュ分割数（ゼンリン住宅地図用）
  Ｘ方向メッシュ数　，　Ｙ方向メッシュ数　の２つの数をカンマ（,）で区切って入力
- メッシュ分割数（道路 GIS 用）
  Ｘ方向メッシュ数　，　Ｙ方向メッシュ数　の２つの数をカンマ（,）で区切って入力
- 初期表示領域サイズにおける原点 X,Y,終点 X,Y
  原点Ｘ　，　原点Ｙ　，　終点Ｘ　，　終点Ｙ　の４つの実座標（平面直角座標）をカンマ（,）で区切って入力

表 2-47　ファイルフォーマット

ファイル名	地図表示設定（その２）	形式	TXT
説明	背景地図のデータ範囲や初期表示位置、設定ファイル名の情報等を記述したファイル。		

フォーマット

- 全体地図ファイル数
- 全体地図ファイル名

  全体地図ファイル数分の行数入力。例えば、全体地図数が８の場合、８行入力する
- 全体地図の実地図領域における原点X,Y,終点X,Y

  原点Ｘ　，　原点Ｙ　，　終点Ｘ　，　終点Ｙ　の４つの実座標（平面直角座標）をカンマ（,）で区切って入力

  全体地図ファイル数分の行数入力。例えば、全体地図数が８の場合、８行入力する
- 表示中心変位

  0.0の場合画面中央を中心に地図を表示。1.0で画面の下を中心、-1.0で画面の上を中心。

  -1.0～1.0の間の数値を指定
- スケール設定（拡大縮小時変化率、等倍は 1.0），自車位置の表示スケール，探査地物の表示スケール

  ３つのスケール値をカンマ（,）で区切って入力

  スケール設定は、拡大・縮小ボタンを押した時の拡大・縮小率

  自車位置の表示スケールは、地図上に表示する車両のマークの大きさ

  探査地物の表示スケールは、地図上に表示する探査地物の大きさ
- スクロール設定（１回の移動距離）

  スクロールボタンが押された時の１回のスクロール量
- 作業記録ファイル作業用ファイル（HD用）

  ディレクトリは事前に作成しておく必要あり
- 作業記録ファイル名（CF用）

  CFのドライブ名もいっしょに指定する
- 地物検索データファイル数
- 地物検索データファイル（データファイル数分の行数入力）

  地物検索ファイル数分の行数入力。例えば、検索地物ファイル数が６の場合、６行入力する
- 地物検索および散布制御範囲判定用メッシュファイル
- 作業員名一覧ファイル

表 2-48　ファイルフォーマット

ファイル名	センサー設定	形式	TXT
説明	位置特定から位置データを取得する際のセンサーの設定および散布制御ファイルを記述したファイル。		

フォーマット

以下の項目を順番に、書式に従って記述する。項目の抜けや順番違いの場合エラーとなるので注意。

- 地図表示設定ファイル名
- 使用センサーの配列番号（複数使用できる場合の配列番号）；

GPS センサ　，　自律センサ　，　車速センサ　，　レーンマーカセンサ　の配列番号をカンマ(,)で区切って入力

- 位置計算処理ロジック

停止判定　，　ドリフト補正　，　パルススケール補正　，　姿勢補正　，　方位算出　，　速度算出　，

マルチパス補正　，　マーカ補正　の計算方法をカンマ（,）で区切って入力

- センサーデータ取得インターバル

設定値は、0:10Hz 1:5Hz 2:2Hz 3:1Hz 4:0.5Hz

- 背景地図表示インターバル

0:システム依存、それ以外は秒

- 散布制御装置接続 RS-232C ポート名

例）　COM1

- 散布制御ファイル数
- 散布制御ファイル名

散布制御ファイル数分の行数入力。例えば、散布制御ファイル数が 5 の場合、5 行入力する

表 2-49　ファイルフォーマット

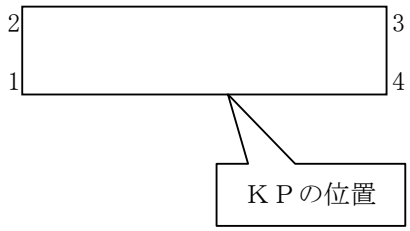
ファイル名	メッシュ	形式	CSV
説明	地物検索および散布制御に用いる道路車線のメッシュファイル		

フォーマット

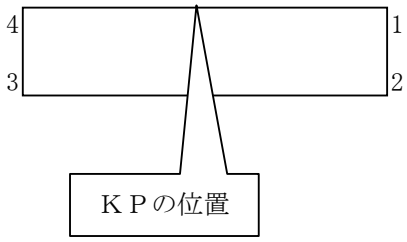
1 行 1 メッシュとし、以下の項目をカンマで区切った CSV 形式とする。メッシュ数分の行数が必要。

- メッシュ ID
- 車線 ID
- KP
- x1
- y1
- x2
- y2
- x3
- y3
- x4
- y4

メッシュＩＤ＞０（上り）



メッシュＩＤ＜０（下り）



上りメッシュと下りメッシュでＫＰの位置が異なるので注意

表 2-50　ファイルフォーマット

ファイル名	地物検索	形式	CSV
説明	地物検索に用いる地物情報のファイル。地物毎に別ファイルとすることによって、画面から検索対象の設定変更が可能。		

フォーマット

1 行 1 地物とし、以下の項目をカンマで区切った CSV 形式とする。地物数分の行数が必要。

- 地物の経度（世界測地系）
- 地物の緯度（世界測地系）
- 地物名称

表 2-51　ファイルフォーマット

ファイル名	散布制御	形式	CSV
説明	散布制御に用いるファイル。支援システムによって生成されるデータファイル。		

フォーマット

1行目：散布標準パターン名

2行目以降：以下の項目をカンマで区切ったCSV形式とする。散布メッシュ数分の行数が必要。

- 車線ID
- 開始KP
- 終了KP
- 散布量
- 散布幅
- 散布方向（0:左 1:右）
- 湿式割合（現散布車では0で固定）
- 薬剤種類（1:フレーク 2:粒 3:粒大 4:原塩）
- 散布順

（注意事項）

車線IDが（＋）の場合は、必ず　開始KP＞終了KP

車線IDが（―）の場合は、必ず　開始KP＜終了KP　とする。

２車線同時散布の場合は、散布順を同じにする。

※空行または1カラム目が；の場合コメントとみなす。

表 2-52　ファイルフォーマット

ファイル名	散布結果作業用	形式	TXT
説明	散布結果を記録するための作業用のファイル		

フォーマット

1行目：作業者名

2行目：標準散布パターン名

3行目以降：以下の項目をカンマで区切った CSV 形式とする。散布変更数分の行数が必要。

- 開始日時(yyyy/MM/dd HH:mm:ss)
- 自動・手動種別（0:自動 1:手動）
- 動作（0:散布停止 1:散布待機 2:散布）
- 車線 ID
- 開始 KP
- 散布量
- 散布幅
- 散布方向（0:左 1:右）
- 湿式割合（現散布車では 0 で固定）
- 薬剤種類（1:フレーク 2:粒 3:粒大 4:原塩）

表 2-53　ファイルフォーマット

ファイル名	散布結果	形式	CSV
説明	散布結果を保存したファイル		

フォーマット

1行目：作業者名

2行目：標準散布パターン名

3行目：開始日時

4行目：終了日時

5行目以降：以下の項目をカンマで区切った CSV 形式とする。散布変更数分の行数が必要。

- 日時
- 自動・手動種別（0:自動 1:手動）
- 車線 ID
- 開始 KP
- 終了 KP
- 散布量
- 散布幅
- 散布方向（0:左 1:右）
- 湿式割合（現散布車では 0 で固定）
- 薬剤種類（1:フレーク 2:粒 3:粒大 4:原塩）

表 2-54　ファイルフォーマット

ファイル名	全体地図	形式	GIF, JPEG
説明	全体地図を表示するための画像ファイル		
フォーマット			

280×180 ドットの GIF または JPEG

左下、右上の座標（平面直角座標）を地図表示設定ファイルに記述する。画像ファイルは、範囲が大きい場合や地図を大きく表示したい場合に対応するため複数指定可。ただし、枚数は縦枚数×横枚数分とする（例　縦２枚、横３枚の場合は６枚）。

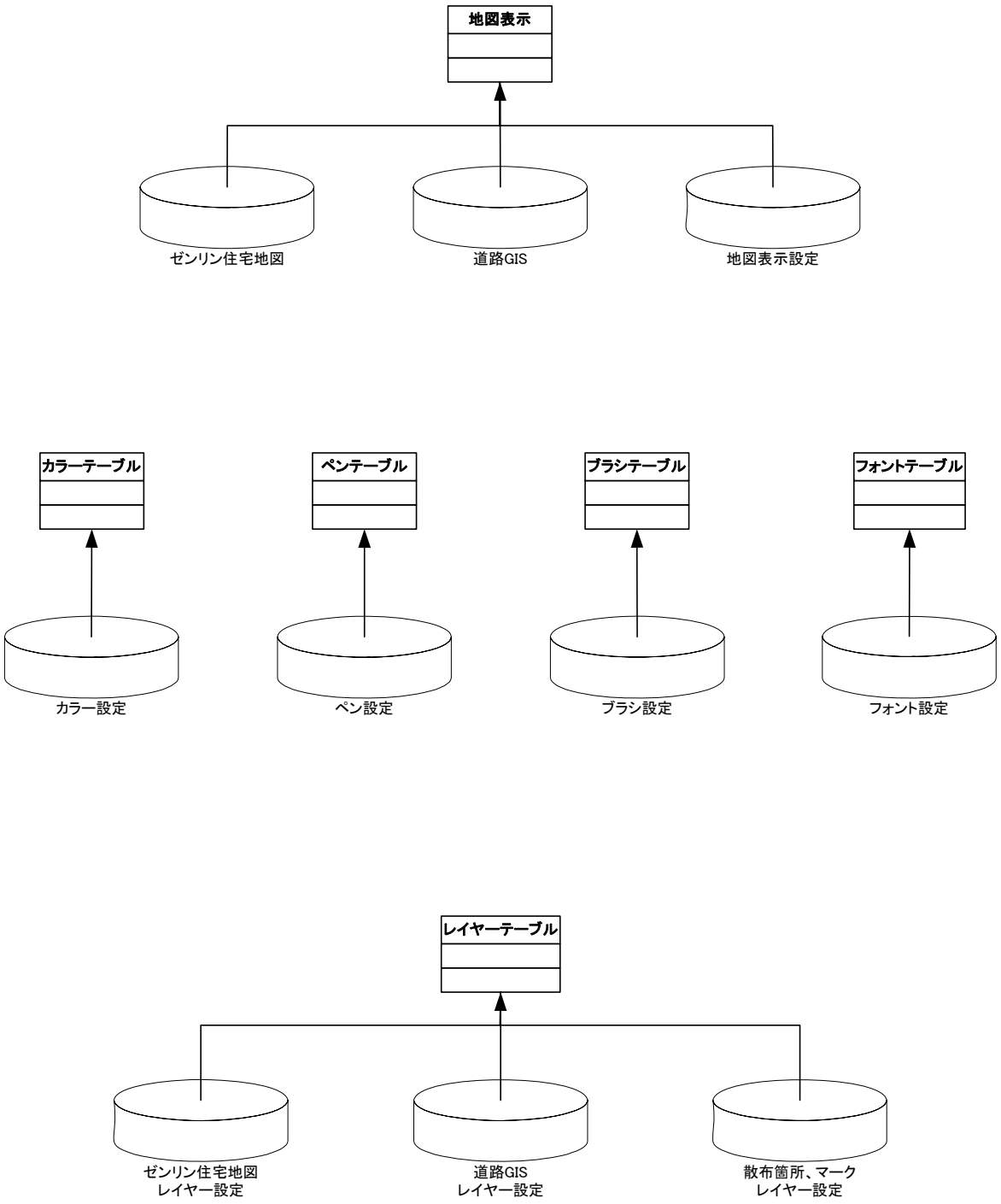


図 2-48　クラスファイル対応図（その１）

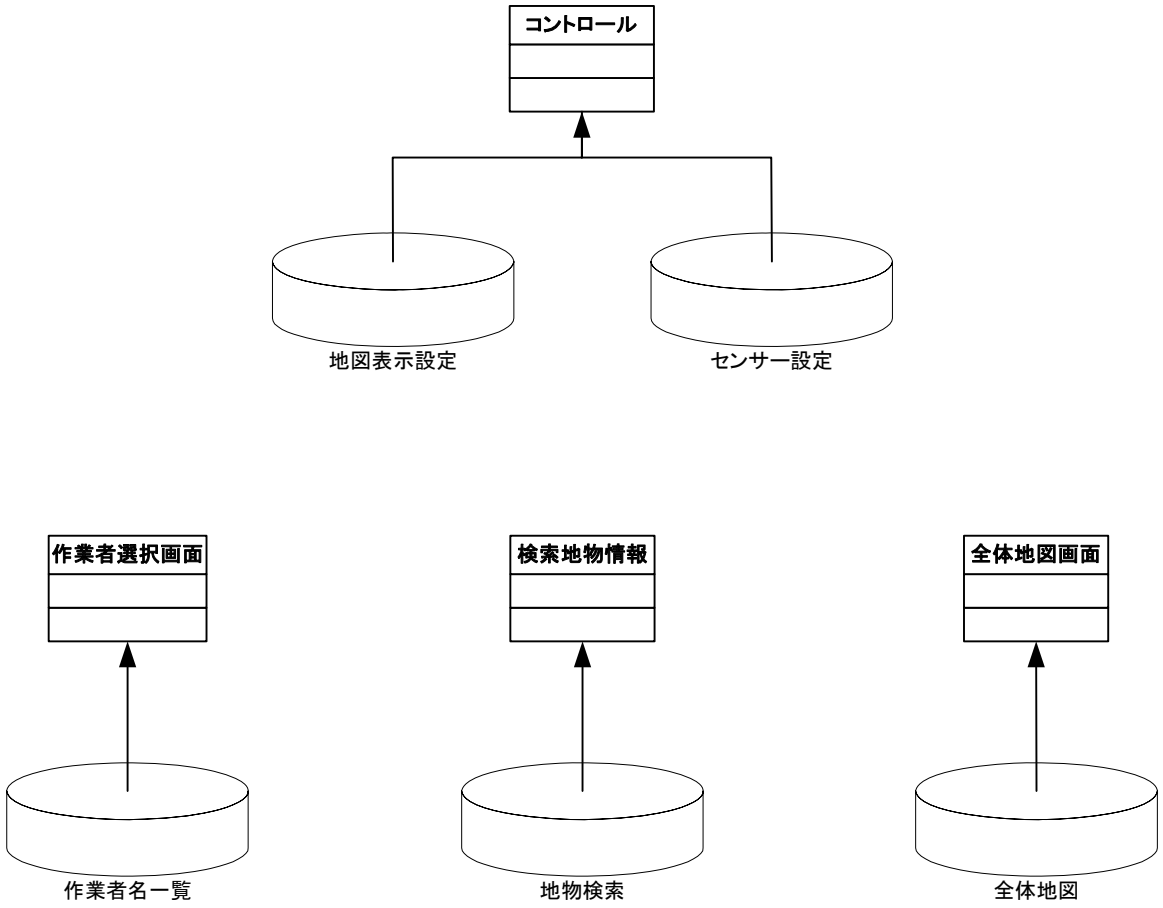


図 2-49　クラスファイル対応図（その２）

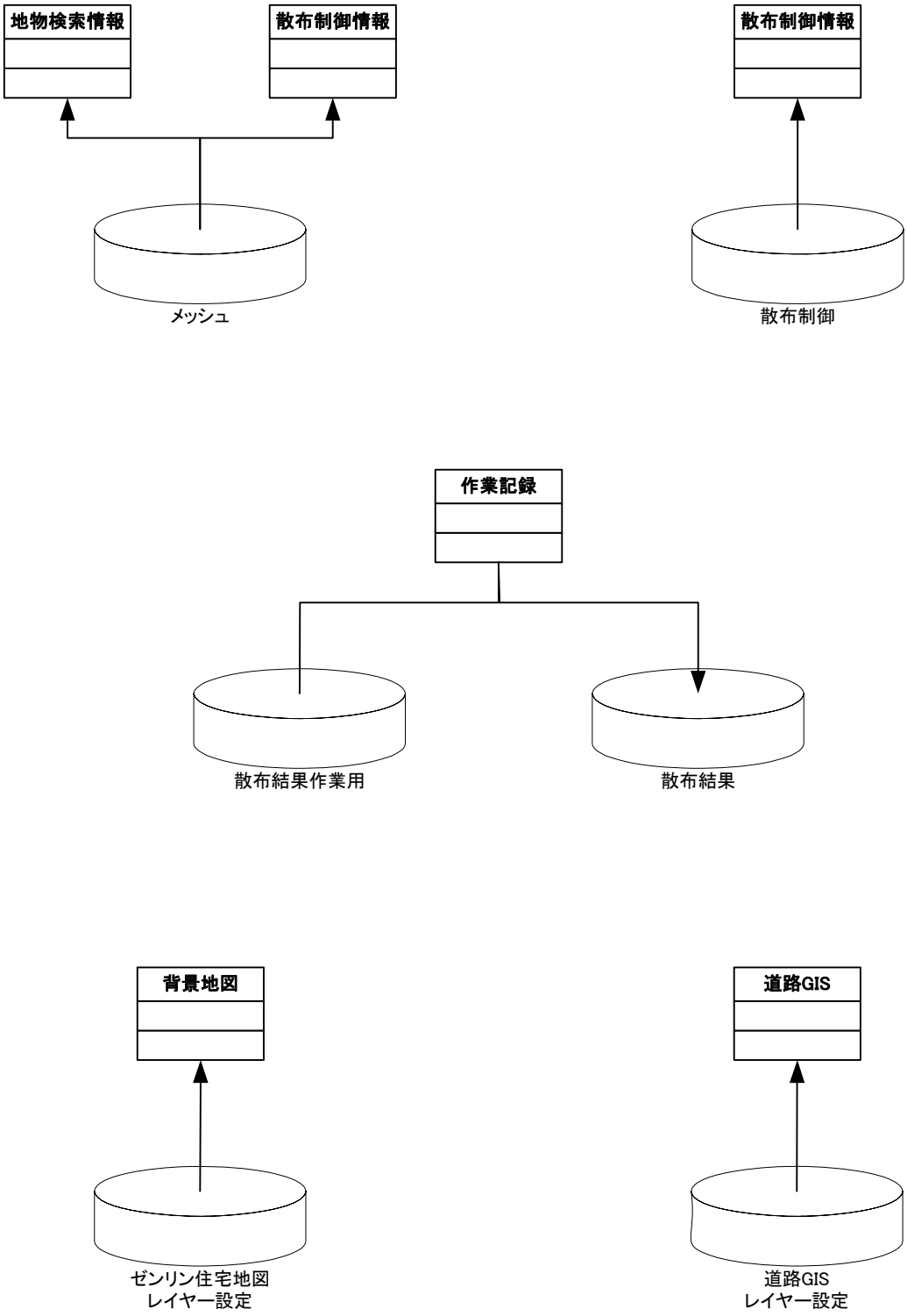


図 2-50　クラスファイル対応図（その３）

(4) 制御アルゴリズム

a） 散布制御

散布制御は、散布装置に対して散布量、散布幅、散布方向等の散布制御情報を送信することによって制御を行う。散布装置の制御仕様およびデータフォーマットを以下に示す。

1. 制御信号（ＲＳ２３２Ｃ）の送受信について

●１－１　切り替えスイッチを「ＰＣ側入り」に選択されている時

１－１－１　ＰＣ側より送信されて来るデータを、散布制御装置が規定したバイト数を受信し終った時点から受信したデータを検証し約0．１秒後にＰＣ側へ送信する。但し、通信エラーまたはＢＣＣエラーが発生した場合は、エラー情報（１bit ）を付加して送信する。

又、エラーが発生した場合は、散布制御装置内の散布諸元は更新されない。

１－１－２　ＰＣ側より送信されて来るデータの更新には、１秒以上の間隔を開ける。

●１－２　切り替えスイッチを「ＰＣ側切り」に選択されている時

１－２－１　散布制御装置の散布諸元が、変更された時のみＰＣ側へ、そのデータを送信する。

１－２－２　散布制御装置から送信しますデータの更新は、１秒の間隔を空ける。

●１－３　切り替えスイッチを「ＰＣ側切り」から「ＰＣ側入り」に選択した時

１－３－１　散布制御装置の現在設定されている散布諸元を、１秒以内にＰＣ側へ送信する。

１－３－２　現在設定されている散布諸元は、ＰＣ側からのデータ更新がない限り変わらない。

●１－４　切り替えスイッチを「ＰＣ側入り」から「ＰＣ側切り」に選択した時

１－４－１　散布制御装置の現在設定されている散布諸元を、１秒以内にＰＣ側へ送信する。

１－４－２　散布幅補正、散布薬剤、散布方向、散布/停止の各諸元は、散布制御装置の操作パネル面のスイッチ及びモノレバーの設定位置により決定される。

●１－５　散布制御装置からＰＣ側への送信時（１－２，１－３，１－４）にはデータ構造の「散布０Ｎ－０ＦＦ」の２bit=１にする。

●１－６　切り替えスイッチについて

切り替えスイッチは、サブコントローラの操作パネル面に配置し,トグルスイッチを上側に倒すと「ＰＣ側入り」に、下側に倒すと「ＰＣ側切り」となる。

２．ＰＣ側から散布制御装置への送信モード（散布０Ｎ－０ＦＦ）について

●２－１　散布停止（ゲート閉、散布円盤停止、コンベアー停止)。

データ構造の「散布０Ｎ－０ＦＦ」を　１bit=０　０bit=０　とする。

●２－２　散布（ゲート開、散布円盤作動、コンベアー作動)。

データ構造の「散布０Ｎ－０ＦＦ」を　１bit=１　０bit=０　とする。

●２－３　散布待機（ゲート開、散布円盤作動、コンベアー停止)。

データ構造の「散布０Ｎ－０ＦＦ」を　1 bit=０　０bit=１　とする。

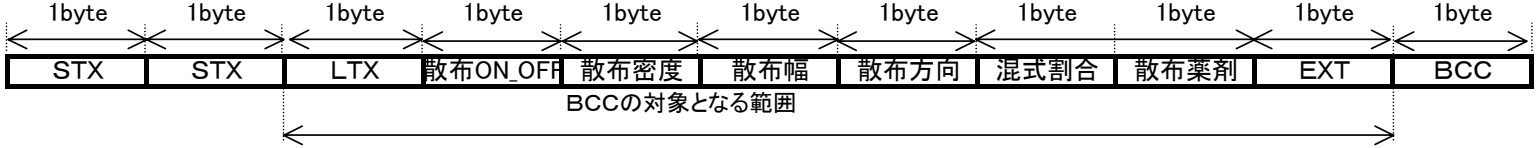
４．制御データ注意事項

・散布停止、散布待機状態に変更する場合に、散布量、散布幅等のデータを指定しない場合通信エラーとなる。

・通信エラー時の再送は、1 秒以上の間隔を空けて送信する。

表 2-55　散布制御装置データフォーマット

1 ﾃﾞｰﾀ構造

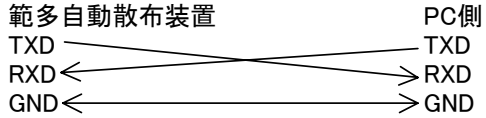


名称	内容（各1byte）
STX	先頭ﾃｷｽﾄﾃﾞｰﾀ　（0xE2）
STX	先頭ﾃｷｽﾄﾃﾞｰﾀ　（0xE2）
LTX	通信するﾃﾞｰﾀの長さ　（0x06）
散布ON_OFF	別紙参照（データ詳細）
散布密度	（例）散布密度50g/m^2（0x32）
散布幅	（例）散布幅3.0m（0x1E）、散布幅7.0m（0x46）
散布方向	右方向（0×01）、左方向（0×00）
混式割合	今回は常に(0x00)
散布薬剤	フレーク（0x01）、粒（0x02）、粒大（0×03）、原塩（0×04）
EXT	終端ﾃｷｽﾄﾃﾞｰﾀ　（0xE3）
BCC	LTXからETXまでのXOR結果

2 伝送諸元及びキャラクター構成

①同期方式　調歩同期
②通信速度　9600bps
③ｷｬﾗｸﾀ長　8ﾋﾞｯﾄ
④ｽﾄｯﾌﾟﾋﾞｯﾄ　1ﾋﾞｯﾄ
⑤ﾊﾟﾘﾃｨﾋﾞｯﾄ　偶数

3 接続



4 自動散布装置側のコネクター仕様

・DSUB（9ピン）メス。
・RXD（2番ピン）、TXD（3番ピン）、GND（5番ピン）、その他のピンは全てNC。

表 2-56　散布制御装置データフォーマット（その２）

5-1 「散布ON－OFF」

D7	D6	D5	D4	D3	D2	D1	D0

Bit 1	Bit 0	動作
0	0	散布停止
0	1	散布待機
1	0	散布

Bit 2	PC側への送信
0	「PC側入り」
1	「PC側切り」

Bit 7	Bit 6	Bit 5	Bit 4	エラー情報
×	×	×	0	通信、BCCエラーなし
×	×	×	1	通信、BCCエラーあり

b） 散布制御アルゴリズム

散布制御については、散布メッシュを用いて散布制御を行う。散布制御に際し散布メッシュを内部的に以下の配列を作成して、配列の情報を参照あるいは更新を行って制御を行う。なお、配列は車線メッシュを以下の手順に従って変換を行い格納する。

制御配列は二次元配列とし、1 列目を車線、2 列目を車線延長方向のメッシュ番号（10m メッシュ）とする。車線は 0 を追い越し車線、1 を走行車線、2 以上順番に副道１、副道２・・・というように格納する。例えば、追い越し車線、走行車線、副道の３つの車線が存在する箇所においては、配列は[0][n]～[2][n]の配列となる（n は車線延長方向のメッシュ数）。

なお、車線延長方向のメッシュ番号は、散布変更箇所検索の効率化のため、上り車線の後ろに下り車線を続けて格納し、上り車線と下り車線の境部および下り車線の最後には、メッシュの存在有無を表すフラグ（次に説明する mN[][]）の車線方向の値を全て-1 とする。

- 制御配列

mN[][] ――――――＞　メッシュの存在有無を表すフラグ

値が 0 の場合、メッシュが存在しない

値が 1 の場合、メッシュが存在する。

たとえば、ある地点において走行車線しか存在しない場合は、mN[0][p]=0、mN[1][p]=1、mN[2][p]=0（p はある地点におけるメッシュ配列番号）となる。

この配列は、散布制御メッシュデータの初期読み込み時に生成され、以後変化しない。

mD[][] ――――――＞　散布制御メッシュと制御配列を対応させるための配列

散布制御メッシュの配列番号が格納される。散布制御メッシュが存在しない場合は値は-1 とする。

この配列は、散布制御メッシュデータの初期読み込み時に生成され、以後変化しない。

mM[][] ――――――＞　散布制御情報と制御配列を対応させるための配列

散布制御情報（車線に対する散布量や散布幅、散布方向等）データの配列番号が格納される。散布メッシュが存在しない場合は値は-1 とする。

この配列は、散布制御メッシュデータの初期読み込み時に生成され、以後変化しない。

mF[][] ――――――＞　散布の指示開始、終了、変更を表すフラグを格納するための配列

散布を開始する地点及び変更する地点（散布制御装置に散布を指示する地点）において、値は 1 とする。

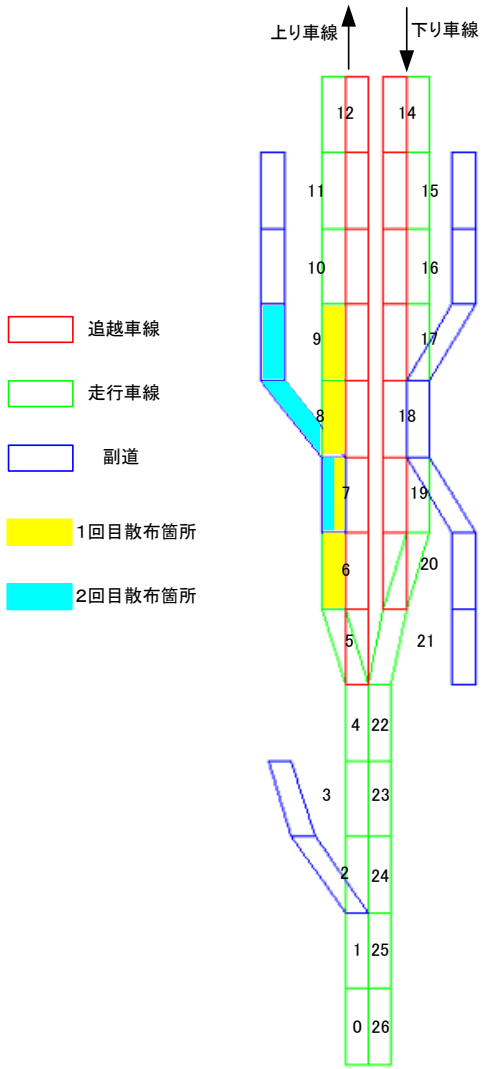
散布を停止する地点（散布制御装置に散布停止を指示する地点）において、値は-1 とする。散布を変更しない地点において、値は 0 とする。

この配列は、散布制御メッシュデータの初期読み込み時に生成され、以後変化しない。

mE[][] ――――――＞　散布を制御するための配列

初期状態では、散布順番号が格納される。散布を行わない箇所においては値は-1、散布済み箇所は 0、1 回目走行時に散布する箇所は 1、2 回目走行時に散布する箇所は 2 が設定される。走行するごとに値がカウントダウンされ、値が 1 の時のみ散布制御装置へ指示が出される。したがって、既に散布済みの箇所や 2 回目走行時のみ散布する箇所は、値が 1 でないため散布されない。

以下に、テストデータにおける配列数値の変化を示す。



mN□□の値

メッシュ番号	追越車線	走行車線	副道	
27	0	0	0	メッシュ終点
26	0	1	0	下り1
25	0	1	0	下り2
24	0	1	0	下り3
23	0	1	0	下り4
22	0	1	0	下り5
21	0	1	1	下り6
20	1	1	1	下り7
19	1	1	1	下り8
18	1	1	1	下り9
17	1	1	1	下り10
16	1	1	1	下り11
15	1	1	1	下り12
14	1	1	0	下り13
13	0	0	0	折り返し地点
12	1	1	0	上り13
11	1	1	1	上り12
10	1	1	1	上り11
9	1	1	1	上り10
8	1	1	1	上り9
7	1	1	1	上り8
6	1	1	0	上り7
5	1	1	0	上り6
4	0	1	0	上り5
3	0	1	1	上り4
2	0	1	1	上り3
1	0	1	0	上り2
0	0	1	0	上り1

図 2-51　散布制御アルゴリズム（その１）

mD□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	28	−1	下り1
25	−1	29	−1	下り2
24	−1	30	−1	下り3
23	−1	31	−1	下り4
22	−1	32	−1	下り5
21	−1	33	48	下り6
20	34	41	49	下り7
19	35	42	50	下り8
18	36	43	51	下り9
17	37	44	52	下り10
16	38	45	53	下り11
15	39	46	54	下り12
14	40	47	−1	下り13
13	−1	−1	−1	折り返し地点
12	12	20	−1	上り13
11	11	19	27	上り12
10	10	18	26	上り11
9	9	17	25	上り10
8	8	16	24	上り9
7	7	15	23	上り8
6	6	14	−1	上り7
5	5	13	−1	上り6
4	−1	4	−1	上り5
3	−1	3	22	上り4
2	−1	2	21	上り3
1	−1	1	−1	上り2
0	−1	−1	−1	上り1

mM□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	−1	−1	下り1
25	−1	−1	−1	下り2
24	−1	−1	−1	下り3
23	−1	−1	−1	下り4
22	−1	−1	−1	下り5
21	−1	−1	−1	下り6
20	−1	−1	−1	下り7
19	−1	−1	−1	下り8
18	−1	−1	−1	下り9
17	−1	−1	−1	下り10
16	−1	−1	−1	下り11
15	−1	−1	−1	下り12
14	−1	−1	−1	下り13
13	−1	−1	−1	折り返し地点
12	−1	−1	−1	上り13
11	−1	−1	−1	上り12
10	−1	−1	−1	上り11
9	−1	1	2	上り10
8	−1	1	2	上り9
7	−1	0	2	上り8
6	−1	0	−1	上り7
5	−1	−1	−1	上り6
4	−1	−1	−1	上り5
3	−1	−1	−1	上り4
2	−1	−1	−1	上り3
1	−1	−1	−1	上り2
0	−1	−1	−1	上り1

図 2-52　散布制御アルゴリズム（その２）

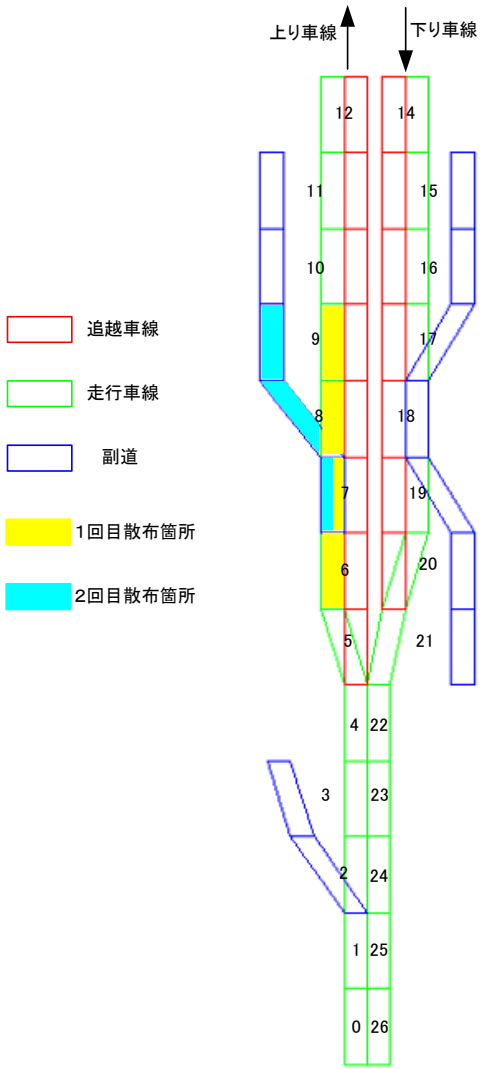
mF□□の値

メッシュ番号	追越車線	走行車線	副道	
27	0	0	0	メッシュ終点
26	0	0	0	下り1
25	0	0	0	下り2
24	0	0	0	下り3
23	0	0	0	下り4
22	0	0	0	下り5
21	0	0	0	下り6
20	0	0	0	下り7
19	0	0	0	下り8
18	0	0	0	下り9
17	0	0	0	下り10
16	0	0	0	下り11
15	0	0	0	下り12
14	0	0	0	下り13
13	0	0	0	折り返し地点
12	0	0	0	上り13
11	0	0	0	上り12
10	0	−1	−1	上り11
9	0	0	0	上り10
8	0	1	0	上り9
7	0	0	2	上り8
6	0	1	0	上り7
5	0	0	0	上り6
4	0	0	0	上り5
3	0	0	0	上り4
2	0	0	0	上り3
1	0	0	0	上り2
0	0	0	0	上り1

mE□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	−1	−1	下り1
25	−1	−1	−1	下り2
24	−1	−1	−1	下り3
23	−1	−1	−1	下り4
22	−1	−1	−1	下り5
21	−1	−1	−1	下り6
20	−1	−1	−1	下り7
19	−1	−1	−1	下り8
18	−1	−1	−1	下り9
17	−1	−1	−1	下り10
16	−1	−1	−1	下り11
15	−1	−1	−1	下り12
14	−1	−1	−1	下り13
13	−1	−1	−1	折り返し地点
12	−1	−1	−1	上り13
11	−1	−1	−1	上り12
10	−1	−1	−1	上り11
9	−1	1	2	上り10
8	−1	1	2	上り9
7	−1	1	2	上り8
6	−1	1	0	上り7
5	−1	−1	−1	上り6
4	−1	−1	−1	上り5
3	−1	−1	−1	上り4
2	−1	−1	−1	上り3
1	−1	−1	−1	上り2
0	−1	−1	−1	上り1

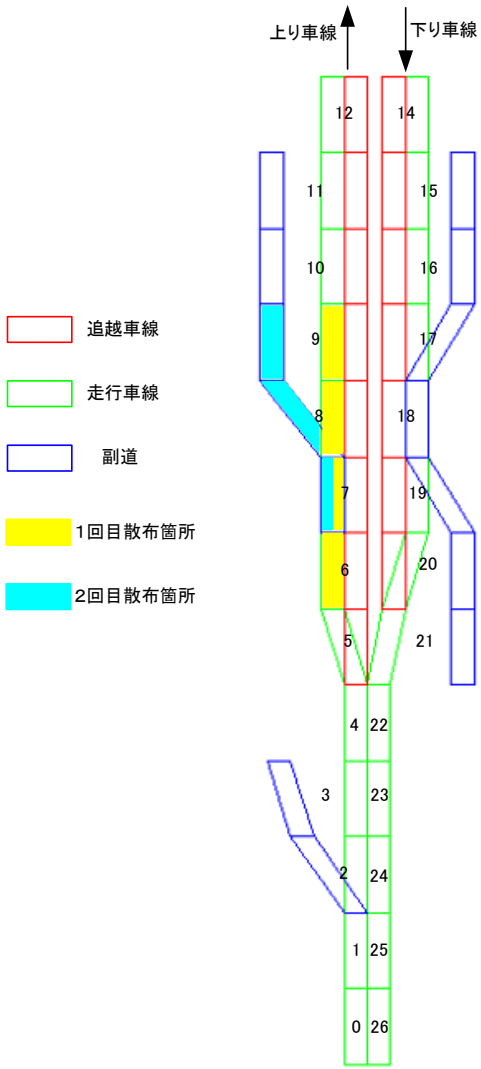
図 2-53　散布制御アルゴリズム（その３）



mE□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	−1	−1	下り1
25	−1	−1	−1	下り2
24	−1	−1	−1	下り3
23	−1	−1	−1	下り4
22	−1	−1	−1	下り5
21	−1	−1	−1	下り6
20	−1	−1	−1	下り7
19	−1	−1	−1	下り8
18	−1	−1	−1	下り9
17	−1	−1	−1	下り10
16	−1	−1	−1	下り11
15	−1	−1	−1	下り12
14	−1	−1	−1	下り13
13	−1	−1	−1	折り返し地点
12	−1	−1	−1	上り13
11	−1	−1	−1	上り12
10	−1	−1	−1	上り11
9	−1	1	2	上り10
8	−1	1	2	上り9
7	−1	1	2	上り8
6	−1	1	0	上り7
5	−1	−1	−1	上り6
4	−1	−1	−1	上り5
3	−1	−1	−1	上り4
2	−1	−1	−1	上り3
1	−1	−1	−1	上り2
0	−1	−1	−1	上り1

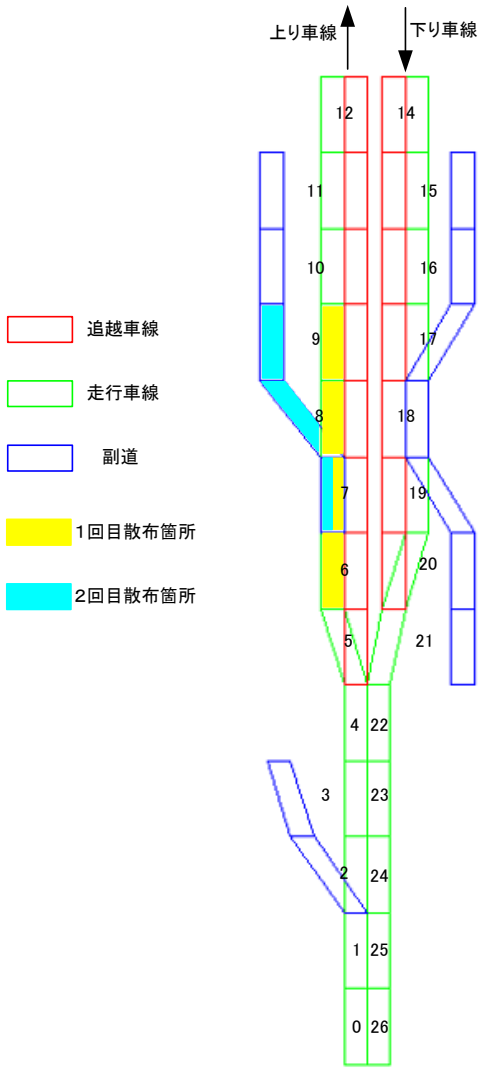
図 2-54　散布制御アルゴリズム（その４）



mE□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	−1	−1	下り1
25	−1	−1	−1	下り2
24	−1	−1	−1	下り3
23	−1	−1	−1	下り4
22	−1	−1	−1	下り5
21	−1	−1	−1	下り6
20	−1	−1	−1	下り7
19	−1	−1	−1	下り8
18	−1	−1	−1	下り9
17	−1	−1	−1	下り10
16	−1	−1	−1	下り11
15	−1	−1	−1	下り12
14	−1	−1	−1	下り13
13	−1	−1	−1	折り返し地点
12	−1	−1	−1	上り13
11	−1	−1	−1	上り12
10	−1	−1	−1	上り11
9	−1	1	2	上り10
8	−1	1	2	上り9
7	−1	1	2	上り8
6	−1	0	0	上り7
5	−1	−1	−1	上り6
4	−1	−1	−1	上り5
3	−1	−1	−1	上り4
2	−1	−1	−1	上り3
1	−1	−1	−1	上り2
0	−1	−1	−1	上り1

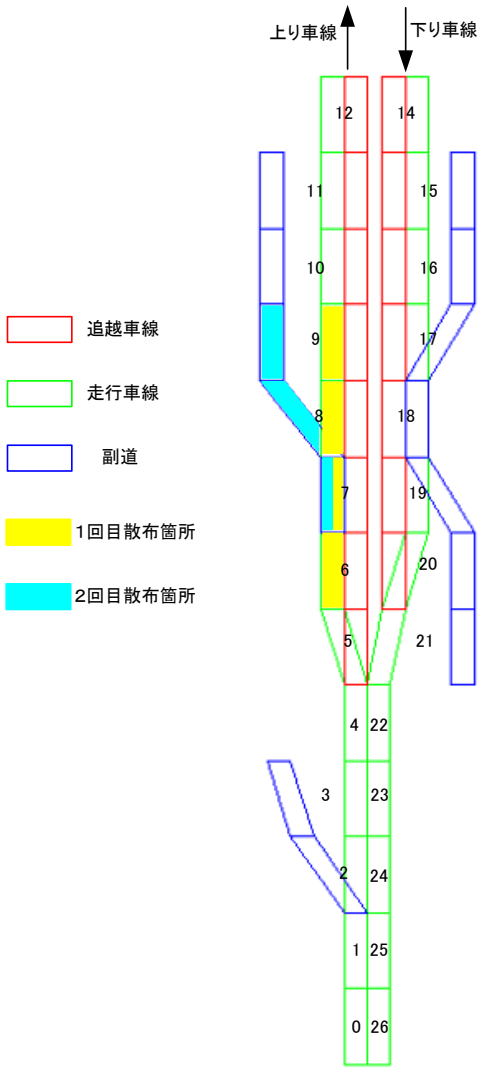
図 2-55　散布制御アルゴリズム（その５）



mE□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	−1	−1	下り1
25	−1	−1	−1	下り2
24	−1	−1	−1	下り3
23	−1	−1	−1	下り4
22	−1	−1	−1	下り5
21	−1	−1	−1	下り6
20	−1	−1	−1	下り7
19	−1	−1	−1	下り8
18	−1	−1	−1	下り9
17	−1	−1	−1	下り10
16	−1	−1	−1	下り11
15	−1	−1	−1	下り12
14	−1	−1	−1	下り13
13	−1	−1	−1	折り返し地点
12	−1	−1	−1	上り13
11	−1	−1	−1	上り12
10	−1	−1	−1	上り11
9	−1	1	2	上り10
8	−1	1	2	上り9
7	−1	0	1	上り8
6	−1	0	0	上り7
5	−1	−1	−1	上り6
4	−1	−1	−1	上り5
3	−1	−1	−1	上り4
2	−1	−1	−1	上り3
1	−1	−1	−1	上り2
0	−1	−1	−1	上り1

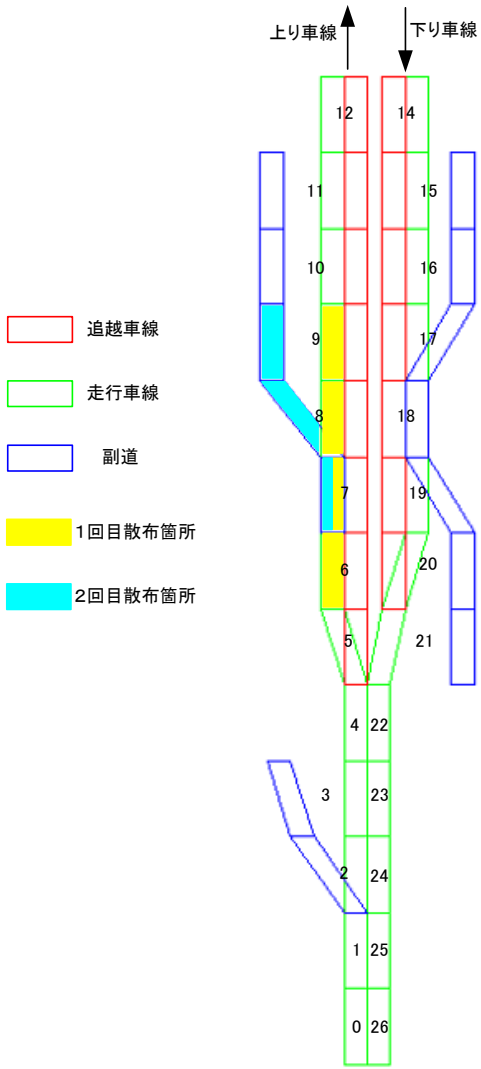
図 2-56　散布制御アルゴリズム（その６）



mE□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	−1	−1	下り1
25	−1	−1	−1	下り2
24	−1	−1	−1	下り3
23	−1	−1	−1	下り4
22	−1	−1	−1	下り5
21	−1	−1	−1	下り6
20	−1	−1	−1	下り7
19	−1	−1	−1	下り8
18	−1	−1	−1	下り9
17	−1	−1	−1	下り10
16	−1	−1	−1	下り11
15	−1	−1	−1	下り12
14	−1	−1	−1	下り13
13	−1	−1	−1	折り返し地点
12	−1	−1	−1	上り13
11	−1	−1	−1	上り12
10	−1	−1	−1	上り11
9	−1	0	1	上り10
8	−1	0	1	上り9
7	−1	0	1	上り8
6	−1	0	0	上り7
5	−1	−1	−1	上り6
4	−1	−1	−1	上り5
3	−1	−1	−1	上り4
2	−1	−1	−1	上り3
1	−1	−1	−1	上り2
0	−1	−1	−1	上り1

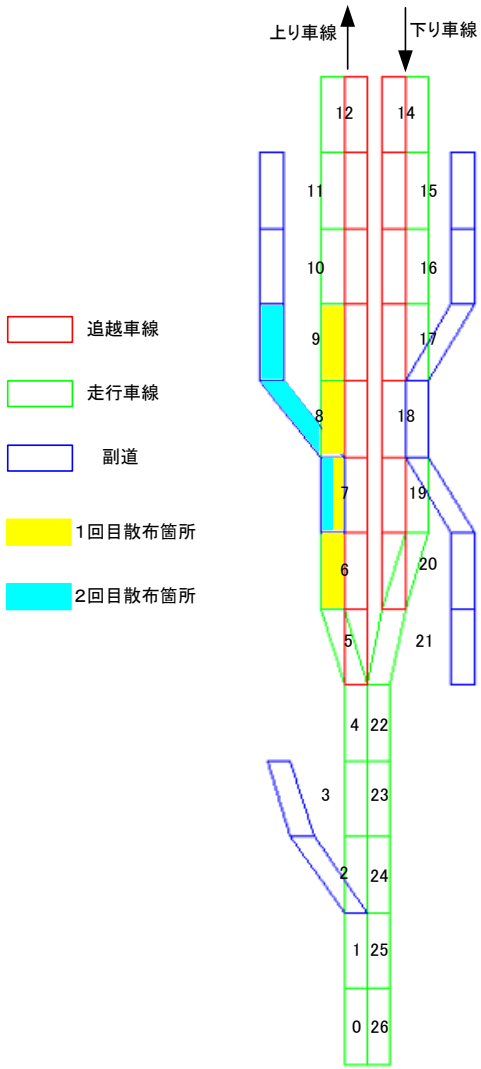
図 2-57　散布制御アルゴリズム（その７）



mE□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	−1	−1	下り1
25	−1	−1	−1	下り2
24	−1	−1	−1	下り3
23	−1	−1	−1	下り4
22	−1	−1	−1	下り5
21	−1	−1	−1	下り6
20	−1	−1	−1	下り7
19	−1	−1	−1	下り8
18	−1	−1	−1	下り9
17	−1	−1	−1	下り10
16	−1	−1	−1	下り11
15	−1	−1	−1	下り12
14	−1	−1	−1	下り13
13	−1	−1	−1	折り返し地点
12	−1	−1	−1	上り13
11	−1	−1	−1	上り12
10	−1	−1	−1	上り11
9	−1	0	1	上り10
8	−1	0	1	上り9
7	−1	0	1	上り8
6	−1	0	0	上り7
5	−1	−1	−1	上り6
4	−1	−1	−1	上り5
3	−1	−1	−1	上り4
2	−1	−1	−1	上り3
1	−1	−1	−1	上り2
0	−1	−1	−1	上り1

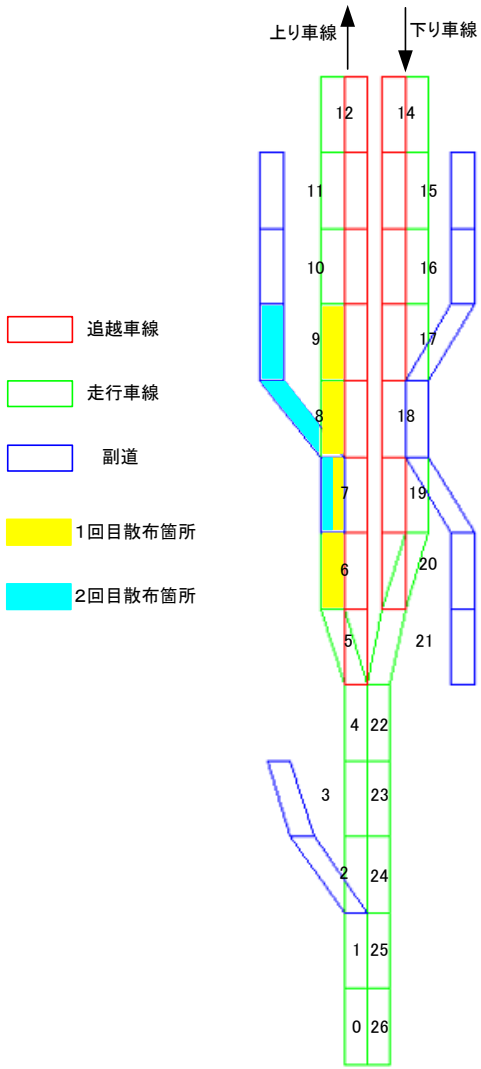
図 2-58　散布制御アルゴリズム（その８）



mE□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	−1	−1	下り1
25	−1	−1	−1	下り2
24	−1	−1	−1	下り3
23	−1	−1	−1	下り4
22	−1	−1	−1	下り5
21	−1	−1	−1	下り6
20	−1	−1	−1	下り7
19	−1	−1	−1	下り8
18	−1	−1	−1	下り9
17	−1	−1	−1	下り10
16	−1	−1	−1	下り11
15	−1	−1	−1	下り12
14	−1	−1	−1	下り13
13	−1	−1	−1	折り返し地点
12	−1	−1	−1	上り13
11	−1	−1	−1	上り12
10	−1	−1	−1	上り11
9	−1	0	1	上り10
8	−1	0	1	上り9
7	−1	0	0	上り8
6	−1	0	0	上り7
5	−1	−1	−1	上り6
4	−1	−1	−1	上り5
3	−1	−1	−1	上り4
2	−1	−1	−1	上り3
1	−1	−1	−1	上り2
0	−1	−1	−1	上り1

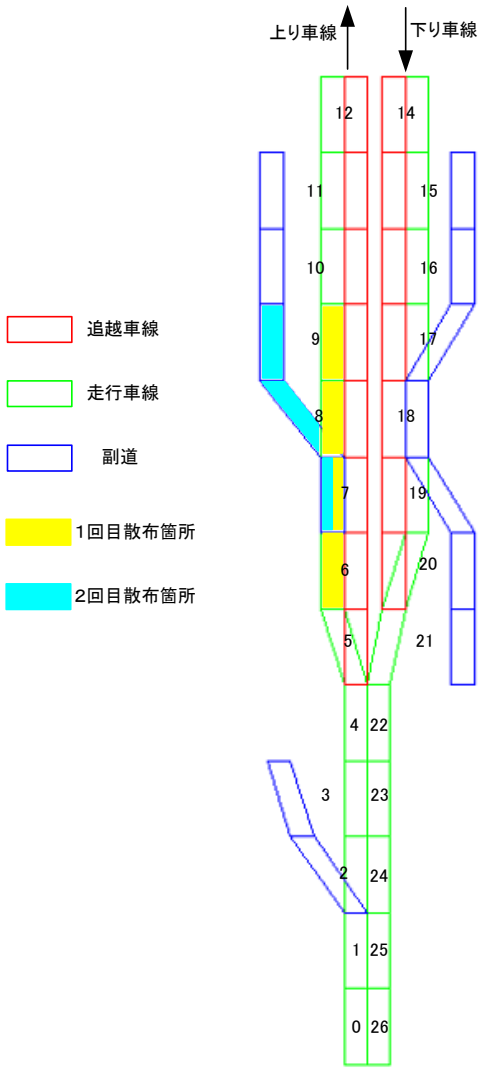
図 2-59　散布制御アルゴリズム（その９）



mE□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	−1	−1	下り1
25	−1	−1	−1	下り2
24	−1	−1	−1	下り3
23	−1	−1	−1	下り4
22	−1	−1	−1	下り5
21	−1	−1	−1	下り6
20	−1	−1	−1	下り7
19	−1	−1	−1	下り8
18	−1	−1	−1	下り9
17	−1	−1	−1	下り10
16	−1	−1	−1	下り11
15	−1	−1	−1	下り12
14	−1	−1	−1	下り13
13	−1	−1	−1	折り返し地点
12	−1	−1	−1	上り13
11	−1	−1	−1	上り12
10	−1	−1	−1	上り11
9	−1	0	1	上り10
8	−1	0	0	上り9
7	−1	0	0	上り8
6	−1	0	0	上り7
5	−1	−1	−1	上り6
4	−1	−1	−1	上り5
3	−1	−1	−1	上り4
2	−1	−1	−1	上り3
1	−1	−1	−1	上り2
0	−1	−1	−1	上り1

図 2-60　散布制御アルゴリズム（その１０）



mE□□の値

メッシュ番号	追越車線	走行車線	副道	
27	−1	−1	−1	メッシュ終点
26	−1	−1	−1	下り1
25	−1	−1	−1	下り2
24	−1	−1	−1	下り3
23	−1	−1	−1	下り4
22	−1	−1	−1	下り5
21	−1	−1	−1	下り6
20	−1	−1	−1	下り7
19	−1	−1	−1	下り8
18	−1	−1	−1	下り9
17	−1	−1	−1	下り10
16	−1	−1	−1	下り11
15	−1	−1	−1	下り12
14	−1	−1	−1	下り13
13	−1	−1	−1	折り返し地点
12	−1	−1	−1	上り13
11	−1	−1	−1	上り12
10	−1	−1	−1	上り11
9	−1	0	0	上り10
8	−1	0	0	上り9
7	−1	0	0	上り8
6	−1	0	0	上り7
5	−1	−1	−1	上り6
4	−1	−1	−1	上り5
3	−1	−1	−1	上り4
2	−1	−1	−1	上り3
1	−1	−1	−1	上り2
0	−1	−1	−1	上り1

図 2-61　散布制御アルゴリズム（その１１）

2.3. 操作支援システムの実装設計

(1) 適用範囲

本節は、操作支援システムにおける実装設計についてのみ記載するものとする。

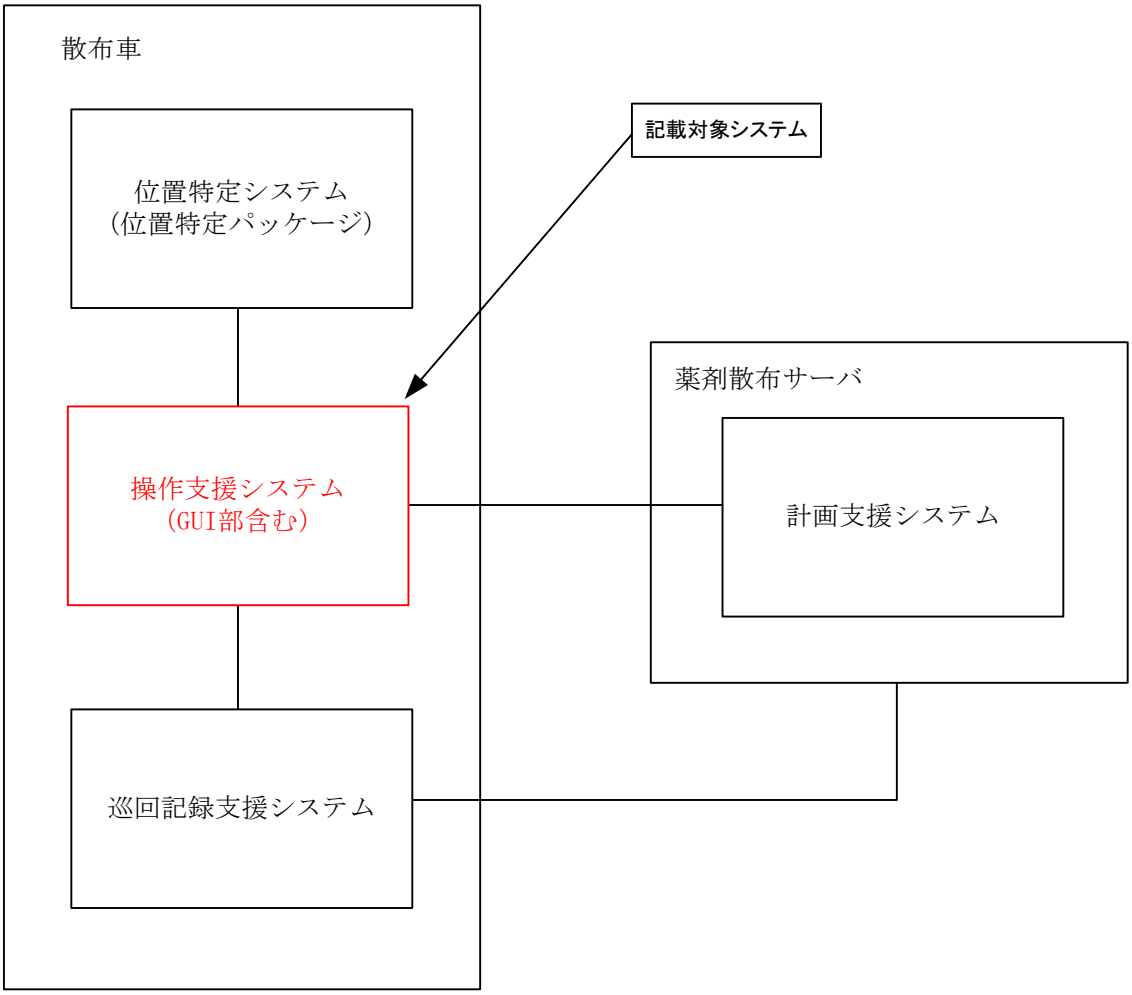


図 2-62　記載範囲

(2) ソフトウェア構成

ソフトウェアの基本構成を図 2-63 に示す。

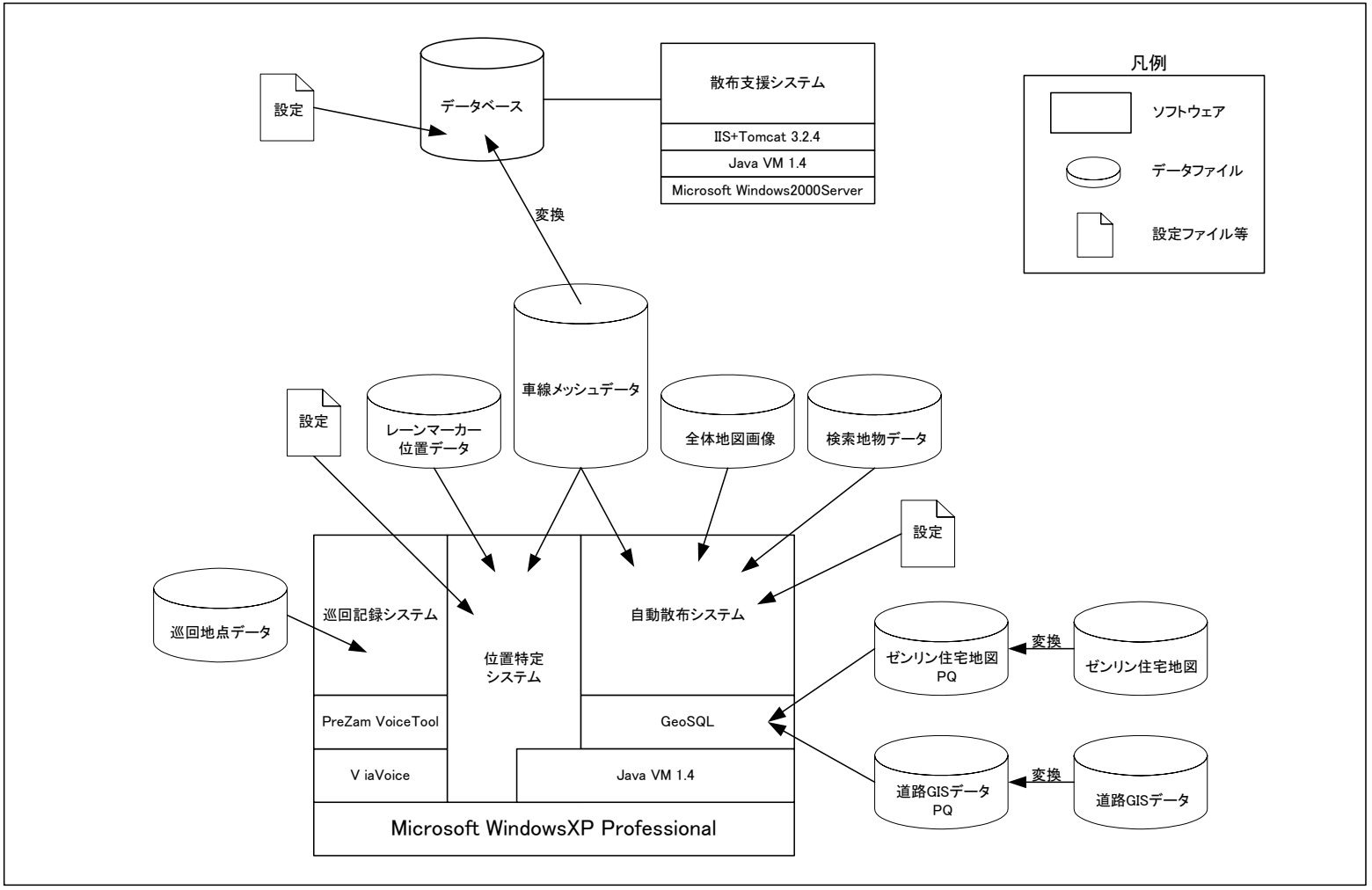


図 2-63　ソフトウェア構成図

(3) モジュール設計

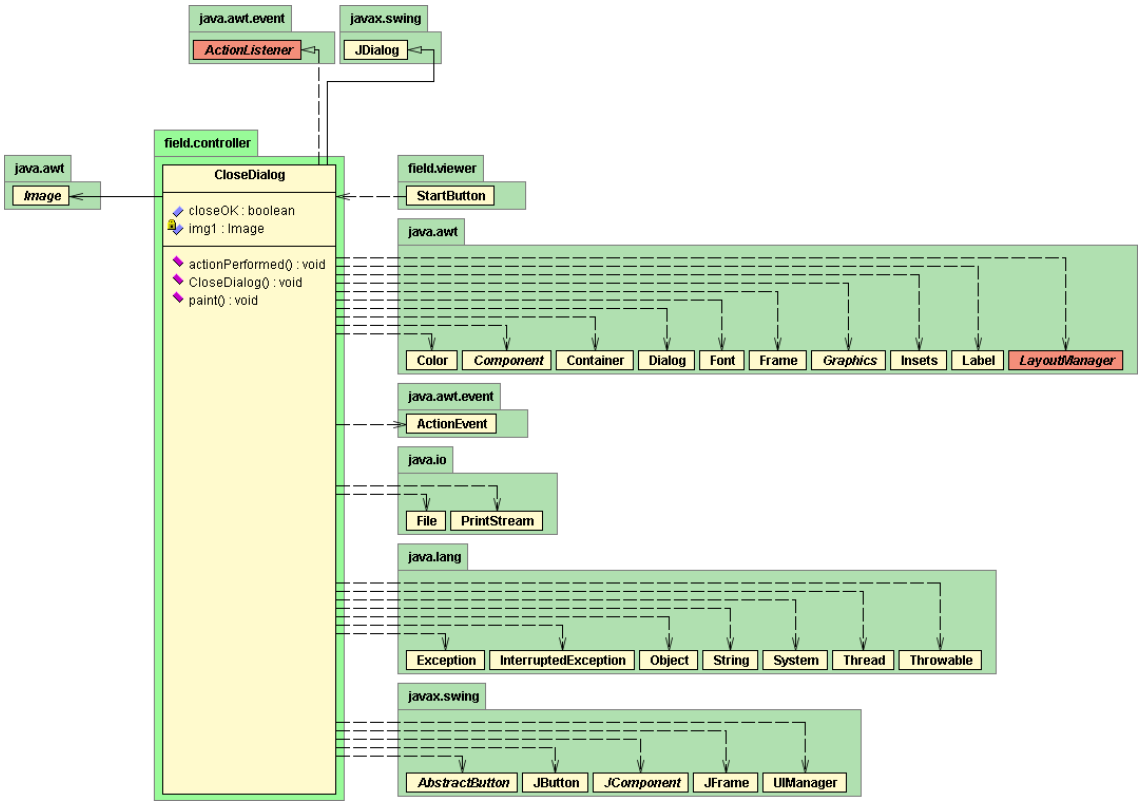
a） パッケージ一覧

パッケージ	
field.controller	全体のコントロールを行う
field.map	背景地図の表示を行う
field.navigation	地物検索処理を行う
field.sprinkle	散布制御を行う
field.viewer	画面レイアウトを行う

b） パッケージ field.controller

クラスの概要	
CloseDialog	システム終了確認画面クラス
Coord	座標計算等を行うユーティリティクラス
ExperimentController	システム全体のコントロールを行うクラス（メインクラス）
FileDialog	地図設定ファイル選択画面クラス
LogicDialog	位置計算処理ロジック選択画面クラス
MapIntervalDialog	画面表示切換えインターバル選択画面クラス
PortList	使用できる COM ポートの一覧を取得する
PosChange	世界測地系緯度経度を日本測地系緯度経度に変換するクラス
ReadFileConfig	設定ファイルの内容を読込クラス
SelectDialog	位置取得情報設定選択確認画面クラス
SencerDialog	使用センサー選択画面クラス。
SencerIntervalDialog	センサーデータ取得インターバル選択画面クラス
SprinkleComportDialog	散布車制御用 COM ポート選択画面クラス
SprinkleDialog	散布計画ファイル選択画面クラス
WaitDialog	センサー初期化待ち画面クラス
WorkmanDialog	作業者名を選択する画面クラス

■ 1　クラス CloseDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.controller.CloseDialog
```

---

public class **CloseDialog**

extends javax.swing.JDialog

implements java.awt.event.ActionListener

システム終了確認画面クラス

フィールドの概要	
boolean	**closeOK** システム終了 OK か NG かのフラグ
private java.awt.Image	**img1** 画像アイコン

コンストラクタの概要
CloseDialog() コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント
void	**paint**(java.awt.Graphics g) ペイント処理

フィールドの詳細

closeOK

public boolean **closeOK**

システム終了 OK か NG かのフラグ

---

img1

private java.awt.Image **img1**

画像アイコン

コンストラクタの詳細

CloseDialog

public **CloseDialog**()

コンストラクタ

メソッドの詳細

paint

public void **paint**(java.awt.Graphics g)

ペイント処理

オーバーライド:

クラス java.awt.Container 内の paint

パラメータ:

g - Graphics

---

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

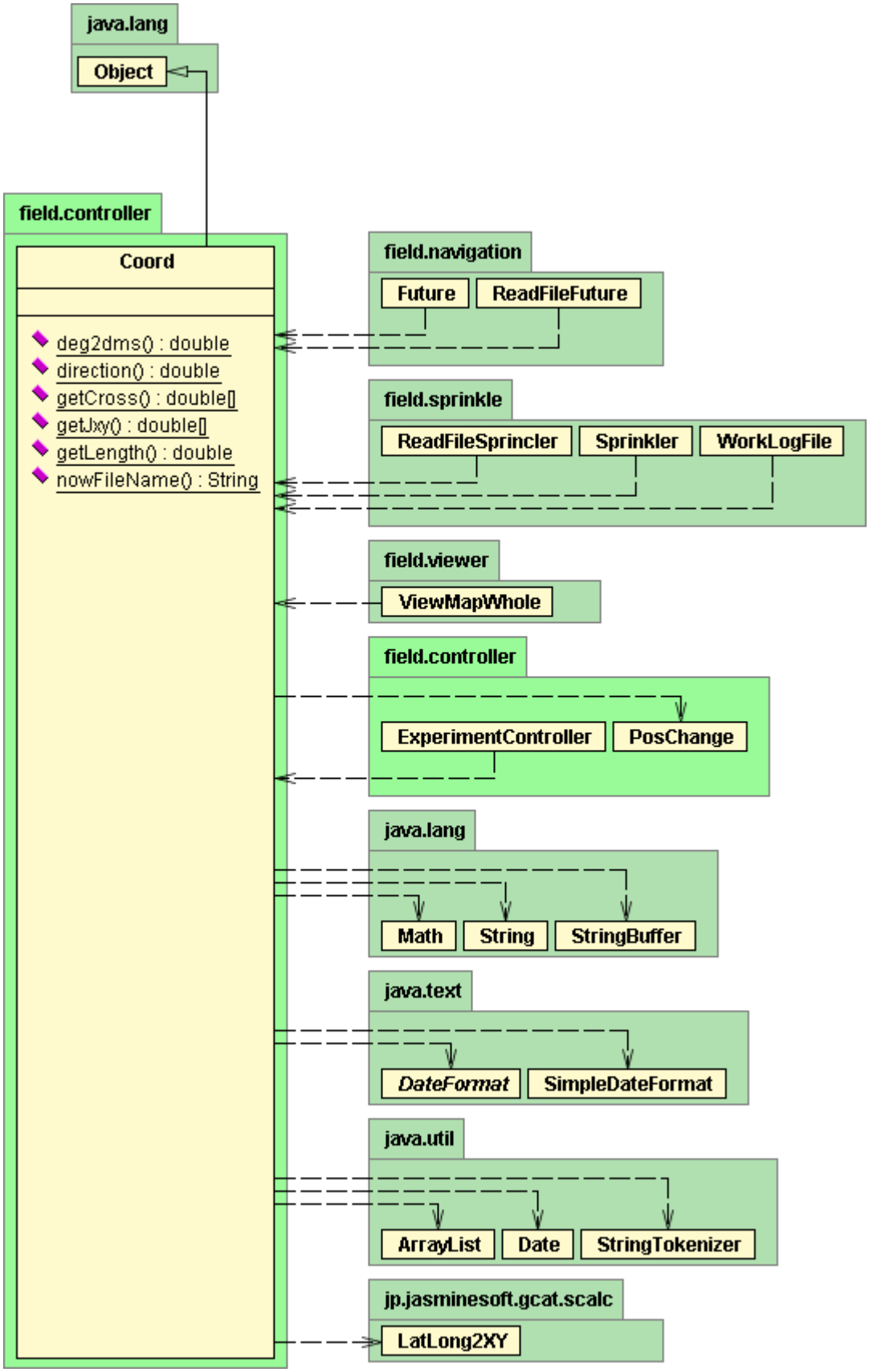
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

■ 2　クラス Coord



java.lang.Object

|

+--**field.controller.Coord**

---

public class **Coord**

extends java.lang.Object

座標計算等を行うユーティリティクラス

コンストラクタの概要
Coord()

メソッドの概要	
static double	**deg2dms**(double f) 10 進表記の緯度経度を DDD.MMSS 形式に変換する（平面直角座標変換クラスへ受け渡すため変換が必要)。
static double	**direction**(double x0, double y0, double x1, double y1) 進行方向角を計算し、degree で返す。
static double[]	**getCross**(double x0, double y0, double x1, double y1, double x2, double y2, double x3, double y3, double xx, double yy) (x0,y0)-(x1,y1)と平行で(xx,yy)を通る線と(x2,y2)-(x3,y3)との交点を求める (x0,y0)-(x1,y1)と(x2,y2)-(x3,y3)は交差する線と線
static double[]	**getJxy**(double longitude, double latitude) 世界測地系の経度・緯度を日本測地系の平面直角座標に変換する
static double	**getLength**(double x0, double y0, double x1, double y1) ２点間の距離を求める
static java.lang.String	**nowFileName**(java.lang.String fileName) 受け渡した文字列（ファイル名）に現在の日時をつけて返す

コンストラクタの詳細

Coord

public **Coord**()

メソッドの詳細

direction

public static double **direction**(double x0,
double y0,
double x1,
double y1)

進行方向角を計算し、degree で返す。

パラメータ:

x0 - 側線始点の X 座標

y0 - 側線始点の Y 座標

x1 - 側線終点の X 座標

y1 - 側線終点の Y 座標

戻り値:

進行方向角（degree）

---

getLength

public static double **getLength**(double x0,
double y0,
double x1,
double y1)

２点間の距離を求める

パラメータ:

x0 - 線の始点 X 座標列

y0 - 線の始点 Y 座標列

x1 - 線の終点 X 座標列

y1 - 線の終点 Y 座標列

戻り値:

距離

---

getCross

```
public static double[] getCross(double x0,
                                double y0,
                                double x1,
                                double y1,
                                double x2,
                                double y2,
                                double x3,
                                double y3,
                                double xx,
                                double yy)
```

(x0,y0)-(x1,y1)と平行で(xx,yy)を通る線と(x2,y2)-(x3,y3)との交点を求める (x0,y0)-(x1,y1)と(x2,y2)-(x3,y3)は交差する線と線

パラメータ:

x0 -

y0 -

x1 -

y1 -

x2 -

y2 -

x3 -

y3 -

xx -

yy -

戻り値:

交点の X,Y 座標 [0]:X [1]:Y

---

deg2dms

```
public static double deg2dms(double f)
```

10 進表記の緯度経度を DDD.MMSS 形式に変換する
（平面直角座標変換クラスへ受け渡すため変換が必要)。
パラメータ:
f - 経度または緯度
戻り値:
DDD.MMSS 形式の経度または緯度

---

getJxy

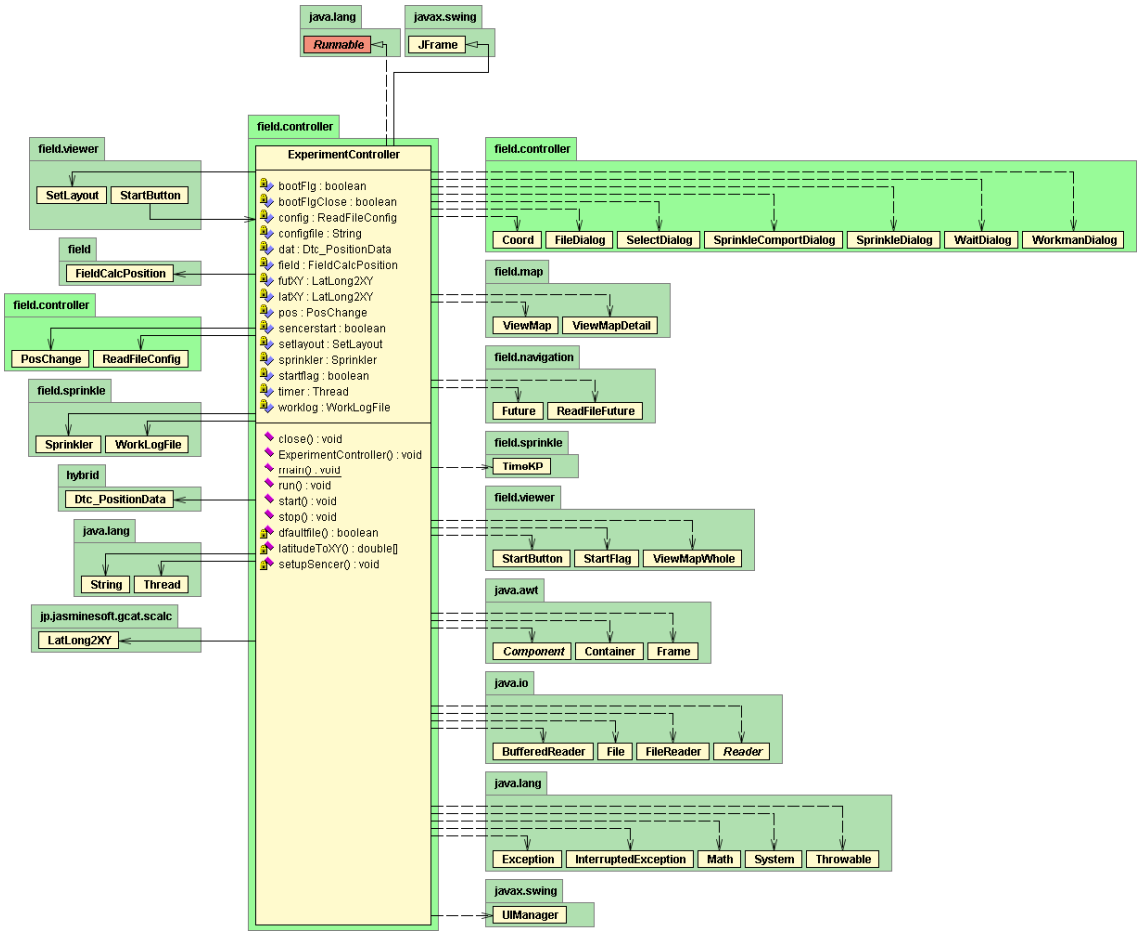
public static double[] **getJxy**(double longitude,
double latitude)
世界測地系の経度・緯度を日本測地系の平面直角座標に変換する
パラメータ:
longitude - 世界測地系の経度
lat - 世界測地系の緯度
戻り値:
日本測地系の平面直角座標 [0]:X [1]:Y

---

nowFileName

public static java.lang.String **nowFileName**(java.lang.String fileName)
受け渡した文字列（ファイル名）に現在の日時をつけて返す
パラメータ:
fileName - 文字列（ファイル名）+ 拡張子
戻り値:
文字列（ファイル名）+ _現在の日時 + 拡張子

■ ３　クラス ExperimentController




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Frame
                          |
                          +--javax.swing.JFrame
                                |
                                +--field.controller.ExperimentController
```

public class ExperimentController
extends javax.swing.JFrame
implements java.lang.Runnable

システム全体のコントロールを行うクラス（メインクラス）

フィールドの概要	
private boolean	**bootFlg** センサー初期化ダイアログの表示フラグ（表示用）
private boolean	**bootFlgClose** センサー初期化ダイアログの表示フラグ（クローズ用）
private ReadFileConfig	**config** 設定ファイル読込クラスのインスタンス
private java.lang.String	**configfile** 設定ファイル名
private Dtc_PositionData	**dat** 位置情報クラスのインスタンス
private FieldCalcPosition	**field** 位置特定処理クラスのインスタンス
private jp.jasminesoft.gcat.scalc.LatLong2XY	**futXY** 道路地物の緯度経度から平面直角座標系に変換するクラスのインスタンス
private jp.jasminesoft.gcat.scalc.LatLong2XY	**latXY** 背景地図の緯度経度から平面直角座標系に変換するクラスのインスタンス
private PosChange	**pos** 世界座標系の緯度経度を日本座標系の緯度経度に変換するクラスのインスタンス

private boolean	**sencerstart** 位置特定処理データの異常を表すフラグ
private SetLayout	**setlayout** 画面レイアウトクラスのインスタンス
private Sprinkler	**sprinkler** 散布車制御クラスのインスタンス
private boolean	**startflag** 描画開始フラグ
private java.lang.Thread	**timer** スレッドタイマークラス
private WorkLogFile	**worklog** 作業記録クラスのインスタンス

コンストラクタの概要
ExperimentController(java.lang.String[] args) コンストラクタ

メソッドの概要	
void	**close**() システム終了処理
private boolean	**dfaultfile**(java.lang.String properties) 設定ファイルの判定
private double[]	**latitudeToXY**(jp.jasminesoft.gcat.scalc.LatLong2XY lat, double x, double y, double r) 緯度、経度を平面直角座標系に変換する。
static void	**main**(java.lang.String[] args) メインプログラム、全てのコントロールを行う。
void	**run**()

	スレッドのメイン処理
private void	**setupSencer**() GPS、車速計、加速度計、ジャイロ等のセンサーの設定を行う
void	**start**() スレッドのスタート

フィールドの詳細

timer

private java.lang.Thread **timer**

スレッドタイマークラス

---

setlayout

private SetLayout **setlayout**

画面レイアウトクラスのインスタンス

---

startflag

private boolean **startflag**

描画開始フラグ

---

config

private ReadFileConfig **config**

設定ファイル読込クラスのインスタンス

---

field

private FieldCalcPosition **field**

位置特定処理クラスのインスタンス

sencerstart

private boolean **sencerstart**

位置特定処理データの異常を表すフラグ

---

dat

private Dtc_PositionData **dat**

位置情報クラスのインスタンス

---

pos

private PosChange **pos**

世界座標系の緯度経度を日本座標系の緯度経度に変換するクラスのインスタンス

---

latXY

private jp.jasminesoft.gcat.scalc.LatLong2XY **latXY**

背景地図の緯度経度から平面直角座標系に変換するクラスのインスタンス

---

futXY

private jp.jasminesoft.gcat.scalc.LatLong2XY **futXY**

道路地物の緯度経度から平面直角座標系に変換するクラスのインスタンス

---

configfile

private java.lang.String **configfile**

設定ファイル名

---

worklog

private WorkLogFile **worklog**

作業記録クラスのインスタンス

---

sprinkler

private Sprinkler **sprinkler**

散布車制御クラスのインスタンス

---

bootFlg

private boolean **bootFlg**

センサー初期化ダイアログの表示フラグ（表示用）

---

bootFlgClose

private boolean **bootFlgClose**

センサー初期化ダイアログの表示フラグ（クローズ用）

コンストラクタの詳細

ExperimentController

public **ExperimentController**(java.lang.String[] args)

コンストラクタ

メソッドの詳細

main

public static void **main**(java.lang.String[] args)

メインプログラム、全てのコントロールを行う。

---

close

public void **close**()

システム終了処理

---

run

public void **run**()
スレッドのメイン処理
定義:
インタフェース java.lang.Runnable 内の run

---

start

public void **start**()
スレッドのスタート

---

stop

public void **stop**()
スレッドの終了

---

dfaultfile

private boolean **dfaultfile**(java.lang.String properties)
設定ファイルの判定
パラメータ:
properties - センサー設定ファイル名
戻り値:
センサー設定ファイルの指定の有無 true:指定なし（ファイルが存在しない） false:指定あり

---

setupSencer

private void **setupSencer**()
GPS、車速計、加速度計、ジャイロ等のセンサーの設定を行う

---

latitudeToXY

```
private double[] latitudeToXY(jp.jasminesoft.gcat.scalc.LatLong2XY lat,
                              double x,
                              double y,
                              double r)
```

緯度、経度を平面直角座標系に変換する。また、方向角をラジアンに変換する。

パラメータ:

lat - 座標変換クラスのインスタンス

x - 経度

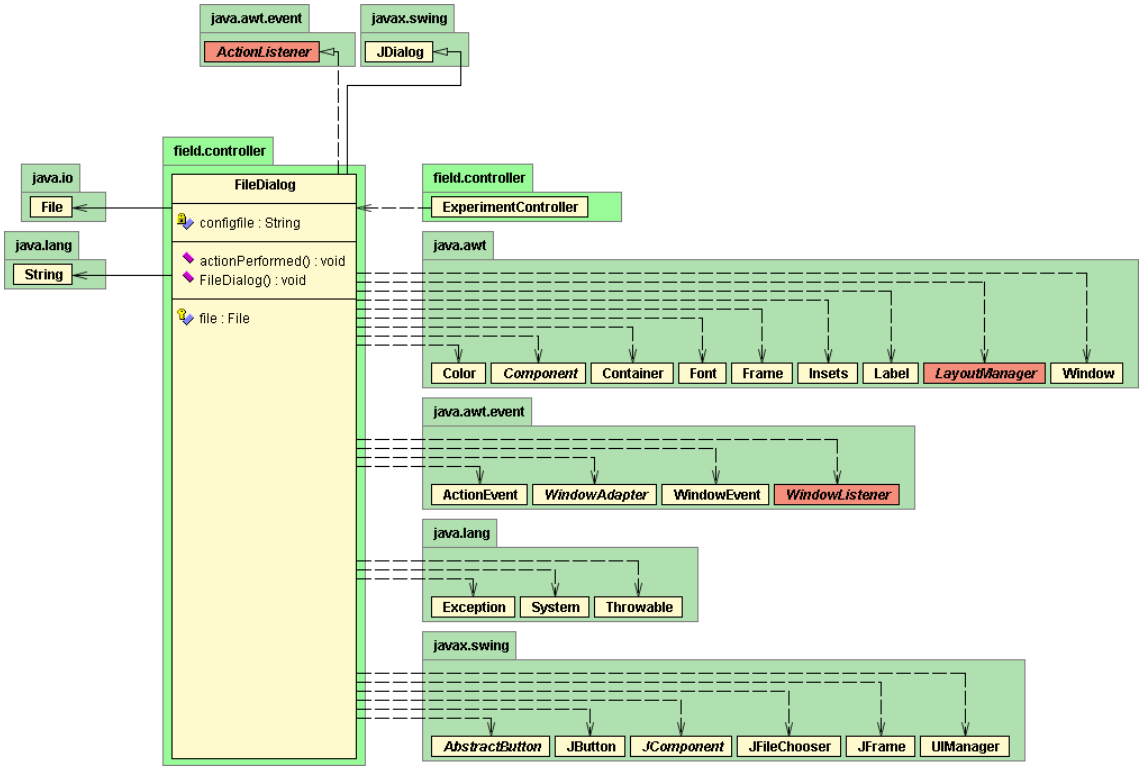
y - 緯度

r - 方向角

戻り値:

平面直角座標(X,Y)、方向角（ラジアン）

■ 4　クラス FileDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.controller.FileDialog
```

---

public class **FileDialog**

extends javax.swing.JDialog

implements java.awt.event.ActionListener

地図設定ファイル選択画面クラス

フィールドの概要	
private java.lang.String	**configfile** 地図設定ファイル名
private java.io.File	**file** ファイル名

コンストラクタの概要
FileDialog(java.lang.String[] args) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント
protected java.io.File	**getFile**() 地図設定ファイル名の取得

フィールドの詳細

file

private java.io.File **file**

ファイル名

---

configfile

private java.lang.String **configfile**

地図設定ファイル名

コンストラクタの詳細

FileDialog

public **FileDialog**(java.lang.String[] args)

コンストラクタ

パラメータ:

args - 実行時受け渡しパラメータ（設定ファイル名）

メソッドの詳細

getFile

protected java.io.File **getFile**()

地図設定ファイル名の取得

戻り値:

ファイルクラス

---

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

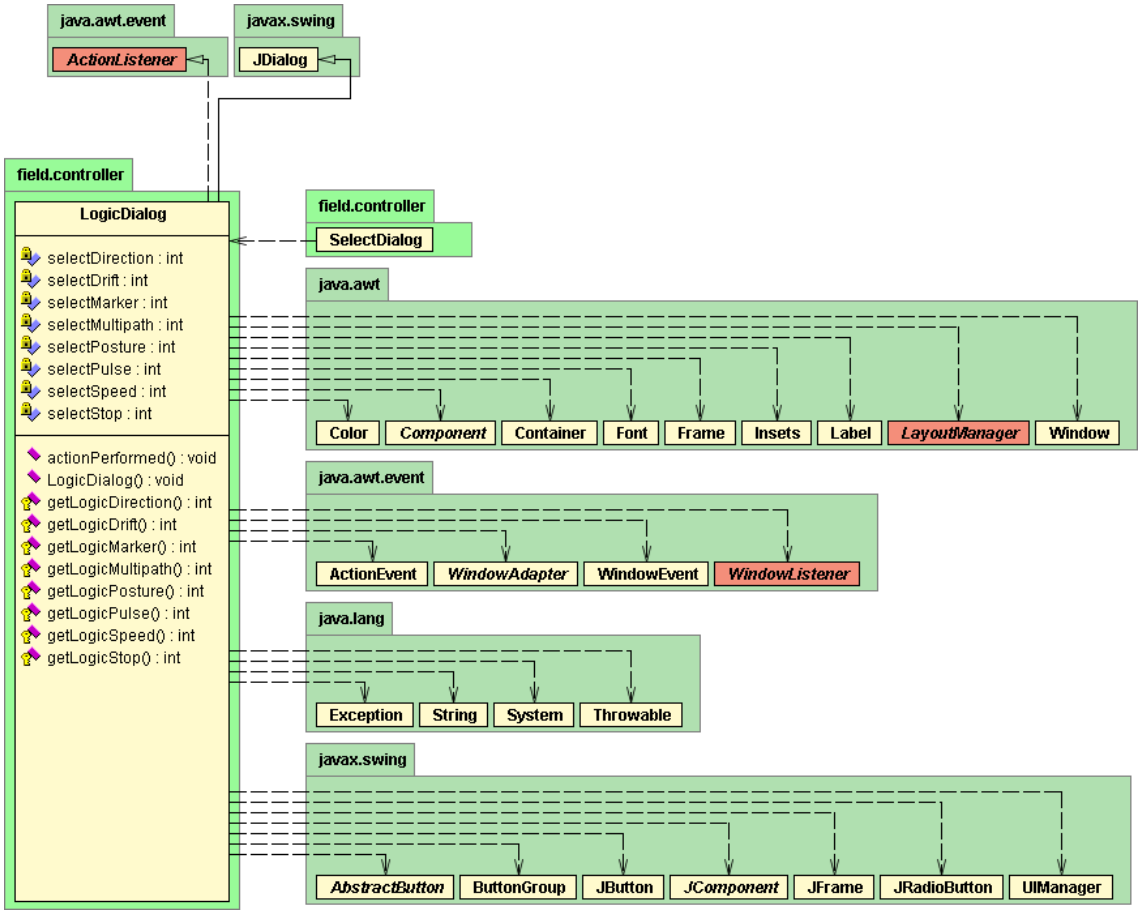
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

■ ５　クラス LogicDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.controller.LogicDialog
```

---

public class **LogicDialog**

extends javax.swing.JDialog

implements java.awt.event.ActionListener

位置計算処理ロジック選択画面クラス

フィールドの概要	
private int	**selectDirection** 方位算出選択値
private int	**selectDrift** ドリフト補正選択値
private int	**selectMarker** マーカー補正選択値
private int	**selectMultipath** マルチパス補正選択値
private int	**selectPosture** 姿勢補正選択値
private int	**selectPulse** ﾊﾟﾙｽｽｹｰﾙ補正選択値
private int	**selectSpeed** 速度算出選択値
private int	**selectStop** 停止判定選択値

コンストラクタの概要
LogicDialog() コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント

protected int	**getLogicDirection**() 方位算出選択値の取得
protected int	**getLogicDrift**() ドリフト補正選択値の取得
protected int	**getLogicMarker**() マーカー補正選択値の取得
protected int	**getLogicMultipath**() マルチパス補正選択値の取得
protected int	**getLogicPosture**() 姿勢補正選択値の取得
protected int	**getLogicPulse**() ﾊﾟﾙｽｽｹｰﾙ補正選択値の取得
protected int	**getLogicSpeed**() 速度算出選択値の取得
protected int	**getLogicStop**() 停止判定選択値の取得

フィールドの詳細

selectStop

private int **selectStop**

停止判定選択値

---

selectDrift

private int **selectDrift**

ドリフト補正選択値

---

selectPulse

private int **selectPulse**

ﾊﾟﾙｽｽｹｰﾙ補正選択値

---

selectPosture

private int **selectPosture**

姿勢補正選択値

---

selectDirection

private int **selectDirection**

方位算出選択値

---

selectSpeed

private int **selectSpeed**

速度算出選択値

---

selectMultipath

private int **selectMultipath**

マルチパス補正選択値

---

selectMarker

private int **selectMarker**

マーカー補正選択値

コンストラクタの詳細

LogicDialog

public **LogicDialog**()

コンストラクタ

メソッドの詳細

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

---

getLogicStop

protected int **getLogicStop**()

停止判定選択値の取得

戻り値:

停止判定選択値

---

getLogicDrift

protected int **getLogicDrift**()

ドリフト補正選択値の取得

戻り値:

ドリフト補正選択値

---

getLogicPulse

protected int **getLogicPulse**()

ﾊﾟﾙｽｽｹｰﾙ補正選択値の取得

戻り値:

ﾊﾟﾙｽｽｹｰﾙ補正選択値

---

getLogicPosture

protected int **getLogicPosture**()

姿勢補正選択値の取得

戻り値:

姿勢補正選択値

---

getLogicDirection

protected int **getLogicDirection**()

方位算出選択値の取得

戻り値:

方位算出選択値

---

getLogicSpeed

protected int **getLogicSpeed**()

速度算出選択値の取得

戻り値:

速度算出選択値

---

getLogicMultipath

protected int **getLogicMultipath**()

マルチパス補正選択値の取得

戻り値:

マルチパス補正選択値

---

getLogicMarker

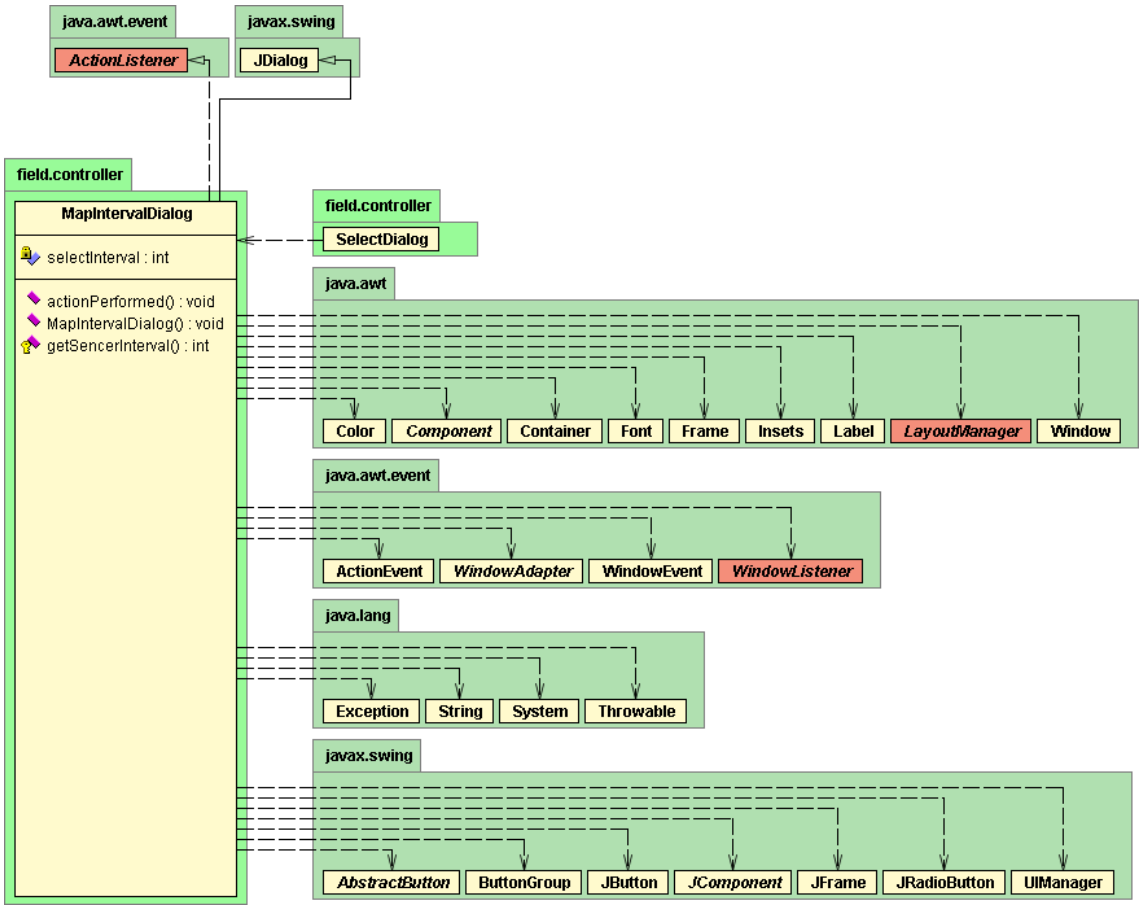
protected int **getLogicMarker**()

マーカー補正選択値の取得

戻り値:

マーカー補正選択値

■ ６　クラス MapIntervalDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.controller.MapIntervalDialog
```

---

public class MapIntervalDialog

extends javax.swing.JDialog

implements java.awt.event.ActionListener

画面表示切換えインターバル選択画面クラス

フィールドの概要	
private int	**selectInterval** インターバル選択値

コンストラクタの概要
MapIntervalDialog() コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント
protected int	**getSencerInterval**() 地図描画インターバル選択値の取得

フィールドの詳細

selectInterval

private int **selectInterval**

インターバル選択値

コンストラクタの詳細

MapIntervalDialog

public **MapIntervalDialog**()

コンストラクタ

メソッドの詳細

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

---

getSencerInterval

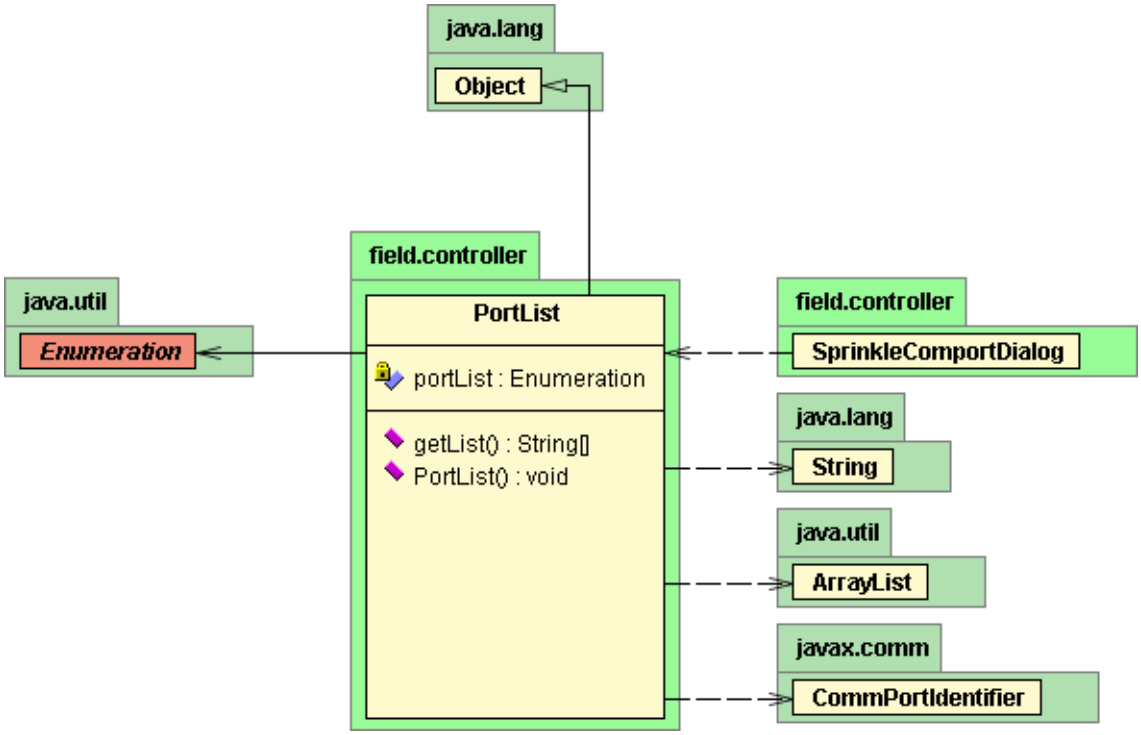
protected int **getSencerInterval**()

地図描画インターバル選択値の取得

戻り値:

インターバル値

■ 7　クラス PortList



java.lang.Object

　|

　+--**field.controller.PortList**

---

public class **PortList**

extends java.lang.Object

使用できる COM ポートの一覧を取得する

フィールドの概要	
private　java.util.Enumeration	**portList** 　　ポートリスト

コンストラクタの概要
PortList() 　　コンストラクタ

メソッドの概要	
java.lang.String[]	**getList**() ポートのリストを取得

フィールドの詳細

portList

private java.util.Enumeration **portList**

ポートリスト

コンストラクタの詳細

PortList

public **PortList**()

コンストラクタ

メソッドの詳細

getList

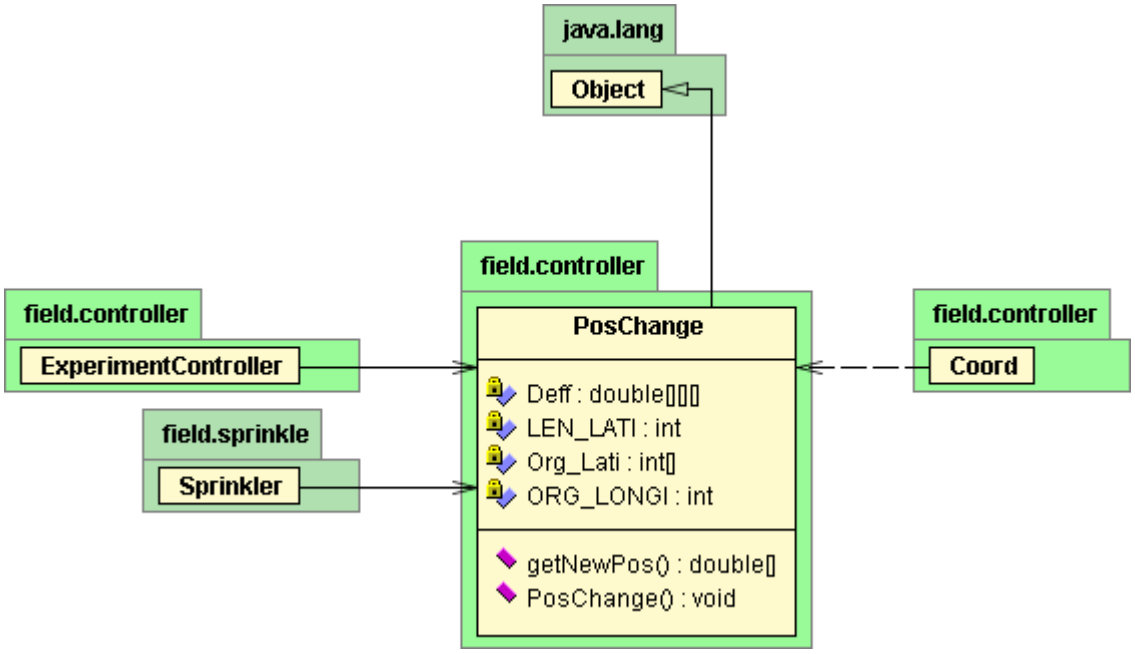
public java.lang.String[] **getList**()

ポートのリストを取得

戻り値:

ポート名の格納された配列

■ 8　クラス PosChange



java.lang.Object

　|

　+--**field.controller.PosChange**

---

public class **PosChange**

extends java.lang.Object

世界測地系緯度経度を日本測地系緯度経度に変換するクラス

フィールドの概要	
private　double[][][]	**Deff** 緯度経度変換テーブル
private　int	**LEN_LATI** 緯度のﾃｰﾌﾞﾙの幅（40′）
private　int[]	**Org_Lati** 緯度の対象区域判定テーブル
private　int	**ORG_LONGI** 変換対象区域の経度設定 （対象区域が経度 129°～）

コンストラクタの概要
PosChange() 　　コンストラクタ

メソッドの概要	
double[]	**getNewPos**(double lLongitude, double lLatitude, int sw) 　　世界測地系<->日本測地系の緯度経度の変換を行う

フィールドの詳細

ORG_LONGI

private int **ORG_LONGI**

変換対象区域の経度設定

（対象区域が経度 129°～）

LEN_LATI

private int **LEN_LATI**

緯度のﾃｰﾌﾞﾙの幅（40′）

Org_Lati

private int[] **Org_Lati**

緯度の対象区域判定テーブル

Deff

private double[][][] **Deff**

緯度経度変換テーブル

コンストラクタの詳細

PosChange

public **PosChange**()

コンストラクタ

メソッドの詳細

getNewPos

public double[] **getNewPos**(double lLongitude,
double lLatitude,
int sw)

世界測地系<->日本測地系の緯度経度の変換を行う

パラメータ:

lLongitude - 経度

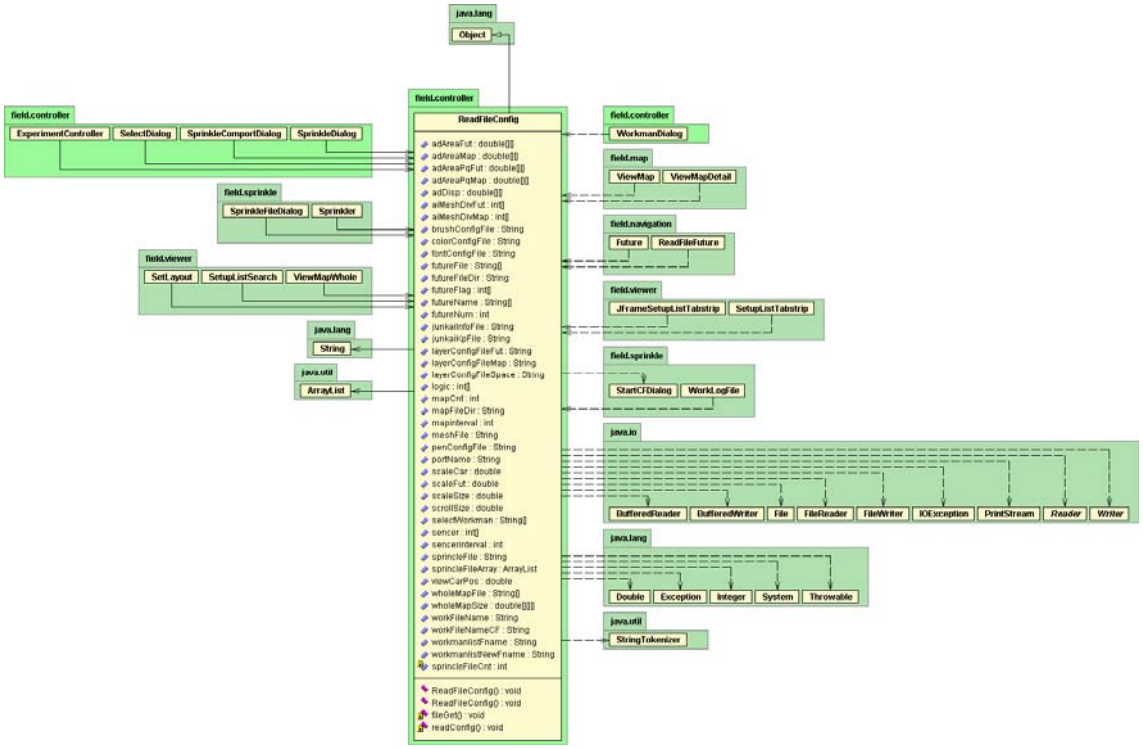
lLatitude - 緯度

sw - 0:世界->日本 1:日本->世界

戻り値:

変換後緯度経度 [0]:経度 [1]:緯度

■ ９　クラス ReadFileConfig



java.lang.Object

  |

  +--**field.controller.ReadFileConfig**

---

public class **ReadFileConfig**

extends java.lang.Object

設定ファイルの内容を読込クラス

フィールドの概要	
double[][]	**adAreaFut** 道路地物の実地図領域における原点 X,Y,終点 X,Y
double[][]	**adAreaMap** 背景地図の実地図領域における原点 X,Y,終点 X,Y
double[][]	**adAreaPqFut** 道路地物の PQ ファイル領域における原点 X,Y,終点 X,Y
double[][]	**adAreaPqMap** 背景地図の PQ ファイル領域における原点 X,Y,終点 X,Y

double[][]	**adDisp** 初期表示領域サイズにおける原点 X,Y,終点 X,Y
int[]	**aiMeshDivFut** 道路地物のメッシュ分割数
int[]	**aiMeshDivMap** 背景地図のメッシュ分割数
java.lang.String	**brushConfigFile** ブラシ設定ファイル名
java.lang.String	**colorConfigFile** カラー設定ファイル名
java.lang.String	**fontConfigFile** フォント設定ファイル名
java.lang.String[]	**futureFile** 地物検索情報ファイル名
java.lang.String	**futureFileDir** 道路地物ファイル配置ディレクトリ名
int[]	**futureFlag** 地物検索情報検索フラグ
java.lang.String[]	**futureName** 地物検索情報地物名
int	**futureNum** 地物検索ファイル数
java.lang.String	**junkaiInfoFile** 巡回記録用巡回情報ファイル名
java.lang.String	**junkaiKpFile** 巡回記録用時間 KP 対応ファイル名
java.lang.String	**layerConfigFileFut** 道路地物レイヤー設定ファイル名
java.lang.String	**layerConfigFileMap** 背景地図レイヤー設定ファイル名
java.lang.String	**layerConfigFileSpace**

	自車位置等レイヤー設定ファイル名
int[]	**logic** 位置計算処理ロジック 0:停止判定、1:ドリフト補正、2:パルススケール補正、3:姿勢補正、4:方位算出、5:速度算出、6:マルチパス補正、7:マーカ補正
int	**mapCnt** 地図ファイル数
java.lang.String	**mapFileDir** 背景地図 PQ ファイル配置ディレクトリ名
int	**mapinterval** 背景地図表示インターバル
java.lang.String	**meshFile** 地物検索範囲判定用メッシュファイル名
java.lang.String	**penConfigFile** ペン設定ファイル名
java.lang.String	**portName** 散布制御装置接続 RS-232C ポート名
double	**scaleCar** 自車位置マークのスケール
double	**scaleFut** 探査地物マークのスケール
double	**scaleSize** スケール設定（拡大縮小時変化率）
double	**scrollSize** スクロール設定（１回の移動距離）
java.lang.String[]	**selectWorkman** 作業者名リスト
int[]	**sencer** 使用センサーの配列番号（複数使用できる場合の配列番号）0:GPS センサ、1:自律センサ、2:車速センサ、3:レーンマーカセンサ
int	**sencerinterval**

	センサーデータ取得インターバル（Hz で指定）
java.lang.String	**sprincleFile** 散布制御ファイル名（デフォルトファイル）
java.util.ArrayList	**sprincleFileArray** 散布制御ファイル名（全てのファイル）
private int	**sprincleFileCnt** 散布制御ファイル数
double	**viewCarPos** 表示中心変位
java.lang.String[]	**wholeMapFile** 全体地図ファイル名(ファイルは 320×200 で GIF または JPEG で作成)
double[][][]	**wholeMapSize** 全体地図の実地図領域における原点 X,Y,終点 X,Y
java.lang.String	**workFileName** 作業記録格納ファイル名（作業用）
java.lang.String	**workFileNameCF** 作業記録格納ファイル名（コンパクトフラッシュ用）
java.lang.String	**workmanlistFname** 作業者名ファイル名
java.lang.String	**workmanlistNewFname** 作業者名ファイル名（書き換え用）

コンストラクタの概要
ReadFileConfig(java.lang.String fileName) コンストラクタ
ReadFileConfig(java.lang.String propertyName, boolean ini) コンストラクタ

メソッドの概要	
private void	**fileGet**(java.lang.String targetFile) ファイルフィルターによるファイル名の取得
private void	**readConfig**(java.lang.String fileName) 設定ファイルの読込

フィールドの詳細

mapFileDir

public java.lang.String **mapFileDir**

背景地図 PQ ファイル配置ディレクトリ名

---

futureFileDir

public java.lang.String **futureFileDir**

道路地物ファイル配置ディレクトリ名

---

colorConfigFile

public java.lang.String **colorConfigFile**

カラー設定ファイル名

---

brushConfigFile

public java.lang.String **brushConfigFile**

ブラシ設定ファイル名

---

penConfigFile

public java.lang.String **penConfigFile**

ペン設定ファイル名

fontConfigFile

public java.lang.String **fontConfigFile**
フォント設定ファイル名

---

layerConfigFileMap

public java.lang.String **layerConfigFileMap**
背景地図レイヤー設定ファイル名

---

layerConfigFileFut

public java.lang.String **layerConfigFileFut**
道路地物レイヤー設定ファイル名

---

layerConfigFileSpace

public java.lang.String **layerConfigFileSpace**
自車位置等レイヤー設定ファイル名

---

adAreaMap

public double[][] **adAreaMap**
背景地図の実地図領域における原点 X,Y,終点 X,Y

---

adAreaFut

public double[][] **adAreaFut**
道路地物の実地図領域における原点 X,Y,終点 X,Y

---

adAreaPqMap

public double[][] **adAreaPqMap**

背景地図の PQ ファイル領域における原点 X,Y,終点 X,Y

---

adAreaPqFut

public double[][] **adAreaPqFut**

道路地物の PQ ファイル領域における原点 X,Y,終点 X,Y

---

aiMeshDivMap

public int[] **aiMeshDivMap**

背景地図のメッシュ分割数

---

aiMeshDivFut

public int[] **aiMeshDivFut**

道路地物のメッシュ分割数

---

adDisp

public double[][] **adDisp**

初期表示領域サイズにおける原点 X,Y,終点 X,Y

---

wholeMapFile

public java.lang.String[] **wholeMapFile**

全体地図ファイル名（ファイルは 320×200 で GIF または JPEG で作成）

---

wholeMapSize

public double[][][] **wholeMapSize**

全体地図の実地図領域における原点 X,Y,終点 X,Y

---

viewCarPos

public double **viewCarPos**

表示中心変位

---

scaleSize

public double **scaleSize**

スケール設定（拡大縮小時変化率）

---

scaleCar

public double **scaleCar**

自車位置マークのスケール

---

scaleFut

public double **scaleFut**

探査地物マークのスケール

---

scrollSize

public double **scrollSize**

スクロール設定（１回の移動距離）

---

mapinterval

public int **mapinterval**

背景地図表示インターバル

---

sencer

public int[] **sencer**

使用センサーの配列番号（複数使用できる場合の配列番号） 0:GPS センサ、1:自律センサ、2:車

速センサ、3:レーンマーカセンサ

---

sencerinterval

public int **sencerinterval**
センサーデータ取得インターバル（Hz で指定）

---

logic

public int[] **logic**
位置計算処理ロジック 0:停止判定、1:ドリフト補正、2:パルススケール補正、3:姿勢補正、4:方位算出、5:速度算出、6:マルチパス補正、7:マーカ補正

---

selectWorkman

public java.lang.String[] **selectWorkman**
作業者名リスト

---

workFileName

public java.lang.String **workFileName**
作業記録格納ファイル名（作業用）

---

workFileNameCF

public java.lang.String **workFileNameCF**
作業記録格納ファイル名（コンパクトフラッシュ用）

---

futureNum

public int **futureNum**
地物検索ファイル数

---

futureName

public java.lang.String[] **futureName**

地物検索情報地物名

---

futureFlag

public int[] **futureFlag**

地物検索情報検索フラグ

---

futureFile

public java.lang.String[] **futureFile**

地物検索情報ファイル名

---

meshFile

public java.lang.String **meshFile**

地物検索範囲判定用メッシュファイル名

---

portName

public java.lang.String **portName**

散布制御装置接続 RS-232C ポート名

---

sprincleFileCnt

private int **sprincleFileCnt**

散布制御ファイル数

---

sprincleFileArray

public java.util.ArrayList **sprincleFileArray**

散布制御ファイル名（全てのファイル）

sprincleFile

public java.lang.String **sprincleFile**

散布制御ファイル名（デフォルトファイル）

---

mapCnt

public int **mapCnt**

地図ファイル数

---

workmanlistFname

public java.lang.String **workmanlistFname**

作業者名ファイル名

---

workmanlistNewFname

public java.lang.String **workmanlistNewFname**

作業者名ファイル名（書き換え用）

---

junkaiKpFile

public java.lang.String **junkaiKpFile**

巡回記録用時間 KP 対応ファイル名

---

junkaiInfoFile

public java.lang.String **junkaiInfoFile**

巡回記録用巡回情報ファイル名

コンストラクタの詳細

ReadFileConfig

public **ReadFileConfig**(java.lang.String fileName)

コンストラクタ

パラメータ:

fileName - 設定ファイル名

ReadFileConfig

public **ReadFileConfig**(java.lang.String propertyName,
boolean ini)

コンストラクタ

パラメータ:

fileName - 設定ファイル名

ini - 設定ファイルの有無 true:有 false:無

メソッドの詳細

readConfig

private void **readConfig**(java.lang.String fileName)

設定ファイルの読込

パラメータ:

fileName - 設定ファイル名

---

fileGet

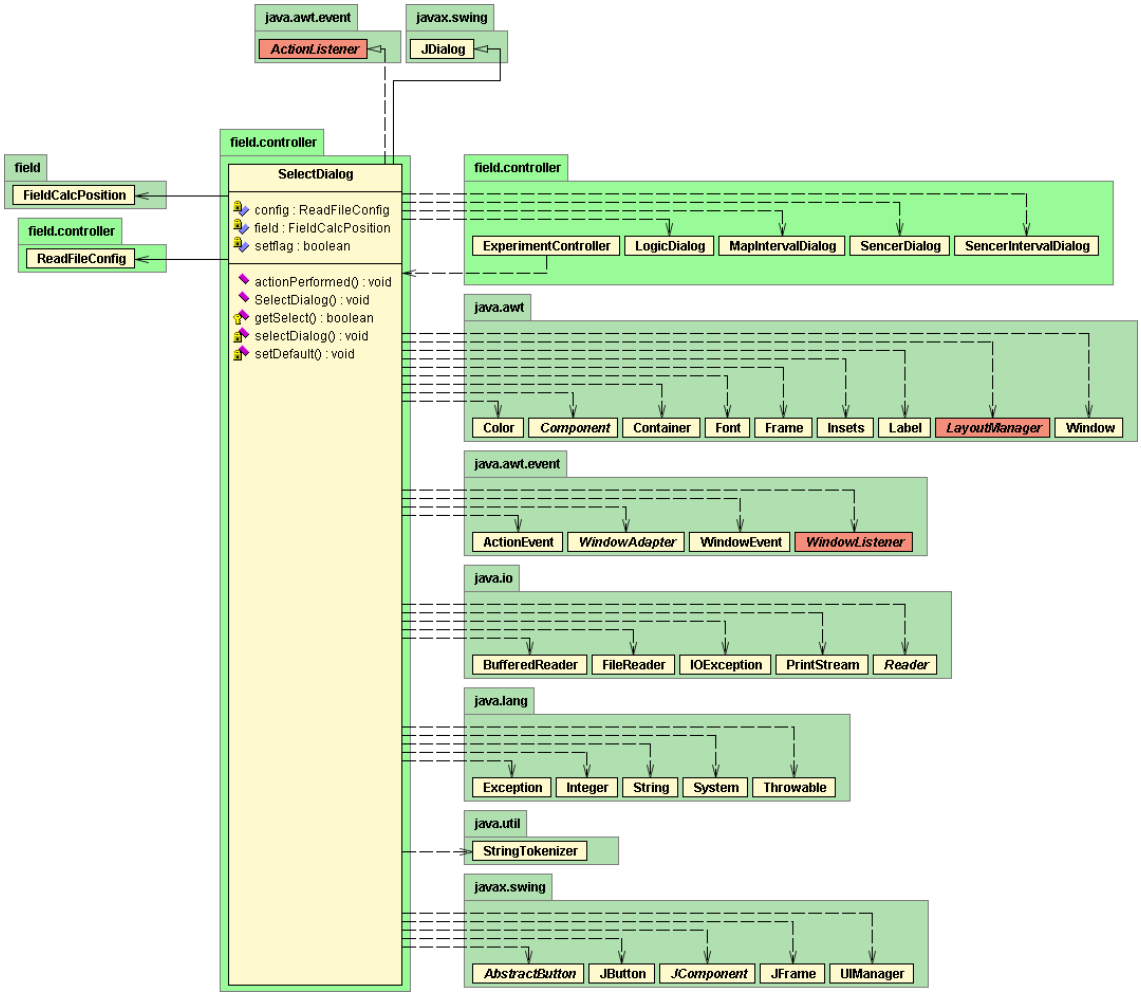
private void **fileGet**(java.lang.String targetFile)

ファイルフィルターによるファイル名の取得

パラメータ:

targetFile - ワイルドカード対応ファイル名

■ １０　クラス SelectDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.controller.SelectDialog
```

public class **SelectDialog**

extends javax.swing.JDialog

implements java.awt.event.ActionListener

位置取得情報設定選択確認画面クラス

フィールドの概要	
private ReadFileConfig	**config** ReadFileConfig クラスのインスタンス
private FieldCalcPosition	**field** FieldCalcPosition クラスのインスタンス
private boolean	**setflag** 設定の有無

コンストラクタの概要
SelectDialog(ReadFileConfig readfileconfig, FieldCalcPosition fieldcalcposition) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント
protected boolean	**getSelect**() 位置情報設定有無状態の取得
private void	**selectDialog**() 位置取得情報設定用各ダイアログの表示
private void	**setDefault**()

	位置取得情報設定を標準設定ファイルから読込

フィールドの詳細

setflag

private boolean setflag

設定の有無

---

config

private ReadFileConfig config

ReadFileConfig クラスのインスタンス

---

field

private FieldCalcPosition field

FieldCalcPosition クラスのインスタンス

コンストラクタの詳細

SelectDialog

public SelectDialog(ReadFileConfig readfileconfig,
FieldCalcPosition fieldcalcposition)

コンストラクタ

パラメータ:

readfileconfig - 設定ファイル読込クラス

fieldcalcposition - 位置情報取得クラス

メソッドの詳細

getSelect

protected boolean **getSelect**()

位置情報設定有無状態の取得

戻り値:

true:設定する false:デフォルトを使用

---

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

---

selectDialog

private void **selectDialog**()

位置取得情報設定用各ダイアログの表示

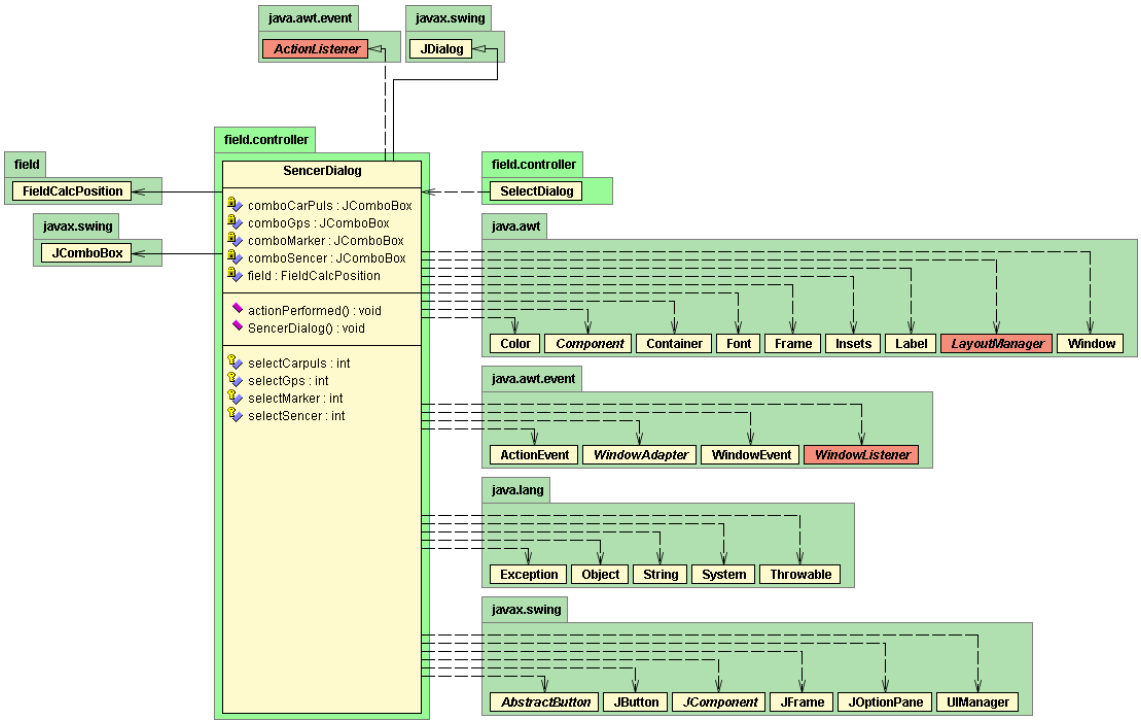
---

setDefault

private void **setDefault**()

位置取得情報設定を標準設定ファイルから読込

■ １１　クラス SencerDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.controller.SencerDialog
```

---

public class **SencerDialog**

extends javax.swing.JDialog

implements java.awt.event.ActionListener

使用センサー選択画面クラス。

フィールドの概要	
private javax.swing.JComboBox	**comboCarPuls** 車速センサー用コンボボックス
private javax.swing.JComboBox	**comboGps** GPS センサー用コンボボックス
private javax.swing.JComboBox	**comboMarker** マーカーセンサー用コンボボックス
private javax.swing.JComboBox	**comboSencer** 自律センサー用コンボボックス
private FieldCalcPosition	**field** 位置特定処理クラスのインスタンス
private int	**selectCarpuls** 選択された車速センサー名
private int	**selectGps** 選択された GPS センサー名
private int	**selectMarker** 選択されたマーカーセンサー名
private int	**selectSencer** 選択された自律センサー名

コンストラクタの概要
SencerDialog(FieldCalcPosition field) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント
protected int	**getSelectCarpuls**()

	車速センサー番号の取得
protected int	**getSelectGps**() GPS センサー番号の取得
protected int	**getSelectMarker**() マーカーセンサー番号の取得
protected int	**getSelectSencer**() 自律センサー番号の取得

フィールドの詳細

field

private FieldCalcPosition **field**

位置特定処理クラスのインスタンス

---

comboGps

private javax.swing.JComboBox **comboGps**

GPS センサー用コンボボックス

---

comboSencer

private javax.swing.JComboBox **comboSencer**

自律センサー用コンボボックス

---

comboCarPuls

private javax.swing.JComboBox **comboCarPuls**

車速センサー用コンボボックス

---

comboMarker

private javax.swing.JComboBox **comboMarker**

マーカーセンサー用コンボボックス

---

selectGps

private int **selectGps**

選択された GPS センサー名

---

selectSencer

private int **selectSencer**

選択された自律センサー名

---

selectCarpuls

private int **selectCarpuls**

選択された車速センサー名

---

selectMarker

private int **selectMarker**

選択されたマーカーセンサー名

コンストラクタの詳細

SencerDialog

public **SencerDialog**(FieldCalcPosition field)

コンストラクタ

パラメータ:

field - 位置特定情報取得クラス

メソッドの詳細

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

---

getSelectGps

protected int **getSelectGps**()

GPS センサー番号の取得

戻り値:

GPS センサー番号

---

getSelectSencer

protected int **getSelectSencer**()

自律センサー番号の取得

戻り値:

自律センサー番号

---

getSelectCarpuls

protected int **getSelectCarpuls**()

車速センサー番号の取得

戻り値:

車速センサー番号

---

getSelectMarker

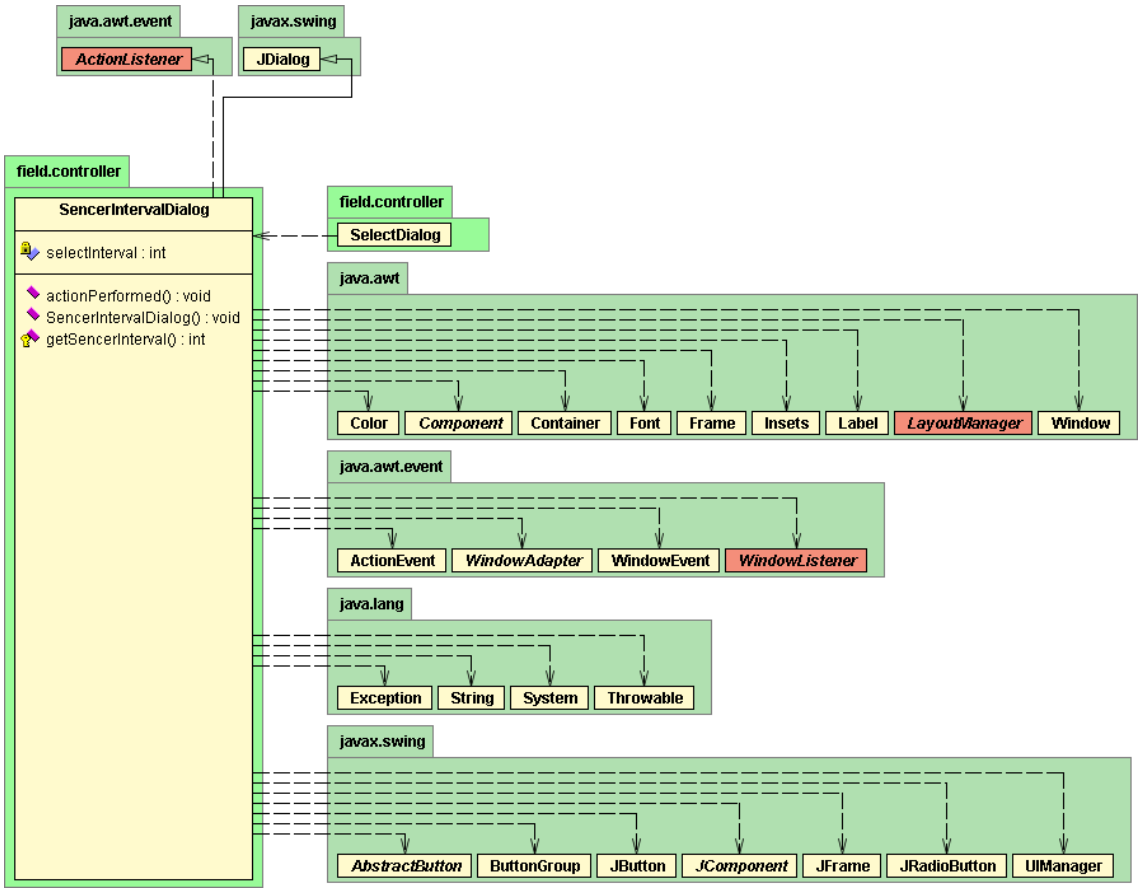
protected int **getSelectMarker**()

マーカーセンサー番号の取得

戻り値:

マーカーセンサー番号

■ １２　クラス SencerIntervalDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.controller.SencerIntervalDialog
```

---

public class SencerIntervalDialog

extends javax.swing.JDialog

implements java.awt.event.ActionListener

センサーデータ取得インターバル選択画面クラス

フィールドの概要	
private int	**selectInterval** インターバル選択値

コンストラクタの概要
SencerIntervalDialog() コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント
protected int	**getSencerInterval**() センサーインターバル選択値の取得

フィールドの詳細

selectInterval

private int **selectInterval**

インターバル選択値

コンストラクタの詳細

SencerIntervalDialog

public **SencerIntervalDialog**()

コンストラクタ

メソッドの詳細

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

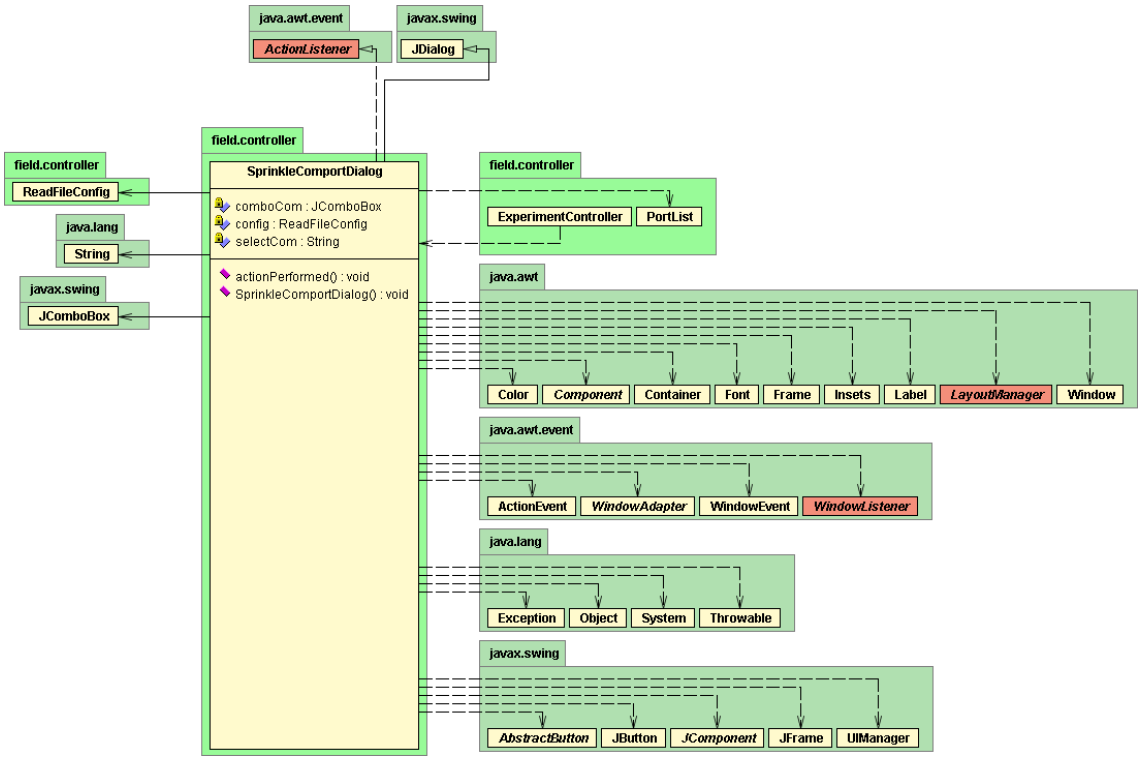
---

getSencerInterval

protected int **getSencerInterval**()

センサーインターバル選択値の取得

■ １３　クラス SprinkleComportDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.controller.SprinkleComportDialog
```

---

public class SprinkleComportDialog

extends javax.swing.JDialog

implements java.awt.event.ActionListener

散布車制御用 COM ポート選択画面クラス

フィールドの概要	
private javax.swing.JComboBox	**comboCom** COM ポート用コンボボックス
private ReadFileConfig	**config** 設定ファイル読込クラス
private java.lang.String	**selectCom** 選択された COM ポート名

コンストラクタの概要
SprinkleComportDialog(ReadFileConfig readfileconfig) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント

フィールドの詳細

comboCom

private javax.swing.JComboBox **comboCom**

COM ポート用コンボボックス

---

selectCom

private java.lang.String **selectCom**

選択された COM ポート名

config

private ReadFileConfig **config**

設定ファイル読込クラス

コンストラクタの詳細

SprinkleComportDialog

public **SprinkleComportDialog**(ReadFileConfig readfileconfig)

コンストラクタ

パラメータ:

field - 位置特定情報取得クラス

メソッドの詳細

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

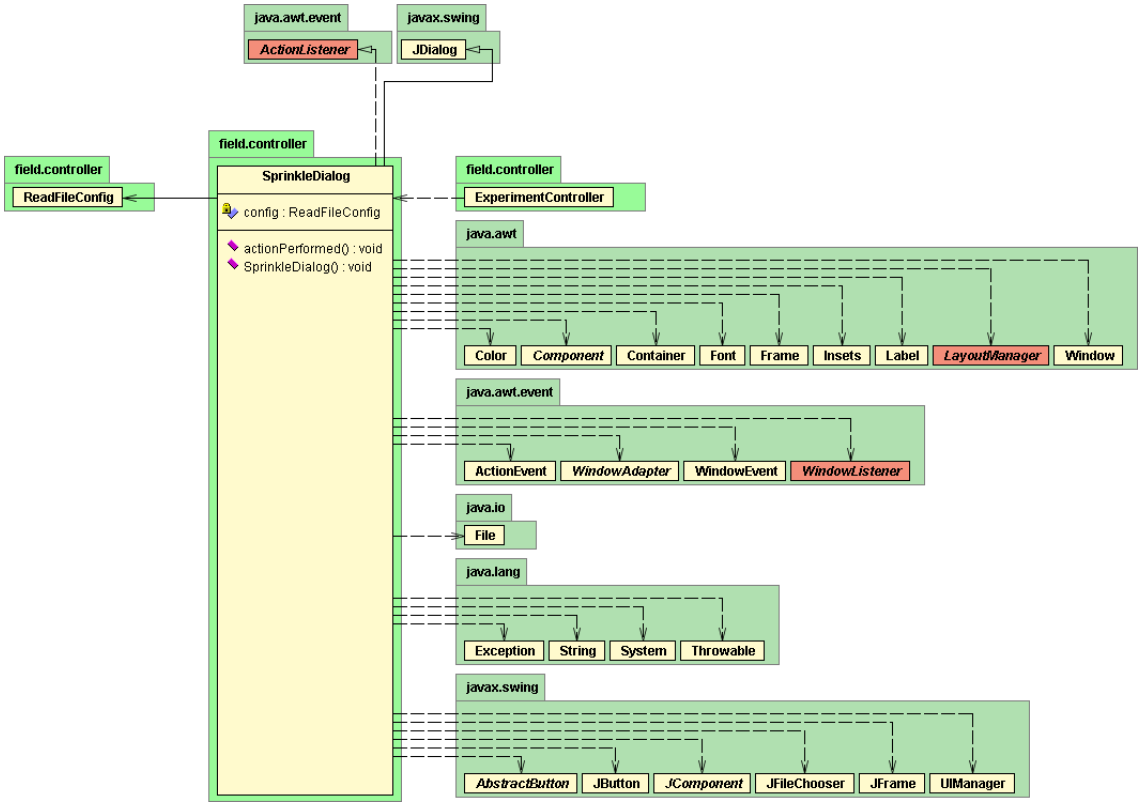
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

■ １４　クラス SprinkleDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.controller.SprinkleDialog
```

---

public class **SprinkleDialog**

extends javax.swing.JDialog

implements java.awt.event.ActionListener

散布計画ファイル選択画面クラス

フィールドの概要	
private ReadFileConfig	**config** 設定ファイル読込クラス

コンストラクタの概要
SprinkleDialog(ReadFileConfig readfileconfig) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント

フィールドの詳細

config

private ReadFileConfig **config**

設定ファイル読込クラス

コンストラクタの詳細

SprinkleDialog

public **SprinkleDialog**(ReadFileConfig readfileconfig)

コンストラクタ

パラメータ:

readfileconfig - 設定ファイル読込クラス

メソッドの詳細

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

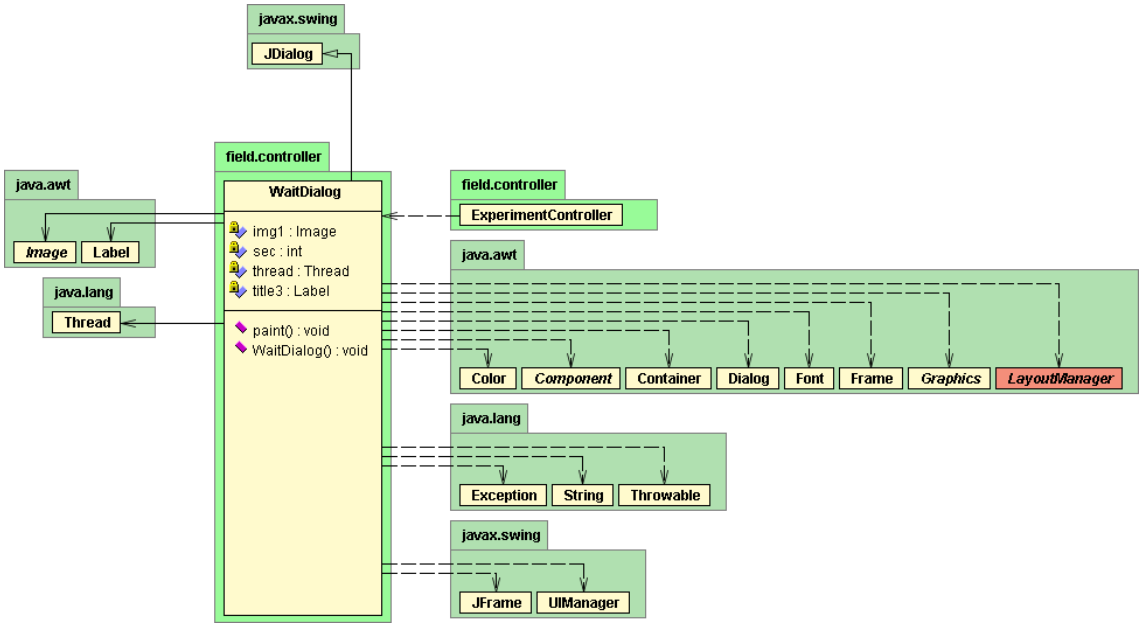
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

■ １５　クラス WaitDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.controller.WaitDialog
```

---

public class **WaitDialog**

extends javax.swing.JDialog

センサー初期化待ち画面クラス

フィールドの概要

private java.awt.Image	**img1** アイコンイメージ
private int	**sec** 待ち時間（秒）
private java.lang.Thread	**thread** スレッドクラス
private java.awt.Label	**title3** 待ち時間表示ラベル

コンストラクタの概要
WaitDialog() コンストラクタ

メソッドの概要	
void	**paint**(java.awt.Graphics g) ペイント処理 待ち時間の描画

フィールドの詳細

img1

private java.awt.Image **img1**

アイコンイメージ

---

title3

private java.awt.Label **title3**

待ち時間表示ラベル

sec

private int **sec**

待ち時間（秒）

---

thread

private java.lang.Thread **thread**

スレッドクラス

コンストラクタの詳細

WaitDialog

public **WaitDialog**()

コンストラクタ

メソッドの詳細

paint

public void **paint**(java.awt.Graphics g)

ペイント処理

待ち時間の描画

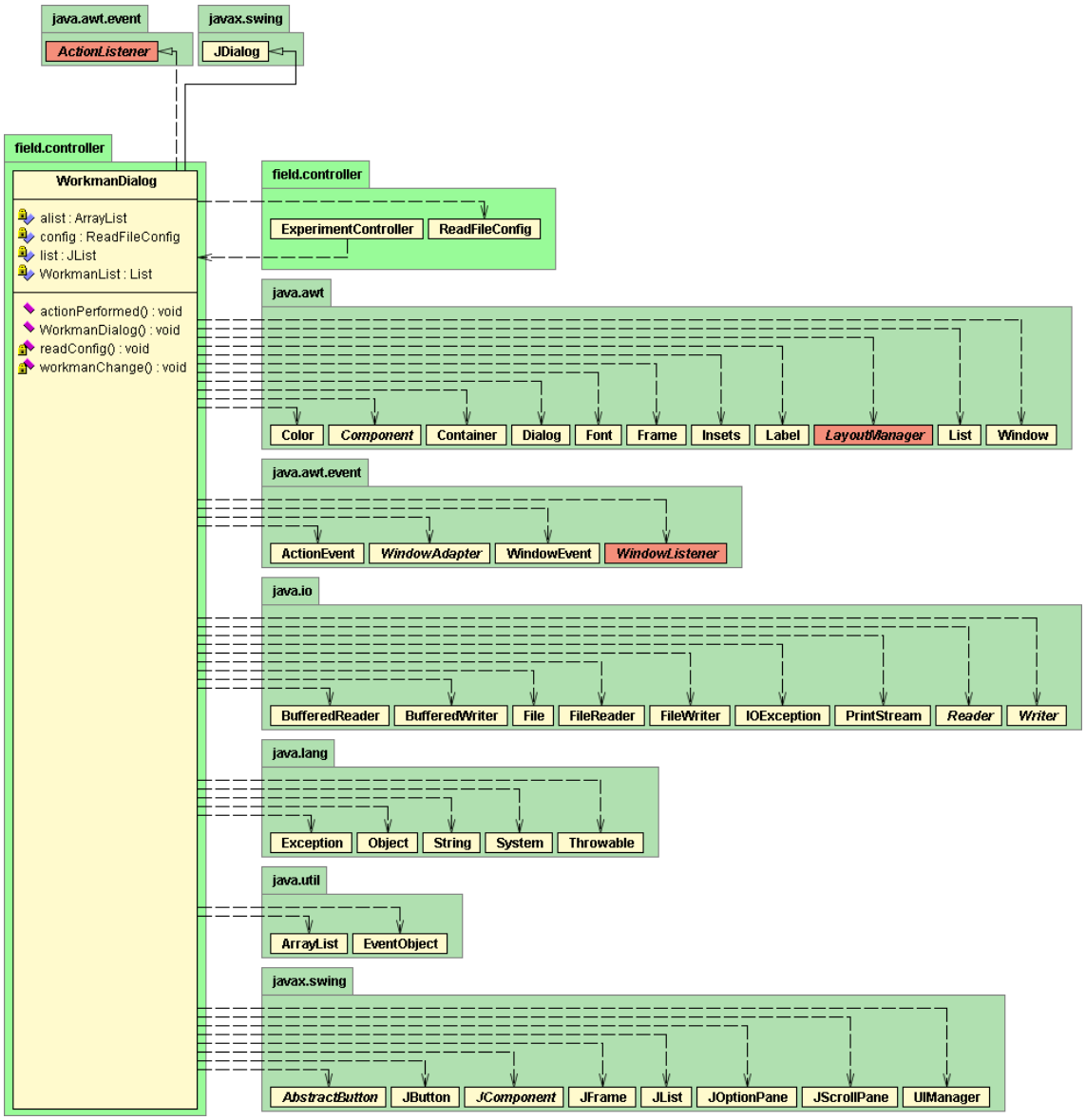
オーバーライド:

クラス java.awt.Container 内の paint

パラメータ:

g - Graphics

■ １６　クラス WorkmanDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
```

public class **WorkmanDialog**

extends javax.swing.JDialog

implements java.awt.event.ActionListener

作業者名を選択する画面クラス

フィールドの概要	
private java.util.ArrayList	**alist** リストボックス表示用作業領域
private ReadFileConfig	**config** 設定ファイルクラス
private javax.swing.JList	**list** 選択作業者名
private java.awt.List	**WorkmanList** 作業者名用コンボボックス

コンストラクタの概要
WorkmanDialog(ReadFileConfig conf) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント
private void	**readConfig**(java.lang.String fileName) 作業員名リストファイルの読込
private void	**workmanChange**(java.lang.String oldFileName,

	java.lang.String newFileName) 　　作業員名リストファイルの書き換え

フィールドの詳細

WorkmanList

private java.awt.List WorkmanList

作業者名用コンボボックス

---

list

private javax.swing.JList list

選択作業者名

---

alist

private java.util.ArrayList alist

リストボックス表示用作業領域

---

config

private ReadFileConfig config

設定ファイルクラス

コンストラクタの詳細

WorkmanDialog

public WorkmanDialog(ReadFileConfig conf)

コンストラクタ

パラメータ:

conf - 設定ファイル読込クラス

メソッドの詳細

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

---

readConfig

private void **readConfig**(java.lang.String fileName)

作業員名リストファイルの読込

パラメータ:

fileName 作業員名リストファイル名 -

---

workmanChange

private void **workmanChange**(java.lang.String oldFileName,
java.lang.String newFileName)

作業員名リストファイルの書き換え

パラメータ:

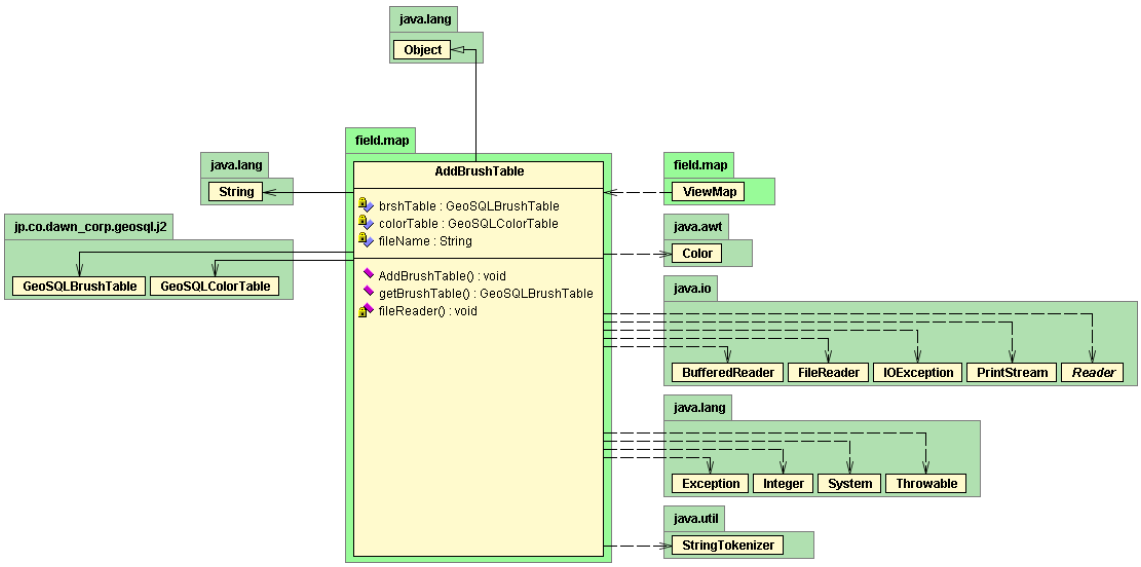
fileName 作業員名リストファイル名 -

---

c） パッケージ field.map

クラスの概要	
AddBrushTable	GeoSQL 表示用ブラシ設定ファイルを読み込んで、設定内容を保持するクラス
AddColorTable	GeoSQL 表示用カラー設定ファイルを読み込んで、設定内容を保持するクラス
AddFontTable	GeoSQL 表示用フォント設定ファイルを読み込んで、設定内容を保持するクラス
AddLayerTable	GeoSQL 表示用レイヤー設定ファイルを読み込んで、設定内容を保持するクラス
AddPenTable	GeoSQL 表示用ペン設定ファイルを読み込んで、設定内容を保持するクラス
AddSymbolTable	シンボルデータを作成するクラス（GeoSQL 用）
ViewMap	GeoSQL を使用して、背景地図、道路地物表示を表示するクラス
ViewMapDetail	地図上に現在位置と接近地物を表示するクラス

■ 1　クラス AddBrushTable



java.lang.Object

　|

　+--**field.map.AddBrushTable**

---

public class **AddBrushTable**

extends java.lang.Object

GeoSQL 表示用ブラシ設定ファイルを読み込んで、設定内容を保持するクラス

フィールドの概要	
private　jp.co.dawn_corp.geosql.j2.GeoSQLBrushTable	**brshTable** ブラシ設定テーブルクラス
private　jp.co.dawn_corp.geosql.j2.GeoSQLColorTable	**colorTable** カラー設定テーブルクラスのインスタンス
private　java.lang.String	**fileName** ブラシ設定ファイル名

コンストラクタの概要
AddBrushTable(java.lang.String f, jp.co.dawn_corp.geosql.j2.GeoSQLColorTable col) コンストラクタ

メソッドの概要	
private void	**fileReader**() ブラシ設定ファイルを読み込む。
jp.co.dawn_corp.geosql.j2.GeoSQLBrushTable	**getBrushTable**() ブラシ設定クラスを返す。

フィールドの詳細

brshTable

private jp.co.dawn_corp.geosql.j2.GeoSQLBrushTable **brshTable**

ブラシ設定テーブルクラス

---

fileName

private java.lang.String **fileName**

ブラシ設定ファイル名

---

colorTable

private jp.co.dawn_corp.geosql.j2.GeoSQLColorTable **colorTable**

カラー設定テーブルクラスのインスタンス

コンストラクタの詳細

AddBrushTable

public **AddBrushTable**(java.lang.String f,
jp.co.dawn_corp.geosql.j2.GeoSQLColorTable col)

コンストラクタ

パラメータ:

f - ブラシ設定ファイル名

メソッドの詳細

getBrushTable

public jp.co.dawn_corp.geosql.j2.GeoSQLBrushTable **getBrushTable**()

ブラシ設定クラスを返す。

戻り値:

ブラシ設定ファイルクラス

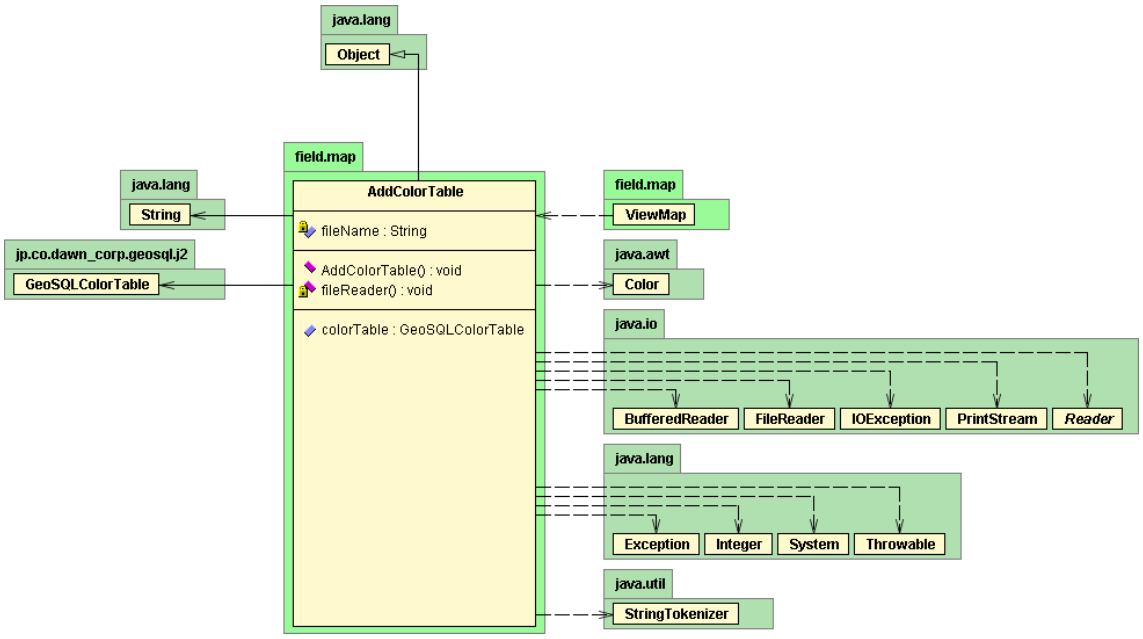
---

fileReader

private void **fileReader**()

ブラシ設定ファイルを読み込む。

■ 2　クラス AddColorTable



java.lang.Object

|

+--**field.map.AddColorTable**

---

public class **AddColorTable**

extends java.lang.Object

GeoSQL 表示用カラー設定ファイルを読み込んで、設定内容を保持するクラス

フィールドの概要	
private jp.co.dawn_corp.geosql.j2.GeoSQLColorTable	**colorTable** カラー設定テーブルクラス
private java.lang.String	**fileName** カラー設定ファイル名

コンストラクタの概要
AddColorTable(java.lang.String f) コンストラクタ

メソッドの概要	
private void	**fileReader**() カラー設定ファイルを読み込む。
jp.co.dawn_corp.geosql.j2.GeoSQLColorTable	**getColorTable**() カラー設定クラスを返す。

フィールドの詳細

colorTable

private jp.co.dawn_corp.geosql.j2.GeoSQLColorTable **colorTable**

カラー設定テーブルクラス

---

fileName

private java.lang.String **fileName**

カラー設定ファイル名

コンストラクタの詳細

AddColorTable

public **AddColorTable**(java.lang.String f)

コンストラクタ

パラメータ:

f - カラー設定ファイル名

メソッドの詳細

getColorTable

public jp.co.dawn_corp.geosql.j2.GeoSQLColorTable **getColorTable**()

カラー設定クラスを返す。

戻り値:

カラー設定ファイルクラス

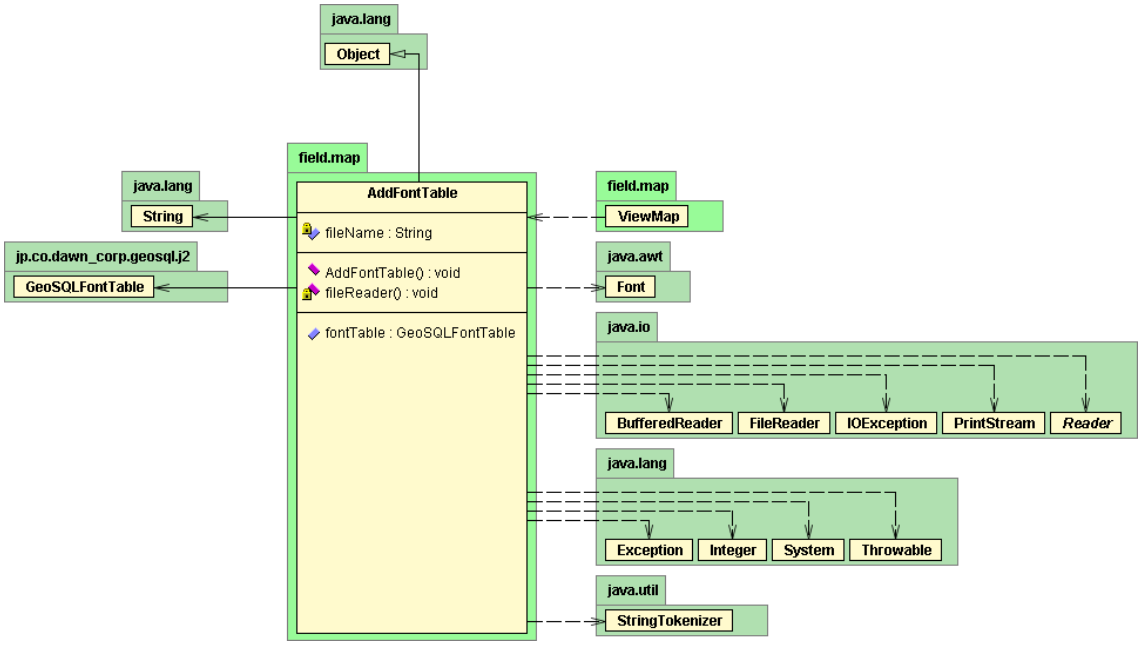
---

fileReader

private void **fileReader**()

カラー設定ファイルを読み込む。

■ 3　クラス AddFontTable



java.lang.Object

　|

　+--**field.map.AddFontTable**

---

public class **AddFontTable**

extends java.lang.Object

GeoSQL 表示用フォント設定ファイルを読み込んで、設定内容を保持するクラス

フィールドの概要	
private　java.lang.String	**fileName** 　　フォント設定ファイル名
private　jp.co.dawn_corp.geosql.j2.GeoSQLFontTable	**fontTable** 　　フォント設定テーブルクラス

コンストラクタの概要
AddFontTable(java.lang.String f) 　　コンストラクタ

メソッドの概要	
private void	**fileReader**() フォント設定ファイルを読み込む。
jp.co.dawn_corp.geosql.j2.GeoSQLFontTable	**getFontTable**() フォント設定クラスを返す。

フィールドの詳細

fontTable

private jp.co.dawn_corp.geosql.j2.GeoSQLFontTable **fontTable**

フォント設定テーブルクラス

---

fileName

private java.lang.String **fileName**

フォント設定ファイル名

コンストラクタの詳細

AddFontTable

public **AddFontTable**(java.lang.String f)

コンストラクタ

パラメータ:

f - フォント設定ファイル名

メソッドの詳細

getFontTable

public jp.co.dawn_corp.geosql.j2.GeoSQLFontTable **getFontTable**()

フォント設定クラスを返す。

戻り値:

フォント設定ファイルクラス

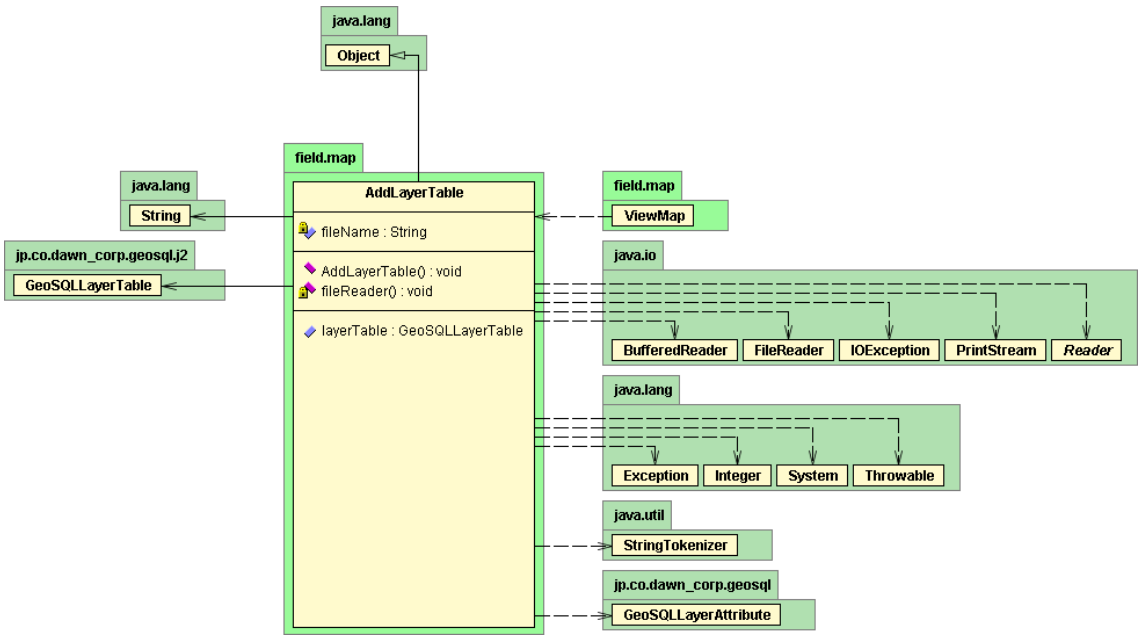
---

fileReader

private void **fileReader**()

フォント設定ファイルを読み込む。

■ 4　クラス AddLayerTable



java.lang.Object

　|

　+--**field.map.AddLayerTable**

---

public class **AddLayerTable**

extends java.lang.Object

GeoSQL 表示用レイヤー設定ファイルを読み込んで、設定内容を保持するクラス

フィールドの概要	
private　java.lang.String	**fileName** 　　レイヤー設定ファイル名
private　jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable	**layerTable** 　　レイヤー設定テーブルクラス

コンストラクタの概要
AddLayerTable(java.lang.String f) 　　コンストラクタ

メソッドの概要	
private void	**fileReader**() レイヤー設定ファイルを読み込む。
jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable	**getLayerTable**() レイヤー設定クラスを返す。

フィールドの詳細

layerTable

private jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable **layerTable**

レイヤー設定テーブルクラス

---

fileName

private java.lang.String **fileName**

レイヤー設定ファイル名

コンストラクタの詳細

AddLayerTable

public **AddLayerTable**(java.lang.String f)

コンストラクタ

パラメータ:

f - レイヤー設定ファイル名

メソッドの詳細

getLayerTable

public jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable **getLayerTable**()

レイヤー設定クラスを返す。

戻り値:

レイヤー設定ファイルクラス

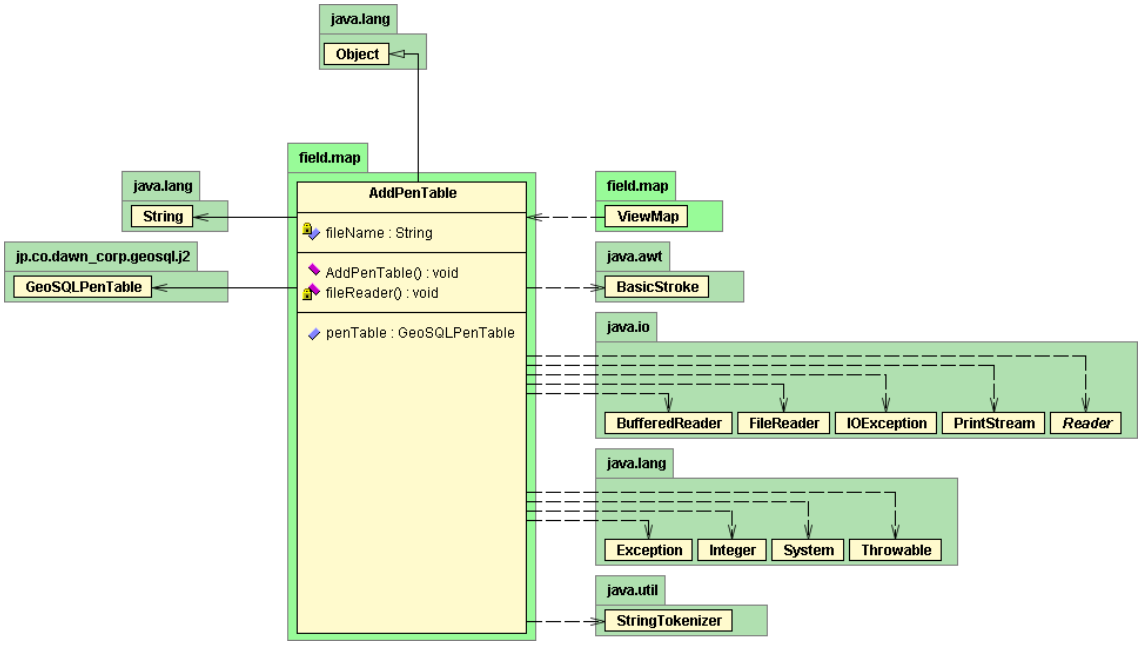
---

fileReader

private void **fileReader**()

レイヤー設定ファイルを読み込む。

■ ５　クラス AddPenTable



java.lang.Object

　|

　+--**field.map.AddPenTable**

---

public class **AddPenTable**

extends java.lang.Object

GeoSQL 表示用ペン設定ファイルを読み込んで、設定内容を保持するクラス

フィールドの概要	
private　java.lang.String	**fileName** 　　ペン設定ファイル名
private　jp.co.dawn_corp.geosql.j2.GeoSQLPenTable	**penTable** 　　ペン設定テーブルクラス

コンストラクタの概要
AddPenTable(java.lang.String f) 　　コンストラクタ

メソッドの概要	
private void	**fileReader**() ペン設定ファイルを読み込む。
jp.co.dawn_corp.geosql.j2.GeoSQLPenTable	**getPenTable**() ペン設定クラスを返す。

フィールドの詳細

penTable

private jp.co.dawn_corp.geosql.j2.GeoSQLPenTable **penTable**

ペン設定テーブルクラス

---

fileName

private java.lang.String **fileName**

ペン設定ファイル名

コンストラクタの詳細

AddPenTable

public **AddPenTable**(java.lang.String f)

コンストラクタ

パラメータ:

f - ペン設定ファイル名

メソッドの詳細

getPenTable

public jp.co.dawn_corp.geosql.j2.GeoSQLPenTable **getPenTable**()

ペン設定クラスを返す。

戻り値:

ペン設定ファイルクラス

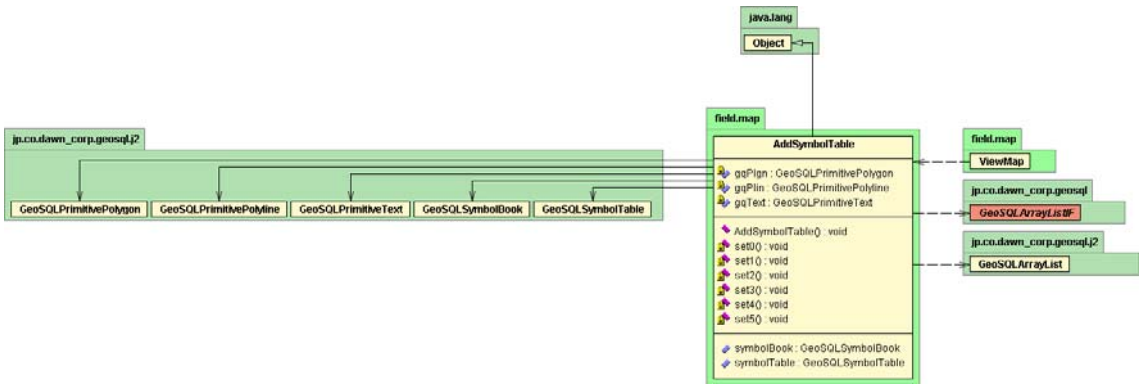
---

fileReader

private void **fileReader**()

ペン設定ファイルを読み込む。

■ ６　クラス AddSymbolTable



java.lang.Object

　|

　+--**field.map.AddSymbolTable**

---

public class **AddSymbolTable**

extends java.lang.Object

シンボルデータを作成するクラス（GeoSQL 用）

フィールドの概要	
private　jp.co.dawn_corp.geosql.j2.GeoSQLPrimitivePolygon	**gqPlgn** GeoSQLPrimitivePolygon
private　jp.co.dawn_corp.geosql.j2.GeoSQLPrimitivePolyline	**gqPlin** GeoSQLPrimitivePolyline
private　jp.co.dawn_corp.geosql.j2.GeoSQLPrimitiveText	**gqText** GeoSQLPrimitiveText
private　jp.co.dawn_corp.geosql.j2.GeoSQLSymbolBook	**symbolBook** GeoSQLSymbolBook
private　jp.co.dawn_corp.geosql.j2.GeoSQLSymbolTable	**symbolTable** GeoSQLSymbolTable

コンストラクタの概要

AddSymbolTable() 　　コンストラクタ

メソッドの概要	
jp.co.dawn_corp.geosql.j2.GeoSQLSymbolBook	**getSymbolBook**() 　　シンボルブック設定クラスを返す。
jp.co.dawn_corp.geosql.j2.GeoSQLSymbolTable	**getSymbolTable**() 　　シンボル設定クラスを返す。
private void	**set0**(int id)
private void	**set1**(int id)
private void	**set2**(int id)
private void	**set3**(int id)
private void	**set4**(int id)
private void	**set5**(int id)

フィールドの詳細

symbolBook

private jp.co.dawn_corp.geosql.j2.GeoSQLSymbolBook **symbolBook**

GeoSQLSymbolBook

---

symbolTable

private jp.co.dawn_corp.geosql.j2.GeoSQLSymbolTable **symbolTable**

GeoSQLSymbolTable

---

gqPlin

private jp.co.dawn_corp.geosql.j2.GeoSQLPrimitivePolyline **gqPlin**

GeoSQLPrimitivePolyline

---

gqText

private jp.co.dawn_corp.geosql.j2.GeoSQLPrimitiveText **gqText**

GeoSQLPrimitiveText

---

gqPlgn

private jp.co.dawn_corp.geosql.j2.GeoSQLPrimitivePolygon **gqPlgn**

GeoSQLPrimitivePolygon

コンストラクタの詳細

AddSymbolTable

public **AddSymbolTable**()

コンストラクタ

メソッドの詳細

getSymbolBook

public jp.co.dawn_corp.geosql.j2.GeoSQLSymbolBook **getSymbolBook**()

シンボルブック設定クラスを返す。

戻り値:

シンボルブック設定ファイルクラス

---

getSymbolTable

public jp.co.dawn_corp.geosql.j2.GeoSQLSymbolTable **getSymbolTable**()

シンボル設定クラスを返す。

戻り値:

シンボル設定ファイルクラス

---

set0

private void **set0**(int id)

---

set1

private void **set1**(int id)

---

set2

private void **set2**(int id)

---

set3

private void **set3**(int id)

---

set4

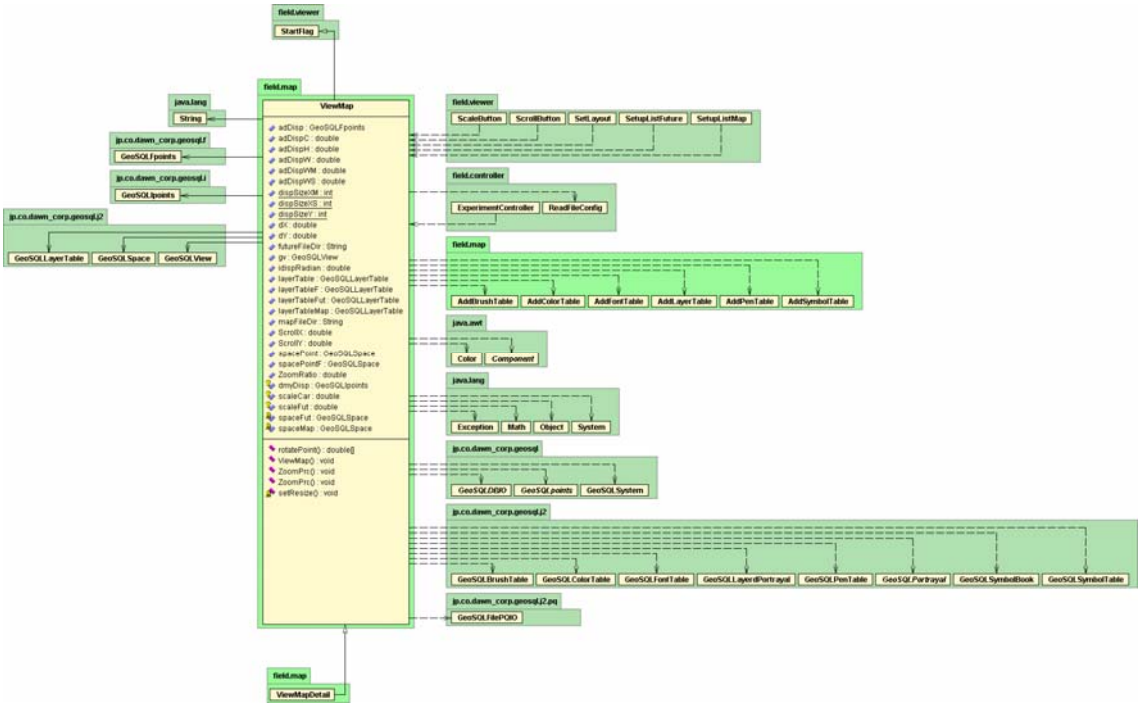
private void **set4**(int id)

---

set5

private void **set5**(int id)

■ 7　クラス ViewMap



java.lang.Object

|

+--java.awt.Component

|

+--java.awt.Container

|

+--javax.swing.JComponent

|

+--javax.swing.JPanel

|

+--field.viewer.StartFlag

|

+--**field.map.ViewMap**

直系の既知のサブクラス:

ViewMapDetail

---

public class **ViewMap**

extends StartFlag

GeoSQL を使用して、背景地図、道路地物表示を表示するクラス

フィールドの概要	
jp.co.dawn_corp.geosql.f.GeoSQLFpoints	**adDisp** 初期表示領域（標準）
double	**adDispC** 表示中心変位
double	**adDispH** 初期表示高さ
double	**adDispW** 初期表示幅
double	**adDispWM** 最大表示幅
double	**adDispWS** 標準表示幅
static int	**dispSizeXM** X 方向最大表示サイズ
static int	**dispSizeXS** X 方向標準表示サイズ
static int	**dispSizeY** Y 方向表示サイズ
protected jp.co.dawn_corp.geosql.i.GeoSQLIpoints	**dmyDisp** 画面表示切替時の表示領域を格納する
double	**dX** 表示中点 X
double	**dY** 表示中点 Y
java.lang.String	**futureFileDir** 地物ファイルディレクトリ
jp.co.dawn_corp.geosql.j2.GeoSQLView	**gv** ビュークラス

double	**idispRadian** 表示角度
jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable	**layerTable** 表示レイヤーテーブルクラス
jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable	**layerTableF** 表示レイヤーテーブルクラス（自車位置用）
jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable	**layerTableFut** レイヤー設定情報
jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable	**layerTableMap** レイヤー設定情報
java.lang.String	**mapFileDir** 地図ファイルディレクトリ
protected double	**scaleCar** 自車位置マークのスケール
protected double	**scaleFut** 検索地物マークのスケール
double	**ScrollX** スクロール値（X）
double	**ScrollY** スクロール値（Y）
private jp.co.dawn_corp.geosql.j2.GeoSQLSpace	**spaceFut** 道路地物スペース管理クラス
private jp.co.dawn_corp.geosql.j2.GeoSQLSpace	**spaceMap** 背景地図スペース管理クラス
jp.co.dawn_corp.geosql.j2.GeoSQLSpace	**spacePoint** 現在位置スペース管理クラス
jp.co.dawn_corp.geosql.j2.GeoSQLSpace	**spacePointF** 現在位置スペース管理クラス（自車位置用）
double	**ZoomRatio** 拡大・縮小率

クラス field.viewer.StartFlag から継承したフィールド
startflag

コンストラクタの概要
ViewMap(ReadFileConfig config) コンストラクタ

メソッドの概要	
double[]	**rotatePoint**(double pX, double pY, double dRadian) 回転時の中心からの変位を計算する（パラメータ：double 型)。
private void	**setResize**(ReadFileConfig config) 画面表示領域の調整および標準時・最大時の領域設定
void	**ZoomPrc**() 地図画面再表示 地図を回転、拡大、スクロールして表示する。
void	**ZoomPrc**(double x, double y, double r) 地図画面再表示 ボタンが押された時に画面の表示を切換える時に呼び出す。

フィールドの詳細

mapFileDir

public java.lang.String **mapFileDir**

地図ファイルディレクトリ

---

futureFileDir

public java.lang.String **futureFileDir**

地物ファイルディレクトリ

---

layerTableMap

public jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable **layerTableMap**
レイヤー設定情報

---

layerTableFut

public jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable **layerTableFut**
レイヤー設定情報

---

gv

public jp.co.dawn_corp.geosql.j2.GeoSQLView **gv**
ビュークラス

---

dX

public double **dX**
表示中点 X

---

dY

public double **dY**
表示中点 Y

---

idispRadian

public double **idispRadian**
表示角度

ZoomRatio

public double **ZoomRatio**

拡大・縮小率

ScrollX

public double **ScrollX**

スクロール値（X）

ScrollY

public double **ScrollY**

スクロール値（Y）

---

spacePoint

public jp.co.dawn_corp.geosql.j2.GeoSQLSpace **spacePoint**

現在位置スペース管理クラス

---

spacePointF

public jp.co.dawn_corp.geosql.j2.GeoSQLSpace **spacePointF**

現在位置スペース管理クラス（自車位置用）

---

layerTable

public jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable **layerTable**

表示レイヤーテーブルクラス

---

layerTableF

public jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable **layerTableF**

表示レイヤーテーブルクラス（自車位置用）

---

adDispW

public double **adDispW**

初期表示幅

---

adDispWS

public double **adDispWS**

標準表示幅

---

adDispWM

public double **adDispWM**

最大表示幅

---

adDispH

public double **adDispH**

初期表示高さ

---

adDispC

public double **adDispC**

表示中心変位

---

spaceMap

private jp.co.dawn_corp.geosql.j2.GeoSQLSpace **spaceMap**

背景地図スペース管理クラス

---

spaceFut

private jp.co.dawn_corp.geosql.j2.GeoSQLSpace **spaceFut**

道路地物スペース管理クラス

adDisp

public jp.co.dawn_corp.geosql.f.GeoSQLFpoints **adDisp**

初期表示領域（標準）

---

dispSizeXS

public static final int **dispSizeXS**

X 方向標準表示サイズ

---

dispSizeXM

public static final int **dispSizeXM**

X 方向最大表示サイズ

---

dispSizeY

public static final int **dispSizeY**

Y 方向表示サイズ

---

scaleCar

protected double **scaleCar**

自車位置マークのスケール

---

scaleFut

protected double **scaleFut**

検索地物マークのスケール

---

dmyDisp

protected jp.co.dawn_corp.geosql.i.GeoSQLIpoints **dmyDisp**

画面表示切替時の表示領域を格納する

コンストラクタの詳細

ViewMap

public **ViewMap**(ReadFileConfig config)

コンストラクタ

パラメータ:

config - 設定ファイルファイル読込クラス

メソッドの詳細

setResize

private void **setResize**(ReadFileConfig config)

画面表示領域の調整および標準時・最大時の領域設定

パラメータ:

config - 設定ファイル読込クラス

ZoomPrc

public void **ZoomPrc**(double x,
double y,
double r)

地図画面再表示

ボタンが押された時に画面の表示を切換える時に呼び出す。

パラメータ:

x - 表示中心座標 X（実地図座標）

y - 表示中心座標 Y（実地図座標）

r - 回転角（北がゼロ、時計回り、ラジアン）

ZoomPrc

public void **ZoomPrc**()

地図画面再表示

地図を回転、拡大、スクロールして表示する。

---

rotatePoint

public double[] **rotatePoint**(double pX,
double pY,
double dRadian)

回転時の中心からの変位を計算する（パラメータ：double 型)。

パラメータ:

pX - X 方向変位量

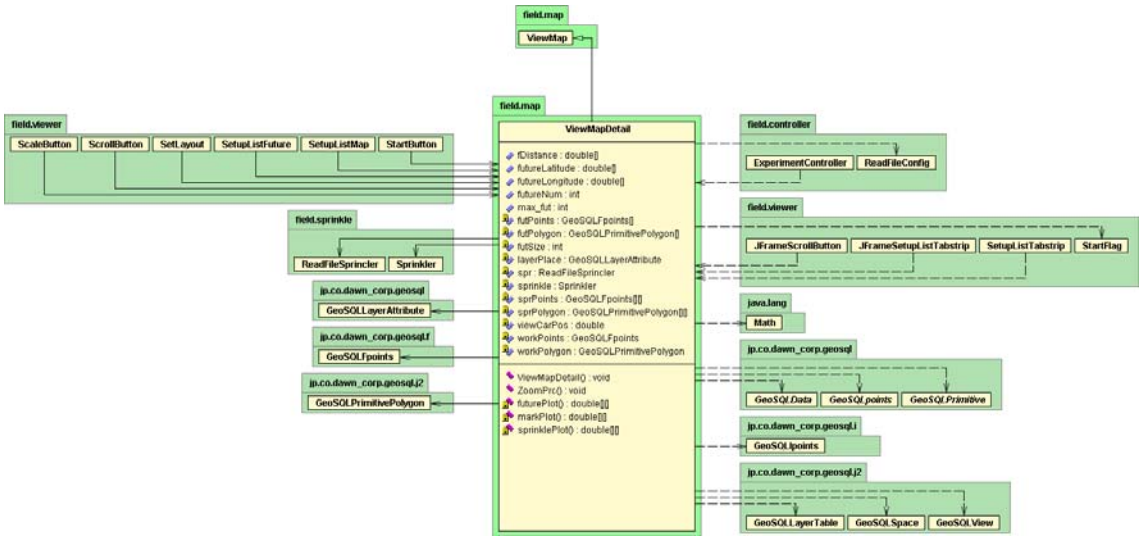
pY - Y 方向変位量

dRadian - 回転角

戻り値:

X 方向、Y 方向の変位

■ ８　クラス ViewMapDetail




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.StartFlag
                                |
                                +--field.map.ViewMap
                                      |
                                      +--field.map.ViewMapDetail
```

---

public class **ViewMapDetail**

extends ViewMap

地図上に現在位置と接近地物を表示するクラス

フィールドの概要

double[]	**fDistance** 地物までの距離
private jp.co.dawn_corp.geosql.f.GeoSQLFpoints[]	**futPoints** 検索地物表示ポリゴンデータ
private jp.co.dawn_corp.geosql.j2.GeoSQLPrimitivePolygon[]	**futPolygon** 検索地物表示ポリゴン
private int	**futSize** 表示する地物の大きさ
double[]	**futureLatitude** 地物の緯度
double[]	**futureLongitude** 地物の経度
int	**futureNum** 表示地物数
private jp.co.dawn_corp.geosql.GeoSQLLayerAttribute	**layerPlace** 現在位置表示属性
int	**max_fut** 表示検索地物の最大数
private ReadFileSprincler	**spr** Sprinkler クラスのインスタンス
private Sprinkler	**sprinkle** 散布制御クラス
private jp.co.dawn_corp.geosql.f.GeoSQLFpoints[][]	**sprPoints** 散布箇所表示ポリゴンデータ
private jp.co.dawn_corp.geosql.j2.GeoSQLPrimitivePolygon[][]	**sprPolygon** 散布箇所表示ポリゴン
private double	**viewCarPos** 自車表示位置補正
private jp.co.dawn_corp.geosql.f.GeoSQLFpoints	**workPoints** 現在位置表示ポリゴンデータ
private jp.co.dawn_corp.	**workPolygon**

geosql.j2.GeoSQLPrimitivePolygon	現在位置表示ポリゴン

クラス field.map.ViewMap から継承したフィールド
adDisp, adDispC, adDispH, adDispW, adDispWM, adDispWS, dispSizeXM, dispSizeXS, dispSizeY, dmyDisp, dX, dY, futureFileDir, gv, idispRadian, layerTable, layerTableF, layerTableFut, layerTableMap, mapFileDir, scaleCar, scaleFut, ScrollX, ScrollY, spaceFut, spaceMap, spacePoint, spacePointF, ZoomRatio

クラス field.viewer.StartFlag から継承したフィールド
startflag

コンストラクタの概要
ViewMapDetail(ReadFileConfig config, Sprinkler sprinkler) コンストラクタ

メソッドの概要	
private double[][]	**futurePlot**(double cx, double cy, double dRadian, double fd) 接近地物のマークを表示する
private double[][]	**markPlot**(double cx, double cy, double dRadian) 現在位置のマークを表示する
private double[][]	**sprinklePlot**(double x0, double y0, double x1, double y1, double x2, double y2, double x3, double y3) 散布箇所を表示する
void	**ZoomPrc**() 地図画面再表示

	地図を回転、拡大、スクロールして表示する。

クラス field.map.ViewMap から継承したメソッド
rotatePoint, setResize, ZoomPrc

フィールドの詳細

max_fut

public int max_fut

表示検索地物の最大数

---

futureLongitude

public double[] futureLongitude

地物の経度

---

futureLatitude

public double[] futureLatitude

地物の緯度

---

fDistance

public double[] fDistance

地物までの距離

---

futureNum

public int futureNum

表示地物数

layerPlace

private jp.co.dawn_corp.geosql.GeoSQLLayerAttribute **layerPlace**

現在位置表示属性

---

workPoints

private jp.co.dawn_corp.geosql.f.GeoSQLFpoints **workPoints**

現在位置表示ポリゴンデータ

---

workPolygon

private jp.co.dawn_corp.geosql.j2.GeoSQLPrimitivePolygon **workPolygon**

現在位置表示ポリゴン

---

futPoints

private jp.co.dawn_corp.geosql.f.GeoSQLFpoints[] **futPoints**

検索地物表示ポリゴンデータ

---

futPolygon

private jp.co.dawn_corp.geosql.j2.GeoSQLPrimitivePolygon[] **futPolygon**

検索地物表示ポリゴン

---

futSize

private int **futSize**

表示する地物の大きさ

---

spr

private ReadFileSprincler **spr**

Sprinkler クラスのインスタンス

---

sprPoints

private jp.co.dawn_corp.geosql.f.GeoSQLFpoints[][] **sprPoints**

散布箇所表示ポリゴンデータ

---

sprPolygon

private jp.co.dawn_corp.geosql.j2.GeoSQLPrimitivePolygon[][] **sprPolygon**

散布箇所表示ポリゴン

---

sprinkle

private Sprinkler **sprinkle**

散布制御クラス

---

viewCarPos

private double **viewCarPos**

自車表示位置補正

コンストラクタの詳細

ViewMapDetail

public **ViewMapDetail**(ReadFileConfig config,
Sprinkler sprinkler)

コンストラクタ

パラメータ:

config - 設定ファイル情報クラス

sprinkler - 散布車制御クラス

メソッドの詳細

ZoomPrc

public void **ZoomPrc**()

地図画面再表示

地図を回転、拡大、スクロールして表示する。

オーバーライド:

クラス ViewMap 内の ZoomPrc

---

markPlot

private double[][] **markPlot**(double cx,
double cy,
double dRadian)

現在位置のマークを表示する

パラメータ:

cx - 現在位置の経度（平面直角座標）

cy - 現在位置の緯度（平面直角座標）

dRadian - 現在の進行方向角

戻り値:

現在位置のマークを構成する点の座標

---

futurePlot

private double[][] **futurePlot**(double cx,
double cy,
double dRadian,
double fd)

接近地物のマークを表示する

パラメータ:

cx - 接近地物の経度（平面直角座標）

cy - 接近地物の緯度（平面直角座標）

dRadian - 現在の進行方向角

戻り値:

接近地物のマークを構成する点の座標

---

sprinklePlot

```
private double[][] sprinklePlot(double x0,
                                double y0,
                                double x1,
                                double y1,
                                double x2,
                                double y2,
                                double x3,
                                double y3)
```

散布箇所を表示する

パラメータ:

cx - 接近地物の経度（平面直角座標）

cy - 接近地物の緯度（平面直角座標）

dRadian - 現在の進行方向角

戻り値:

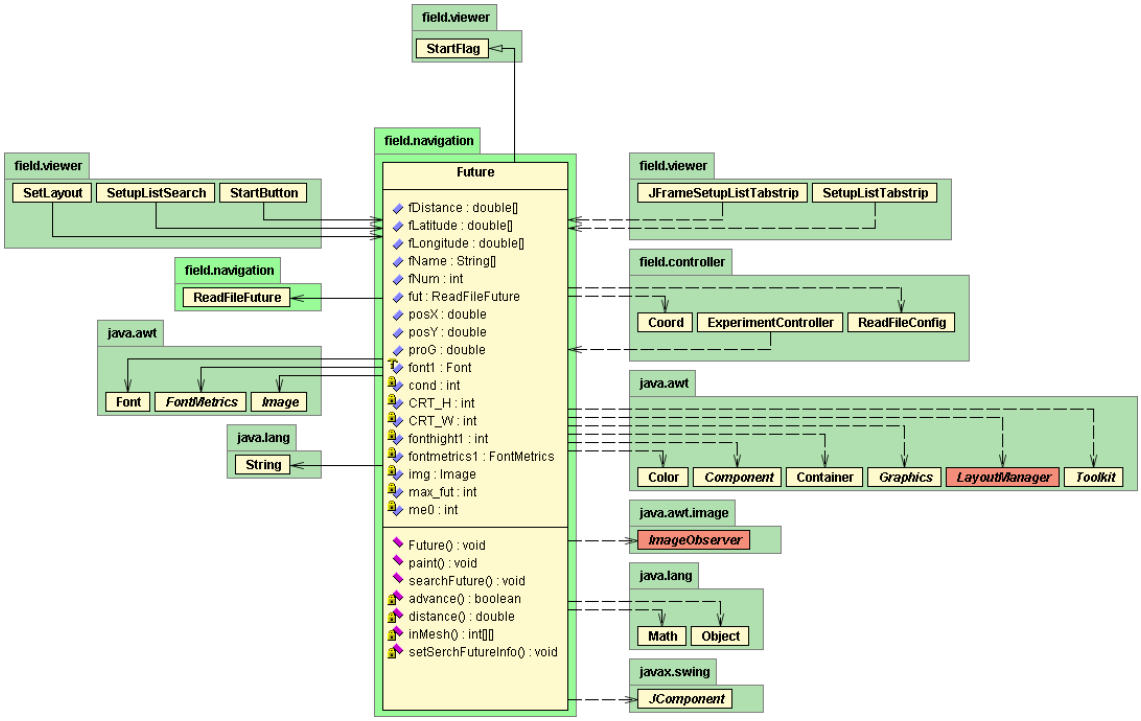
散布箇所を構成する点の座標

---

d） パッケージ field.navigation

クラスの概要	
Future	道路地物接近を判定するクラス
ReadFileFuture	地物検索設定ファイルの読込クラス

■ 1　クラス Future




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.StartFlag
                                |
                                +--field.navigation.Future
```

---

public class **Future**

extends StartFlag

道路地物接近を判定するクラス

フィールドの概要	
private int	**cond** 検索地物無、検索地物有、走行ライン外、逆走メッセージ フラグ 0:検索地物無 1:検索地物有 2:走行ライン外 3:逆走メッセージ
private int	**CRT_H** 画面高さ
private int	**CRT_W** 画面幅
double[]	**fDistance** 地物までの距離
double[]	**fLatitude** 地物緯度
double[]	**fLongitude** 地物経度
java.lang.String[]	**fName** 地物名称
int	**fNum** 表示検索地物数
(パッケージプライベート) java.awt.Font	**font1** フォントの設定（サイズ：26）
private int	**fonthight1** フォントの高さ（サイズ：26）
private java.awt.FontMetrics	**fontmetrics1** フォントの属性（サイズ：26）
ReadFileFuture	**fut** 地物探査ファイル情報クラス
private java.awt.Image	**img** パネル画像イメージ
private int	**max_fut** 表示検索地物の最大数

private int	**me0** メッシュ探査開始メッシュ番号
double	**posX** 現在位置（緯度）
double	**posY** 現在位置（経度）
double	**proG** 現在位置（進行方向角）

クラス field.viewer.StartFlag から継承したフィールド
startflag

コンストラクタの概要
Future(ReadFileConfig configure) コンストラクタ

メソッドの概要	
private boolean	**advance**(int j, int i) 進んでいる方向が、基準点距離が増加方向に進んでいるか減少方向に進んでいるかを判定する。
private double	**distance**(int j, int i, double xx2, double yy2) 探査道路地物までの距離を計算する。
private int[][]	**inMesh**() 現在位置がメッシュ内にあるか判定する。
void	**paint**(java.awt.Graphics g) 注意を喚起する文字を表示する

void	**searchFuture**() 道路地物の探査を行う。
private void	**setSerchFutureInfo**(int n, int j, int i, double dis) 検索地物情報のセット

フィールドの詳細

max_fut

private int **max_fut**

表示検索地物の最大数

---

fDistance

public double[] **fDistance**

地物までの距離

---

fName

public java.lang.String[] **fName**

地物名称

---

fLongitude

public double[] **fLongitude**

地物経度

---

fLatitude

public double[] **fLatitude**

地物緯度

---

fNum

public int **fNum**

表示検索地物数

---

posX

public double **posX**

現在位置（緯度）

---

posY

public double **posY**

現在位置（経度）

---

proG

public double **proG**

現在位置（進行方向角）

CRT_W

private int **CRT_W**

画面幅

CRT_H

private int **CRT_H**

画面高さ

---

fut

public ReadFileFuture **fut**

地物探査ファイル情報クラス

---

me0

private int me0

メッシュ探査開始メッシュ番号

---

cond

private int cond

検索地物無、検索地物有、走行ライン外、逆走メッセージ　フラグ 0:検索地物無 1:検索地物有 2:走行ライン外 3:逆走メッセージ

---

font1

java.awt.Font font1

フォントの設定（サイズ：26）

---

fontmetrics1

private java.awt.FontMetrics fontmetrics1

フォントの属性（サイズ：26）

---

fonthight1

private int fonthight1

フォントの高さ（サイズ：26）

---

img

private java.awt.Image img

パネル画像イメージ

コンストラクタの詳細

Future

public **Future**(ReadFileConfig configure)

コンストラクタ

パラメータ:

fileName - 道路地物探査用ファイル名

メソッドの詳細

paint

public void **paint**(java.awt.Graphics g)

注意を喚起する文字を表示する

オーバーライド:

クラス javax.swing.JComponent 内の paint

パラメータ:

g - java.awt.Graphics

---

searchFuture

public void **searchFuture**()

道路地物の探査を行う。

---

setSerchFutureInfo

private void **setSerchFutureInfo**(int n,
int j,
int i,
double dis)

検索地物情報のセット

パラメータ:

n - 車線番号

j - 車線配列番号

i - 検索地物配列番号

dis - 現在位置とメッシュベース線との距離

---

distance

```
private double distance(int j,
                        int i,
                        double xx2,
                        double yy2)
```

探査道路地物までの距離を計算する。

パラメータ:

j - 車線方向配列番号

i - 進行方向配列番号

xx2 - 現在位置の X 座標

yy2 - 現在位置の Y 座標

戻り値:

最近探査道路地物までの距離

---

advance

```
private boolean advance(int j,
                        int i)
```

進んでいる方向が、基準点距離が増加方向に進んでいるか減少方向に進んでいるかを判定する。

パラメータ:

j - 車線方向配列番号

i - 進行方向配列番号

戻り値:

true:増加方向、false:減少方向

---

inMesh

```
private int[][] inMesh()
```

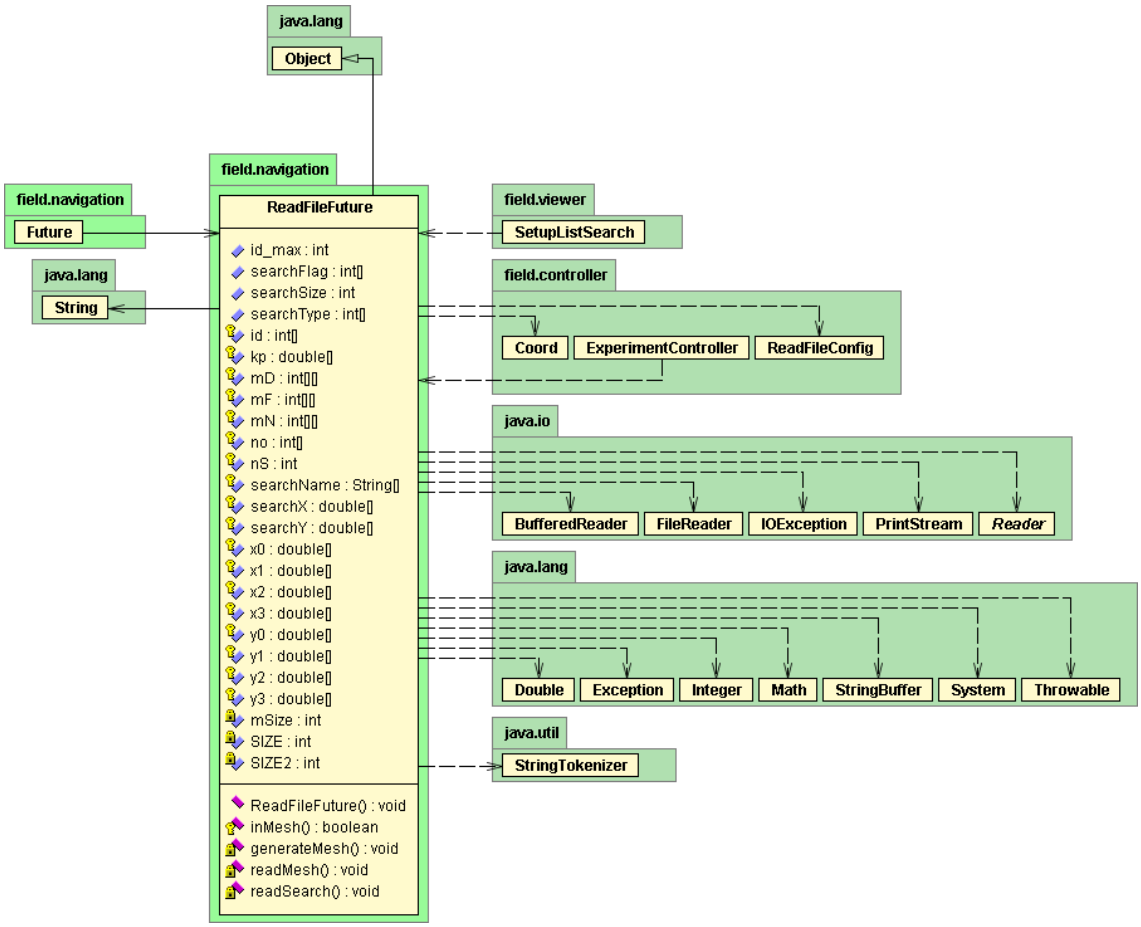
現在位置がメッシュ内にあるか判定する。

（高速化のため、１回前の位置メッシュからサーチ開始）

戻り値:

現在位置のメッシュ番号 [][0]:車線 [][1]:メッシュ番号

■ 2　クラス ReadFileFuture



java.lang.Object

　|

　+--**field.navigation.ReadFileFuture**

---

public class **ReadFileFuture**

extends java.lang.Object

地物検索設定ファイルの読込クラス

フィールドの概要	
protected　int[]	**id** 車線 ID　下り　追越車線:-1　走行車線:-2　副道 1:-3　副道 1:-4　副道 1:-5　上り　追越車線: 1　走行車線: 2　副道 1: 3　副道 1: 4　副道 1: 5

int	**id_max** 車線 ID の最大値
protected double[]	**kp** KP
protected int[][]	**mD** メッシュ格納配列番号
protected int[][]	**mF** メッシュ検索地物の数
protected int[][]	**mN** メッシュの存在の有無 0:無 1:有
private int	**mSize** メッシュデータ数
protected int[]	**no** メッシュ番号
protected int	**nS** サーチメッシュの数
int[]	**searchFlag** 検索地物情報（検索判定）
protected java.lang.String[]	**searchName** 検索地物情報（名称）
int	**searchSize** 検索地物情報（データ数）
int[]	**searchType** 検索地物情報（地物タイプ）
protected double[]	**searchX** 検索地物情報（緯度）
protected double[]	**searchY** 検索地物情報（経度）
private static int	**SIZE** 検索メッシュの最大点数（処理高速化のため、配列を使用）

private static int	**SIZE2** 入力データの最大点数（処理高速化のため、配列を使用）
protected double[]	**x0** メッシュ４点座標（左下 X）
protected double[]	**x1** メッシュ４点座標（左上 X）
protected double[]	**x2** メッシュ４点座標（右上 X）
protected double[]	**x3** メッシュ４点座標（右下 X）
protected double[]	**y0** メッシュ４点座標（左下 Y）
protected double[]	**y1** メッシュ４点座標（左上 Y）
protected double[]	**y2** メッシュ４点座標（右上 Y）
protected double[]	**y3** メッシュ４点座標（右下 Y）

コンストラクタの概要
ReadFileFuture(ReadFileConfig configure) コンストラクタ

メソッドの概要	
private void	**generateMesh**() 検索用メッシュデータの生成
protected boolean	**inMesh**(int i, double x, double y)

	与えられた点がメッシュ内にあるか判定
private void	**readMesh**(java.lang.String fileName) 範囲判定用メッシュファイルの読込
private void	**readSearch**(int i, java.lang.String fileName, int flag) 地物検索ファイルの読込

フィールドの詳細

SIZE

private static final int **SIZE**

検索メッシュの最大点数（処理高速化のため、配列を使用）

SIZE2

private static final int **SIZE2**

入力データの最大点数（処理高速化のため、配列を使用）

---

searchSize

public int **searchSize**

検索地物情報（データ数）

---

searchX

protected double[] **searchX**

検索地物情報（緯度）

---

searchY

protected double[] **searchY**

検索地物情報（経度）

searchName

protected java.lang.String[] **searchName**

検索地物情報（名称）

---

searchType

public int[] **searchType**

検索地物情報（地物タイプ）

---

searchFlag

public int[] **searchFlag**

検索地物情報（検索判定）

---

mSize

private int **mSize**

メッシュデータ数

---

no

protected int[] **no**

メッシュ番号

---

id

protected int[] **id**

車線 ID 下り　追越車線:-1　走行車線:-2　副道 1:-3　副道 1:-4　副道 1:-5 上り　追越車線: 1 走行車線: 2　副道 1: 3　副道 1: 4　副道 1: 5

---

id_max

public int **id_max**

車線 ID の最大値

---

kp

protected double[] **kp**

KP

---

x0

protected double[] **x0**

メッシュ４点座標（左下 X）

---

y0

protected double[] **y0**

メッシュ４点座標（左下 Y）

---

x1

protected double[] **x1**

メッシュ４点座標（左上 X）

---

y1

protected double[] **y1**

メッシュ４点座標（左上 Y）

---

x2

protected double[] **x2**

メッシュ４点座標（右上 X）

---

y2

protected double[] y2

メッシュ４点座標（右上 Y）

---

x3

protected double[] x3

メッシュ４点座標（右下 X）

---

y3

protected double[] **y3**

メッシュ４点座標（右下 Y）

---

nS

protected int nS

サーチメッシュの数

---

mN

protected int[][] mN

メッシュの存在の有無 0:無 1:有

---

mD

protected int[][] mD

メッシュ格納配列番号

---

mF

protected int[][] mF

メッシュ検索地物の数

コンストラクタの詳細

ReadFileFuture

public **ReadFileFuture**(ReadFileConfig configure)
コンストラクタ
パラメータ:
configure - 設定ファイル読込クラス

メソッドの詳細

readSearch

private void **readSearch**(int i,
java.lang.String fileName,
int flag)
地物検索ファイルの読込
パラメータ:
i - ファイルカウンタ
fileName - ファイル名

---

readMesh

private void **readMesh**(java.lang.String fileName)
範囲判定用メッシュファイルの読込
パラメータ:
fileName - 範囲判定用メッシュファイル名

---

generateMesh

private void **generateMesh**()
検索用メッシュデータの生成

---

inMesh

```
protected boolean inMesh(int i,
                         double x,
                         double y)
```

与えられた点がメッシュ内にあるか判定

パラメータ:

i - メッシュの配列番号

x - 与えられた点の X 座標

y - 与えられた点の Y 座標

戻り値:

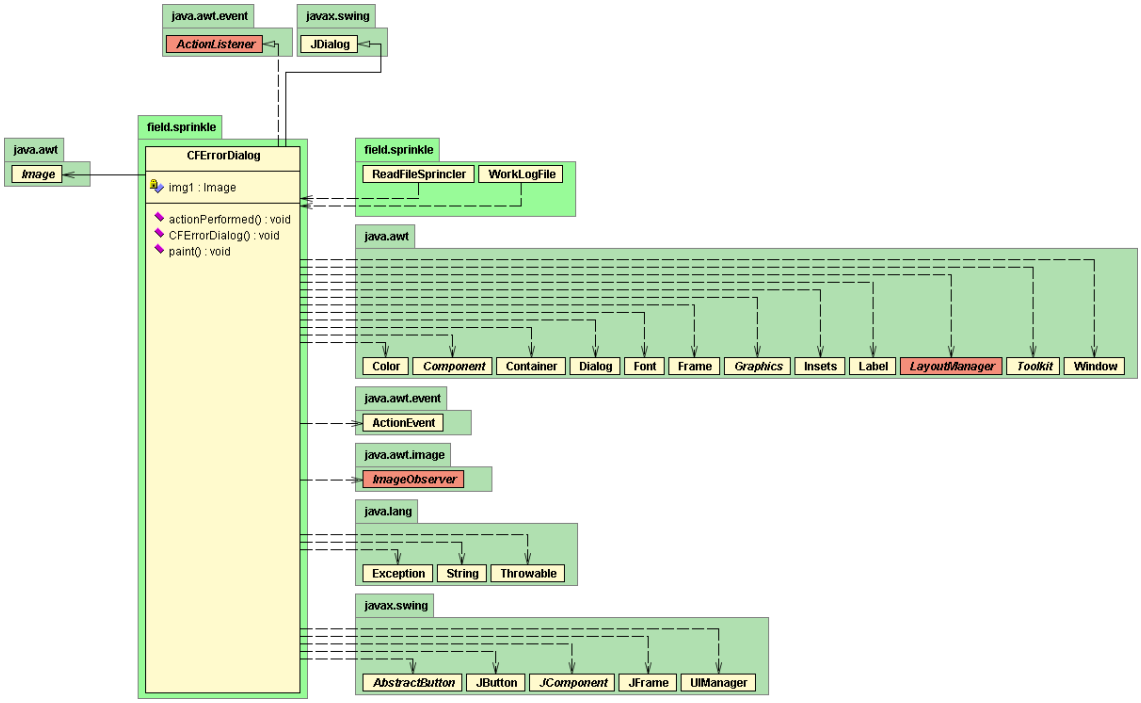
メッシュ内:true メッシュ外:false

---

e） パッケージ field.sprinkle

クラスの概要	
CFErrorDialog	外部記憶装置ファイル確認画面クラス
ReadFileSprincler	散布制御ファイルの読込クラス
SprinkleController	散布車制御データを RS-232C を通して制御するクラス
SprinkleFileDialog	散布ファイル選択画面クラス
Sprinkler	散布車を制御するクラス
StartCFDialog	外部記憶装置ファイル確認画面クラス
TimeKP	時間、KP をファイルに出力するクラス
WorkLogFile	作業記録をファイルに出力するクラス

■ 1　クラス CFErrorDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.sprinkle.CFErrorDialog
```

---

public class **CFErrorDialog**

extends javax.swing.JDialog

implements java.awt.event.ActionListener

外部記憶装置ファイル確認画面クラス

フィールドの概要	
private java.awt.Image	**img1** イメージアイコン

コンストラクタの概要
CFErrorDialog() コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント
void	**paint**(java.awt.Graphics g) ペイント処理

フィールドの詳細

img1

private java.awt.Image **img1**

イメージアイコン

コンストラクタの詳細

CFErrorDialog

public **CFErrorDialog**()

コンストラクタ

メソッドの詳細

paint

public void **paint**(java.awt.Graphics g)

ペイント処理

オーバーライド:

クラス java.awt.Container 内の paint

パラメータ:

g - Graphics

---

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

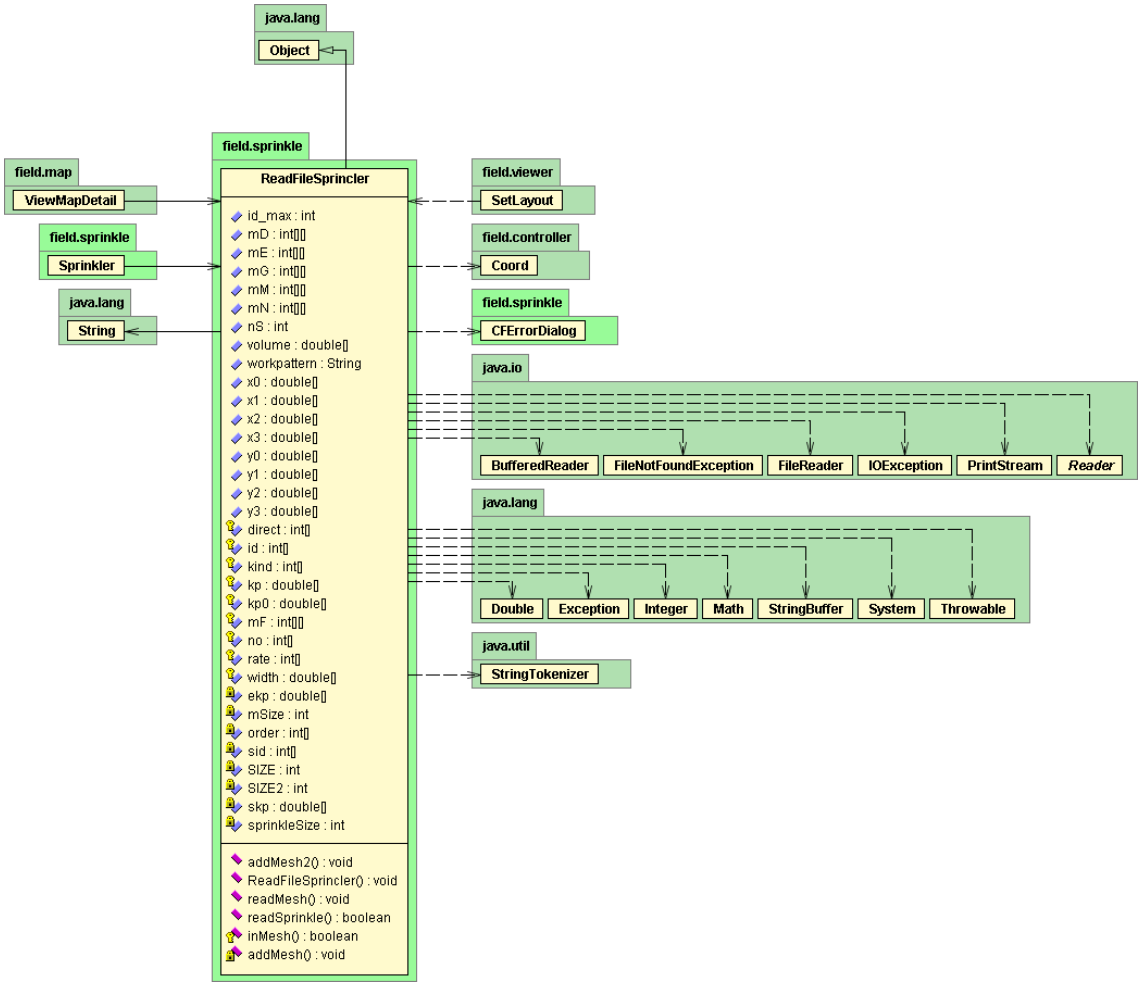
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

■ ２　クラス ReadFileSprincler



java.lang.Object

  |

  +--**field.sprinkle.ReadFileSprincler**

---

public class ReadFileSprincler

extends java.lang.Object

散布制御ファイルの読込クラス

フィールドの概要	
protected　int[]	**direct** 散布車制御（散布方向）
private　double[]	**ekp**

	散布車制御（終了 KP）
protected int[]	**id** 車線 ID 下り 追越車線:-1 走行車線:-2 副道:-3 以下（車線毎に順番に付番） 上り 追越車線: 1 走行車線: 2 副道: 3 以上（車線毎に順番に付番）
int	**id_max** 車線 ID の最大値
protected int[]	**kind** 散布車制御（薬剤種類）
protected double[]	**kp** KP(作業用)
protected double[]	**kp0** KP
int[][]	**mD** メッシュ格納配列番号
int[][]	**mE** 散布済フラグ 0:散布済またはメッシュなし 1 以上:未散布（散布順番号）
protected int[][]	**mF** 散布情報変更フラグ 1:散布開始 0:変更無 -1:散布終了
int[][]	**mG** 散布回数
int[][]	**mM** 散布車制御情報格納配列番号
int[][]	**mN** メッシュの存在の有無 0:無 1:有
private int	**mSize** メッシュデータ数
protected int[]	**no** メッシュ番号
int	**nS**

	サーチメッシュの数
private int[]	**order** 走行順　1から順に設定
protected int[]	**rate** 散布車制御（湿式割合）
private int[]	**sid** 散布車制御（車線 ID）
private static int	**SIZE** 検索メッシュの最大点数（処理高速化のため、配列を使用）
private static int	**SIZE2** 入力データの最大点数（処理高速化のため、配列を使用）
private double[]	**skp** 散布車制御（開始 KP）
private int	**sprinkleSize** 散布車制御（データ数）
double[]	**volume** 散布車制御（散布量）
protected double[]	**width** 散布車制御（散布幅）
java.lang.String	**workpattern** 標準作業パターン格納文字列
double[]	**x0** メッシュ４点座標（左下 X）
double[]	**x1** メッシュ４点座標（左上 X）
double[]	**x2** メッシュ４点座標（右上 X）
double[]	**x3** メッシュ４点座標（右下 X）
double[]	**y0** メッシュ４点座標（左下 Y）

double[]	**y1** メッシュ４点座標（左上 Y）
double[]	**y2** メッシュ４点座標（右上 Y）
double[]	**y3** メッシュ４点座標（右下 Y）

コンストラクタの概要
ReadFileSprincler(java.lang.String meshfile, java.lang.String sprinklefile) コンストラクタ

メソッドの概要	
private void	**addMesh**(java.lang.String fileName) 散布制御用メッシュデータの生成
void	**addMesh2**() 散布制御用メッシュデータの再生成
protected boolean	**inMesh**(int i, double x, double y) 与えられた点がメッシュ内にあるか判定
void	**readMesh**(java.lang.String fileName) 範囲判定用メッシュファイルの読込
boolean	**readSprinkle**(java.lang.String fileName) 散布計画ファイルの読込

フィールドの詳細

SIZE

private static final int **SIZE**

検索メッシュの最大点数（処理高速化のため、配列を使用）

SIZE2

private static final int **SIZE2**

入力データの最大点数（処理高速化のため、配列を使用）

---

sprinkleSize

private int **sprinkleSize**

散布車制御（データ数）

---

sid

private int[] **sid**

散布車制御（車線 ID）

---

skp

private double[] **skp**

散布車制御（開始 KP）

---

ekp

private double[] **ekp**

散布車制御（終了 KP）

---

order

private int[] **order**

走行順　1 から順に設定

---

volume

public double[] **volume**

散布車制御（散布量）

---

width

protected double[] **width**

散布車制御（散布幅）

---

direct

protected int[] **direct**

散布車制御（散布方向）

---

rate

protected int[] **rate**

散布車制御（湿式割合）

---

kind

protected int[] **kind**

散布車制御（薬剤種類）

---

mSize

private int **mSize**

メッシュデータ数

---

no

protected int[] **no**

メッシュ番号

id

protected int[] id

車線 ID 下り　追越車線:-1　走行車線:-2　副道:-3 以下（車線毎に順番に付番） 上り　追越車線:1　走行車線: 2　副道: 3 以上（車線毎に順番に付番）

---

id_max

public int id_max

車線 ID の最大値

---

kp

protected double[] kp

KP(作業用)

---

kp0

protected double[] kp0

KP

---

x0

public double[] x0

メッシュ４点座標（左下 X）

---

y0

public double[] y0

メッシュ４点座標（左下 Y）

---

x1

public double[] **x1**

メッシュ４点座標（左上 X）

---

y1

public double[] **y1**

メッシュ４点座標（左上 Y）

---

x2

public double[] **x2**

メッシュ４点座標（右上 X）

---

y2

public double[] **y2**

メッシュ４点座標（右上 Y）

---

x3

public double[] **x3**

メッシュ４点座標（右下 X）

---

y3

public double[] **y3**

メッシュ４点座標（右下 Y）

---

nS

public int **nS**

サーチメッシュの数

---

mN

public int[][] **mN**

メッシュの存在の有無 0:無 1:有

---

mD

public int[][] **mD**

メッシュ格納配列番号

---

mF

protected int[][] **mF**

散布情報変更フラグ 1:散布開始 0:変更無 -1:散布終了

---

mE

public int[][] **mE**

散布済フラグ 0:散布済またはメッシュなし 1 以上:未散布（散布順番号）

---

mM

public int[][] **mM**

散布車制御情報格納配列番号

---

mG

public int[][] **mG**

散布回数

---

workpattern

public java.lang.String **workpattern**

標準作業パターン格納文字列

コンストラクタの詳細

ReadFileSprincler

public **ReadFileSprincler**(java.lang.String meshfile,
java.lang.String sprinklefile)
コンストラクタ
パラメータ:
meshfile -
sprinclefile -

メソッドの詳細

readMesh

public void **readMesh**(java.lang.String fileName)
範囲判定用メッシュファイルの読込
パラメータ:
fileName -

---

readSprinkle

public boolean **readSprinkle**(java.lang.String fileName)
散布計画ファイルの読込
パラメータ:
fileName - 散布計画ファイル

---

addMesh

private void **addMesh**(java.lang.String fileName)
散布制御用メッシュデータの生成
パラメータ:
fileName - 散布制御用メッシュデータファイル

---

addMesh2

public void **addMesh2**()

散布制御用メッシュデータの再生成

---

inMesh

protected boolean **inMesh**(int i,
double x,
double y)

与えられた点がメッシュ内にあるか判定

パラメータ:

i - メッシュの配列番号

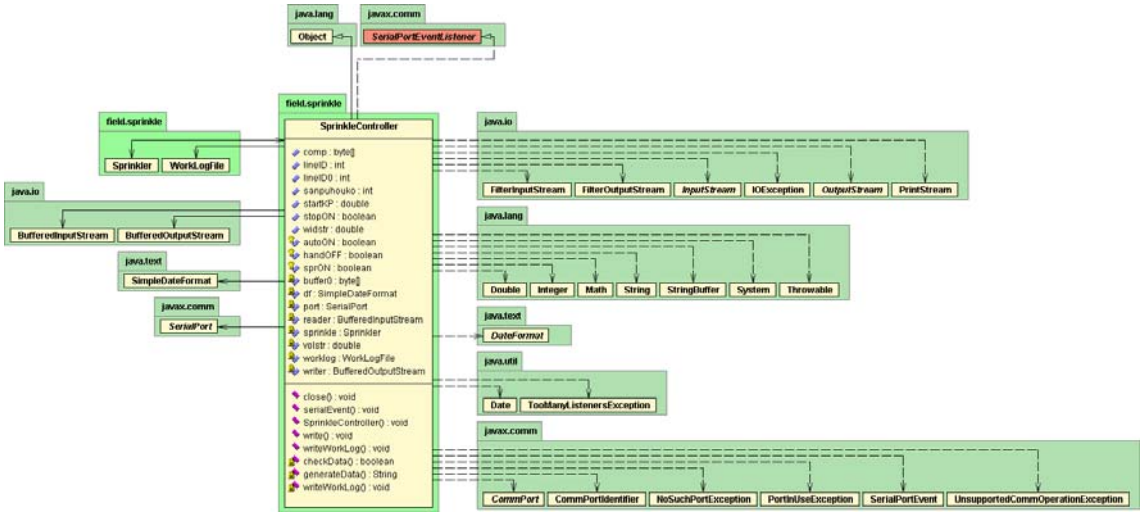
x - 与えられた点の X 座標

y - 与えられた点の Y 座標

戻り値:

メッシュ内:true メッシュ外:false

３　クラス SprinkleController



java.lang.Object

　|

　+--field.sprinkle.SprinkleController

---

public class SprinkleController

extends java.lang.Object

implements javax.comm.SerialPortEventListener

散布車制御データを RS-232C を通して制御するクラス

フィールドの概要	
protected　boolean	**autoON** 散布装置が手動か自動かを表すフラグ　ture:自動　false:手動
private　byte[]	**buffer0** RS-232C からのデータの保存
byte[]	**comp** 散布制御変更チェック用フラグ
private　java.text. SimpleDateFormat	**df** データフォーマットクラスのインスタンス
protected　boolean	**handOFF** 散布装置が手動から自動に変わった状態を表すフラグ　true:手

	動から自動 false:それ以外
int	**lineID** 車線 ID
int	**lineIDO** 前回車線 ID
private javax.comm.SerialPort	**port** SerialPort クラスのインスタンス
private java.io.BufferedInputStream	**reader** BufferedInputStream クラスのインスタンス
int	**sanpuhouko** 散布方向
private Sprinkler	**sprinkle** 散布クラスのインスタンス
protected boolean	**sprON** 散布状態を表すフラグ true:散布 false:散布停止
double	**startKP** 開始 KP
boolean	**stopON** 散布 ON から散布 OFF に変わった場合を表すフラグ true:ON->OFF false:それ以外
private double	**volstr** 散布量
double	**widstr** 散布幅
private WorkLogFile	**worklog** WorkLogFile クラスのインスタンス
private java.io.BufferedOutputStream	**writer** PrintWriter クラスのインスタンス

コンストラクタの概要

SprinkleController(Sprinkler spr, java.lang.String portName, WorkLogFile worklogfile) コンストラクタ

メソッドの概要	
private boolean	**checkData**(byte[] buffer) 散布制御装置からのデータの通信エラーチェック
void	**close**() ポートのクローズ処理
private java.lang.String	**generateData**(int line_id, byte[] buffer) 散布制御装置からのデータを解析して CSV 形式に変換する
void	**serialEvent**(javax.comm.SerialPortEvent event) SerialPortEvent 処理ルーチン
void	**write**(byte[] para) RS-232C に対して出力
void	**writeWorkLog**() 作業記録用作業ファイルへの書出し
private void	**writeWorkLog**(byte[] buffer) 作業記録用作業ファイルへの書出し

フィールドの詳細

port

private javax.comm.SerialPort **port**

SerialPort クラスのインスタンス

---

startKP

public double **startKP**

開始 KP

---

lineID

public int **lineID**
車線 ID

---

lineID0

public int **lineID0**
前回車線 ID

---

writer

private java.io.BufferedOutputStream **writer**
PrintWriter クラスのインスタンス

---

reader

private java.io.BufferedInputStream **reader**
BufferedInputStream クラスのインスタンス

---

worklog

private WorkLogFile **worklog**
WorkLogFile クラスのインスタンス

---

df

private java.text.SimpleDateFormat **df**
データフォーマットクラスのインスタンス

---

autoON

protected boolean **autoON**

散布装置が手動か自動かを表すフラグ ture:自動 false:手動

---

sprON

protected boolean **sprON**

散布状態を表すフラグ true:散布　false:散布停止

---

handOFF

protected boolean **handOFF**

散布装置が手動から自動に変わった状態を表すフラグ true:手動から自動 false:それ以外

---

sprinkle

private Sprinkler **sprinkle**

散布クラスのインスタンス

---

volstr

private double **volstr**

散布量

---

widstr

public double **widstr**

散布幅

---

sanpuhouko

public int **sanpuhouko**

散布方向

---

comp

public byte[] **comp**

散布制御変更チェック用フラグ

---

stopON

public boolean **stopON**

散布 ON から散布 OFF に変わった場合を表すフラグ true:ON->OFF false:それ以外

---

buffer0

private byte[] **buffer0**

RS-232C からのデータの保存

コンストラクタの詳細

SprinkleController

public **SprinkleController**(Sprinkler spr,
java.lang.String portName,
WorkLogFile worklogfile)

コンストラクタ

パラメータ:

portName - ポート名:COM1 etc.

メソッドの詳細

close

public void **close**()

ポートのクローズ処理

---

write

public void **write**(byte[] para)

RS-232C に対して出力

---

serialEvent

public void **serialEvent**(javax.comm.SerialPortEvent event)

SerialPortEvent 処理ルーチン

定義:

インタフェース javax.comm.SerialPortEventListener 内の serialEvent

パラメータ:

event - SerialPortEvent

---

checkData

private boolean **checkData**(byte[] buffer)

散布制御装置からのデータの通信エラーチェック

パラメータ:

buffer - 散布制御装置からのデータ

戻り値:

true エラーなし

---

writeWorkLog

public void **writeWorkLog**()

作業記録用作業ファイルへの書出し

---

writeWorkLog

private void **writeWorkLog**(byte[] buffer)

作業記録用作業ファイルへの書出し

パラメータ:

buffer - 散布制御装置からの出力

---

generateData

private java.lang.String **generateData**(int line_id,
byte[] buffer)

散布制御装置からのデータを解析して CSV 形式に変換する

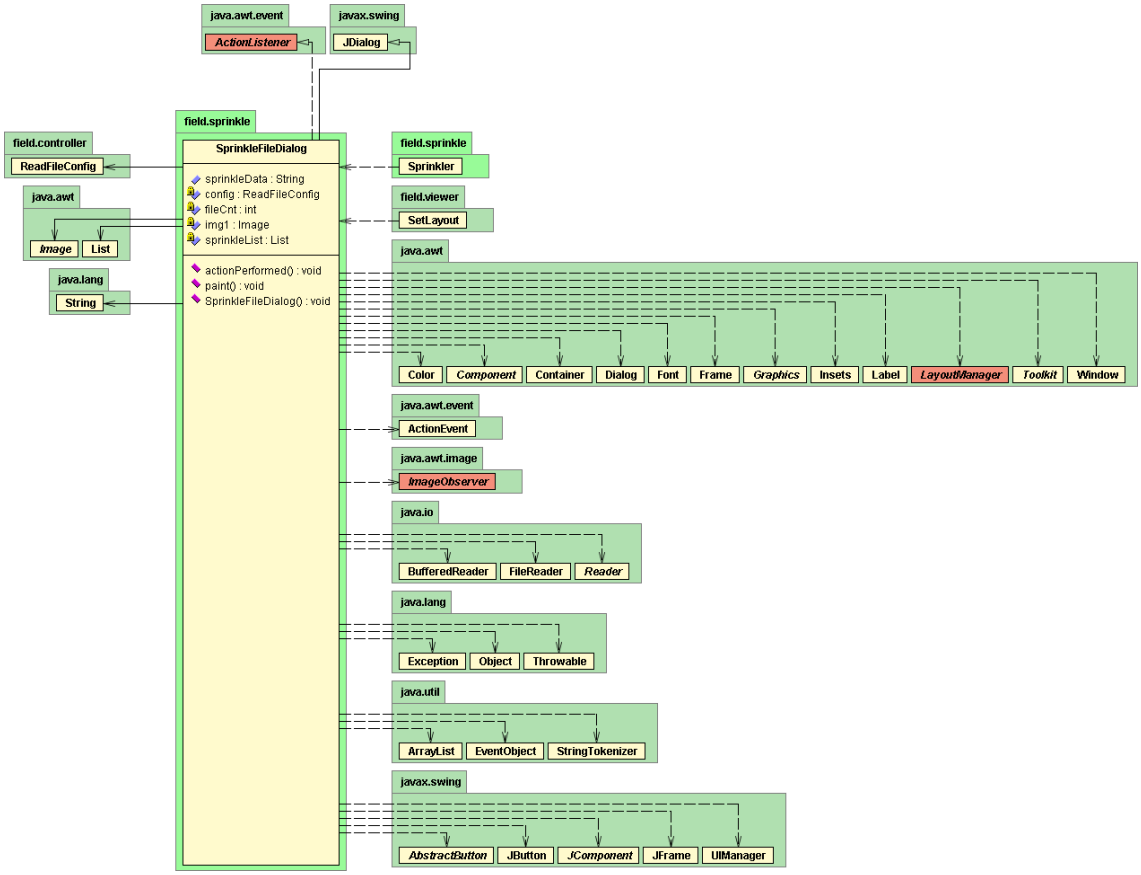
パラメータ:

buffer - 散布制御装置からのデータ

戻り値:

CSV 形式のデータ文字列

■ 4　クラス SprinkleFileDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.sprinkle.SprinkleFileDialog
```

---

public class SprinkleFileDialog

extends javax.swing.JDialog

implements java.awt.event.ActionListener

散布ファイル選択画面クラス

フィールドの概要	
private ReadFileConfig	**config** 設定ファイル読込クラス
private int	**fileCnt** 散布計画ファイル数
private java.awt.Image	**img1** イメージアイコン
java.lang.String	**sprinkleData** 散布計画ファイル選択値
private java.awt.List	**sprinkleList** 散布計画ファイル一覧

コンストラクタの概要
SprinkleFileDialog(ReadFileConfig conf, int spkFlg) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント
void	**paint**(java.awt.Graphics g) ペイント処理

フィールドの詳細

fileCnt

private int **fileCnt**

散布計画ファイル数

---

img1

private java.awt.Image **img1**

イメージアイコン

---

sprinkleList

private java.awt.List **sprinkleList**

散布計画ファイル一覧

---

sprinkleData

public java.lang.String **sprinkleData**

散布計画ファイル選択値

---

config

private ReadFileConfig **config**

設定ファイル読込クラス

コンストラクタの詳細

SprinkleFileDialog

public **SprinkleFileDialog**(ReadFileConfig conf,
int spkFlg)

コンストラクタ

パラメータ:

conf - 設定ファイル

spkFlg - 0:起動時の呼び出し, 1:２回目以降の呼び出し

メソッドの詳細

paint

public void **paint**(java.awt.Graphics g)

ペイント処理

オーバーライド:

クラス java.awt.Container 内の paint

パラメータ:

g - Graphics

---

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

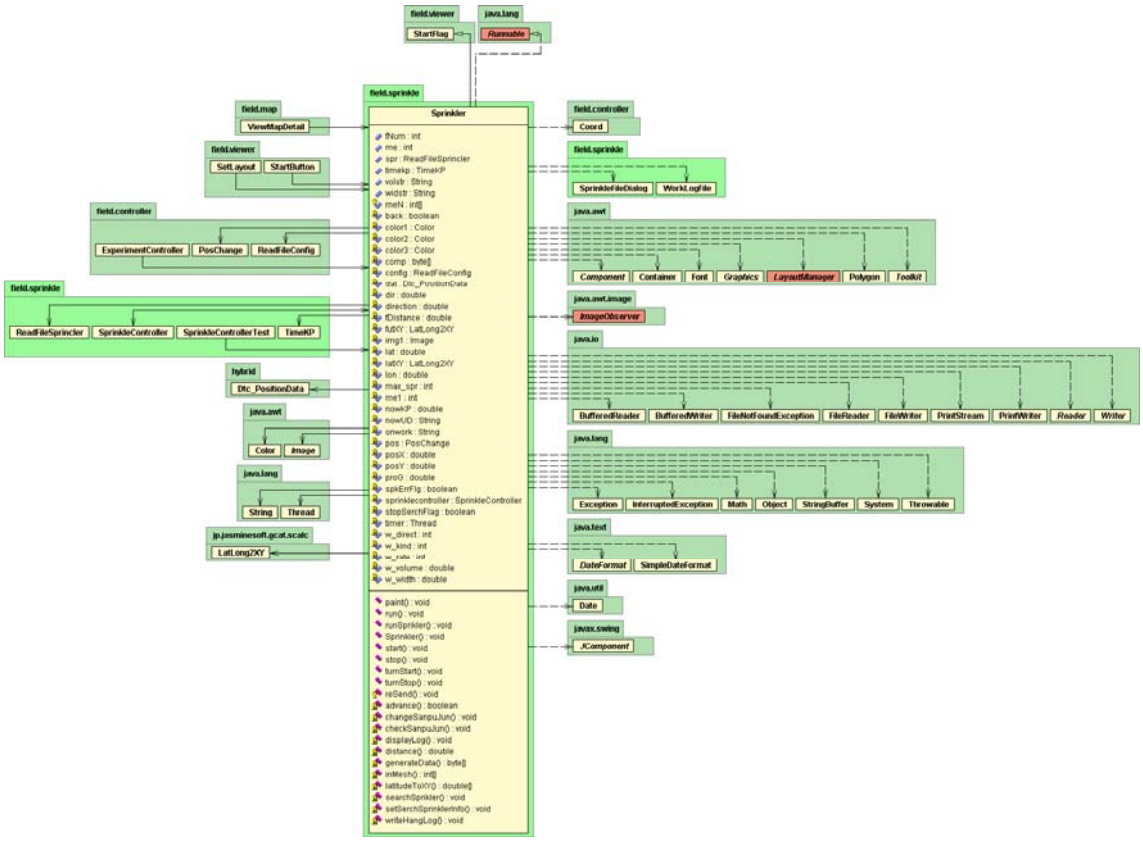
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

■ ５　クラス Sprinkler




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.StartFlag
                                |
                                +--field.sprinkle.Sprinkler
```

---

public class **Sprinkler**

extends StartFlag

implements java.lang.Runnable

散布車を制御するクラス

フィールドの概要	
private boolean	**back** 走行バックフラグ
private java.awt.Color	**color1** 背景色１
private java.awt.Color	**color2** 背景色２
private java.awt.Color	**color3** 背景色３
private byte[]	**comp** 散布制御変更チェック用フラグ
private ReadFileConfig	**config** 設定ファイル読込クラス
private Dtc_PositionData	**dat** 位置情報クラスのインスタンス
private double	**dir** 位置特定パッケージからの進行方向
private double	**direction** 方位計表示用方位角
private double	**fDistance** 次回散布指示までの距離
int	**fNum** 現在いる位置のメッシュ数
private jp.jasminesoft.gcat.scalc.LatLong2XY	**futXY** 道路地物の緯度経度から平面直角座標系に変換するクラスのインスタンス
private java.awt.Image	**img1** パネル画像イメージ

private double	**lat** 位置特定パッケージからの緯度
private jp.jasminesoft.gcat.scalc.LatLong2XY	**latXY** 背景地図の緯度経度から平面直角座標系に変換するクラスのインスタンス
private double	**lon** 位置特定パッケージからの経度
private int	**max_spr** 車線 ID 数
int	**me** 現在のメッシュ番号
private int	**me1** メッシュ探査開始メッシュ番号
protected int[]	**meN** 作業記録複数車線書き出しフラグ
private double	**nowKP** KP 距離（KP 表示用）
private java.lang.String	**nowUD** 上り下りの別（KP 表示用）
private java.lang.String	**onwork** "散布中"または"散布停止中"文字
private PosChange	**pos** 世界座標系の緯度経度を日本座標系の緯度経度に変換するクラスのインスタンス
private double	**posX** 現在位置（経度）
private double	**posY** 現在位置（緯度）
private double	**proG** 現在位置（進行方向角）
private boolean	**spkErrFlg**

	散布順エラーフラグ true:散布順ＯＫ false:散布順エラー
ReadFileSprincler	**spr** 散布制御データ読込クラス
private SprinkleController	**sprinklecontroller** 散布車制御装置コントロールクラスのインスタンス
private boolean	**stopSerchFlag** 散布変更箇所が散布停止箇所かどうかを表すフラグ true:散布停止箇所 false:散布変更箇所
TimeKP	**timekp** KP ファイル出力クラス
private java.lang.Thread	**timer** スレッドタイマークラス
java.lang.String	**volstr** 散布量表示文字列
private int	**w_direct** ＲＳ２３２Ｃデータ転送用（散布方向）
private int	**w_kind** ＲＳ２３２Ｃデータ転送用（薬剤種類）
private int	**w_rate** ＲＳ２３２Ｃデータ転送用（湿式割合）
private double	**w_volume** ＲＳ２３２Ｃデータ転送用（散布量）
private double	**w_width** ＲＳ２３２Ｃデータ転送用（散布幅）
java.lang.String	**widstr** 散布幅表示文字列

クラス field.viewer.StartFlag から継承したフィールド
startflag

コンストラクタの概要
Sprinkler(ReadFileConfig con, Dtc_PositionData d, WorkLogFile worklog, boolean ini) コンストラクタ

メソッドの概要	
private boolean	**advance**(int j, int i) 進んでいる方向が、基準点距離が増加方向に進んでいるか減少方向に進んでいるかを判定する。
private void	**changeSanpuJun**(int me0, int me9, int j, int lineID) 散布後の散布順変更
private void	**checkSanpuJun**(int me0, int me9, int j, int lineID) 散布順エラー時の散布指示フラグ変更
private void	**displayLog**() 散布指示ログの表示
private double	**distance**(int j, int i, double xx2, double yy2) 探査道路地物までの距離を計算する。
private byte[]	**generateData**(int mode, double volume, double width, int direct, int rate, int kind) 散布車制御データ（ON）を生成する
private int[]	**inMesh**() 現在位置がメッシュ内にあるか判定する。
private double[]	**latitudeToXY**(jp.jasminesoft.gcat.scalc.LatLong2XY lat, double x, double y, double r) 緯度、経度を平面直角座標系に変換する。
void	**paint**(java.awt.Graphics g) 散布情報を表示する
protected void	**reSend**() RS-232C へのデータ再送

void	**run**() スレッドのメイン処理
void	**runSprikler**() 緯度・経度情報を取得して、散布制御を行う
private void	**searchSprikler**() 散布制御の探査を行う。
private void	**setSerchSprinklerInfo**(int j, int i, double dis) 次の散布情報のセット
void	**start**() スレッドのスタート
void	**stop**() スレッドの終了
void	**turnStart**() 円盤の回転を開始する
void	**turnStop**() 円盤の回転を停止する
private void	**writeHangLog**() ハングアップ時、復活用ログへの出力

フィールドの詳細

max_spr

private int **max_spr**

車線 ID 数

---

dat

private Dtc_PositionData **dat**

位置情報クラスのインスタンス

---

timer

private java.lang.Thread **timer**

スレッドタイマークラス

---

spr

public ReadFileSprincler **spr**

散布制御データ読込クラス

---

posX

private double **posX**

現在位置（経度）

---

posY

private double **posY**

現在位置（緯度）

---

proG

private double **proG**

現在位置（進行方向角）

---

pos

private PosChange **pos**

世界座標系の緯度経度を日本座標系の緯度経度に変換するクラスのインスタンス

---

latXY

private jp.jasminesoft.gcat.scalc.LatLong2XY **latXY**

背景地図の緯度経度から平面直角座標系に変換するクラスのインスタンス

futXY

private jp.jasminesoft.gcat.scalc.LatLong2XY **futXY**

道路地物の緯度経度から平面直角座標系に変換するクラスのインスタンス

---

fNum

public int **fNum**

現在いる位置のメッシュ数

---

me1

private int **me1**

メッシュ探査開始メッシュ番号

---

me

public int **me**

現在のメッシュ番号

---

color1

private java.awt.Color **color1**

背景色１

---

color2

private java.awt.Color **color2**

背景色２

---

color3

private java.awt.Color **color3**

背景色３

---

onwork

private java.lang.String **onwork**

"散布中"または"散布停止中"文字

---

fDistance

private double **fDistance**

次回散布指示までの距離

---

volstr

public java.lang.String **volstr**

散布量表示文字列

---

widstr

public java.lang.String **widstr**

散布幅表示文字列

---

nowKP

private double **nowKP**

KP 距離（KP 表示用）

---

nowUD

private java.lang.String **nowUD**

上り下りの別（KP 表示用）

---

direction

private double **direction**

方位計表示用方位角

---

sprinklecontroller

private SprinkleController **sprinklecontroller**

散布車制御装置コントロールクラスのインスタンス

---

stopSerchFlag

private boolean **stopSerchFlag**

散布変更箇所が散布停止箇所かどうかを表すフラグ true:散布停止箇所 false:散布変更箇所

---

spkErrFlg

private boolean **spkErrFlg**

散布順エラーフラグ true:散布順ＯＫ false:散布順エラー

---

w_volume

private double **w_volume**

ＲＳ２３２Ｃデータ転送用（散布量）

---

w_width

private double **w_width**

ＲＳ２３２Ｃデータ転送用（散布幅）

---

w_direct

private int **w_direct**

ＲＳ２３２Ｃデータ転送用（散布方向）

w_rate

private int **w_rate**

ＲＳ２３２Ｃデータ転送用（湿式割合）

---

w_kind

private int **w_kind**

ＲＳ２３２Ｃデータ転送用（薬剤種類）

---

config

private ReadFileConfig **config**

設定ファイル読込クラス

---

lon

private double **lon**

位置特定パッケージからの経度

---

lat

private double **lat**

位置特定パッケージからの緯度

---

dir

private double **dir**

位置特定パッケージからの進行方向

---

img1

private java.awt.Image **img1**

パネル画像イメージ

---

meN

protected int[] **meN**

作業記録複数車線書き出しフラグ

---

comp

private byte[] **comp**

散布制御変更チェック用フラグ

---

timekp

public TimeKP **timekp**

KP ファイル出力クラス

---

back

private boolean **back**

走行バックフラグ

コンストラクタの詳細

Sprinkler

public **Sprinkler**(ReadFileConfig con,
DtcPositionData d,
WorkLogFile worklog,
boolean ini)

コンストラクタ

パラメータ:

config - 設定ファイル読込クラス

d - 位置情報クラス

メソッドの詳細

paint

public void **paint**(java.awt.Graphics g)

散布情報を表示する

オーバーライド:

クラス javax.swing.JComponent 内の paint

パラメータ:

g - java.awt.Graphics

---

run

public void **run**()

スレッドのメイン処理

定義:

インタフェース java.lang.Runnable 内の run

---

start

public void **start**()

スレッドのスタート

---

stop

public void **stop**()

スレッドの終了

---

runSprikler

public void **runSprikler**()

緯度・経度情報を取得して、散布制御を行う

---

searchSprikler

private void **searchSprikler**()

散布制御の探査を行う。

---

changeSanpuJun

private void **changeSanpuJun**(int me0,
int me9,
int j,
int lineID)

散布後の散布順変更

パラメータ:

me0 -

me9 -

j -

i -

---

displayLog

private void **displayLog**()

散布指示ログの表示

---

checkSanpuJun

private void **checkSanpuJun**(int me0,
int me9,
int j,
int lineID)

散布順エラー時の散布指示フラグ変更

パラメータ:

j - 現在車線の配列番号

i - 現在走行位置の配列番号

---

setSerchSprinklerInfo

```
private void setSerchSprinklerInfo(int j,
                                   int i,
                                   double dis)
```

次の散布情報のセット

パラメータ:

j -

i -

dis - 次散布箇所

---

distance

```
private double distance(int j,
                        int i,
                        double xx2,
                        double yy2)
```

探査道路地物までの距離を計算する。

パラメータ:

j - 散布制御メッシュの車線方向配列番号

i - 散布制御メッシュの進行方向配列番号

xx2 - 現在位置 X 座標

yy2 - 現在位置 Y 座標

戻り値:

最近探査道路地物までの距離

---

advance

```
private boolean advance(int j,
                        int i)
```

進んでいる方向が、基準点距離が増加方向に進んでいるか減少方向に進んでいるかを判定する。

パラメータ:

j - 散布制御メッシュの車線方向配列番号

i - 散布制御メッシュの進行方向配列番号

戻り値:

true:増加方向、false:減少方向

---

inMesh

```
private int[] inMesh()
```

現在位置がメッシュ内にあるか判定する。

（高速化のため、１回前の位置メッシュからサーチ開始）

戻り値:

現在位置のメッシュ番号（メッシュ内にない場合は-1）

---

latitudeToXY

```
private double[] latitudeToXY(jp.jasminesoft.gcat.scalc.LatLong2XY lat,
                              double x,
                              double y,
                              double r)
```

緯度、経度を平面直角座標系に変換する。また、方向角をラジアンに変換する。

パラメータ:

lat - 座標変換クラスのインスタンス

x - 経度

y - 緯度

r - 方向角

戻り値:

平面直角座標(X,Y)、方向角（ラジアン）

---

turnStart

```
public void turnStart()
```

円盤の回転を開始する

---

turnStop

public void **turnStop**()

円盤の回転を停止する

---

generateData

private byte[] **generateData**(int mode,
double volume,
double width,
int direct,
int rate,
int kind)

散布車制御データ（ON）を生成する

パラメータ:

mode - 散布装置 ON-OFF 制御 0x00:散布停止 0x01:散布待機 0x02:散布

volume - 散布密度

width - 散布幅

direct - 散布方向

rate - 湿式割合

kind - 散布薬剤 0x01:フレーク 0x02:粒 0x03:粒大 0x04:原塩

戻り値:

RS-232C 送信用データ(散布開始・変更)

---

reSend

protected void **reSend**()

RS-232C へのデータ再送

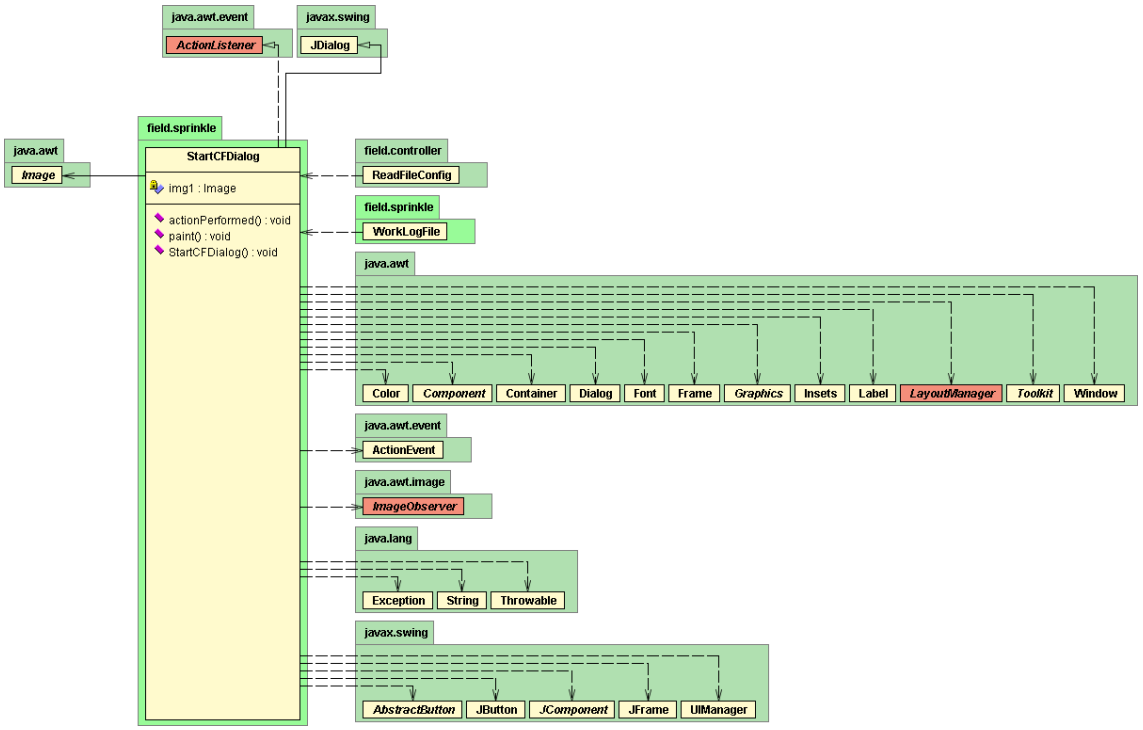
---

writeHangLog

private void **writeHangLog**()

ハングアップ時、復活用ログへの出力

■ ６　クラス StartCFDialog




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Dialog
                          |
                          +--javax.swing.JDialog
                                |
                                +--field.sprinkle.StartCFDialog
```

---

public class **StartCFDialog**

extends javax.swing.JDialog

implements java.awt.event.ActionListener

外部記憶装置ファイル確認画面クラス

フィールドの概要	
private java.awt.Image	**img1** イメージアイコン

コンストラクタの概要
StartCFDialog() コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンイベント
void	**paint**(java.awt.Graphics g) ペイント処理

フィールドの詳細

img1

private java.awt.Image **img1**

イメージアイコン

コンストラクタの詳細

StartCFDialog

public **StartCFDialog**()

コンストラクタ

メソッドの詳細

paint

public void **paint**(java.awt.Graphics g)

ペイント処理

オーバーライド:

クラス java.awt.Container 内の paint

パラメータ:

g - Graphics

---

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンイベント

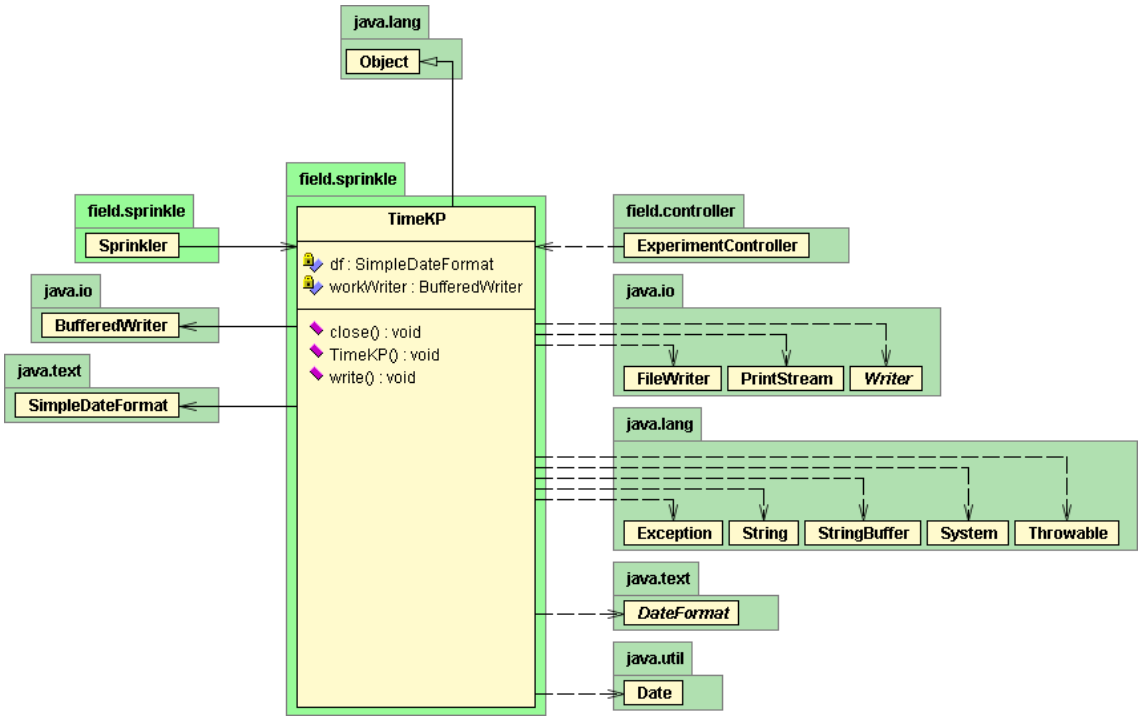
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

■ 7　クラス TimeKP



java.lang.Object

　|

　+--**field.sprinkle.TimeKP**

---

public class **TimeKP**

extends java.lang.Object

時間、KP をファイルに出力するクラス

フィールドの概要	
private　java.text.SimpleDateFormat	**df** データフォーマットクラスのインスタンス
private　java.io.BufferedWriter	**workWriter** ファイル書込みクラスのインスタンス

コンストラクタの概要
TimeKP(java.lang.String junkaiKpFile) コンストラクタ

メソッドの概要	
void	**close**() KP ファイルのクローズ処理
void	**write**(int id, double kp) KP の出力

フィールドの詳細

workWriter

private java.io.BufferedWriter **workWriter**

ファイル書込みクラスのインスタンス

---

df

private java.text.SimpleDateFormat **df**

データフォーマットクラスのインスタンス

コンストラクタの詳細

TimeKP

public **TimeKP**(java.lang.String junkaiKpFile)

コンストラクタ

メソッドの詳細

write

```
public void write(int id,
                  double kp)
```

KP の出力

パラメータ:

str - 散布履歴情報用文字列

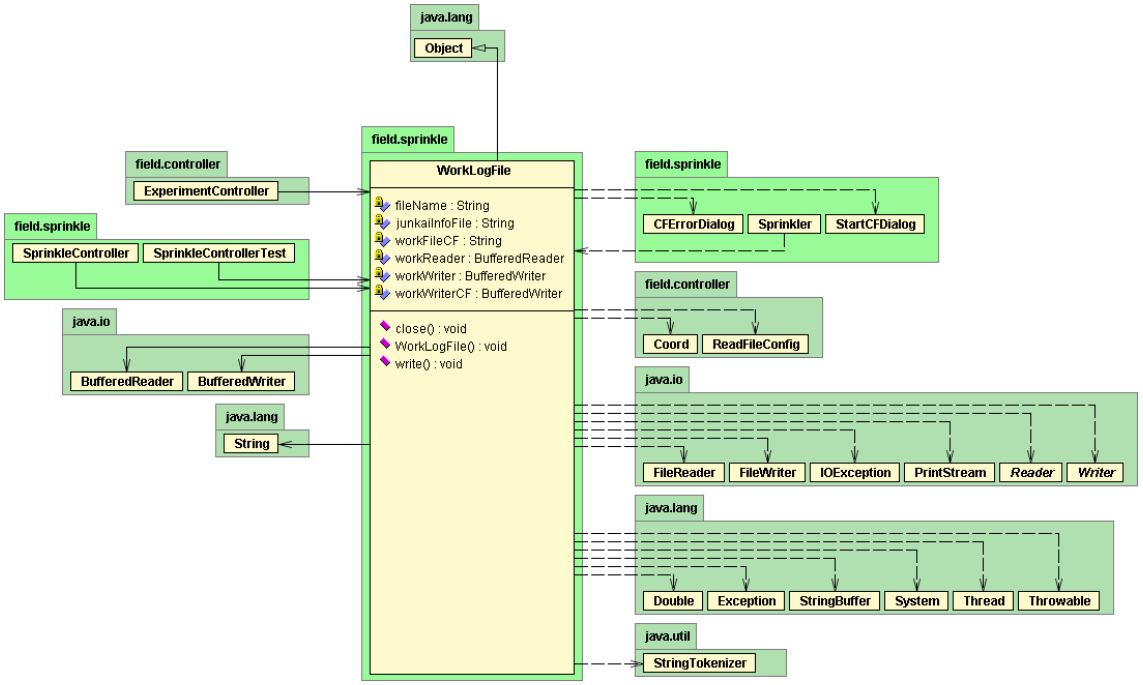
---

close

```
public void close()
```

KP ファイルのクローズ処理

■ 8　クラス WorkLogFile



java.lang.Object

　|

　+--**field.sprinkle.WorkLogFile**

---

public class **WorkLogFile**

extends java.lang.Object

作業記録をファイルに出力するクラス

フィールドの概要	
private java.lang.String	**fileName** 作業記録保存ワークファイル名
private java.lang.String	**junkaiInfoFile** 巡回情報ファイル名
private java.lang.String	**workFileCF** 作業記録コンパクトフラッシュ用ファイル名
private java.io.BufferedReader	**workReader** ファイル読込みクラスのインスタンス

private java.io.BufferedWriter	**workWriter** ファイル書込みクラスのインスタンス
private java.io.BufferedWriter	**workWriterCF** ファイル書込みクラス（コンパクトフラッシュ用）のインスタンス

コンストラクタの概要
WorkLogFile(ReadFileConfig config) コンストラクタ

メソッドの概要	
void	**close**() 作業記録ファイルのクローズ処理
void	**write**(java.lang.String str) 散布履歴情報用文字列の出力

フィールドの詳細

workWriter

private java.io.BufferedWriter **workWriter**

ファイル書込みクラスのインスタンス

---

workWriterCF

private java.io.BufferedWriter **workWriterCF**

ファイル書込みクラス（コンパクトフラッシュ用）のインスタンス

---

workReader

private java.io.BufferedReader **workReader**

ファイル読込みクラスのインスタンス

---

fileName

private java.lang.String **fileName**

作業記録保存ワークファイル名

---

workFileCF

private java.lang.String **workFileCF**

作業記録コンパクトフラッシュ用ファイル名

---

junkaiInfoFile

private java.lang.String **junkaiInfoFile**

巡回情報ファイル名

コンストラクタの詳細

WorkLogFile

public **WorkLogFile**(ReadFileConfig config)

コンストラクタ

パラメータ:

config - field.controller.ReadFileConfig

メソッドの詳細

write

public void **write**(java.lang.String str)

散布履歴情報用文字列の出力

パラメータ:

str - 散布履歴情報用文字列

close

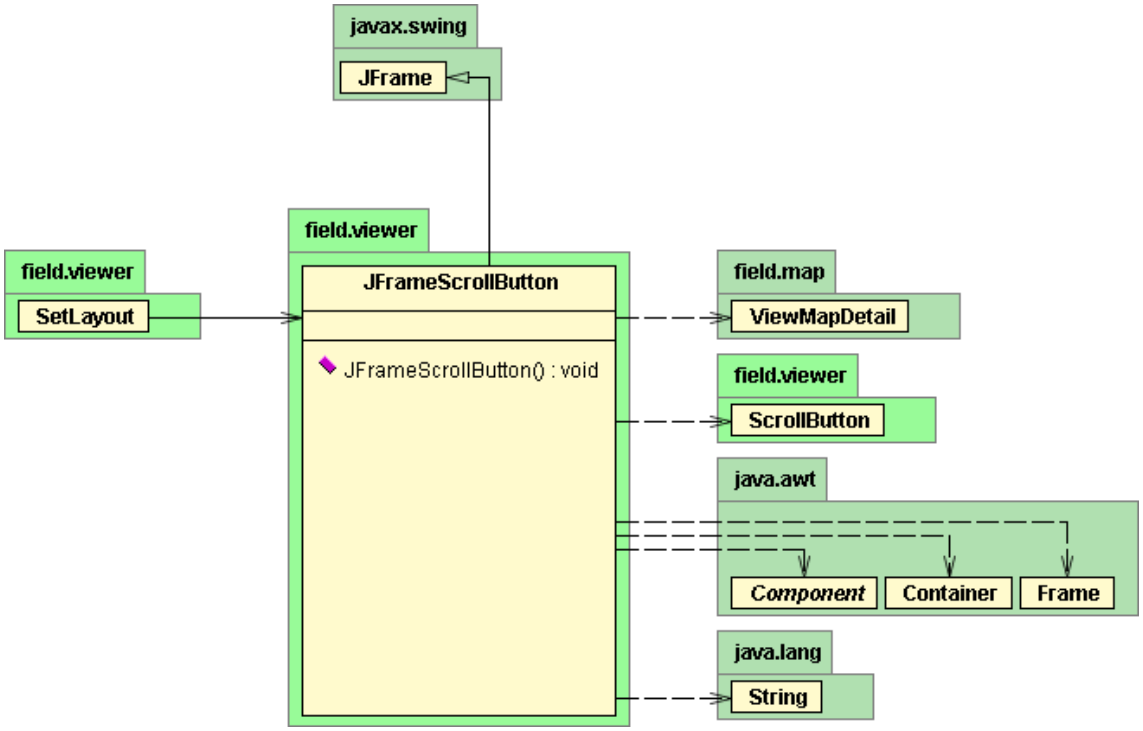
public void **close**()

作業記録ファイルのクローズ処理

f） パッケージ field.viewer

クラスの概要	
JFrameScrollButton	スクロールフレームクラス
JFrameSetupListTabstrip	表示変更フレームクラス
ScaleButton	スケールボタンの表示およびコントロールを行うクラス
ScrollButton	スクロールボタンの表示およびコントロールを行うクラス
SetLayout	ナビゲーション画面の各パネルの配置を行うクラス
SetupListFuture	表示設定タブ内に地物表示設定のチェックボックス群を表示するクラス
SetupListMap	表示設定タブ内に背景地図表示設定のチェックボックス群を表示するクラス
SetupListSearch	表示設定タブ内に地物検索設定のチェックボックス群を表示するクラス
SetupListTabstrip	表示設定画面のタブ画面を生成するクラス
StartButton	画面上に「作業開始」「作業終了」ボタンを表示、コントロールするクラス。
StartFlag	開始・終了ステータスの共有化クラス。
ViewMapWhole	画面に全体地図を表示するクラス 画像は 280x180 ピクセルで作成

■ 1　クラス JFrameScrollButton




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Frame
                          |
                          +--javax.swing.JFrame
                                |
                                +--field.viewer.JFrameScrollButton
```

---

public class JFrameScrollButton

extends javax.swing.JFrame

スクロールフレームクラス

コンストラクタの概要

JFrameScrollButton(ViewMapDetail viewmap, double scrollSize) 　　コンストラクタ

コンストラクタの詳細

JFrameScrollButton

public **JFrameScrollButton**(ViewMapDetail viewmap,
double scrollSize)

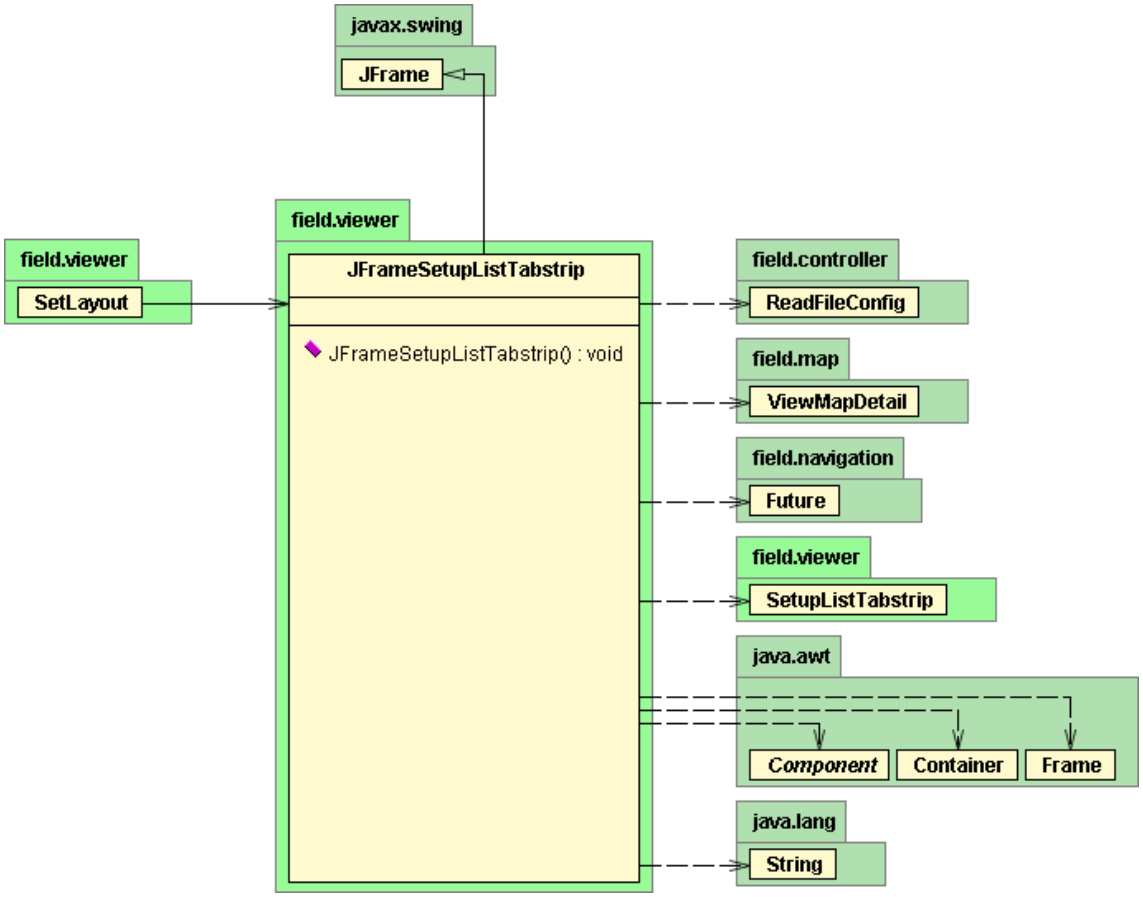
コンストラクタ

パラメータ:

viewmap - ViewMapDetail

scrollSize - スクロール量

■ 2　クラス JFrameSetupListTabstrip




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Frame
                          |
                          +--javax.swing.JFrame
                                |
                                +--field.viewer.JFrameSetupListTabstrip
```

---

public class JFrameSetupListTabstrip

extends javax.swing.JFrame

表示変更フレームクラス

コンストラクタの概要
JFrameSetupListTabstrip(ViewMapDetail viewmap, ReadFileConfig config, Future future) コンストラクタ

コンストラクタの詳細

JFrameSetupListTabstrip

public **JFrameSetupListTabstrip**(ViewMapDetail viewmap,
ReadFileConfig config,
Future future)

コンストラクタ

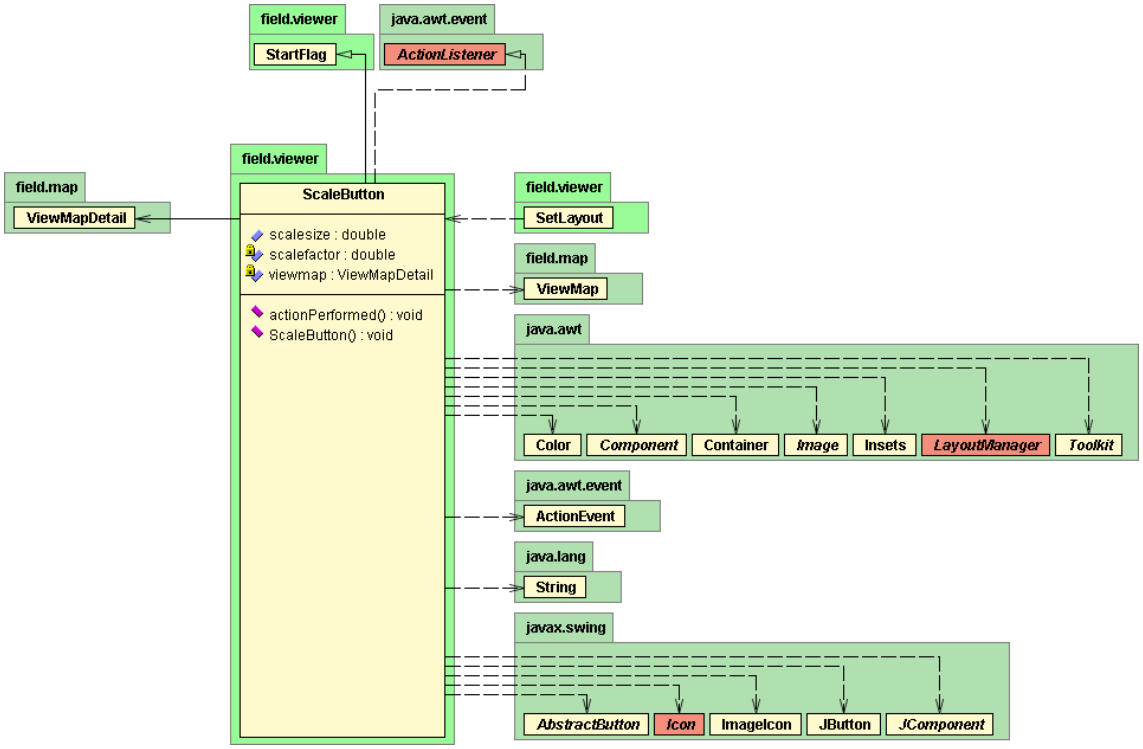
パラメータ:

viewmap - ViewMapDetail

config - ReadFileConfig

future - Future

■ 3　クラス ScaleButton




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.StartFlag
                                |
                                +--field.viewer.ScaleButton
```

---

public class **ScaleButton**

extends StartFlag

implements java.awt.event.ActionListener

スケールボタンの表示およびコントロールを行うクラス

フィールドの概要	
private double	**scalefactor** スケールファクター
double	**scalesize** スケールの拡大縮小率
private ViewMapDetail	**viewmap** jp.co.dawn_corp.geosql.j2.GeoSQLView

クラス field.viewer.StartFlag から継承したフィールド
startflag

コンストラクタの概要
ScaleButton(ViewMapDetail v) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) スケールボタンの判定および処理を行う

フィールドの詳細

scalefactor

private double **scalefactor**

スケールファクター

---

scalesize

public double **scalesize**

スケールの拡大縮小率

---

viewmap

private ViewMapDetail **viewmap**

jp.co.dawn_corp.geosql.j2.GeoSQLView

コンストラクタの詳細

ScaleButton

public **ScaleButton**(ViewMapDetail v)

コンストラクタ

パラメータ:

v - jp.co.dawn_corp.geosql.j2.GeoSQLView

メソッドの詳細

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

スケールボタンの判定および処理を行う

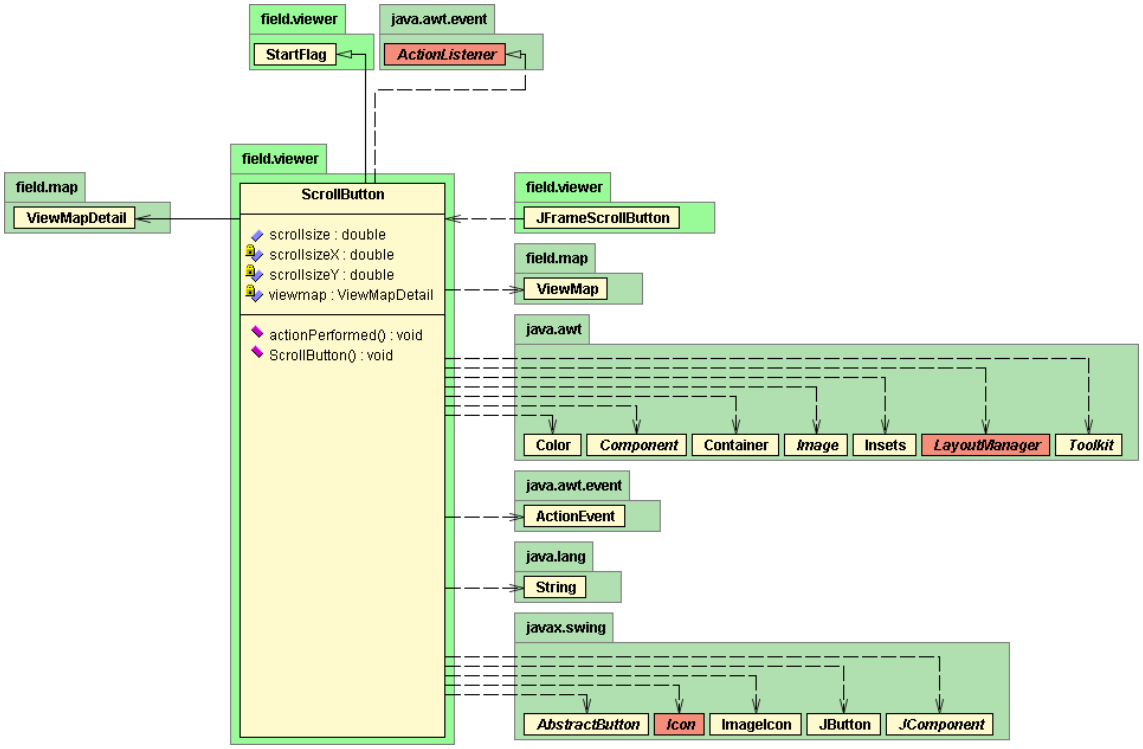
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

■ 4　クラス ScrollButton




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.StartFlag
                                |
                                +--field.viewer.ScrollButton
```

---

public class **ScrollButton**

extends StartFlag

implements java.awt.event.ActionListener

スクロールボタンの表示およびコントロールを行うクラス

フィールドの概要	
double	**scrollsize** ボタン１回当たりのスクロール移動値
private double	**scrollsizeX** 初期状態を０とした時の現在のＸ方向のスクロール値
private double	**scrollsizeY** 初期状態を０とした時の現在のＹ方向のスクロール値
private ViewMapDetail	**viewmap** jp.co.dawn_corp.geosql.j2.GeoSQLView

クラス field.viewer.StartFlag から継承したフィールド
startflag

コンストラクタの概要
ScrollButton(ViewMapDetail v) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) スクロールボタンの判定および処理を行う

フィールドの詳細

scrollsizeX

private double scrollsizeX

初期状態を０とした時の現在のＸ方向のスクロール値

---

scrollsizeY

private double scrollsizeY

初期状態を０とした時の現在のＹ方向のスクロール値

---

scrollsize

public double scrollsize

ボタン１回当たりのスクロール移動値

---

viewmap

private ViewMapDetail viewmap

jp.co.dawn_corp.geosql.j2.GeoSQLView

コンストラクタの詳細

ScrollButton

public ScrollButton(ViewMapDetail v)

コンストラクタ

パラメータ:

v - jp.co.dawn_corp.geosql.j2.GeoSQLView

メソッドの詳細

actionPerformed

public void actionPerformed(java.awt.event.ActionEvent e)

スクロールボタンの判定および処理を行う

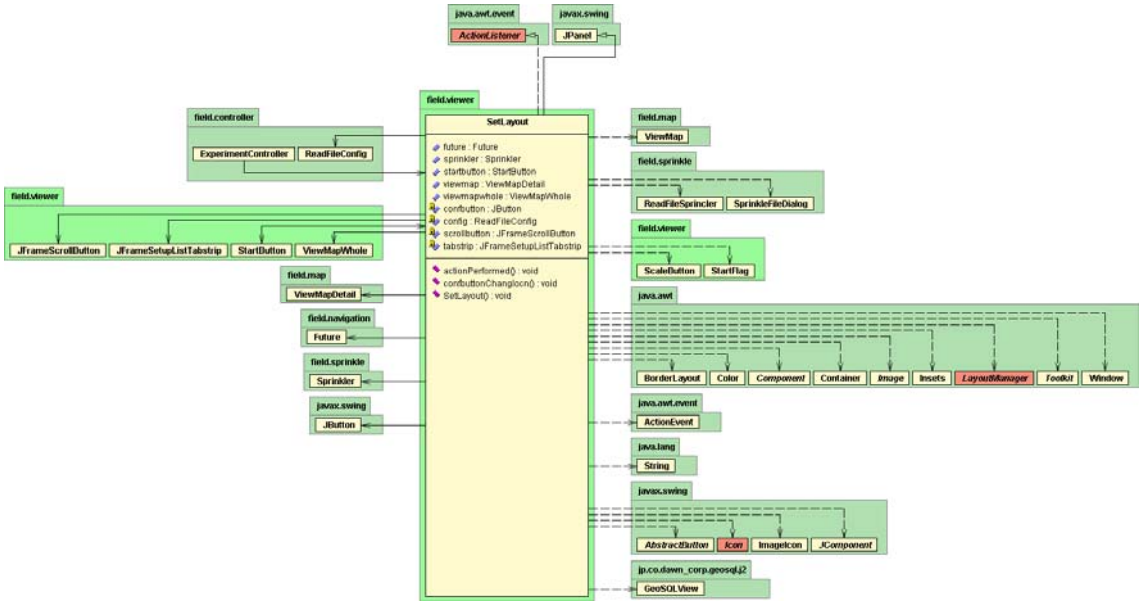
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

■ ５　クラス SetLayout




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.SetLayout
```

---

public class **SetLayout**

extends javax.swing.JPanel

implements java.awt.event.ActionListener

ナビゲーション画面の各パネルの配置を行うクラス

フィールドの概要	
private　javax.	**confbutton**

swing.JButton	散布計画選択ボタン
private ReadFileConfig	**config** 設定ファイルクラス
Future	**future** 地物検索クラスのインスタンス
private JFrameScrollButton	**scrollbutton** フレームクラスのインスタンス
Sprinkler	**sprinkler** 散布車制御クラスのインスタンス
StartButton	**startbutton** スタートボタンクラスのインスタンス
private JFrameSetupListTabstrip	**tabstrip** 表示変更フレームクラスのインスタンス
ViewMapDetail	**viewmap** jp.co.dawn_corp.geosql.j2.GeoSQLView
ViewMapWhole	**viewmapwhole** 全体地図クラスのインスタンス

コンストラクタの概要
SetLayout(ReadFileConfig conf, Sprinkler sprink) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) ボタンの判定および処理を行う
void	**confbuttonChanglocn**(boolean iconFlg) 散布計画選択ボタンの Icon の貼り付け

フィールドの詳細

viewmap

public ViewMapDetail **viewmap**

jp.co.dawn_corp.geosql.j2.GeoSQLView

---

startbutton

public StartButton **startbutton**

スタートボタンクラスのインスタンス

---

viewmapwhole

public ViewMapWhole **viewmapwhole**

全体地図クラスのインスタンス

---

future

public Future **future**

地物検索クラスのインスタンス

---

sprinkler

public Sprinkler **sprinkler**

散布車制御クラスのインスタンス

---

config

private ReadFileConfig **config**

設定ファイルクラス

---

scrollbutton

private JFrameScrollButton **scrollbutton**

フレームクラスのインスタンス

---

tabstrip

private JFrameSetupListTabstrip **tabstrip**

表示変更フレームクラスのインスタンス

---

confbutton

private javax.swing.JButton **confbutton**

散布計画選択ボタン

コンストラクタの詳細

SetLayout

public **SetLayout**(ReadFileConfig conf,
Sprinkler sprink)

コンストラクタ

パラメータ:

config - ReadFileConfig

sprink – Sprinkler

メソッドの詳細

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

ボタンの判定および処理を行う

定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

---

confbuttonChangIocn

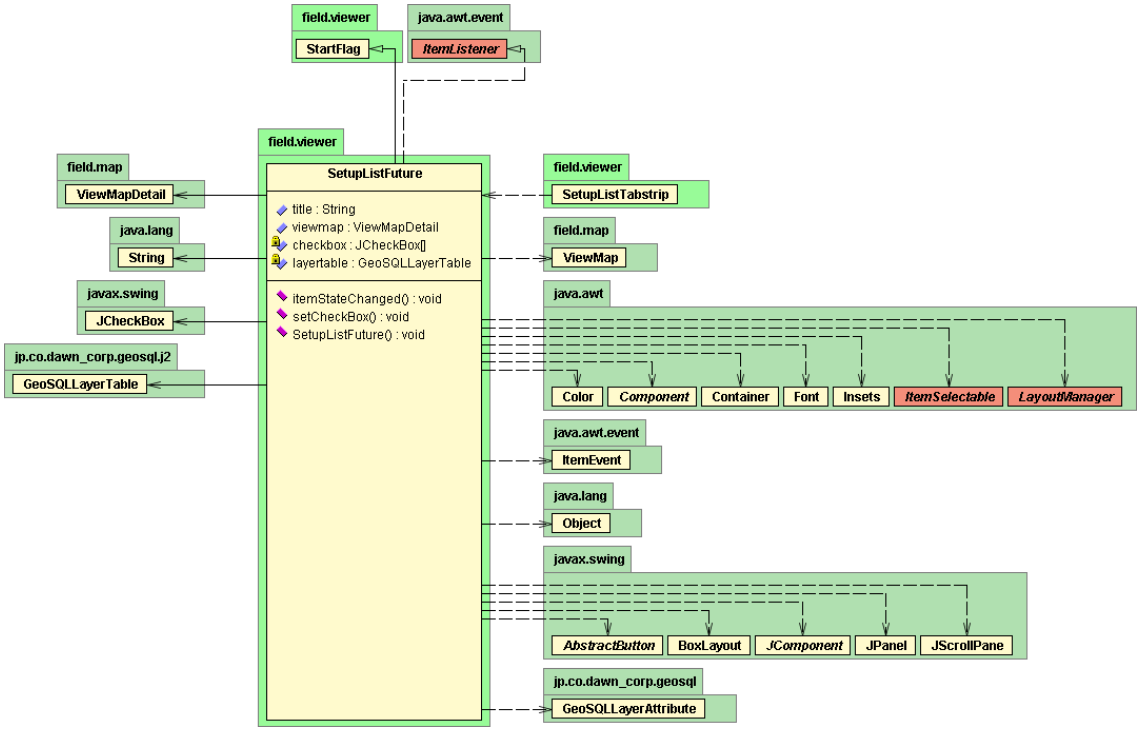
public void **confbuttonChangIocn**(boolean iconFlg)

散布計画選択ボタンの Icon の貼り付け

パラメータ:

iconFlg - true:OK Icon , false:NG Icon

■ ６　クラス SetupListFuture




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.StartFlag
                                |
                                +--field.viewer.SetupListFuture
```

---

public class SetupListFuture

extends StartFlag

implements java.awt.event.ItemListener

表示設定タブ内に地物表示設定のチェックボックス群を表示するクラス

フィールドの概要	
private javax.swing.JCheckBox[]	**checkbox** 道路地物表示変更用チェックボックス群
private jp.co.dawn_corp. geosql.j2.GeoSQLLayerTable	**layertable** 道路地物表示用レイヤーテーブル
java.lang.String	**title** 表示設定タブに表示する文字
ViewMapDetail	**viewmap** jp.co.dawn_corp.geosql.j2.GeoSQLView

クラス field.viewer.StartFlag から継承したフィールド
startflag

コンストラクタの概要
SetupListFuture(ViewMapDetail view) コンストラクタ

メソッドの概要	
void	**itemStateChanged**(java.awt.event.ItemEvent e) チェックボックスのチェックが変更されたときの処理
void	**setCheckBox**(javax.swing.JPanel panel) 表示するチェックボックス群を生成する

フィールドの詳細

title

public java.lang.String **title**

表示設定タブに表示する文字

---

viewmap

public ViewMapDetail **viewmap**

jp.co.dawn_corp.geosql.j2.GeoSQLView

---

layertable

private jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable **layertable**

道路地物表示用レイヤーテーブル

---

checkbox

private javax.swing.JCheckBox[] **checkbox**

道路地物表示変更用チェックボックス群

コンストラクタの詳細

SetupListFuture

public **SetupListFuture**(ViewMapDetail view)

コンストラクタ

パラメータ:

view - jp.co.dawn_corp.geosql.j2.GeoSQLView

メソッドの詳細

itemStateChanged

public void **itemStateChanged**(java.awt.event.ItemEvent e)

チェックボックスのチェックが変更されたときの処理

定義:

インタフェース java.awt.event.ItemListener 内の itemStateChanged

パラメータ:

e - ItemEvent

---

setCheckBox

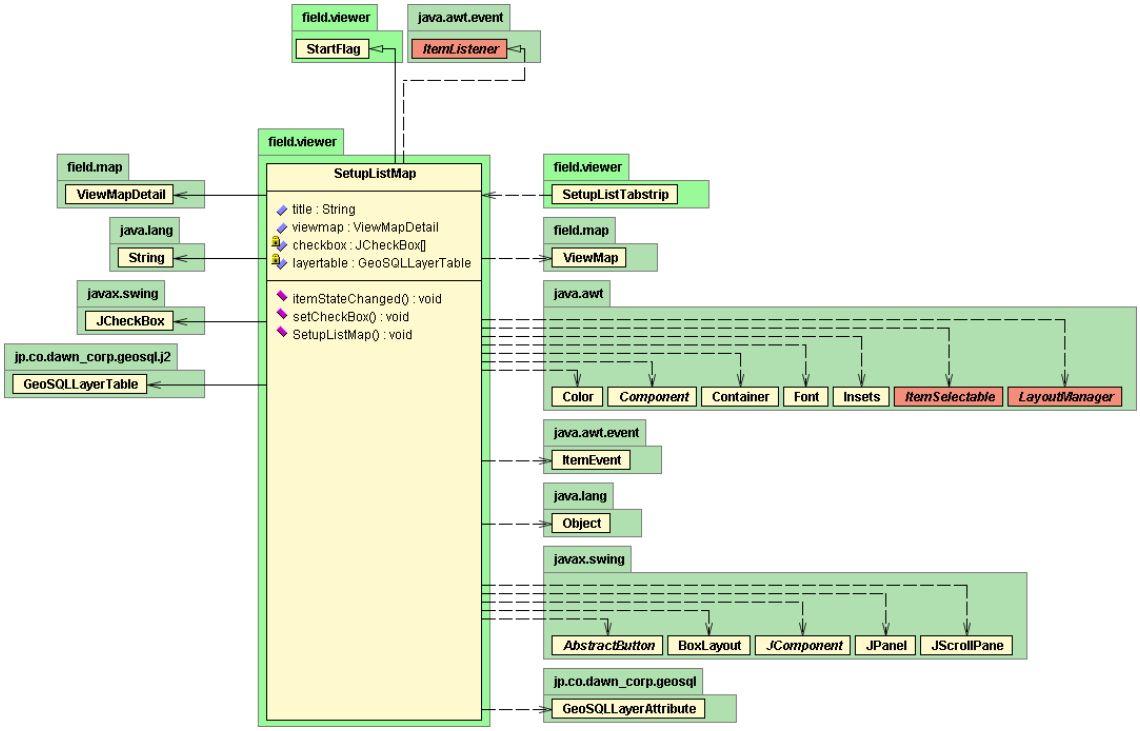
public void **setCheckBox**(javax.swing.JPanel panel)

表示するチェックボックス群を生成する

パラメータ:

panel - JPanel

■ 7　クラス SetupListMap




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.StartFlag
                                |
                                +--field.viewer.SetupListMap
```

---

public class **SetupListMap**

extends StartFlag

implements java.awt.event.ItemListener

表示設定タブ内に背景地図表示設定のチェックボックス群を表示するクラス

フィールドの概要	
private javax.swing.JCheckBox[]	**checkbox** 背景地図表示変更用チェックボックス群
private jp.co.dawn_corp. geosql.j2.GeoSQLLayerTable	**layertable** 背景地図表示用レイヤーテーブル
java.lang.String	**title** 表示設定タブに表示する文字
ViewMapDetail	**viewmap** jp.co.dawn_corp.geosql.j2.GeoSQLView

クラス field.viewer.StartFlag から継承したフィールド
startflag

コンストラクタの概要
SetupListMap(ViewMapDetail view) コンストラクタ

メソッドの概要	
void	**itemStateChanged**(java.awt.event.ItemEvent e) チェックボックスのチェックが変更されたときの処理
void	**setCheckBox**(javax.swing.JPanel panel) 表示するチェックボックス群を生成する

フィールドの詳細

title

public java.lang.String **title**

表示設定タブに表示する文字

---

viewmap

public ViewMapDetail **viewmap**

jp.co.dawn_corp.geosql.j2.GeoSQLView

---

layertable

private jp.co.dawn_corp.geosql.j2.GeoSQLLayerTable **layertable**

背景地図表示用レイヤーテーブル

---

checkbox

private javax.swing.JCheckBox[] **checkbox**

背景地図表示変更用チェックボックス群

コンストラクタの詳細

SetupListMap

public **SetupListMap**(ViewMapDetail view)

コンストラクタ

パラメータ:

view - jp.co.dawn_corp.geosql.j2.GeoSQLView

メソッドの詳細

itemStateChanged

public void **itemStateChanged**(java.awt.event.ItemEvent e)

チェックボックスのチェックが変更されたときの処理

定義:

インタフェース java.awt.event.ItemListener 内の itemStateChanged

パラメータ:

e - ItemEvent

---

setCheckBox

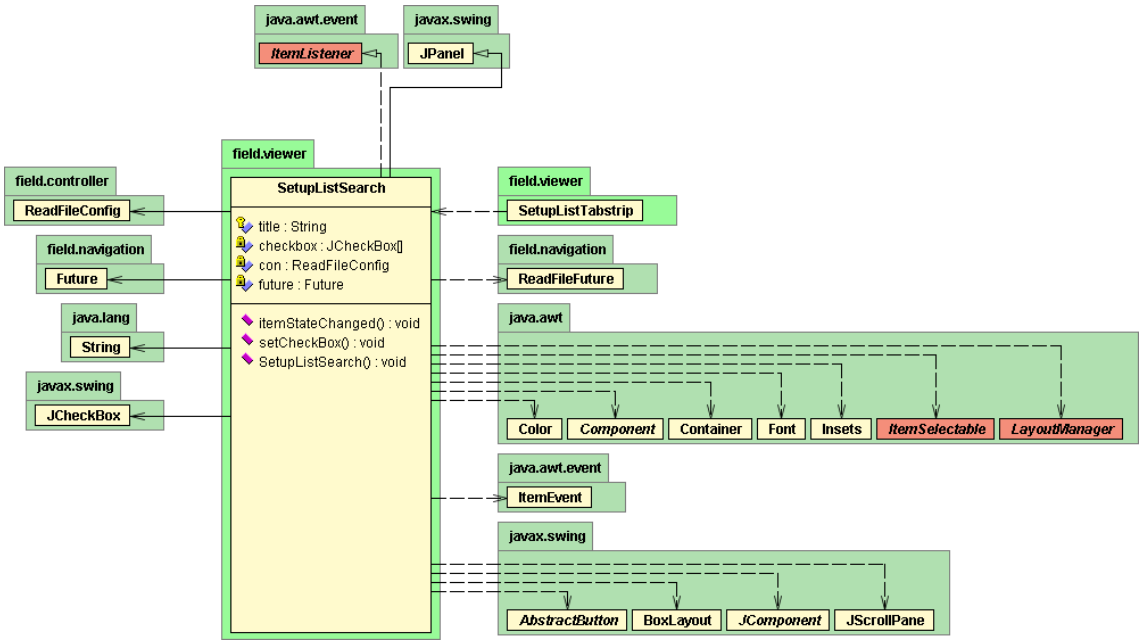
public void **setCheckBox**(javax.swing.JPanel panel)

表示するチェックボックス群を生成する

パラメータ:

panel - JPanel

■ 8　クラス SetupListSearch




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.SetupListSearch
```

---

public class SetupListSearch

extends javax.swing.JPanel

implements java.awt.event.ItemListener

表示設定タブ内に地物検索設定のチェックボックス群を表示するクラス

フィールドの概要	
private	checkbox

javax.swing.JCheckBox[]	背景地図表示変更用チェックボックス群
private ReadFileConfig	**con** 設定ファイルクラス
private Future	**future** 地物検索クラス
protected java.lang.String	**title** 表示設定タブに表示する文字

コンストラクタの概要
SetupListSearch(ReadFileConfig config, Future fut) コンストラクタ

メソッドの概要	
void	**itemStateChanged**(java.awt.event.ItemEvent e) チェックボックスのチェックが変更されたときの処理
void	**setCheckBox**(javax.swing.JPanel panel) 表示するチェックボックス群を生成する

フィールドの詳細

title

protected java.lang.String **title**

表示設定タブに表示する文字

---

con

private ReadFileConfig **con**

設定ファイルクラス

checkbox

private javax.swing.JCheckBox[] **checkbox**

背景地図表示変更用チェックボックス群

---

future

private Future **future**

地物検索クラス

コンストラクタの詳細

SetupListSearch

public **SetupListSearch**(ReadFileConfig config,
Future fut)

コンストラクタ

パラメータ:

view - jp.co.dawn_corp.geosql.j2.GeoSQLView

メソッドの詳細

itemStateChanged

public void **itemStateChanged**(java.awt.event.ItemEvent e)

チェックボックスのチェックが変更されたときの処理

定義:

インタフェース java.awt.event.ItemListener 内の itemStateChanged

パラメータ:

e - ItemEvent

---

setCheckBox

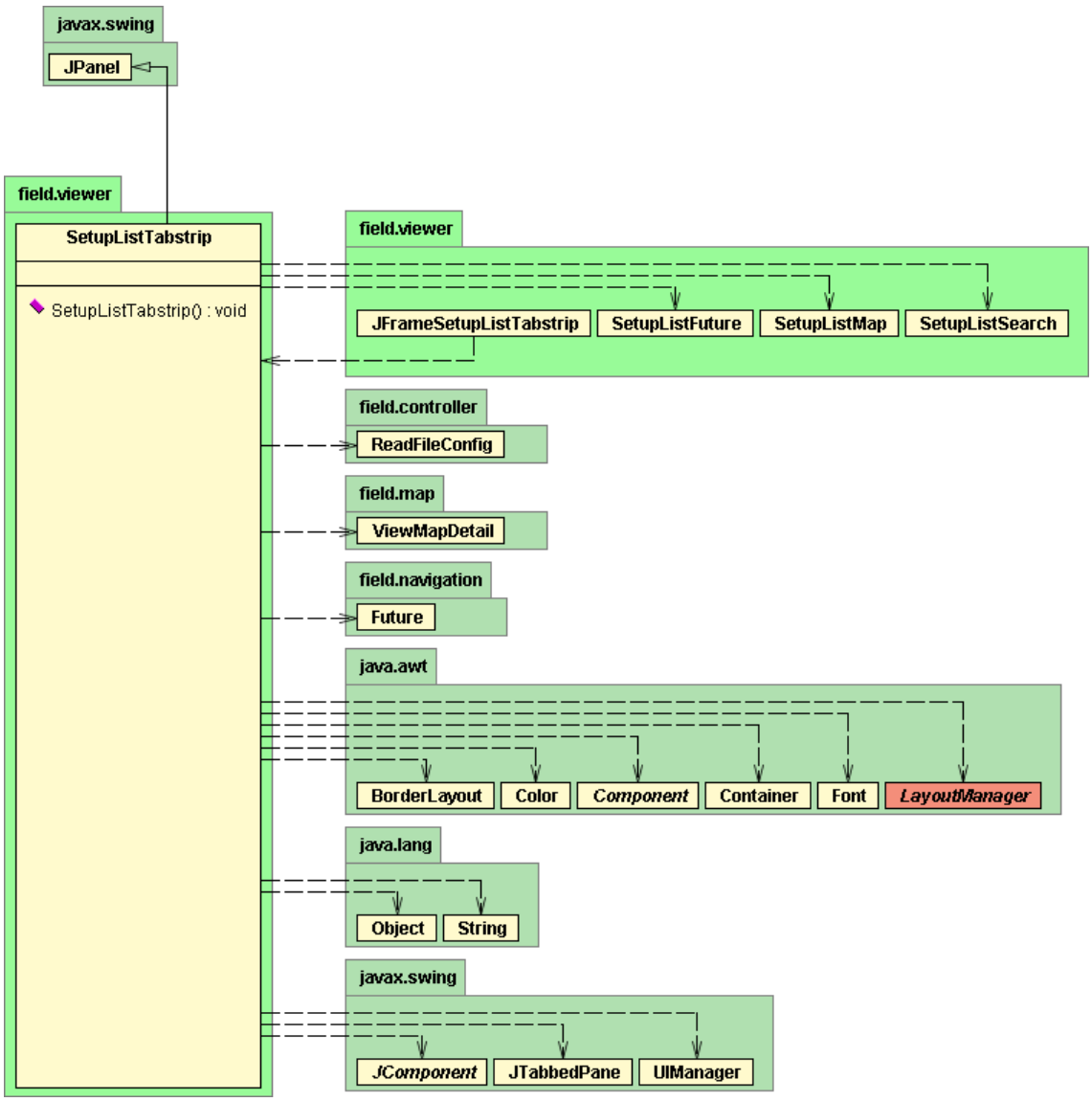
public void **setCheckBox**(javax.swing.JPanel panel)

表示するチェックボックス群を生成する

パラメータ:

panel - JPanel

■ ９　クラス SetupListTabstrip




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
```

```
+--javax.swing.JPanel
      |
      +--field.viewer.SetupListTabstrip
```

---

public class SetupListTabstrip
extends javax.swing.JPanel

表示設定画面のタブ画面を生成するクラス

コンストラクタの概要
SetupListTabstrip(ViewMapDetail viewmap, ReadFileConfig config, Future future) コンストラクタ

コンストラクタの詳細

SetupListTabstrip

public **SetupListTabstrip**(ViewMapDetail viewmap,
ReadFileConfig config,
Future future)

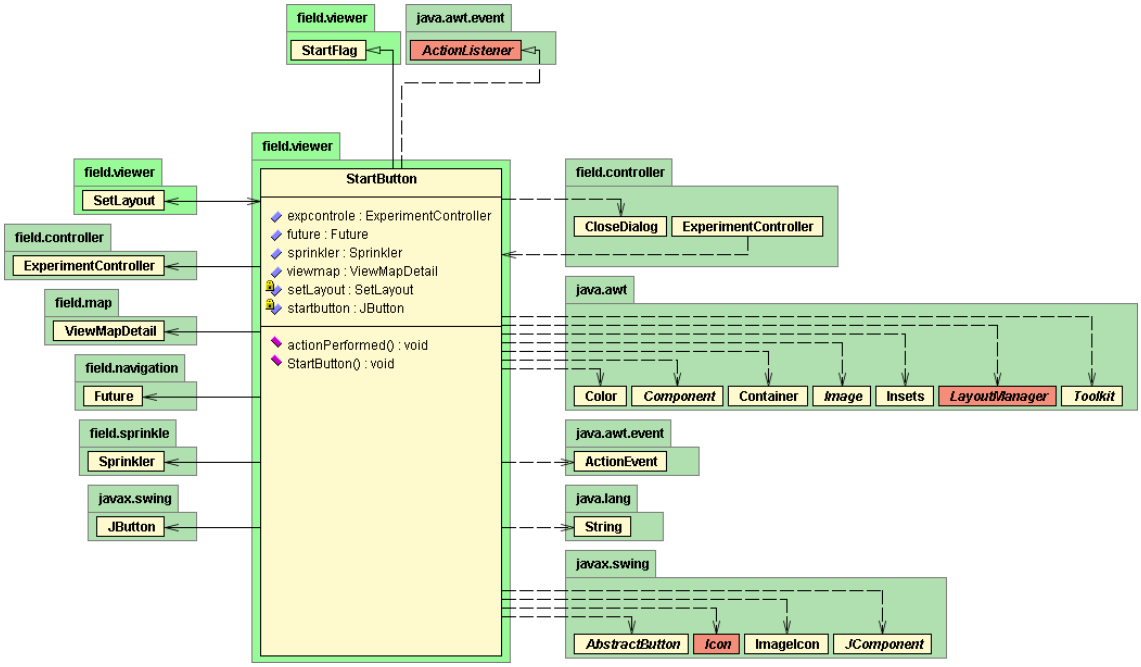
コンストラクタ

パラメータ:

view - jp.co.dawn_corp.geosql.j2.GeoSQLView

config - ReadFileConfig

■ １０　クラス StartButton




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.StartFlag
                                |
                                +--field.viewer.StartButton
```

---

public class **StartButton**

extends StartFlag

implements java.awt.event.ActionListener

画面上に「作業開始」「作業終了」ボタンを表示、コントロールするクラス。

フィールドの概要	
ExperimentController	**expcontrole** メインコントローラ システム終了「はい」の時の再描画用
Future	**future** Future クラスのインスタンス システム終了「いいえ」の時の再描画用
private SetLayout	**setLayout** ナビゲーション画面の各パネルの配置を行うクラスのインスタンス
Sprinkler	**sprinkler** Sprinkler クラスのインスタンス システム終了「いいえ」の時の再描画用
private javax.swing.JButton	**startbutton** ボタンクラス
ViewMapDetail	**viewmap** jp.co.dawn_corp.geosql.j2.GeoSQLView

クラス field.viewer.StartFlag から継承したフィールド
startflag

コンストラクタの概要
StartButton(SetLayout parSetLayout) コンストラクタ

メソッドの概要	
void	**actionPerformed**(java.awt.event.ActionEvent e) スタートボタンの判定および処理を行う

フィールドの詳細

startbutton

private javax.swing.JButton **startbutton**

ボタンクラス

---

expcontrole

public ExperimentController **expcontrole**

メインコントローラ システム終了「はい」の時の再描画用

---

future

public Future **future**

Future クラスのインスタンス システム終了「いいえ」の時の再描画用

---

sprinkler

public Sprinkler **sprinkler**

Sprinkler クラスのインスタンス システム終了「いいえ」の時の再描画用

---

viewmap

public ViewMapDetail **viewmap**

jp.co.dawn_corp.geosql.j2.GeoSQLView

---

setLayout

private SetLayout **setLayout**

ナビゲーション画面の各パネルの配置を行うクラスのインスタンス

コンストラクタの詳細

StartButton

public **StartButton**(SetLayout parSetLayout)

コンストラクタ

パラメータ:

parSetLayout – SetLayout

メソッドの詳細

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

スタートボタンの判定および処理を行う

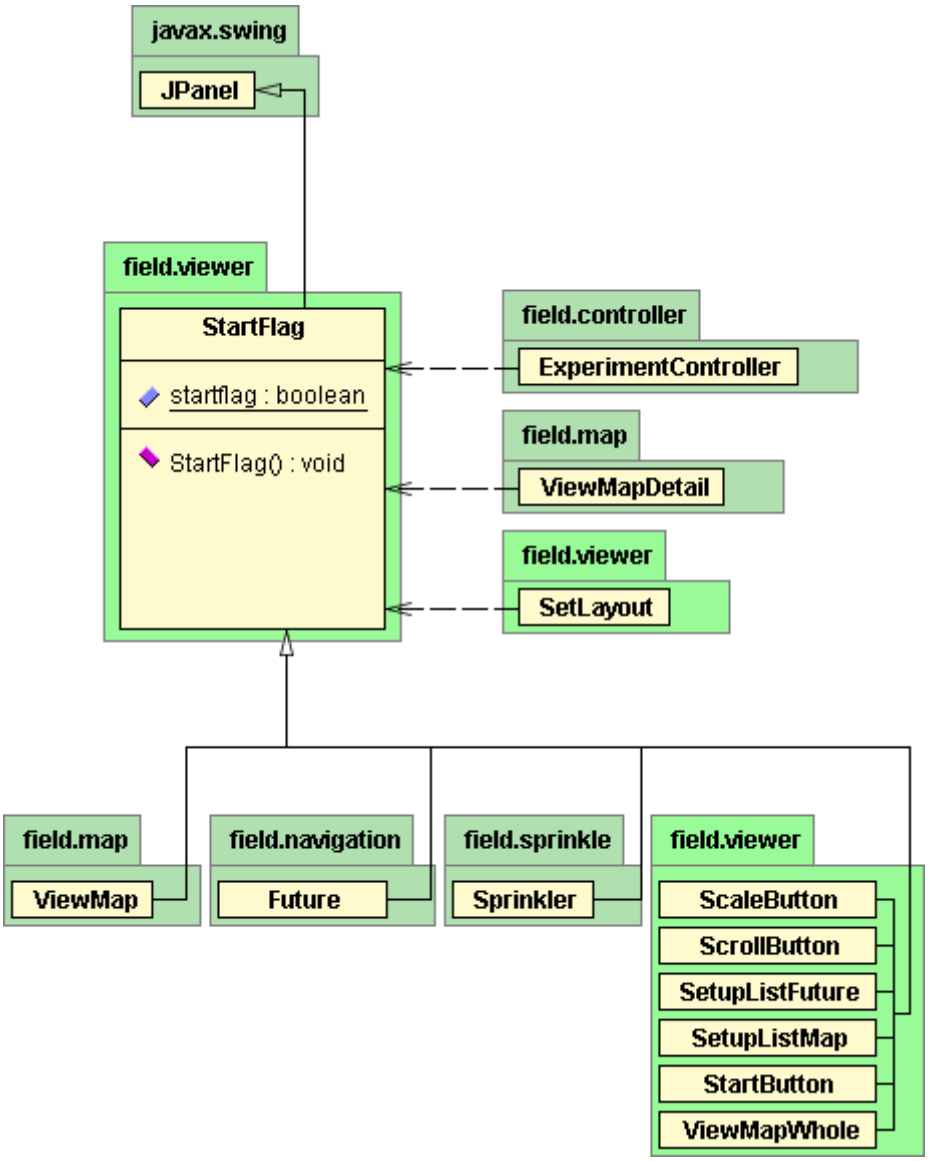
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e - ActionEvent

■ １１　クラス StartFlag




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.StartFlag
```

直系の既知のサブクラス:

Future, ScaleButton, ScrollButton, SetupListFuture, SetupListMap, Sprinkler, StartButton, ViewMap, ViewMapWhole

---

public class **StartFlag**

extends javax.swing.JPanel

開始・終了ステータスの共有化クラス。このクラスを継承することによって、
開始ボタンが押されてるかの判断を行う。 startflag = true ---> 開始状態 startflag = false --> 停止状態

フィールドの概要	
static boolean	**startflag** 開始終了状態を持つフラグ

コンストラクタの概要
StartFlag() コンストラクタ

フィールドの詳細

startflag

public static boolean **startflag**

開始終了状態を持つフラグ

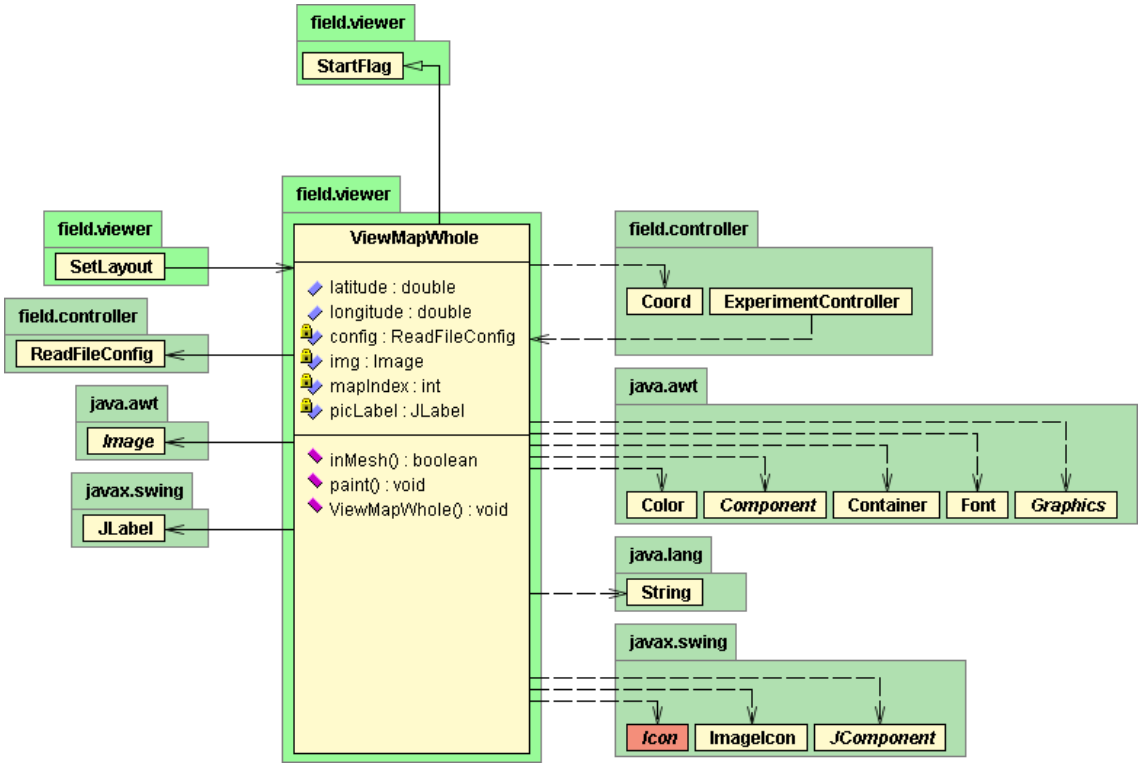
コンストラクタの詳細

StartFlag

public **StartFlag**()

コンストラクタ

■ １２　クラス ViewMapWhole




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--field.viewer.StartFlag
                                |
                                +--field.viewer.ViewMapWhole
```

---

public class **ViewMapWhole**

extends StartFlag

画面に全体地図を表示するクラス

画像は 280x180 ピクセルで作成

フィールドの概要	
private ReadFileConfig	**config** 設定ファイル読込クラス
private java.awt.Image	**img** 全体地図として表示するイメージ（イメージ）
double	**latitude** 緯度
double	**longitude** 経度
private int	**mapIndex** 全体地図として表示するイメージ（画像番号）
private javax.swing.JLabel	**picLabel** 全体地図として表示するイメージ（ラベル）

クラス field.viewer.StartFlag から継承したフィールド
startflag

コンストラクタの概要
ViewMapWhole(ReadFileConfig con) コンストラクタ

メソッドの概要	
boolean	**inMesh**(double xPos, double yPos, double x1, double y1, double x2, double y2)

	与えられた点がメッシュ内にあるか判定
void	**paint**(java.awt.Graphics g) 全体地図中に現在位置を表示する。

フィールドの詳細

latitude

public double **latitude**

緯度

---

longitude

public double **longitude**

経度

---

picLabel

private javax.swing.JLabel **picLabel**

全体地図として表示するイメージ（ラベル）

---

img

private java.awt.Image **img**

全体地図として表示するイメージ（イメージ）

---

mapIndex

private int **mapIndex**

全体地図として表示するイメージ（画像番号）

---

config

private ReadFileConfig **config**

設定ファイル読込クラス

コンストラクタの詳細

ViewMapWhole

public **ViewMapWhole**(ReadFileConfig con)

コンストラクタ

パラメータ:

config - 設定ファイル情報クラス

メソッドの詳細

paint

public void **paint**(java.awt.Graphics g)

全体地図中に現在位置を表示する。

オーバーライド:

クラス javax.swing.JComponent 内の paint

パラメータ:

g - java.awt.Graphics

---

inMesh

public boolean **inMesh**(double xPos,
double yPos,
double x1,
double y1,
double x2,
double y2)

与えられた点がメッシュ内にあるか判定

パラメータ:

xPos - 与えられた点の X 座標

yPos - 与えられた点の Y 座標

x1 - 地図の右下 x 座標

y1 - 地図の右下 y 座標

x2 - 地図の左上 x 座標

y2 - 地図の左上 y 座標

戻り値:

メッシュ内:true メッシュ外:false

2.4. 操作支援システムの製作

製作した本システムのソースコードを別添資料に添付する。

2.5. 操作支援システムの試験（単体）

(1) 試験概要

本試験成績書は、操作支援システムについての単体試験結果を示すものである。単体試験方針としては、地物検索機能および自動散布制御機能の動作確認を中心として同時に画面表示、音声案内についての動作確認を行うものとする。動作確認は、位置特定パッケージが連結されていないため、仮想路線メッシュを作成しその仮想路線上を走行する仮想走行データをデータファイルから読み込むことによって、擬似的に地図上を走行する動作を実現することにより実施した。また、ホワイトボックステストでは全体的な検索や制御の確認が不可能なため、機能部分のテストを重点においたブラックボックステストを行うこととした。

(2) 試験結果

a） 散布制御

散布制御には、2 種類のテストメッシュを作成し、走行データ、散布計画データを変化させて、テストを実施した。単体テストに使用した車線メッシュ、試験パターンおよび試験結果を以下に示す。

なお、試験結果の結果欄に１回目、２回目、３回目とあるが、１回目でテストが失敗した場合にプログラムの修正を行い２回目、３回目において再度テストを行った状況を示すものである。

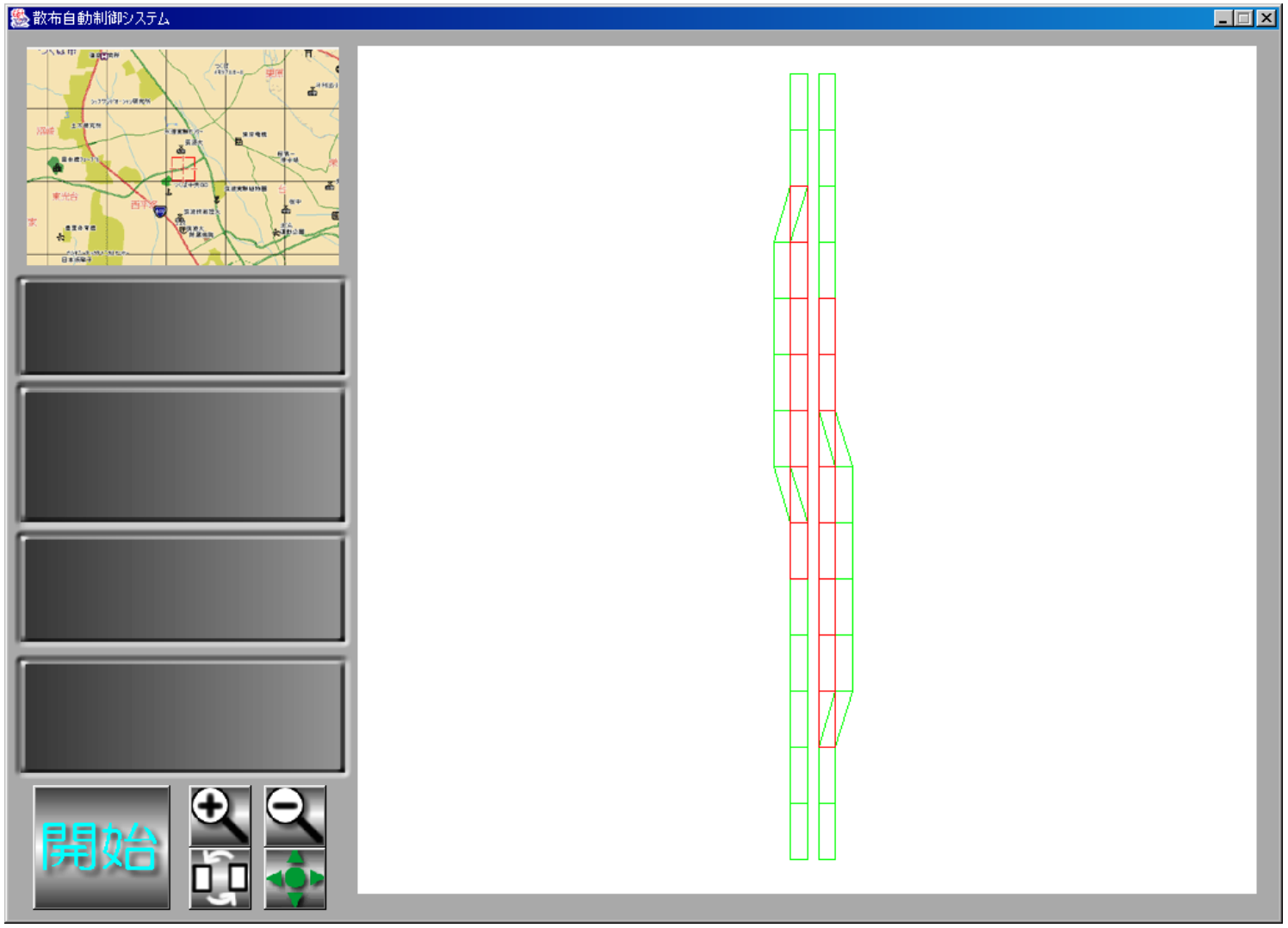


図 2-64　散布制御テスト用メッシュ 1

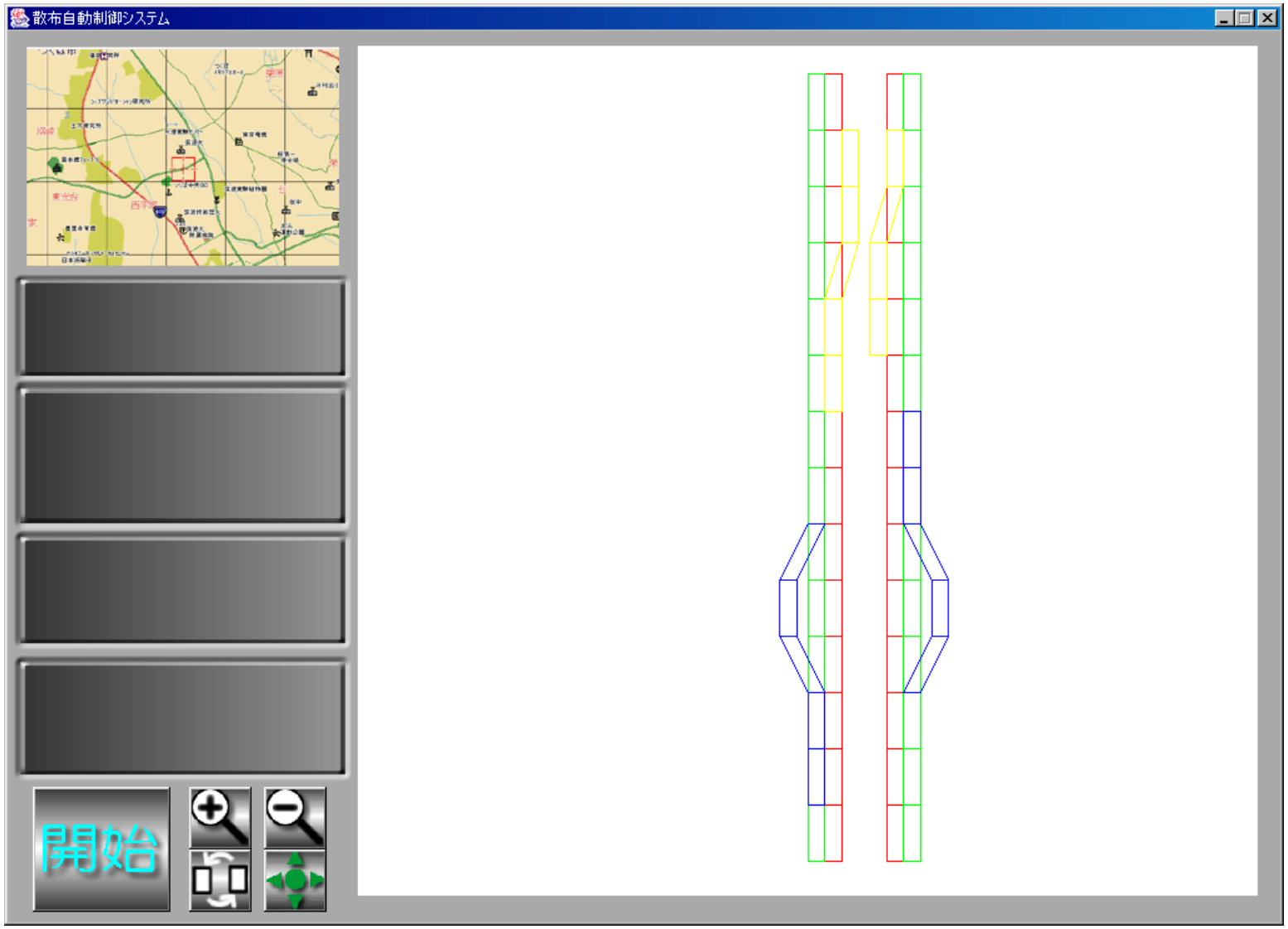


図 2-65　散布制御テスト用メッシュ２

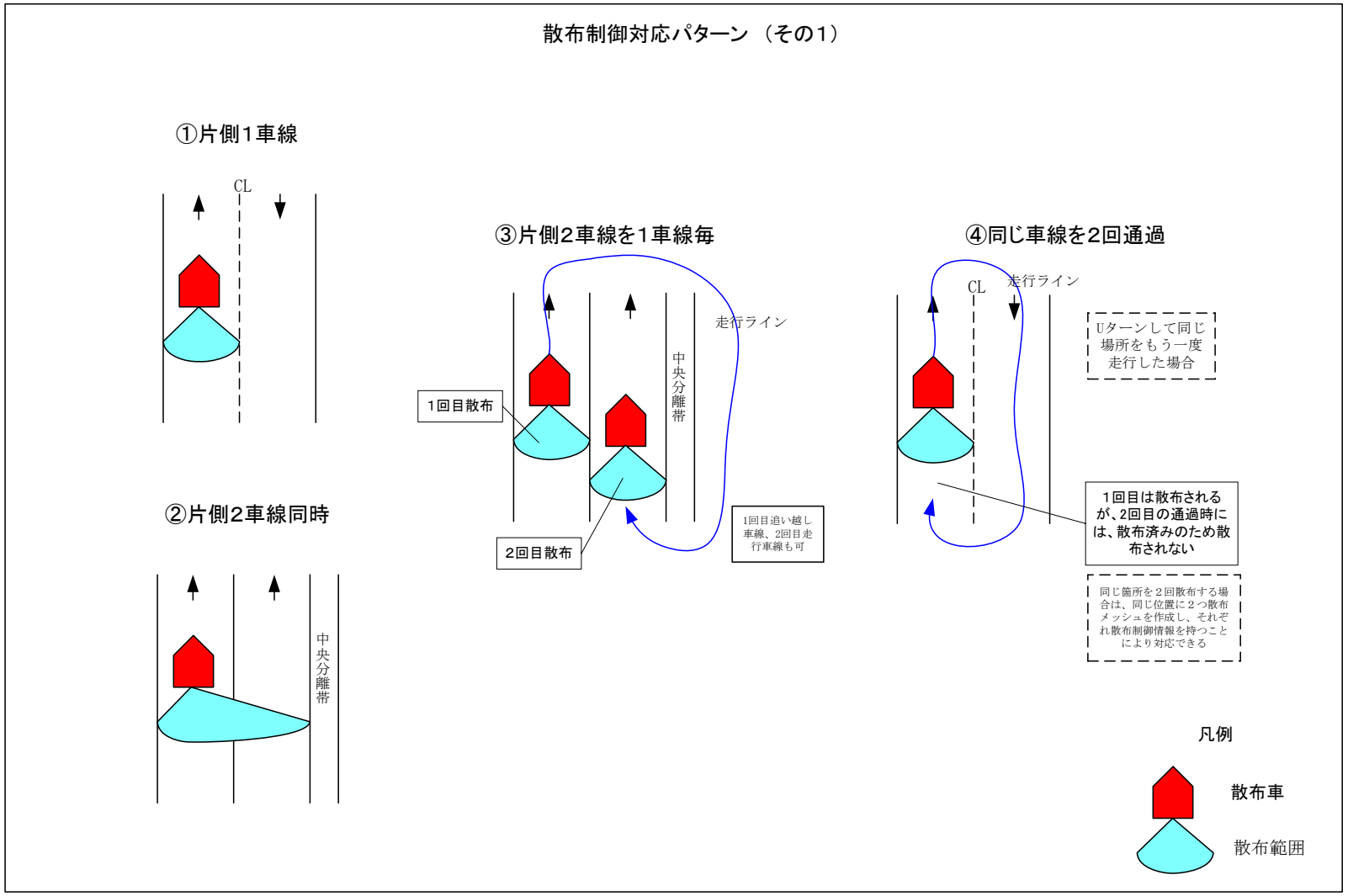


図 2-66　試験パターン（その１）

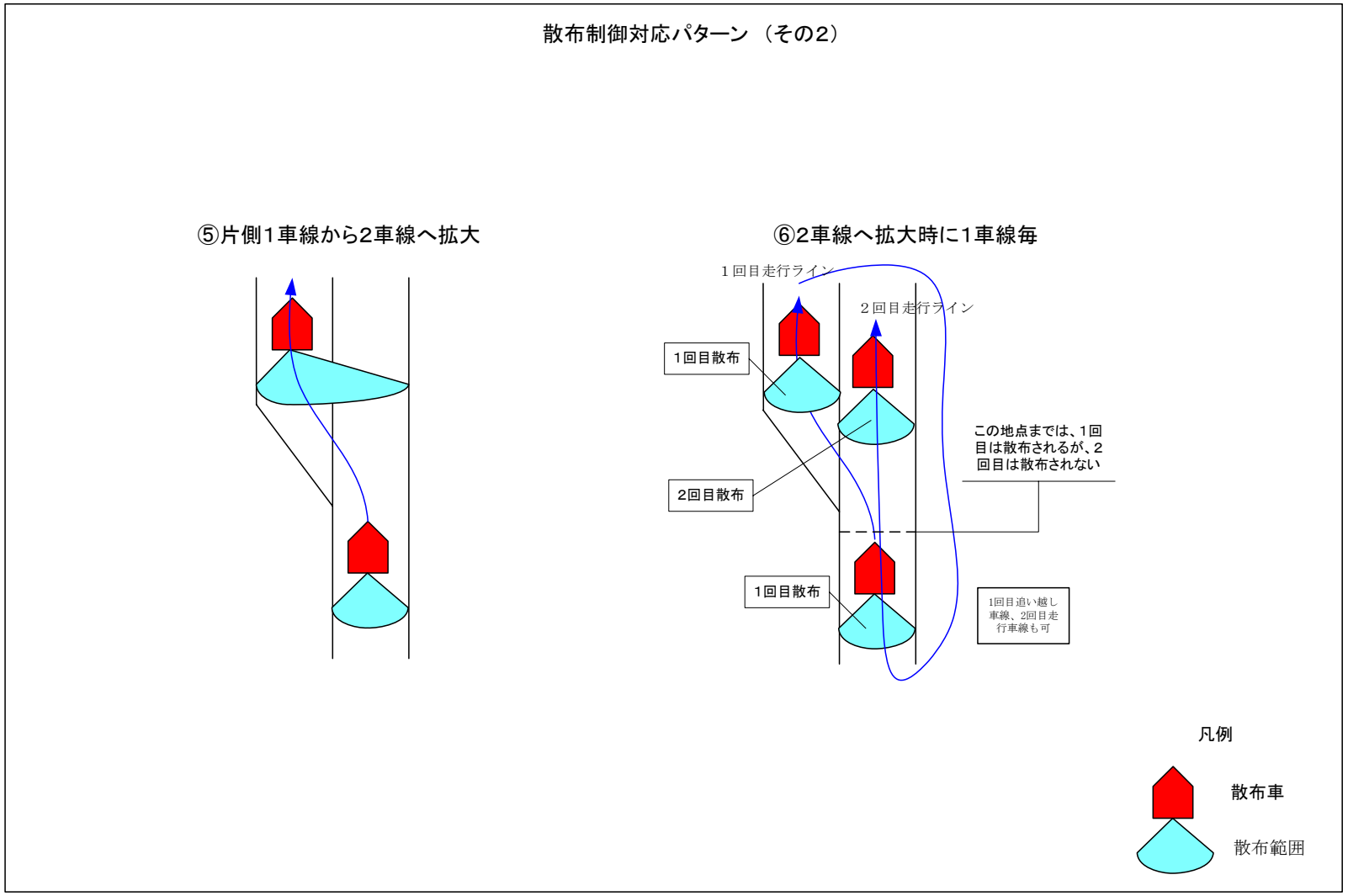


図 2-67　試験パターン（その２）

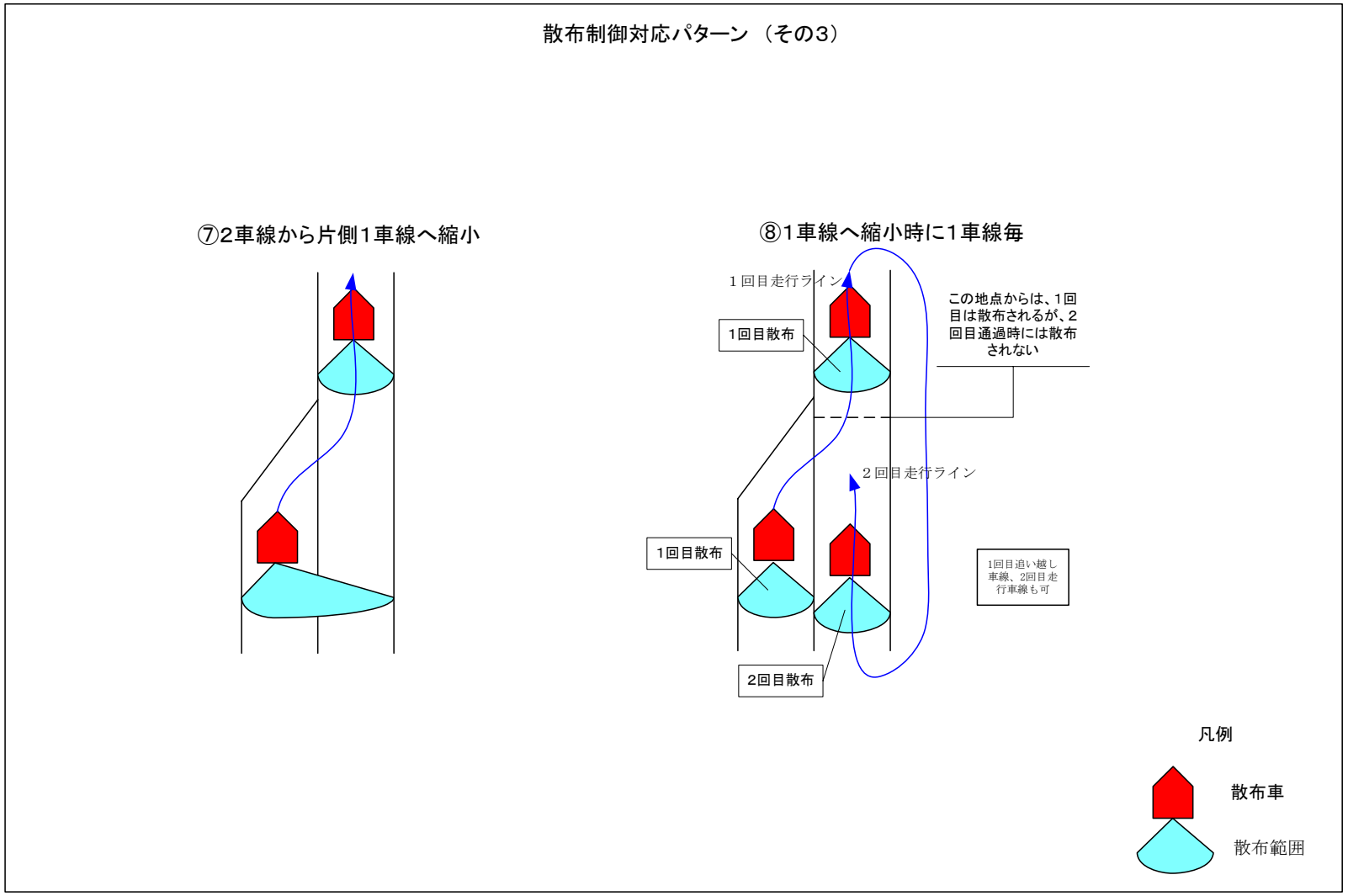


図 2-68　試験パターン（その３）

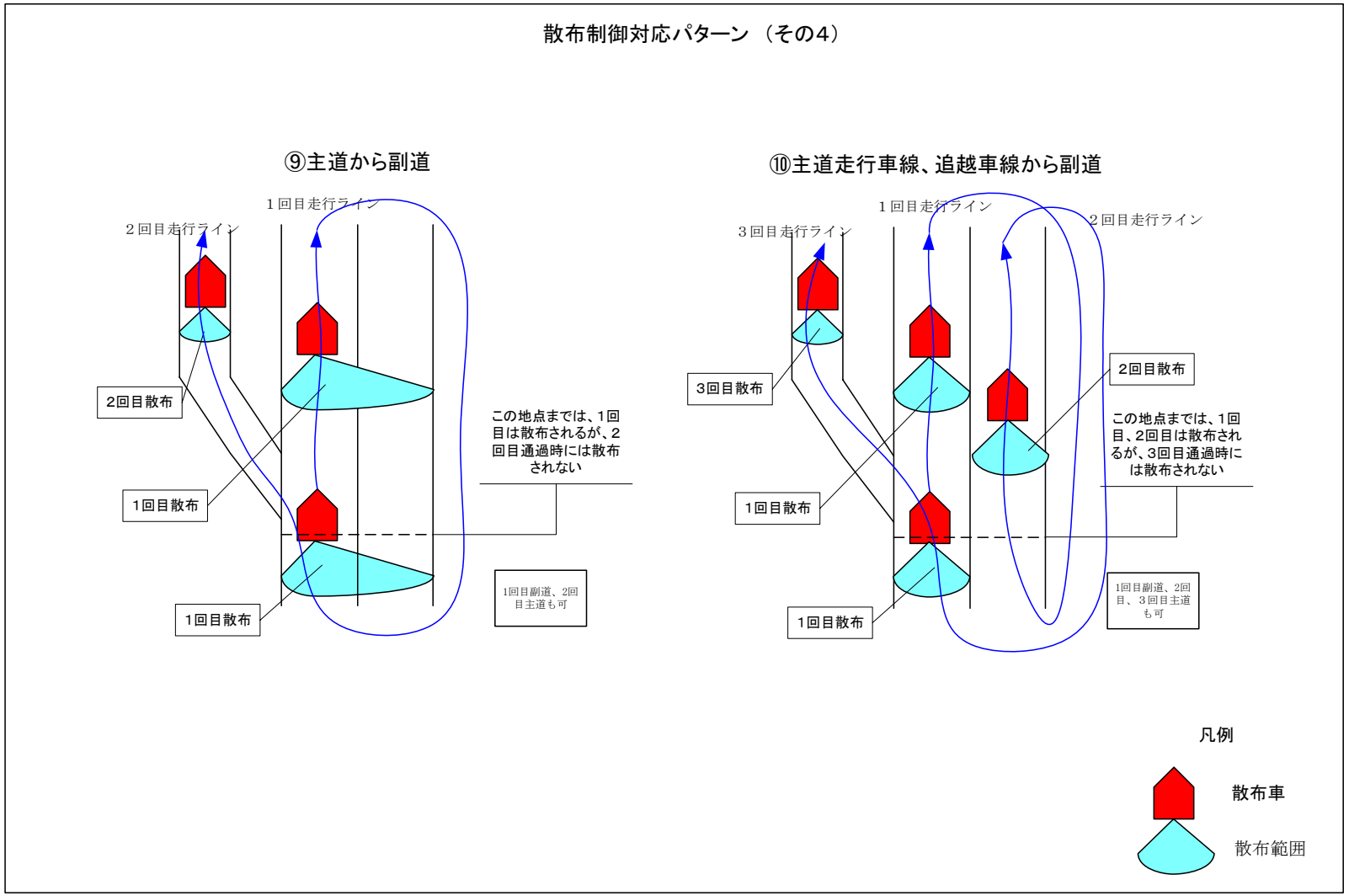


図 2-69　試験パターン（その４）

表 2-57　散布制御単体試験結果（その１）

走行パターン	周回	上／下	テスト内容	結果			備　　考
				１回目	２回目	３回目	
片側１車線	１周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
	２周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目も散布する	○	○	○	
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
			２周目も散布する	○	○	○	
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた

表 2-58　散布制御単体試験結果（その２）

走行パターン	周回	上／下	テスト内容	結果			備　　考
				１回目	２回目	３回目	
２車線へ拡大時に同時	１周	上り	散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
			２車線へ拡大してから散布する	○	○	○	
		下り	散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
			２車線へ拡大してから散布する	○	○	○	
	２周	上り	２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
		下り	２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた

表 2-59　散布制御単体試験結果（その３）

走行パターン	周回	上／下	テスト内容	結果			備　　考
				１回目	２回目	３回目	
２車線へ拡大時に１車線毎	１周	上り	散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
			２車線へ拡大してから散布する	○	○	○	
		下り	散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
			２車線へ拡大してから散布する	○	○	○	
	２周	上り	散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
			２車線へ拡大してから散布する	○	○	○	
			１周目は散布しないで２周目の１車線のみ散布する	○	○	○	
		下り	散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
			２車線へ拡大してから散布する	○	○	○	
			１周目は散布しないで２周目の１車線のみ散布する	○	○	○	

表 2-60　散布制御単体試験結果（その４）

走行パターン	周回	上／下	テスト内容	結果			備　　考
				１回目	２回目	３回目	
片側２車線同時	１周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
	２周	上り	２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
		下り	２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた

表 2-61　散布制御単体試験結果（その５）

走行パターン	周回	上／下	テスト内容	結果			備　　考
				１回目	２回目	３回目	
片側２車線を１車線毎	１周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
	２周	上り	散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
			１周目は散布しないで２周目の１車線のみ散布する	○	○	○	
		下り	散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
			１周目は散布しないで２周目の１車線のみ散布する	○	○	○	

表 2-62　散布制御単体試験結果（その６）

走行パターン	周回	上／下	テスト内容	結果			備　　考
				１回目	２回目	３回目	
２車線同時から１車線へ縮小	１周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
	２周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた

表 2-63　散布制御単体試験結果（その７）

走行パターン	周回	上／下	テスト内容	結果			備　　考
				１回目	２回目	３回目	
片側２車線を１車線毎から１車線へ縮小	１周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
	２周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた

表 2-64　散布制御単体試験結果（その８）

走行パターン	周回	上／下	テスト内容	結果			備　　考
				１回目	２回目	３回目	
１周目主道から２周目副道	１周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
	２周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
			１周目の主道は散布しないで２周目の副道のみ散布する	○	○	○	
			副道から主道へ戻った時散布しない	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
			１周目の主道は散布しないで２周目の副道のみ散布する	○	○	○	
			副道から主道へ戻った時散布しない	○	○	○	

表 2-65　散布制御単体試験結果（その９）

走行パターン	周回	上／下	テスト内容	結果			備　　考
				１回目	２回目	３回目	
１周目副道から２周目主道	１周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
			副道から主道へ戻った時その先の主道を散布する	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
			副道から主道へ戻った時その先の主道を散布する	○	○	○	
	２周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
			１周目で副道からの合流先を散布している場合は、２周目は散布しない	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			２周目は走行のみで散布はしない	×	○	○	２周目の走行のみで『散布開始までXXm』と表示されていた
			１周目で副道からの合流先を散布している場合は、２周目は散布しない	○	○	○	

表 2-66　散布制御単体試験結果（その１０）

走行パターン	周回	上／下	テスト内容	結果			備　　考
				１回目	２回目	３回目	
１周目主道走行、２周目追越車線、３周目副道	１周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
			１周目主道走行と３周目副道から合流してきた追越車線を１車線毎に散布する	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	○	○	○	
			１周目主道走行と３周目副道から合流してきた追越車線を１車線毎に散布する	○	○	○	
	２周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
	３周	上り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			３周目は走行のみで散布はしない	×	○	○	３周目の走行のみで『散布開始までXXm』と表示されていた
			１周目の主道は散布しないで３周目の副道のみ散布する	○	○	○	
			副道から主道へ戻った時散布しない	○	○	○	
			１周目主道走行と３周目副道から合流してきた追越車線を１車線毎に散布する	○	○	○	
		下り	散布⇒停止⇒散布を繰り返す	○	○	○	
			散布量一定で散布する	○	○	○	
			各メッシュにて散布量が違う	×	○	○	散布量の表示が変化しなかった
			３周目は走行のみで散布はしない	×	○	○	３周目の走行のみで『散布開始までXXm』と表示されていた
			１周目の主道は散布しないで３周目の副道のみ散布する	○	○	○	
			副道から主道へ戻った時散布しない	○	○	○	
			１周目主道走行と３周目副道から合流してきた追越車線を１車線毎に散布する	×	○	○	散布する箇所が水色に塗りつぶされていなかった

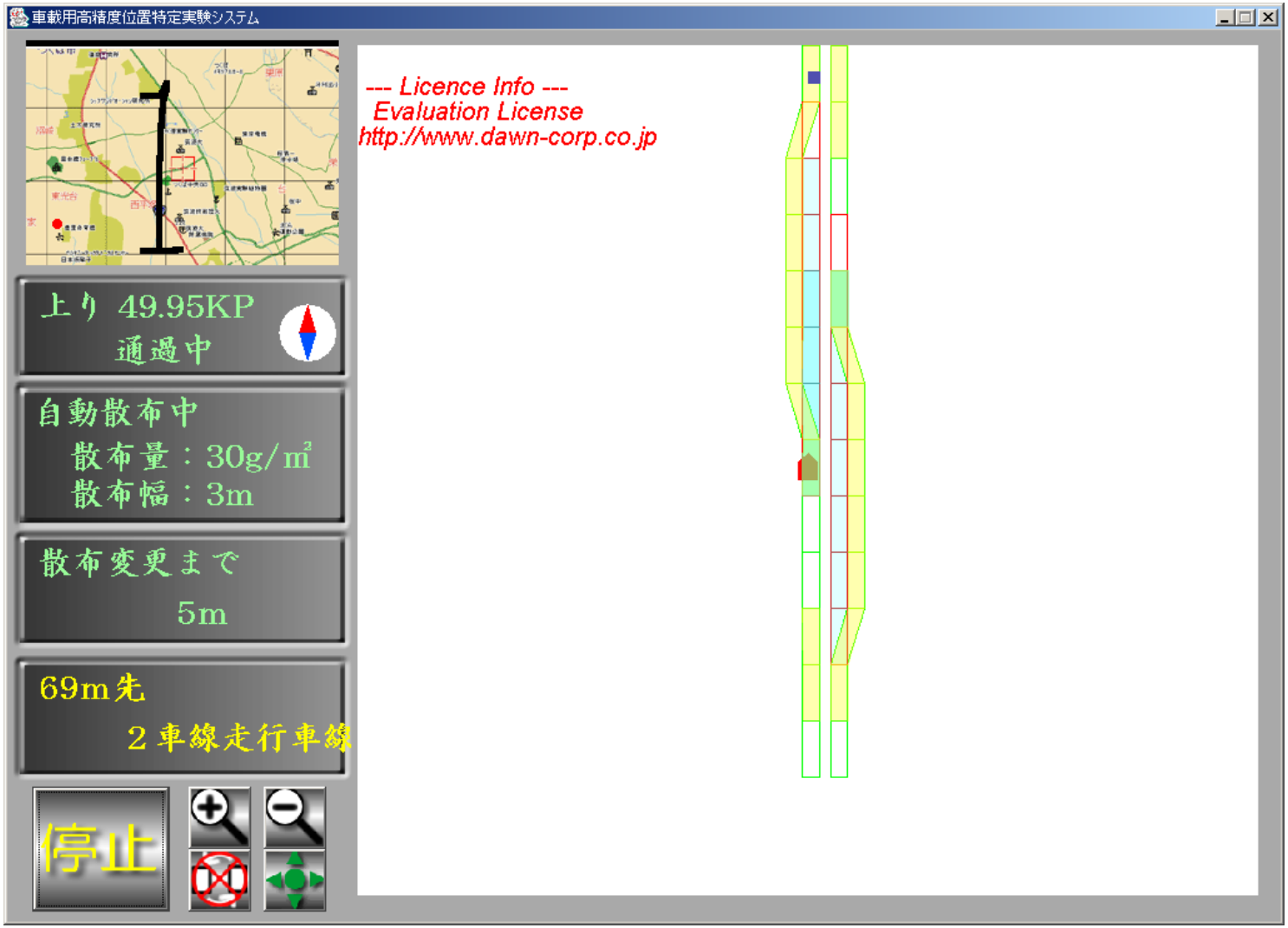


図 2-70　単体試験実施イメージ

b） 地物検索

地物検索には、1 種類のテストメッシュを作成し、走行データ、地物位置を変化させて、テストを実施した。単体テストに使用した車線メッシュ、試験パターンおよび試験結果を以下に示す。

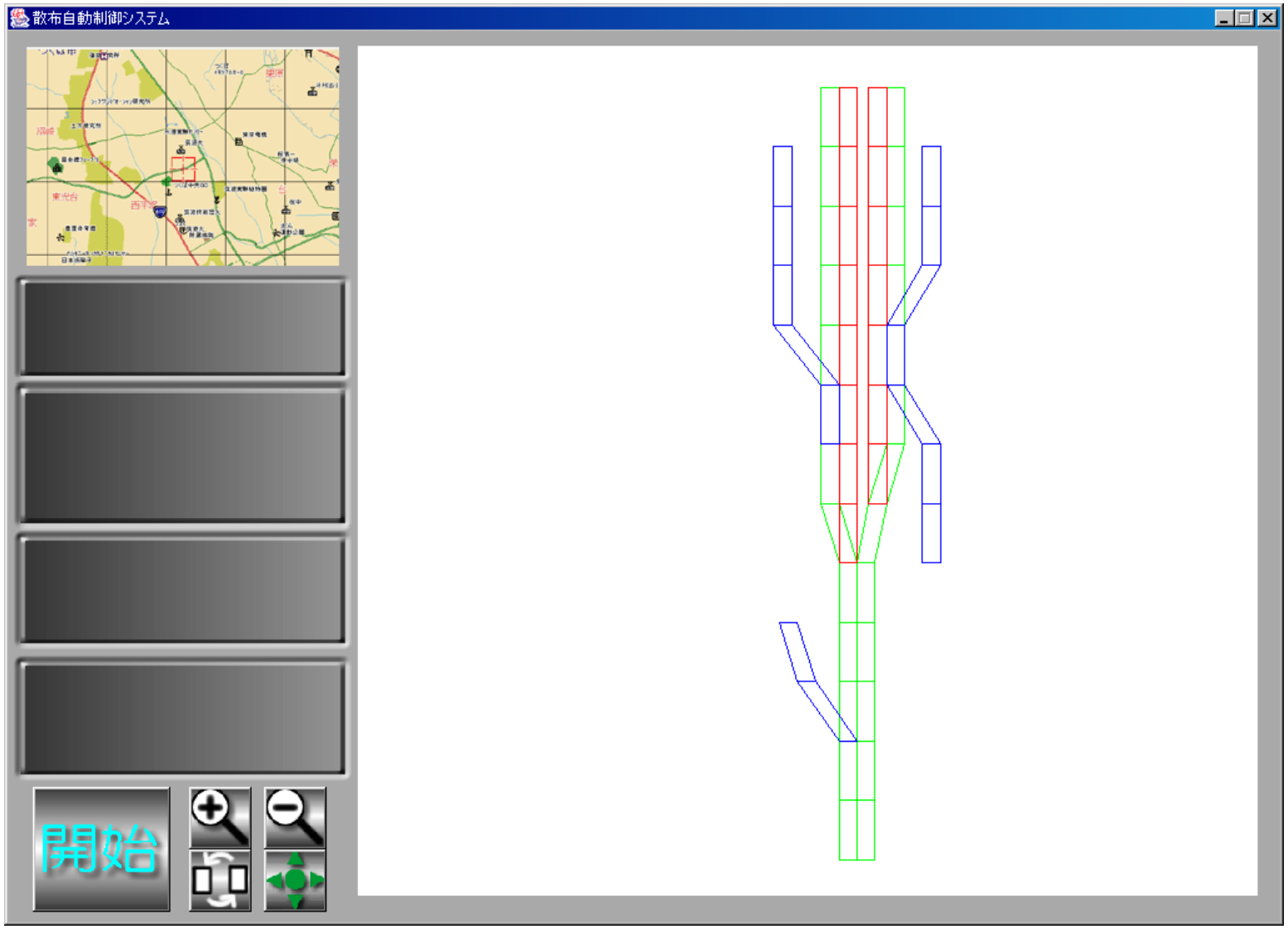


図 2-71　地物検索テスト用メッシュ

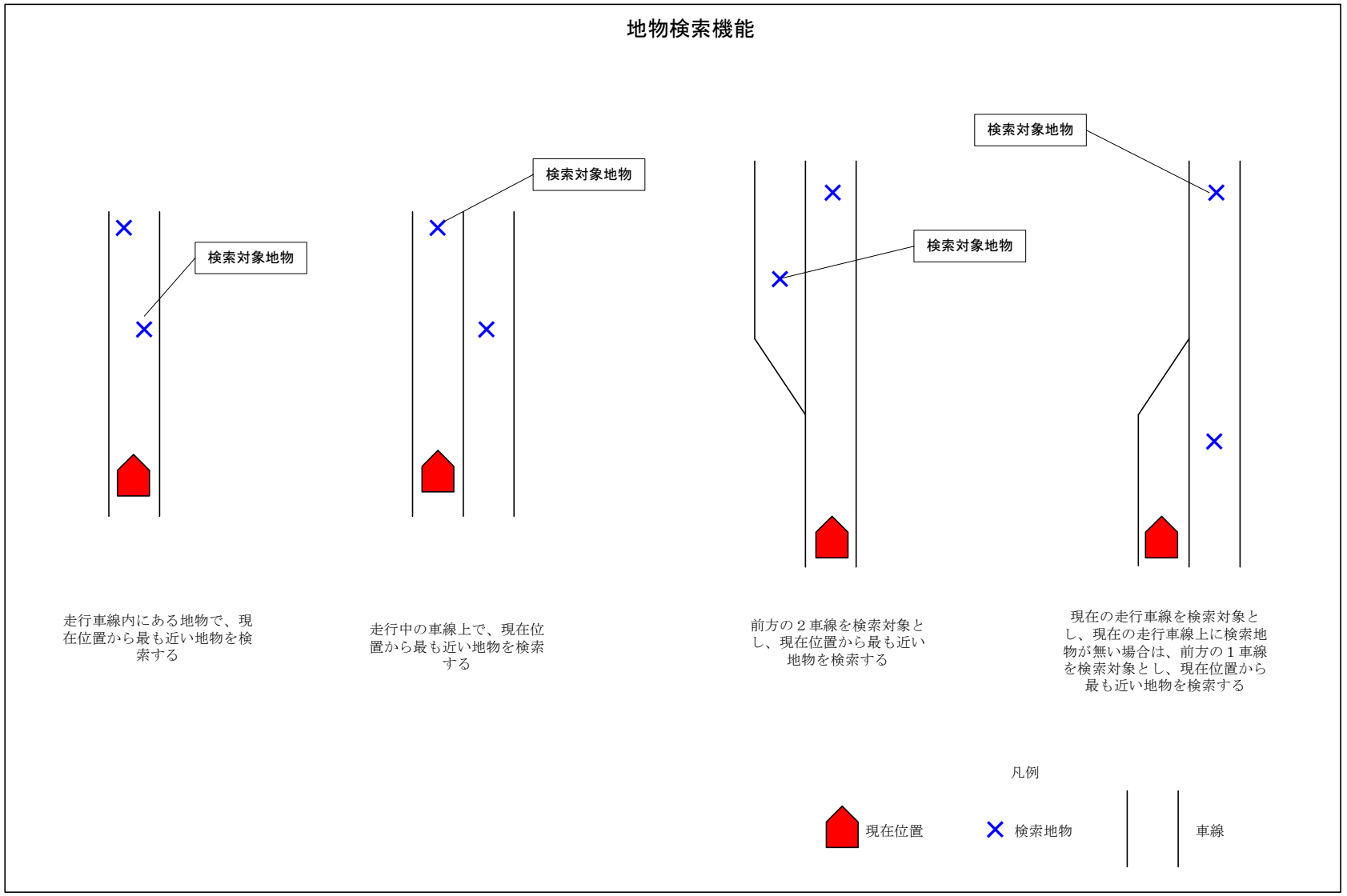


図 2-72　地物検索パターン

表 2-67　地物検索単体試験結果

ケース番号	車線パターン	テスト内容	合否	備　　考
1	１車線を走行中で前方で２車線に分岐	同一メッシュ内の地物が検索	○	
2		前方１車線の地物が検索	○	
3		前方２車線の走行車線の地物が検索	○	
4		前方２車線の追越車線の地物が検索されない	○	
5		前方２車線の走行と追越に地物がある場合に走行車線上の地物が検索	○	
6	２車線を走行	走行車線を走行した場合、走行車線の地物が検索	○	
7		追越車線を走行した場合、追越車線の地物が検索	○	
8	副道を走行	副道上の地物が検索	○	上り車線の確認
9	走行車線と副道の２重メッシュ区間を走行	走行車線と副道の２つの地物が検索され、追越車線の地物が検索されない	○	
10		副道の地物が検索され、追越車線の地物が検索されない	○	
11		走行車線の地物が検索され、追越車線の地物が検索されない	○	
12	２車線を走行中で前方で１車線に合流	走行車線を走行した場合、走行車線の地物が検索	○	
13		走行車線を走行した場合、１車線の地物が検索され、追越車線上の地物が検索されない	○	
14		走行車線を走行した場合、１車線の地物が検索	○	
15		追越車線を走行した場合、追越車線の地物が検索され、走行車線の地物が検索されない	○	
16		追越車線を走行した場合、追越車線の地物が検索され、１車線の地物が検索されない	○	
17		追越車線を走行した場合、１車線の地物が検索	○	
18	副道を走行（前方で１車線に合流）	副道上の最も近い地物が検索	○	
19		副道上の地物が検索	○	下り車線の確認
20		地物が検索されない	○	
21		走行車線と副道の２つの地物が検索	○	
22		前方１車線の地物と副道の地物が検索	○	
23		副道の地物が検索	○	
24		走行車線の地物が検索	○	
25		前方１車線の地物が検索	○	
26	１車線（同一メッシュ内に２つの地物）	近い地物が検索	○	
27		走行位置より後ろの地物が検索されない	○	
28	走行車線と副道の２重メッシュ区間が前方にあり	地物検索の表示が１つになる（同じ地物が２つ表示されない）	○	

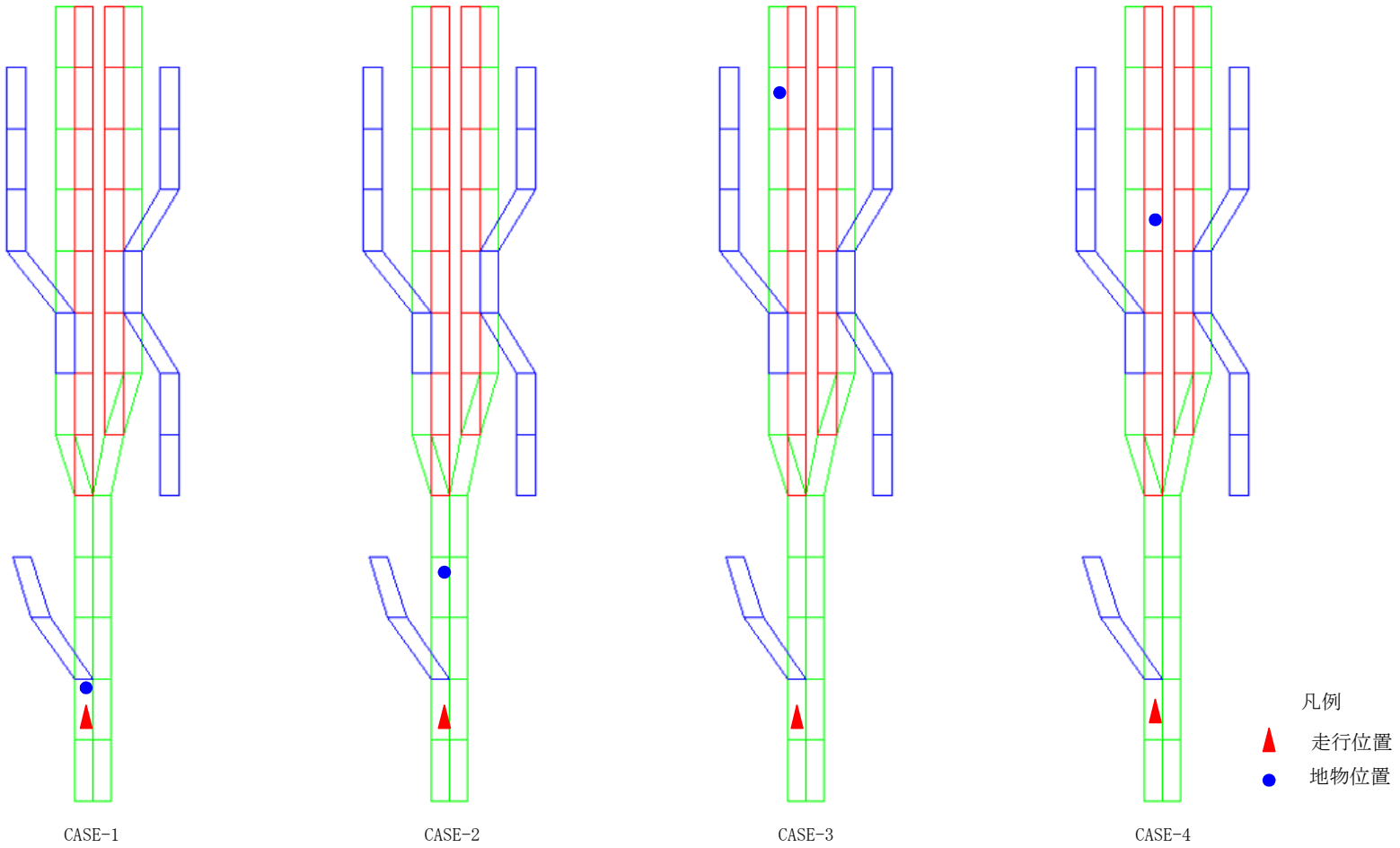


図 2-73　地物検索ケース図（その１）

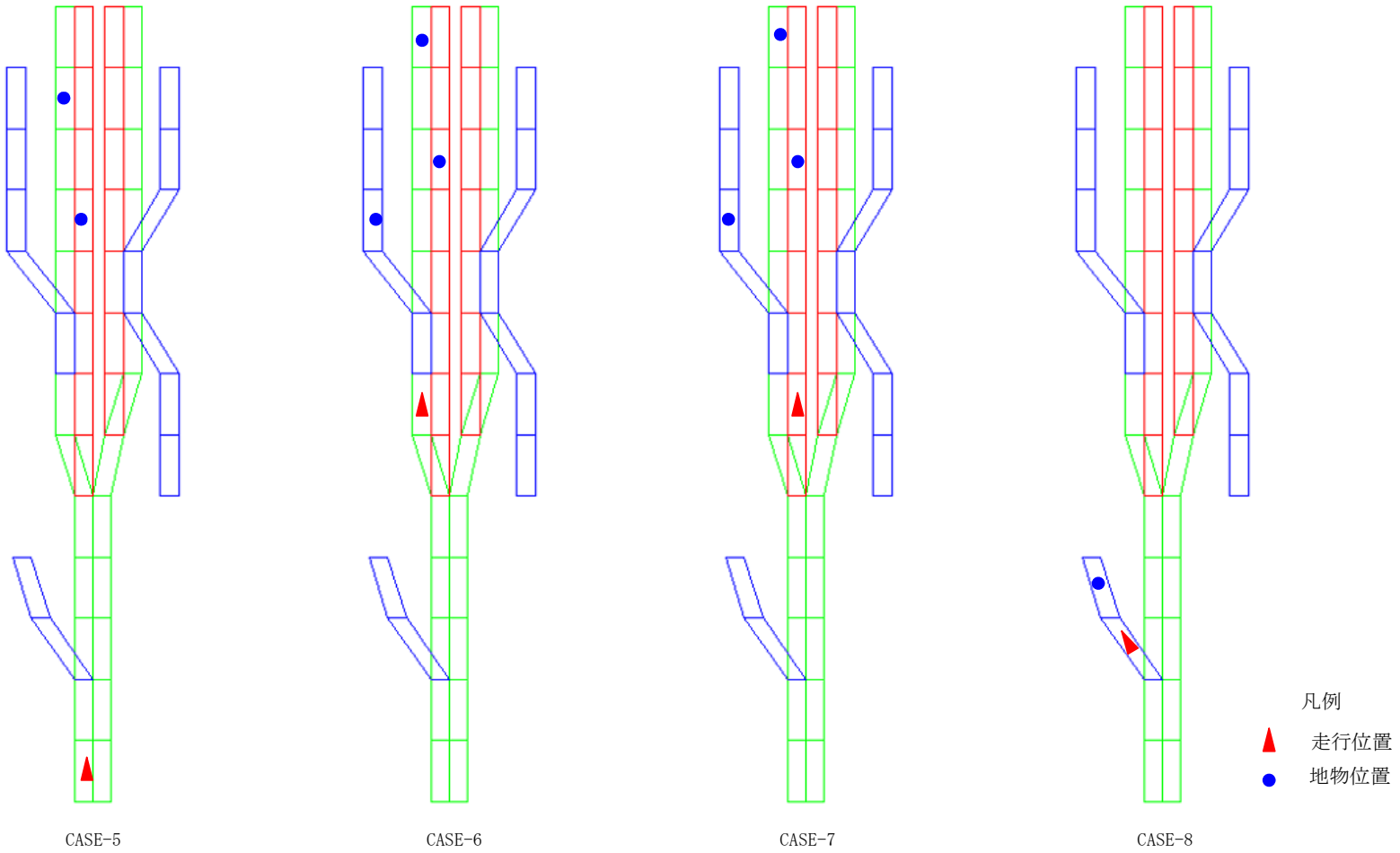


図 2-74　地物検索ケース図（その２）

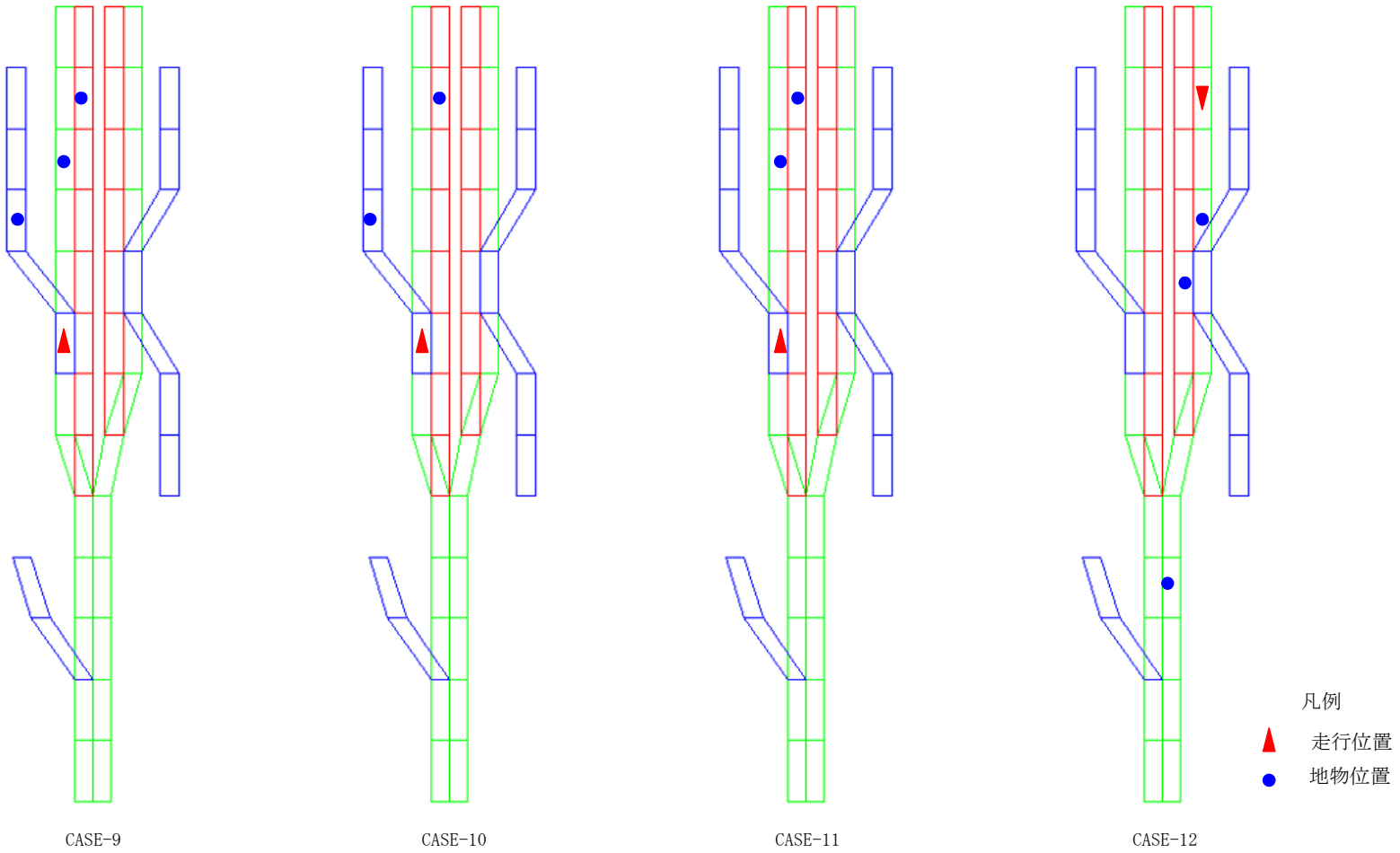


図 2-75　地物検索ケース図（その３）

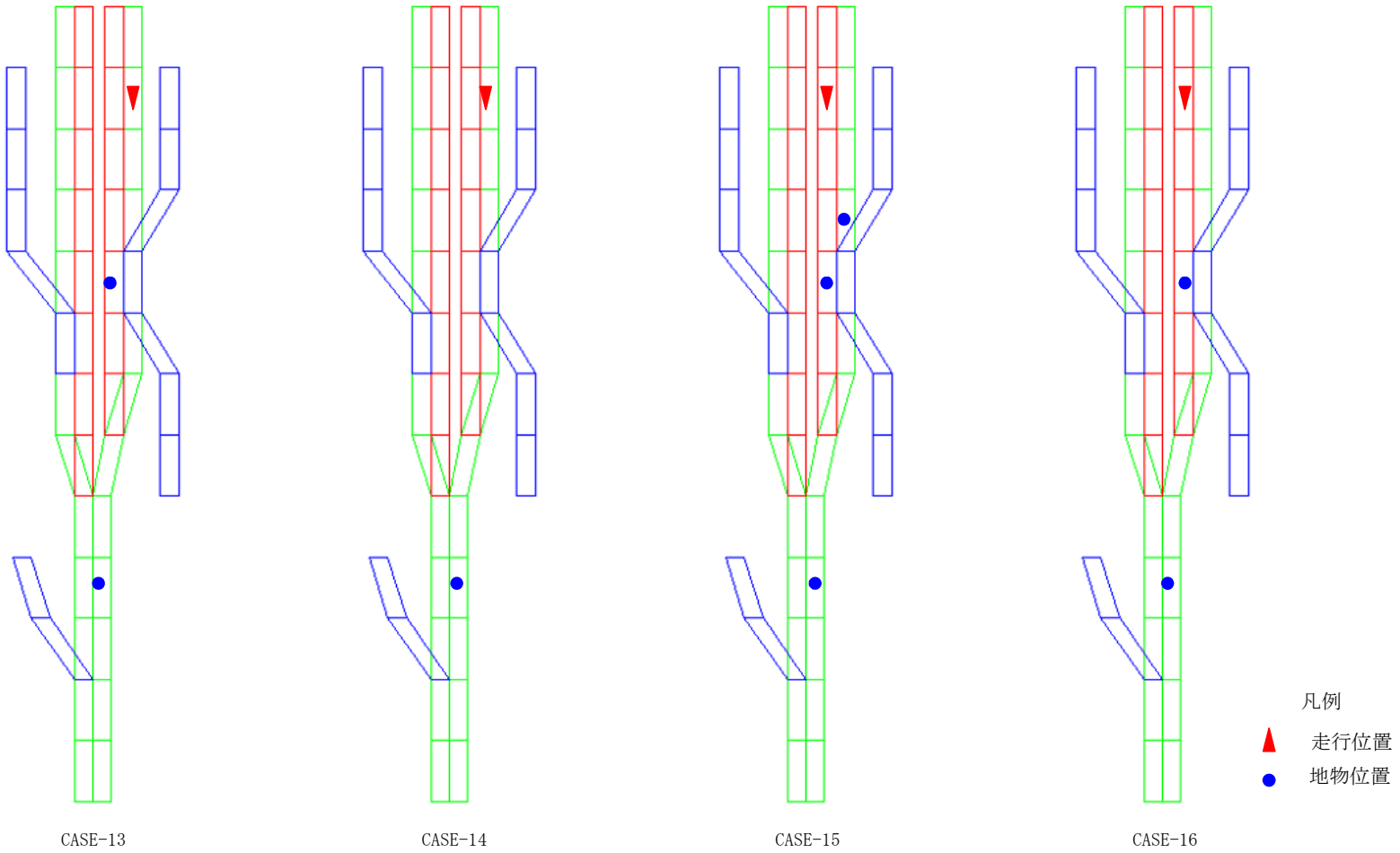


図 2-76　地物検索ケース図（その４）

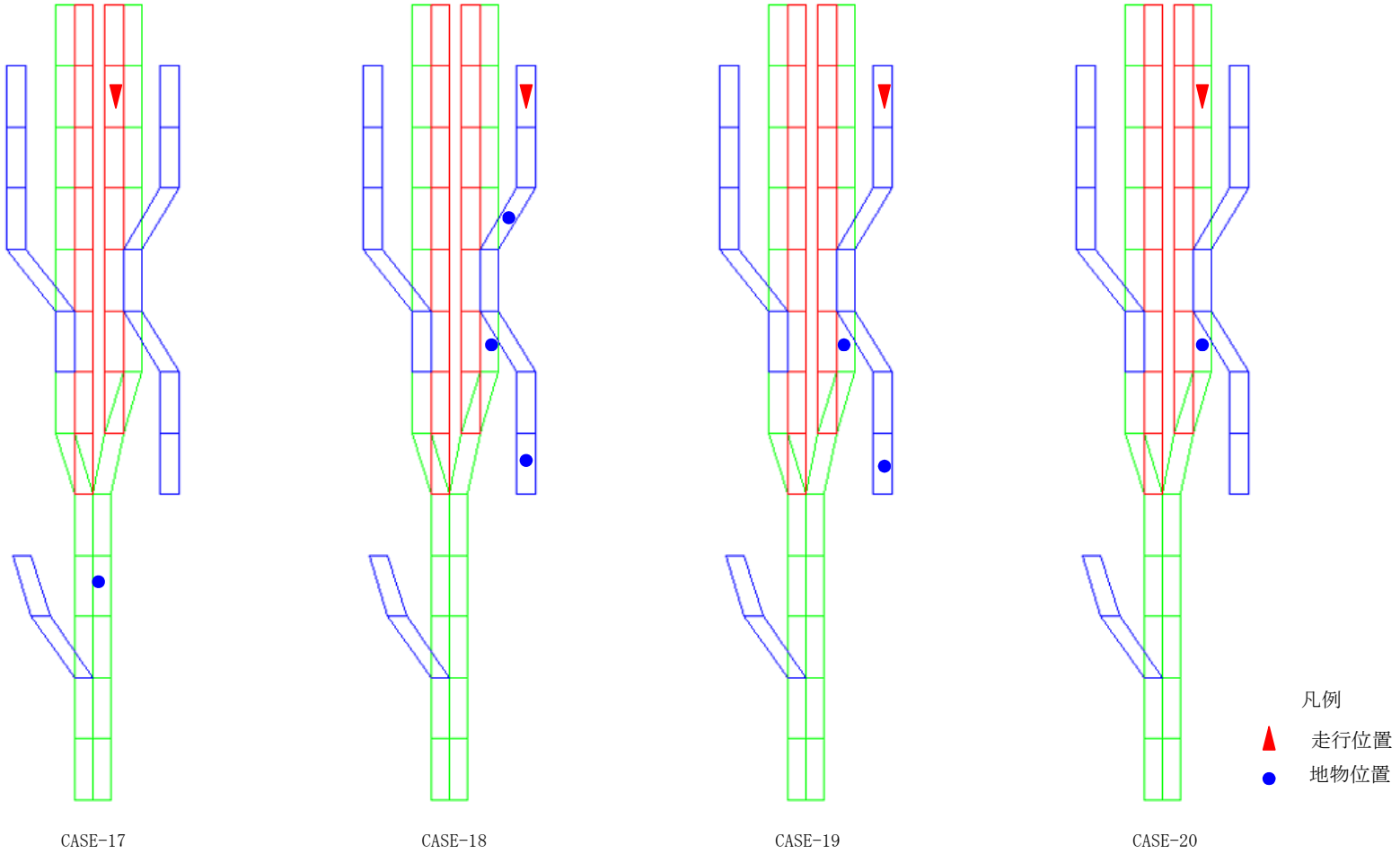


図 2-77　地物検索ケース図（その５）

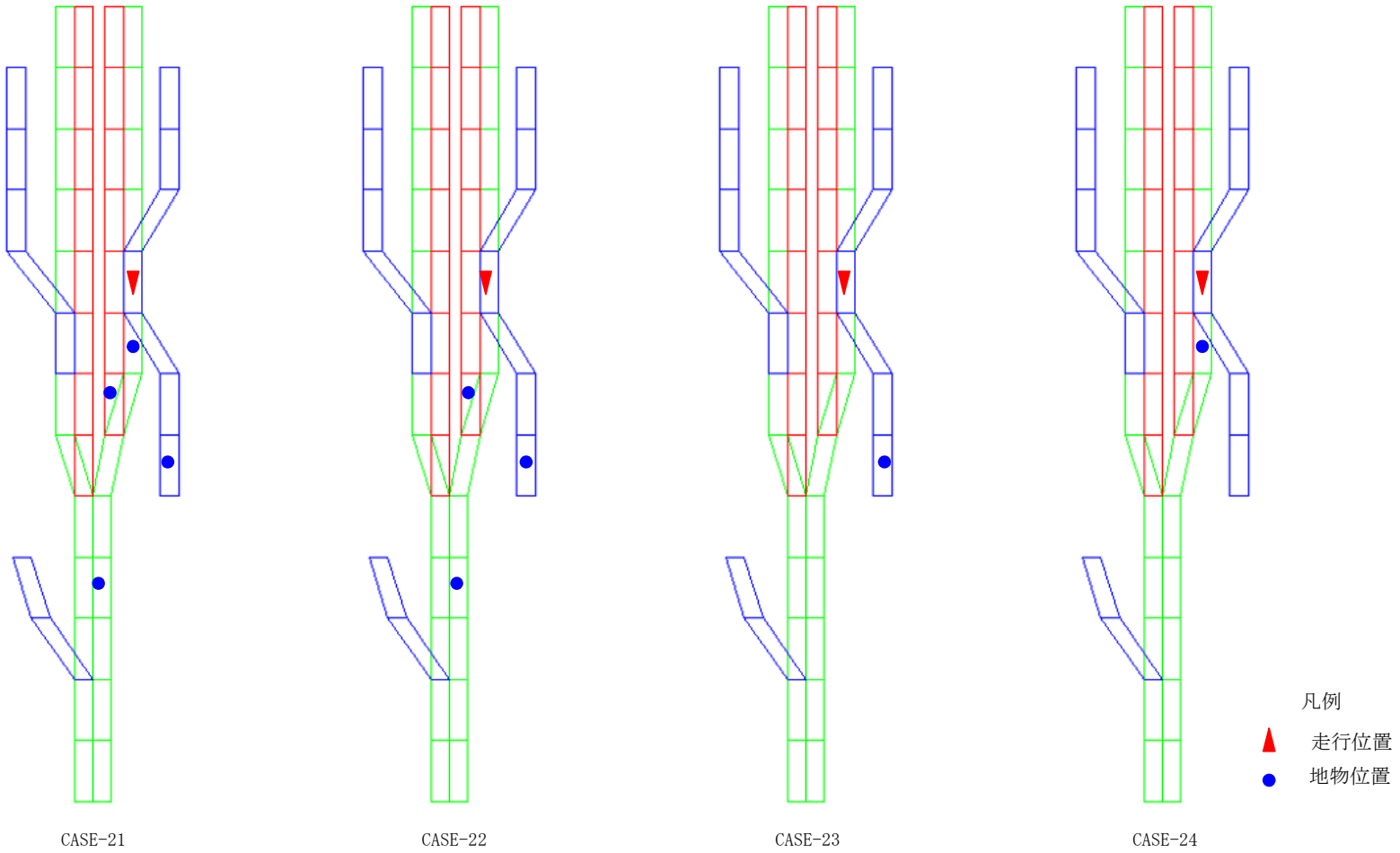


図 2-78　地物検索ケース図（その６）

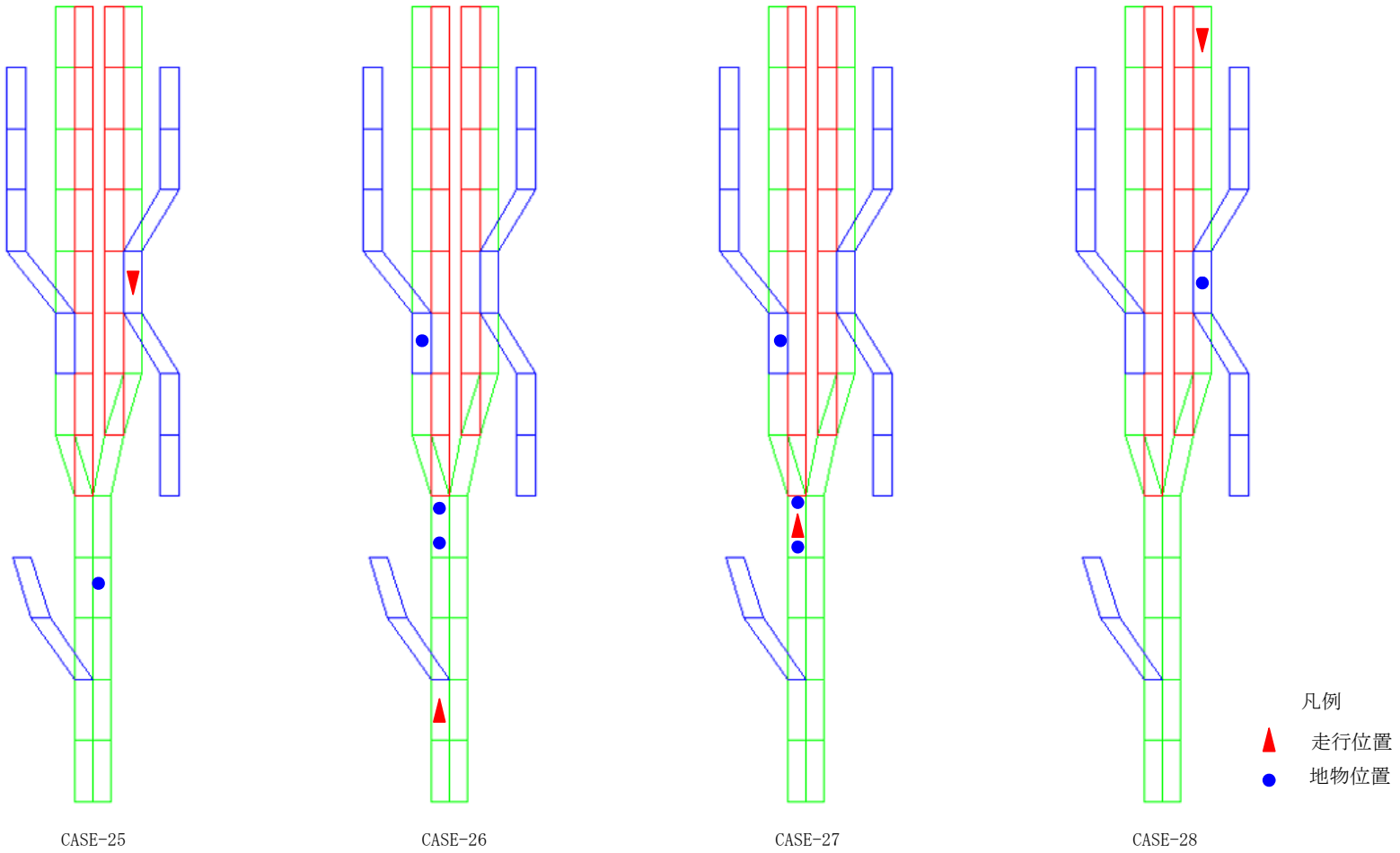


図 2-79　地物検索ケース図（その７）

c） その他機能

散布制御及び地物検索以外の単体試験としては、地図の拡大、縮小、スクロール、表示地物の表示切替え、全体地図の表示切替え、散布時の表示変化、音声案内（ビープ音）について、散布制御の単体試験と同時に実施した。以下に単体試験結果を示す。

表 2-68　その他機能単体試験結果

ケース番号	機能	テスト内容	合否	備　　考
1	画面表示	散布制御停止時に地図のスクロールができること	○	
2		散布制御時に地図のスクロールができること	○	
3		散布制御停止時に地図の拡大・縮小ができること	○	
4		散布制御時に地図の拡大・縮小ができること	○	
5	散布制御時画面表示	未散布区間が散布予定数量で色分けされ表示されること	○	
6		散布済み区間が灰色に変化していること	○	
7		散布の制御が切り変わるタイミングでビープ音がなること	○	
8	全体地図	全体地図が現在位置に応じて正しく切り変わること	○	
9	散布制御	散布履歴ファイルが生成されること	○	内容については結合時に最終確認
10		散布計画ファイルの切り替えができること	○	
11		システム強制終了時にバックアップファイルができること	○	

表 2-69　単体試験結果

No	テスト項目	テスト概要	結果
27	操作支援システム	操作支援システムが起動されること	○
28	との連動	巡回支援システム画面が最前面に表示されること	○
29		操作支援システム終了時に巡回支援システムが終了すること	○

表 2-70　単体試験結果

No	テスト項目	テスト概要	結果
30	巡回記録	指定巡回地点における時刻が正しく巡回記録登録用ファイルに記録されていること	○
31		指定巡回地点における気温が正しく巡回記録登録用ファイルに記録されていること	○
32		指定巡回地点における天候が正しく巡回記録登録用ファイルに記録されていること	○
33		指定巡回地点における路面状況が正しく巡回記録登録用ファイルに記録されていること	○
34		指定巡回地点における交通状況が正しく巡回記録登録用ファイルに記録されていること	○
35		指定巡回地点における消雪パイプ稼動状況が正しく巡回記録登録用ファイルに記録されていること	○
36		指定巡回地点における雪種別が正しく巡回記録登録用ファイルに記録されていること	○
37		指定巡回地点における積雪深が正しく巡回記録登録用ファイルに記録されていること	○

3. 計画支援システムの製作

3.1. 計画支援システムの基本設計

(1) 適用範囲

本節は、散布計画支援システムのおける基本設計についてのみ記載するものとする。

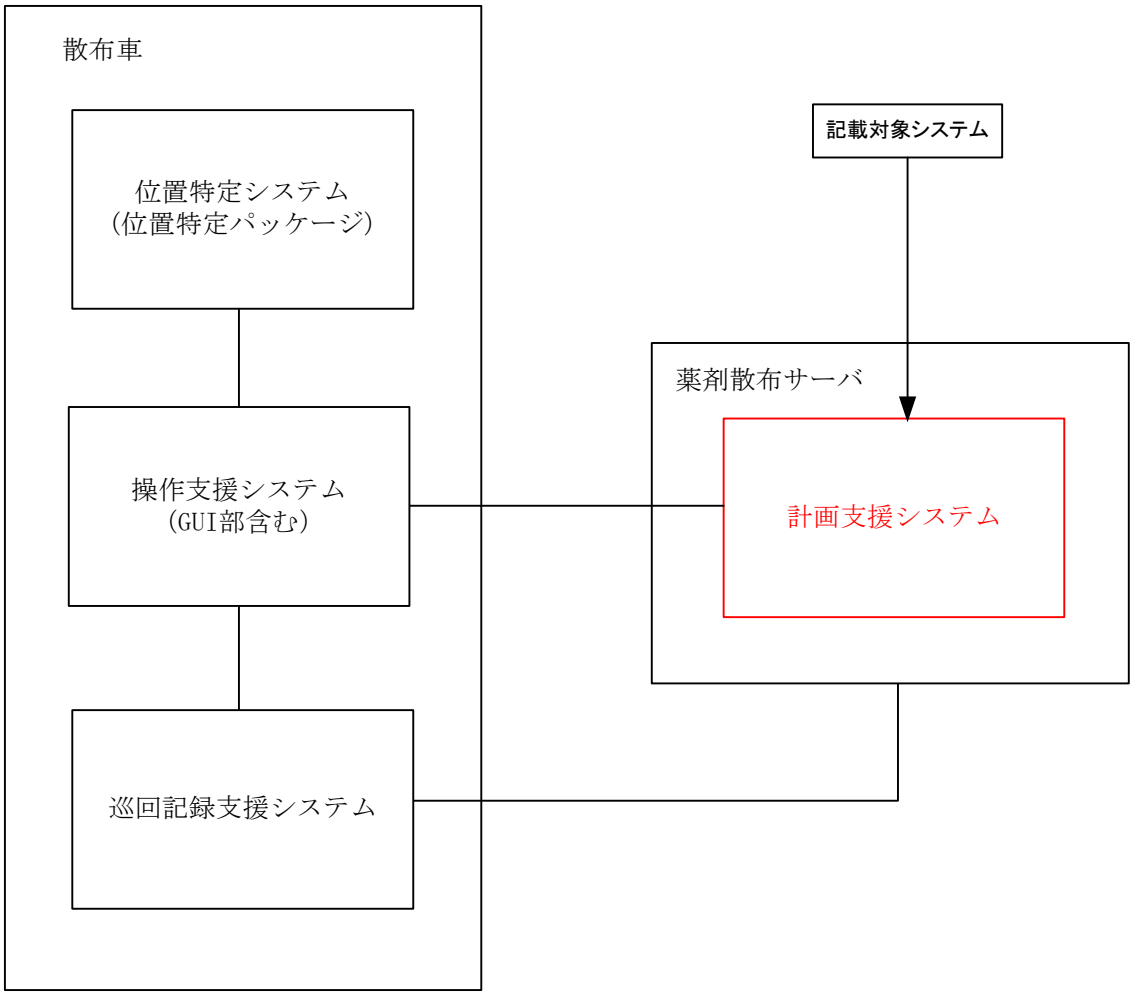


図 3-1　記載範囲

(2) 要件定義の確認

本システム全体における要件を以下のユースケース図に示す。なお、本書に記載する要件については、赤色で示されるユースケース部分である。

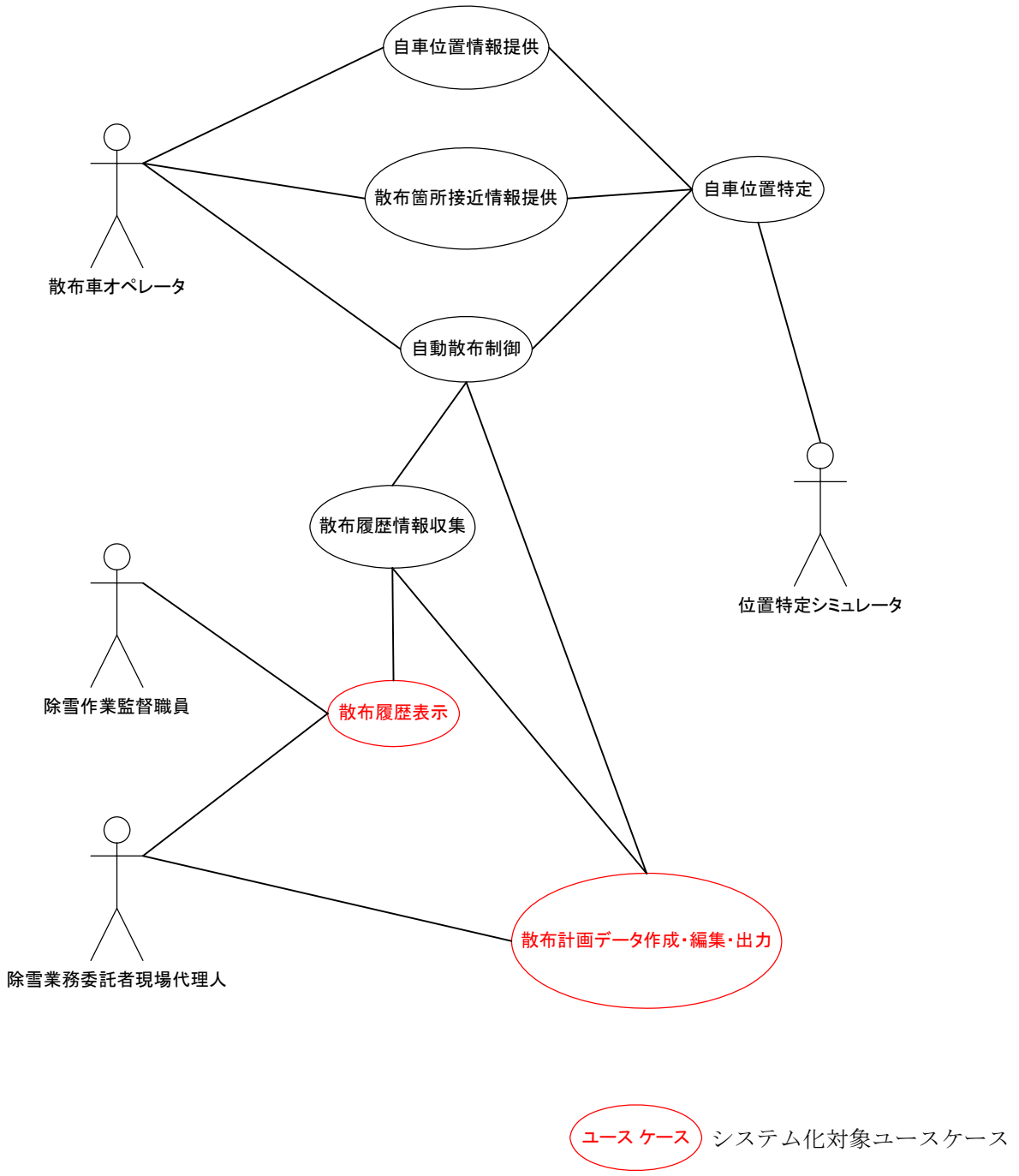


図 3-2　システム全体ユースケース図

(3) システム全体構成

システムの全体構成図を以下に示す。

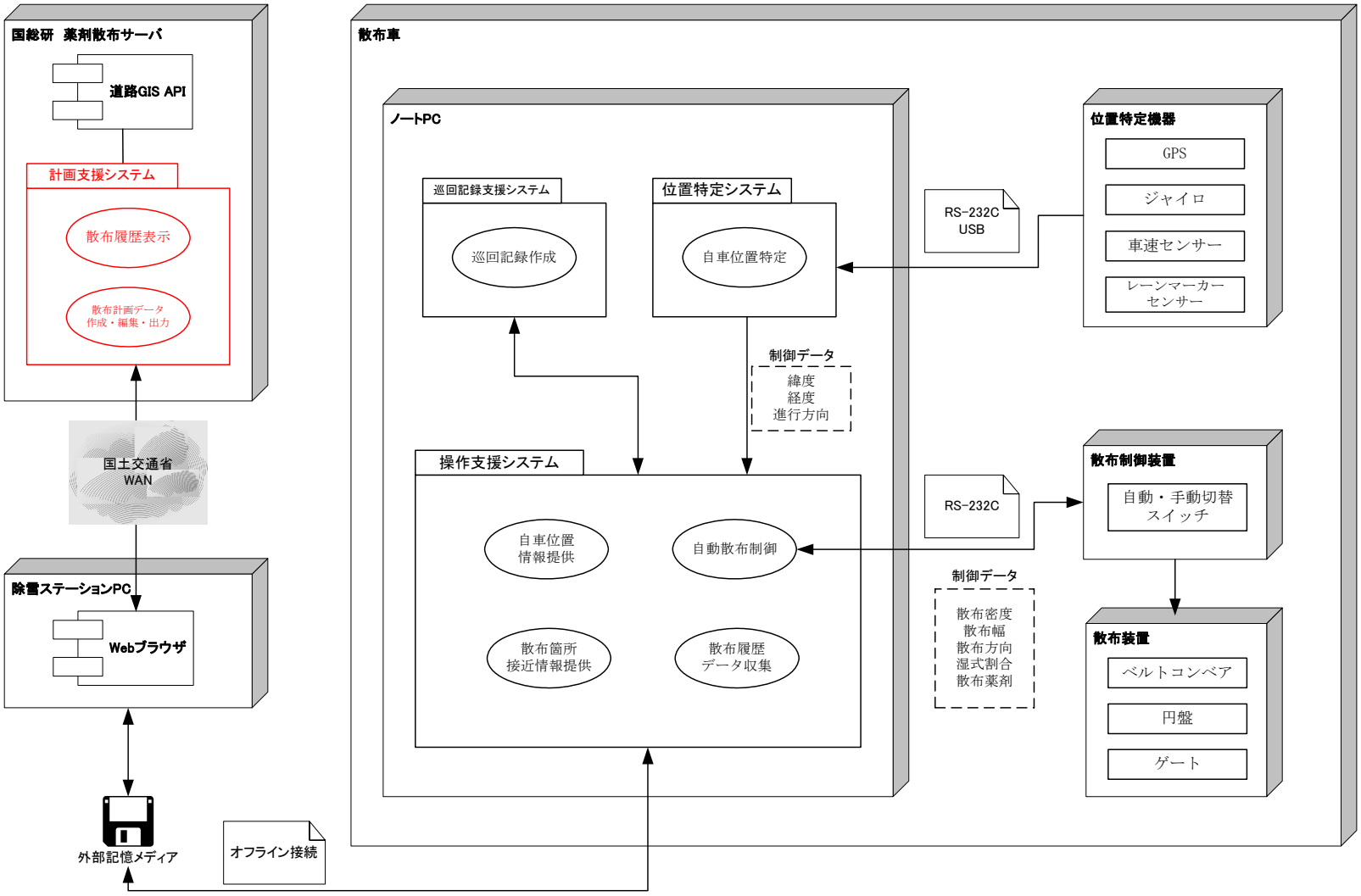


図 3-3　システム全体構成図

また、システム全体における処理の流れを以下のシーケンス図に示す。なお、図中の赤色の部分が本基本設計書に該当する箇所である。

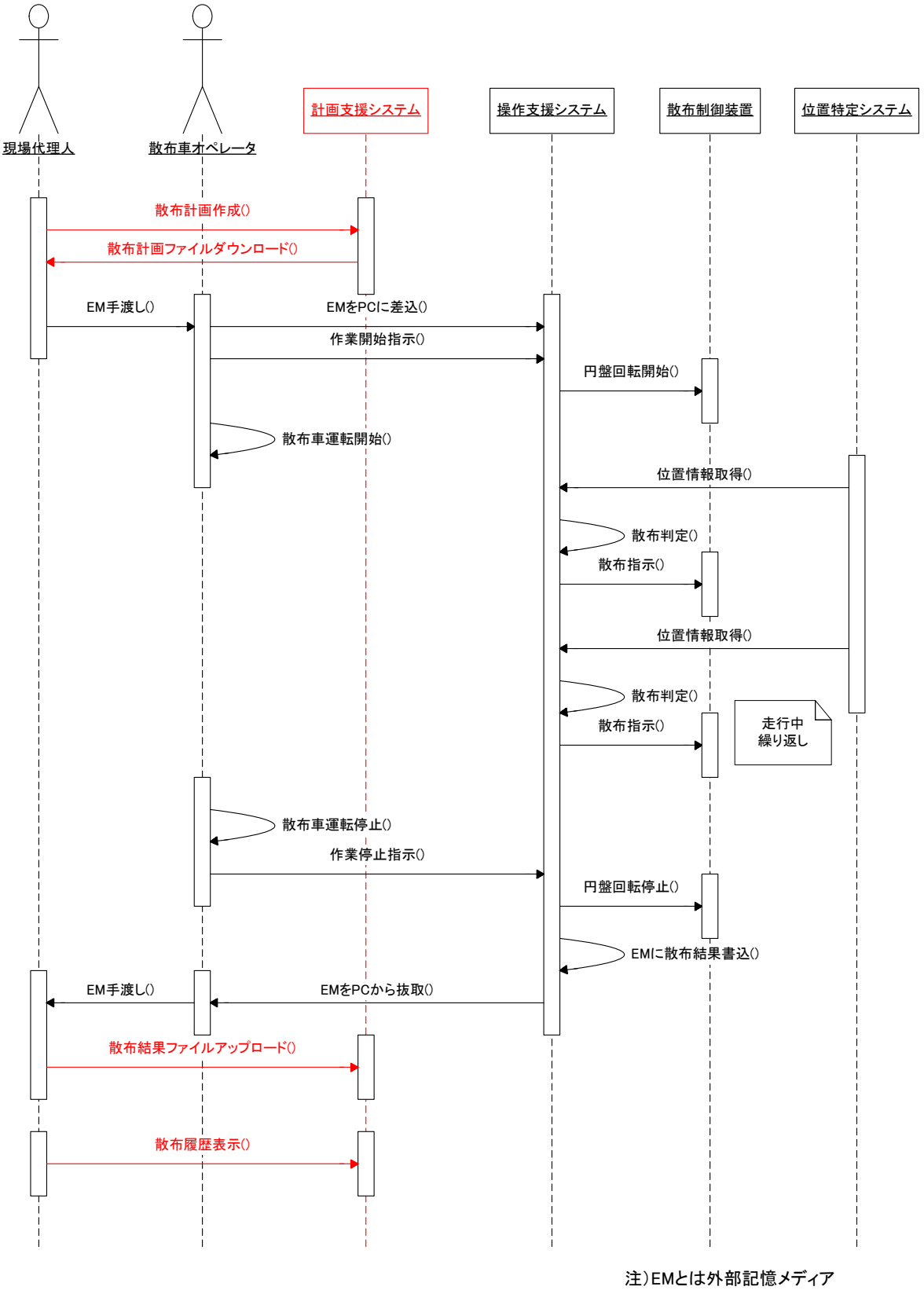


図 3-4　システム全体シーケンス図

(4)　機能設計

a） ユースケースの詳細化

システムとして必要な機能を詳細化するために、各ユースケースをさらに詳細化する。詳細化した機能を下表に示す。

表 3-1　システムに必要な機能

機能		詳細機能	機能概要
散布計画作成支援	散布履歴表示	散布履歴検索機能	過去の散布履歴データの中からユーザの指定した散布履歴データを検索する機能。
		散布履歴表示機能	検索した散布履歴データを地図上に表示する機能。
		気温情報表示機能	過去の気温データをの中からユーザの指定した地図上に表示する機能。
	散布計画データの作成・編集・出力	散布計画箇所選択支援機能	地図上から散布計画箇所を任意に選択できるようにする機能。
		散布計画内容登録支援機能	ユーザが選択した散布計画箇所の計画内容を登録できるようにする機能。
		散布計画表示機能	登録した散布計画箇所および内容を地図上に表示する機能。
		散布計画編集支援機能	登録した散布計画箇所および内容を変更・登録できるようにする機能。
		車両用散布計画データ出力機能	ユーザが指定した任意の散布計画データを車両提供用記憶媒体に出力する機能。
	作業者登録	作業者リスト登録支援機能	実験システムの作業者を変更・登録する機能。

b） 散布履歴表示機能

散布履歴表示機能は、散布操作支援システムによって散布時に自動生成された散布結果データを登録後、道路 GIS サーバーに登録されている地図を背景として、散布結果を表示するものである。また、気温取得システム（仮名）によって得られた気温データを本システムに登録し表示するものである。機能の制約としては、以下とする。

- 道路 GIS サーバーを使用したシステムとすること
- 背景地図には道路 GIS API を使用すること
- 背景地図上に、散布領域（メッシュ）を表示し、その領域をクリックした時に散布履歴情報が確認できること
- 地図表示には、拡大・縮小およびスクロール機能があること
- 散布履歴および気温に関しては、最新情報の表示と過去の履歴が選択可能なこと

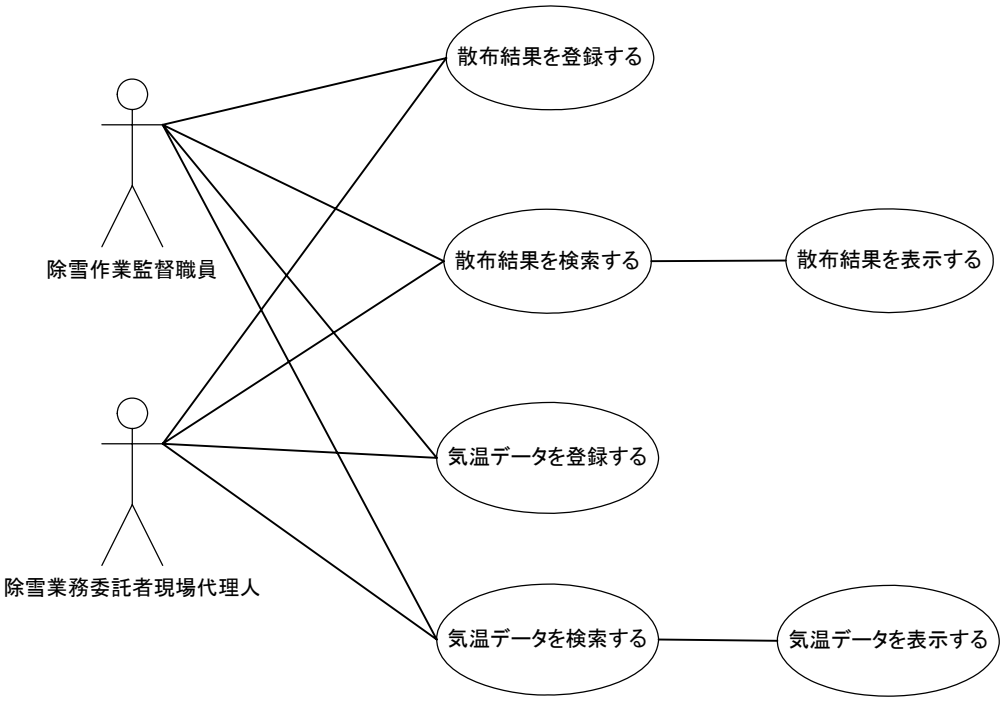


図 3-5　散布履歴表示　詳細ユースケース図

c） 散布計画データの作成・編集・出力機能

散布計画データの作成・編集・出力機能は、散布計画の基本となる散布メッシュや散布量、散布幅などの散布情報の登録・編集およびこの基本データを組み合わせて散布車の制御を行うための散布計画データの作成、車両提供用記録媒体に保存するものである。機能の制約としては、以下とする。

- 道路 GIS サーバーを使用したシステムとすること
- 背景地図には道路 GIS API を使用すること
- 散布計画データの基本となる散布メッシュおよび散布情報の登録・編集な可能なこと
- 散布メッシュおよび散布情報を組み合わせて、散布計画データが作成可能なこと
- 背景地図上に、散布メッシュを表示し、その領域をクリックした時に散布計画情報が確認でき、さらに編集して散布計画データに出力できること
- 地図表示には、拡大・縮小およびスクロール機能があること

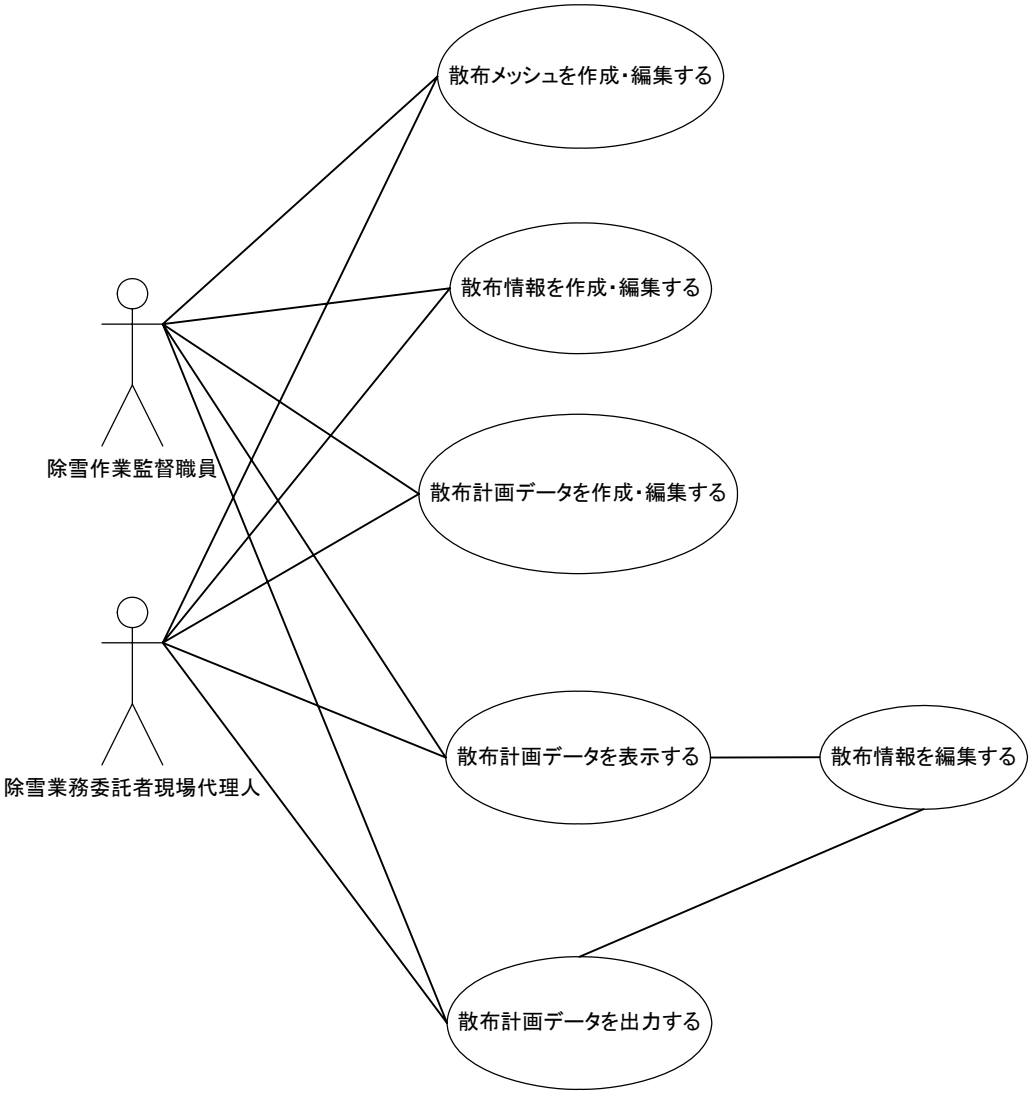


図 3-6　散布計画データの作成・編集・出力　詳細ユースケース図

(5) システムモデリング

a） サブシステム分割

本システムのサブシステム構成を以下に示す。

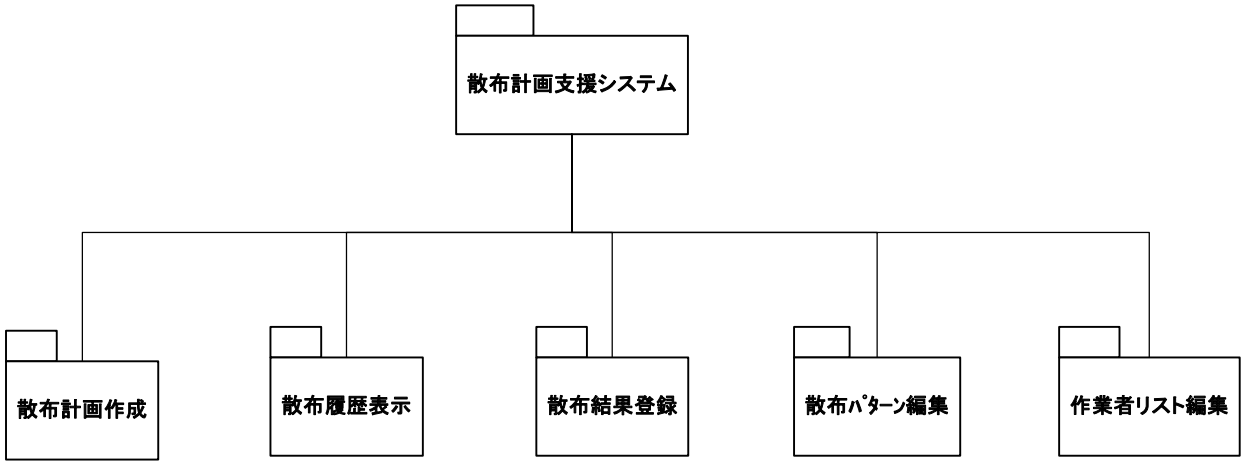


図 3-7　サブシステム構成図

b） 散布計画作成

散布計画作成サブシステムにおけるクラス図およびシーケンス図を以下に示す。

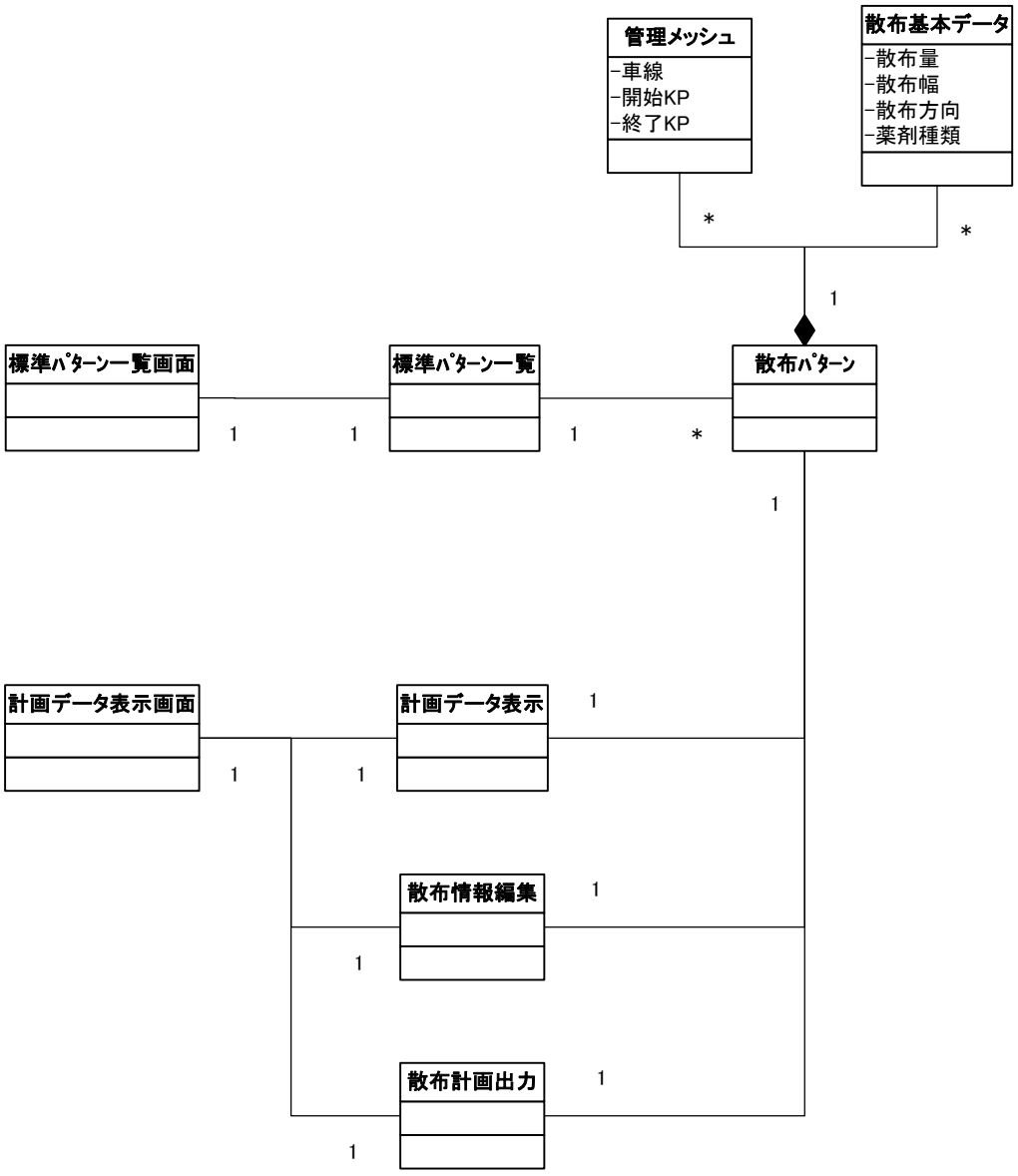


図 3-8　散布計画作成クラス図

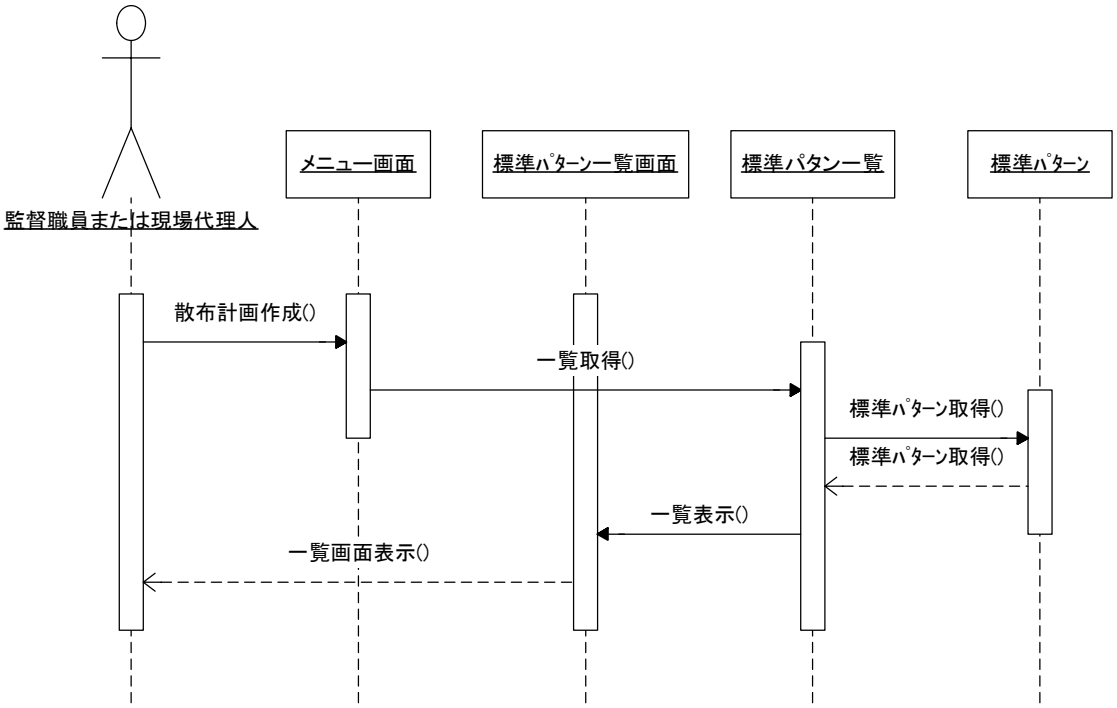


図 3-9　散布計画作成シーケンス図（その１）

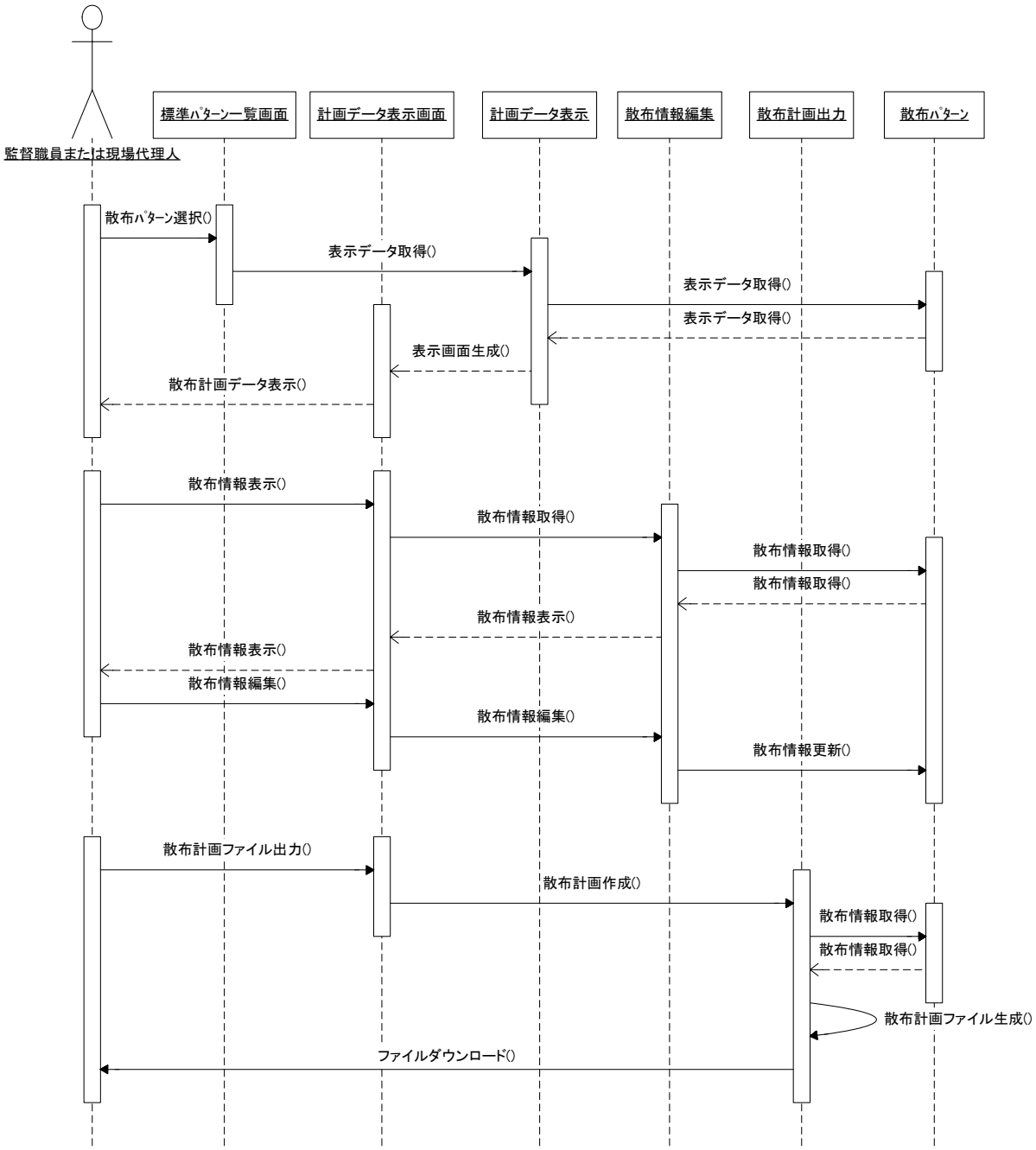


図 3-10　散布計画作成シーケンス図（その２）

c） 散布履歴表示

散布履歴表示サブシステムにおけるクラス図およびシーケンス図を以下に示す。

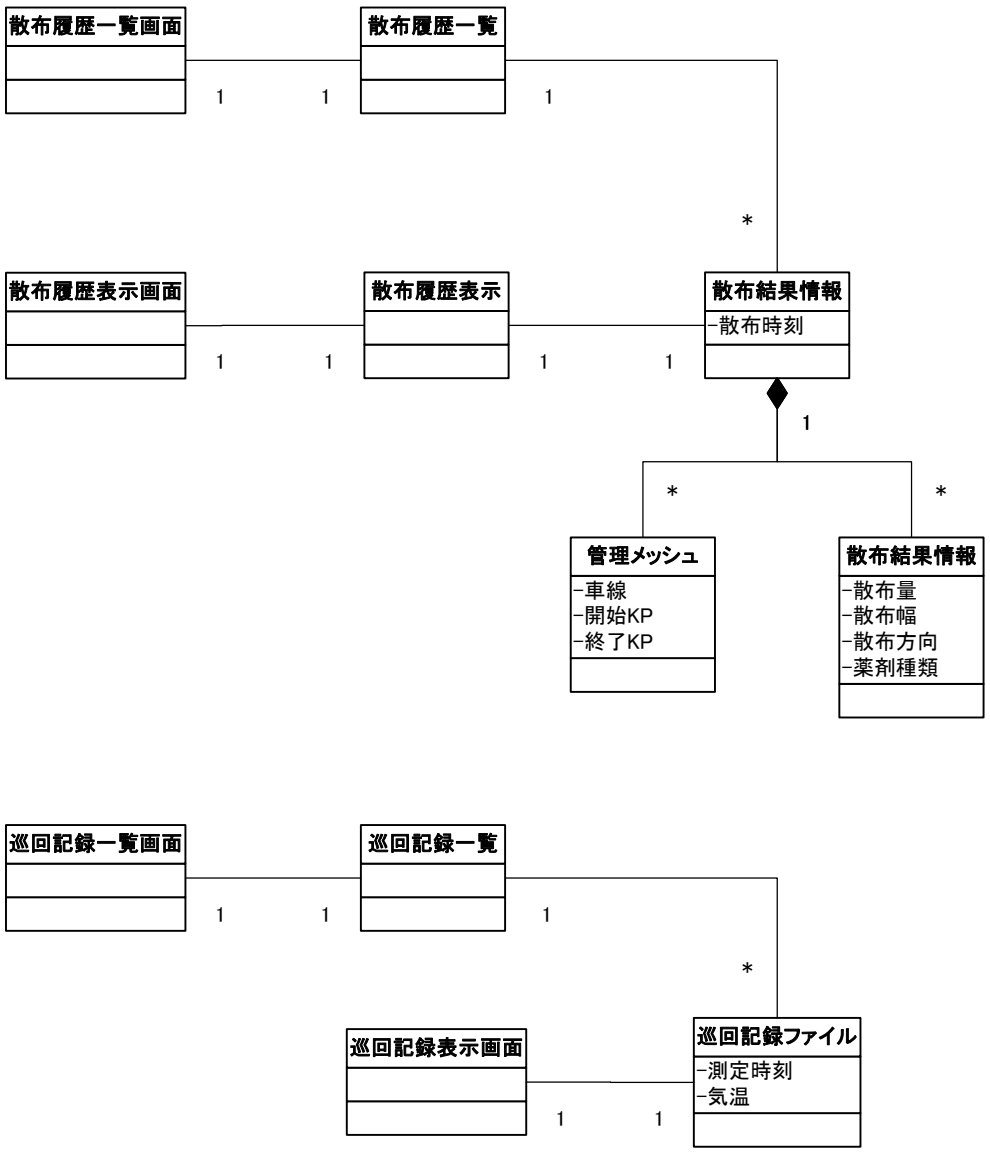


図 3-11　散布履歴表示クラス図

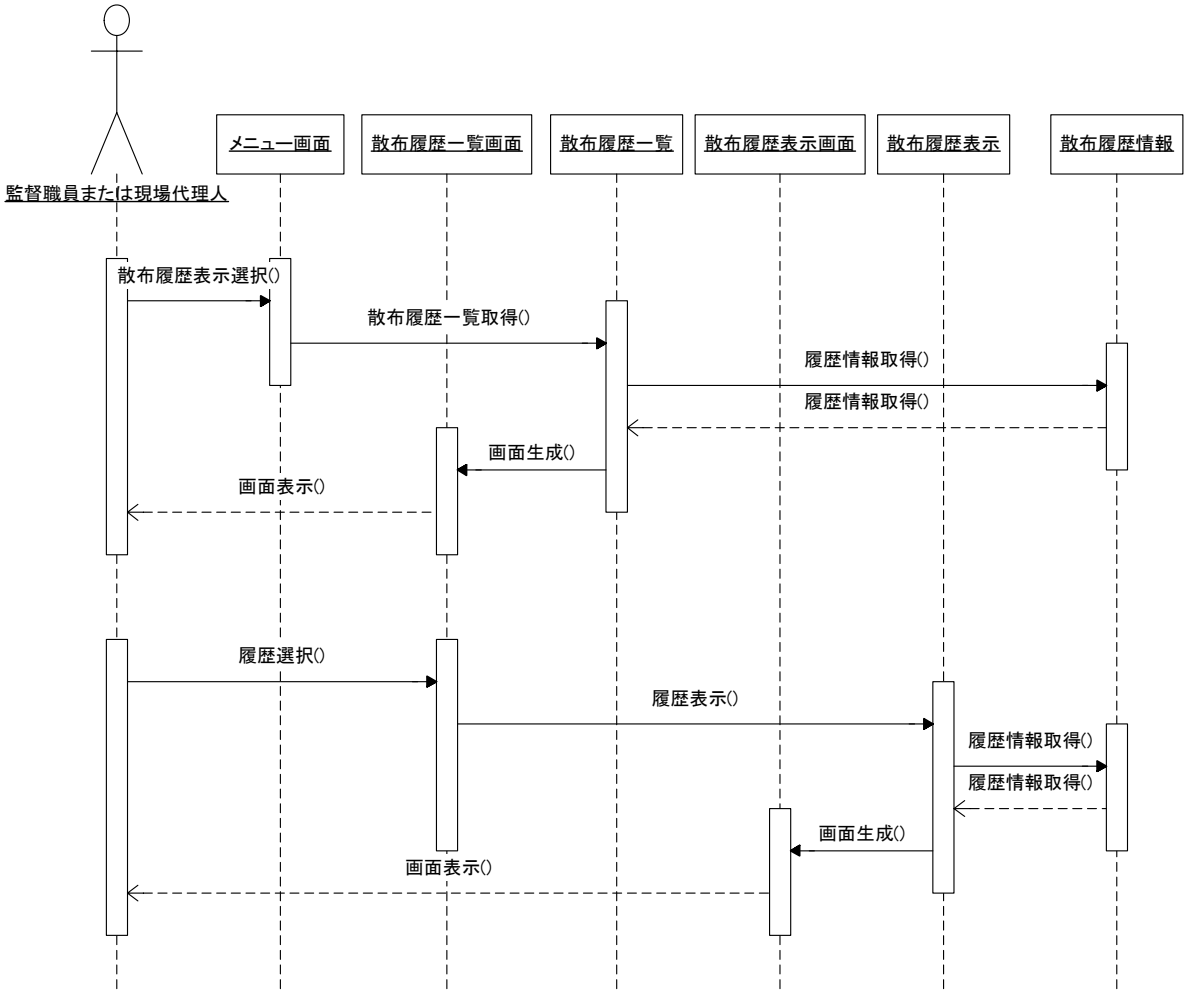


図 3-12　散布履歴表示シーケンス図（散布履歴）

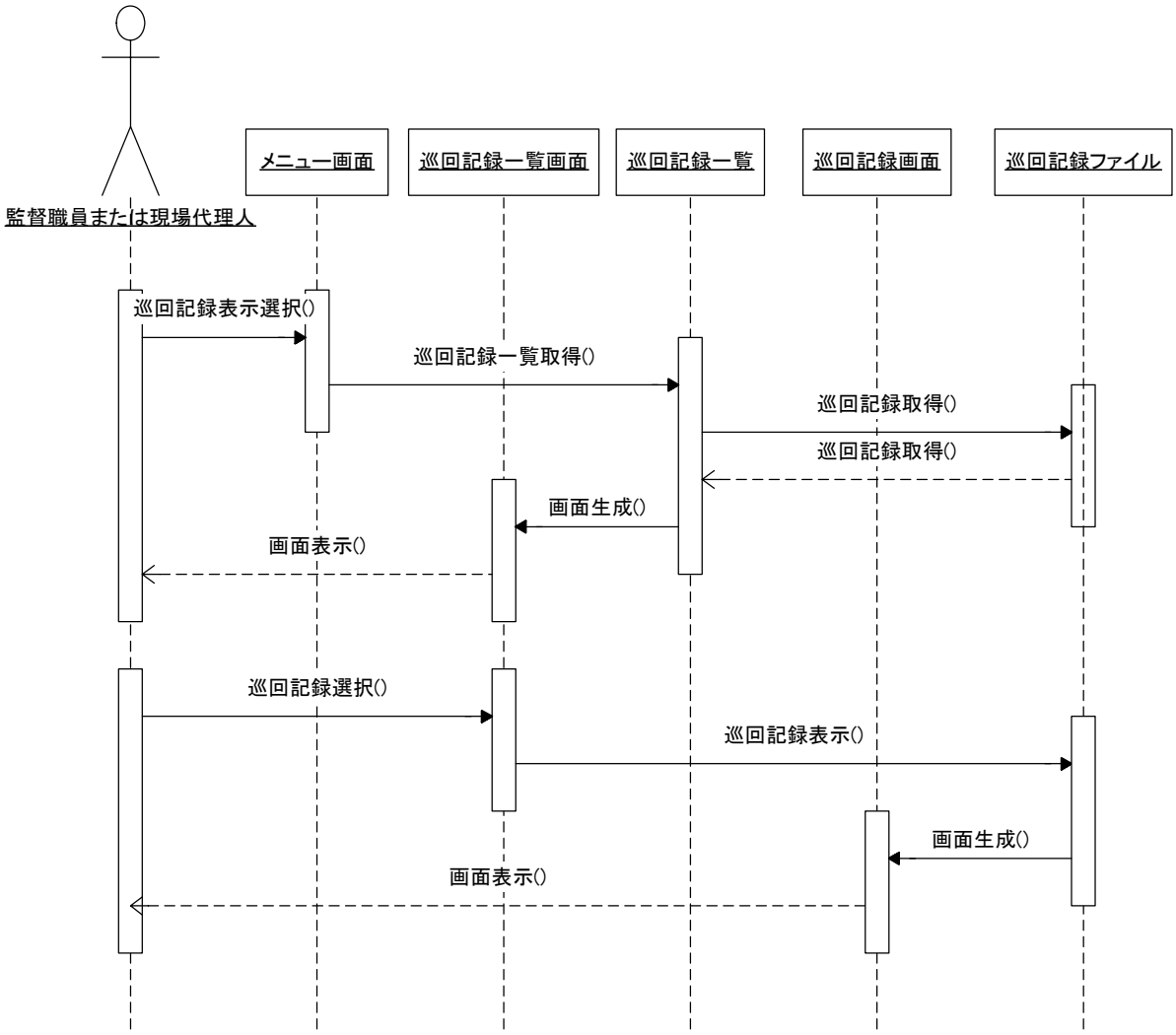


図 3-13　散布履歴表示シーケンス図（巡回記録）

d） 散布結果登録

散布結果登録サブシステムにおけるクラス図およびシーケンス図を以下に示す。

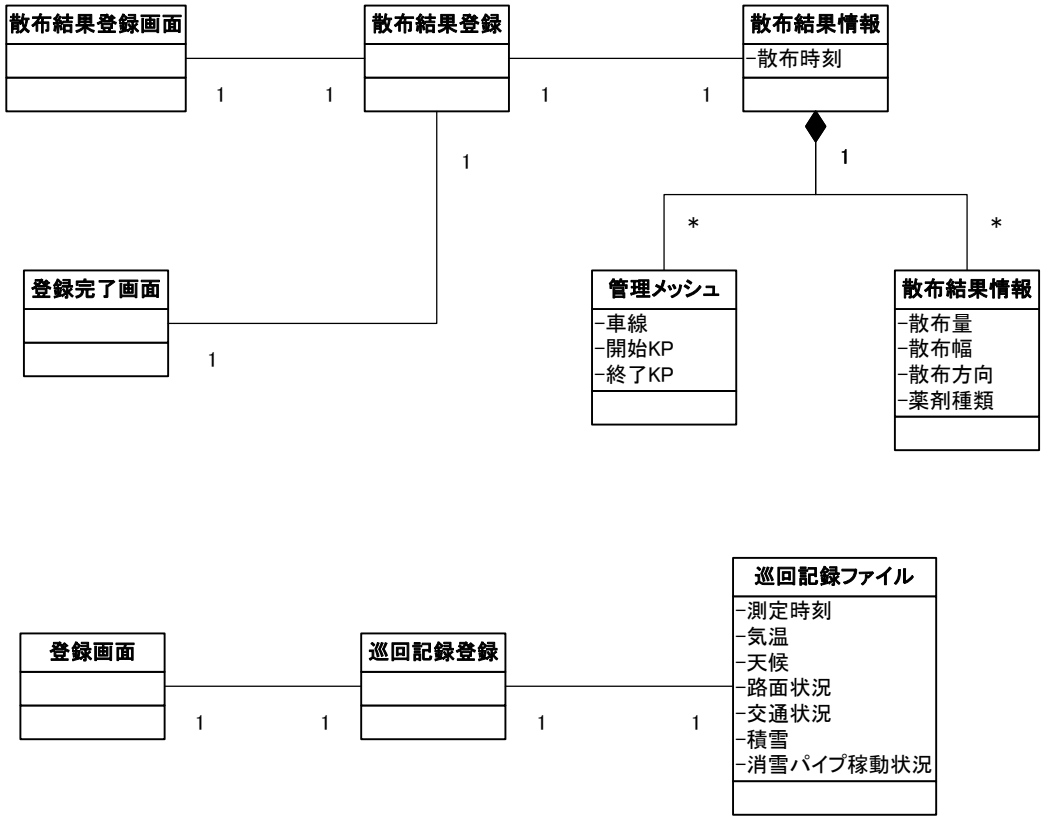


図 3-14　散布結果登録クラス図

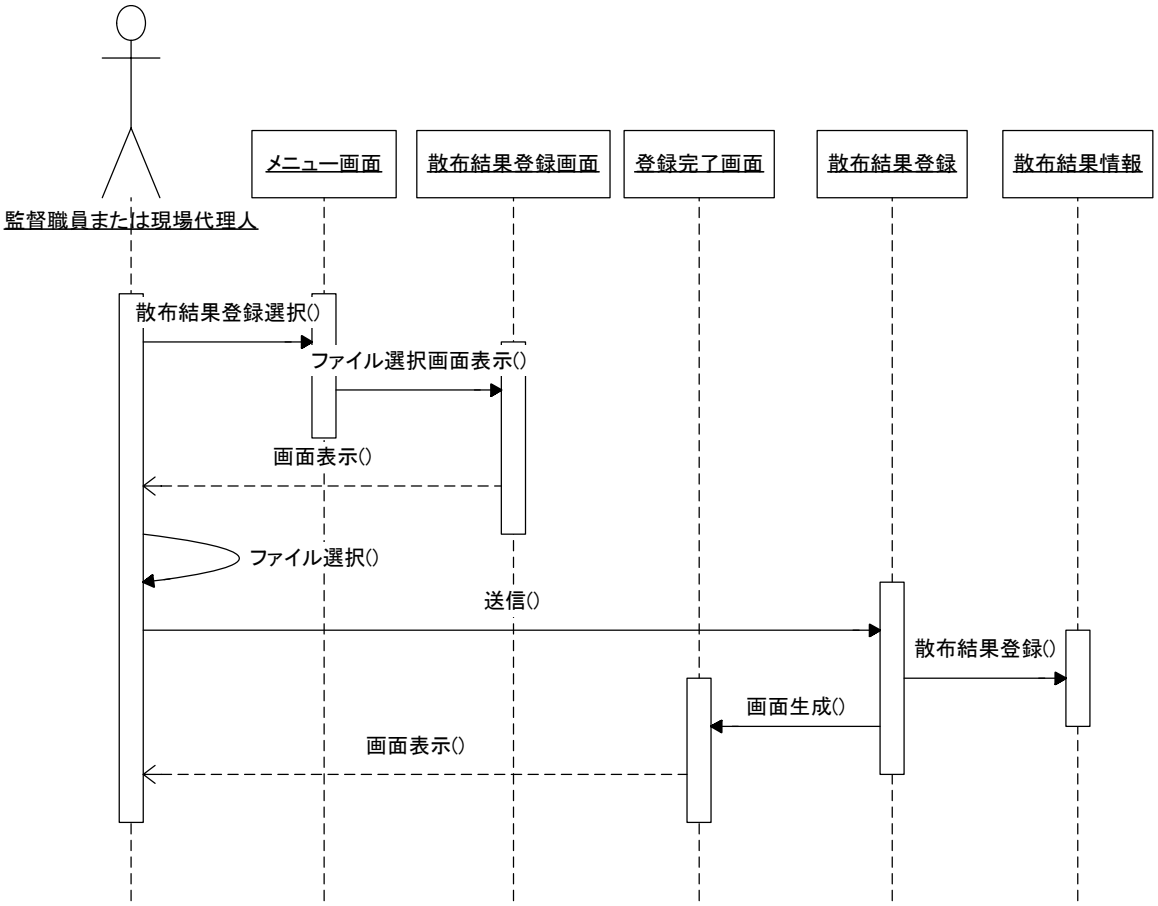


図 3-15　散布結果登録シーケンス図（散布履歴）

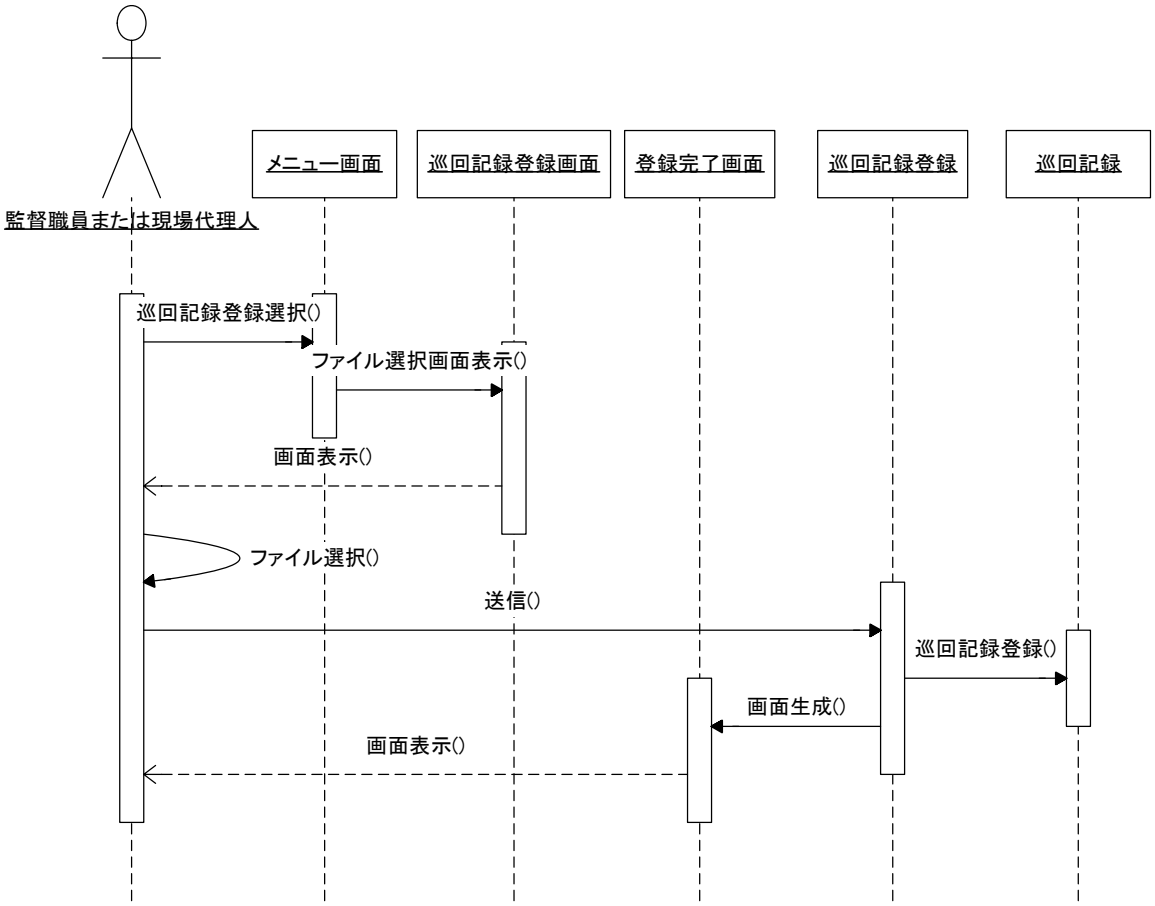


図 3-16　散布結果登録シーケンス図（巡回記録）

e） 標準パターン編集

標準パターン編集サブシステムにおけるクラス図およびシーケンス図を以下に示す。

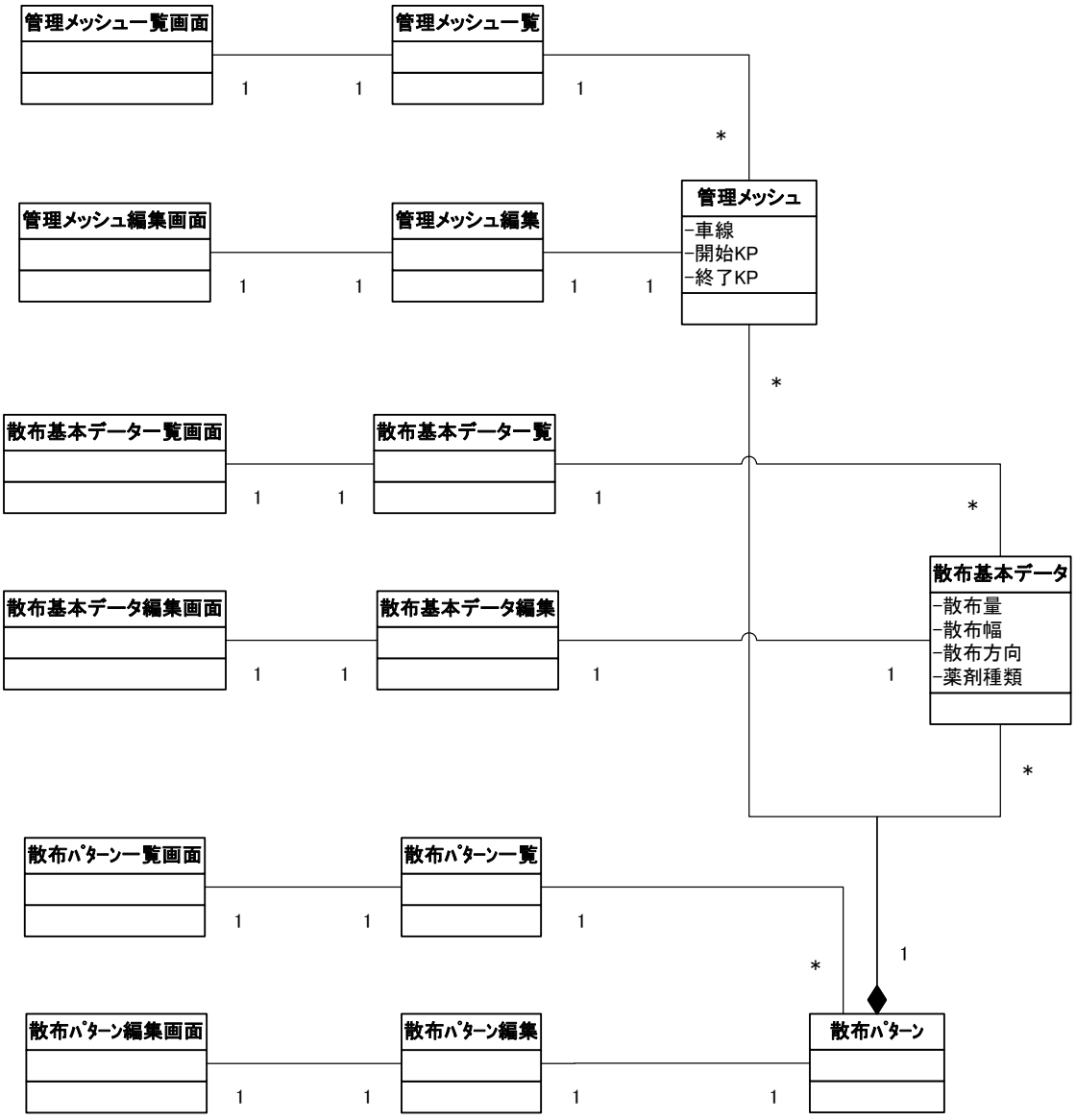


図 3-17　標準パターン編集クラス図

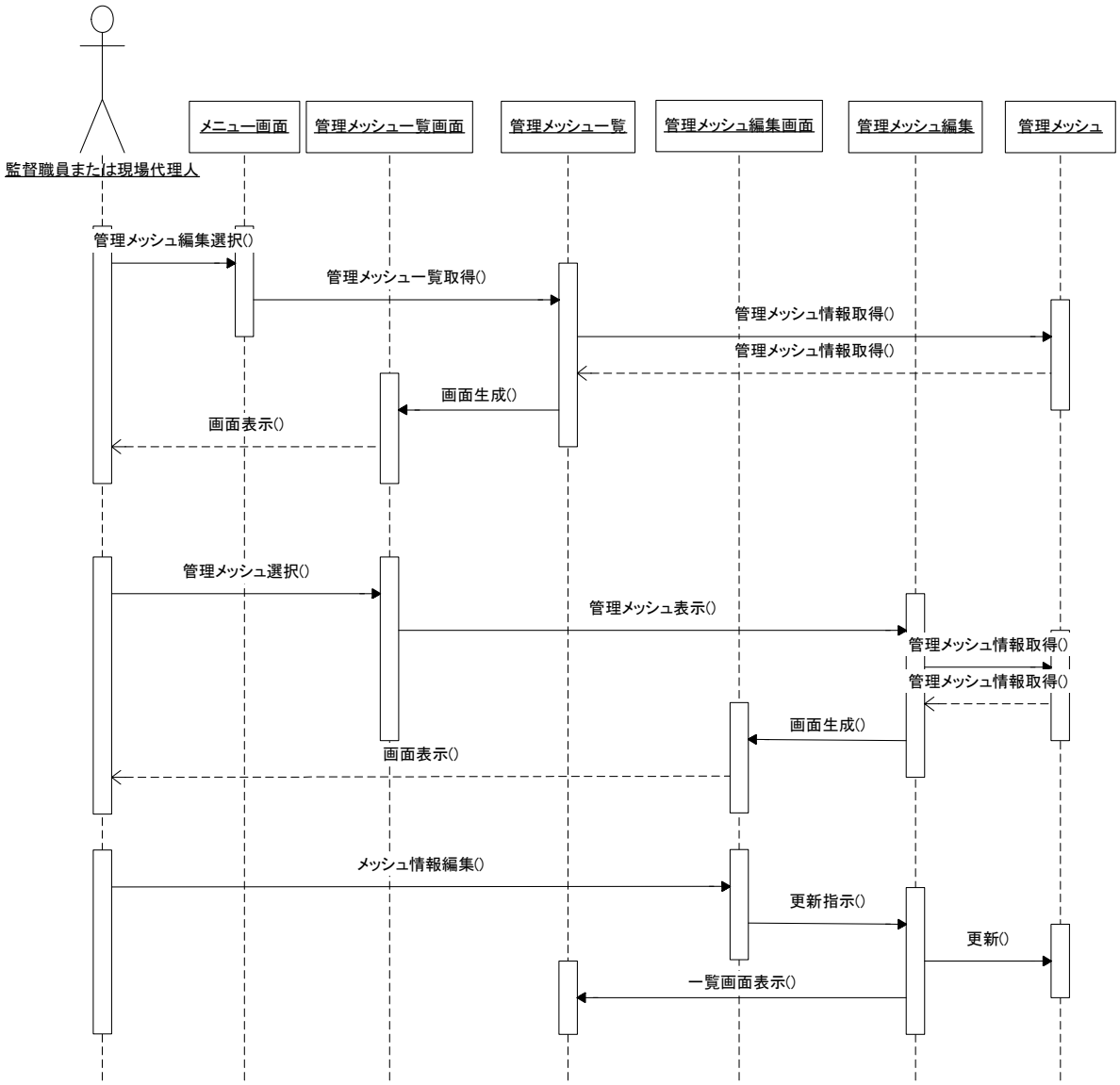


図 3-18　標準パターン編集シーケンス図（その１）

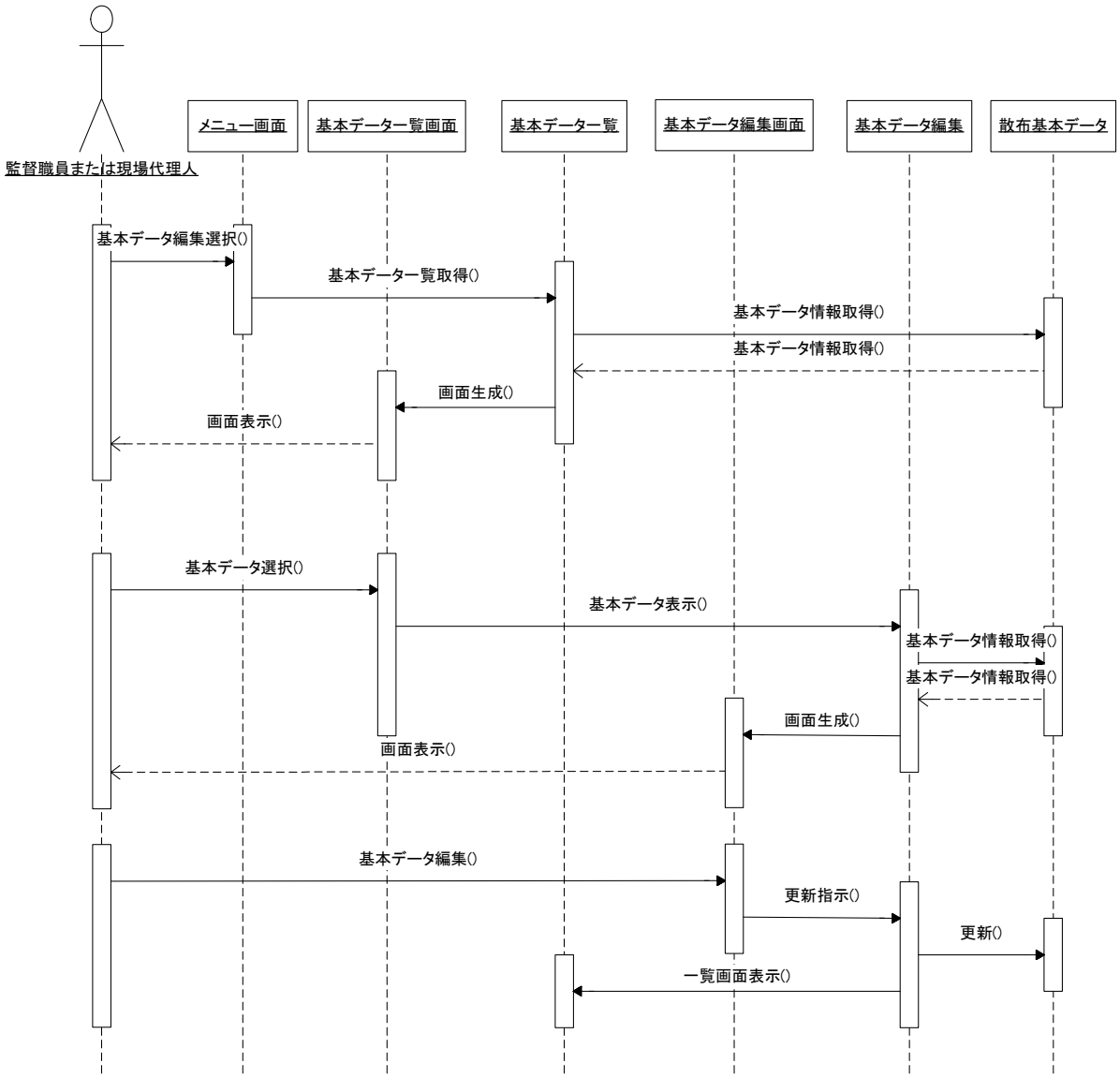


図 3-19　標準パターン編集シーケンス図（その２）

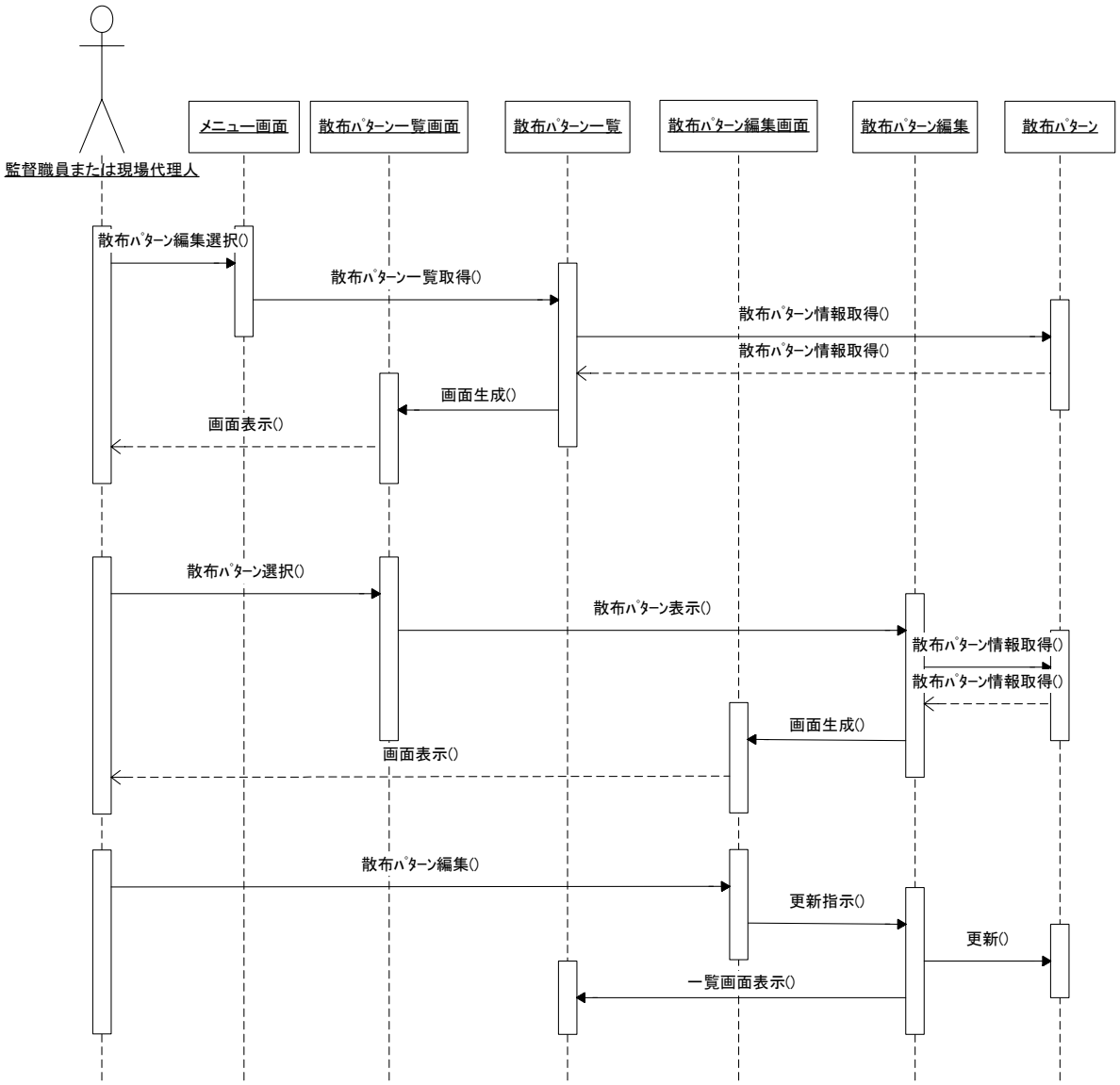


図 3-20　標準パターン編集シーケンス図（その３）

(6) 画面設計

本システムにおける画面構成、画面および遷移図を以下に示す。

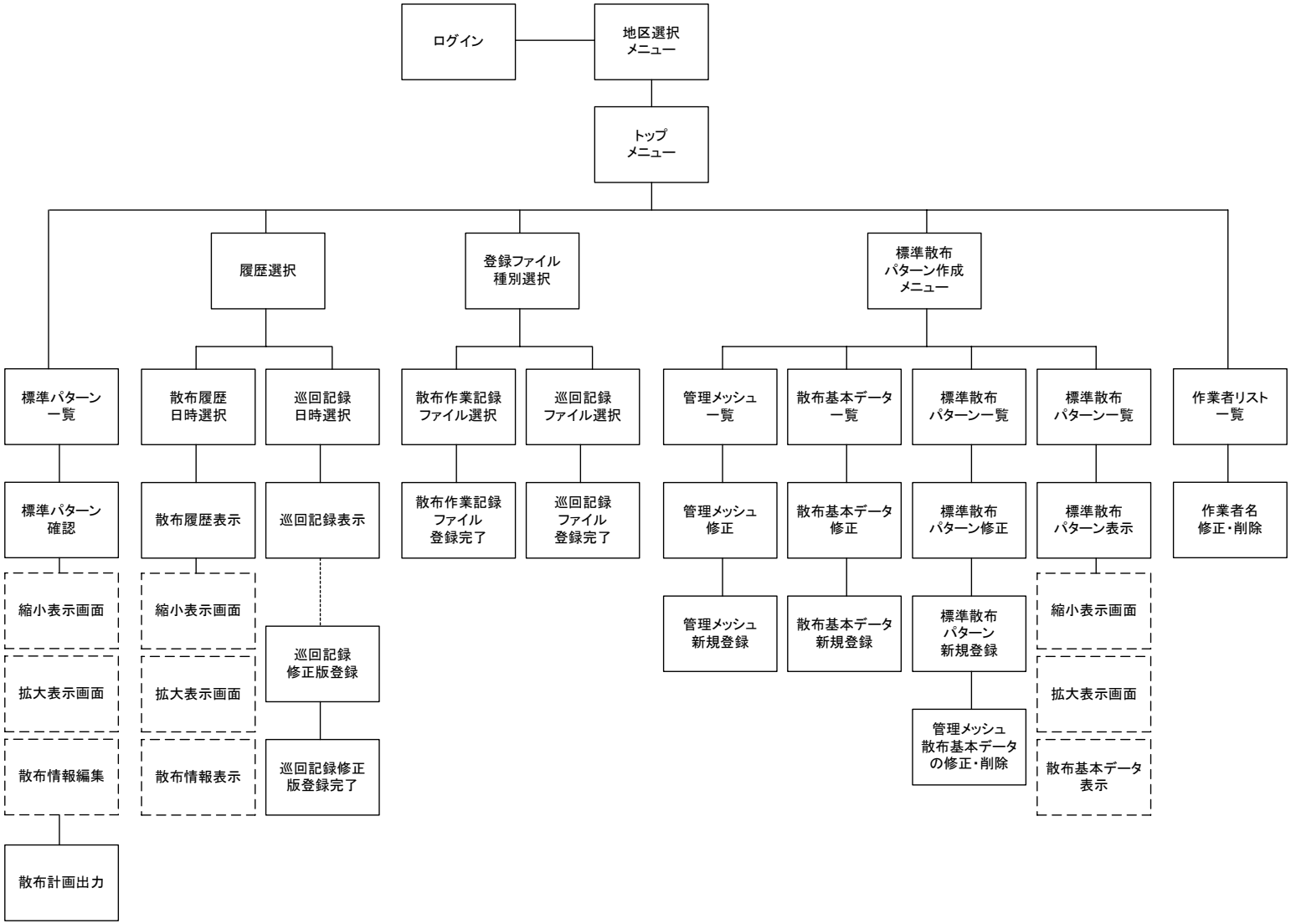
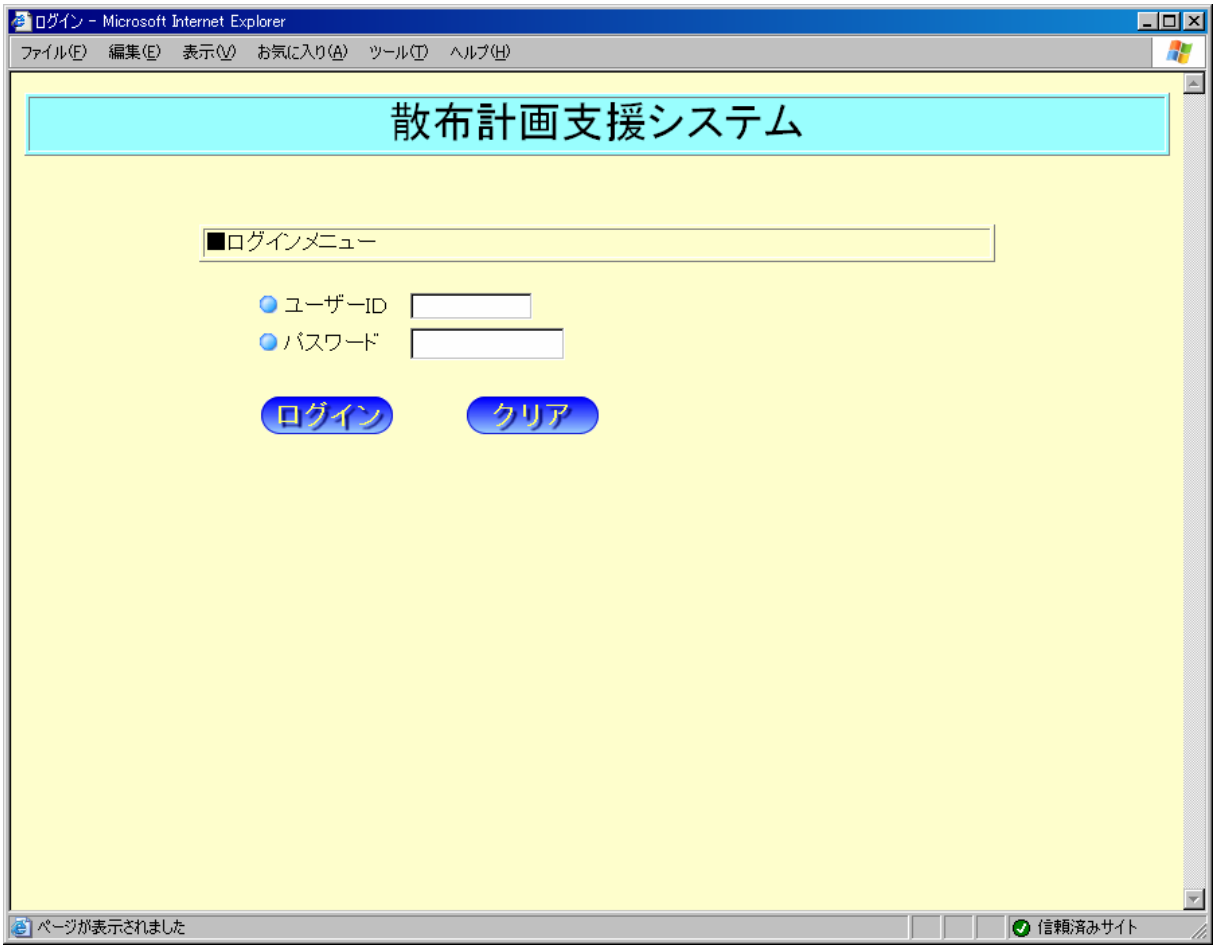

ログイン
地区選択
メニュー
トップ
メニュー
履歴選択
登録ファイル
種別選択
標準散布
パターン作成
メニュー
標準パターン
一覧
散布履歴
日時選択
巡回記録
日時選択
散布作業記録
ファイル選択
巡回記録
ファイル選択
管理メッシュ
一覧
散布基本データ
一覧
標準散布
パターン一覧
標準散布
パターン一覧
作業者リスト
一覧
標準パターン
確認
散布履歴表示
巡回記録表示
散布作業記録
ファイル
登録完了
巡回記録
ファイル
登録完了
管理メッシュ
修正
散布基本データ
修正
標準散布
パターン修正
標準散布
パターン表示
作業者名
修正・削除
縮小表示画面
縮小表示画面
巡回記録
修正版登録
管理メッシュ
新規登録
散布基本データ
新規登録
標準散布
パターン
新規登録
縮小表示画面
拡大表示画面
拡大表示画面
拡大表示画面
散布情報編集
散布情報表示
巡回記録修正
版登録完了
管理メッシュ
散布基本データ
の修正・削除
散布基本データ
表示
散布計画出力

図 3-21　画面構成

表 3-2　画面レイアウト

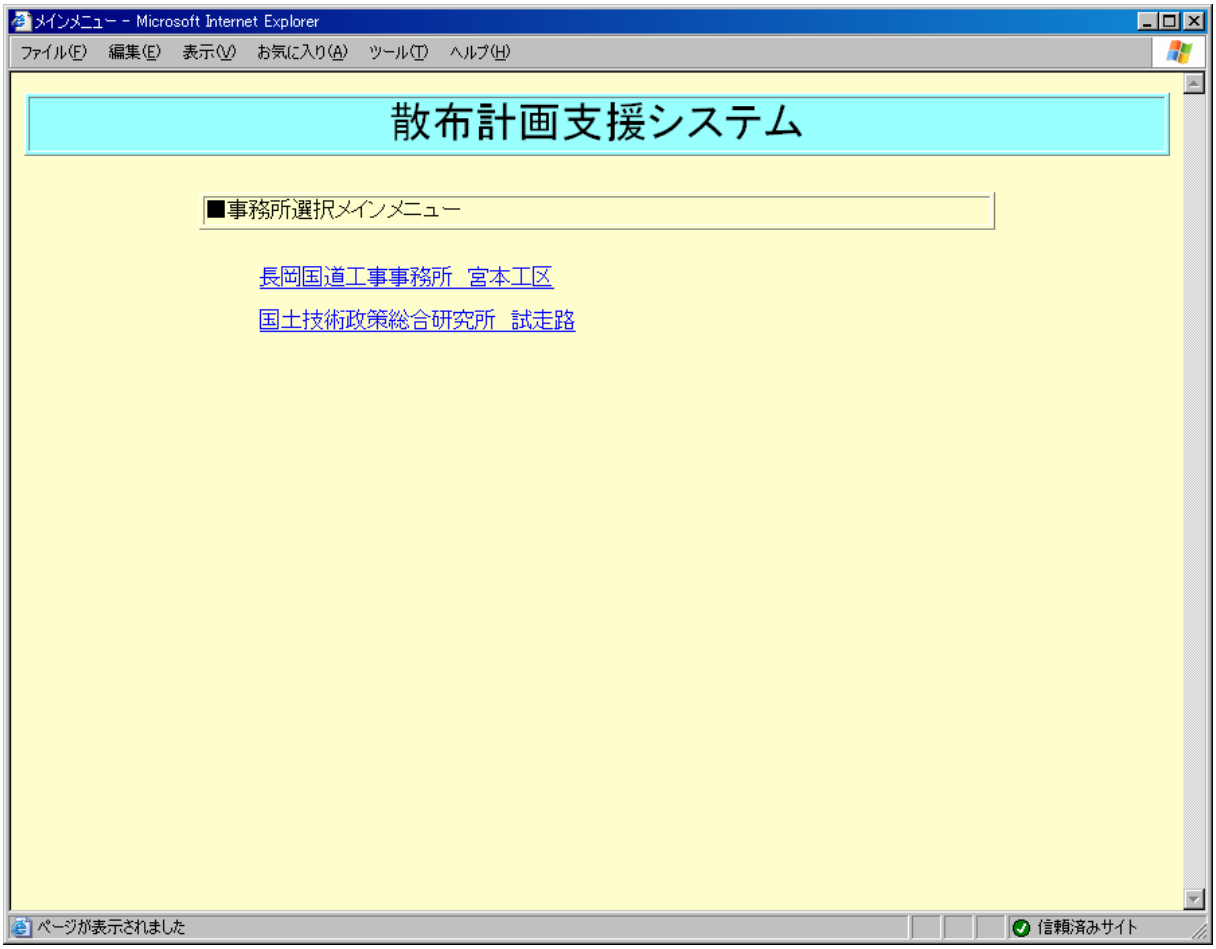
画　面　名	ログイン		
概要	国総研メニューよりリンクした時に、表示される画面。ログイン画面で入力されたユーザーID とパスワードが認証されたら、地区選択メニュー画面へ遷移する。		
遷移元	国総研メニュー	遷移先	地区選択メニュー



項目名	型	説　　明
ユーザーID	インプットボックス	ログインするユーザーID
パスワード	インプットボックス	ユーザーID に対するパスワード

表 3-3　画面レイアウト

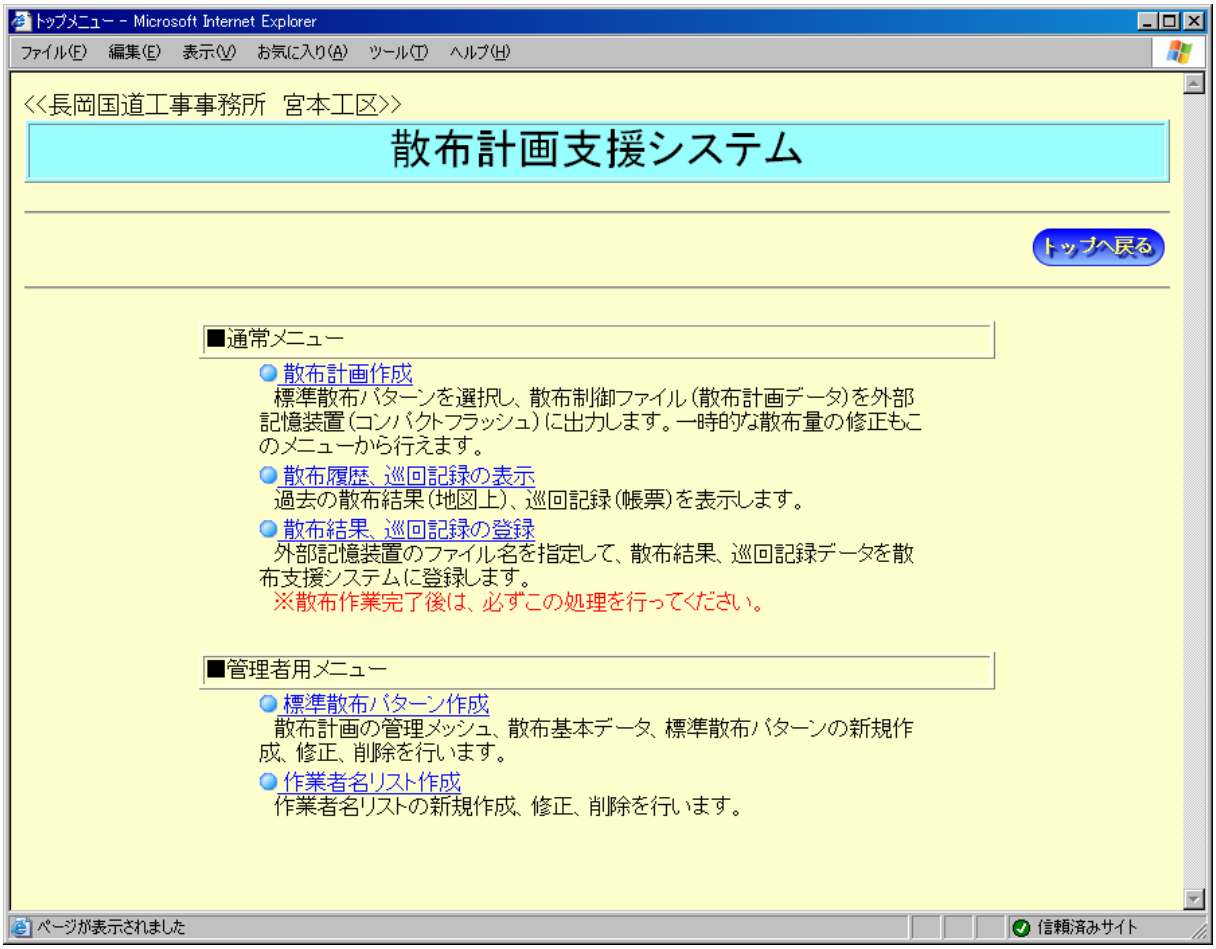
画　面　名	地区選択メニュー		
概要	ユーザ毎に利用できる地区名のリンクを表示する。地区名リンクをクリックしてトップメニューへ遷移する。		
遷移元	ログイン	遷移先	トップメニュー



項目名	型	説　　明
地区名	リンク	利用できる地区名を表示する

表 3-4　画面レイアウト

画　面　名	トップメニュー		
概要	地区名選択後に表示される画面。リンクによりそれぞれの処理画面へ遷移する。		
遷移元	地区選択メニュー	遷移先	リンク毎の遷移先



項目名	型	説　　明
散布計画作成	リンク	散布計画作成へのリンク
散布履歴、巡回記録の表示	リンク	散布履歴または巡回記録の表示へのリンク
散布結果、巡回記録の登録	リンク	散布結果または巡回記録の登録へのリンク
標準散布パターン作成	リンク	標準散布パターン作成へのリンク
作業者名リスト作成	リンク	作業者名リストへのリンク

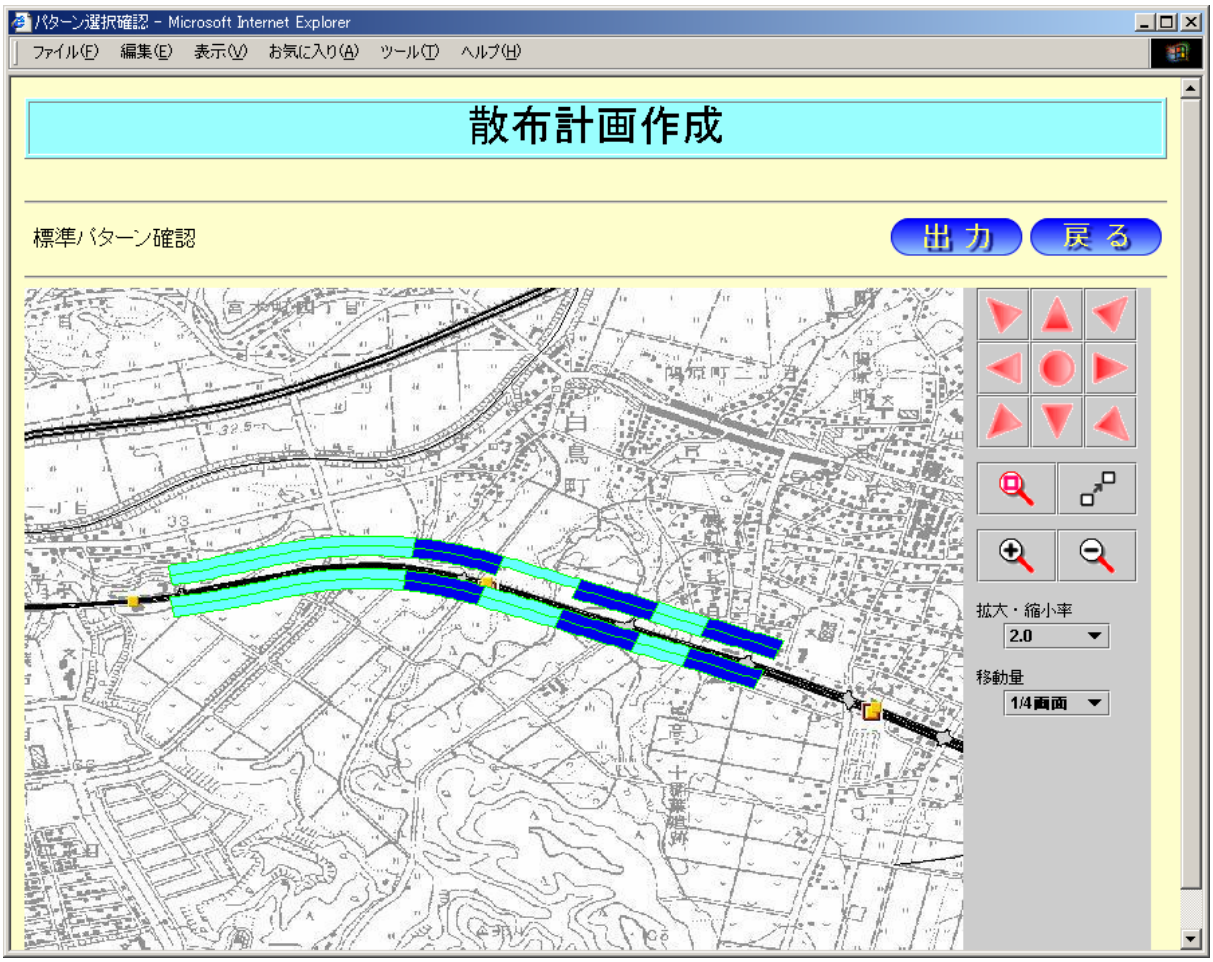
表 3-5　画面レイアウト

画　面　名	散布計画作成：標準パターン名一覧		
概要	散布計画として登録されている標準散布パターンの一覧を表示する。		
遷移元	トップメニュー	遷移先	パターン選択確認

項目名	型	説　　明
標準パターン名	リンク	登録されている標準パターン名
説明	テキスト	標準パターンの説明文
戻る	ボタン	遷移元へ戻るためのボタン

表 3-6　画面レイアウト

画　面　名	散布計画作成：標準パターン確認（縮小画面）		
概要	登録されている標準散布パターンを背景地図上に散布メッシュとして表示する 地図画面はアプレットにより、縮小画面、拡大画面、散布情報修正ダイアログの表示を切替える		
遷移元	標準パターン名一覧	遷移先	散布計画出力完了

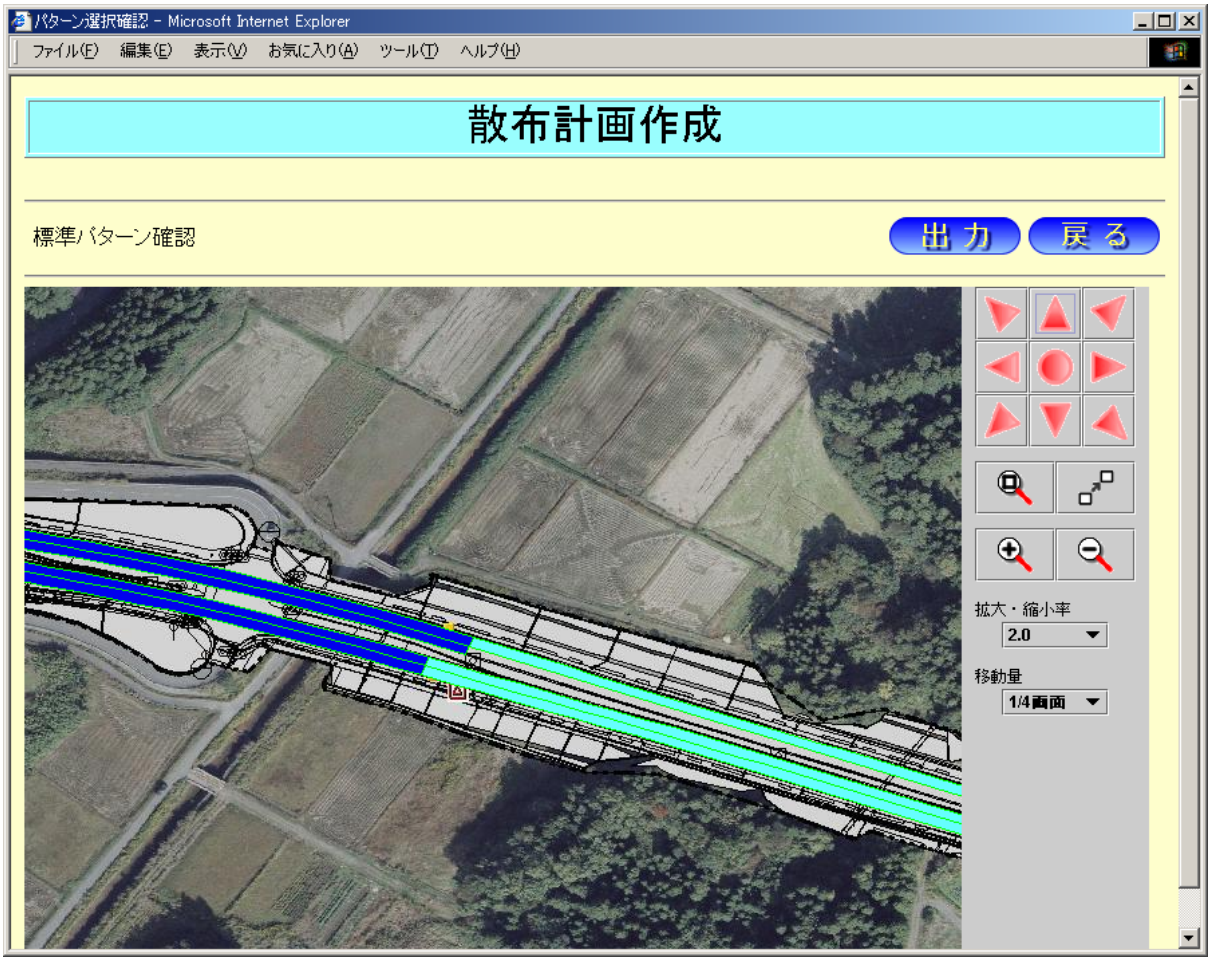


項目名	型	説　　明
出力	ボタン	散布計画ファイルを出力（保存）する
戻る	ボタン	遷移元へ戻るためのボタン
地図	アプレット	背景地図と散布情報を表示する

表 3-7　画面レイアウト

画　面　名	散布計画作成：標準パターン確認（拡大画面）
概要	登録されている標準散布パターンを背景地図上に散布メッシュとして表示する 地図画面はアプレットにより、縮小画面、拡大画面、散布情報修正ダイアログの表示を切替える

遷移元	標準パターン名一覧	遷移先	散布計画出力完了

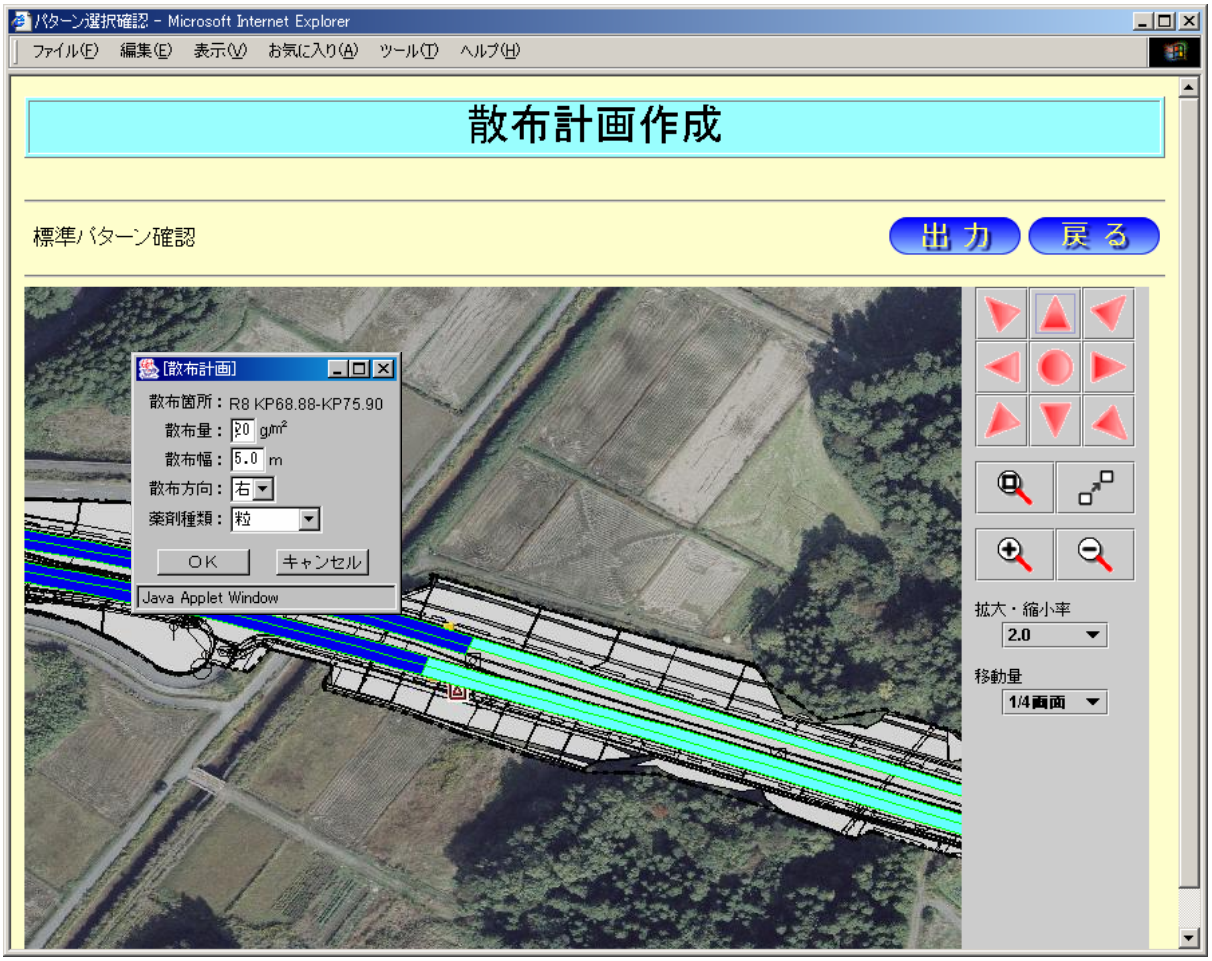


項目名	型	説　　明
出力	ボタン	散布計画ファイルを出力（保存）する
戻る	ボタン	遷移元へ戻るためのボタン
地図	アプレット	背景地図と散布情報を表示する

表 3-8　画面レイアウト

画　面　名	散布計画作成：標準パターン確認（散布情報修正ダイアログ）
概要	登録されている標準散布パターンを背景地図上に散布メッシュとして表示する 地図画面はアプレットにより、縮小画面、拡大画面、散布情報修正ダイアログの表示を切替える

遷移元	標準パターン名一覧	遷移先	散布計画出力完了



項目名	型	説　　明
出力	ボタン	散布計画ファイルを出力（保存）する
戻る	ボタン	遷移元へ戻るためのボタン
地図	アプレット	背景地図と散布情報を表示する

表 3-9　画面レイアウト

画　面　名	散布計画作成：散布計画出力完了
概要	散布計画ファイルを外部記録メディアに出力する画面。出力する標準散布パターン名とその説明を表示する。ＯＫボタン押下で散布計画ファイルがダウンロードされる。

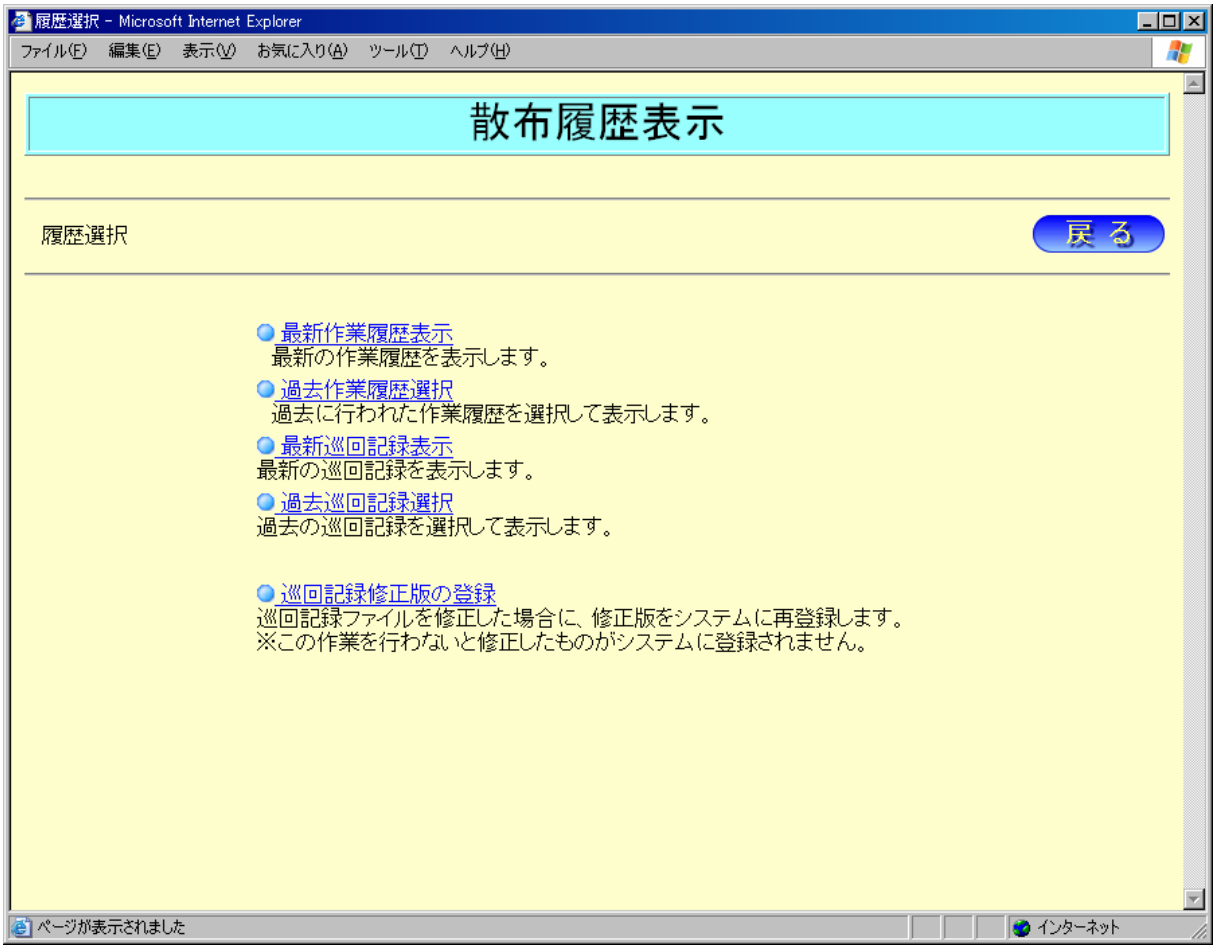
遷移元	標準パターン確認	遷移先	一覧またはトップメニュー

項目名	型	説　　明
標準パターン名	テキスト	出力された標準パターン名
ＯＫ	ボタン	散布計画ファイルをダウンロードするボタン
戻る	ボタン	標準パターン確認へ戻るボタン
一覧へ戻る	ボタン	パターン名一覧へ戻るボタン
メニューへ	ボタン	トップメニューへ戻るボタン

表 3-10　画面レイアウト

画　面　名	散布履歴表示：履歴選択
概要	散布履歴を表示するための表示項目を選択するサブメニュー。

遷移元	トップメニュー	遷移先	リンク毎の遷移先



項目名	型	説　　明
最新作業履歴表示	リンク	最新の作業履歴を散布履歴画面に表示するためのリンク
過去作業履歴選択	リンク	作業履歴一覧を表示するためのリンク
最新気温データ表示	リンク	最新の気温データを気温データ画面に表示するためのリンク
過去気温データ選択	リンク	気温データ一覧を表示するためのリンク

表 3-11　画面レイアウト

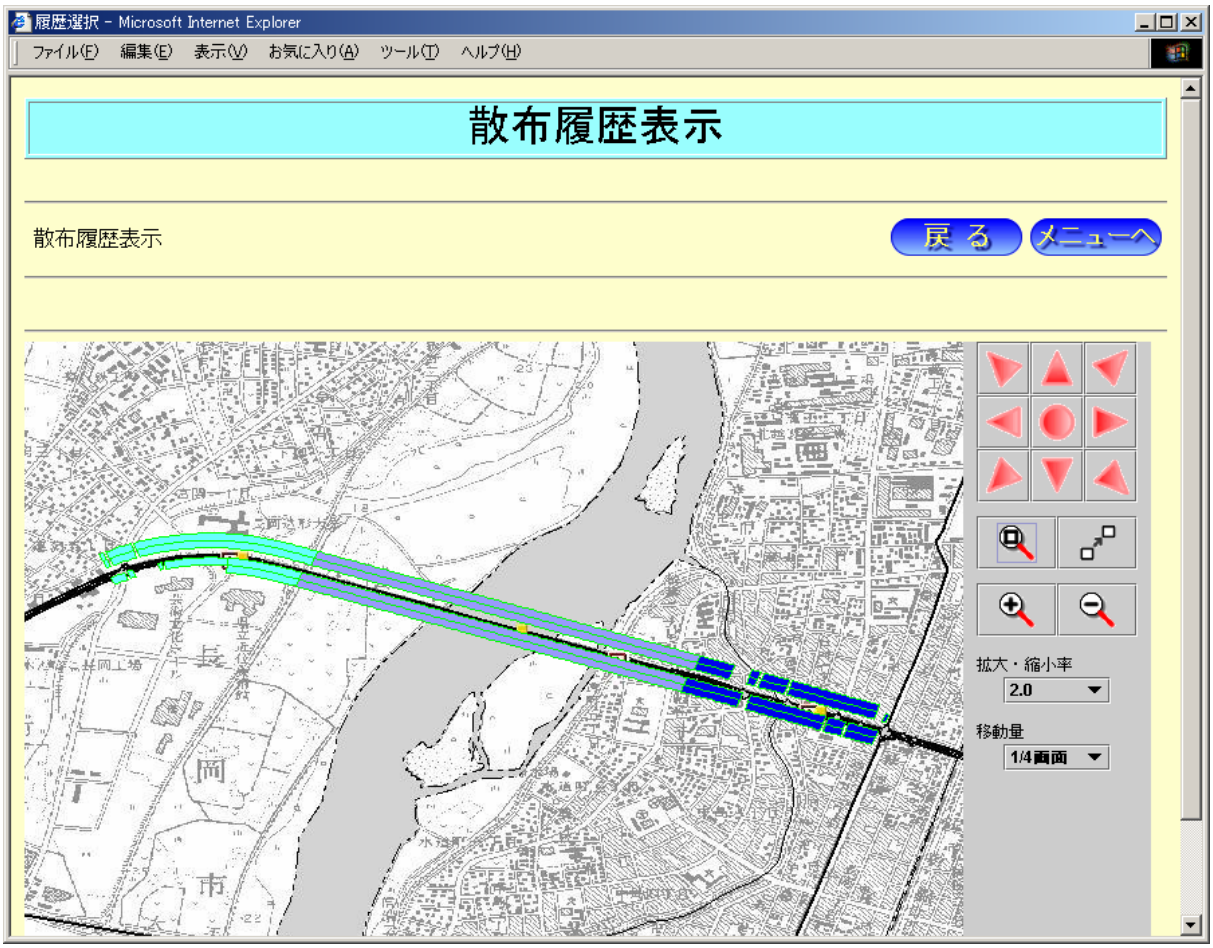
画　面　名	散布履歴表示：散布履歴日時選択
概要	散布履歴を表示するために、表示する日時を選択するための散布履歴一覧画面。

遷移元	履歴選択	遷移先	散布履歴表示

項目名	型	説　　明
作業日	リンク	作業実施日
作業時間	テキスト	作業開始時間～作業終了時間
作業者名	テキスト	作業実施者の名前
戻る	ボタン	遷移元へ戻るためのボタン

表 3-12　画面レイアウト

画　面　名	散布履歴表示：散布履歴表示（縮小画面）		
概要	登録されている作業履歴を背景地図上に散布メッシュとして表示する 地図画面はアプレットにより、縮小画面、拡大画面、散布情報表示ダイアログの表示を切替える		
遷移元	履歴選択または散布履歴日時選択	遷移先	散布履歴日時選択またはトップメニュー



項目名	型	説　　明
戻る	ボタン	遷移元へ戻るためのボタン
メニューへ	ボタン	トップメニューへ戻るためのボタン
地図	アプレット	背景地図と散布履歴情報を表示する

表 3-13　画面レイアウト

画　面　名	散布履歴表示：散布履歴表示（拡大画面）		
概要	登録されている作業履歴を背景地図上に散布メッシュとして表示する 地図画面はアプレットにより、縮小画面、拡大画面、散布情報表示ダイアログの表示を切替える		
遷移元	履歴選択または散布履歴日時選択	遷移先	散布履歴日時選択またはトップメニュー

項目名	型	説　　明
戻る	ボタン	遷移元へ戻るためのボタン
メニューへ	ボタン	トップメニューへ戻るためのボタン
地図	アプレット	背景地図と散布履歴情報を表示する

表 3-14　画面レイアウト

画　面　名	散布履歴表示：散布履歴表示（散布履歴情報表示画面）		
概要	登録されている作業履歴を背景地図上に散布メッシュとして表示する 地図画面はアプレットにより、縮小画面、拡大画面、散布情報表示ダイアログの表示を切替える		
遷移元	履歴選択または散布履歴日時選択	遷移先	散布履歴日時選択またはトップメニュー

項目名	型	説　　明
戻る	ボタン	遷移元へ戻るためのボタン
メニューへ	ボタン	トップメニューへ戻るためのボタン
地図	アプレット	背景地図と散布履歴情報を表示する

表 3-15　画面レイアウト

画　面　名	散布履歴表示：巡回記録日時選択		
概要	巡回記録を表示するために、表示する日時を選択するための巡回記録一覧画面。		
遷移元	履歴選択	遷移先	巡回記録表示（Excel）

項目名	型	説　　明
巡回日	リンク	巡回実施日
巡回時間	テキスト	巡回開始時間～巡回終了時間
運転者名	テキスト	巡回実施者の名前
戻る	ボタン	遷移元へ戻るためのボタン

表 3-16　画面レイアウト

画　面　名	散布履歴表示：巡回記録表示（Excel）		
概要	登録されている巡回記録を Excel に表示する。Excel ウィンドウを新規にオープンする。		
遷移元	履歴選択または巡回記録日時選択	遷移先	なし

項目名	型	説　　明

表 3-17　画面レイアウト

画　面　名	散布履歴表示：巡回記録修正版登録		
概要	巡回記録を Excel 上で修正した場合に、修正版をサーバー側へアップロードし登録する。		
遷移元	履歴選択	遷移先	巡回記録修正版登録完了

項目名	型	説　　明
参照	ボタン	巡回記録ファイルを選択するボタン
巡回日	テキスト	巡回日
巡回時間	テキスト	巡回時間
巡回記録ファイル名	テキスト	巡回記録ファイル名
送信	ボタン	散布作業記録ファイルをシステムに登録するボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-18　画面レイアウト

画　面　名	散布履歴表示：巡回記録修正版登録完了
概要	巡回記録を Excel 上で修正したファイルをサーバー上へアップロードし、正常にサーバーに登録されたときに表示される画面。

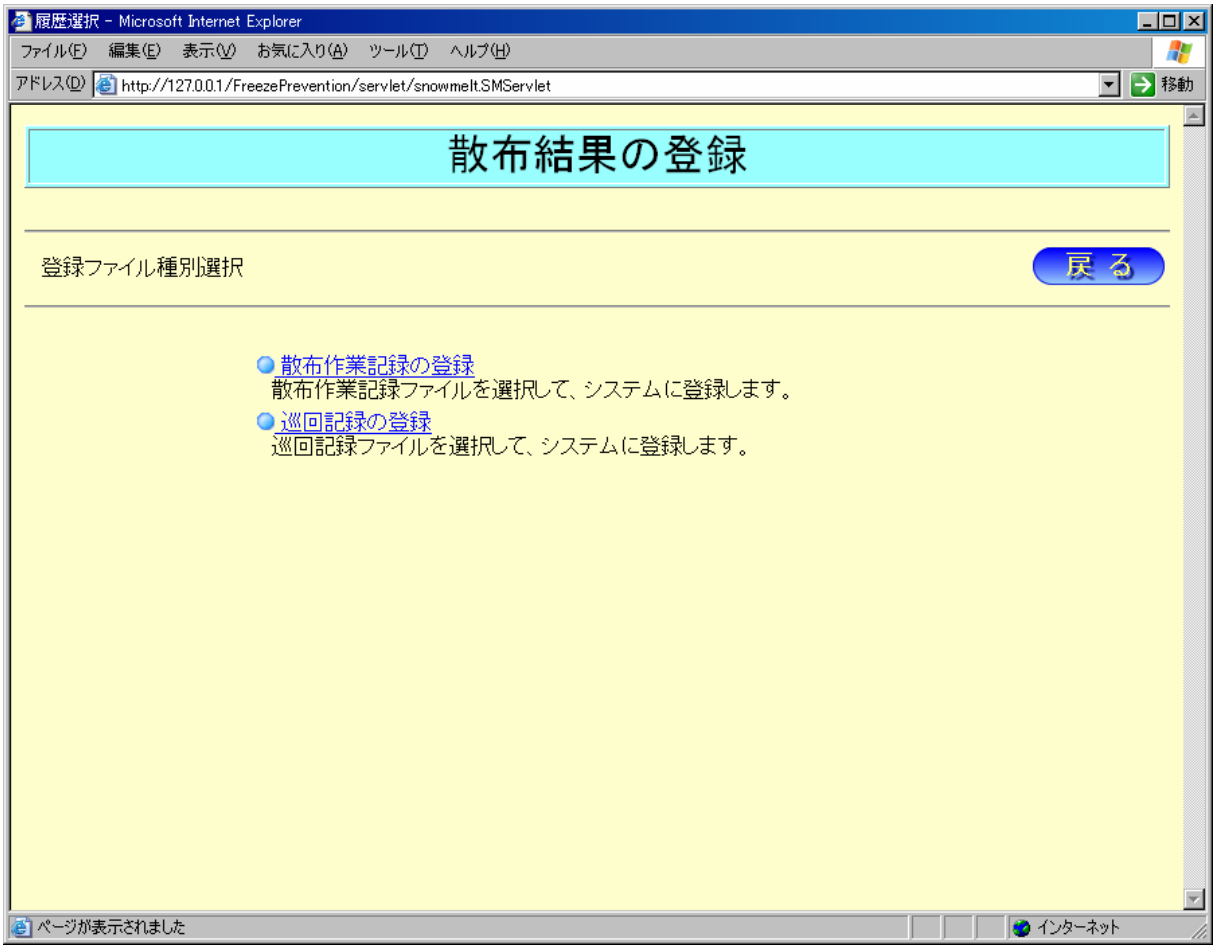
遷移元	巡回記録修正版登録	遷移先	履歴選択またはトップメニュー

項目名	型	説　　明
戻る	ボタン	遷移元へ戻るためのボタン
メニューへ	ボタン	トップメニューへ戻るためのボタン

表 3-19　画面レイアウト

画　面　名	散布結果の登録：登録ファイル種別選択
概要	散布作業記録ファイルを登録するか気温データファイルを登録するかを選択するサブメニュー。

遷移元	トップメニュー	遷移先	リンク毎の遷移先



項目名	型	説　　明
散布作業記録の登録	リンク	散布作業記録ファイル選択画面へのリンク
巡回記録の登録	リンク	巡回記録ファイル選択画面へのリンク
戻る	ボタン	遷移元へ戻るためのボタン

表 3-20　画面レイアウト

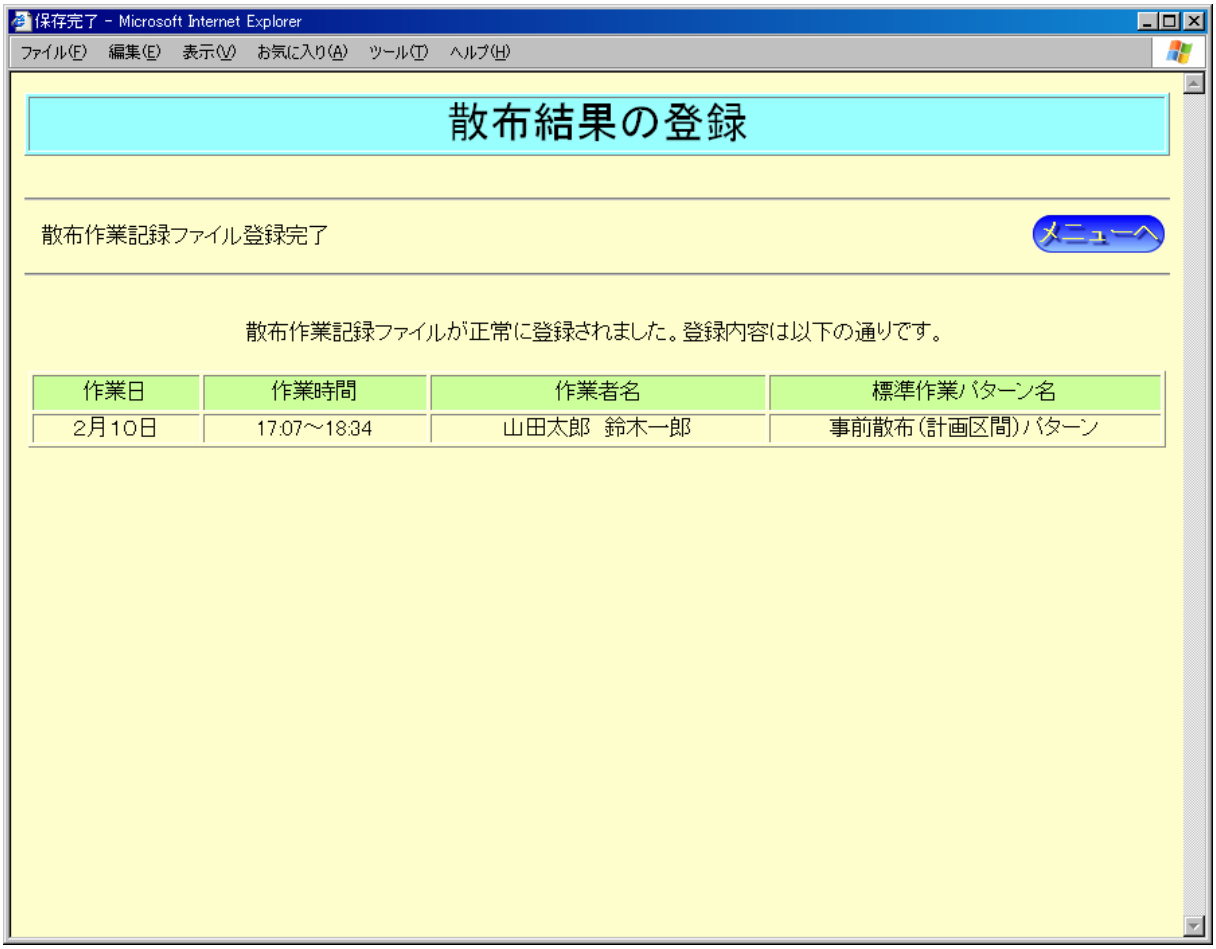
画　面　名	散布結果の登録：散布作業記録ファイル選択
概要	散布作業記録データを本システムに登録するため、散布車 PC から得られた散布作業記録ファイルを選択する。

遷移元	登録ファイル種別選択	遷移先	散布作業記録ファイル登録完了

項目名	型	説　　明
参照	ボタン	散布作業記録ファイルを選択するボタン
送信	ボタン	散布作業記録ファイルをシステムに登録するボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-21　画面レイアウト

画　面　名	散布結果の登録：散布作業記録ファイル登録完了		
概要	散布結果ファイルが正常にシステムに正常に登録された場合に表示する。登録したファイルの情報を表示する。		
遷移元	散布作業記録保存ファイル選択	遷移先	トップメニュー

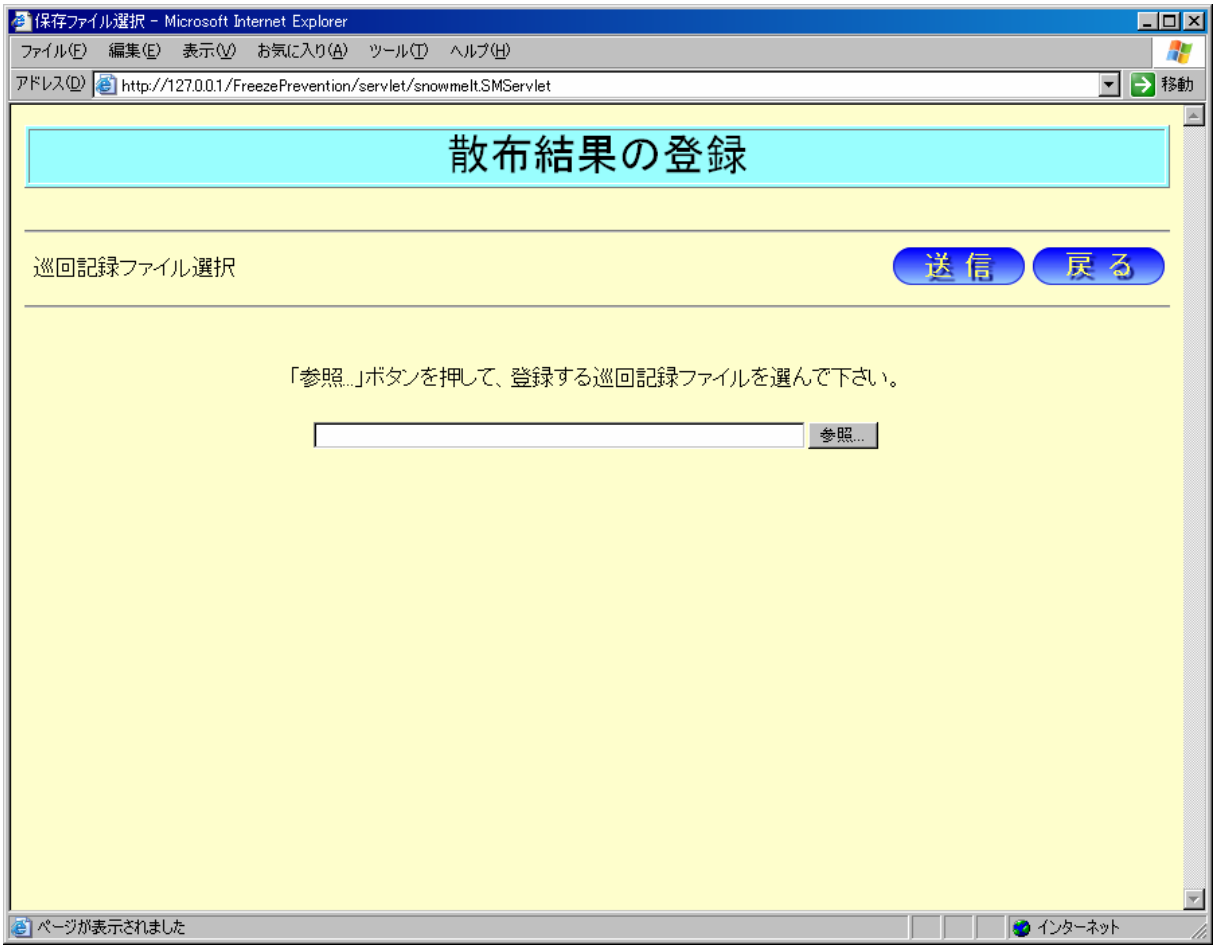


項目名	型	説　　明
作業日	テキスト	散布作業記録ファイルの作業日
作業時間	テキスト	散布作業記録ファイルの作業開始時刻～作業終了時刻
作業者名	テキスト	散布作業記録ファイルの作業者名
標準作業パターン名	テキスト	散布作業記録ファイルの標準作業パターン名
メニューへ	ボタン	トップメニューへ戻るためのボタン

表 3-22　画面レイアウト

画　面　名	散布結果の登録：巡回記録ファイル選択
概要	巡回記録データを本システムに登録するため、巡回記録ファイルを選択する。

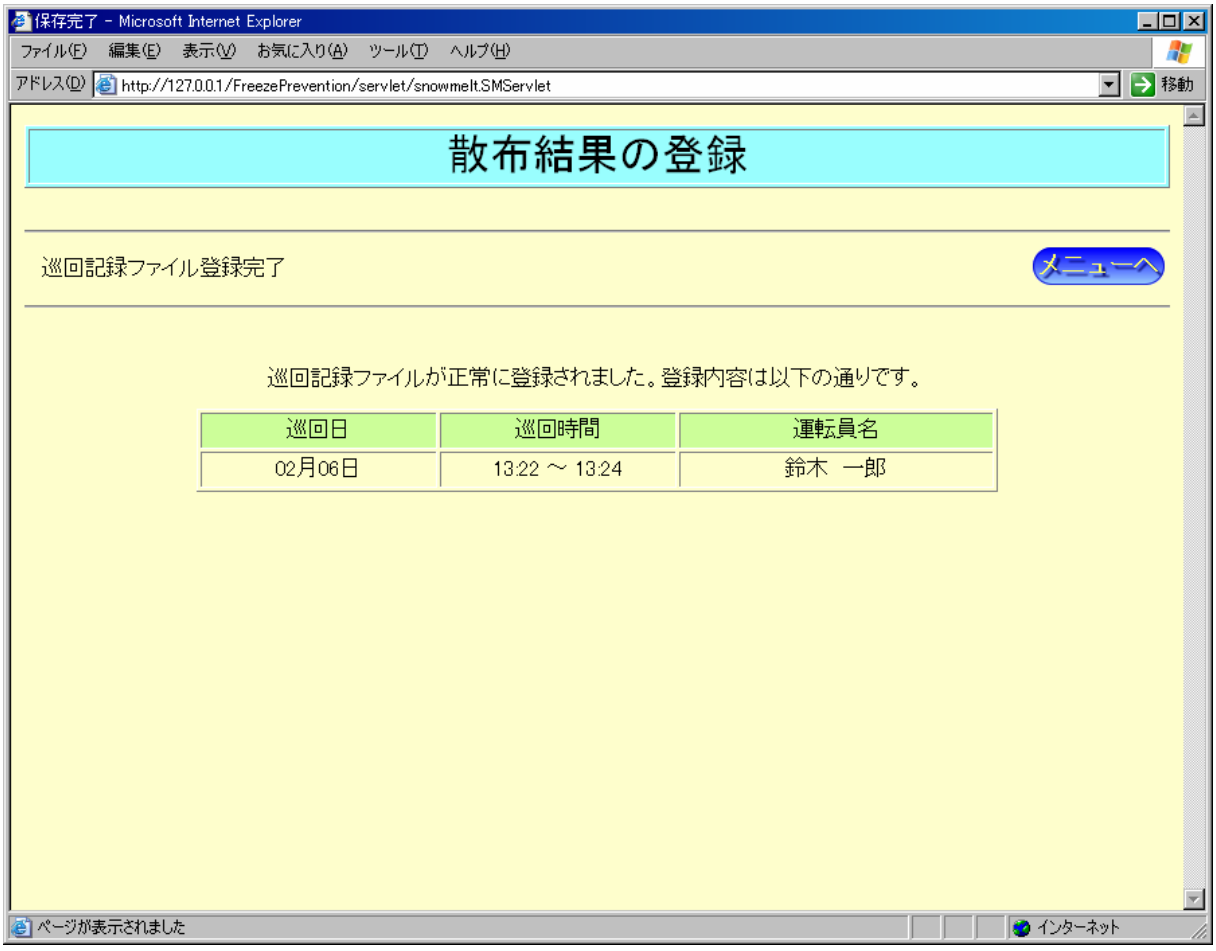
遷移元	登録ファイル種別選択	遷移先	巡回記録登録完了



項目名	型	説　　明
参照	ボタン	巡回記録ファイルを選択するボタン
送信	ボタン	巡回記録ファイルをシステムに登録するボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-23　画面レイアウト

画　面　名	散布結果の登録：巡回記録ファイル保存完了		
概要	巡回記録ファイルが正常にシステムに正常に保存された場合に表示する。保存したファイルの情報を表示する。		
遷移元	巡回記録ファイル選択	遷移先	トップメニュー

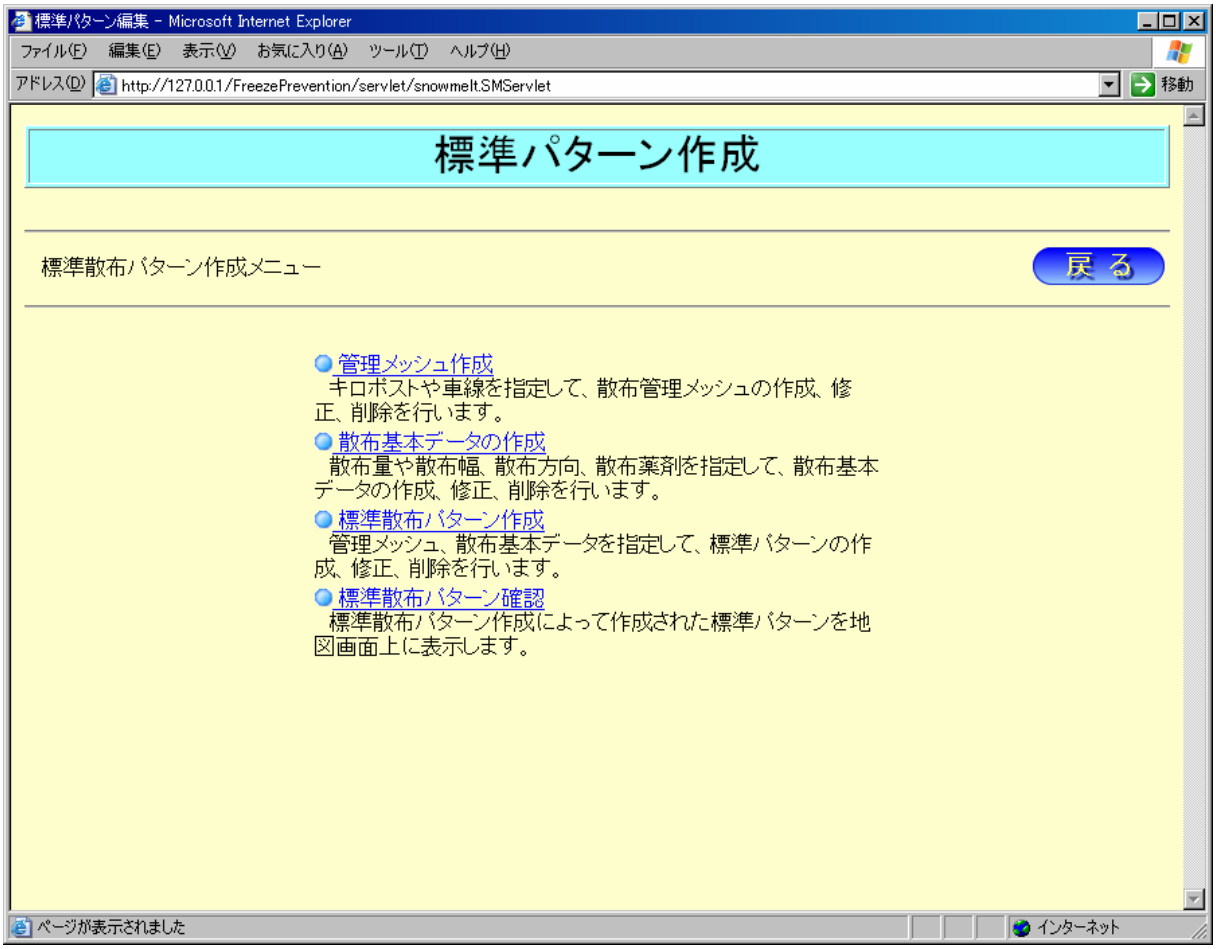


項目名	型	説　　明
巡回日	テキスト	巡回日
巡回時間	テキスト	巡回時間
運転員名	テキスト	巡回を実施した運転員の名前
メニューへ	ボタン	トップメニューへ戻るためのボタン

表 3-24　画面レイアウト

画　面　名	標準散布パターン作成：標準散布パターン作成メニュー
概要	管理メッシュや散布基本データ、標準散布パターンを作成するためのサブメニュー。

遷移元	トップメニュー	遷移先	リンク毎の遷移先



項目名	型	説　　明
管理メッシュ作成	リンク	管理メッシュ作成画面へのリンク
散布基本データ作成	リンク	散布基本データ作成へのリンク
標準パターン作成	リンク	標準パターン作成へのリンク
標準パターン確認	リンク	標準パターン確認へのリンク

表 3-25　画面レイアウト

画　面　名	標準散布パターン作成：管理メッシュ一覧		
概要	編集する管理メッシュの一覧を表示する		
遷移元	標準散布パターン作成メニュー	遷移先	管理メッシュ新規登録または修正

項目名	型	説　　明
メッシュ ID	リンク	修正する散布メッシュのメッシュ ID へのリンク
路線名	テキスト	登録されているメッシュ ID に対する路線名
車線区別	テキスト	登録されているメッシュ ID に対する車線名
開始 KP	テキスト	登録されているメッシュ ID に対する開始 KP
終了 KP	テキスト	登録されているメッシュ ID に対する終了 KP
新規	ボタン	管理メッシュを新規に作成するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン
メニューへ	ボタン	トップメニューへ戻るためのボタン

表 3-26　画面レイアウト

画　面　名	標準散布パターン作成：管理メッシュ新規登録		
概要	管理メッシュの新規登録を行う		
遷移元	管理メッシュ一覧	遷移先	管理メッシュ一覧

項目名	型	説　　明
メッシュ ID	インプットボックス	新規登録する管理メッシュのメッシュ ID
路線名	インプットボックス	新規登録する管理メッシュの路線名
車線区別	ドロップダウン	新規登録する管理メッシュの車線名
開始 KP	インプットボックス	新規登録する管理メッシュの開始 KP
終了 KP	インプットボックス	新規登録する管理メッシュの終了 KP
登録	ボタン	管理メッシュを新規登録するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-27　画面レイアウト

画　面　名	標準散布パターン作成：管理メッシュ修正		
概要	管理メッシュの修正を行う		
遷移元	管理メッシュ一覧	遷移先	管理メッシュ一覧または新規登録

項目名	型	説　　明
メッシュ ID	テキスト	修正する管理メッシュのメッシュ ID
路線名	インプットボックス	修正する管理メッシュの路線名
車線区別	ドロップダウン	修正する管理メッシュの車線名
開始 KP	インプットボックス	修正する管理メッシュの開始 KP
終了 KP	インプットボックス	修正する管理メッシュの終了 KP
修正	ボタン	管理メッシュを修正するためのボタン
複製	ボタン	現在の管理メッシュの複製を作成するためのボタン
削除	ボタン	現在の管理メッシュを削除するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-28　画面レイアウト

画　面　名	標準散布パターン作成：管理メッシュ新規登録		
概要	管理メッシュ修正から複製ボタンにより複製を作成し、その複製から管理メッシュの新規登録を行う		
遷移元	管理メッシュ修正	遷移先	管理メッシュ一覧

項目名	型	説　　明
メッシュ ID	インプットボックス	新規登録する管理メッシュのメッシュ ID
路線名	インプットボックス	新規登録する管理メッシュの路線名
車線区別	ドロップダウン	新規登録する管理メッシュの車線名
開始 KP	インプットボックス	新規登録する管理メッシュの開始 KP
終了 KP	インプットボックス	新規登録する管理メッシュの終了 KP
登録	ボタン	管理メッシュを新規登録するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-29　画面レイアウト

画　面　名	標準散布パターン作成：散布基本データ一覧		
概要	編集する散布基本データの一覧を表示する		
遷移元	標準パターン作成メニュー	遷移先	散布基本データ新規登録または修正

項目名	型	説　　明
データ ID	リンク	修正する散布基本データのデータ ID へのリンク
散布量	テキスト	登録されているデータ ID に対する散布量
散布幅	テキスト	登録されているデータ ID に対する散布幅
散布方向	テキスト	登録されているデータ ID に対する散布方向
散布薬剤	テキスト	登録されているデータ ID に対する散布薬剤
新規	ボタン	散布基本データを新規に作成するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン
メニューへ	ボタン	トップメニューへ戻るためのボタン

表 3-30　画面レイアウト

画　面　名	標準散布パターン作成：散布基本データ新規登録		
概要	散布基本データの新規登録を行う		
遷移元	散布基本データ一覧	遷移先	散布基本データ一覧

項目名	型	説　　明
データ ID	インプットボックス	新規登録する散布基本データのデータ ID
散布量	ドロップダウン	新規登録する散布基本データの散布量
散布幅	ドロップダウン	新規登録する散布基本データの散布幅
散布方向	ドロップダウン	新規登録する散布基本データの散布方向
散布薬剤	ドロップダウン	新規登録する散布基本データの散布薬剤
登録	ボタン	散布基本データを新規登録するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-31　画面レイアウト

画　面　名	標準散布パターン作成：散布基本データ修正		
概要	散布基本データの修正を行う		
遷移元	散布基本データ一覧	遷移先	散布基本データ一覧または新規登録

項目名	型	説　　明
散布量	ドロップダウン	修正する散布基本データの散布量
散布幅	ドロップダウン	修正する散布基本データの散布幅
散布方向	ドロップダウン	修正する散布基本データの散布方向
散布薬剤	ドロップダウン	修正する散布基本データの散布薬剤
修正	ボタン	散布基本データを修正するためのボタン
複製	ボタン	現在の散布基本データの複製を作成するためのボタン
削除	ボタン	現在の散布基本データを削除するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-32　画面レイアウト

画　面　名	標準散布パターン作成：散布基本データ新規登録		
概要	散布基本データ修正から複製ボタンにより複製を作成し、その複製から散布基本データの新規登録を行う		
遷移元	散布基本データ修正	遷移先	散布基本データ一覧

項目名	型	説　　明
データ ID	インプットボックス	新規登録する散布基本データのデータ ID
散布量	ドロップダウン	新規登録する散布基本データの散布量
散布幅	ドロップダウン	新規登録する散布基本データの散布幅
散布方向	ドロップダウン	新規登録する散布基本データの散布方向
散布薬剤	ドロップダウン	新規登録する散布基本データの散布薬剤
登録	ボタン	散布基本データを新規登録するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-33　画面レイアウト

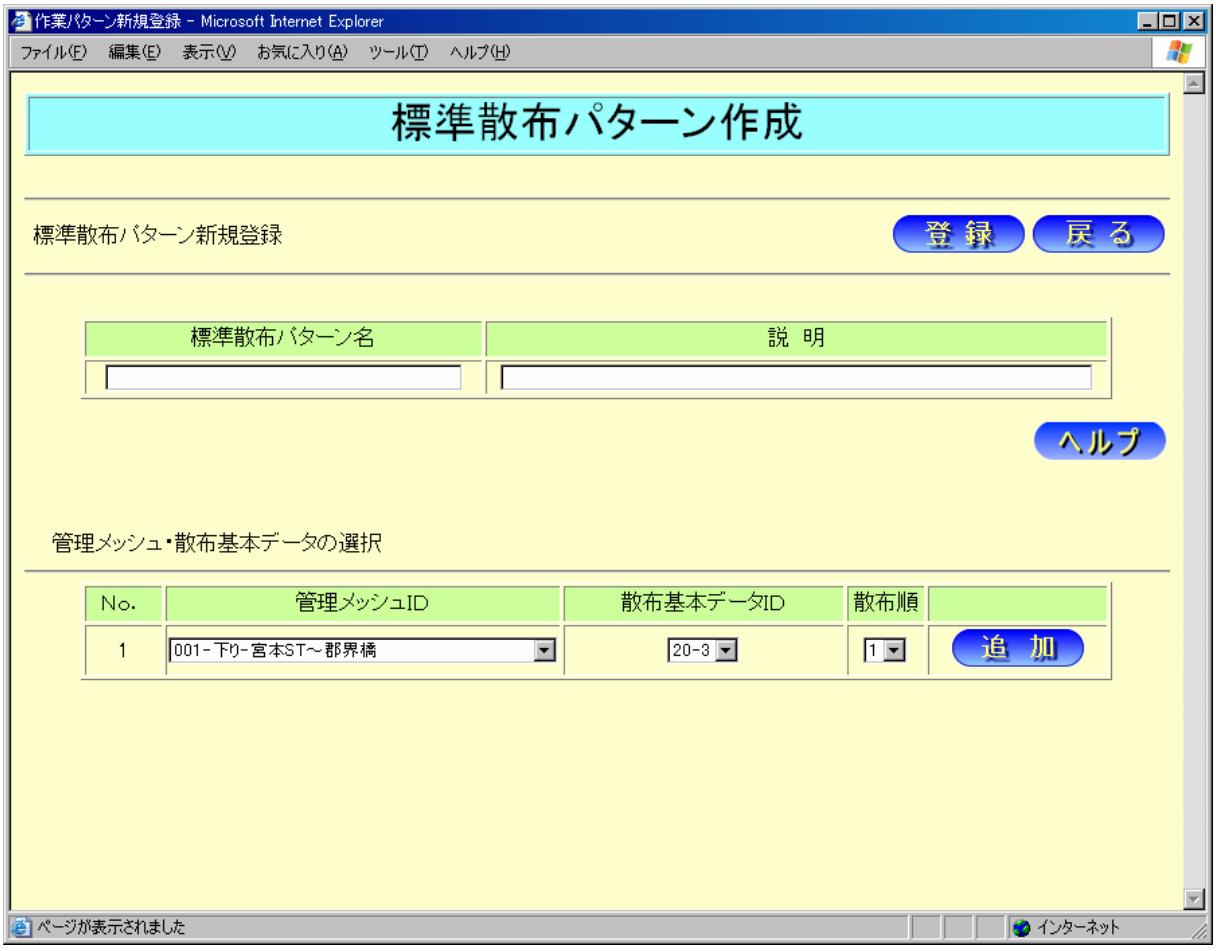
画　面　名	標準散布パターン作成：標準散布パターン一覧		
概要	編集する標準散布パターンの一覧を表示する		
遷移元	標準散布パターン作成メニュー	遷移先	標準散布パターン新規登録または修正

項目名	型	説　　明
標準散布パターン名	リンク	修正する標準散布パターンへのリンク
説明	テキスト	標準散布パターンの説明
総散布量	テキスト	追越車線を除く車線の総散布量計算値
新規	ボタン	標準散布パターンを新規に作成するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン
メニューへ	ボタン	トップメニューへ戻るためのボタン

表 3-34　画面レイアウト

画　面　名	標準散布パターン作成：標準散布パターン新規登録
概要	標準散布パターンの新規登録を行う

遷移元	標準散布パターン一覧	遷移先	標準散布パターン一覧



項目名	型	説　　明
標準散布パターン名	インプットボックス	標準散布パターンの名前
説明	インプットボックス	標準散布パターンの説明文
登録	ボタン	標準散布パターンを新規登録するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

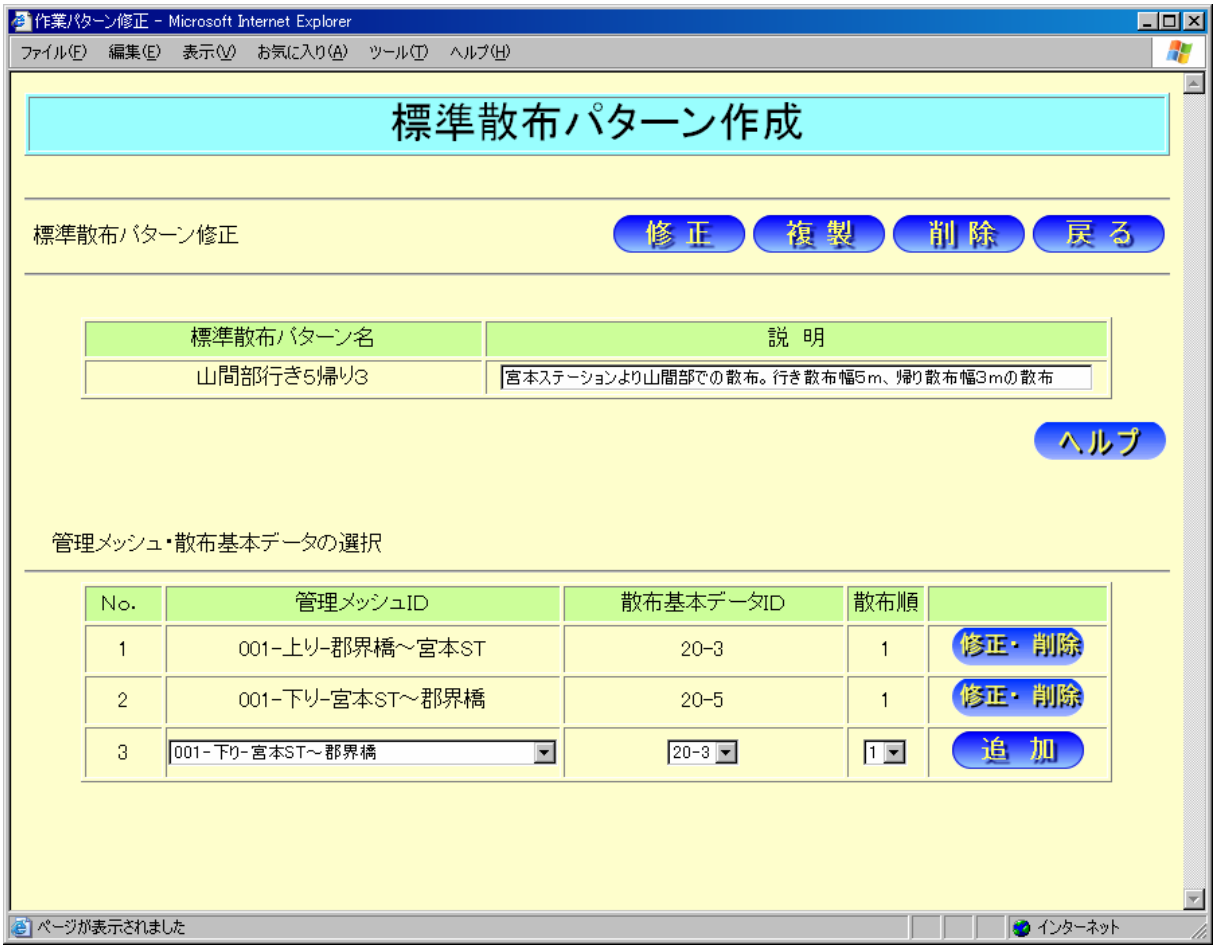
（次ページへ続く）

項目名	型	説　　明
No	テキスト	管理メッシュ、散布基本データの組合せ連番
管理メッシュ ID	テキスト	既作成済みの管理メッシュ ID
散布基本データ ID	テキスト	既作成済みの散布基本データ ID
散布順	テキスト	既作成済みの散布順
管理メッシュ ID	ドロップダウン	これから登録する管理メッシュ ID
散布基本データ ID	ドロップダウン	これから登録する散布基本データ ID
散布順	ドロップダウン	これから登録する散布順
修正・削除	ボタン	既作成済みの管理メッシュ、散布基本データを修正・削除するためのボタン
追加	ボタン	管理メッシュ、散布基本データを追加するためのボタン

表 3-35　画面レイアウト

画　面　名	標準散布パターン作成：標準散布パターン修正
概要	標準散布パターンの修正を行う

遷移元	標準散布パターン一覧	遷移先	標準散布パターン一覧



項目名	型	説　　明
標準散布パターン名	テキスト	標準散布パターンの名前
説明	インプットボックス	標準散布パターンの説明文
修正	ボタン	標準散布パターンを修正するためのボタン
複製	ボタン	標準散布パターンの複製を作成するためのボタン
削除	ボタン	標準散布パターンを削除するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

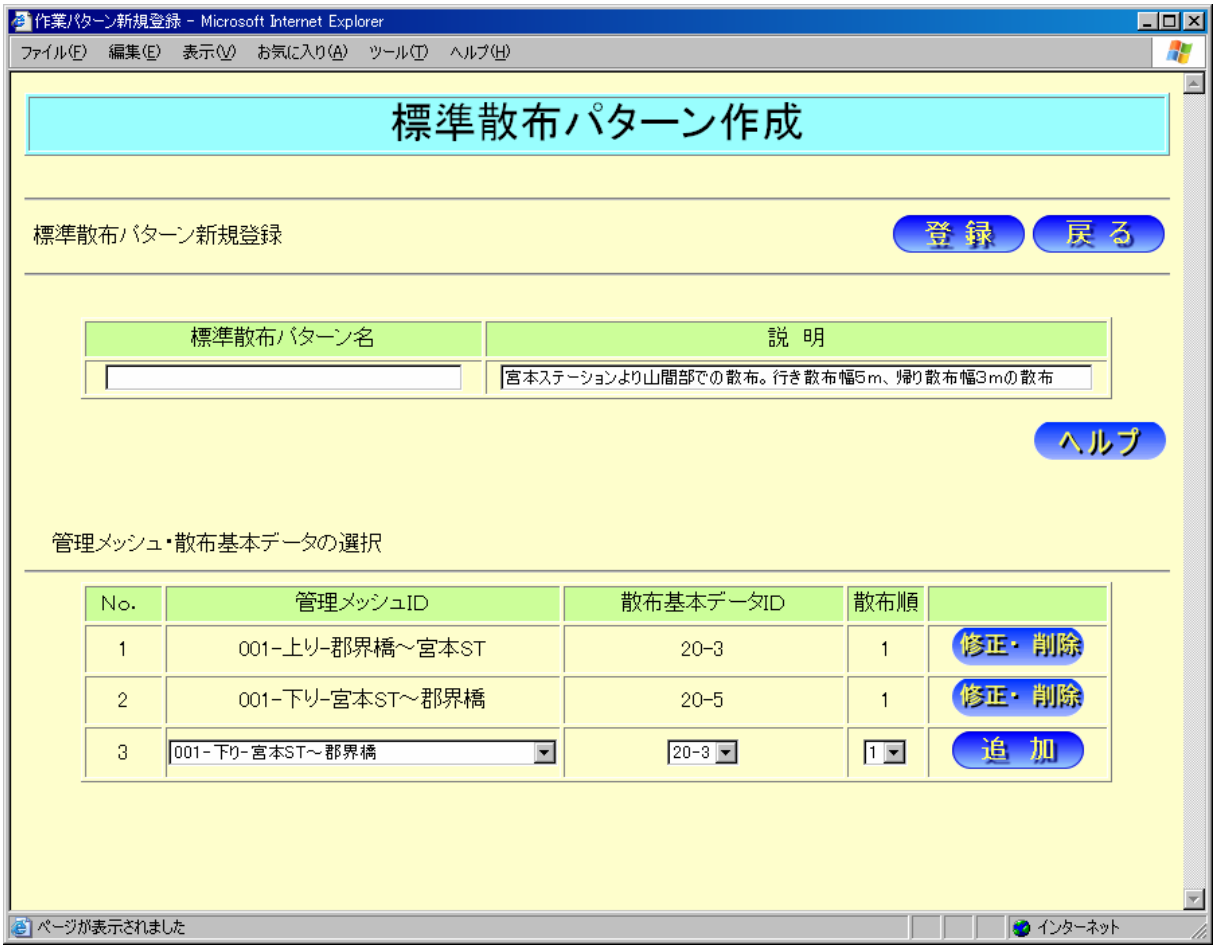
（次ページへ続く）

項目名	型	説　　明
No	テキスト	管理メッシュ、散布基本データの組合せ連番
管理メッシュ ID	テキスト	既作成済みの管理メッシュ ID
散布基本データ ID	テキスト	既作成済みの散布基本データ ID
散布順	テキスト	既作成済みの散布順
管理メッシュ ID	ドロップダウン	追加する管理メッシュ ID
散布基本データ ID	ドロップダウン	追加する散布基本データ ID
散布順	ドロップダウン	追加する散布順
修正・削除	ボタン	既作成済みの管理メッシュ、散布基本データを修正・削除するためのボタン
追加	ボタン	管理メッシュ、散布基本データを追加するためのボタン

表 3-36　画面レイアウト

画　面　名	標準散布パターン作成：標準散布パターン新規登録
概要	標準散布パターン修正から複製ボタンにより複製を作成し、その複製から標準散布パターンの新規登録を行う

遷移元	標準散布パターン修正	遷移先	標準散布パターン一覧



項目名	型	説　　明
標準散布パターン名	インプットボックス	標準散布パターンの名前
説明	インプットボックス	標準散布パターンの説明文
登録	ボタン	標準散布パターンを新規登録するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

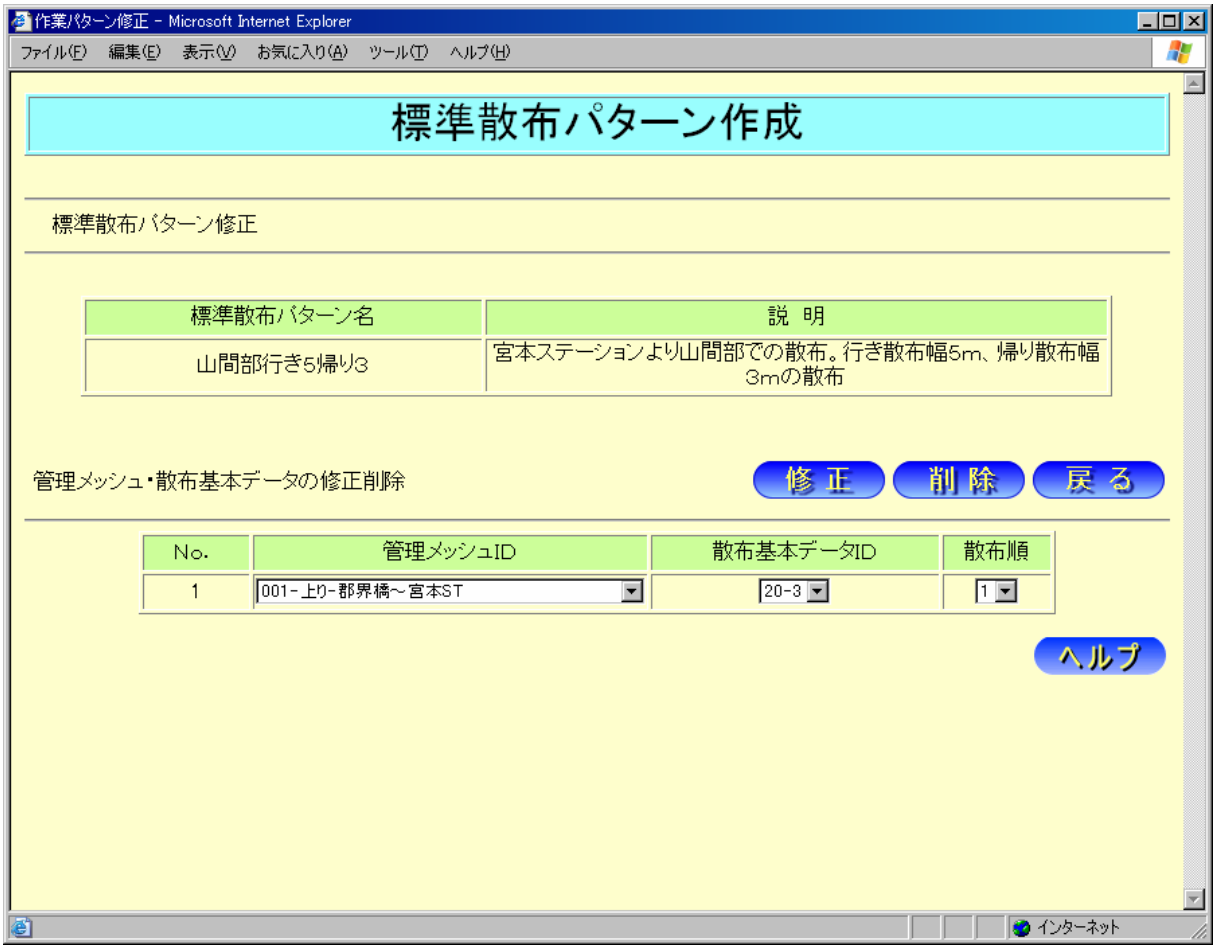
（次ページへ続く）

項目名	型	説　　明
No	テキスト	管理メッシュ、散布基本データの組合せ連番
管理メッシュ ID	テキスト	既作成済みの管理メッシュ ID
散布基本データ ID	テキスト	既作成済みの散布基本データ ID
散布順	テキスト	既作成済みの散布順
管理メッシュ ID	ドロップダウン	これから登録する管理メッシュ ID
散布基本データ ID	ドロップダウン	これから登録する散布基本データ ID
散布順	ドロップダウン	これから登録する散布順
修正・削除	ボタン	既作成済みの管理メッシュ、散布基本データを修正・削除するためのボタン
追加	ボタン	管理メッシュ、散布基本データを追加するためのボタン

表 3-37　画面レイアウト

画　面　名	標準散布パターン作成：標準散布パターン修正（管理メッシュ・散布基本データの修正・削除）
概要	標準散布パターン修正から各管理メッシュ・散布基本データの修正・削除ボタンから遷移し、各管理メッシュ・散布基本データの修正・削除を行う。

遷移元	標準散布パターン修正	遷移先	標準散布パターン修正



項目名	型	説　　明
標準散布パターン名	テキスト	標準散布パターンの名前
説明	テキスト	標準散布パターンの説明文

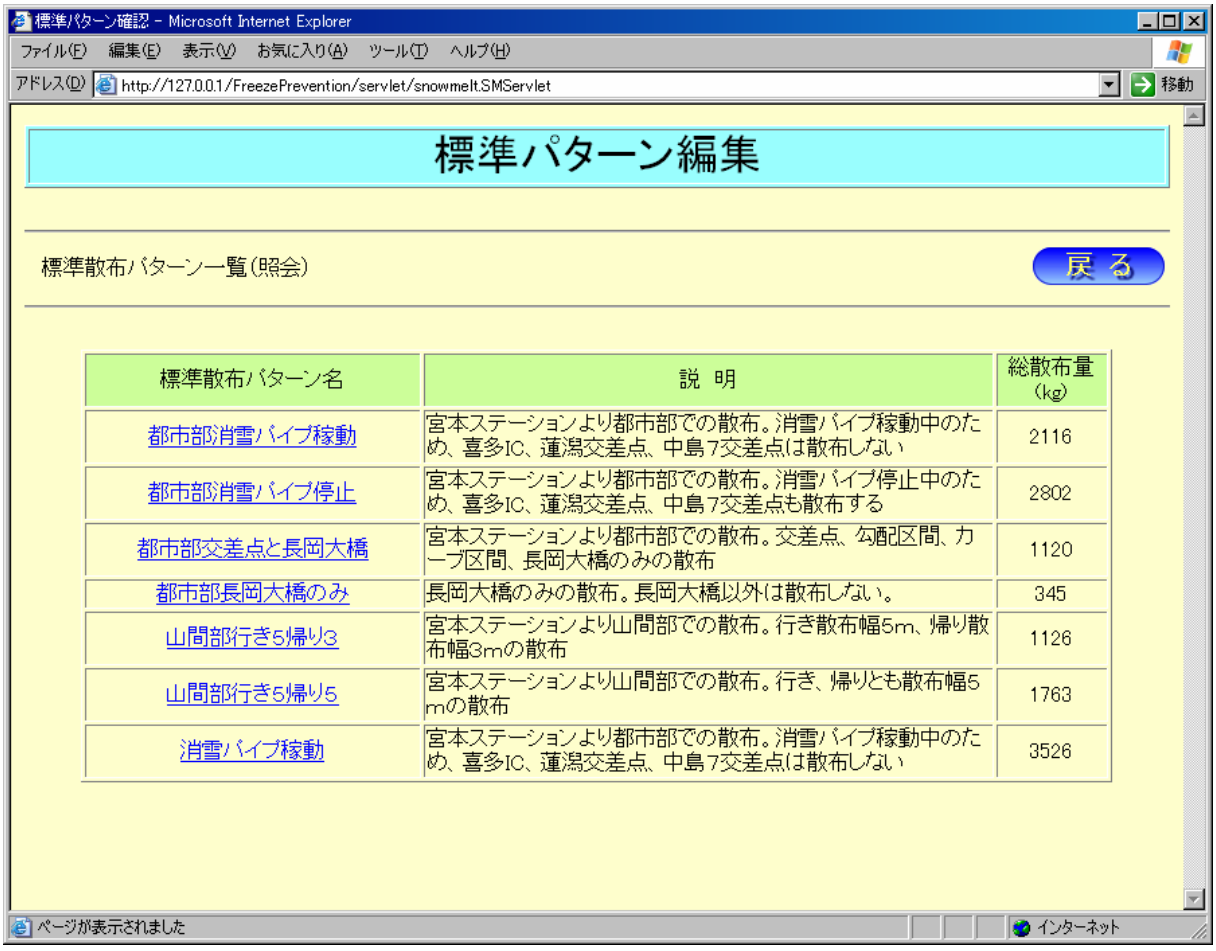
（次ページへ続く）

項目名	型	説　　明
No	テキスト	管理メッシュ、散布基本データの組合せ連番
管理メッシュ ID	ドロップダウン	修正する管理メッシュ ID
散布基本データ ID	ドロップダウン	修正する散布基本データ ID
散布順	ドロップダウン	修正する散布順
修正	ボタン	修正を確定するためのボタン
削除	ボタン	管理メッシュ、散布基本データの組合せを削除するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-38　画面レイアウト

画　面　名	標準散布パターン作成：標準散布パターン一覧
概要	登録されている標準散布パターンの一覧を表示する。

遷移元	標準散布パターン作成メニュー	遷移先	標準散布パターン確認

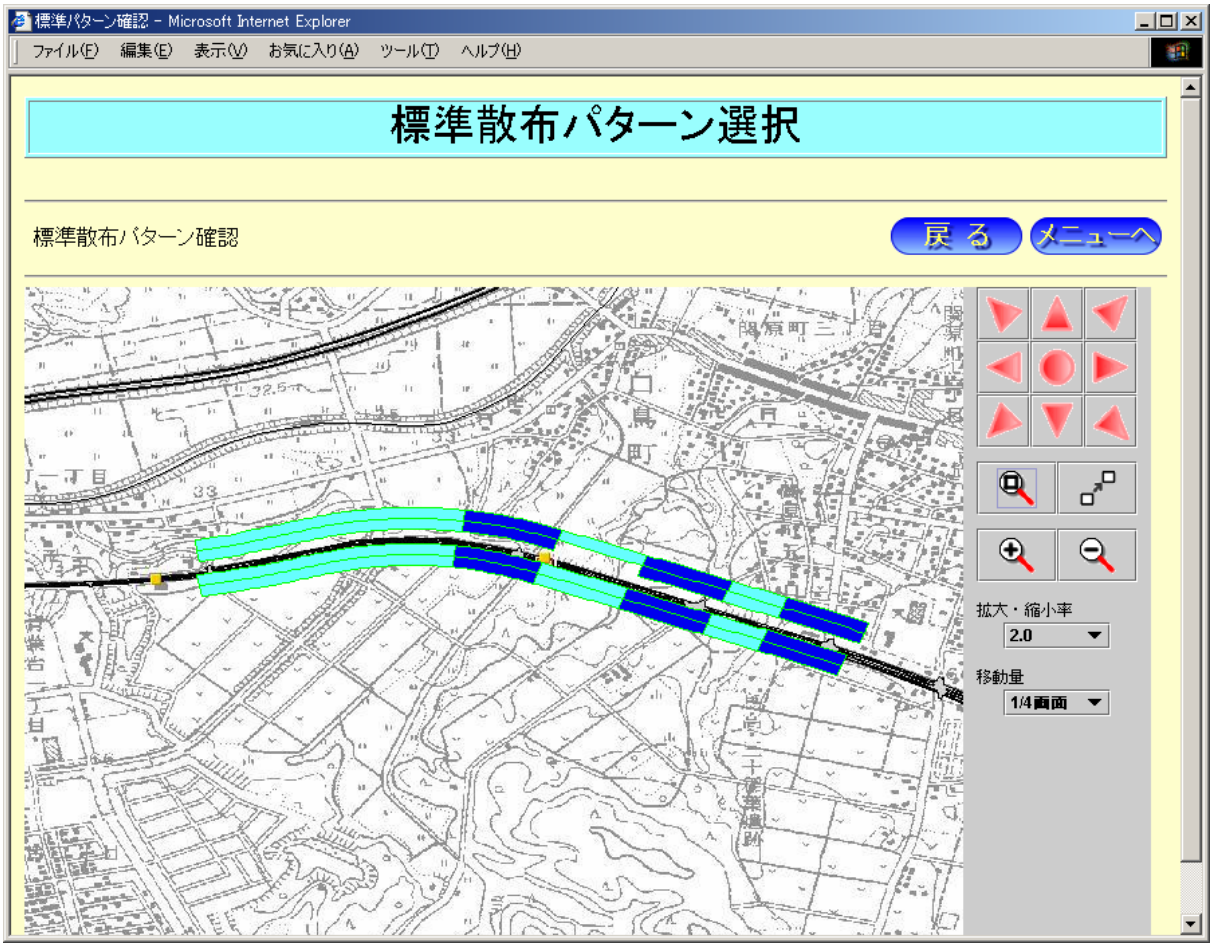


項目名	型	説　　明
標準散布パターン名	リンク	登録されている標準散布パターン名
説明	テキスト	標準散布パターンの説明文
総散布量	テキスト	追越車線を除く車線の総散布量計算値
戻る	ボタン	遷移元へ戻るためのボタン

表 3-39　画面レイアウト

画　面　名	標準散布パターン作成：標準散布パターン確認（縮小画面）
概要	登録されている標準散布パターンを背景地図上に散布メッシュとして表示する 地図画面はアプレットにより、縮小画面、拡大画面、散布情報確認ダイアログの表示を切替える

遷移元	標準散布パターン名一覧	遷移先	標準散布パターン名一覧

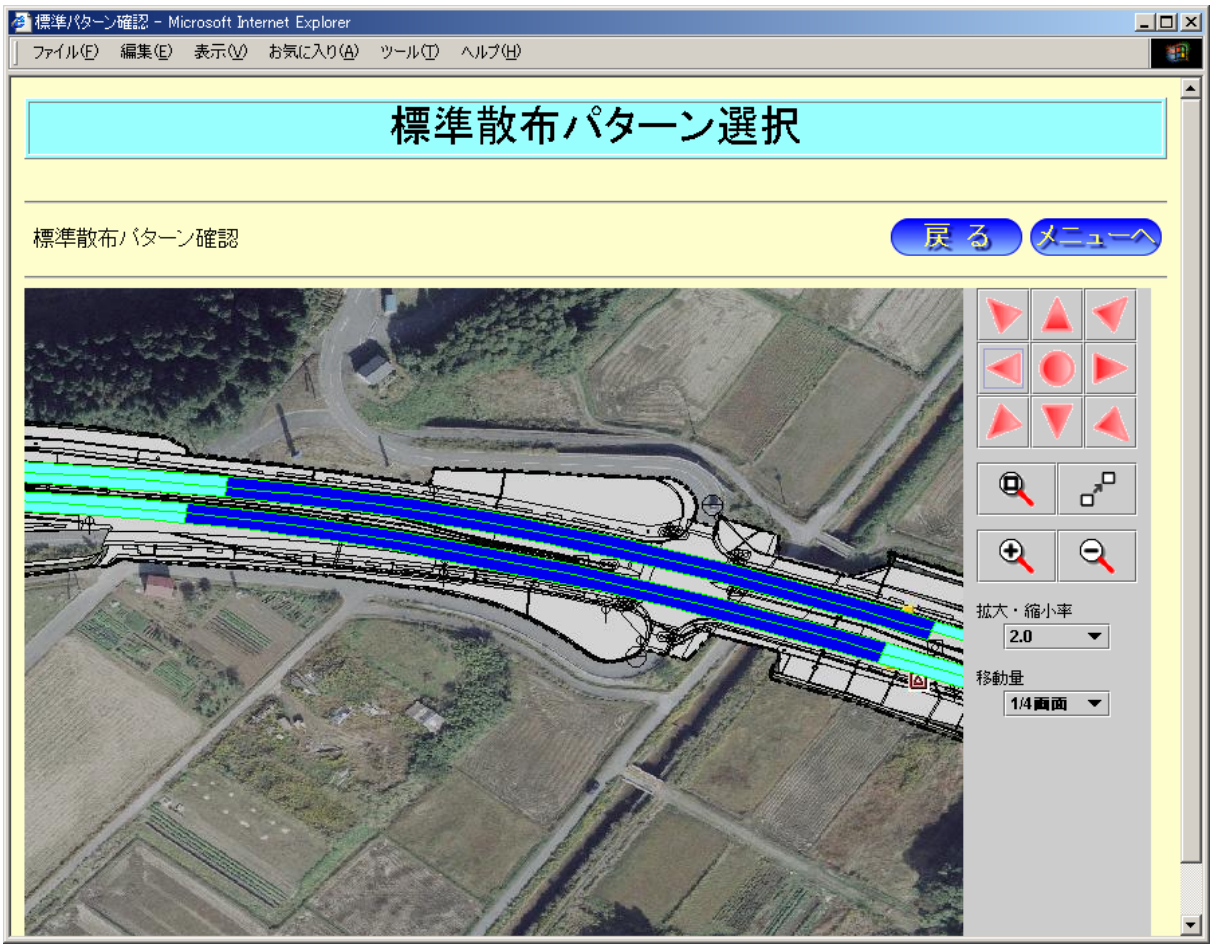


項目名	型	説　　明
戻る	ボタン	遷移元へ戻るためのボタン
メニュー	ボタン	トップメニューへ戻るためのボタン
地図	アプレット	背景地図と散布計画情報を表示する

表 3-40　画面レイアウト

画　面　名	標準散布パターン作成：標準散布パターン確認（拡大画面）
概要	登録されている標準散布パターンを背景地図上に散布メッシュとして表示する 地図画面はアプレットにより、縮小画面、拡大画面、散布情報確認ダイアログの表示を切替える

遷移元	標準散布パターン名一覧	遷移先	標準散布パターン名一覧



項目名	型	説　　明
戻る	ボタン	遷移元へ戻るためのボタン
メニュー	ボタン	トップメニューへ戻るためのボタン
地図	アプレット	背景地図と散布計画情報を表示する

表 3-41　画面レイアウト

画　面　名	標準散布パターン作成：標準散布パターン確認（散布情報確認ダイアログ）
概要	登録されている標準散布パターンを背景地図上に散布メッシュとして表示する 地図画面はアプレットにより、縮小画面、拡大画面、散布情報確認ダイアログの表示を切替える

遷移元	標準散布パターン名一覧	遷移先	標準散布パターン名一覧

項目名	型	説　　明
戻る	ボタン	遷移元へ戻るためのボタン
メニュー	ボタン	トップメニューへ戻るためのボタン
地図	アプレット	背景地図と散布計画情報を表示する

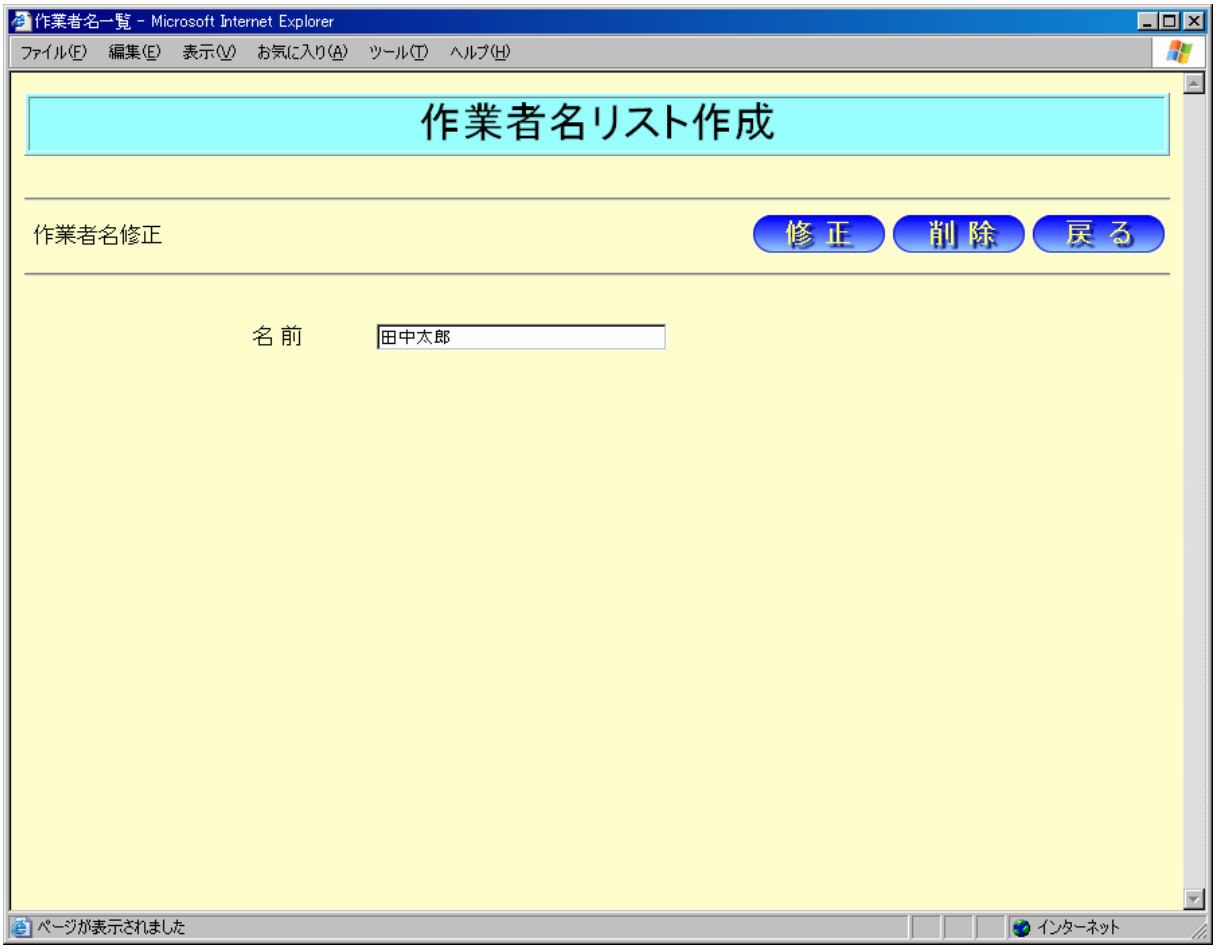
表 3-42　画面レイアウト

画　面　名	作業者名リスト作成：作業者名リスト一覧		
概要	操作支援システムで使用する作業者名の一覧を表示する。		
遷移元	トップメニュー	遷移先	作業者名修正

項目名	型	説　　明
Ｎｏ.	テキスト	連番
名前	テキスト	作業者名
修正・削除	ボタン	作業者名の修正・削除へ遷移するためのボタン
追加	ボタン	作業者名を追加するためのボタン
出力	ボタン	作業者名リストをファイルに出力するためのボタン
戻る	ボタン	遷移元へ戻るためのボタン

表 3-43　画面レイアウト

画　面　名	作業者名リスト作成：作業者名修正		
概要	作業者名の修正・削除を行う。		
遷移元	作業者名リスト一覧	遷移先	作業者名リスト一覧



項目名	型	説　　明
名前	インプットボックス	名前の入力フィールド
修正	ボタン	名前を修正するボタン
削除	ボタン	名前を削除するボタン
戻る	ボタン	遷移元へ戻るためのボタン

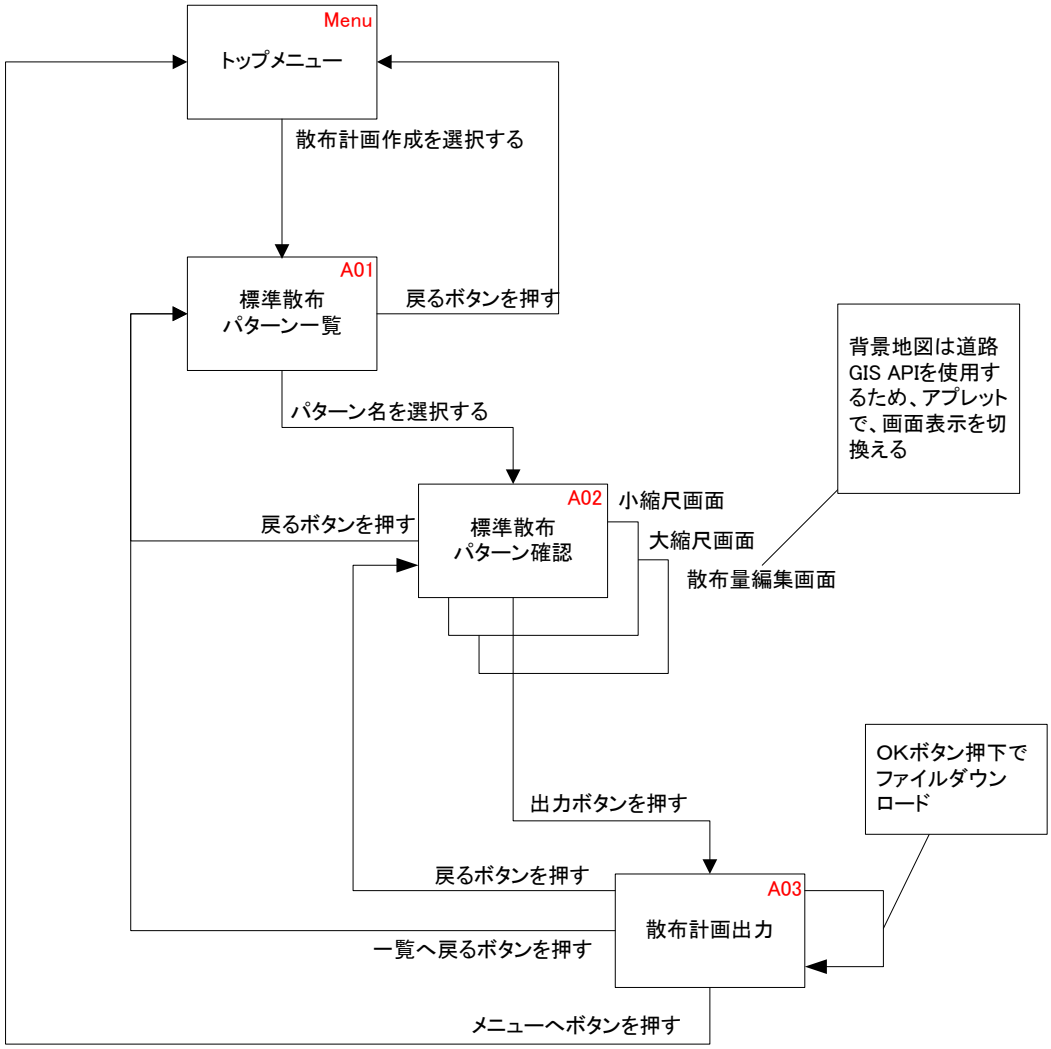


図 3-22　散布計画作成画面遷移図

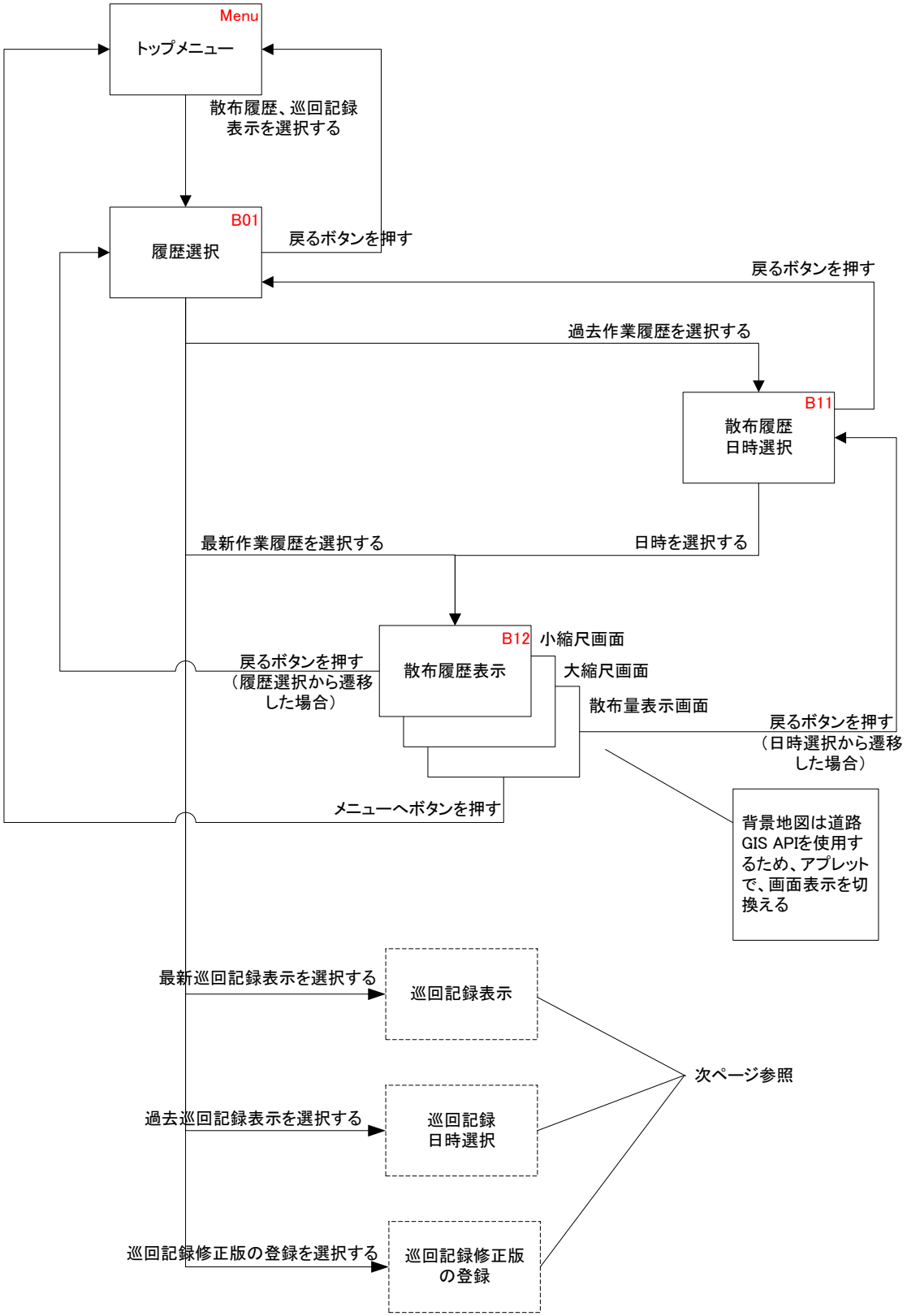


図 3-23　散布履歴表示画面遷移図（その１）

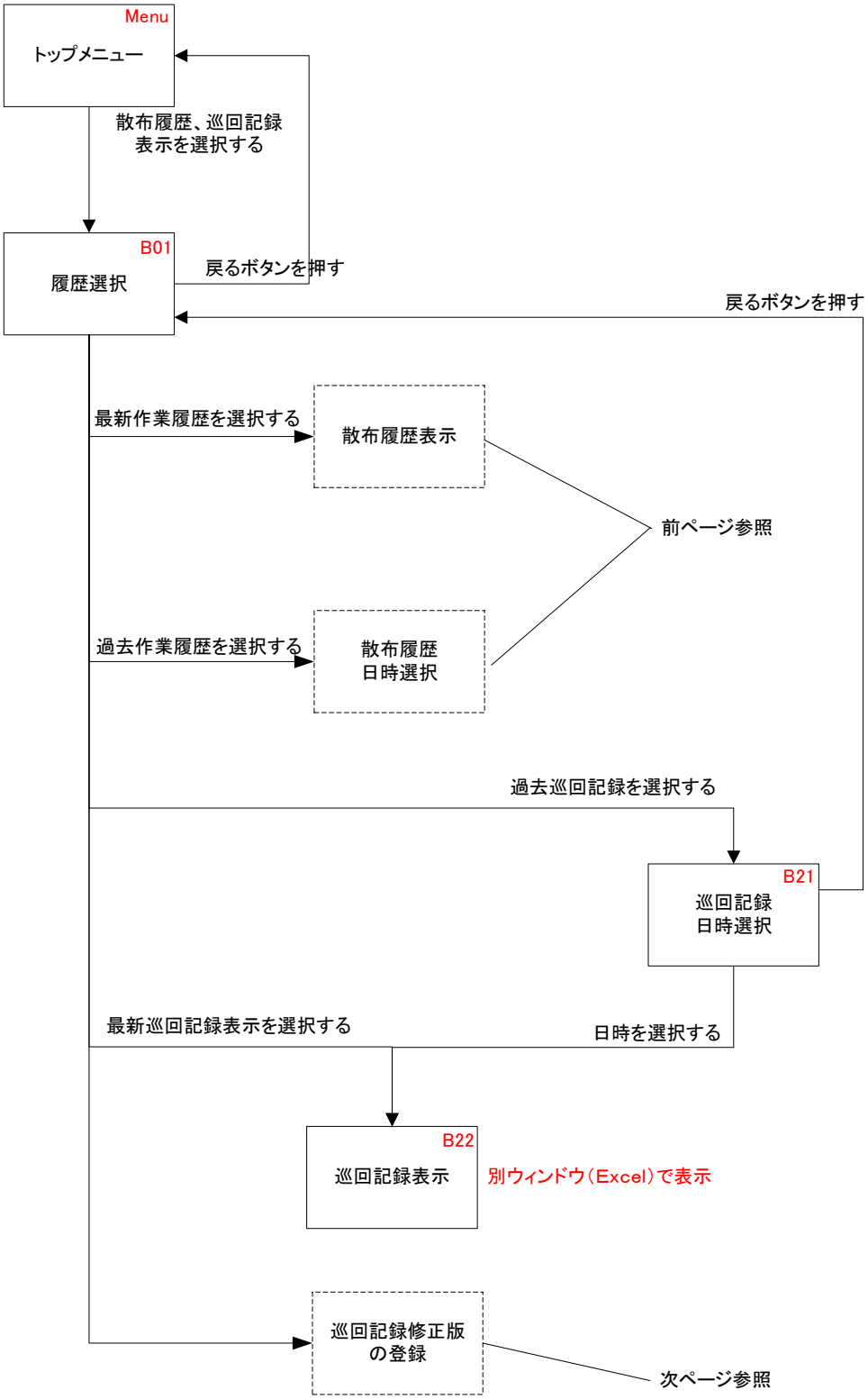


図 3-24　散布履歴表示画面遷移図（その２）

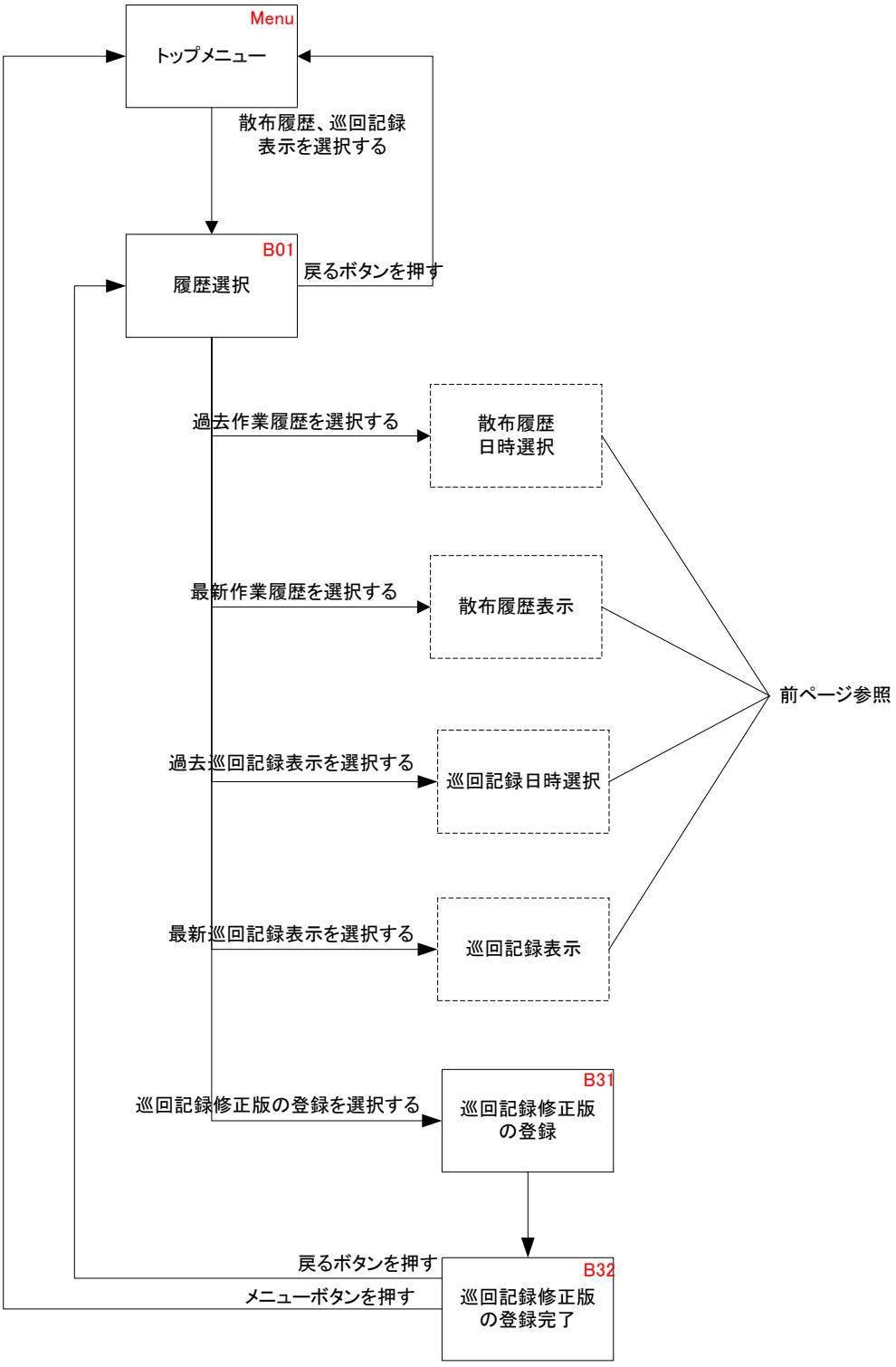


図 3-25　散布履歴表示画面遷移図（その３）

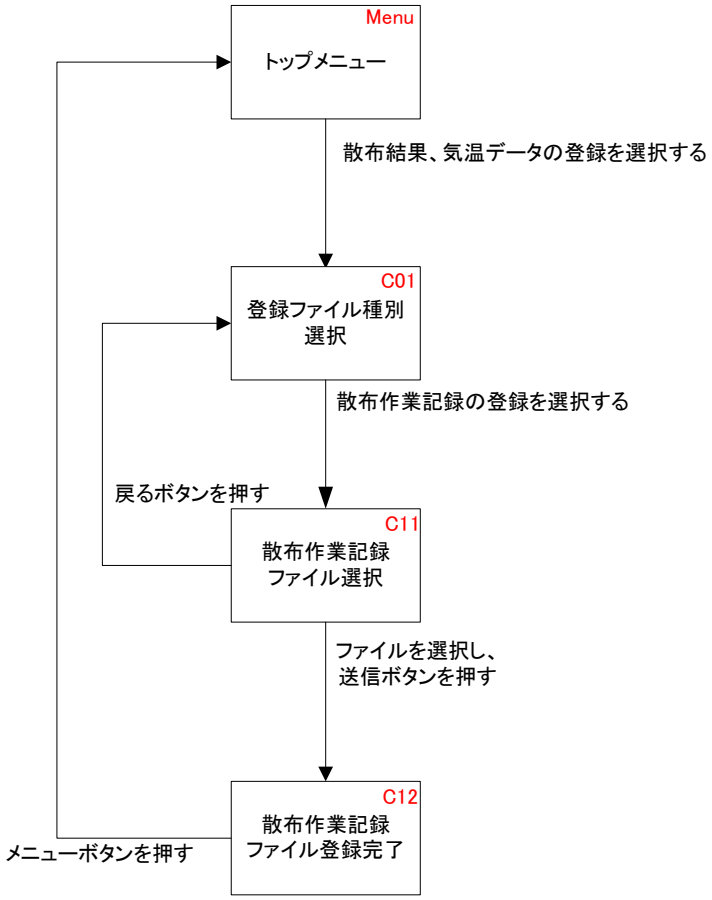


図 3-26　散布結果、巡回記録の登録遷移図（その１）

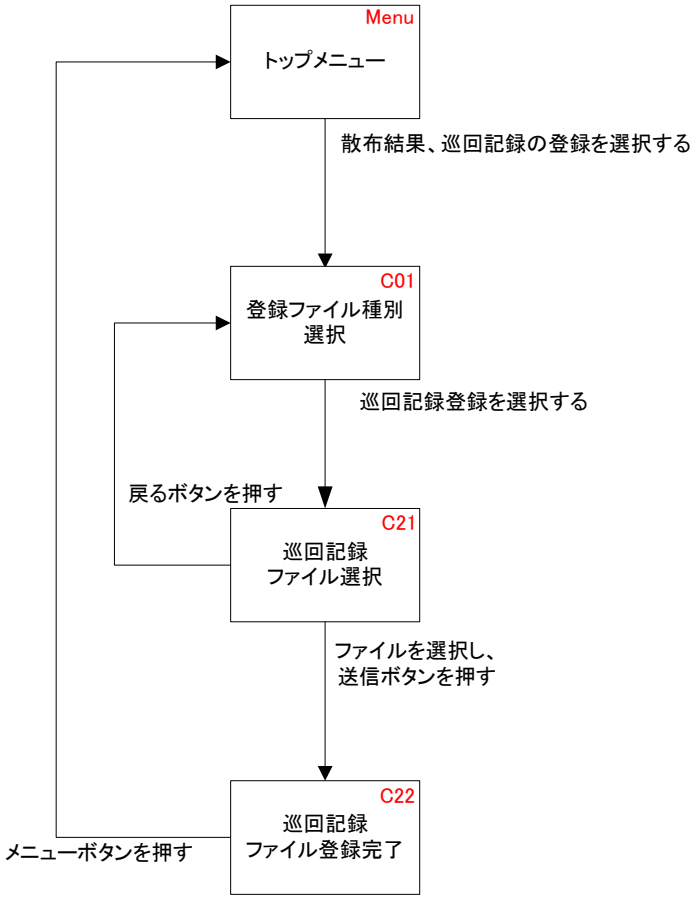


図 3-27　散布結果、巡回記録の登録遷移図（その２）

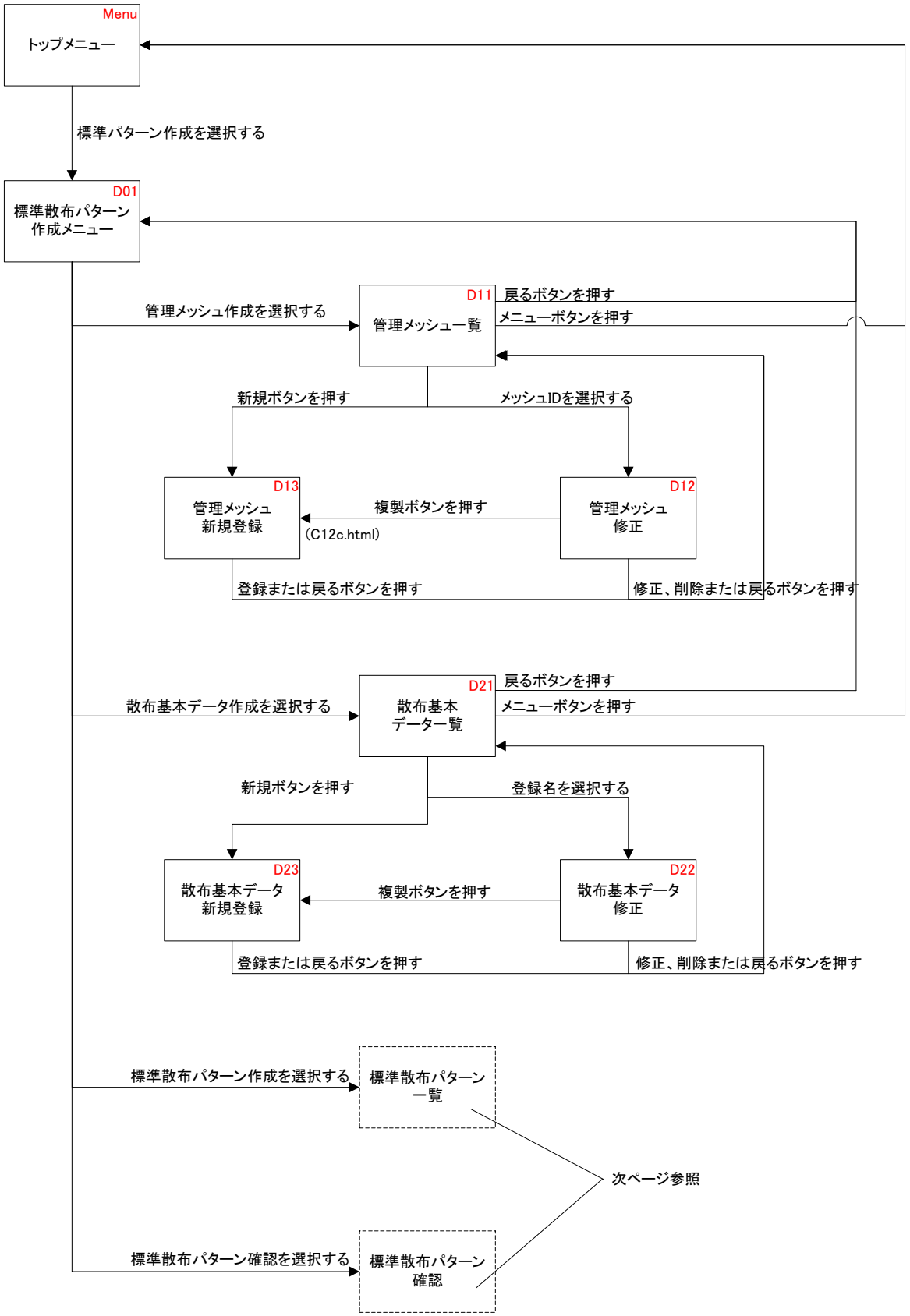


図 3-28　標準散布パターン作成遷移図（その１）

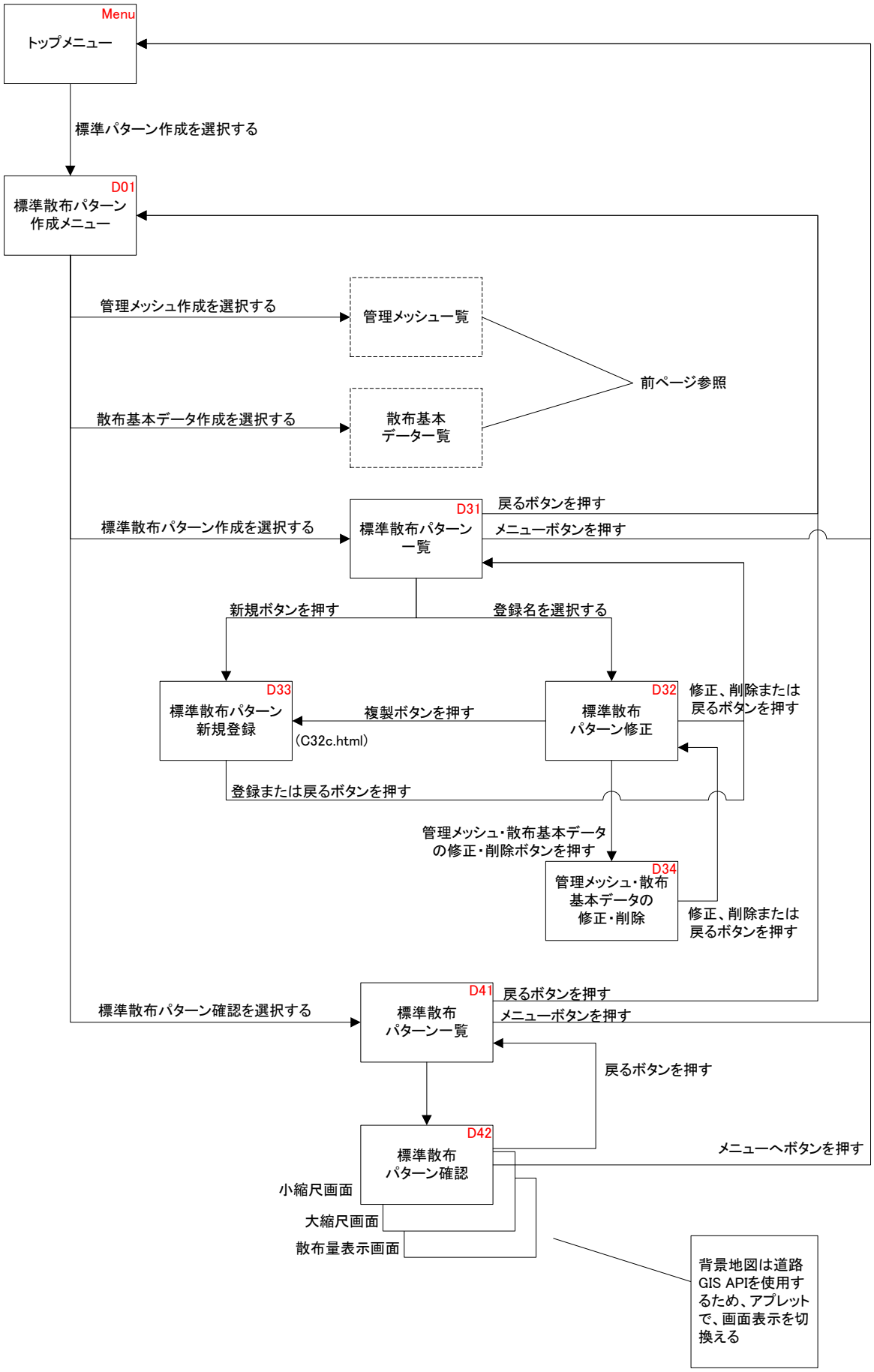


図 3-29　標準散布パターン作成遷移図（その２）

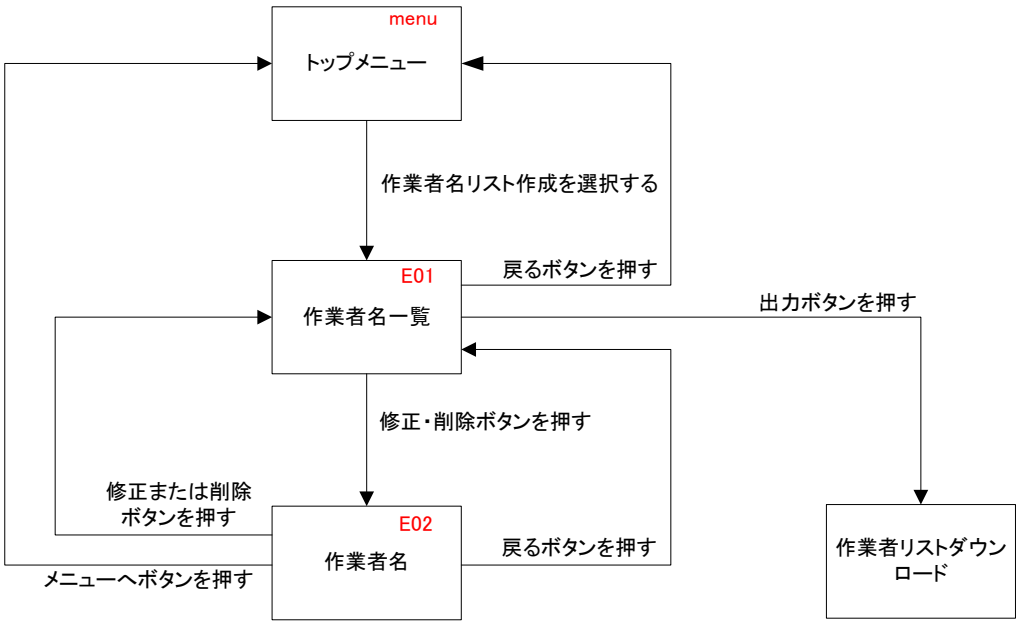


図 3-30　作業者リスト作成遷移図

3.2. 計画支援システムの詳細設計

(1) 適用範囲

本節は、計画支援システムにおける詳細設計についてのみ記載するものとする。

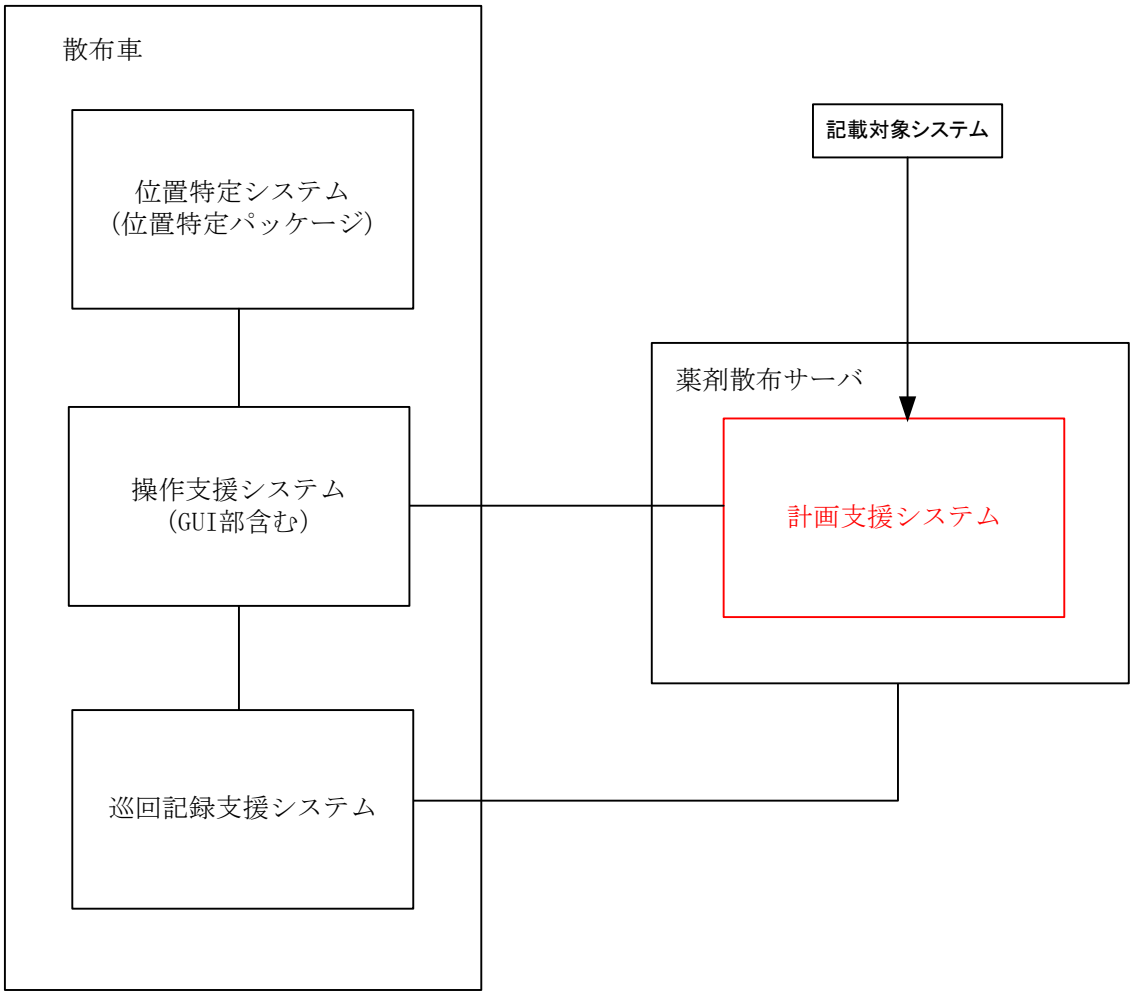


図 3-31　記載範囲

(2) システム構成

ネットワーク構成、システム構成、ソフトウェア構成を以下の図に示す。

なお、本システムは、道路 GIS API を用いた Web システムとなり、また、ネットワークの構成上 HTTP（ポート番号 80）の通信のみ許可されるため、アプレットからは直接データベースにアクセスすることが不可能である。このため、システムデータはサーブレットからアクセス可能な情報に関してはデータベースに格納し、アプレットからアクセスが必要なデータに関しては情報ファイルを用いてアクセスを行う構造とする。

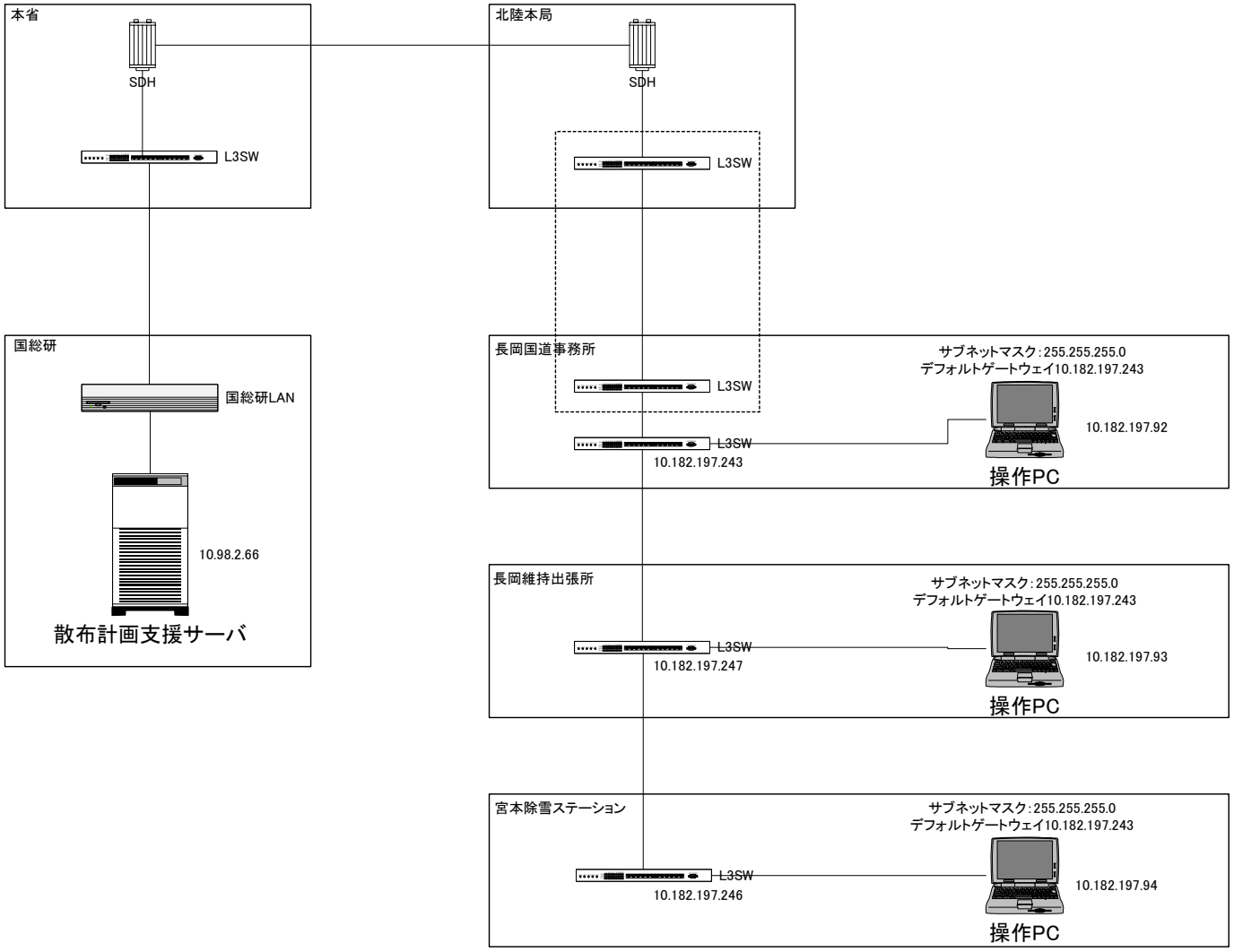


図 3-32　ネットワーク構成

散布支援システム構成図

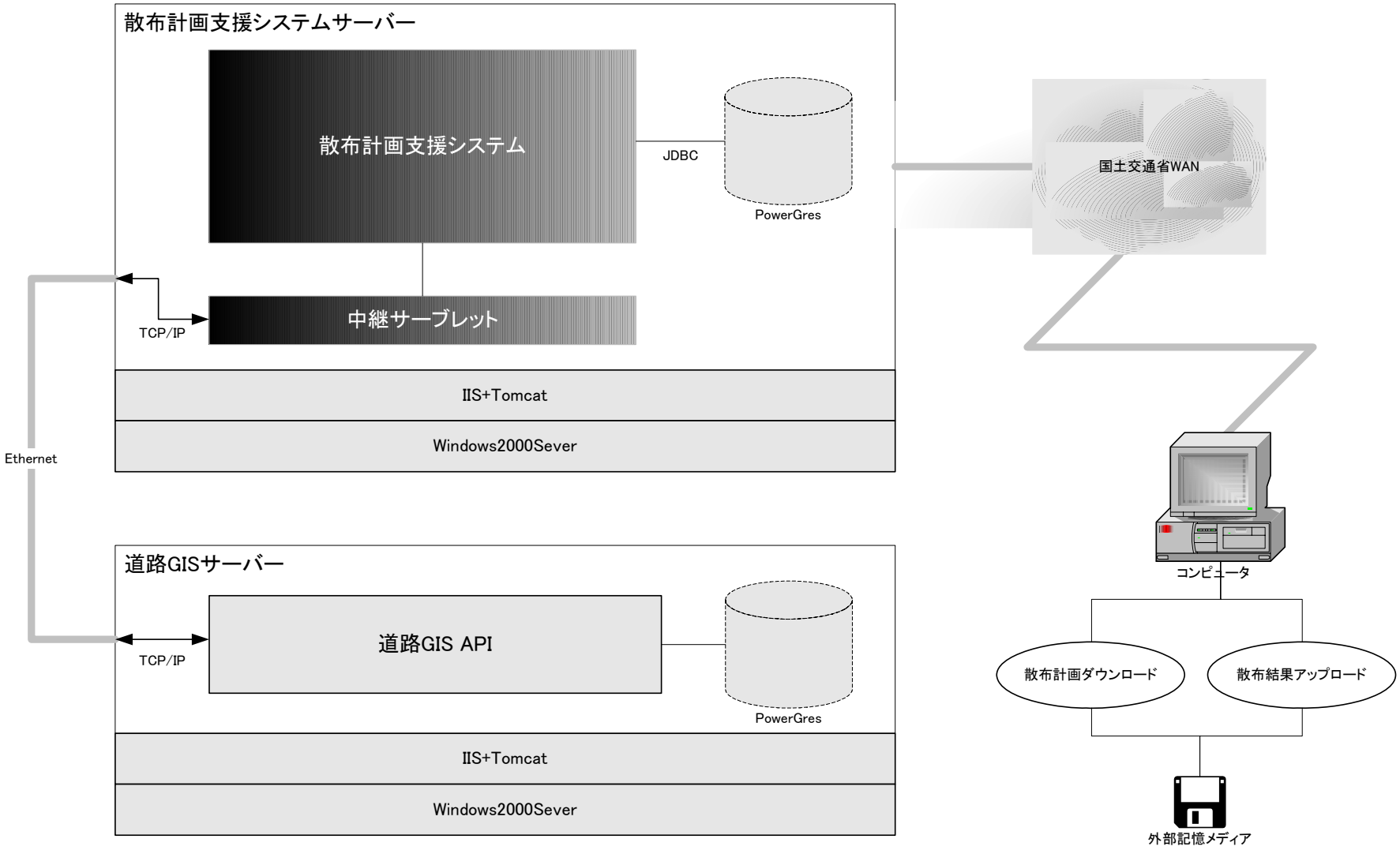


図 3-33　システム構成

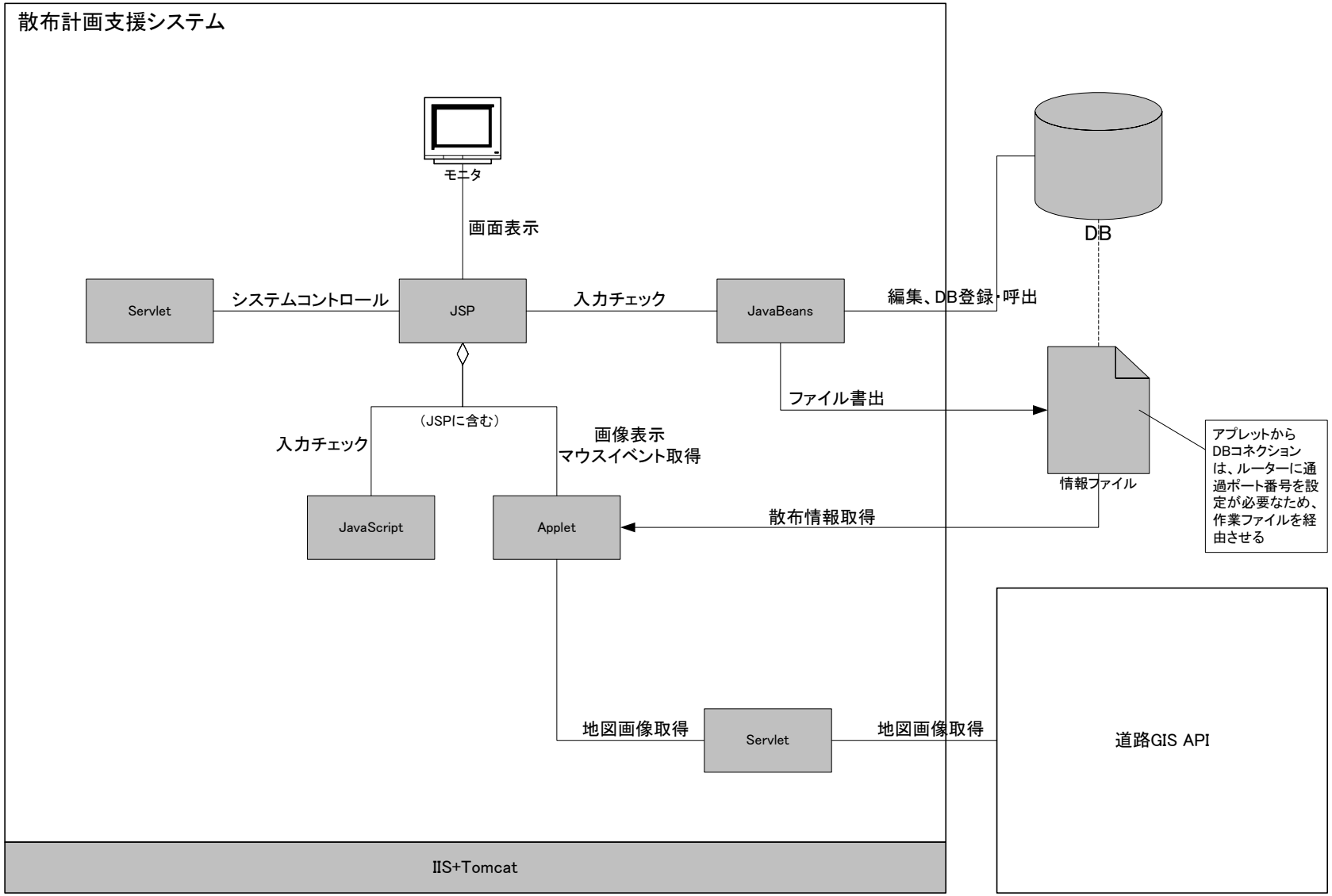


図 3-34　ソフトウェア構成

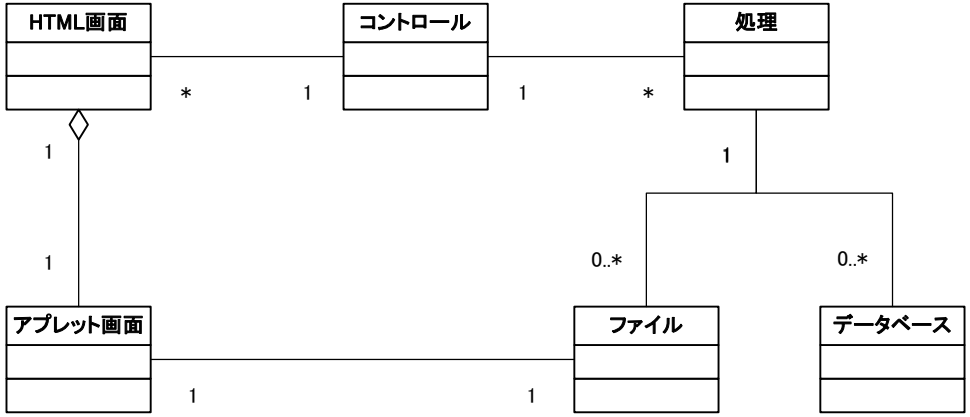


図 3-35　処理方式説明用クラス図

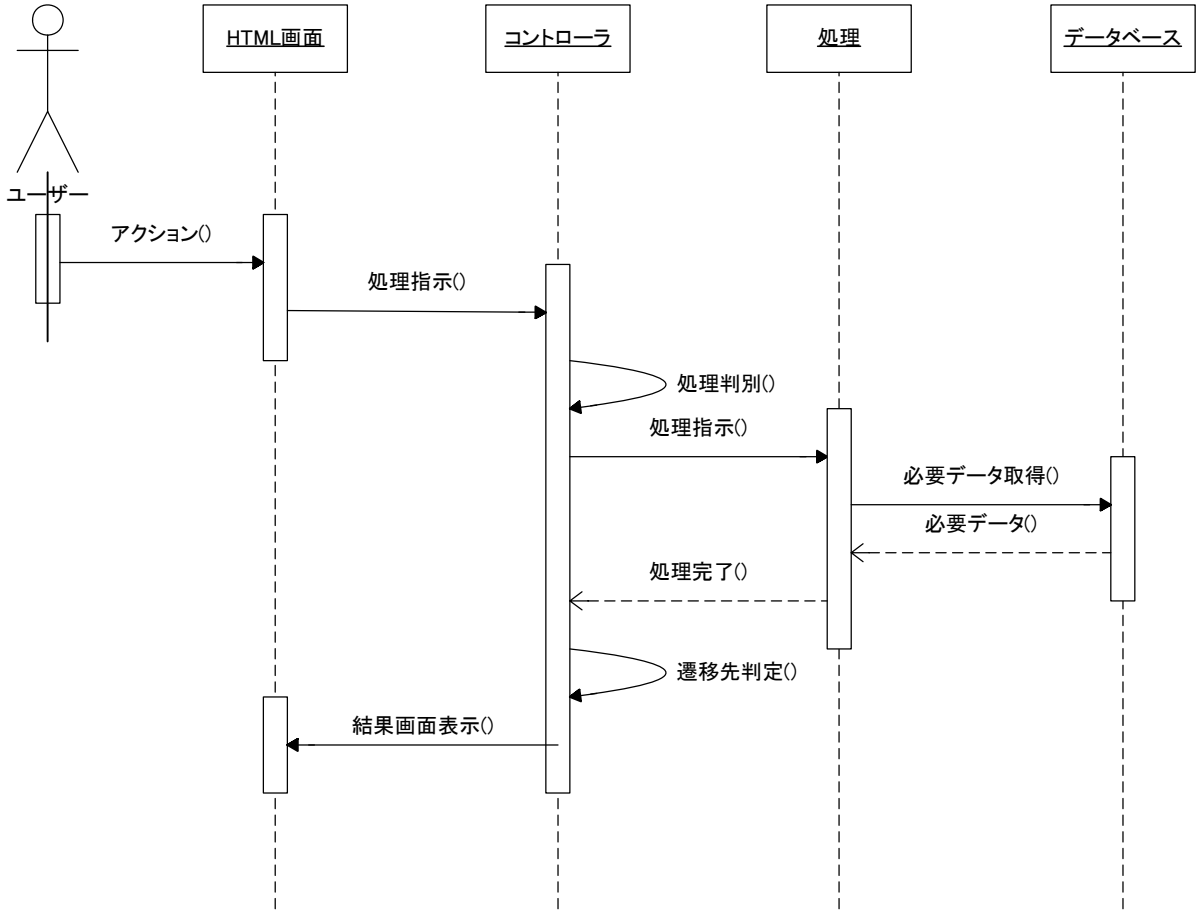


図 3-36　処理方式説明用シーケンス図（アプレットなし参照処理）

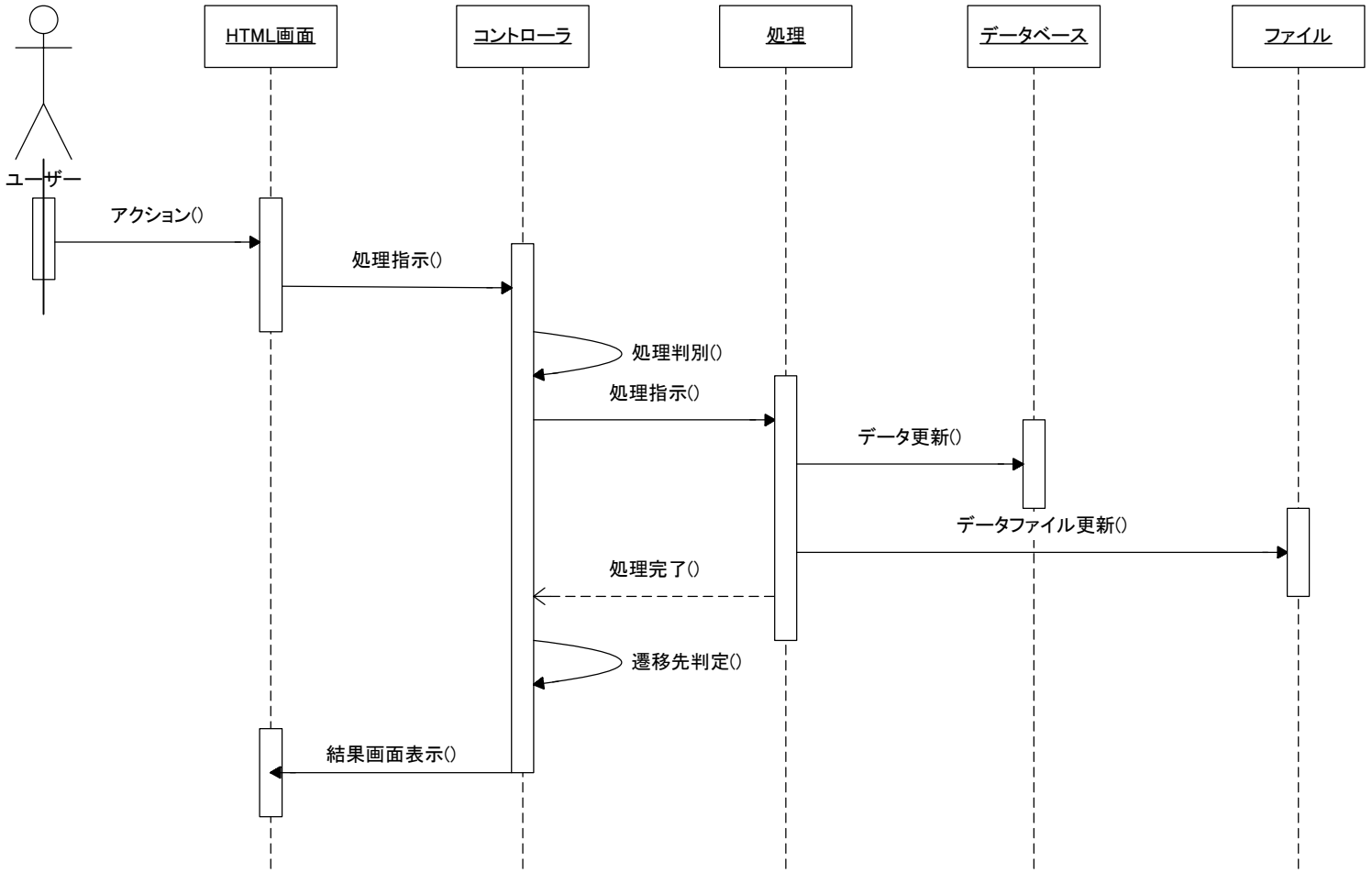


図 3-37　処理方式説明用シーケンス図（アプレットなし更新処理）

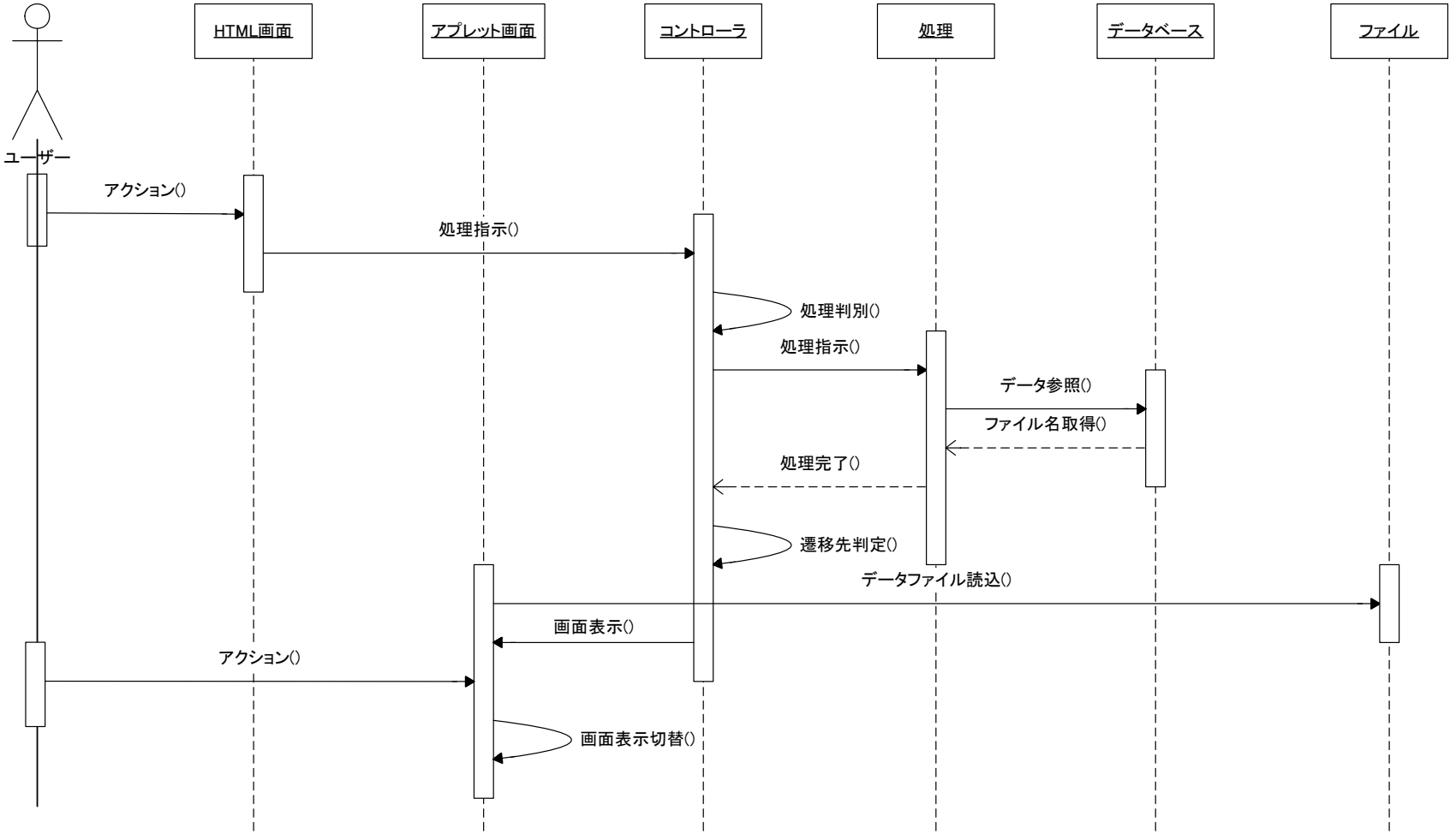


図 3-38　処理方式説明用シーケンス図（アプレットあり参照処理）

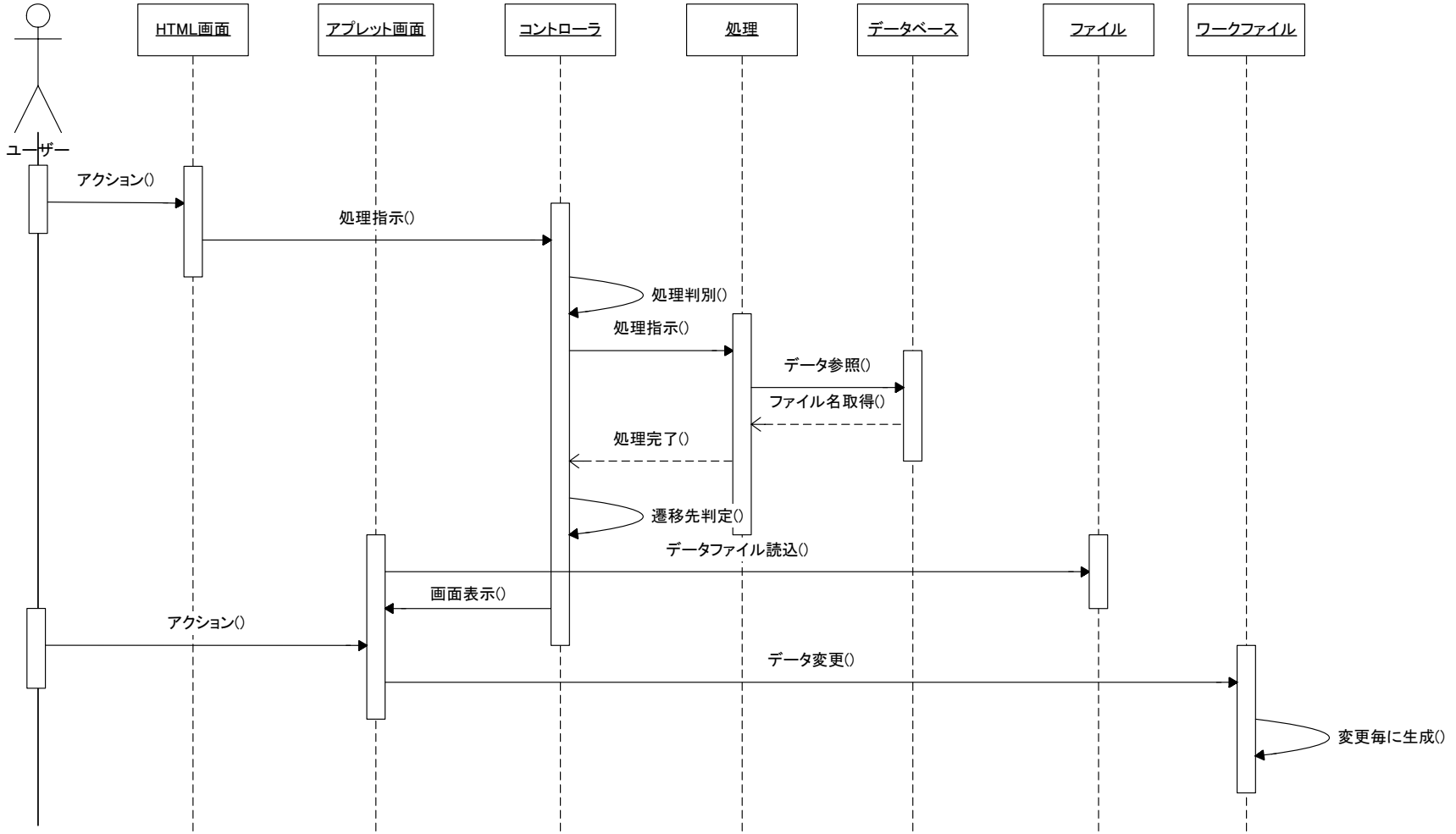


図 3-39　処理方式説明用シーケンス図（アプレットあり更新処理）

(3) コード設計

本システムに使用するコードを以下の表示す。また、画面遷移時に使用する画面 ID を図に示す。

表 3-44　コード表

コード名	車線コード
目　的	車線を識別するためのコード
コード体系	車線毎に-9～9 の整数で指定。1 から順に作成する。
備　考	上り車線（＋）、下り車線（－） 地区によりコードが異なる場合がある

コード	内　　容	備　　考
1	上り追越車線	
2	上り走行車線	
3	上り副道	
4	上り右折車線	
-1	下り追越車線	
-2	下り走行車線	
-3	下り副道	
-4	下り右折車線	

表 3-45　コード表

コード名	散布方向コード
目　的	散布方向の識別および名称を表示するために使用
コード体系	プラスの１桁の整数値
備　考	

コード	内　　容	備　　考
0	左	
1	右	

表 3-46　コード表

コード名	散布薬剤コード
目　的	散布薬剤の識別および名称を表示するために使用
コード体系	プラスの１桁の整数値
備　考	

コード	内　　容	備　　考
1	フレーク	
2	粒	
3	粒大	
4	原塩	

表 3-47　コード表

コード名	所属コード
目　的	所属を識別するためのコード
コード体系	６桁の半角英数
備　考	

コード	内　　容	備　　考
000000	長岡国道宮本工区	

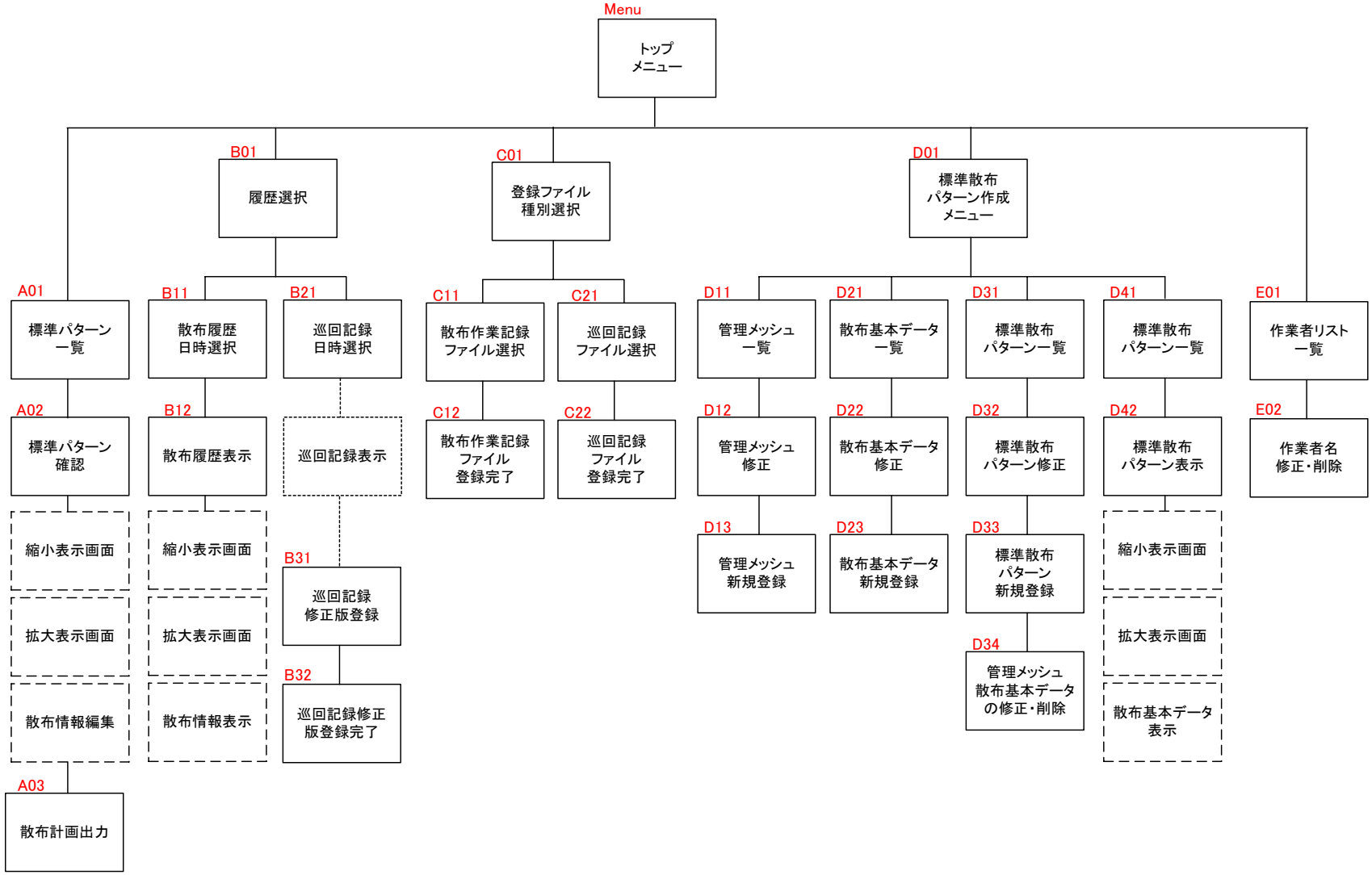

Menu
トップ
メニュー
A01
標準パターン
一覧
A02
標準パターン
確認
縮小表示画面
拡大表示画面
散布情報編集
A03
散布計画出力
B01
履歴選択
B11
散布履歴
日時選択
B12
散布履歴表示
縮小表示画面
拡大表示画面
散布情報表示
B21
巡回記録
日時選択
巡回記録表示
B31
巡回記録
修正版登録
B32
巡回記録修正
版登録完了
C01
登録ファイル
種別選択
C11
散布作業記録
ファイル選択
C12
散布作業記録
ファイル
登録完了
C21
巡回記録
ファイル選択
C22
巡回記録
ファイル
登録完了
D01
標準散布
パターン作成
メニュー
D11
管理メッシュ
一覧
D12
管理メッシュ
修正
D13
管理メッシュ
新規登録
D21
散布基本データ
一覧
D22
散布基本データ
修正
D23
散布基本データ
新規登録
D31
標準散布
パターン一覧
D32
標準散布
パターン修正
D33
標準散布
パターン
新規登録
D34
管理メッシュ
散布基本データ
の修正・削除
D41
標準散布
パターン一覧
D42
標準散布
パターン表示
縮小表示画面
拡大表示画面
散布基本データ
表示
E01
作業者リスト
一覧
E02
作業者名
修正・削除

図 3-40　画面 ID

(4) テーブル設計

本システムに使用するデータベースのテーブルの ER 図、テーブルフォーマットを以下に示す。

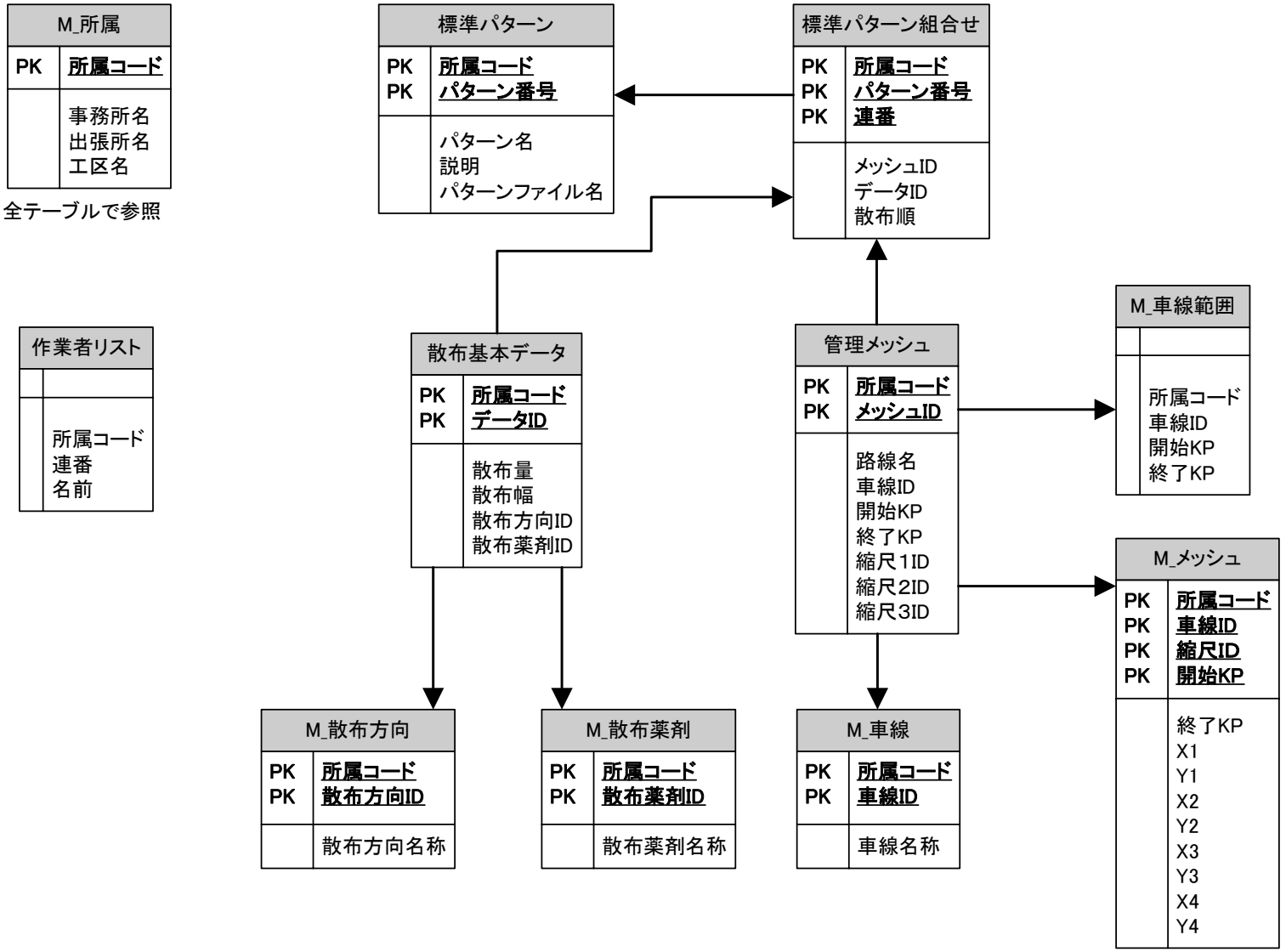

M_所属
PK 所属コード
事務所名
出張所名
工区名
全テーブルで参照
標準パターン
PK 所属コード
PK パターン番号
パターン名
説明
パターンファイル名
標準パターン組合せ
PK 所属コード
PK パターン番号
PK 連番
メッシュID
データID
散布順
作業者リスト
所属コード
連番
名前
散布基本データ
PK 所属コード
PK データID
散布量
散布幅
散布方向ID
散布薬剤ID
管理メッシュ
PK 所属コード
PK メッシュID
路線名
車線ID
開始KP
終了KP
縮尺1ID
縮尺2ID
縮尺3ID
M_車線範囲
所属コード
車線ID
開始KP
終了KP
M_メッシュ
PK 所属コード
PK 車線ID
PK 縮尺ID
PK 開始KP
終了KP
X1
Y1
X2
Y2
X3
Y3
X4
Y4
M_散布方向
PK 所属コード
PK 散布方向ID
散布方向名称
M_散布薬剤
PK 所属コード
PK 散布薬剤ID
散布薬剤名称
M_車線
PK 所属コード
PK 車線ID
車線名称

図 3-41　ER 図（その１）

散布履歴	
PK	**所属コード**
PK	**連番**
	作業日
	開始時刻
	終了時刻
	作業者名
	散布パターン名
	履歴ファイル名

散布履歴詳細	
PK	**所属コード**
PK	**連番**
PK	**詳細連番**
	散布時刻
	自動手動種別
	車線名
	開始KP
	終了KP
	散布量
	散布幅
	散布方向
	薬剤種類

M_帳票位置	
PK	**所属コード**
PK	**種別**
PK	**地点名**
	地点名セル
	KPセル
	区間KPセル
	行き時間セル
	帰り時間セル
	行き気温セル
	帰り気温セル
	行き天候セル
	帰り天候セル
	行き路面状況セル
	帰り路面状況セル
	行き交通状況セル
	帰り交通状況セル
	行き新雪セル
	帰り新雪セル
	行き圧雪セル
	帰り圧雪セル
	行き消雪パイプ地点名セル
	帰り消雪パイプ地点名
	行き消雪パイプ状態セル
	帰り消雪パイプ状態セル

巡回記録	
PK	**所属コード**
PK	**連番**
	巡回日
	出動回数
	出発時間
	帰着時間
	作業者名
	散布パターン名
	帳票ファイル名

巡回記録詳細	
PK	**所属コード**
PK	**連番**
PK	**詳細連番**
	種別
	車線区別
	地点名
	観測時間
	気温
	天候
	路面状況
	新雪深
	圧雪深
	交通状況
	消雪パイプ稼動状況

M_帳票情報	
PK	**所属コード**
	帳票ファイル
	巡回日セル
	回数セル
	出発時間セル
	帰着時間セル
	起点時間

図 3-42　ER 図（その２）

表 3-48　テーブル定義

日本語名	M_所属	物理名	M_SYOZOKU

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	事務所名	JIMUSYU_MEI	VARCHAR	100			
3	出張所名	SYUTTYOUJYO_MEI	VARCHAR	100			
4	工区名	KOUKU_MEI	VARCHAR	100			

表 3-49　テーブル定義

日本語名	M_車線	物理名	M_SYASEN

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	車線ID	SYASEN_ID	NUMERIC		○		
3	車線名称	SYASEN_MEI	VARCHAR	20			

表 3-50　テーブル定義

日本語名	M_散布方向	物理名	M_SANPUHOUKO

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	散布方向ID	SANPUHOUKO_ID	NUMERIC		○		
3	散布方向	SANPUHOUKO	VARCHAR	4			

表 3-51　テーブル定義

日本語名	M_散布薬剤	物理名	M_SANPUYAKUZAI

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	散布薬剤ID	SANPUYAKUZAI_ID	NUMERIC		○		
3	散布薬剤名称	SANPUYAKUZAI_MEI	VARCHAR	10			

表 3-52　テーブル定義

日本語名	標準パターン	物理名	HYOUJYUNPATTERN

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	パターン番号	PTN_NO	NUMERIC		○		
3	パターン名	PATTERN_MEI	VARCHAR	40			
4	説明	SETSUMEI	VARCHAR	300			
5	ﾊﾟﾀｰﾝﾌｧｲﾙ名	PT_FILE	VARCHAR	32			

表 3-53　テーブル定義

日本語名	標準パターン組合せ	物理名	HYOUJYUNPTKUMI

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	パターン番号	PTN_NO	NUMERIC		○	○	FK:標準パターン
3	連番	RENBAN	NUMERIC		○		
4	メッシュID	MESH_ID	VARCHAR	32		○	FK:管理メッシュ
5	データID	DATA_ID	VARCHAR	32		○	FK:散布基本データ
6	散布順	SANPUJUN	NUMERIC				

表 3-54　テーブル定義

日本語名	散布基本データ	物理名	SANPUKIHON

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	データID	DATA_ID	VARCHAR	32	○		
3	散布量	SANPURYO	DECIMAL	4.1			
4	散布幅	SANPUHABA	DECIMAL	3.1			
5	散布方向ID	SANPUHOUKO_ID	NUMERIC			○	FK:M_散布方向
6	散布薬剤ID	SANPUYAKUZAI_ID	NUMERIC			○	FK:M_散布薬剤

表 3-55　テーブル定義

日本語名	管理メッシュ	物理名	KANRIMESH

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	メッシュID	MESH_ID	VARCHAR	32	○		
3	路線名	ROSEN_MEI	VARCHAR	10			
4	車線ID	SYASEN_ID	NUMERIC			○	FK:M_車線
5	開始KP	KAISHI_KP	DECIMAL	6.2			
6	終了KP	SYURYO_KP	DECIMAL	6.2			
7	縮尺１ID	SYKUSYAKU1	NUMERIC				値:1
8	縮尺２ID	SYKUSYAKU2	NUMERIC				値:2
9	縮尺３ID	SYKUSYAKU3	NUMERIC				値:3

表 3-56　テーブル定義

日本語名	メッシュポリゴン	物理名	MESHPOLYGON

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	メッシュID	MESH_ID	VARCHAR	12	○		
3	縮尺ID	SYUKUSYAKU_ID	NUMERIC		○		
4	連番	RENBAN	NUMERIC		○		
5	X	X	DECIMAL	15.10			
6	Y	Y	DECIMAL	15.10			

表 3-57　テーブル定義

日本語名	散布履歴	物理名	SANPURIREKI

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	連番	RENBAN	NUMERIC		○		
3	作業日	SAGYO_DATE	DATE				
4	開始時刻	KAISHI_TIME	TIME				
5	終了時刻	SYURYU_TIME	TIME				
6	作業者名	SAGYOSYA_MEI	VARCHAR	60			
7	散布ﾊﾟﾀｰﾝ名	PATTERN_MEI	VARCHAR	40			
8	履歴ﾌｧｲﾙ名	REREKI_FILE	VARCHAR	32			

表 3-58　テーブル定義

日本語名	M_帳票情報	物理名	M_CHOHYOINFO

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	帳票ファイル	FILE	VARCHAR	32			帳票のﾃﾝﾌﾟﾚｰﾄﾌｧｲﾙ名
3	巡回日セル	JUNKAIBI_CELL	VARCHAR	5			Excelへ出力するセル番号
4	回数セル	KAISU_CELL	VARCHAR	5			Excelへ出力するセル番号
5	出発時間セル	START_CELL	VARCHAR	5			Excelへ出力するセル番号
6	帰着時間セル	END_CELL	VARCHAR	5			Excelへ出力するセル番号
7	起点時間	KITEN_TIME	VARCHAR	5			出動回数を分ける時間

表 3-59　テーブル定義

日本語名	M_帳票位置	物理名	M_CHOHYOCELL

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	種別	SYUBETSU	NUMERIC		○		0:地点 1:消雪
3	地点名	CHITEN_MEI	VARCHAR	30	○		
4	地点名セル	CHITEN_CELL	VARCHAR	5			Excelへ出力するセル番号（以下同じ）
5	KPセル	KP_CELL	VARCHAR	5			種別:0の時、必須
6	区間KPセル	KUKAN_CELL	VARCHAR	5			種別:0の時、必須
7	行き時間セル	TIME1_CELL	VARCHAR	5			種別:0の時、必須
8	帰り時間セル	TIME2_CELL	VARCHAR	5			種別:0の時、必須
9	行き気温セル	KION1_CELL	VARCHAR	5			種別:0の時、必須
10	帰り気温セル	KION2_CELL	VARCHAR	5			種別:0の時、必須
11	行き天候セル	TENKO1_CELL	VARCHAR	5			種別:0の時、必須
12	帰り天候セル	TENKO2_CELL	VARCHAR	5			種別:0の時、必須
13	行き路面状況セル	ROMEN1_CELL	VARCHAR	5			種別:0の時、必須
14	帰り路面状況セル	ROMEN2_CELL	VARCHAR	5			種別:0の時、必須
15	行き交通状況セル	KOUTSU1_CELL	VARCHAR	5			種別:0の時、必須
16	帰り交通状況セル	KOUTSU2_CELL	VARCHAR	5			種別:0の時、必須
17	行き新雪セル	NEW1_CELL	VARCHAR	5			種別:0の時、必須
18	帰り新雪セル	NEW2_CELL	VARCHAR	5			種別:0の時、必須
19	行き圧雪セル	OLD1_CELL	VARCHAR	5			種別:0の時、必須
20	帰り圧雪セル	OLD2_CELL	VARCHAR	5			種別:0の時、必須
21	行き消雪パイプ地点名セル	PIPEMEI1_CELL	VARCHAR	5			種別:1の時、必須
22	帰り消雪パイプ地点名セル	PIPEMEI2_CELL	VARCHAR	5			種別:1の時、必須
23	行き消雪パイプ状態セル	PIPEONOFF1_CELL	VARCHAR	5			種別:1の時、必須
24	帰り消雪パイプ状態セル	PIPEONOFF2_CELL	VARCHAR	5			種別:1の時、必須

表 3-60　テーブル定義

日本語名	巡回記録	物理名	JUNKAIKIROKU

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	連番	RENBAN	NUMERIC		○		
3	巡回日	JUNKAIBI	DATE				
4	出動回数	SYUTSUDOU	NUMERIC				
5	出発時間	START_TIME	TIME				
6	帰着時間	END_TIME	TIME				
7	作業者名	NAME	VARCHAR	20			
8	散布パターン名	SANPU_PATTERN	VARCHAR	40			
9	帳票ファイル名	JUNKAI_FILE	VARCHAR	32			

表 3-61　テーブル定義

日本語名	M_メッシュ	物理名	M_MESH

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	車線ID	SYASEN_ID	NUMERIC		○		
3	縮尺ID	SYUKUSYAKU_ID	NUMERIC		○		
4	開始KP	KAISHI_KP	DECIMAL	6.2			
5	終了KP	SYURYO_KP	DECIMAL	6.2			
6	X1	X1	DECIMAL	15.10			
7	Y1	Y1	DECIMAL	15.10			
8	X2	X2	DECIMAL	15.10			
9	Y2	Y2	DECIMAL	15.10			
10	X3	X3	DECIMAL	15.10			
11	Y3	Y3	DECIMAL	15.10			
12	X4	X4	DECIMAL	15.10			
13	Y4	Y4	DECIMAL	15.10			

表 3-62　テーブル定義

日本語名	M_車線ID範囲マスタ	物理名	M_MESHRANGE

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	車線ID	SYASEN_ID	NUMERIC		○		
3	開始KP	KAISHI_KP	DECIMAL	6.2	○		
4	終了KP	SYURYO_KP	DECIMAL	6.2			

表 3-63　テーブル定義

日本語名	作業者リスト	物理名	WORKMANLIST

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	連番	RENBAN	NUMERIC		○		
3	作業者名	NAME	VARCHAR	20			

表 3-64　テーブル定義

日本語名	散布履歴詳細	物理名	SANPUSYOSAI

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	連番	RENBAN	NUMERIC		○	○	FK:散布履歴
3	詳細連番	SYOREN	NUMERIC		○		
4	散布時刻	SANPU_TIME	TIME				
5	自動手動種別	SYUBETSU	VARCHAR	4			
6	車線名	SYASEN_MEI	VARCHAR	20			
7	開始KP	KAISHI_KP	DECIMAL	6.2			
8	終了KP	SYURYO_KP	DECIMAL	6.2			
9	散布量	SANPU_RYO	DECIMAL	4.1			
10	散布幅	SANPU_HABA	DECIMAL	3.1			
11	散布方向	SANPU_HOUKO	VARCHAR	2			
12	薬剤種類	YAKUZAI_SYURUI	VARCHAR	10			

表 3-65　テーブル定義

日本語名	巡回記録詳細	物理名	JUNKAISYOSAI

No	日本語名	カラム名	型	サイズ	PK	FK	備　考
1	所属コード	SYOZOKU_CD	VARCHAR	6	○		
2	連番	RENBAN	NUMERIC		○	○	FK:巡回記録
3	詳細連番	SYOREN	NUMERIC		○		
4	種別	SYUBETSU	VARCHAR	4			「地点」または「消雪」
5	車線区別	SYASEN	VARCHAR	4			「上り」または「下り」
6	地点名	CHITEN_MEI	VARCHAR	30			
7	観測時間	KANSOKU_TIME	TIME				
8	気温	KION	DECIMAL	5.1			
9	天候	TENKO	VARCHAR	4			
10	路面状況	ROMEN	VARCHAR	12			
11	新雪深	NEW_FUKASA	NUMERIC				
12	圧雪深	OLD_FUKASA	NUMERIC				
13	交通状況	KOUTSU	VARCHAR	4			
14	消雪パイプ稼動状況	PIPE	VARCHAR	4			

(5) ファイル設計

本システムで使用するファイルは、以下のとおりである。データベースとファイルの関連及び各ファイルフォーマットを以下に示す。

表 3-66　ファイル一覧

ファイル名	形式	I/O	説　明
M_メッシュ	CSV	I	管理メッシュの入力チェックおよび表示用メッシュとの対応をとるためのファイル（DB:M_メッシュテーブル登録用のファイルのため直接は使用しない）。散布制御システムの散布メッシュファイルと同じ。
M_表示用メッシュ	CSV	I	管理メッシュの表示メッシュの座標を持ったファイル。散布制御システムの散布メッシュファイルと1対1の対応（DB:M_メッシュテーブル登録用のファイルのため直接は使用しない）。
散布計画	CSV	I/O	散布制御システムで使用する散布計画ファイル。散布計画作成画面から出力ボタンでダウンロードされるファイル。
散布履歴	CSV	I	散布制御システムで散布作業時に自動生成されるファイル。散布作業記録の登録時に指定するファイル。
巡回記録データ	CSV	I	巡回記録システムで自動生成されるファイル。巡回記録の登録時に指定するファイル。
標準パターン表示用	XML	I/O	アプレットで標準パターン確認時に表示するデータを格納したファイル。標準パターン作成時に生成し、標準パターン分生成される。
散布履歴表示用	XML	I/O	アプレットで散布履歴表示時に表示するデータを格納したファイル。散布履歴ファイルアップロード時に生成され、アップロード毎に生成される。
巡回記録ファイル	Excel	O	巡回記録データを格納したExcelファイル。巡回記録ファイルアップロード時に生成され、アップロード毎に生成される。
変更用作業ファイル	CSV	I/O	標準パターン確認アプレットで、散布計画データが変更された場合に、変更情報を一時的に格納しておく作業ファイル。
作業者リスト	TEXT	O	作業者名リスト作成で作成した作業者を TEXT ファイルへ保存するファイル。

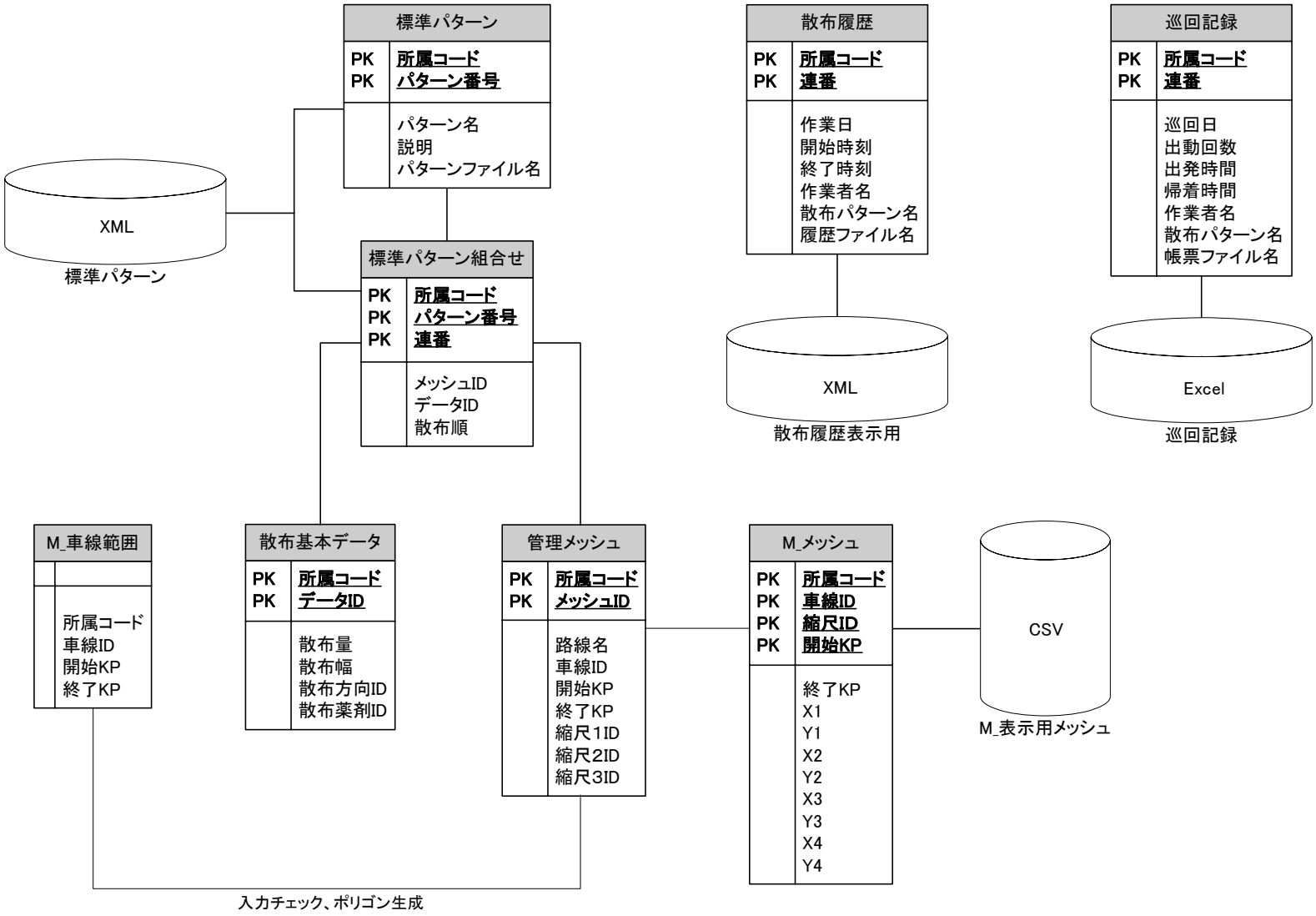


図 3-43　データベースとファイル関連図

表 3-67　ファイルフォーマット

ファイル名	M_メッシュ	形式	CSV
説明	管理メッシュの入力チェックおよび表示用メッシュとの対応をとるためのファイル。散布制御システムの散布メッシュファイルと同じ。		

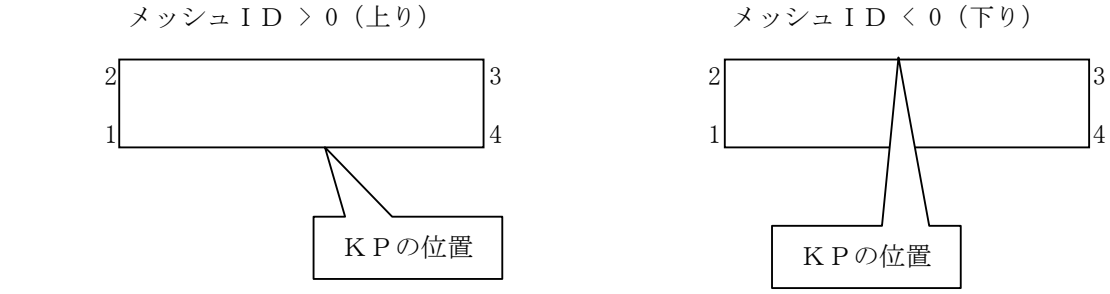
フォーマット

1行1メッシュとし、以下の項目をカンマで区切った CSV 形式とする。メッシュ数分の行数が必要。

- メッシュ番号
- 車線 ID
- メッシュ KP
- メッシュの X 座標１
- メッシュの Y 座標１
- メッシュの X 座標２
- メッシュの Y 座標２
- メッシュの X 座標３
- メッシュの Y 座標３
- メッシュの X 座標４
- メッシュの Y 座標４

（注意事項）

メッシュに対するＫＰの位置



上りメッシュと下りメッシュでＫＰの位置が異なるので注意
メッシュＩＤとメッシュ点の並び順は、上り、下りとも同じ

※DB:M_メッシュテーブル登録用のファイルのため直接は使用しない。

※空行または１カラム目が；の場合コメントとみなす。

表 3-68　ファイルフォーマット

ファイル名	M_表示用メッシュ	形式	CSV
説明	管理メッシュの表示メッシュの座標を持ったファイル。散布制御システムの散布メッシュファイルと１対１の対応。		

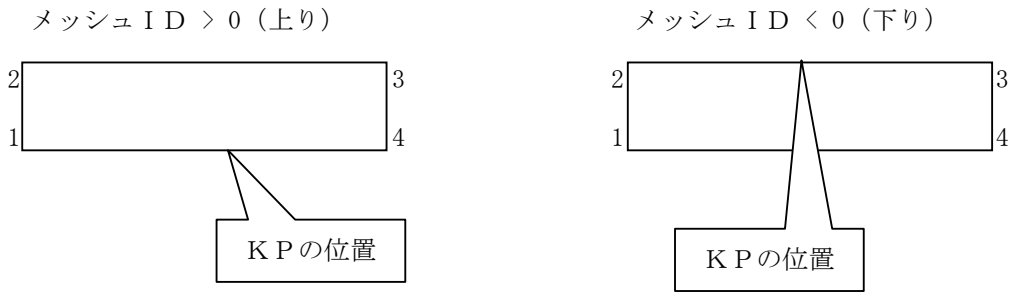
フォーマット

１行１メッシュとし、以下の項目をカンマで区切った CSV 形式とする。メッシュ数分の行数が必要。必ず、M_メッシュと１対１で対応させる必要あり。縮尺数分のファイルが必要。

- メッシュ番号
- 車線 ID
- メッシュ KP
- メッシュの X 座標１（画面表示用）
- メッシュの Y 座標１（画面表示用）
- メッシュの X 座標２（画面表示用）
- メッシュの Y 座標２（画面表示用）
- メッシュの X 座標３（画面表示用）
- メッシュの Y 座標３（画面表示用）
- メッシュの X 座標４（画面表示用）
- メッシュの Y 座標４（画面表示用）

（注意事項）

メッシュに対するＫＰの位置



上りメッシュと下りメッシュでＫＰの位置が異なるので注意
メッシュＩＤとメッシュ点の並び順は、上り、下りとも同じ

※空行または１カラム目が；の場合コメントとみなす。

表 3-69　ファイルフォーマット

ファイル名	散布計画	形式	CSV
説明	散布制御システムで使用する散布計画ファイル。散布計画作成画面から出力ボタンでダウンロードされるファイル。		

フォーマット

1行目：散布標準パターン名

2行目以降：以下の項目をカンマで区切った CSV 形式とする。散布メッシュ数分の行数が必要。

- 車線 ID
- 開始 KP
- 終了 KP
- 散布量
- 散布幅
- 散布方向（0:左 1:右）
- 湿式割合（現散布車では 0 で固定）
- 薬剤種類（1:フレーク 2:粒 3:粒大 4:原塩）
- 散布順

（注意事項）

車線 ID が（＋）の場合は、必ず　開始 KP＞終了 KP

車線 ID が（―）の場合は、必ず　開始 KP＜終了 KP　とする。

２車線同時散布の場合は、散布順を同じにする。

※空行または１カラム目が；の場合コメントとみなす。

表 3-70　ファイルフォーマット

ファイル名	散布履歴	形式	CSV
説明	散布制御システムで散布作業時に自動生成されるファイル。散布作業記録の登録時に指定するファイル。		

フォーマット

1行目：作業者名

2行目：標準散布パターン名

3行目：開始日時

4行目：終了日時

5行目以降：以下の項目をカンマで区切った CSV 形式とする。散布変更数分の行数が必要。

- 日時
- 自動・手動種別（0:自動 1:手動）
- 車線 ID
- 開始 KP
- 終了 KP
- 散布量
- 散布幅
- 散布方向（0:左 1:右）
- 湿式割合（現散布車では 0 で固定）
- 薬剤種類（1:フレーク 2:粒 3:粒大 4:原塩）

表 3-71　ファイルフォーマット

ファイル名	巡回記録データ	形式	CSV
説明	巡回記録システムで自動生成されるファイル。巡回記録の登録時に指定するファイル。		

フォーマット

1行目：作業者名

2行目：標準散布パターン名

3行目：開始日時

4行目：終了日時

5行目以降：以下の項目をカンマで区切った CSV 形式とする。地点数分の行数が必要。

- “地点”で固定
- 巡回地点 KP
- “上り”または“下り“
- 地点名称
- 通過時間
- 気温
- 天候（晴、曇り、雨、雪）
- 路面状況（乾燥、凍結、ザクレ、シャーベット、濡れ）
- 交通状況（良好、渋滞）
- 新雪深（なし、１ｃｍ、２ｃｍ、３ｃｍ、４ｃｍ、５ｃｍ）
- 圧雪深（なし、１ｃｍ、２ｃｍ、３ｃｍ、４ｃｍ、５ｃｍ）
- 表示用地点 KP
- 表示用開始 KP
- 表示用終了 KP

５＋地点数行目以降：以下の項目をカンマで区切った CSV 形式とする。消雪パイプ箇所数分の行数が必要。

- “消雪”で固定
- 消雪パイプ地点 KP
- “上り”または“下り“
- 地点名称
- 通過時間
- 稼動状況（停止、稼動）

表 3-72　ファイルフォーマット

ファイル名	標準パターン表示用	形式	XML
説明	アプレットで標準パターン確認時に表示するデータを格納したファイル。標準パターン作成時に生成し、標準パターン分生成される。		

フォーマット

```
<?xml version="1.0" encoding="Shift-JIS"?>
<!-- 標準パターン -->
<!-- HYOUJYUN_PATTERN.dtd -->
<!DOCTYPE HYOUJYUN_PATTERN [
        <!ELEMENT SANPU_MESH    (SYOZOKU_CD, SYASEN_ID, KAISHI_KP, SYURYO_KP,
                                SYASEN_MEI, KASYO_MEI, SANPU_RYO,
                                SANPU_HABA, SANPU_HOUKOU, SANPU_YAKUZAI,
                                SANPU_JYUN,
                                POLYGON1, POLYGON2, POLYGON3)+>
        <!ELEMENT SYOZOKU_CD    (#PCDATA)>    <!-- 所属コード -->
        <!ELEMENT SYASEN_ID     (#PCDATA)>    <!-- 車線 ID -->
        <!ELEMENT KAISHI_KP     (#PCDATA)>    <!-- 開始 KP-->
        <!ELEMENT SYURYO_KP     (#PCDATA)>    <!-- 終了 KP -->
        <!ELEMENT SYASEN_MEI    (#PCDATA)>    <!-- 車線名      -->
        <!ELEMENT KASYO_MEI     (#PCDATA)>    <!-- 散布箇所名 -->
        <!ELEMENT SANPU_RYO     (#PCDATA)>    <!-- 散布量      -->
        <!ELEMENT SANPU_HABA    (#PCDATA)>    <!-- 散布幅      -->
        <!ELEMENT SANPU_HOUKOU  (#PCDATA)>    <!-- 散布方向    -->
        <!ELEMENT SANPU_YAKUZAI (#PCDATA)>    <!-- 散布薬剤    -->
        <!ELEMENT SANPU_JYUN    (#PCDATA)>    <!-- 散布順      -->
        <!ELEMENT POLYGON1      (#PCDATA)>    <!-- 縮尺１ポリゴン座標 -->
        <!ELEMENT POLYGON2      (#PCDATA)>    <!-- 縮尺２ポリゴン座標 -->
        <!ELEMENT POLYGON3      (#PCDATA)>    <!-- 縮尺３ポリゴン座標 -->
]>
```

車線名は、「上り走行車線」「下り追越車線」 -> M_車線 DB から取得

散布箇所名は、「R8 KP72.5-KP69.8」 -> 路線名と開始 KP、終了 KP から編集

散布方向は、「右」または「左」

縮尺ポリゴンは、x1,y1,x2,y2,x3,y3,・・・・・,のようにカンマで区切って節点数分記述

表 3-73　ファイルフォーマット

ファイル名	散布履歴表示用	形式	XML
説明	アプレットで散布履歴表示時に表示するデータを格納したファイル。散布履歴ファイルアップロード時に生成され、アップロード毎に生成される。		

フォーマット

```
<?xml version="1.0" encoding="Shift-JIS"?>
<!-- 散布履歴 -->
<!-- SANPU_RIREKI.dtd -->
<!DOCTYPE SANPU_RIREKI [
        <!ELEMENT SANPU_MESH    (SYOZOKU_CD, SYASEN_MEI, KASYO_MEI, SANPU_ JIKOKU,
                                SANPU_RYO, SANPU_HABA, SANPU_HOUKOU, SANPU_YAKUZAI,
                                JISYU_SYUBETSU, POLYGON1, POLYGON2, POLYGON3)+>
        <!ELEMENT SYOZOKU_CD     (#PCDATA)>    <!-- 所属コード      -->
        <!ELEMENT SYASEN_MEI     (#PCDATA)>    <!-- 車線名          -->
        <!ELEMENT KASYO_MEI      (#PCDATA)>    <!-- 散布箇所名      -->
        <!ELEMENT SANPU_JIKOKU   (#PCDATA)>    <!-- 散布時刻    -->
        <!ELEMENT SANPU_RYO      (#PCDATA)>    <!-- 散布量          -->
        <!ELEMENT SANPU_HABA     (#PCDATA)>    <!-- 散布幅          -->
        <!ELEMENT SANPU_HOUKOU   (#PCDATA)>    <!-- 散布方向        -->
        <!ELEMENT SANPU_YAKUZAI  (#PCDATA)>    <!-- 散布薬剤        -->
        <!ELEMENT JISYU_SYUBETSU (#PCDATA)>    <!-- 自動・手動種別 -->
        <!ELEMENT POLYGON1       (#PCDATA)>    <!-- 縮尺１ポリゴン座標 -->
        <!ELEMENT POLYGON2       (#PCDATA)>    <!-- 縮尺２ポリゴン座標 -->
        <!ELEMENT POLYGON3       (#PCDATA)>    <!-- 縮尺３ポリゴン座標 -->
]>
```

車線名は、「上り走行車線」「下り追越車線」 -> M_車線 DB から取得

散布箇所名は、「KP72.5-KP69.8」 -> 開始 KP、終了 KP から編集

散布時刻は、「2003.12.05 21:38」 -> 散布開始時刻を編集

散布方向は、「右」または「左」

散布薬剤は、「粒」「フレーク」 -> M_散布薬剤 DB から取得

自動・手動種別は、「自動」または「手動」

縮尺ポリゴンは、x1,y1,x2,y2,x3,y3,・・・・・,のようにカンマで区切って節点数分記述

表 3-74　ファイルフォーマット

ファイル名	巡回記録表示用	形式	Excel
説明	巡回記録を登録した Excel ファイル。		
フォーマット			

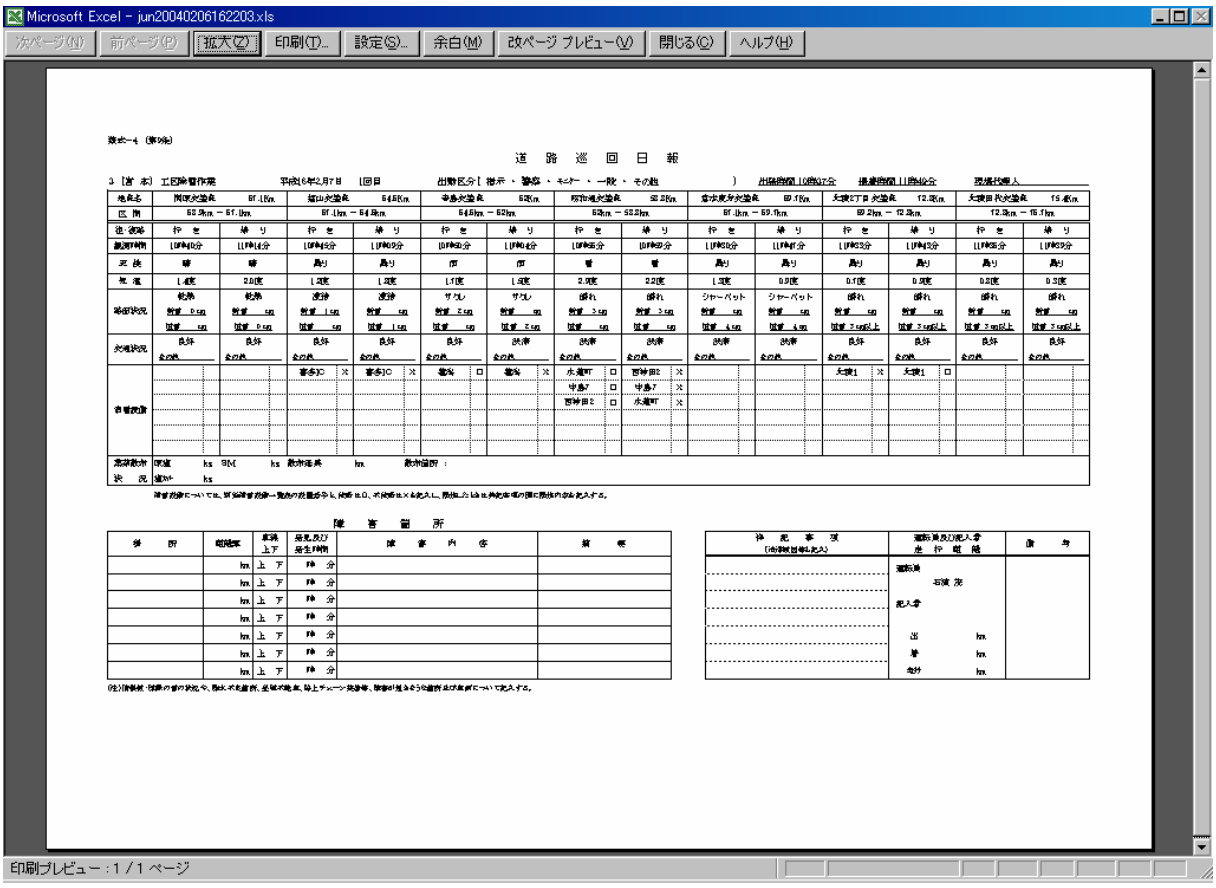


表 3-75　ファイルフォーマット

ファイル名	変更用作業ファイル	形式	CSV
説明	標準パターン確認アプレットで、散布メッシュがクリックされ散布情報ポップアップメニューから散布情報が変更された時に、散布情報を一時的に格納しておくための作業ファイル。標準パターン確認画面から「出力」ボタンが押された時、標準散布パターン XML と結合して標準散布パターン CSV を生成する。		

フォーマット

以下の項目をカンマで区切った CSV 形式とする。散布情報データが更新される毎に追加されるため、OK ボタンが押された分の行数が生成される。

- 車線 ID
- 開始 KP
- 終了 KP
- 散布量
- 散布幅
- 散布方向（0:左 1:右）
- 薬剤種類（1:フレーク 2:粒 3:粒大 4:原塩）

(6) 処理フロー

本システムにおける処理フローを以下に示す。また、遷移一覧を下表に示す。

表 3-76　画面遷移一覧（その１）

遷移元画面	アクション	遷移先画面	ID
トップメニュー	リンク：散布計画作成	標準パターン一覧	A01
標準パターン一覧	ボタン：戻る	トップメニュー	Menu
標準パターン一覧	リンク：標準パターン名	標準パターン確認	A02
標準パターン確認	ボタン：出力	散布計画出力	A03
標準パターン確認	ボタン：戻る	標準パターン一覧	A01
散布計画出力	ボタン：戻る	標準パターン確認	A02
散布計画出力	ボタン：一覧へ戻る	標準パターン一覧	A01
散布計画出力	ボタン：メニューへ	トップメニュー	Menu
トップメニュー	リンク：散布履歴、巡回記録表示	履歴選択	B01
履歴選択	ボタン：戻る	トップメニュー	Menu
履歴選択	リンク：最新作業履歴表示	散布履歴表示	B12N
散布履歴表示	ボタン：戻る（日時経由なし）	履歴選択	B01
履歴選択	リンク：過去作業履歴選択	散布履歴日時選択	B11
散布履歴日時選択	ボタン：戻る	履歴選択	B01
散布履歴日時選択	リンク：作業日	散布履歴表示	B12H
散布履歴表示	ボタン：戻る（日時経由あり）	散布履歴日時選択	B11
散布履歴表示	ボタン：メニューへ	トップメニュー	Menu
履歴選択	リンク：最新巡回記録表示	巡回記録（別ウィンドウ）	B22
履歴選択	リンク：過去巡回記録選択	巡回記録日時選択	B21
巡回記録日時選択	ボタン：戻る	履歴選択	B01
巡回記録日時選択	リンク：巡回日	巡回記録（別ウィンドウ）	
履歴選択	リンク：巡回記録修正版の登録	巡回記録ファイル修正版選択	B31
巡回記録ファイル修正版選択	ボタン：戻る	履歴選択	B01
巡回記録ファイル修正版選択	ボタン：送信	巡回記録ファイル修正版登録完了	B32
巡回記録ファイル修正版登録完了	ボタン：戻る	履歴選択	B01
巡回記録ファイル修正版登録完了	ボタン：メニューへ	トップメニュー	Menu
トップメニュー	リンク：散布結果、巡回記録の登録	登録ファイル種別選択	C01
登録ファイル種別選択	ボタン：戻る	トップメニュー	Menu

（次ページに続く）

表 3-77　画面遷移一覧（その２）

遷移元画面	アクション	遷移先画面	ID
登録ファイル種別選択	リンク：散布作業記録の登録	散布作業記録ファイル選択	C11
散布作業記録ファイル選択	ボタン：送信	散布作業記録ファイル登録完了	C12
散布作業記録ファイル選択	ボタン：戻る	登録ファイル種別選択	C01
散布作業記録ファイル登録完了	ボタン：メニューへ	トップメニュー	Menu
登録ファイル種別選択	リンク：巡回記録の登録	巡回記録ファイル選択	C21
巡回記録ファイル選択	ボタン：送信	巡回記録ファイル登録完了	C22
巡回記録ファイル選択	ボタン：戻る	登録ファイル種別選択	C01
巡回記録ファイル登録完了	ボタン：メニューへ	トップメニュー	Menu
トップメニュー	リンク：標準散布パターン作成	標準散布パターン作成メニュー	D01
標準散布パターン作成メニュー	ボタン：戻る	トップメニュー	Menu
標準散布パターン作成メニュー	リンク：管理メッシュ作成	管理メッシュ一覧	D11
管理メッシュ一覧	リンク：メッシュ ID	管理メッシュ修正	D12
管理メッシュ一覧	ボタン：新規	管理メッシュ新規登録	D13
管理メッシュ一覧	ボタン：戻る	標準散布パターン作成メニュー	D01
管理メッシュ一覧	ボタン：メニューへ	トップメニュー	Menu
管理メッシュ修正	ボタン：修正	管理メッシュ一覧	D11U
管理メッシュ修正	ボタン：複製	管理メッシュ登録	D13C
管理メッシュ修正	ボタン：削除	管理メッシュ一覧	D11D
管理メッシュ修正	ボタン：戻る	管理メッシュ一覧	D11
管理メッシュ登録	ボタン：登録	管理メッシュ一覧	D11I
管理メッシュ登録	ボタン：戻る	管理メッシュ一覧	D11
標準散布パターン作成メニュー	リンク：散布基本データ作成	散布基本データ一覧	D21
散布基本データ一覧	リンク：データ ID	散布基本データ修正	D22
散布基本データ一覧	ボタン：新規	散布基本データ新規登録	D23
散布基本データ一覧	ボタン：戻る	標準散布パターン作成メニュー	D01
散布基本データ一覧	ボタン：メニューへ	トップメニュー	Menu
散布基本データ修正	ボタン：修正	散布基本データ一覧	D21U
散布基本データ修正	ボタン：複製	散布基本データ登録	D23C
散布基本データ修正	ボタン：削除	散布基本データ一覧	D21D
散布基本データ修正	ボタン：戻る	散布基本データ一覧	D21
散布基本データ登録	ボタン：登録	散布基本データ一覧	D21I

（次ページに続く）

表 3-78　画面遷移一覧（その３）

遷移元画面	アクション	遷移先画面	ID
散布基本データ登録	ボタン：戻る	散布基本データ一覧	D21
標準散布パターン作成メニュー	リンク：標準散布パターン作成	標準散布パターン一覧	D31
標準散布パターン一覧	リンク：標準散布パターン名	標準散布パターン修正	D32
標準散布パターン一覧	ボタン：新規	標準散布パターン新規登録	D33
標準散布パターン一覧	ボタン：戻る	標準散布パターン作成メニュー	D01
標準散布パターン一覧	ボタン：メニューへ	トップメニュー	Menu
標準散布パターン修正	ボタン：修正	標準散布パターン一覧	D31U
標準散布パターン修正	ボタン：複製	標準散布パターン登録	D33C
標準散布パターン修正	ボタン：削除	標準散布パターン一覧	D31D
標準散布パターン修正	ボタン：戻る	標準散布パターン一覧	D31
標準散布パターン修正	ボタン：修正・削除	管理メッシュ・散布基本データの修正・削除	D34
標準散布パターン修正	ボタン：追加	標準散布パターン修正	D32A
標準散布パターン登録	ボタン：登録	標準散布パターン一覧	D31I
標準散布パターン登録	ボタン：戻る	標準散布パターン一覧	D31
標準散布パターン登録	ボタン：修正・削除	管理メッシュ・散布基本データの修正・削除	D34
標準散布パターン登録	ボタン：追加	標準散布パターン登録	D33A
管理メッシュ・散布基本データの修正・削除	ボタン：修正	標準散布パターン修正	D32U
管理メッシュ・散布基本データの修正・削除	ボタン：修正	標準散布パターン登録	D33U
管理メッシュ・散布基本データの修正・削除	ボタン：削除	標準散布パターン修正	D32D
管理メッシュ・散布基本データの修正・削除	ボタン：削除	標準散布パターン登録	D33D
管理メッシュ・散布基本データの修正・削除	ボタン：戻る	標準散布パターン修正	D32R
管理メッシュ・散布基本データの修正・削除	ボタン：戻る	標準散布パターン登録	D33R

（次ページに続く）

表 3-79　画面遷移一覧（その４）

遷移元画面	アクション	遷移先画面	ID
標準散布パターン作成メニュー	リンク：標準散布パターン確認	標準散布パターン一覧	D41
標準散布パターン一覧	ボタン：戻る	標準散布パターン作成メニュー	D01
標準散布パターン一覧	リンク：標準散布パターン名	標準散布パターン確認	D42
標準散布パターン確認	ボタン：戻る	標準散布パターン一覧	D41
標準散布パターン一覧	リンク：標準散布パターン名	標準散布パターン確認	D42
標準散布パターン確認	ボタン：戻る	標準散布パターン一覧	D41
標準散布パターン確認	ボタン：メニューへ	トップメニュー	Menu
標準ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	マウスクリック	標準ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	
散布履歴表示ｱﾌﾟﾚｯﾄ	マウスクリック	散布履歴表示ｱﾌﾟﾚｯﾄ	
標準散布ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	マウスクリック	標準散布ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	

表 3-80　処理フロー

遷移元	全画面共通	アクション	画面遷移時における全てのアクション	遷移先	全画面共通

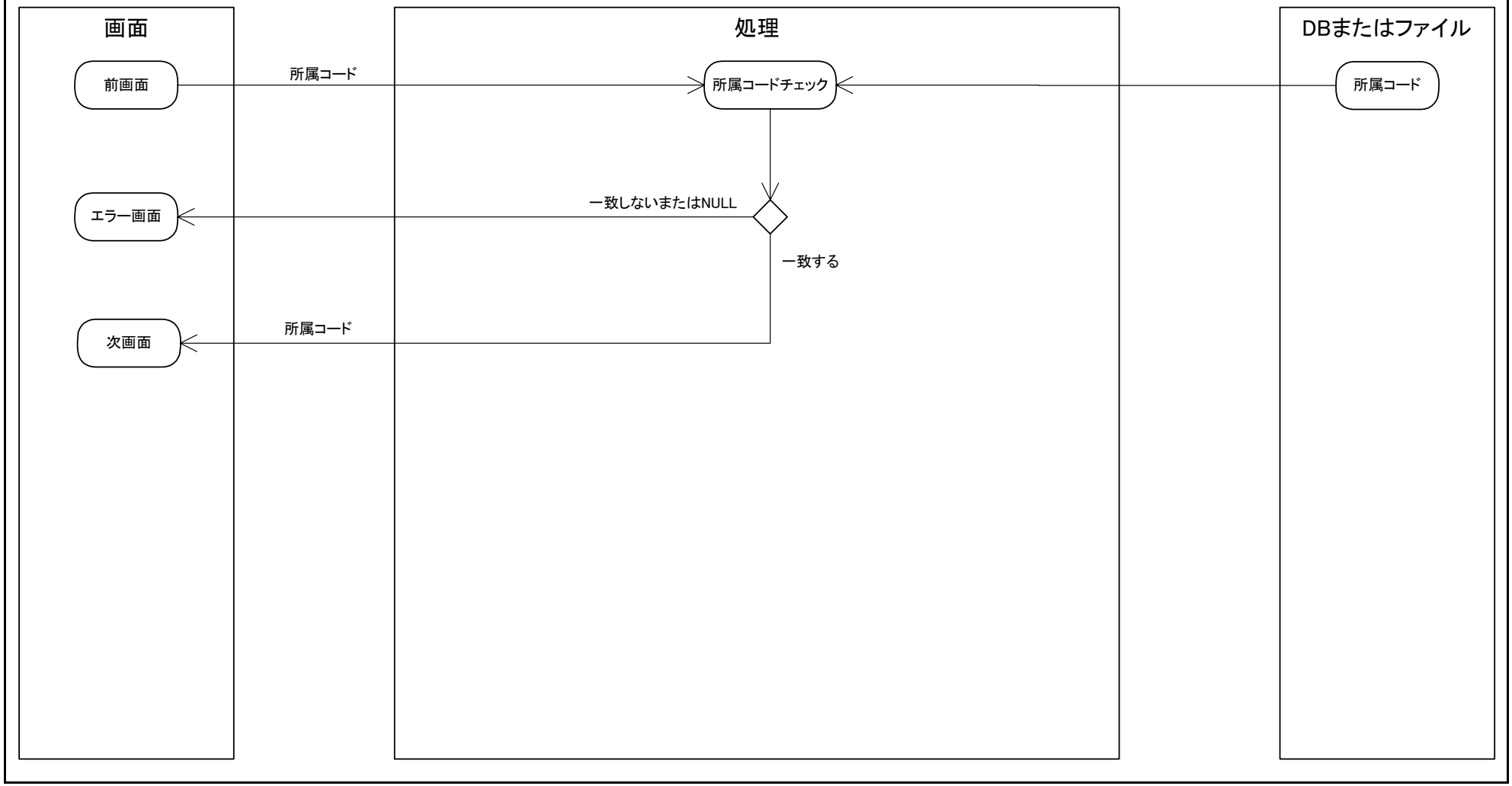


表 3-81　処理フロー

遷移元	全画面共通	アクション	画面遷移時における全てのアクション	遷移先	全画面共通

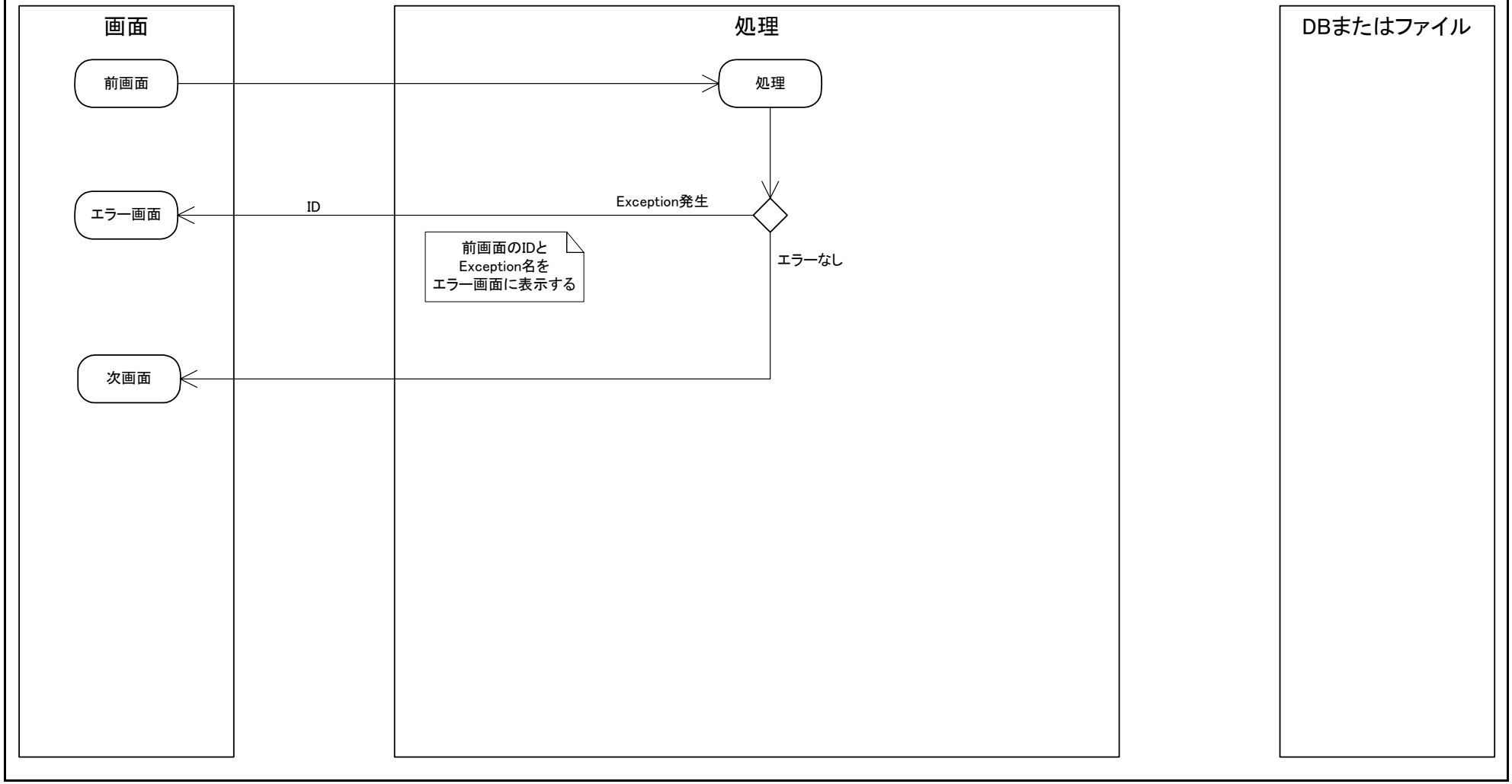


表 3-82　処理フロー

遷移元	トップメニュー	アクション	リンク：散布計画作成	遷移先	標準パターン一覧

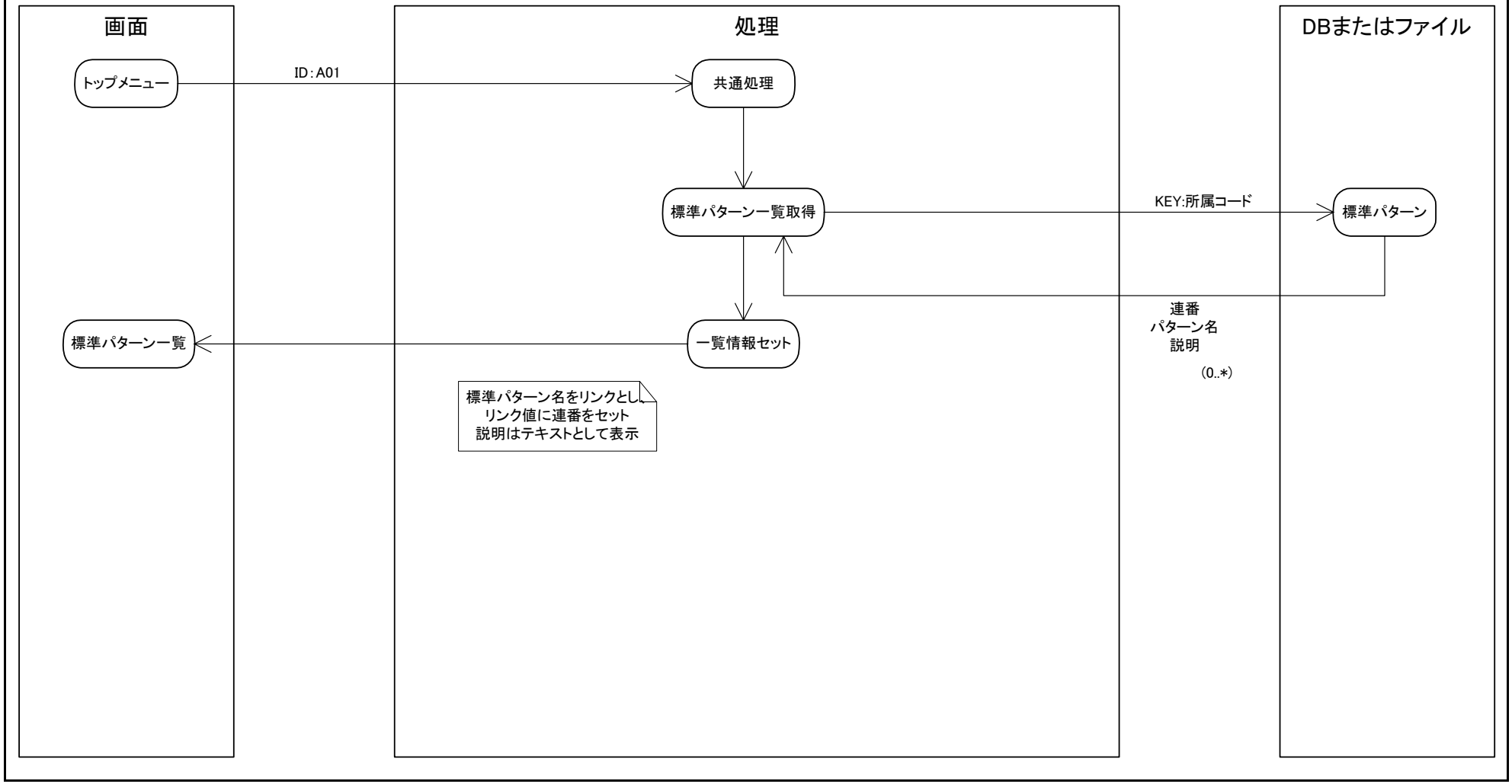


表 3-83　処理フロー

遷移元	標準パターン一覧	アクション	ボタン：戻る	遷移先	トップメニュー

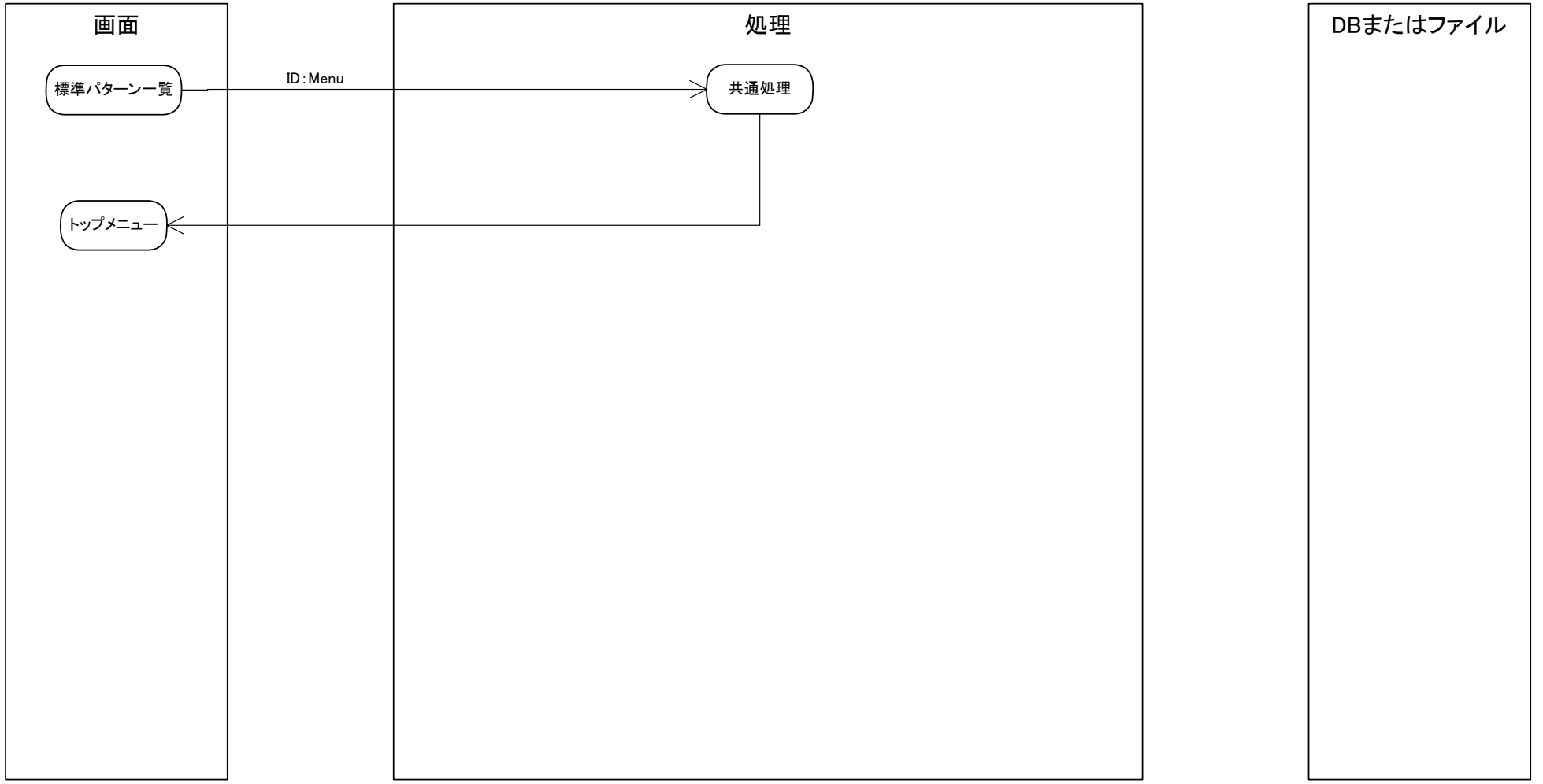


表 3-84　処理フロー

遷移元	標準パターン一覧	アクション	リンク：標準パターン名	遷移先	標準パターン確認

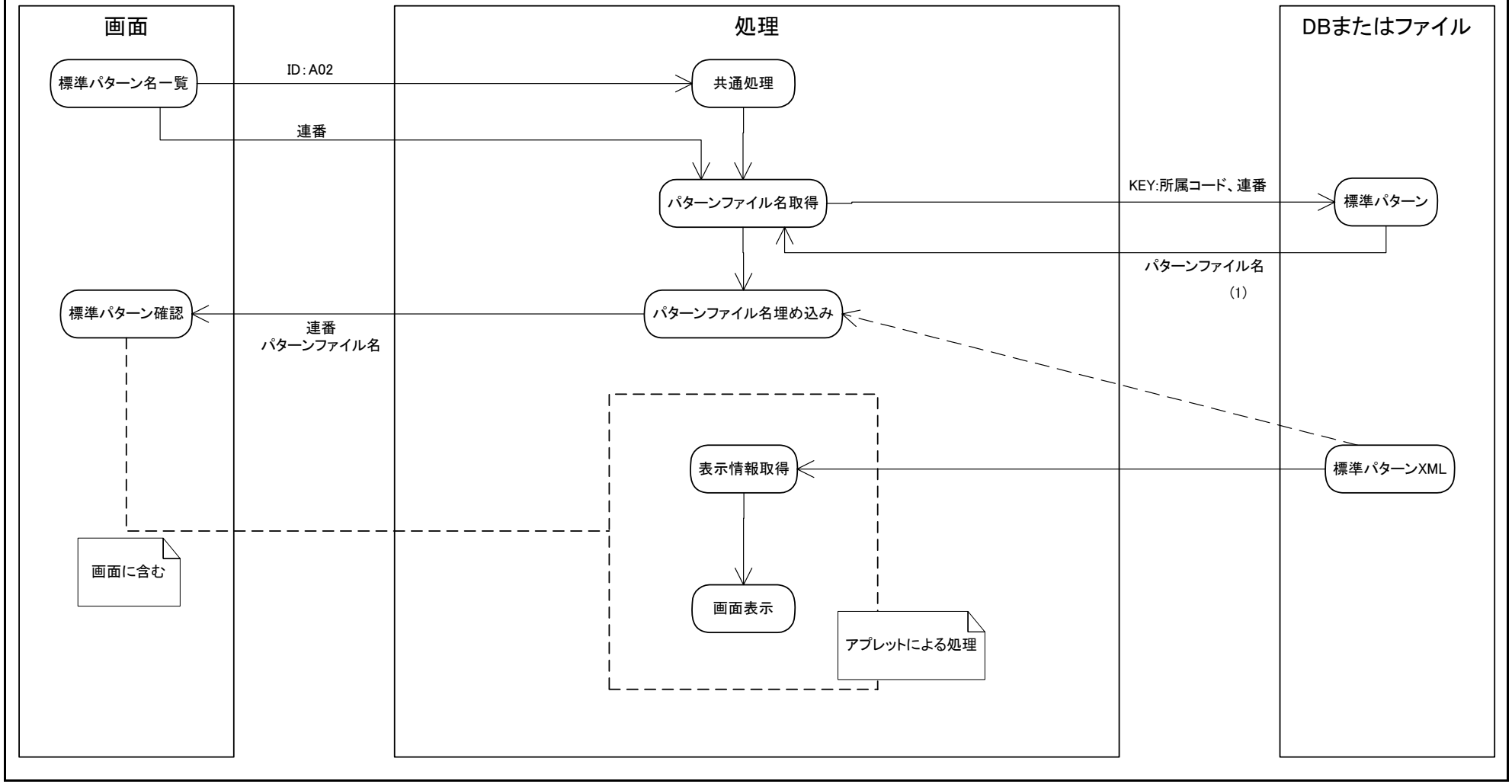


表 3-85　処理フロー

遷移元	標準パターン確認	アクション	ボタン：出力	遷移先	散布計画出力

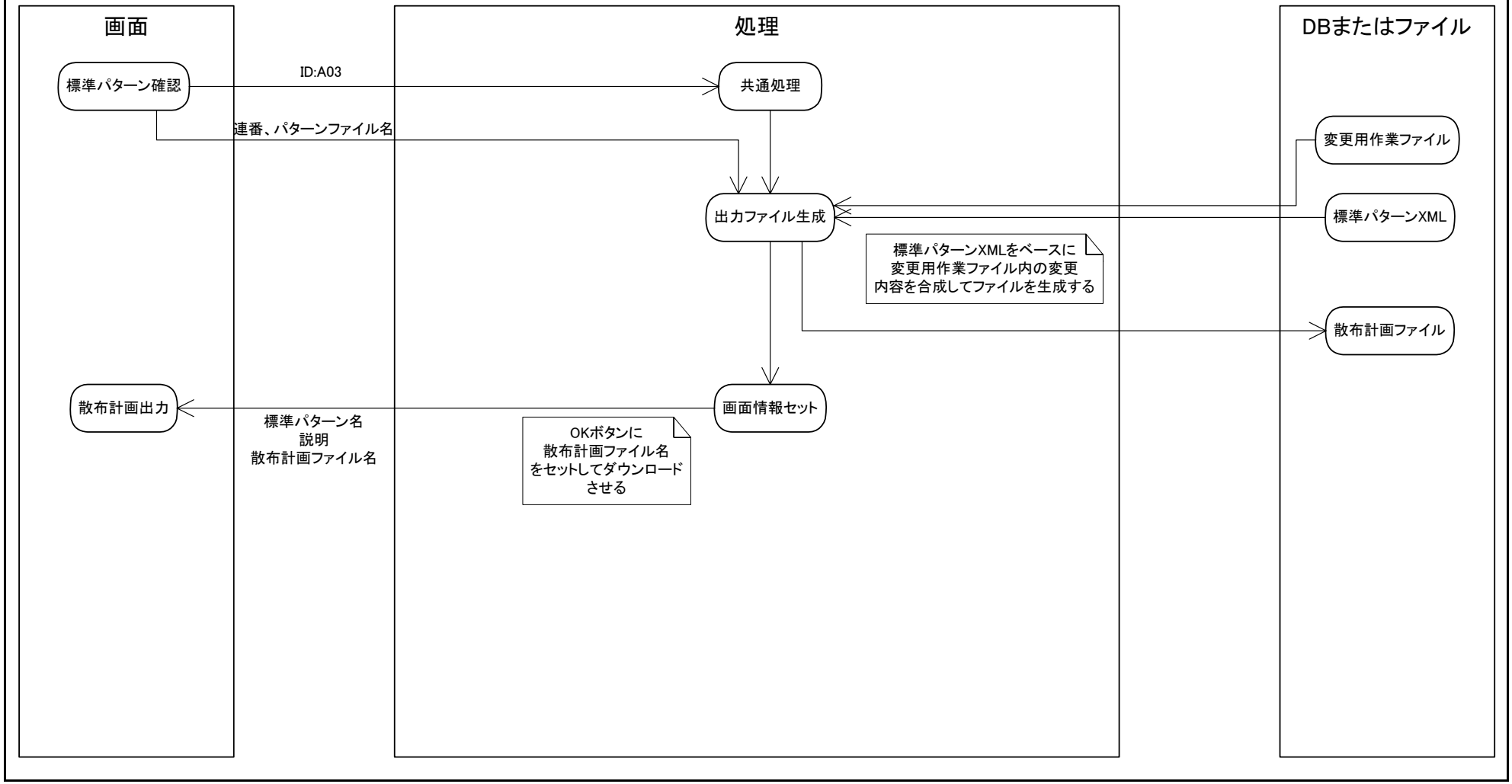


表 3-86　処理フロー

遷移元	標準パターン確認	アクション	ボタン：戻る	遷移先	標準パターン一覧

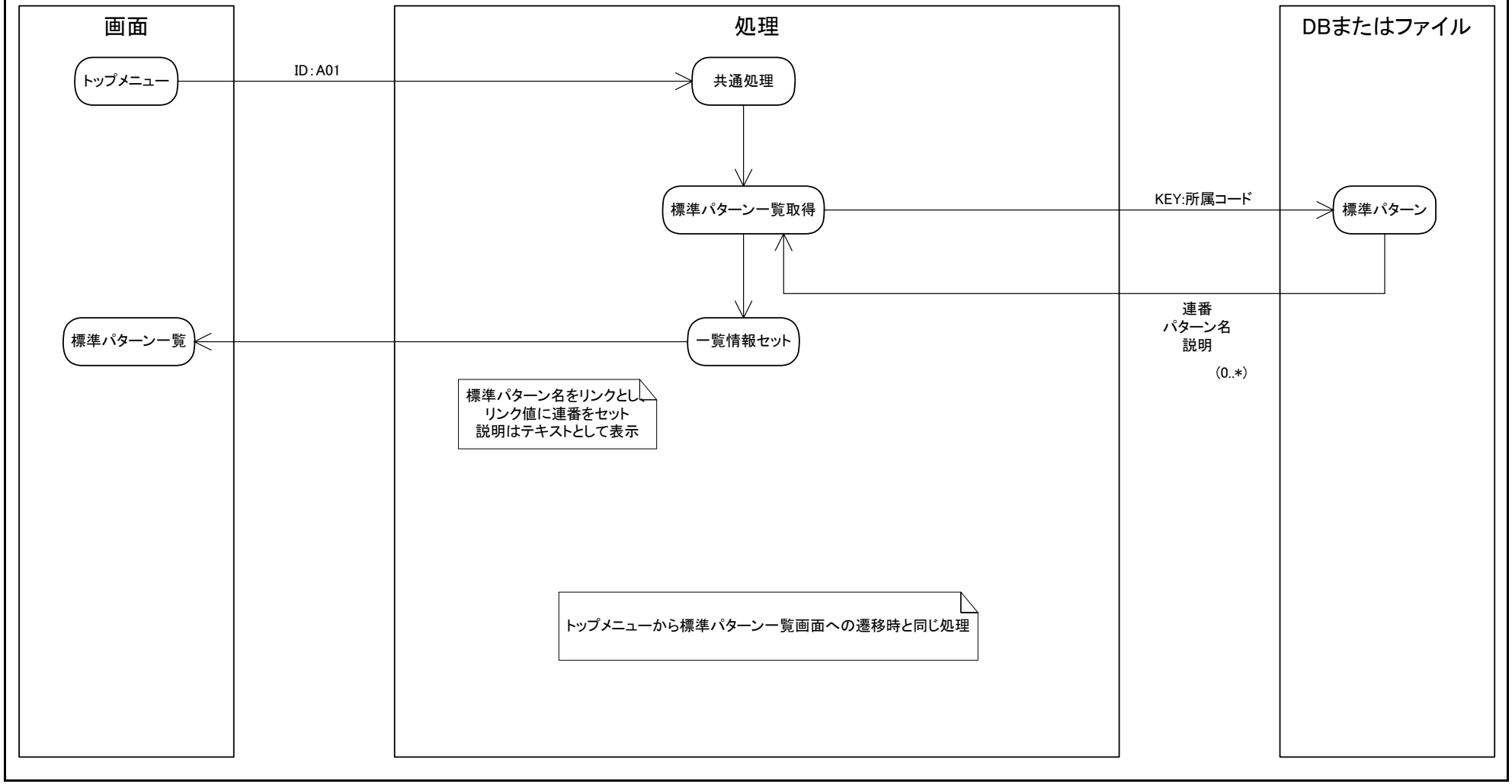


表 3-87　処理フロー

遷移元	散布計画出力	アクション	ボタン：戻る	遷移先	標準パターン確認

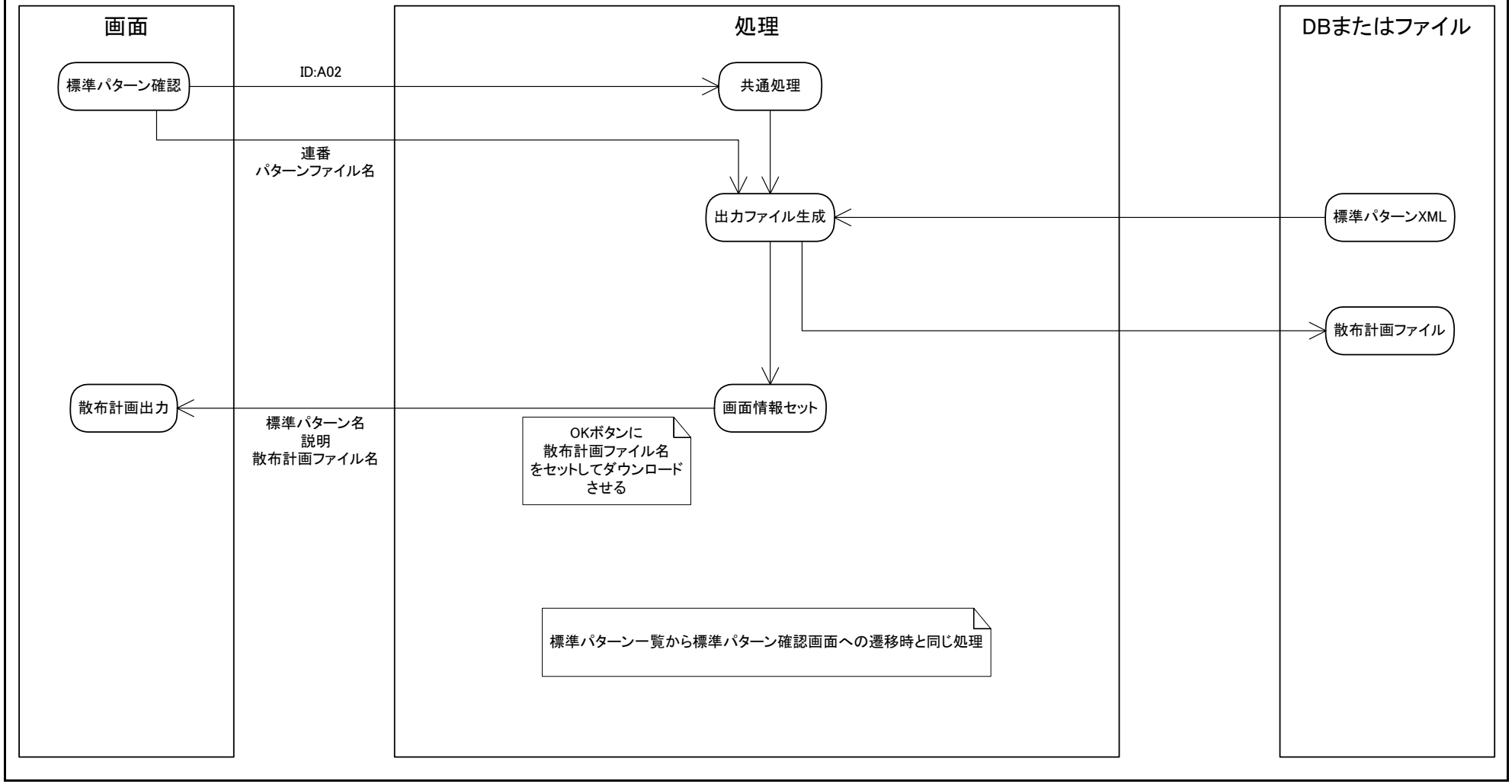


表 3-88　処理フロー

遷移元	散布計画出力	アクション	ボタン：一覧へ戻る	遷移先	標準パターン一覧

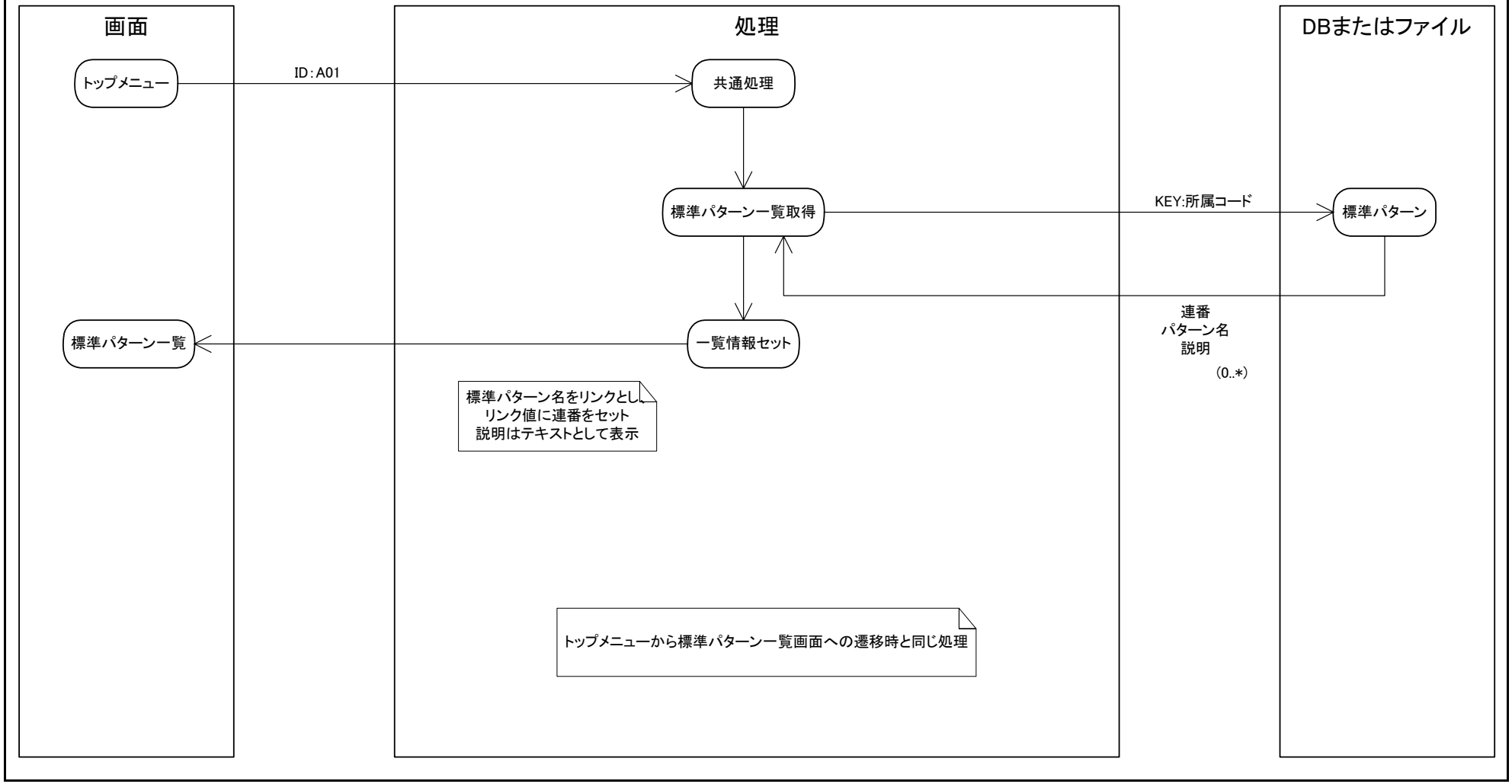


表 3-89　処理フロー

遷移元	散布計画出力	アクション	ボタン：メニューへ	遷移先	トップメニュー

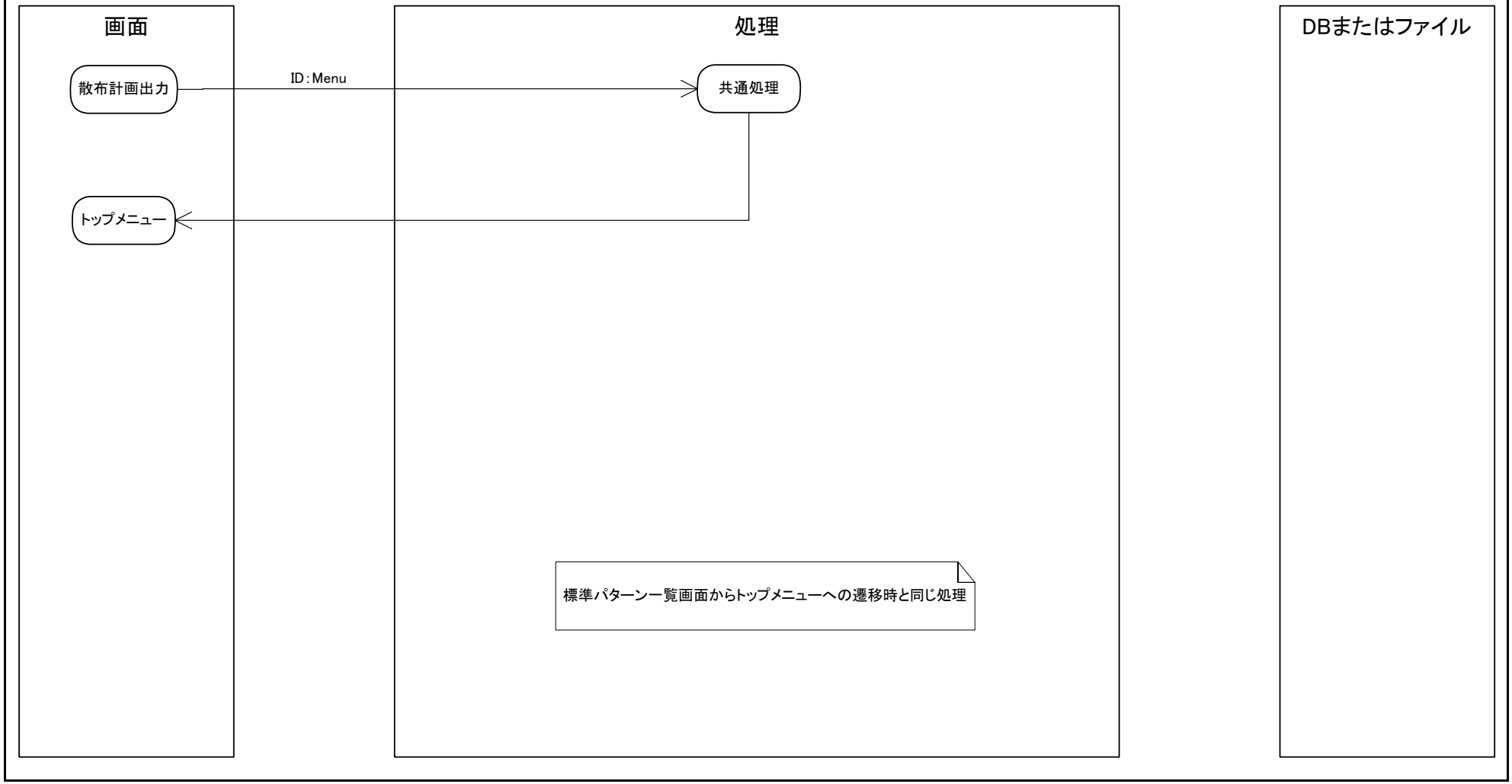


表 3-90　処理フロー

遷移元	トップメニュー	アクション	リンク：散布履歴、巡回記録表示	遷移先	履歴選択

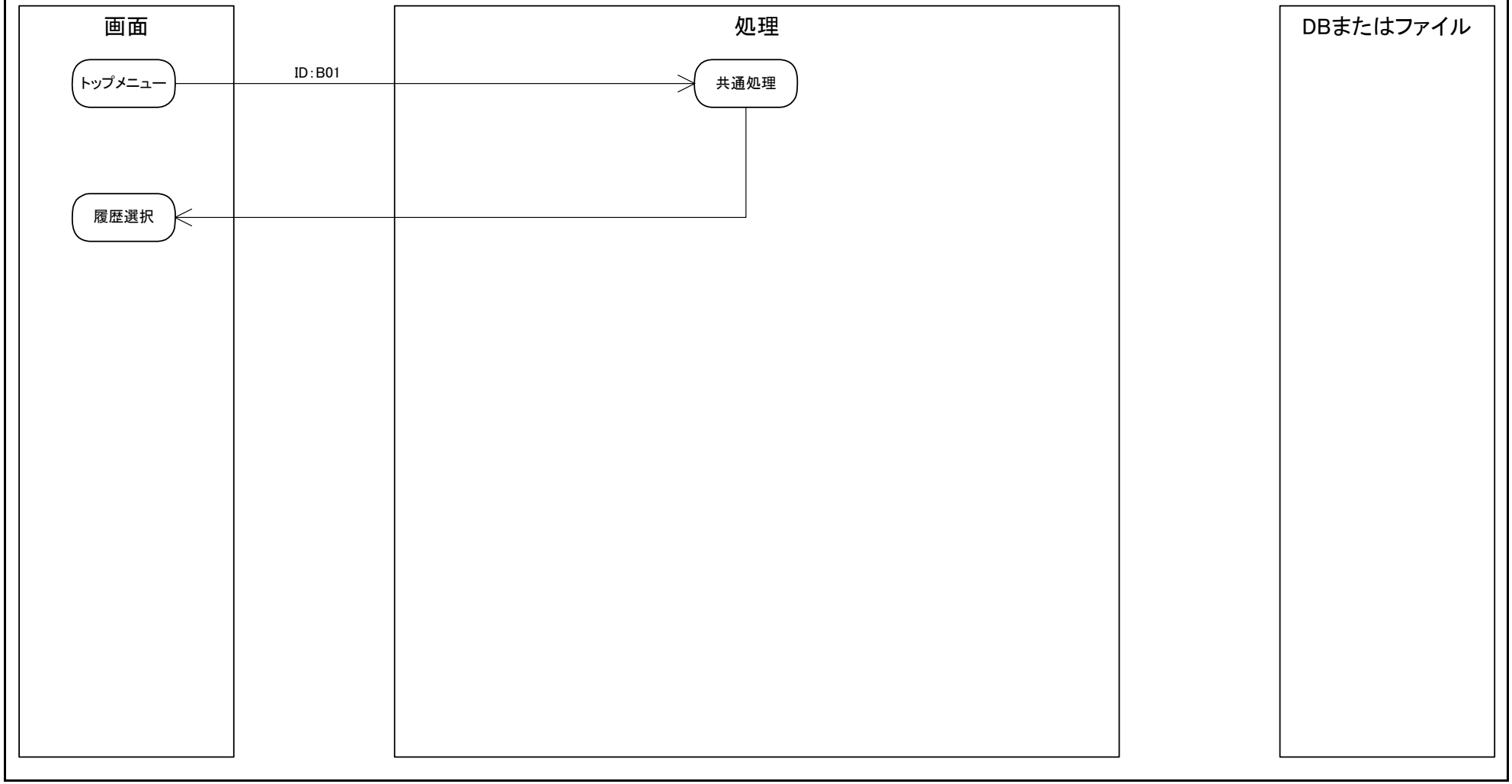


表 3-91　処理フロー

遷移元	履歴選択	アクション	ボタン：戻る	遷移先	トップメニュー

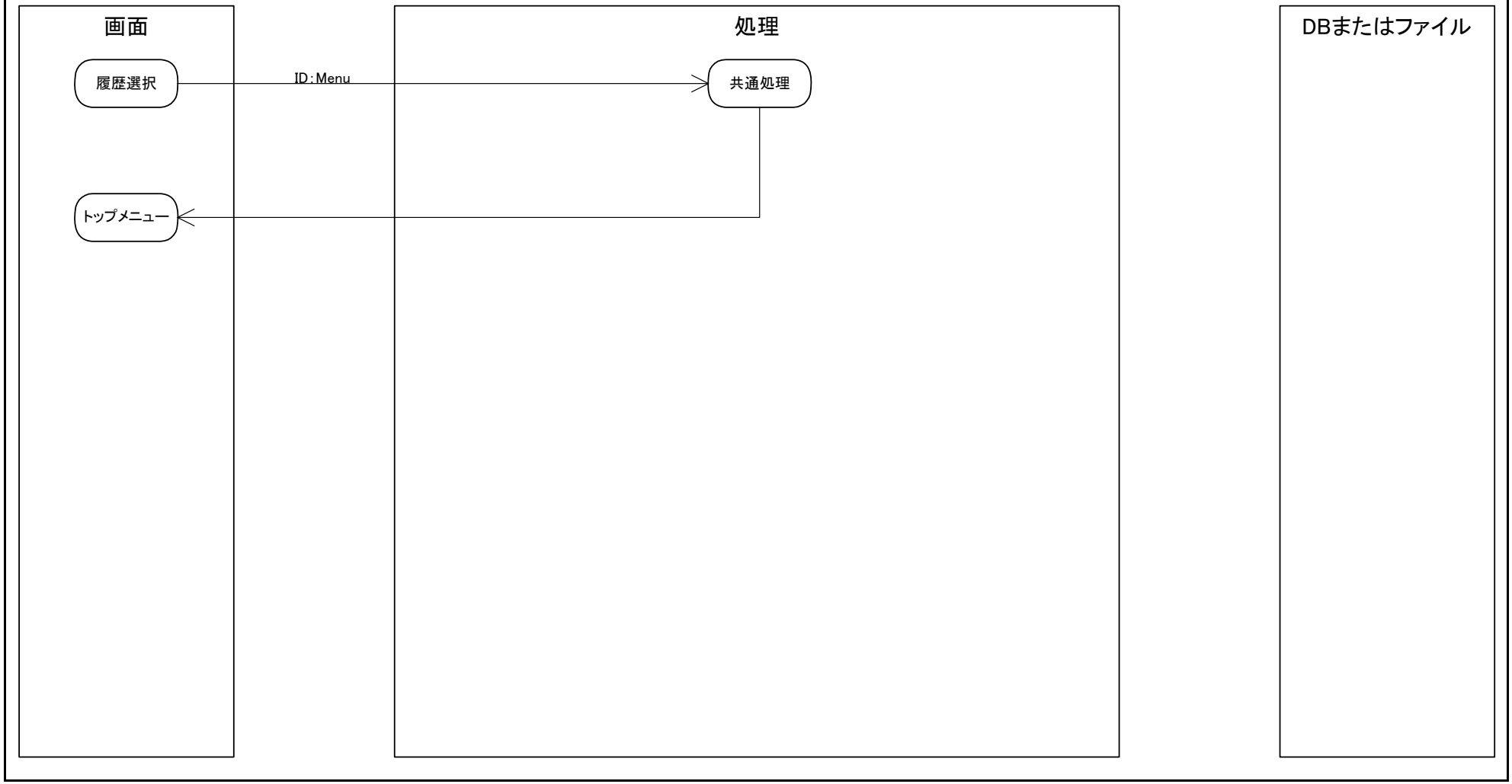


表 3-92　処理フロー

遷移元	履歴選択	アクション	リンク：最新作業履歴表示	遷移先	散布履歴表示

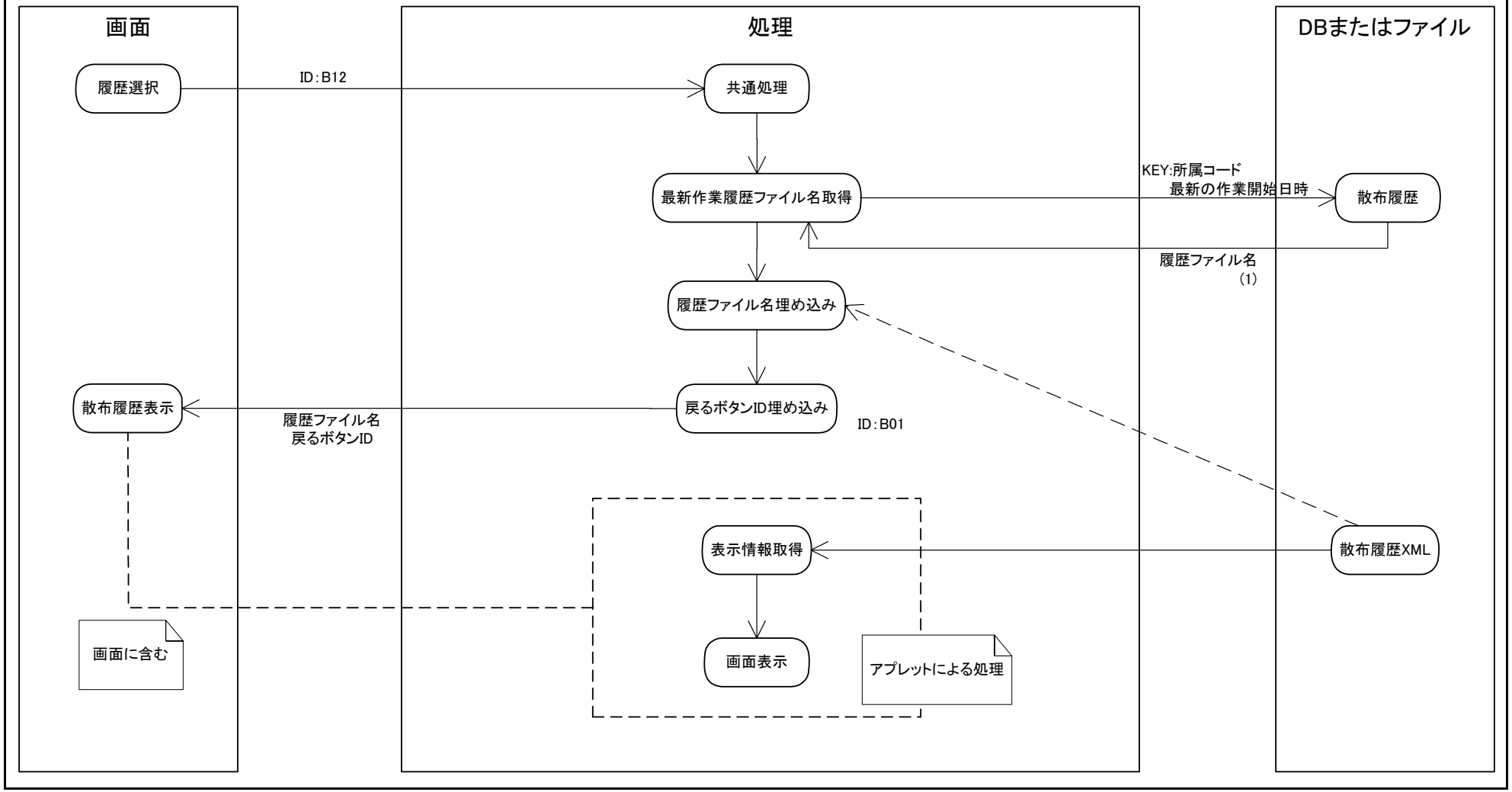


表 3-93　処理フロー

遷移元	散布履歴表示	アクション	ボタン：戻る（散布履歴日時選択画面経由なし）	遷移先	散布履歴

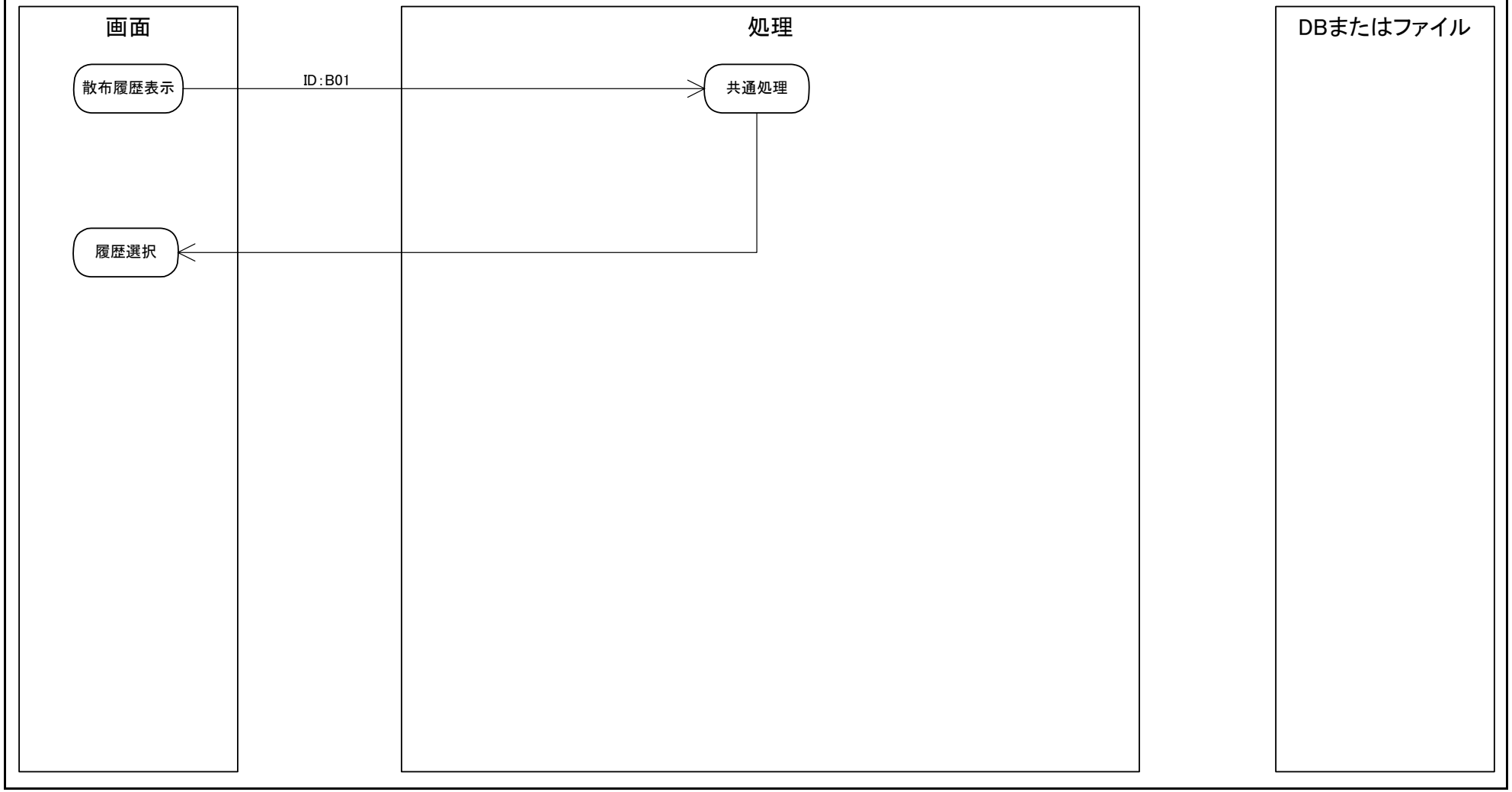


表 3-94　処理フロー

遷移元	履歴選択	アクション	リンク：過去作業履歴選択	遷移先	散布履歴日時選択

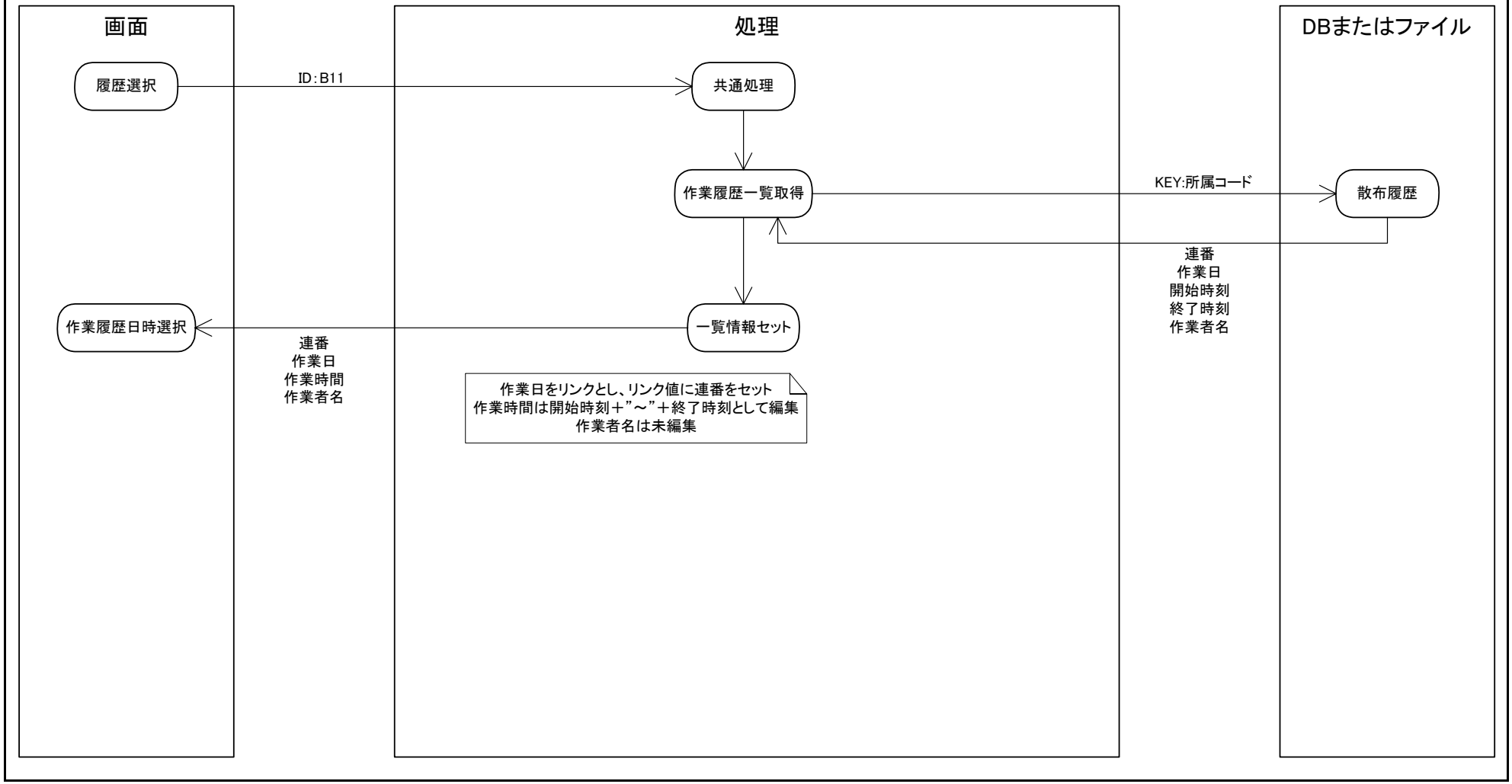


表 3-95　処理フロー

遷移元	散布履歴日時選択	アクション	ボタン：戻る	遷移先	履歴選択

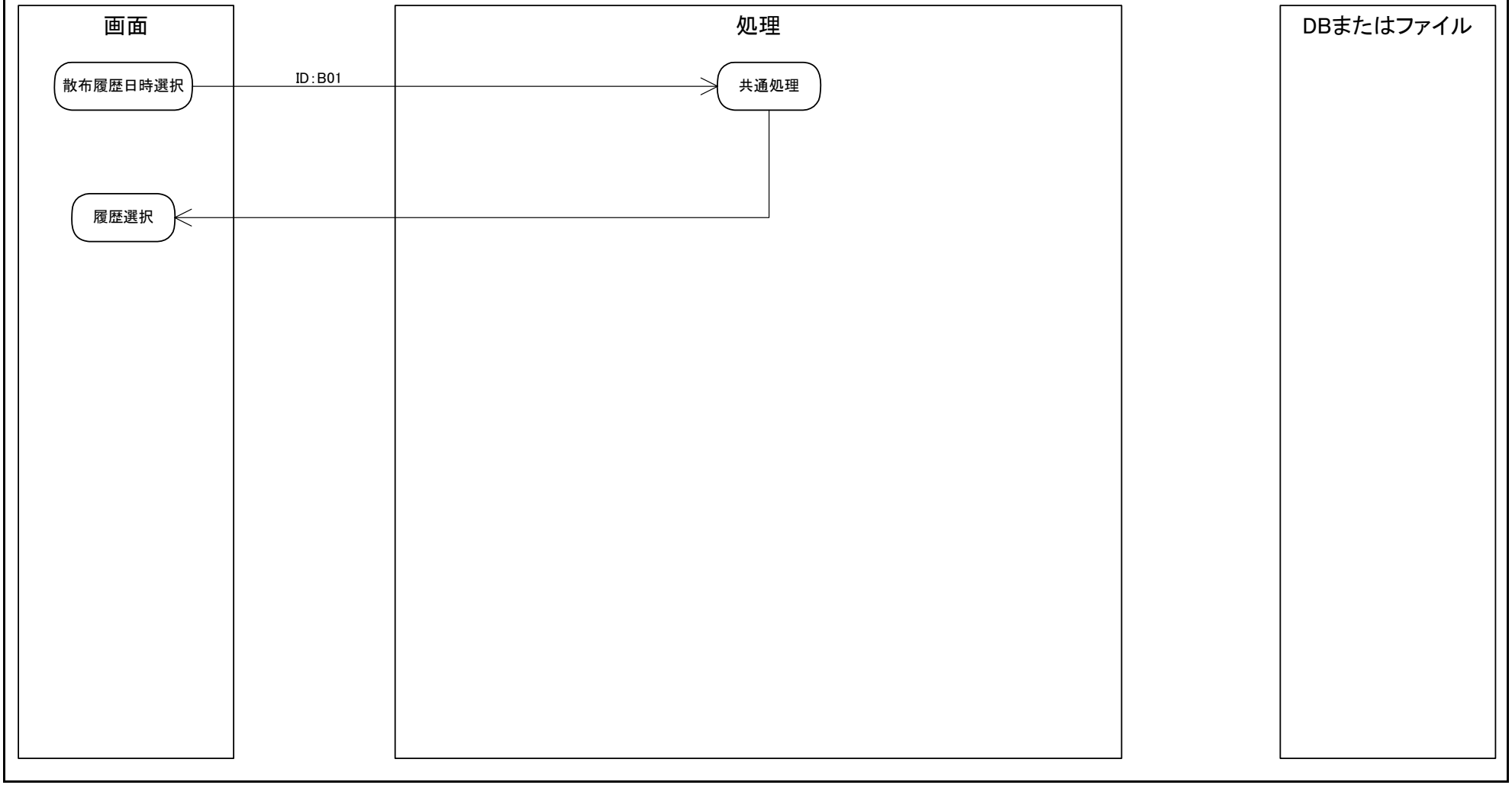


表 3-96　処理フロー

遷移元	散布履歴日時選択	アクション	リンク：作業日	遷移先	散布履歴表示

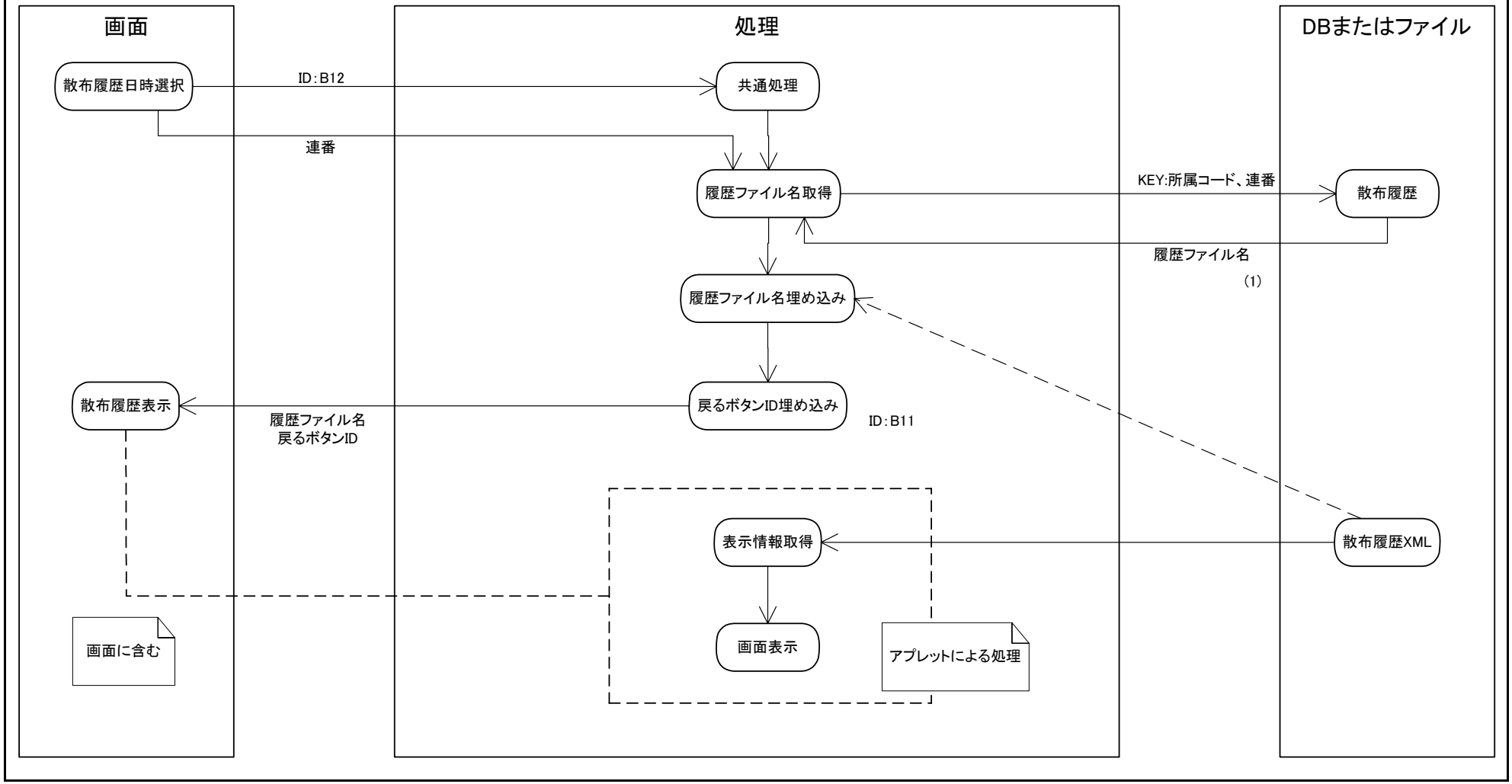


表 3-97　処理フロー

遷移元	散布履歴表示	アクション	ボタン：戻る（散布履歴日時選択画面経由あり）	遷移先	散布履歴日時選択

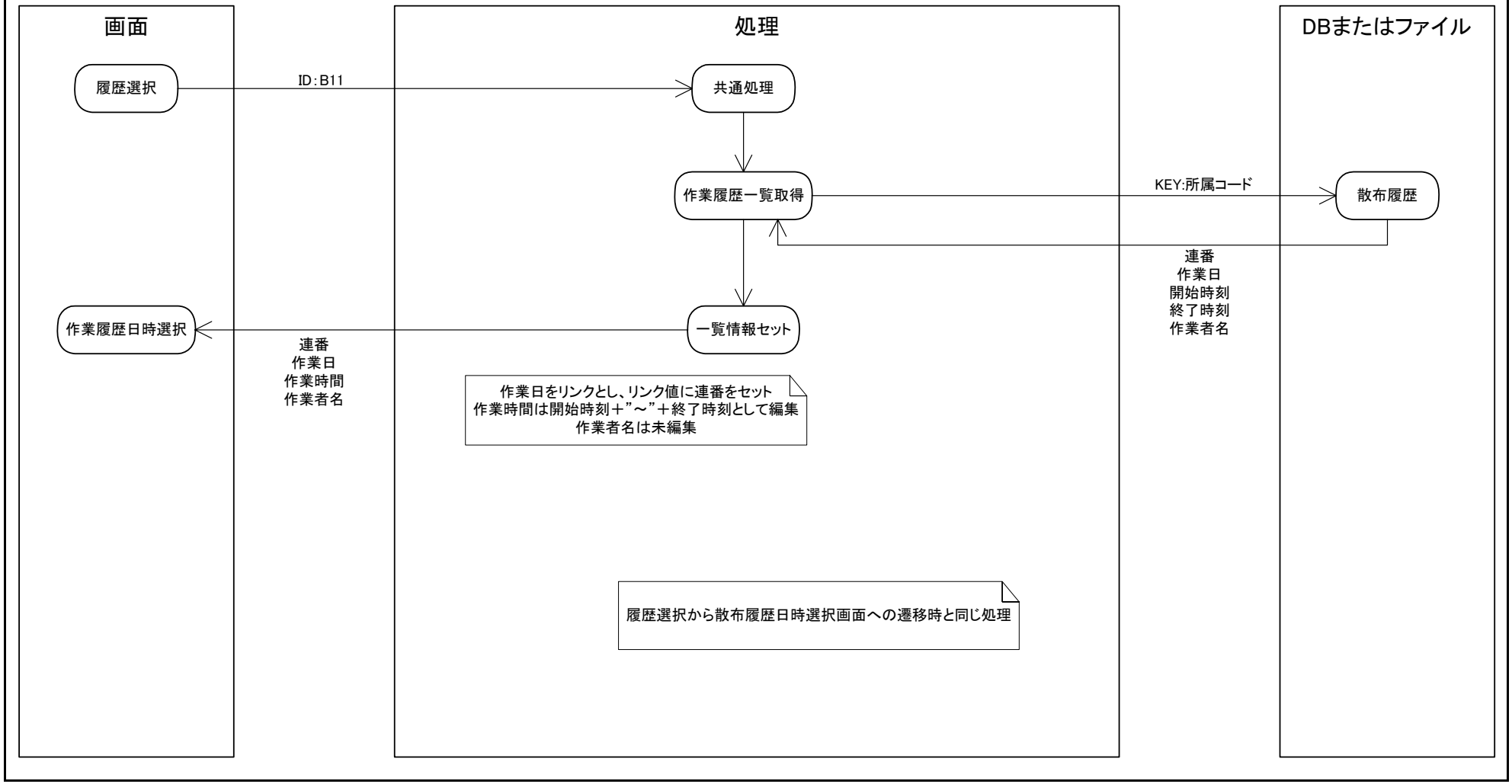


表 3-98　処理フロー

遷移元	散布履歴表示	アクション	ボタン：メニューへ	遷移先	トップメニュー

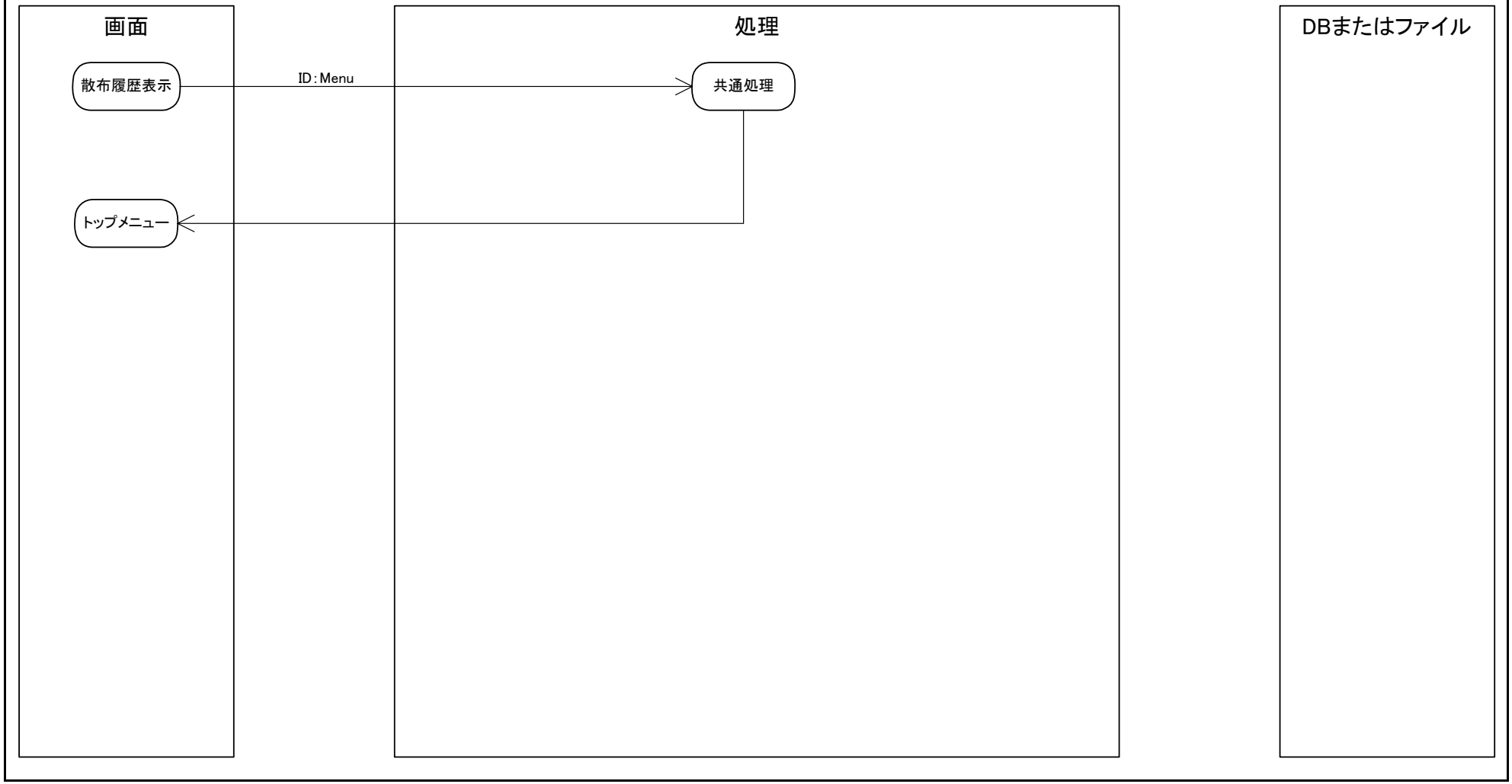


表 3-99　処理フロー

遷移元	履歴選択	アクション	リンク：最新巡回記録表示	遷移先	巡回記録表示

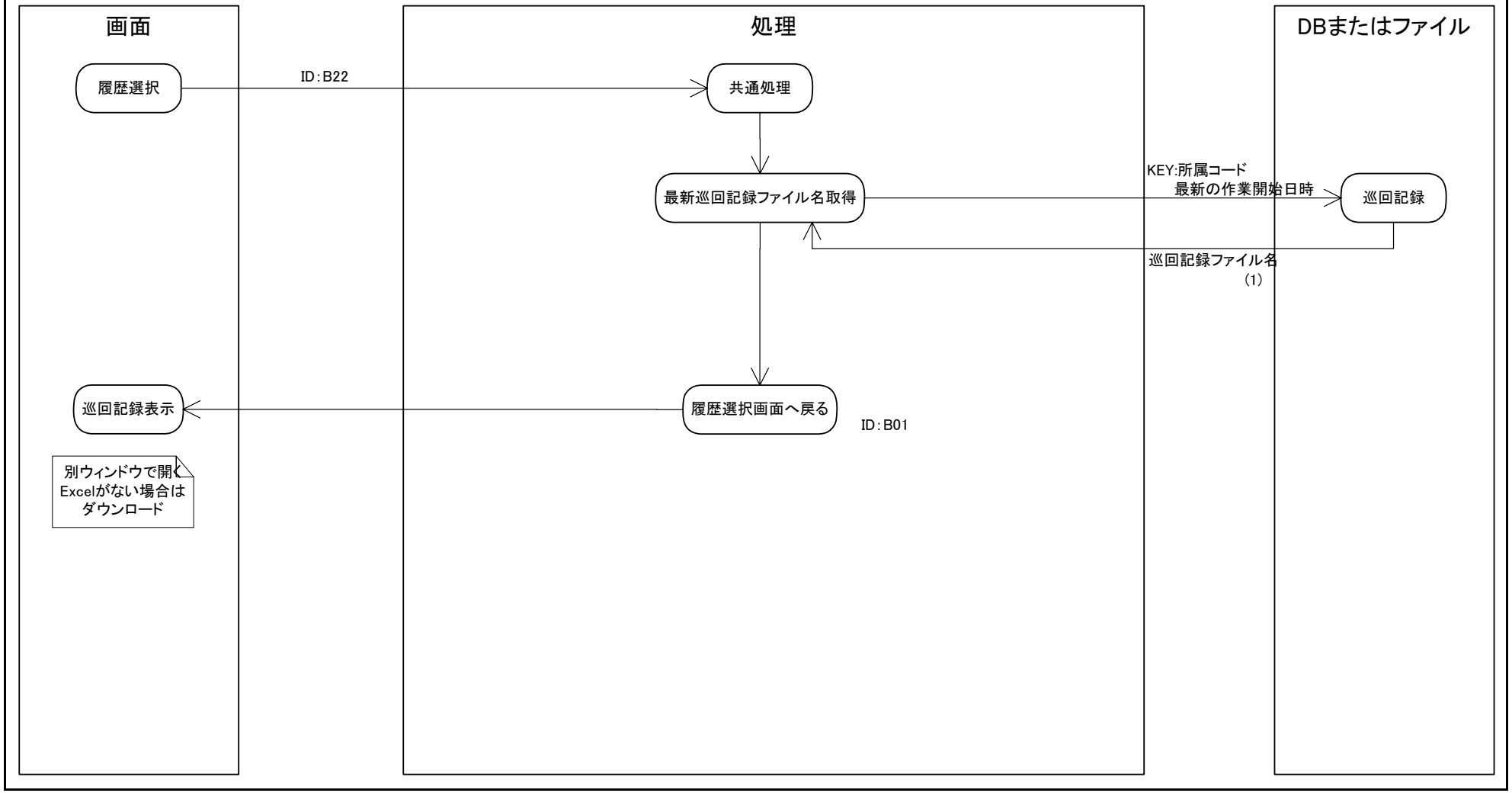


表 3-100　処理フロー

遷移元	履歴選択	アクション	リンク：過去巡回記録選択	遷移先	巡回記録日時選択

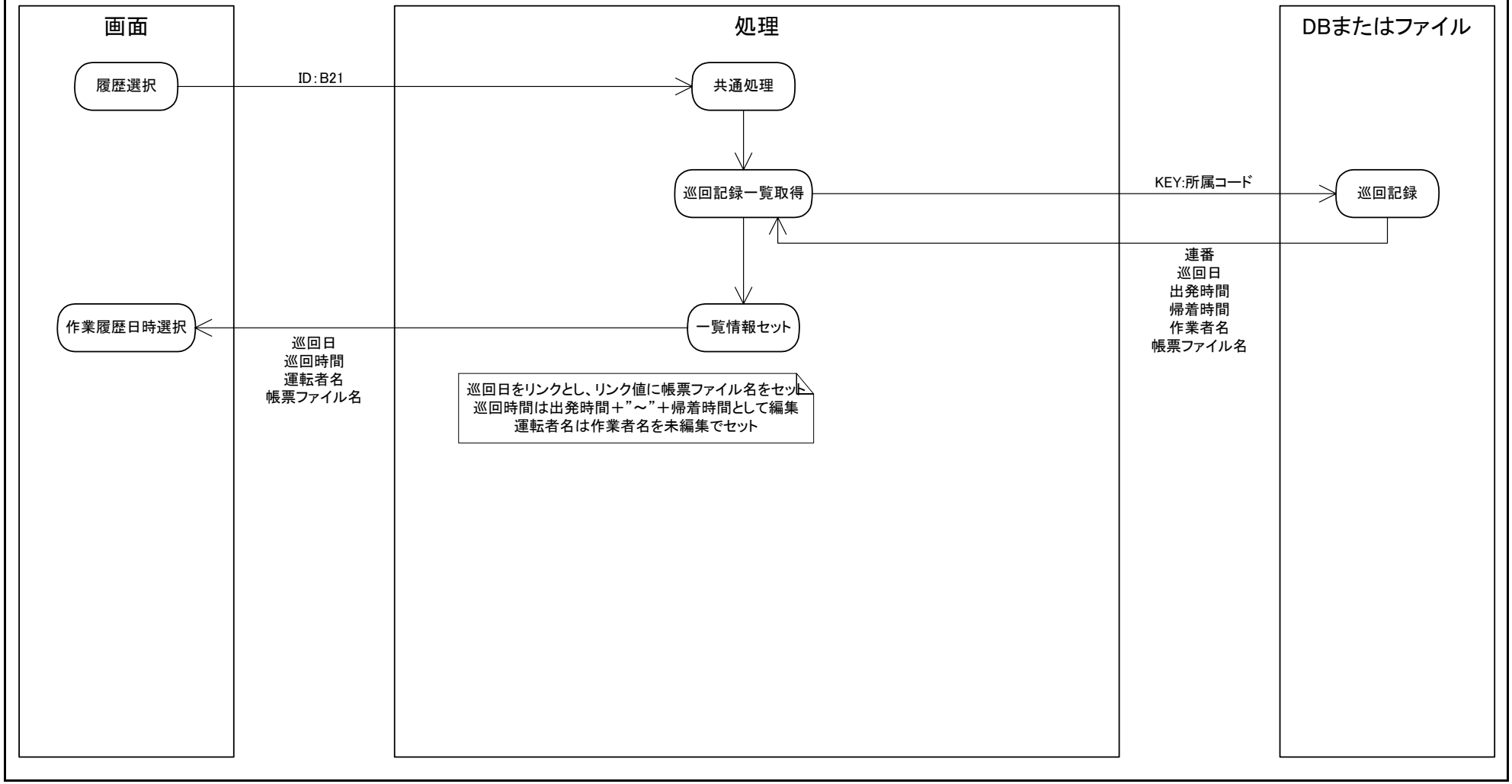


表 3-101　処理フロー

遷移元	巡回記録日時選択	アクション	ボタン：戻る	遷移先	履歴選択

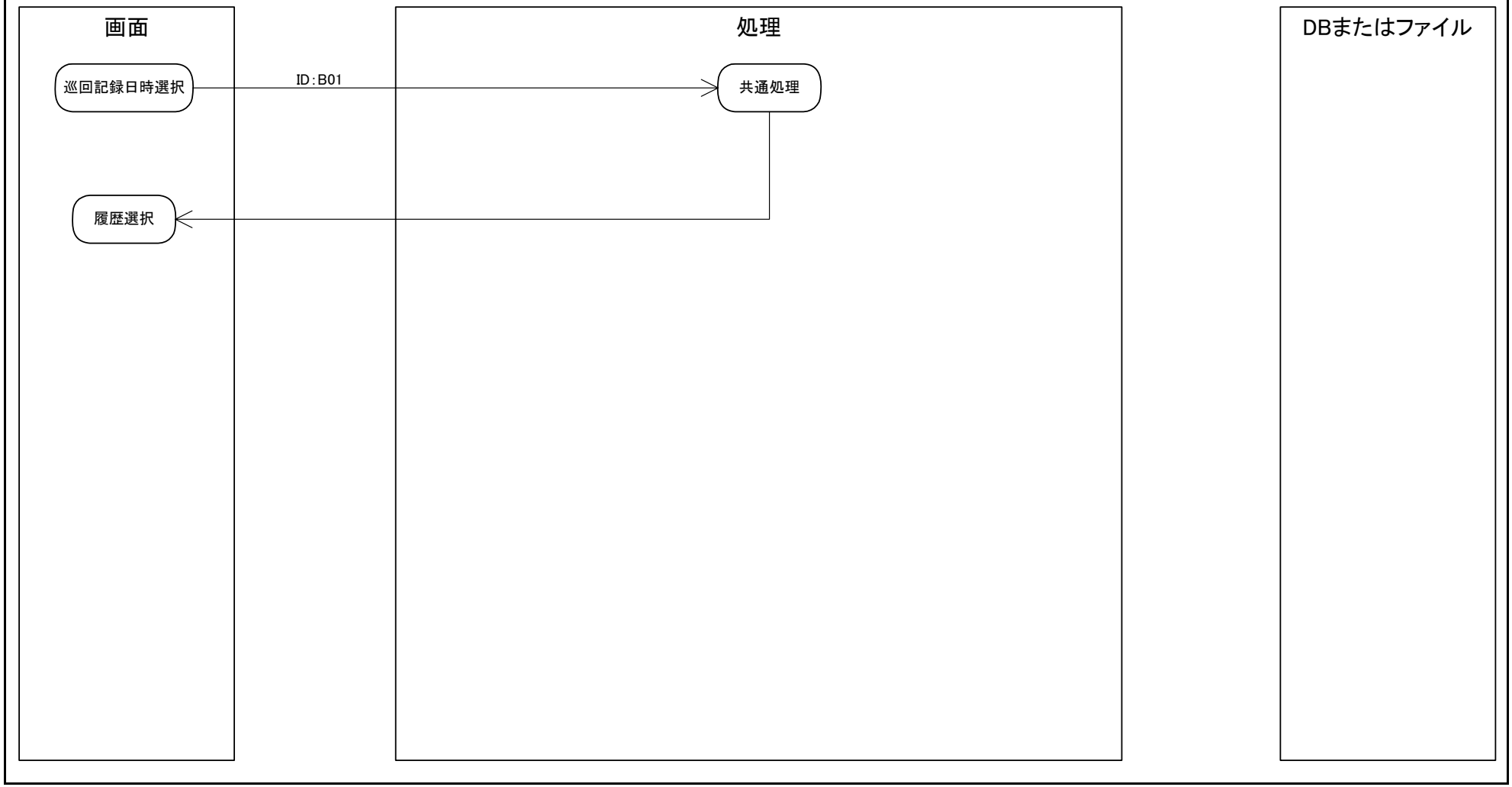


表 3-102　処理フロー

遷移元	巡回記録日時選択	アクション	リンク：巡回日	遷移先	巡回記録（別ウィンドウ）

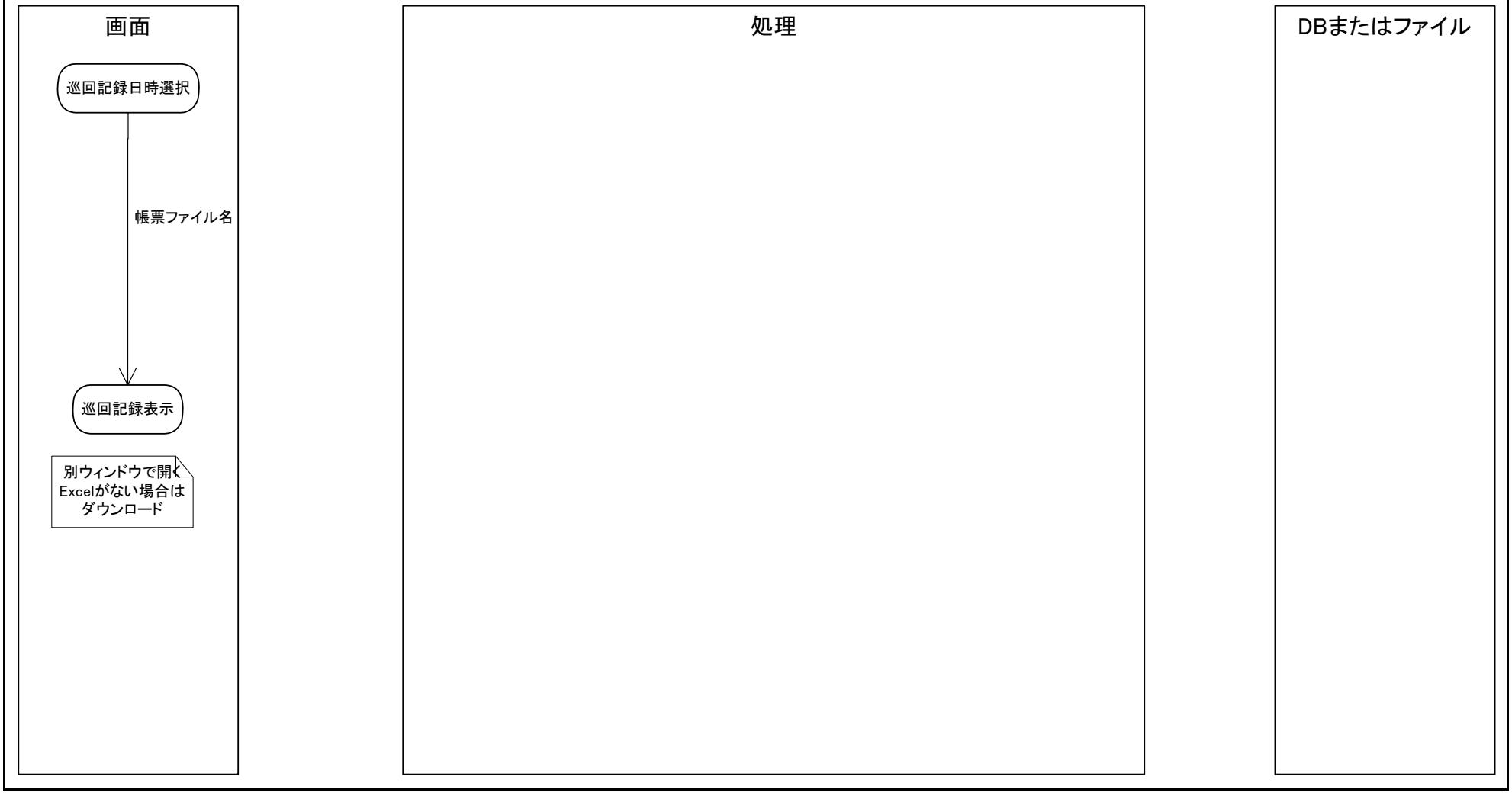


表 3-103　処理フロー

遷移元	履歴選択	アクション	リンク：巡回記録修正版の登録	遷移先	巡回記録修正版選択

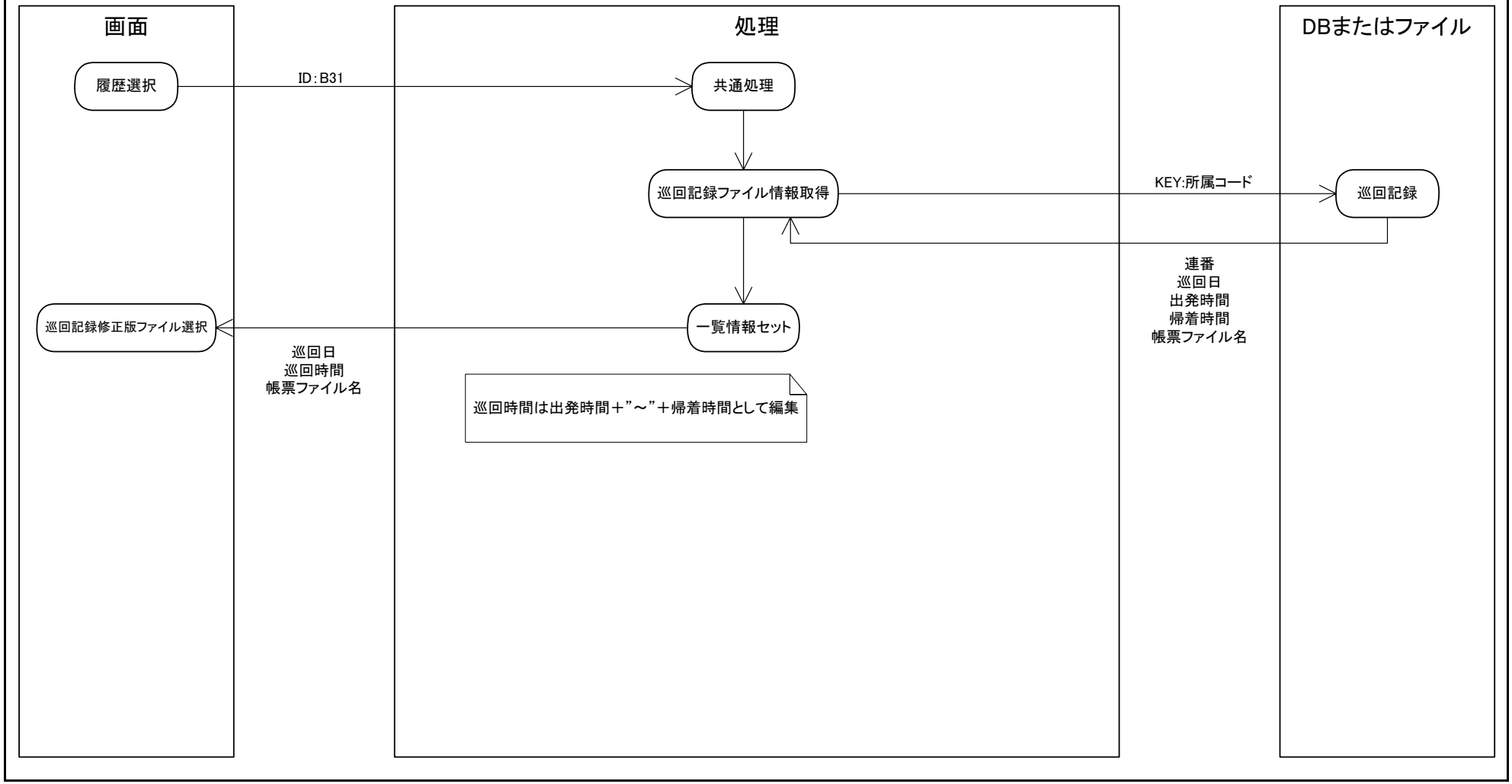


表 3-104　処理フロー

遷移元	巡回記録修正版選択	アクション	ボタン：戻る	遷移先	履歴選択

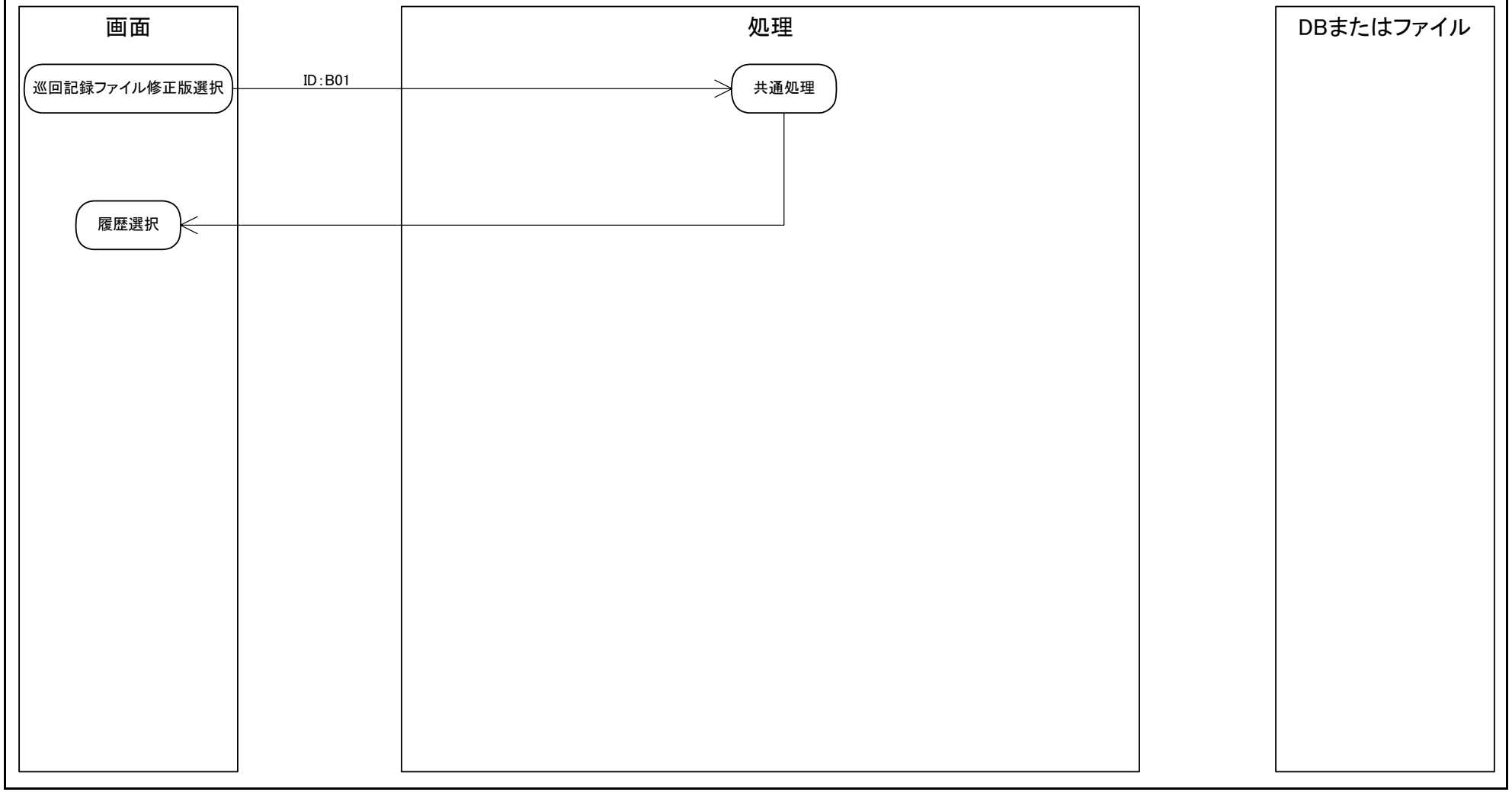


表 3-105　処理フロー

遷移元	巡回記録ファイル修正版選択	アクション	ボタン：送信	遷移先	巡回記録ファイル修正版登録完了

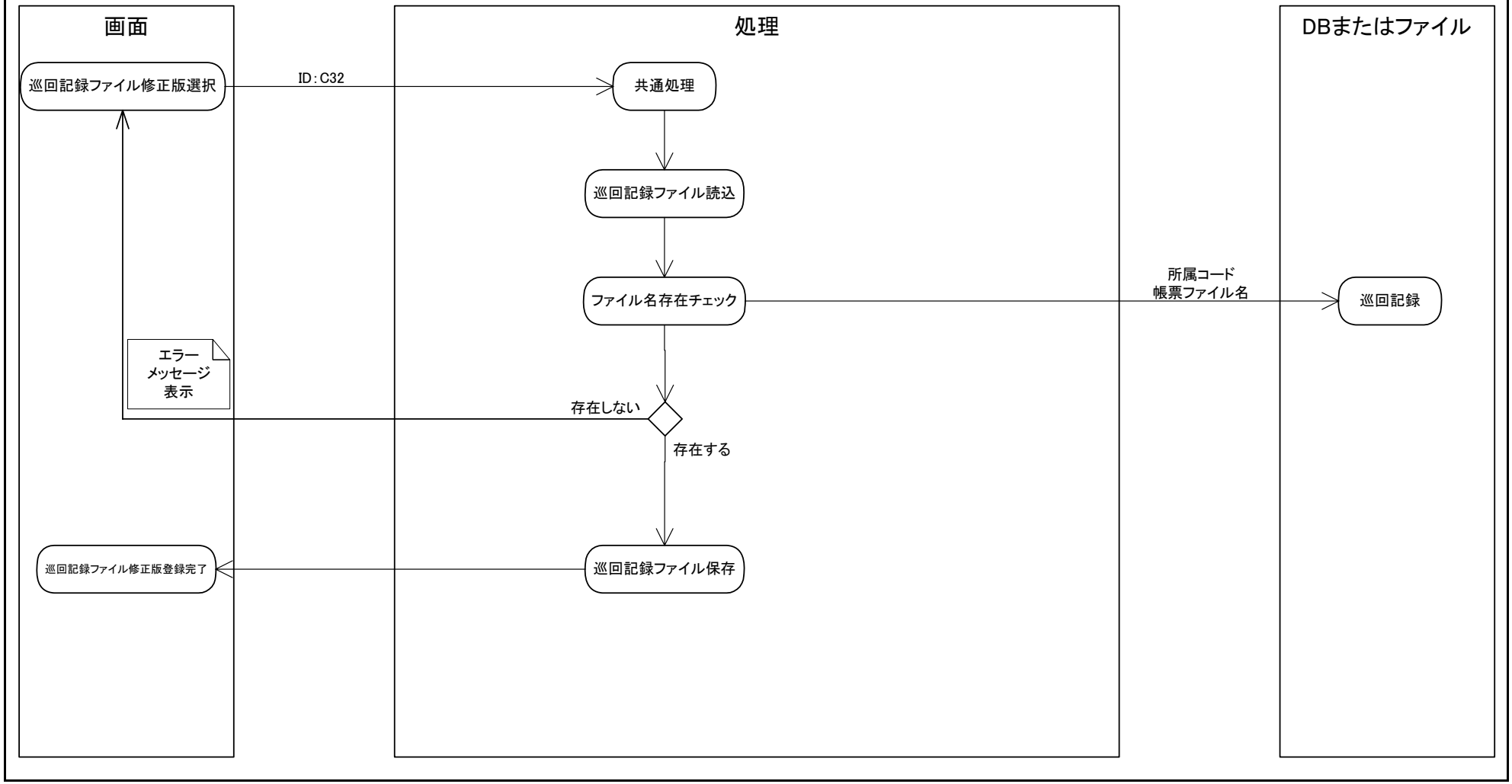


表 3-106　処理フロー

遷移元	巡回記録ファイル修正版登録完了	アクション	ボタン：戻る	遷移先	履歴選択

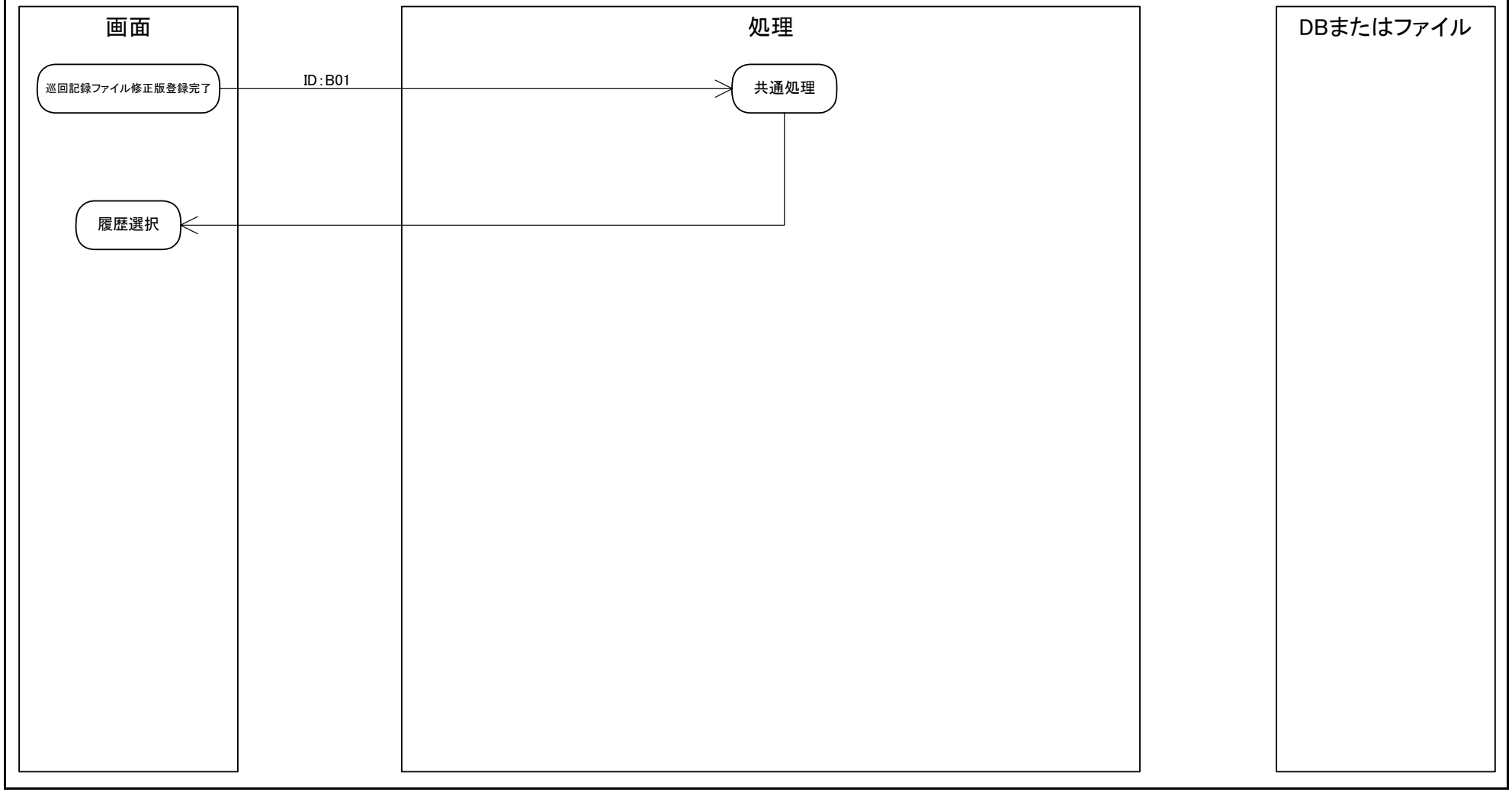


表 3-107　処理フロー

遷移元	巡回記録ファイル修正版登録完了	アクション	ボタン：メニューへ	遷移先	トップメニュー

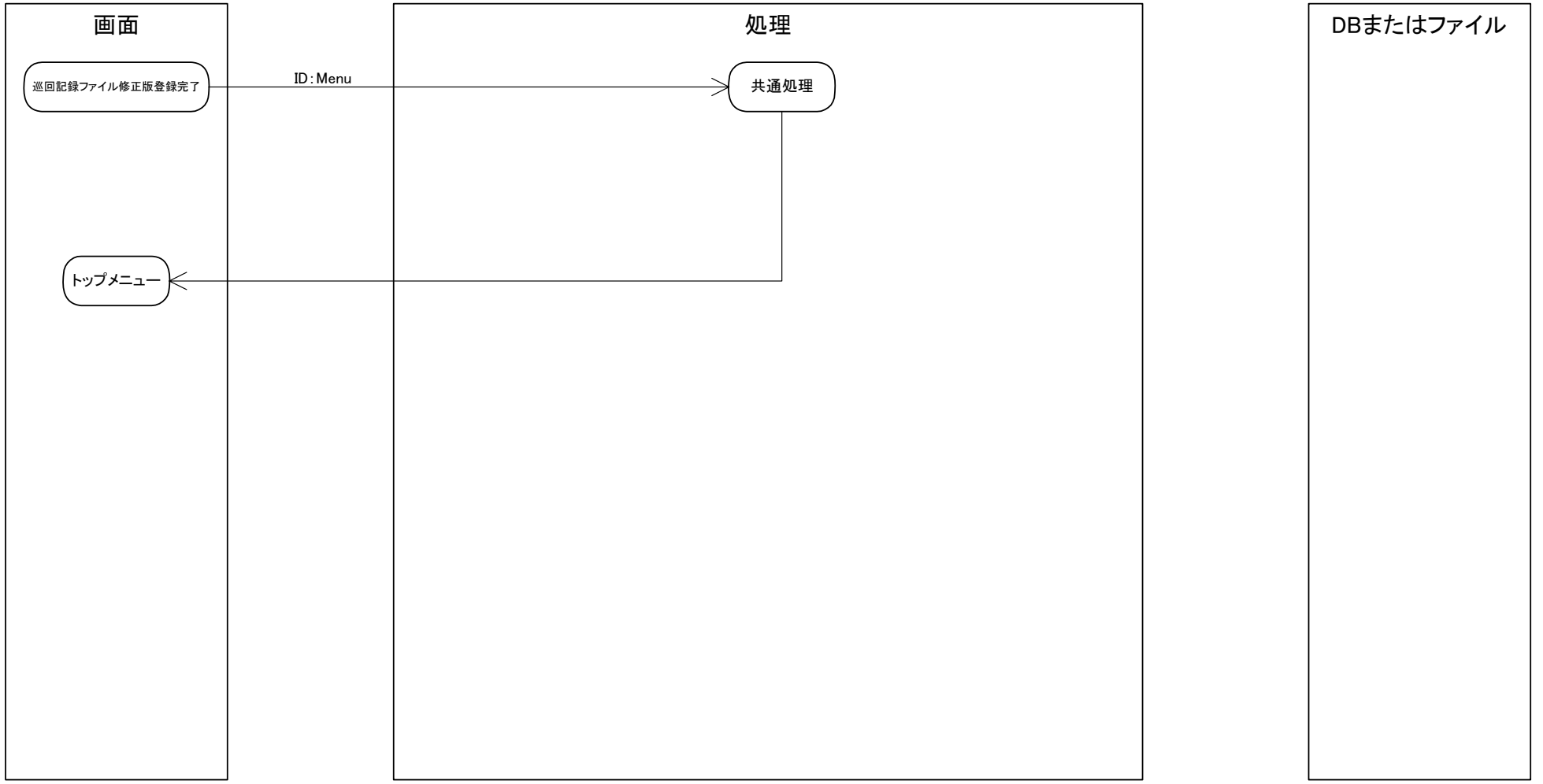


表 3-108　処理フロー

遷移元	トップメニュー	アクション	リンク：散布結果、巡回記録の登録	遷移先	登録ファイル種別選択

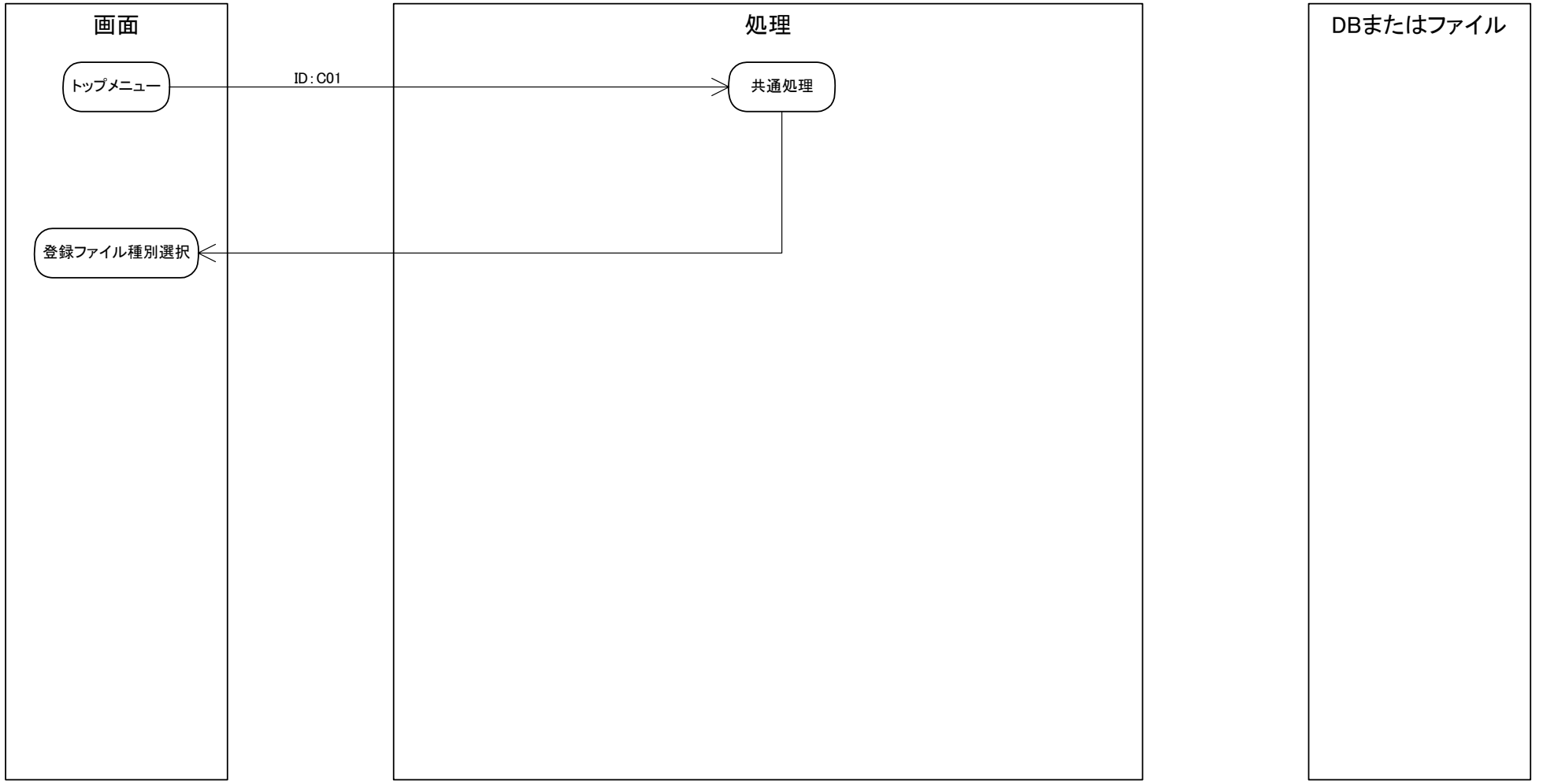


表 3-109　処理フロー

遷移元	登録ファイル種別選択	アクション	ボタン：戻る	遷移先	トップメニュー

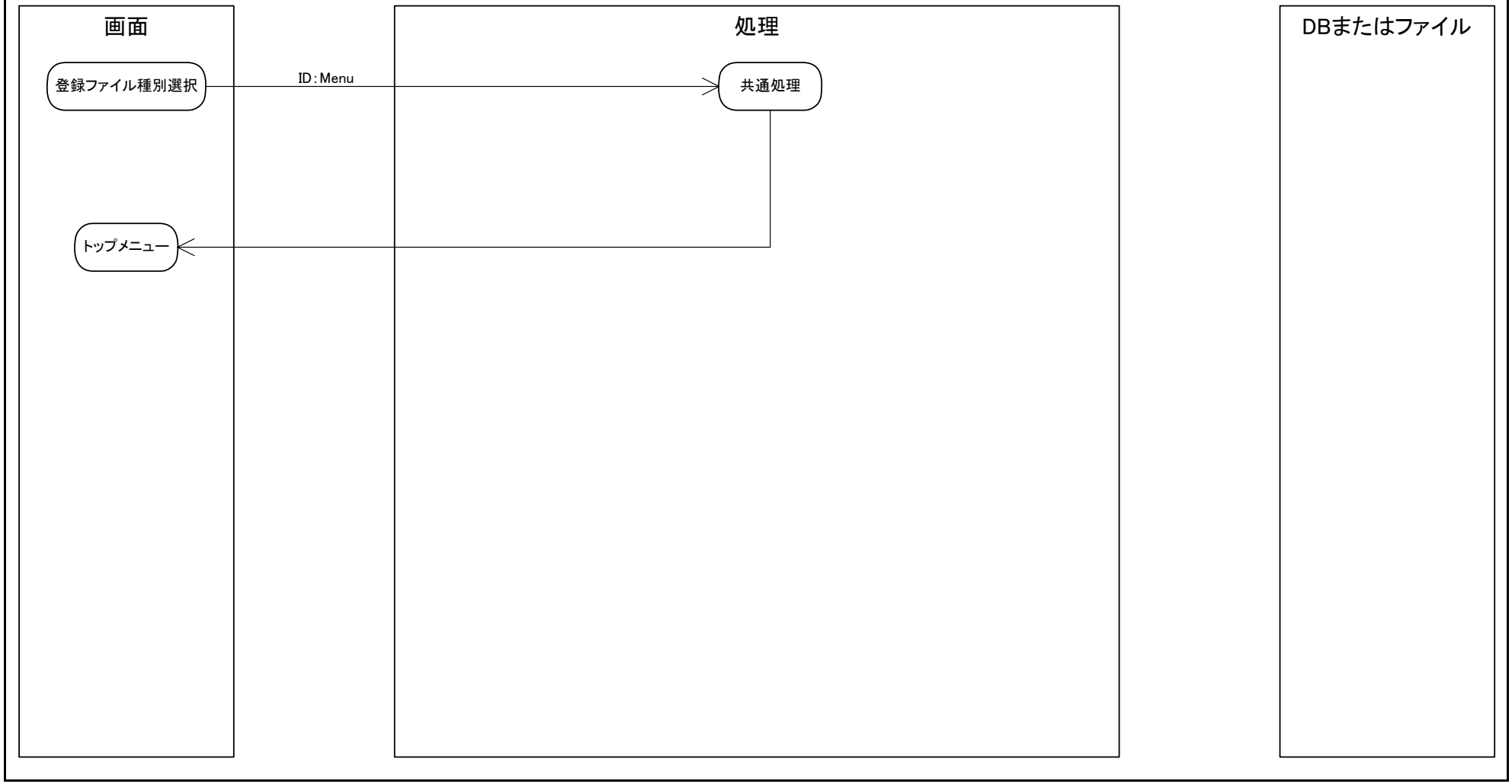


表 3-110　処理フロー

遷移元	登録ファイル種別選択	アクション	リンク：散布作業記録の登録	遷移先	散布作業記録ファイル選択

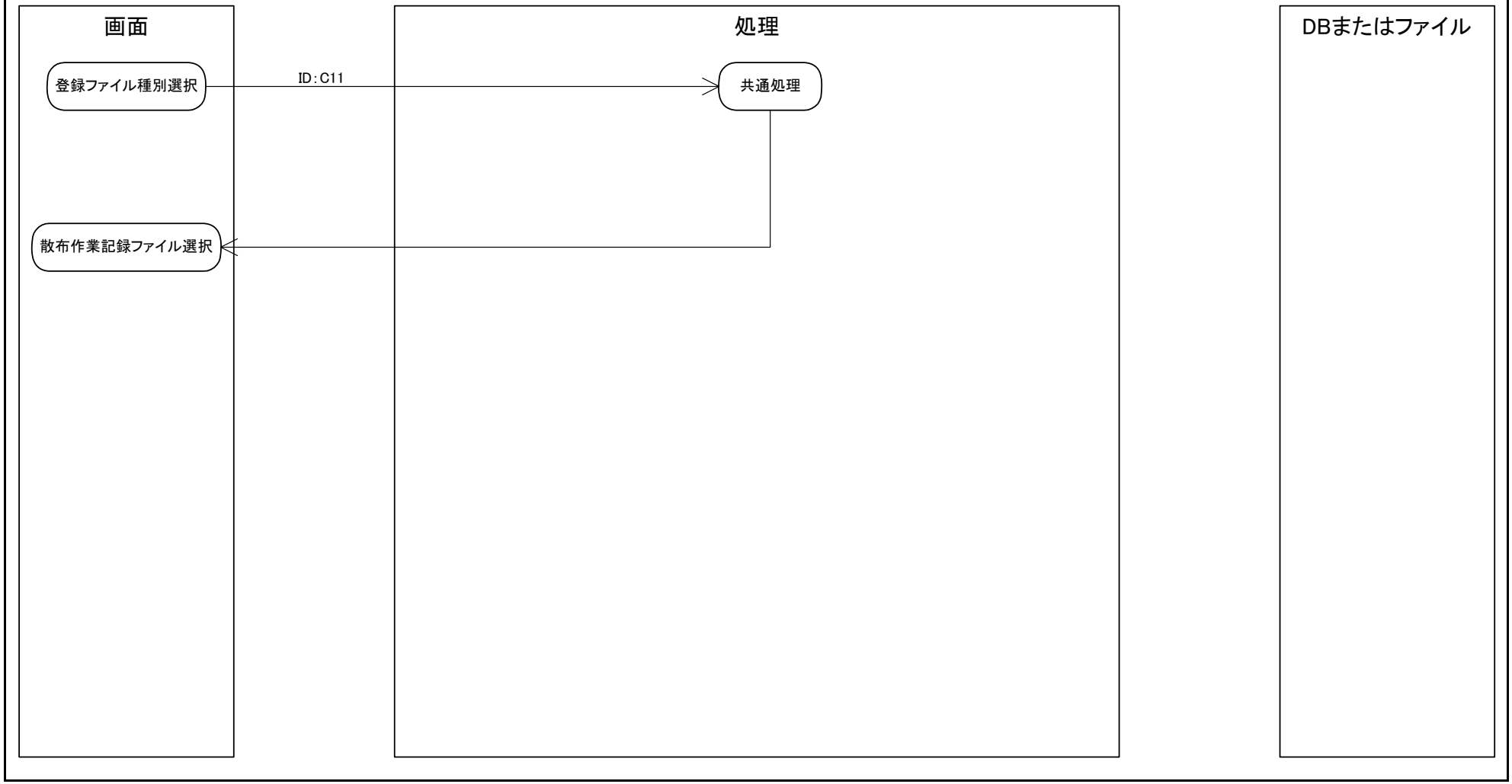


表 3-111　処理フロー

遷移元	散布作業記録ファイル選択	アクション	ボタン：送信	遷移先	散布作業記録ファイル登録完了

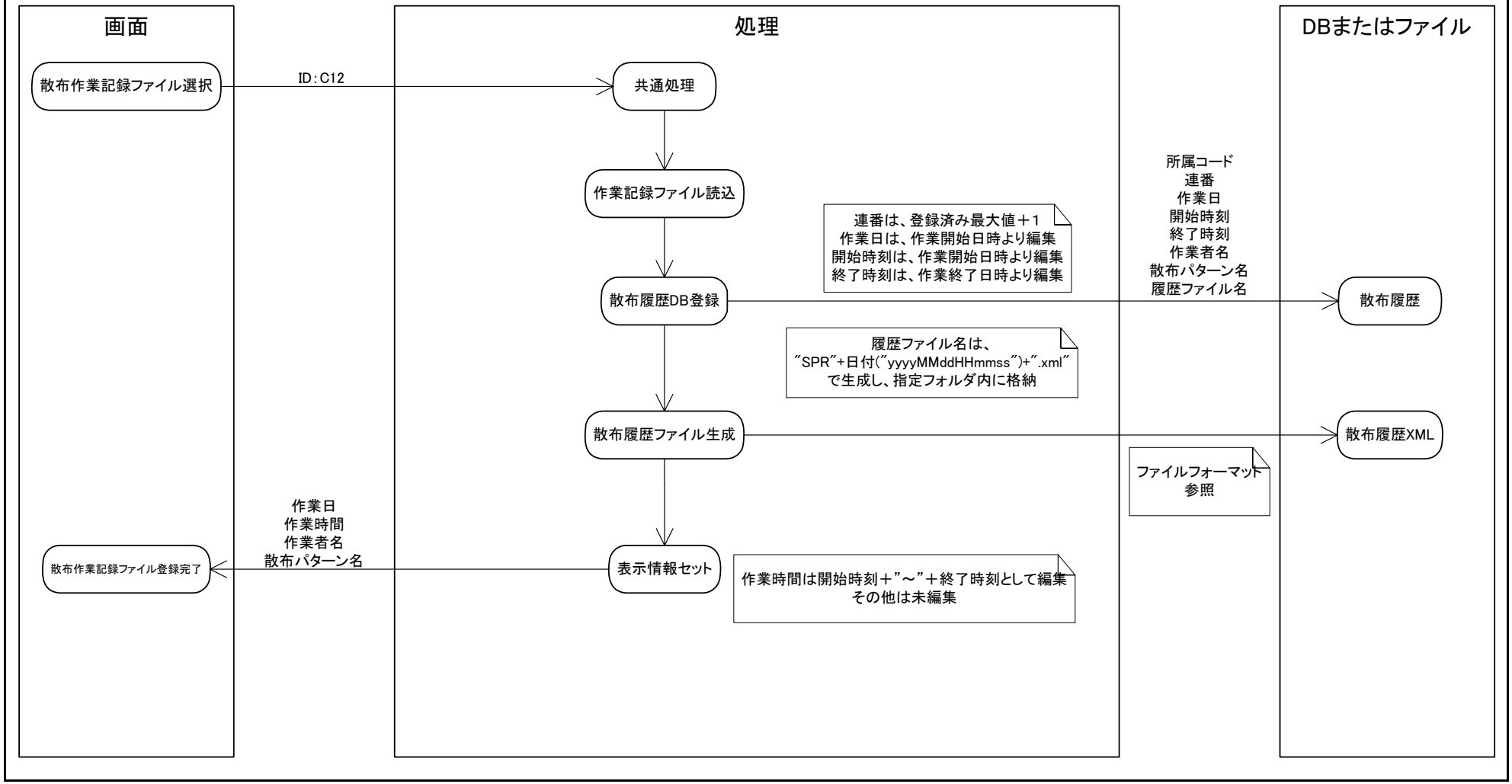


表 3-112　処理フロー

遷移元	散布作業記録ファイル選択	アクション	ボタン：戻る	遷移先	登録ファイル種別選択

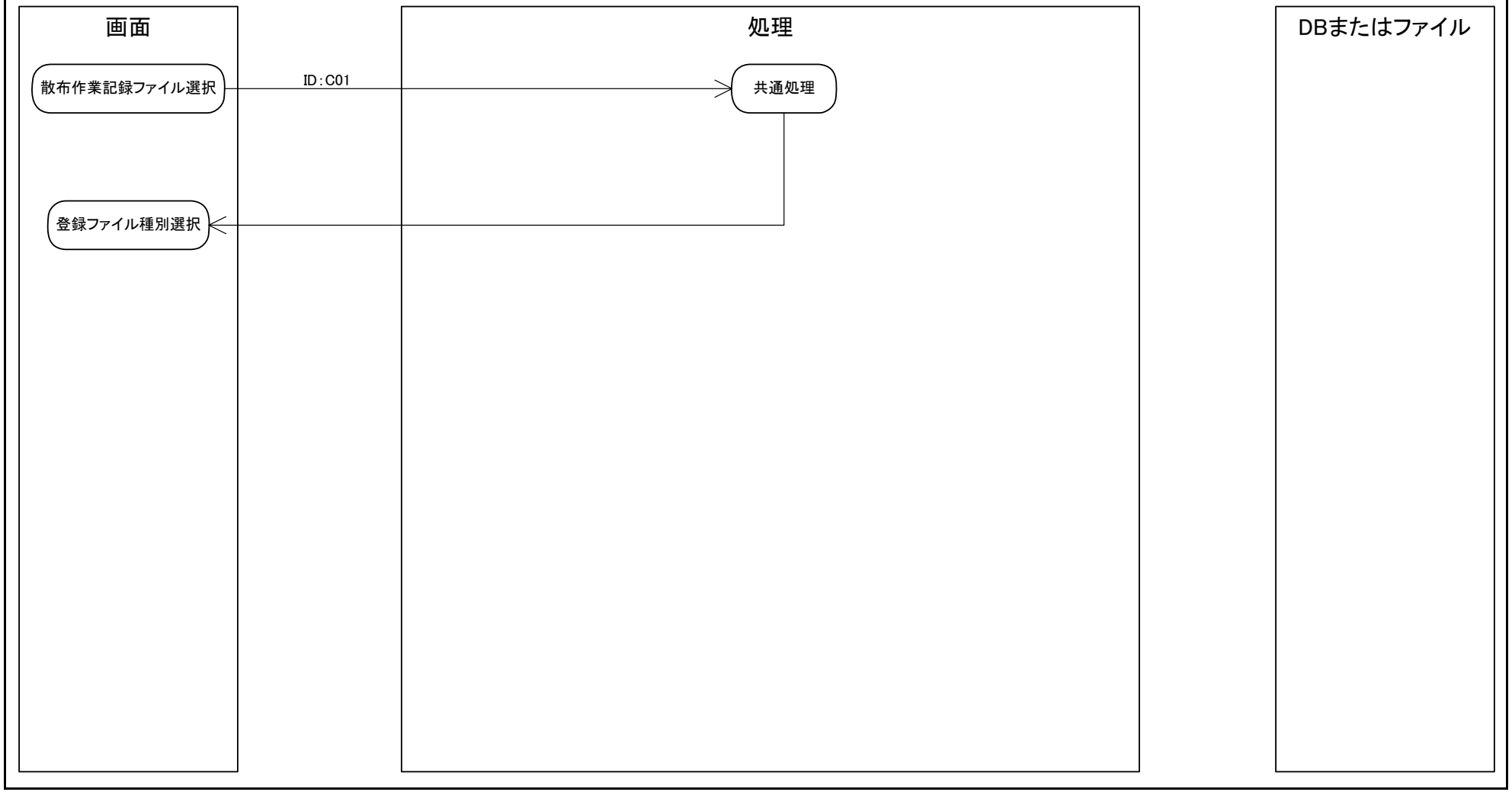


表 3-113　処理フロー

遷移元	散布作業記録ファイル登録完了	アクション	ボタン：メニューへ	遷移先	トップメニュー

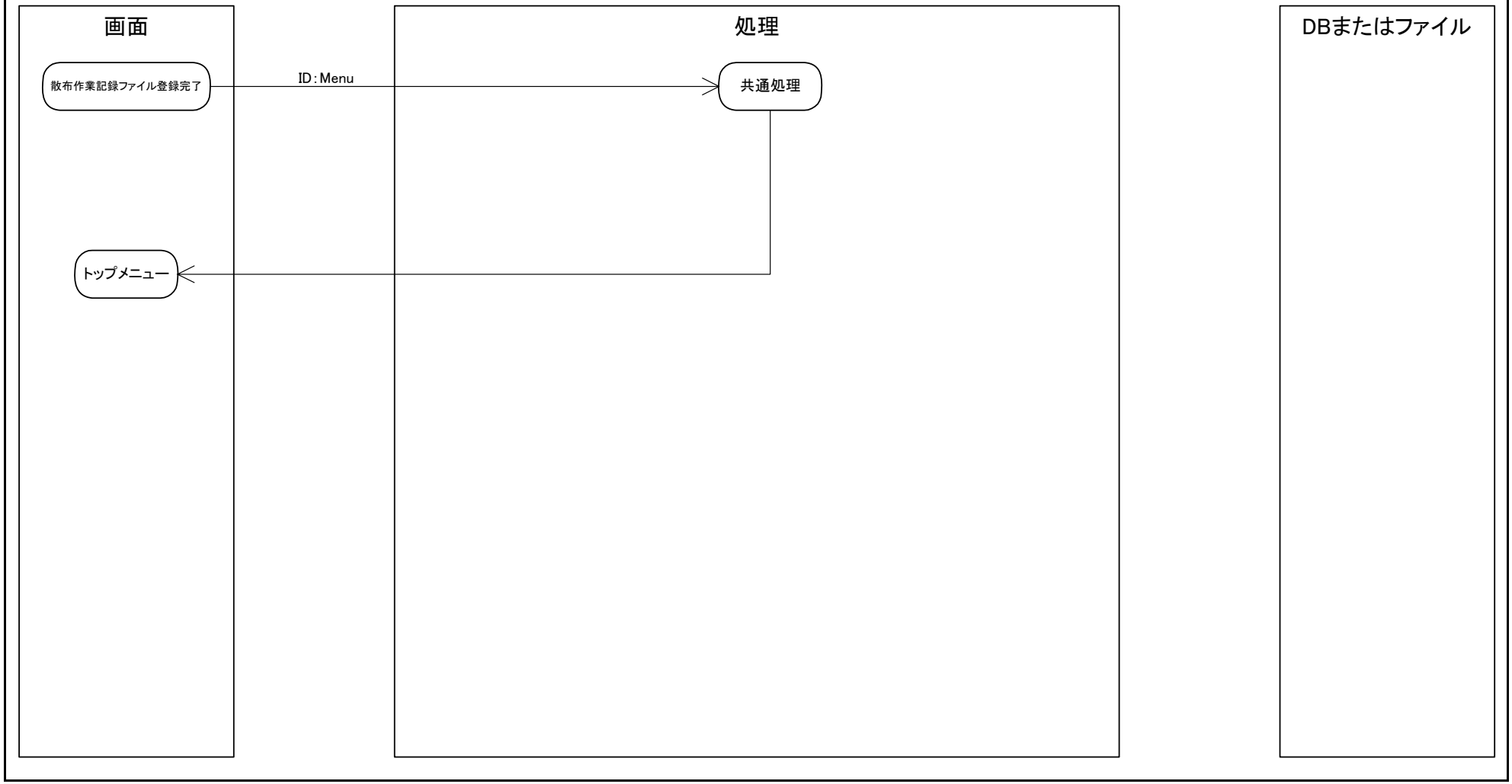


表 3-114　処理フロー

遷移元	登録ファイル種別選択	アクション	リンク：巡回記録の登録	遷移先	巡回記録ファイル選択

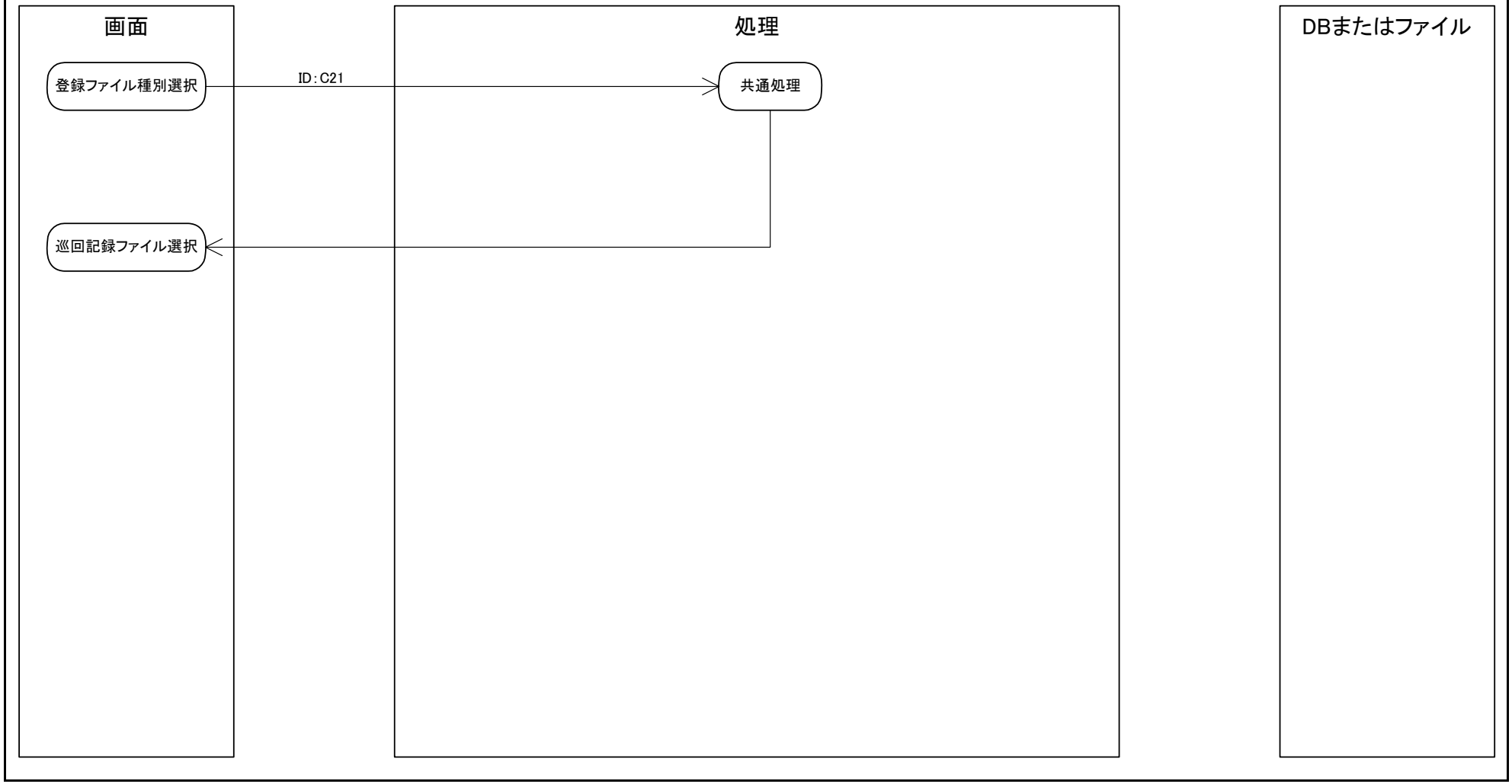


表 3-115　処理フロー

遷移元	巡回記録ファイル選択	アクション	ボタン：送信	遷移先	巡回記録ファイル登録完了

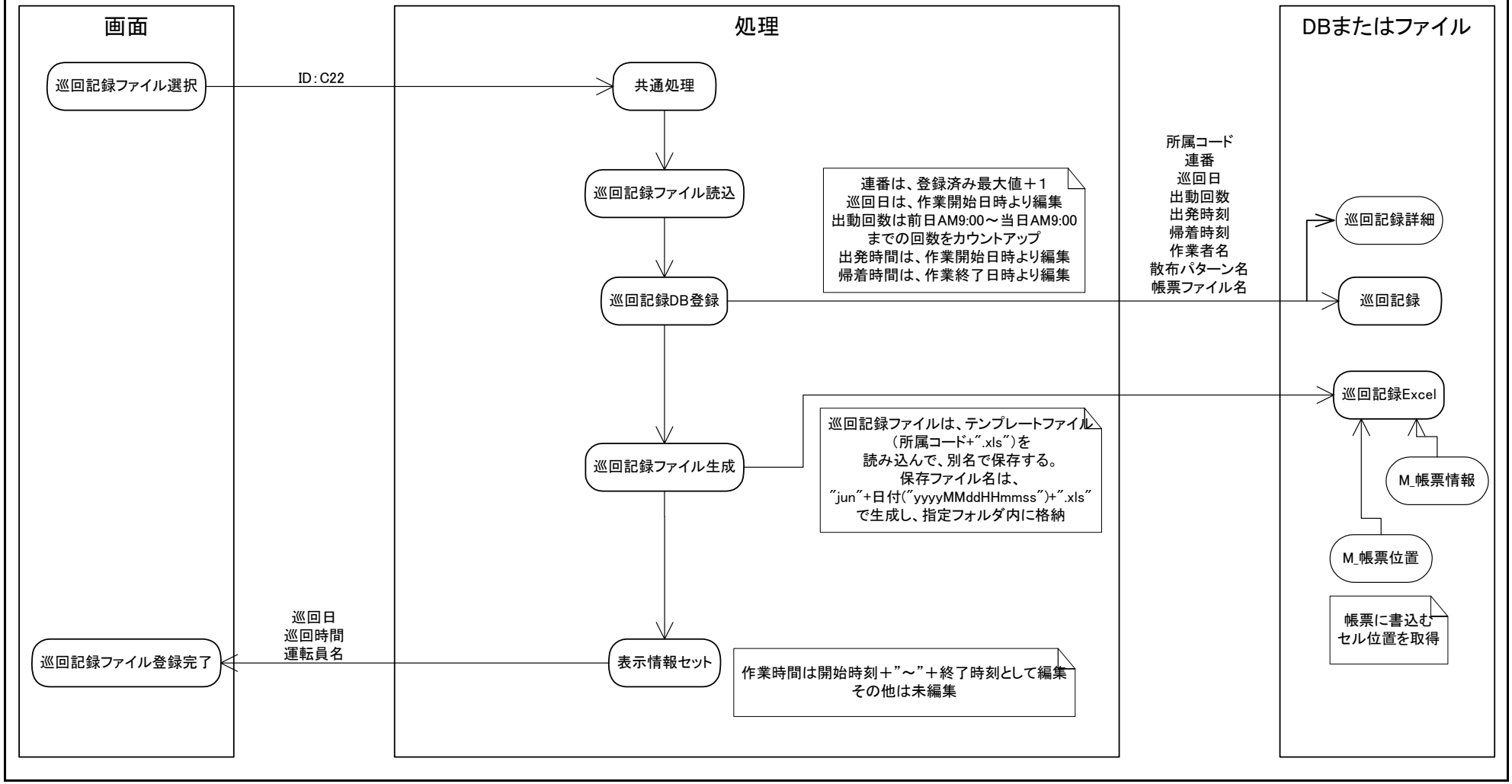


表 3-116　処理フロー

遷移元	巡回記録ファイル選択	アクション	ボタン：戻る	遷移先	登録ファイル種別選択

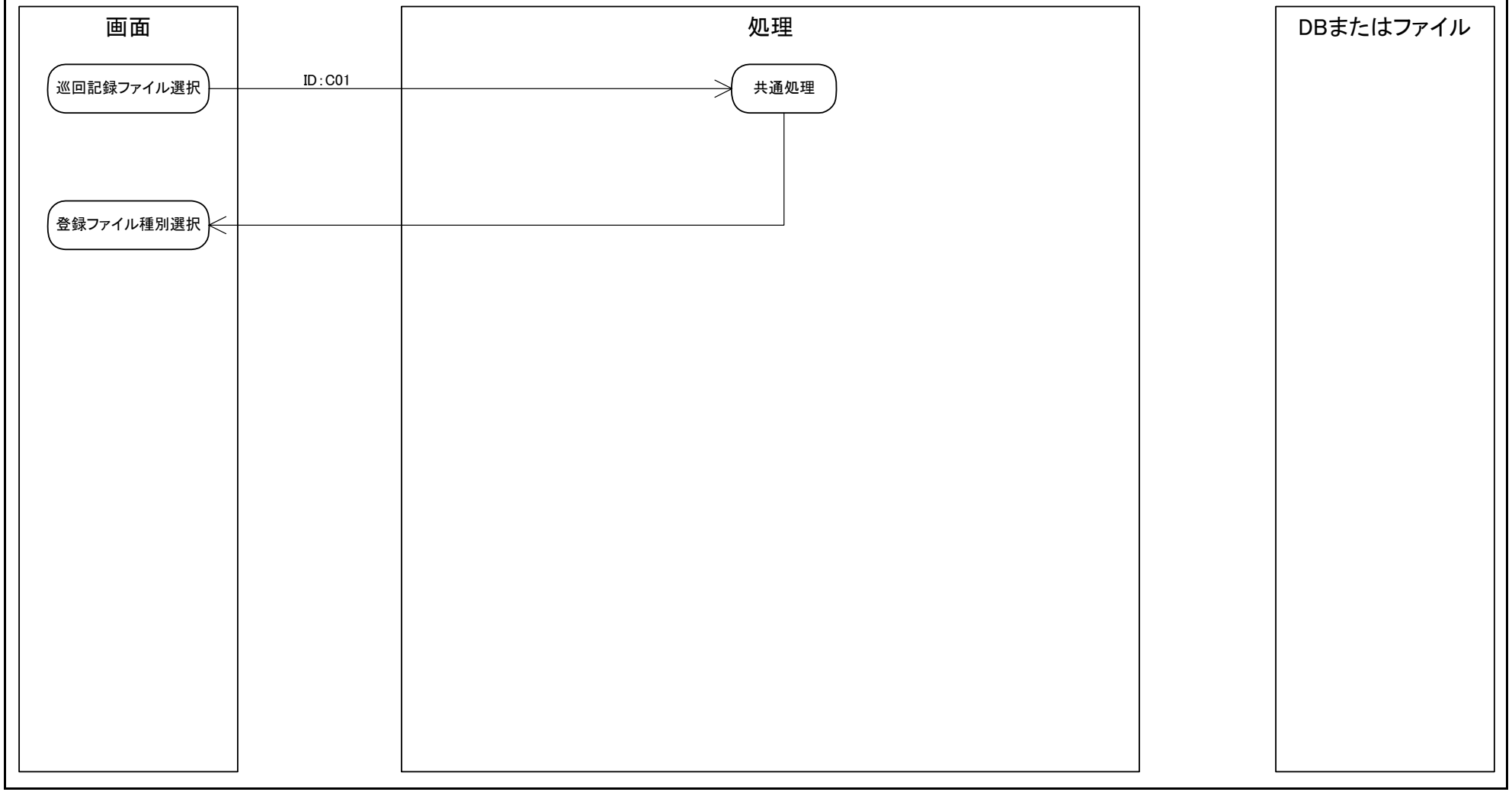


表 3-117　処理フロー

遷移元	巡回記録ファイル登録完了	アクション	ボタン：メニューへ	遷移先	トップメニュー

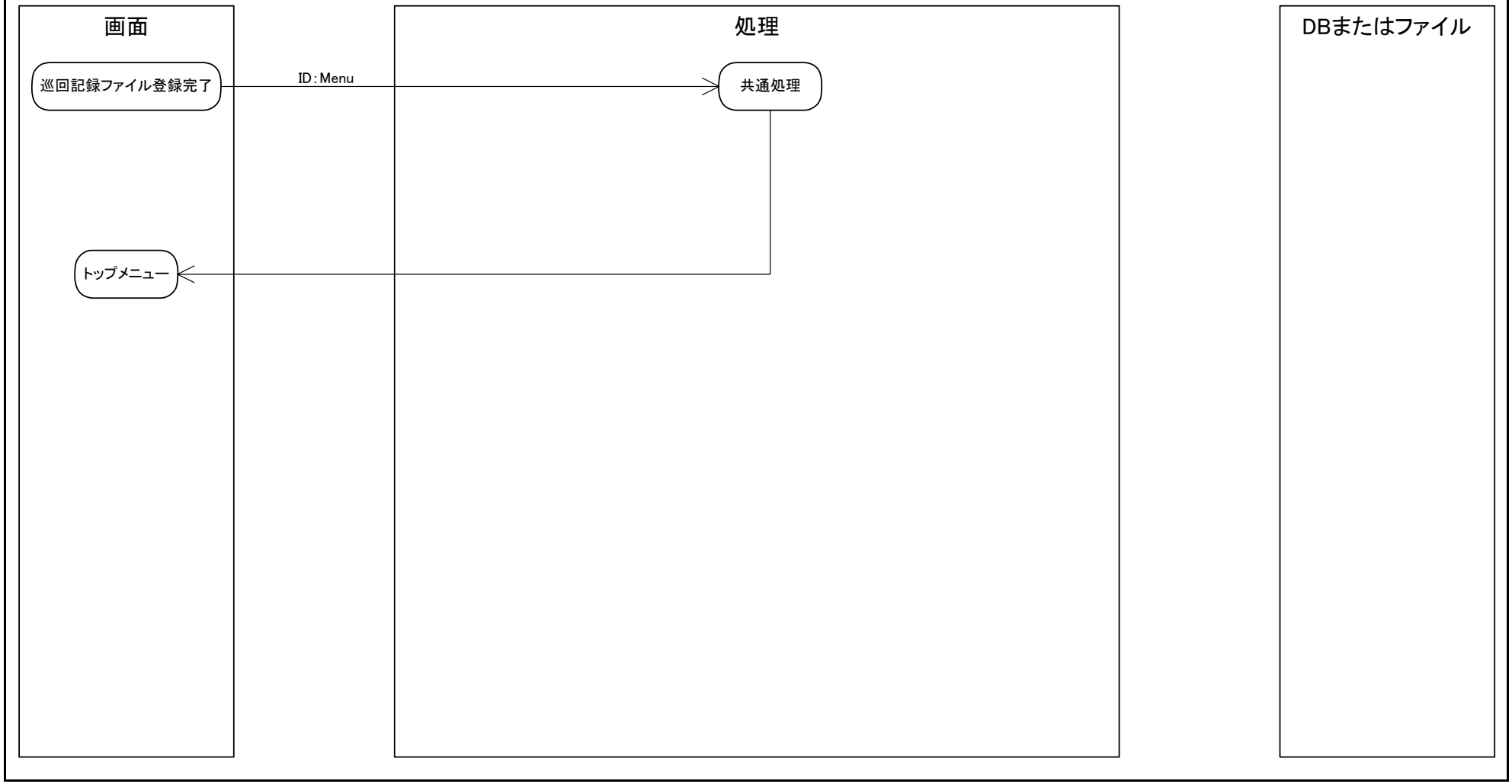


表 3-118　処理フロー

遷移元	トップメニュー	アクション	リンク：標準散布パターン作成	遷移先	標準散布パターン作成メニュー

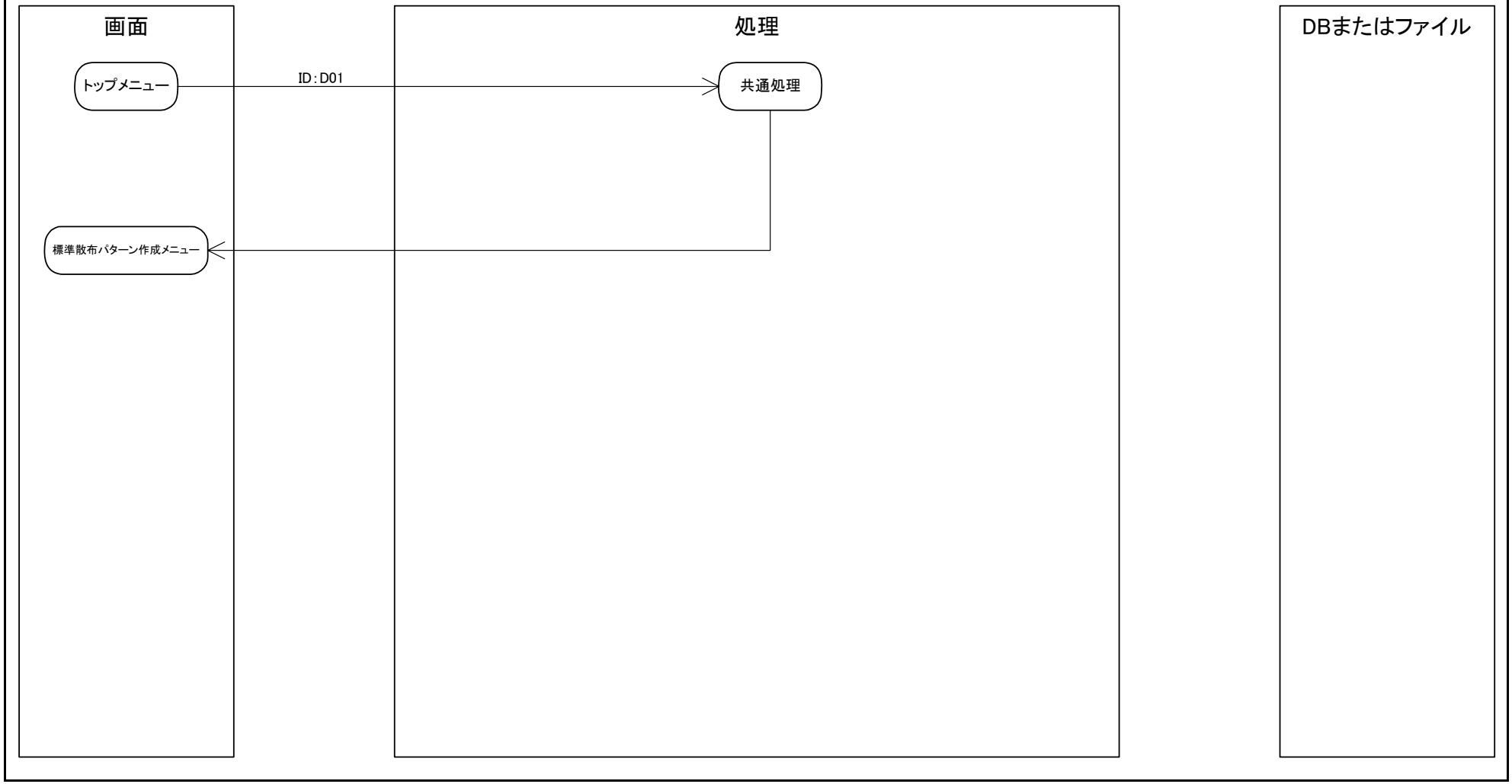


表 3-119　処理フロー

遷移元	標準散布パターン作成メニュー	アクション	ボタン：戻る	遷移先	トップメニュー

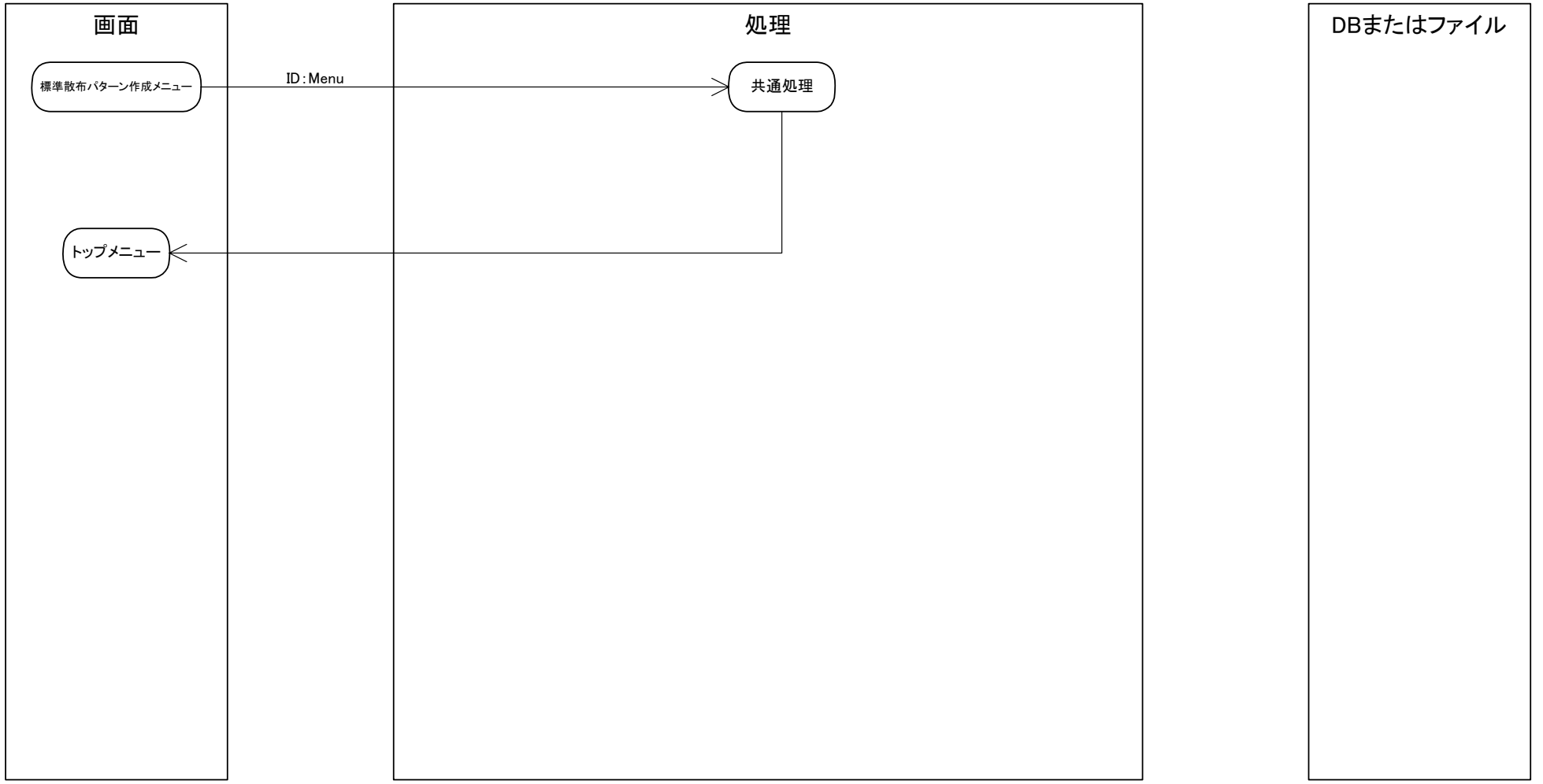


表 3-120　処理フロー

遷移元	標準散布パターン作成メニュー	アクション	リンク：管理メッシュ作成	遷移先	管理メッシュ一覧

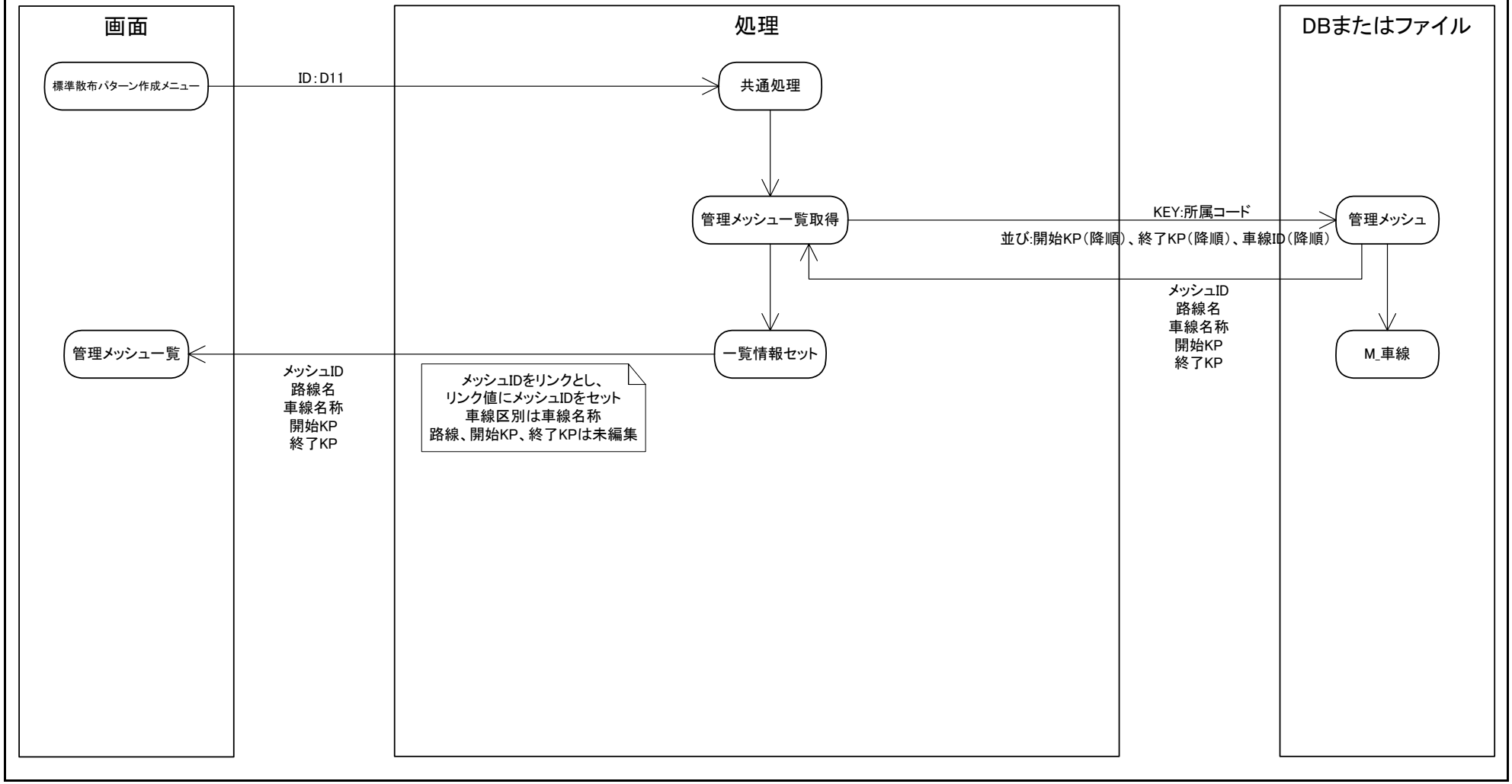


表 3-121　処理フロー

遷移元	管理メッシュ一覧	アクション	リンク：メッシュ ID	遷移先	管理メッシュ修正

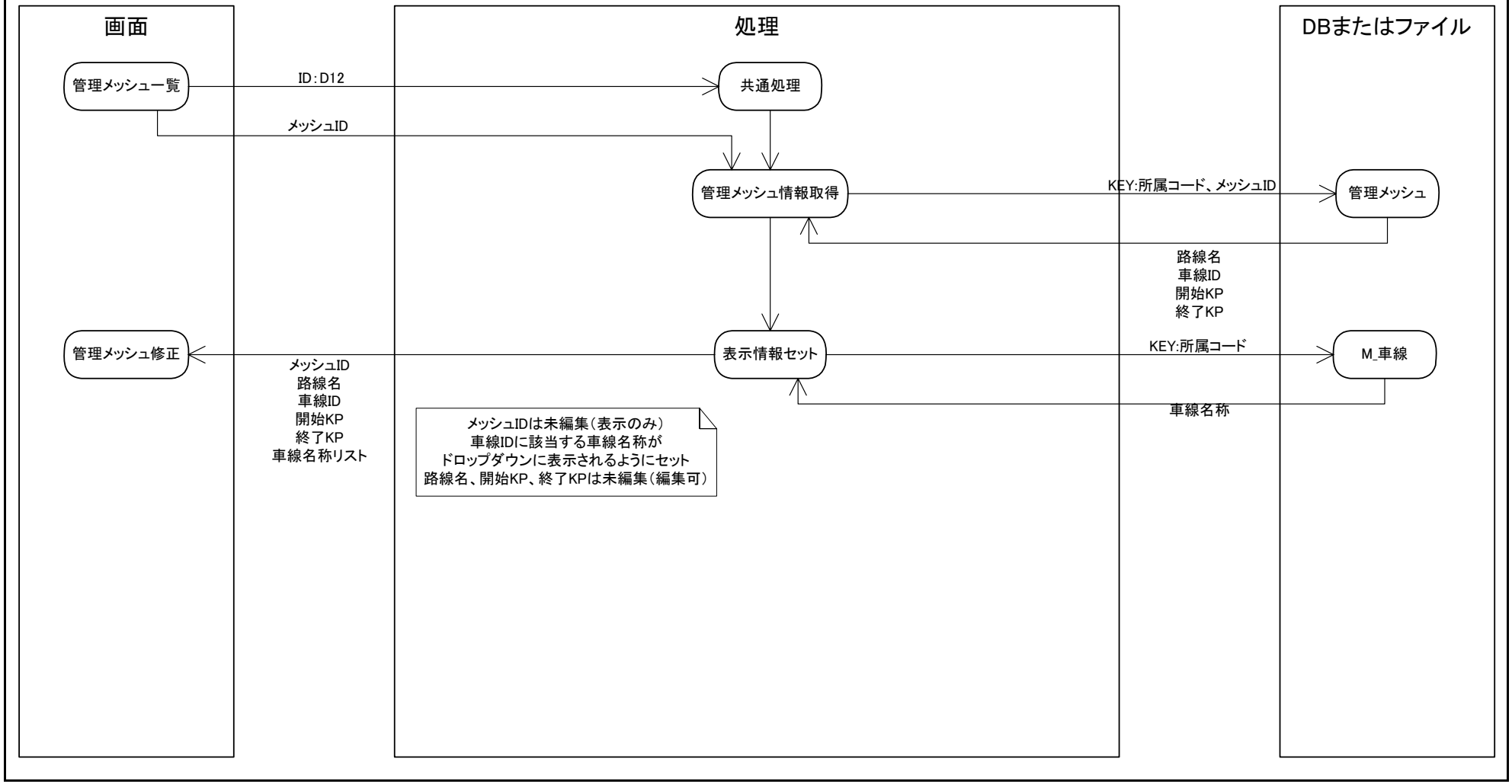


表 3-122　処理フロー

遷移元	管理メッシュ一覧	アクション	ボタン：新規	遷移先	管理メッシュ新規登録

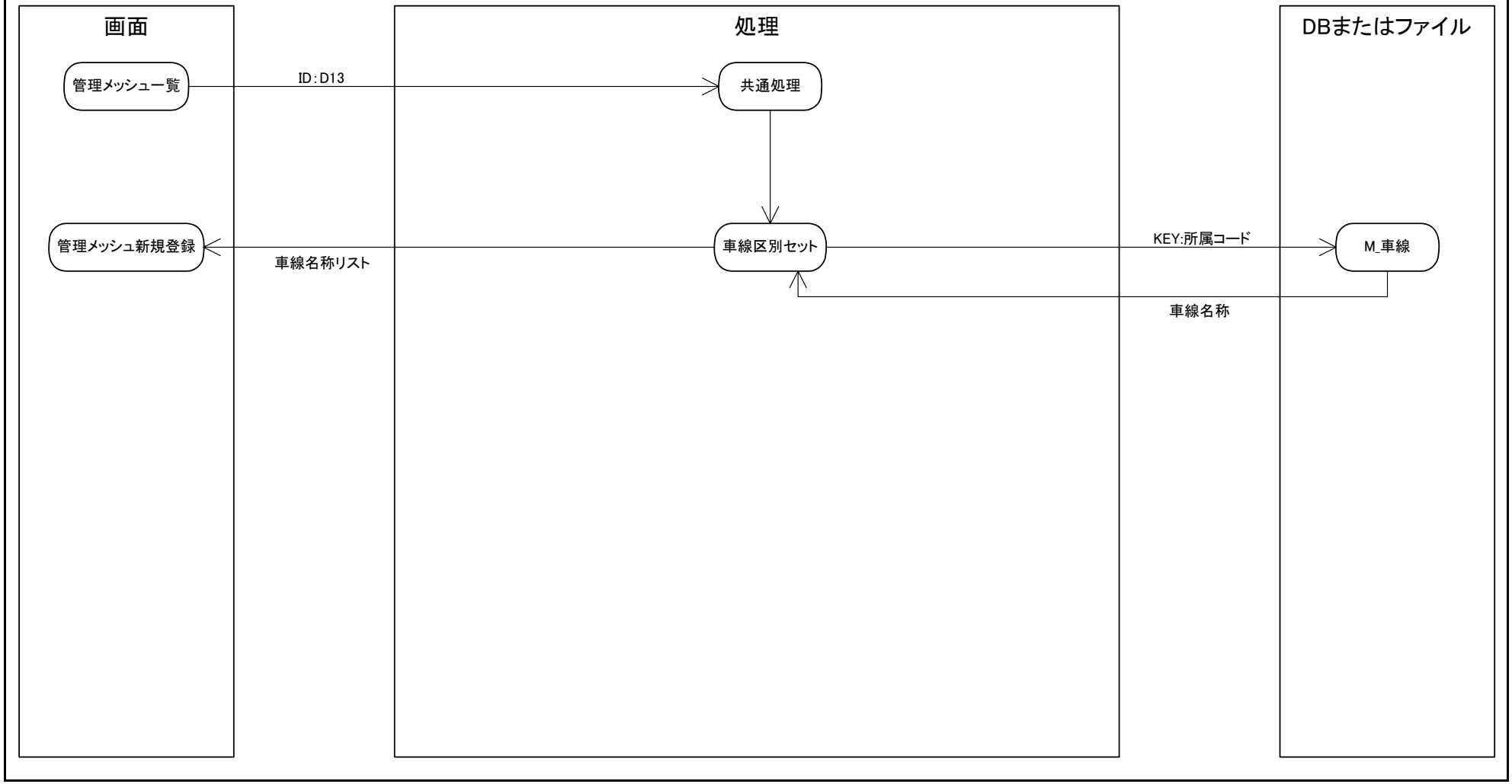


表 3-123　処理フロー

遷移元	管理メッシュ一覧	アクション	ボタン：戻る	遷移先	標準散布パターン作成メニュー

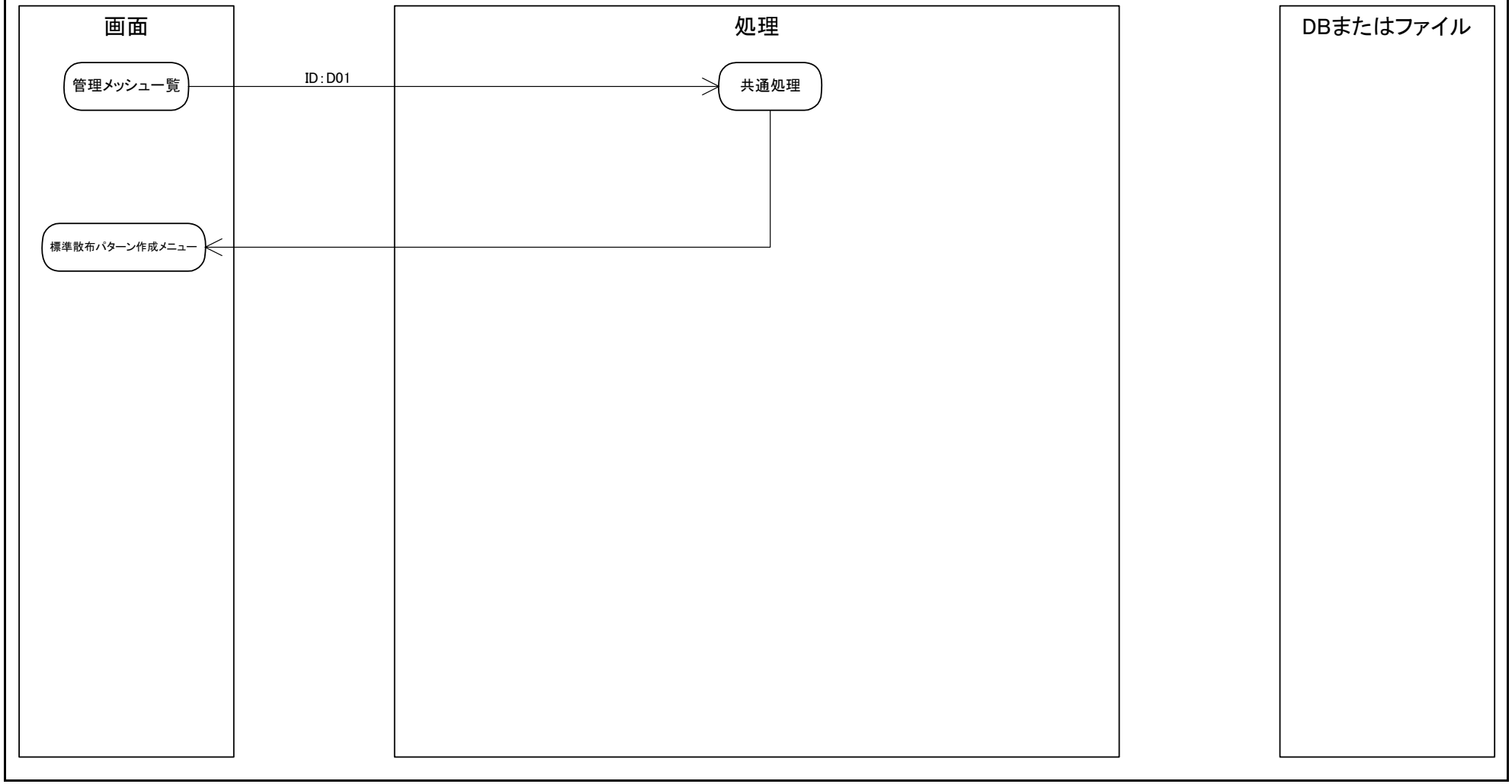


表 3-124　処理フロー

遷移元	管理メッシュ一覧	アクション	ボタン：メニューへ	遷移先	トップメニュー

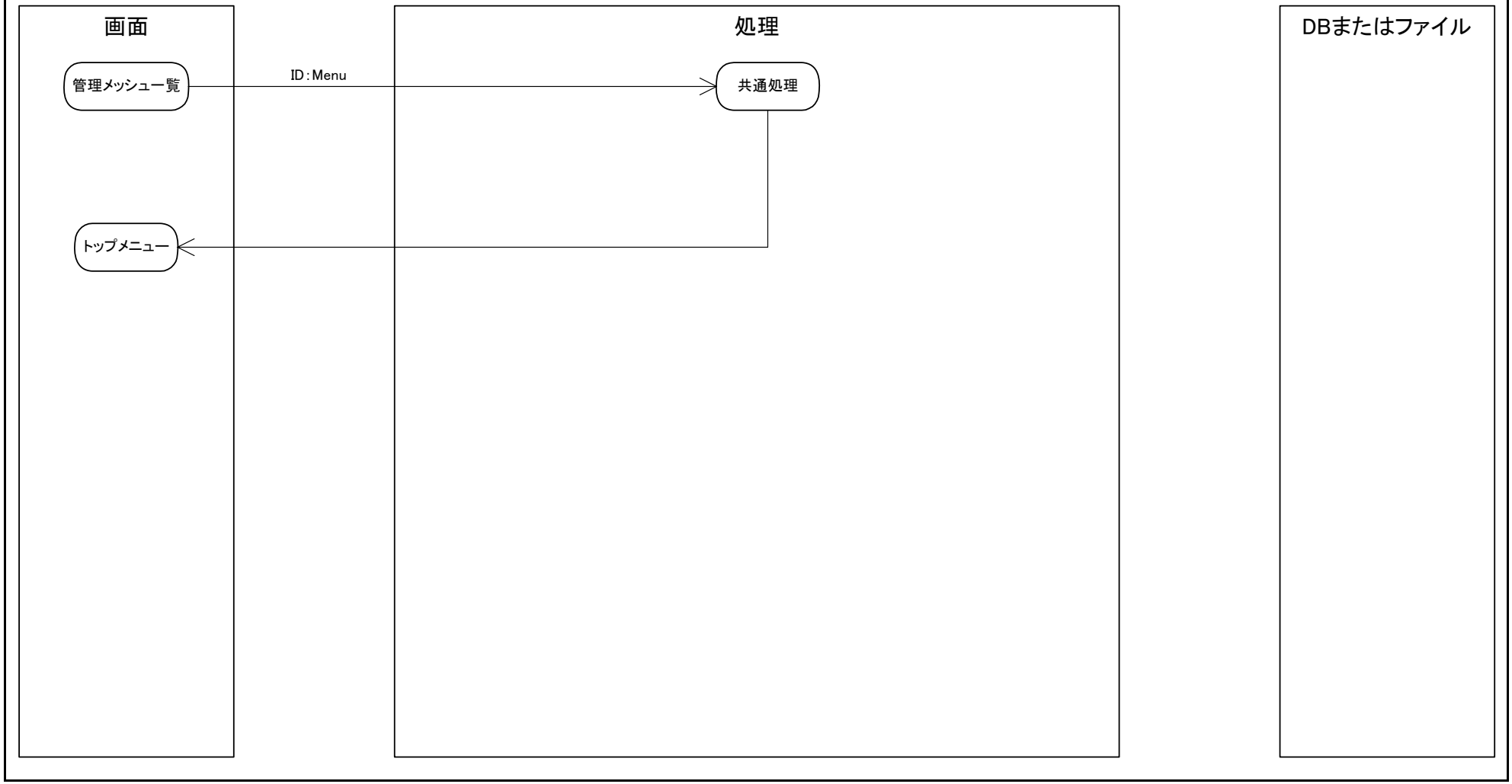


表 3-125　処理フロー

遷移元	管理メッシュ修正	アクション	ボタン：修正	遷移先	管理メッシュ一覧

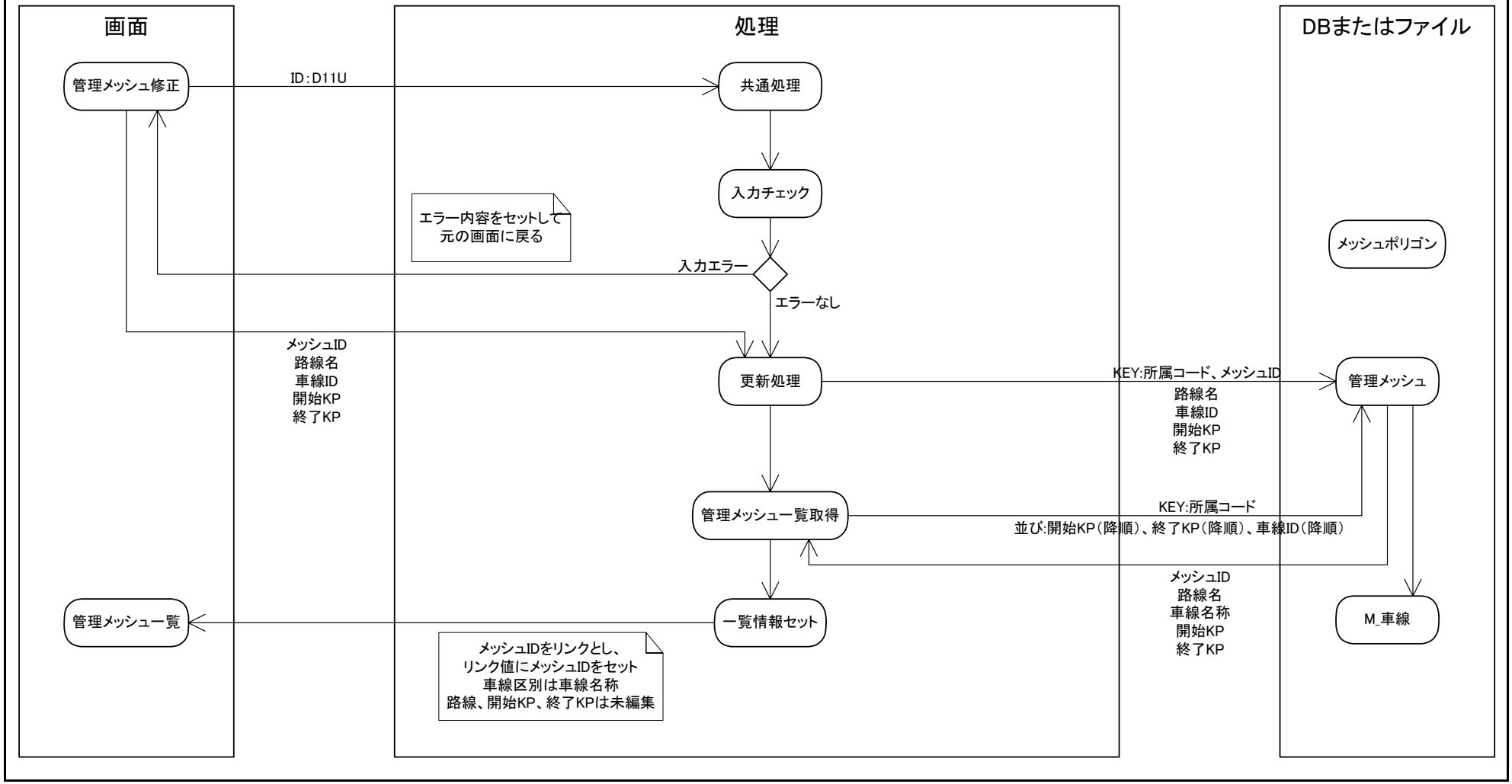


表 3-126　処理フロー

遷移元	管理メッシュ修正	アクション	ボタン：複製	遷移先	管理メッシュ登録

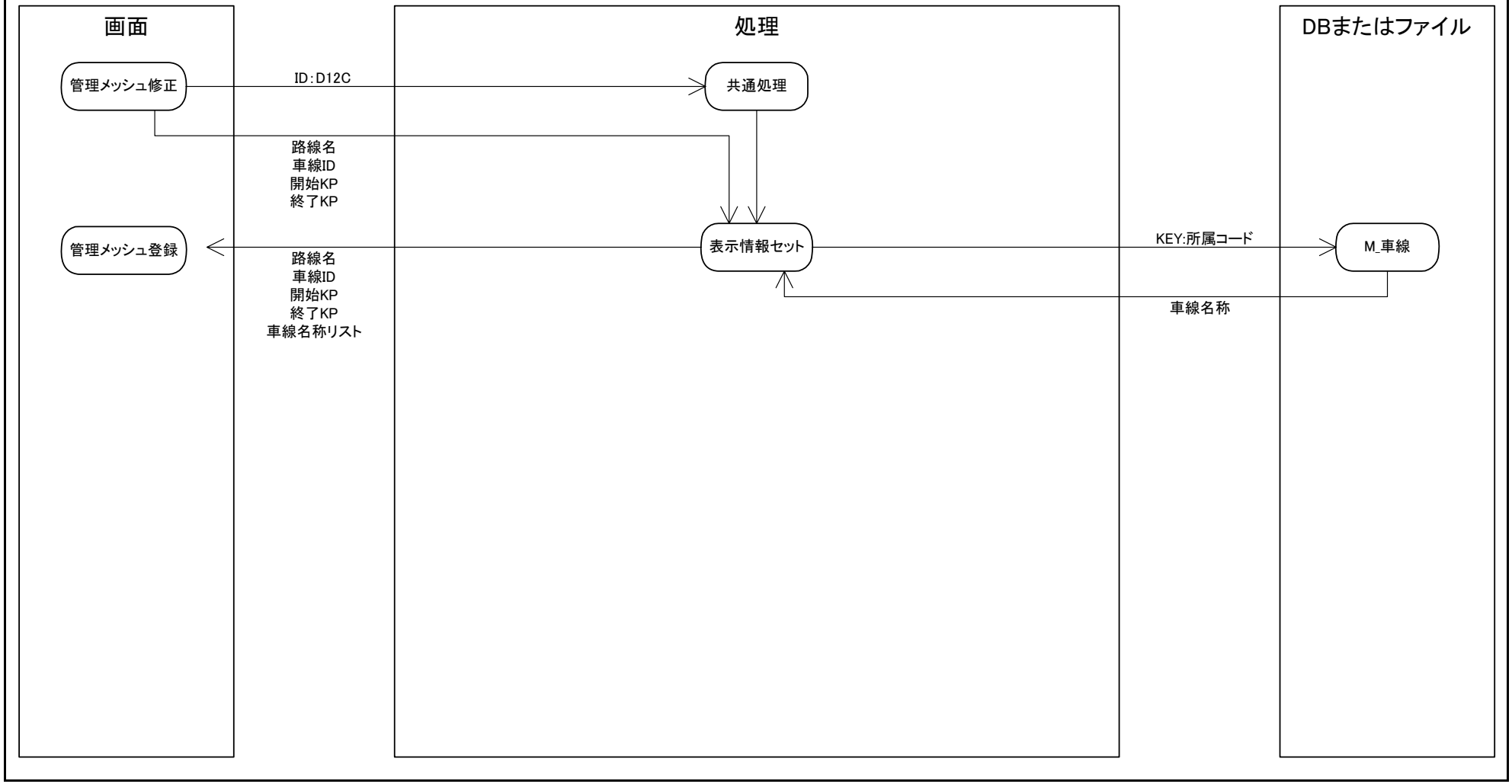


表 3-127　処理フロー

遷移元	管理メッシュ修正	アクション	ボタン：削除	遷移先	管理メッシュ一覧

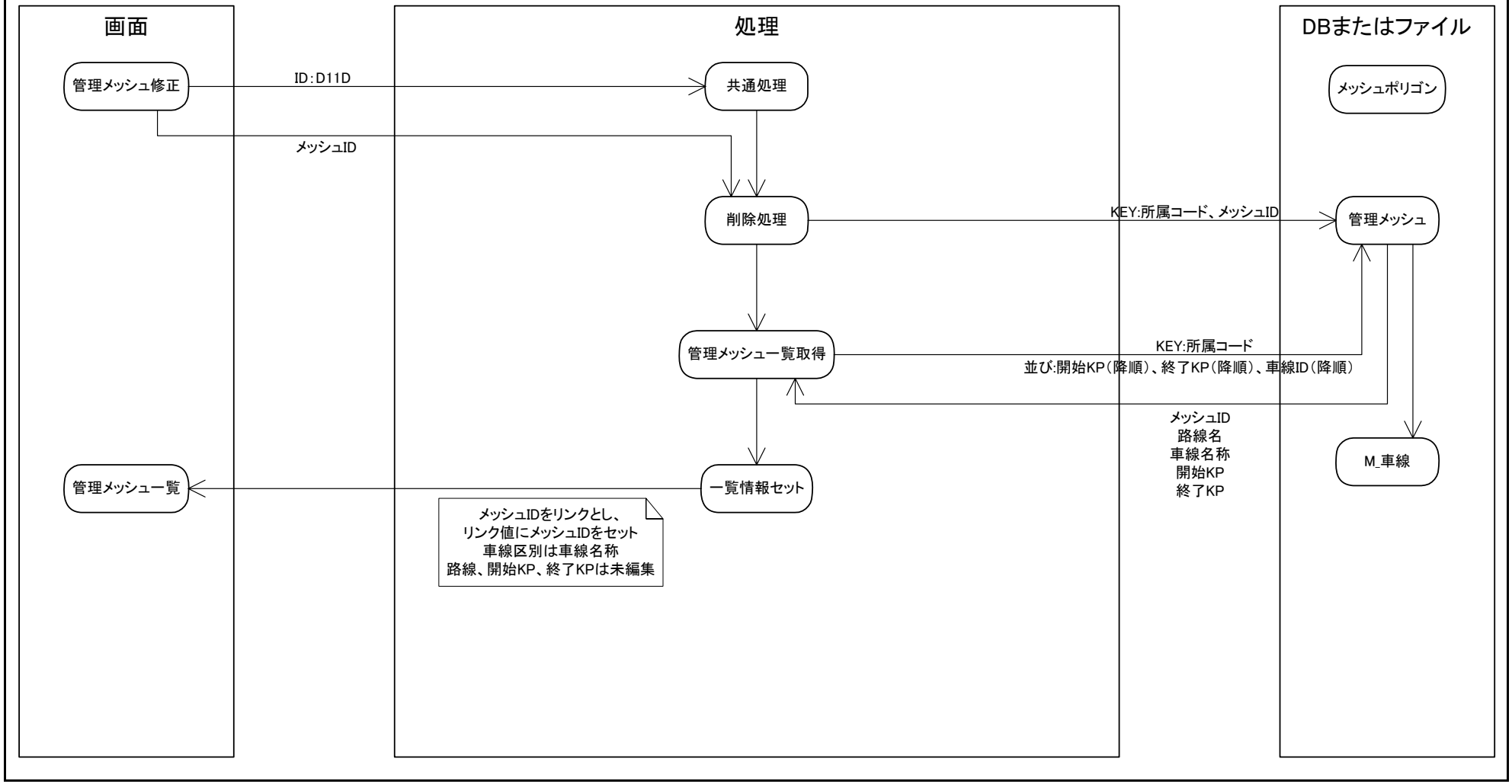


表 3-128　処理フロー

遷移元	管理メッシュ修正	アクション	ボタン：戻る	遷移先	管理メッシュ一覧

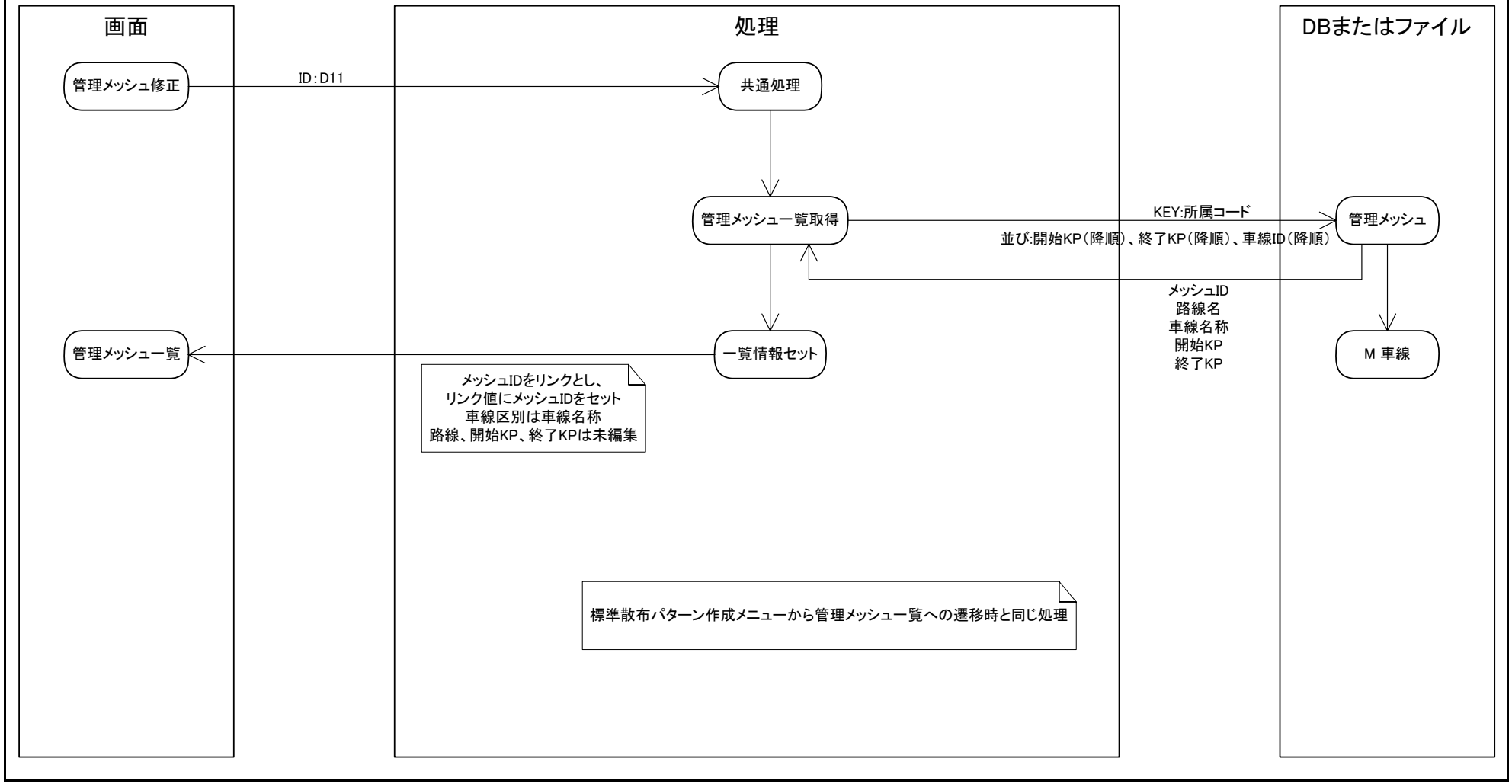


表 3-129　処理フロー

遷移元	管理メッシュ登録	アクション	ボタン：登録	遷移先	管理メッシュ一覧

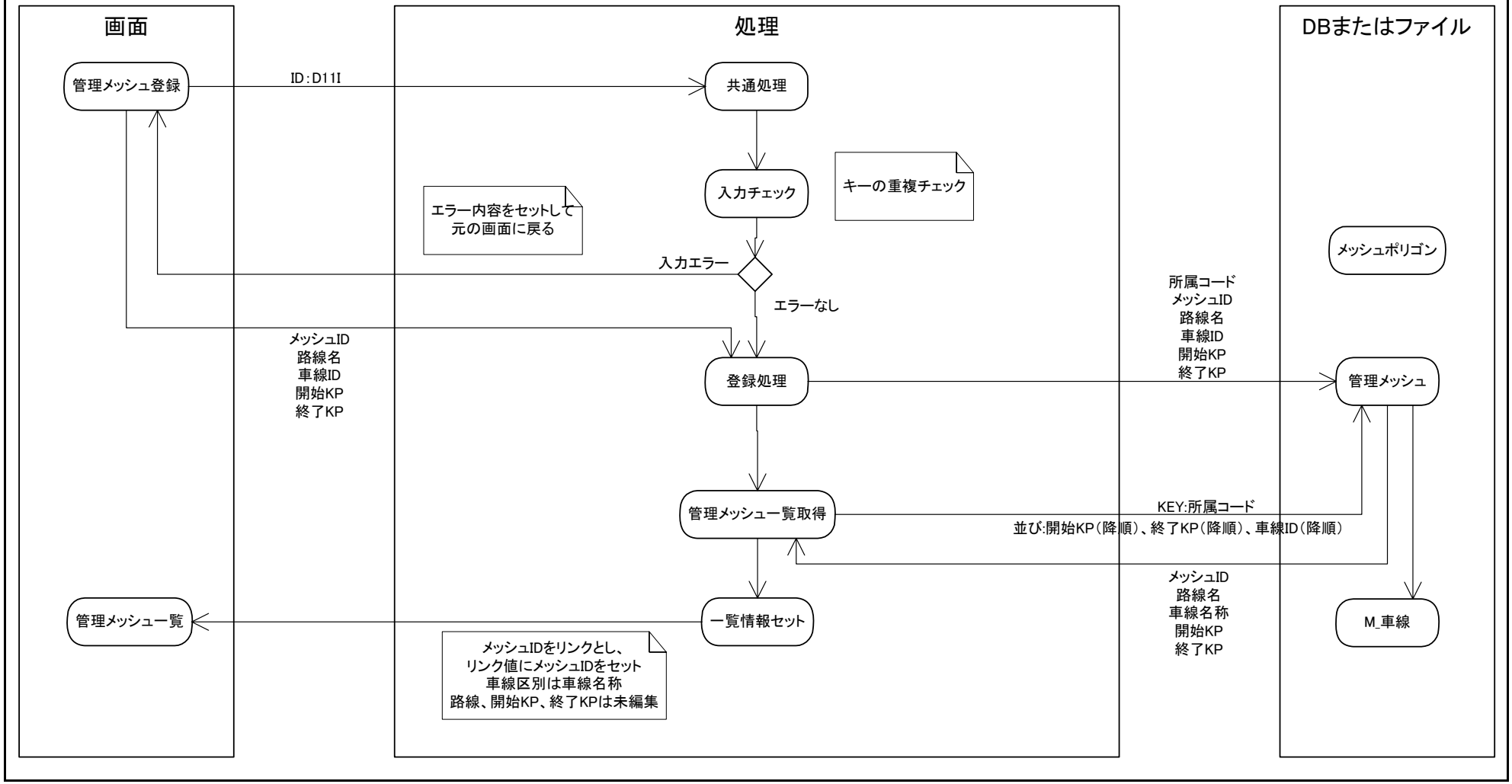


表 3-130　処理フロー

遷移元	管理メッシュ登録	アクション	ボタン：戻る	遷移先	管理メッシュ一覧

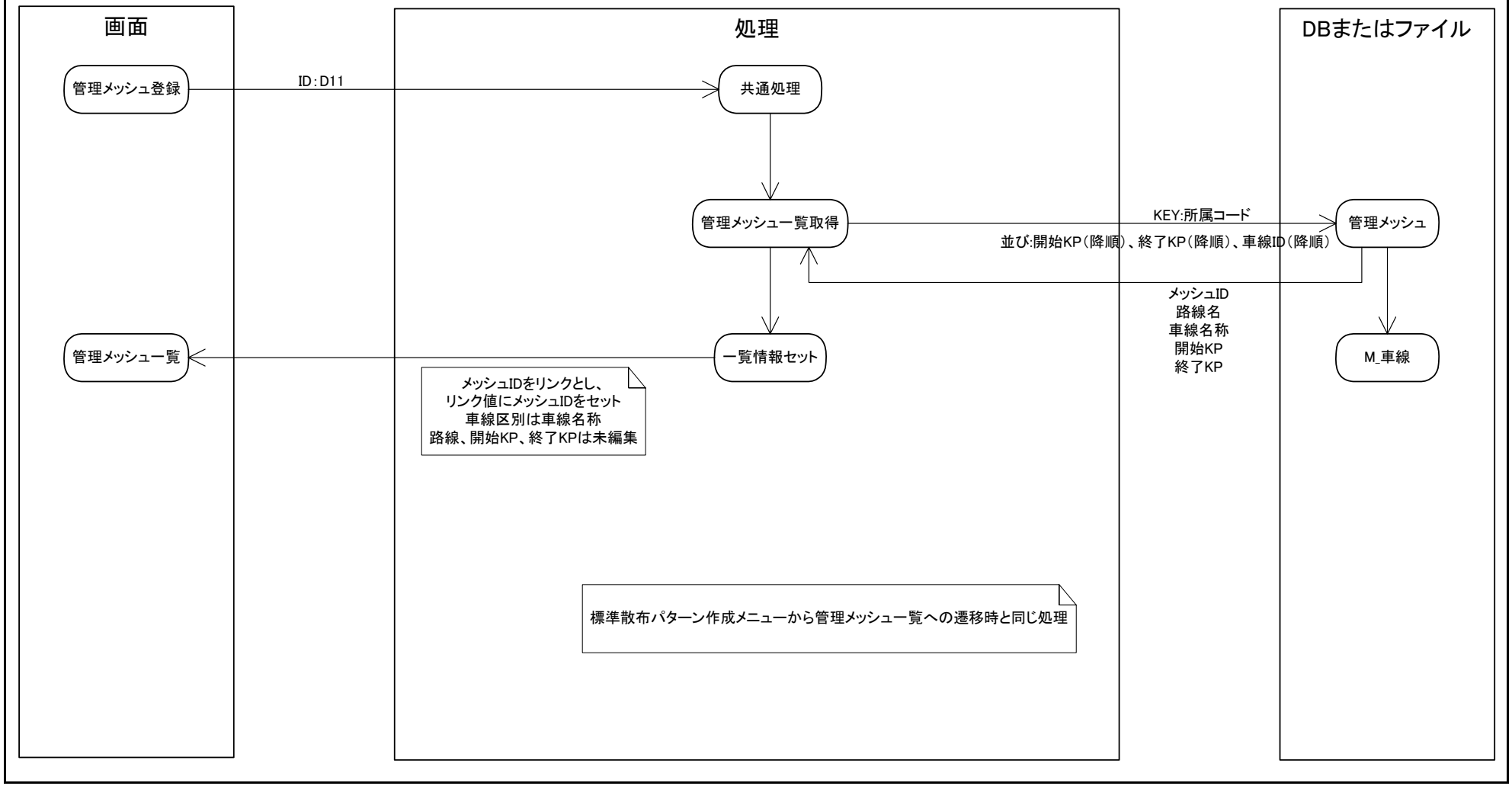


表 3-131　処理フロー

遷移元	標準散布パターン作成メニュー	アクション	リンク：散布基本データ作成	遷移先	散布基本データ一覧

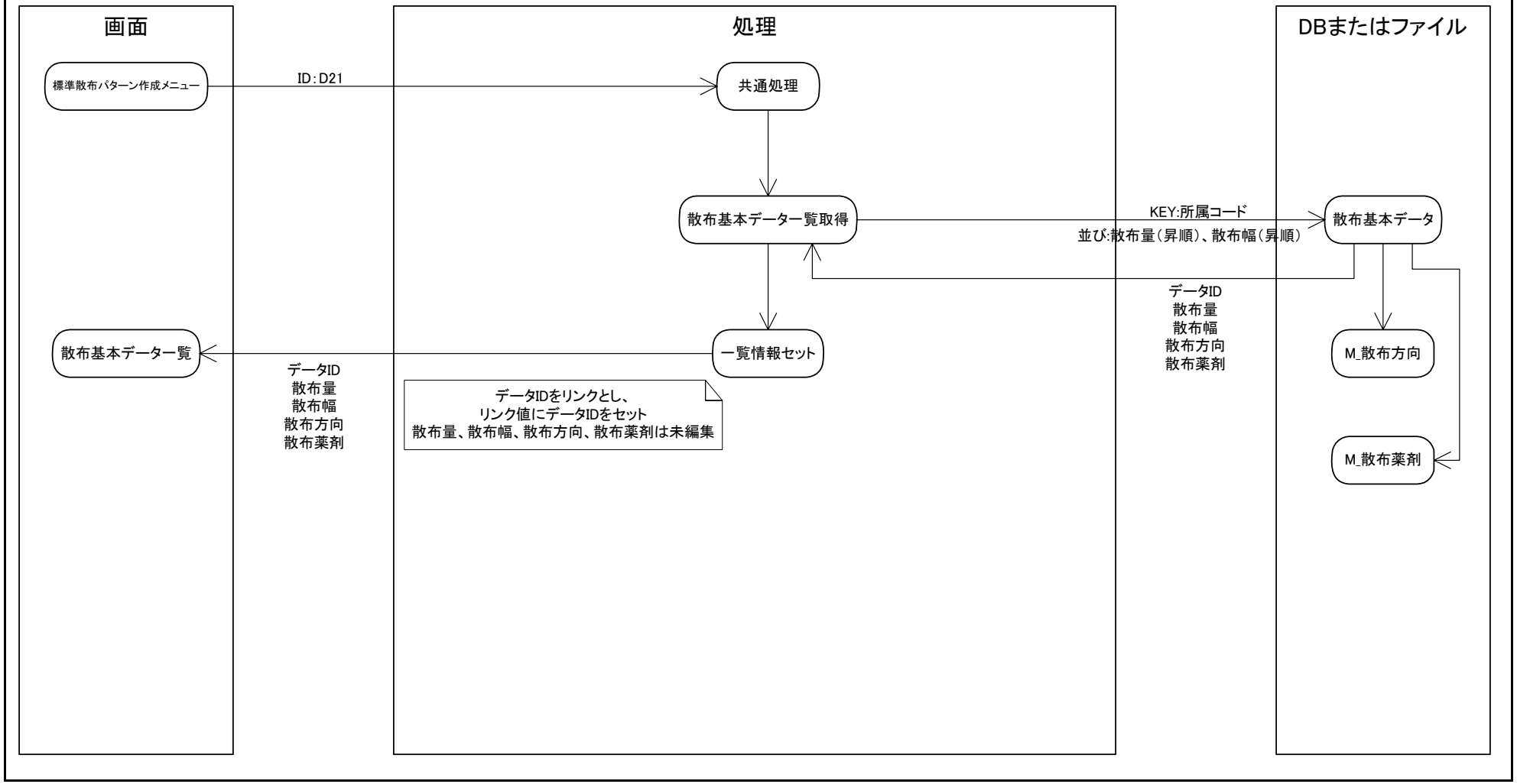


表 3-132　処理フロー

遷移元	散布基本データ	アクション	リンク：データ ID	遷移先	散布基本データ修正

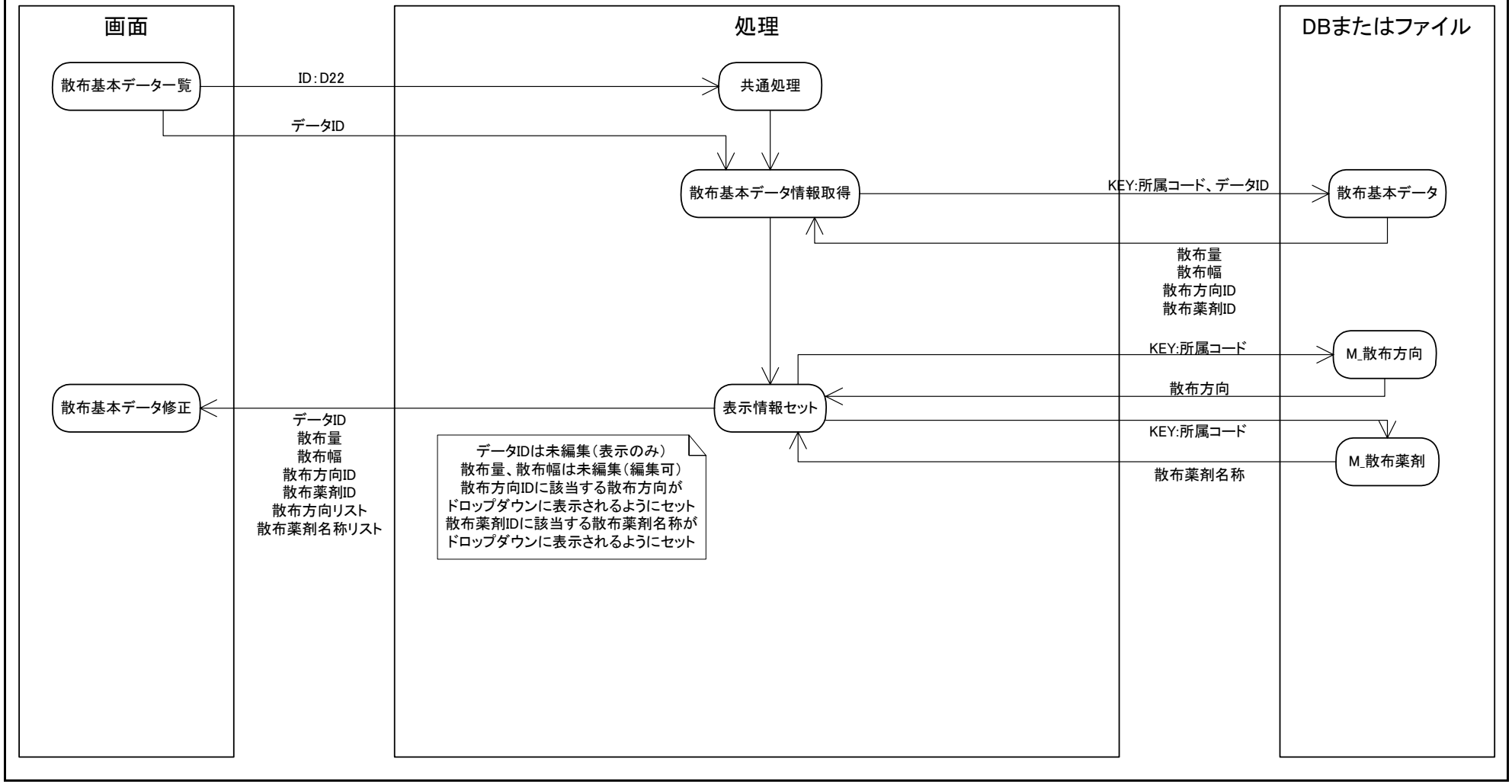


表 3-133　処理フロー

遷移元	散布基本データ一覧	アクション	ボタン：新規	遷移先	散布基本データ新規登録

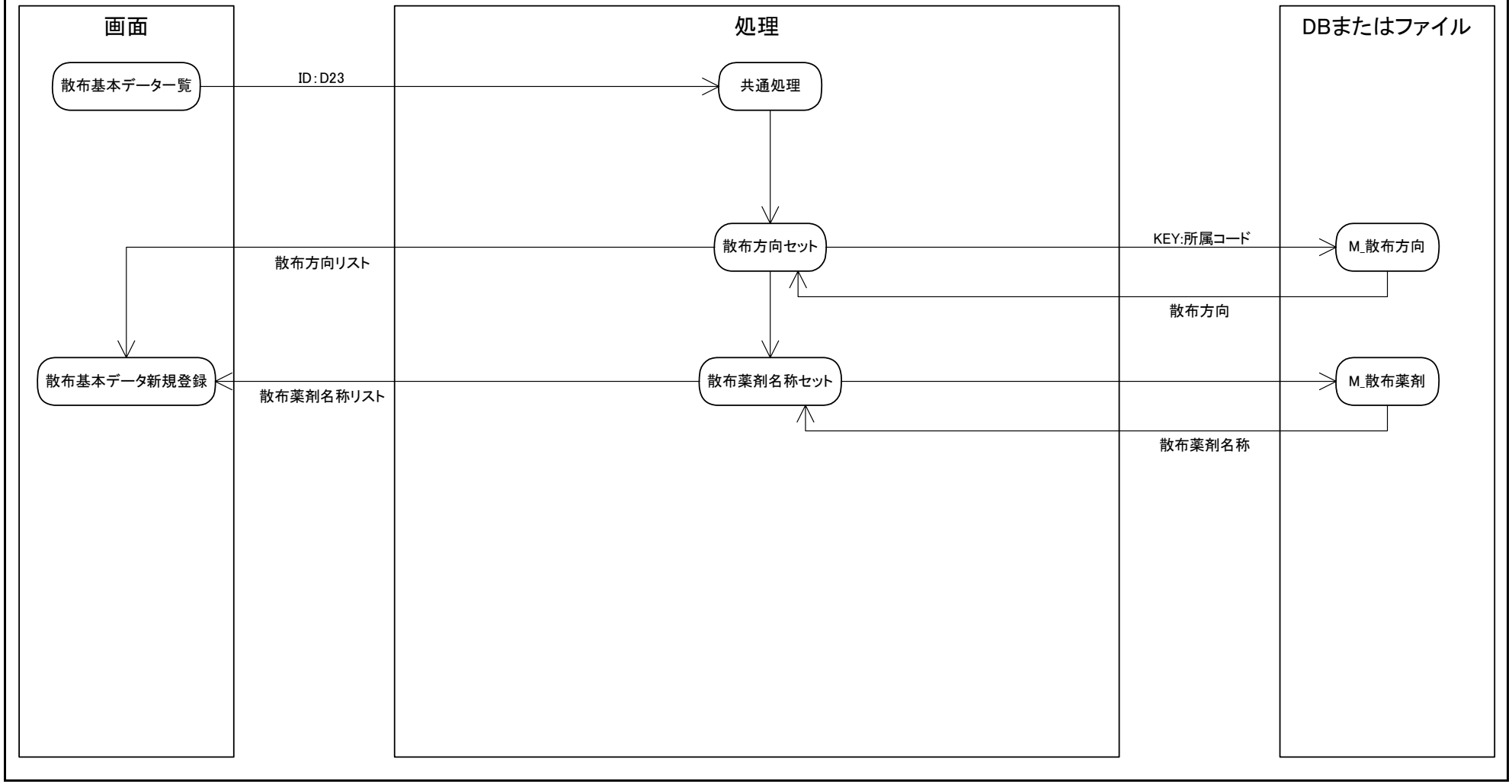


表 3-134　処理フロー

遷移元	散布基本データ一覧	アクション	ボタン：戻る	遷移先	標準散布パターン作成メニュー

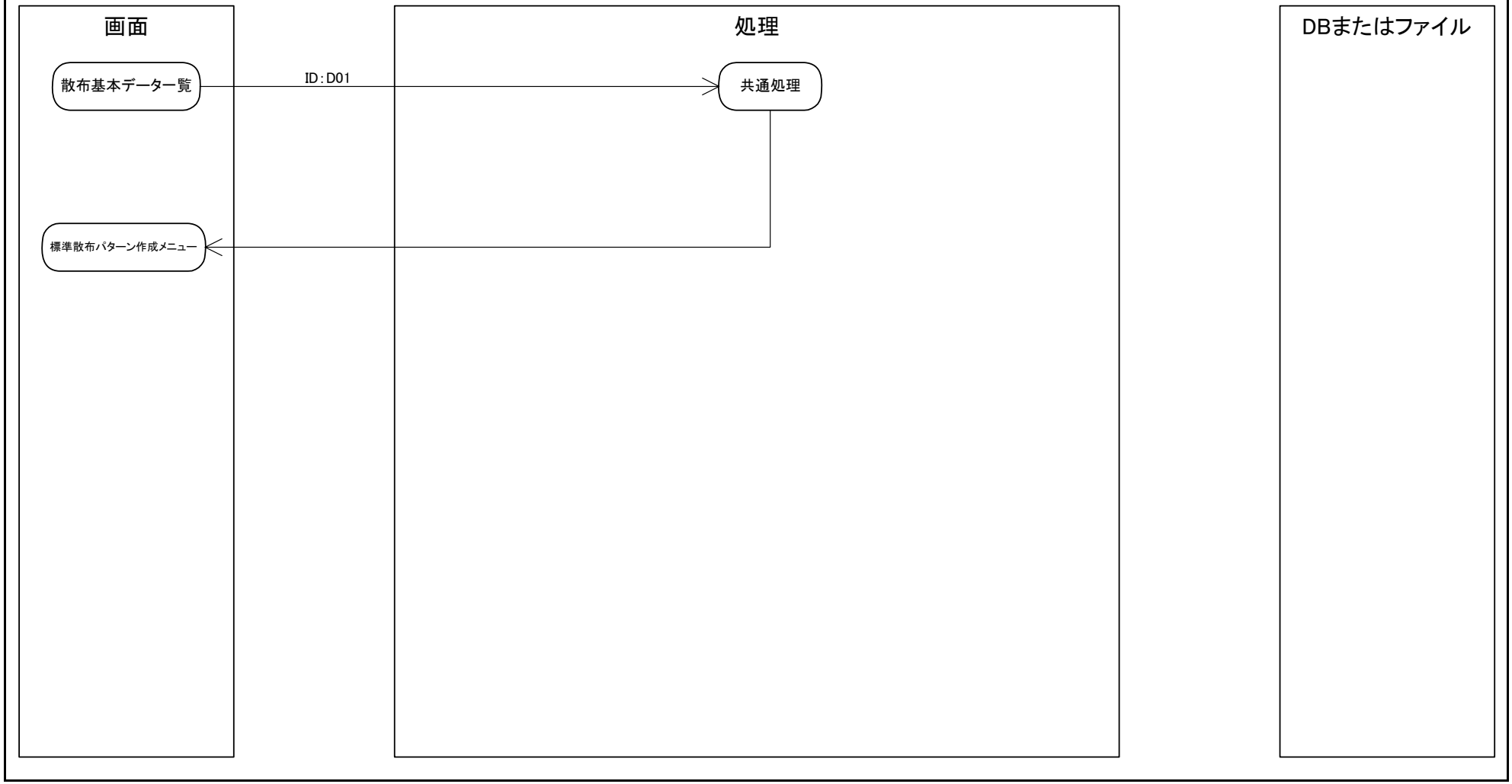


表 3-135　処理フロー

遷移元	散布基本データ一覧	アクション	ボタン：メニューへ	遷移先	トップメニュー

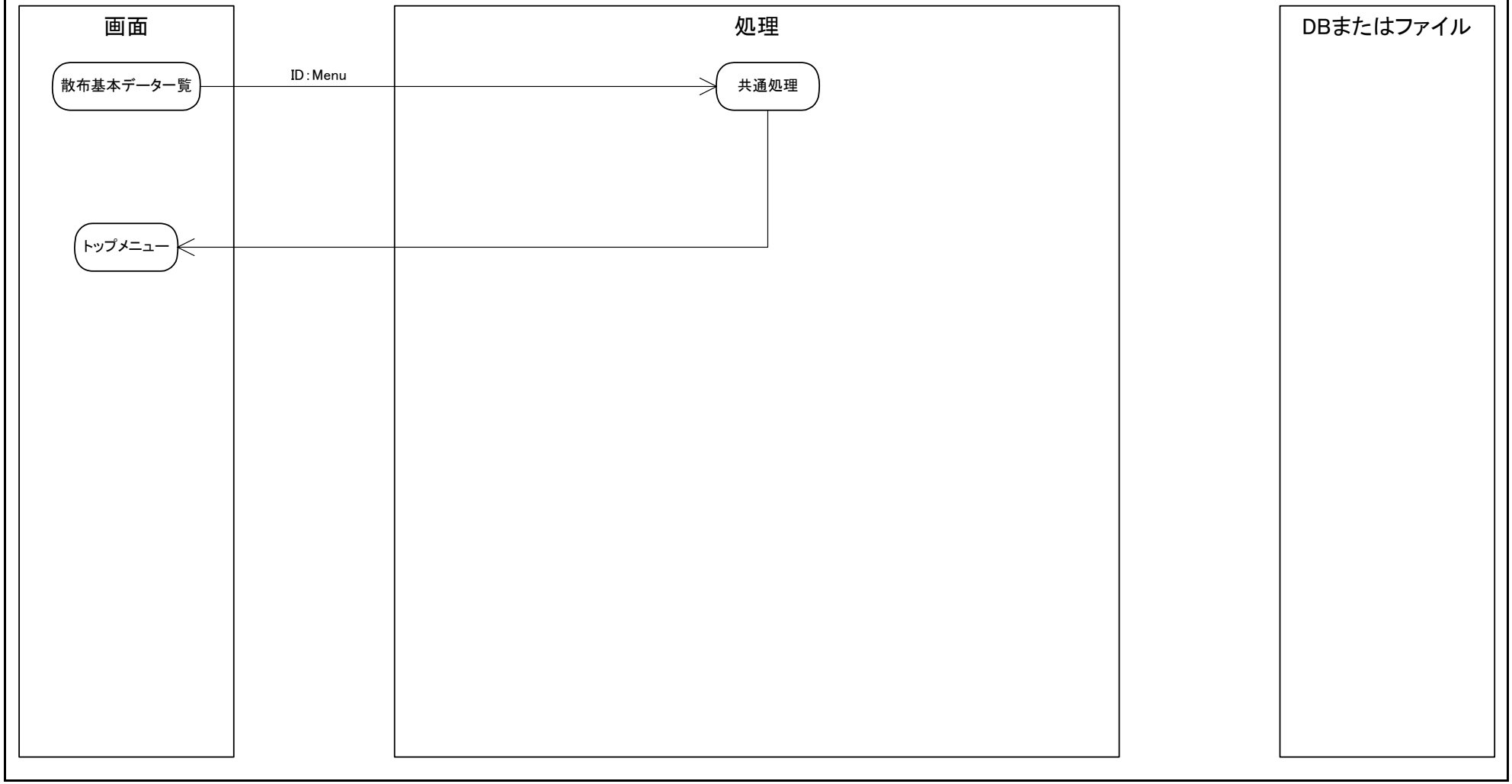


表 3-136　処理フロー

遷移元	散布基本データ修正	アクション	ボタン：修正	遷移先	散布基本データ一覧

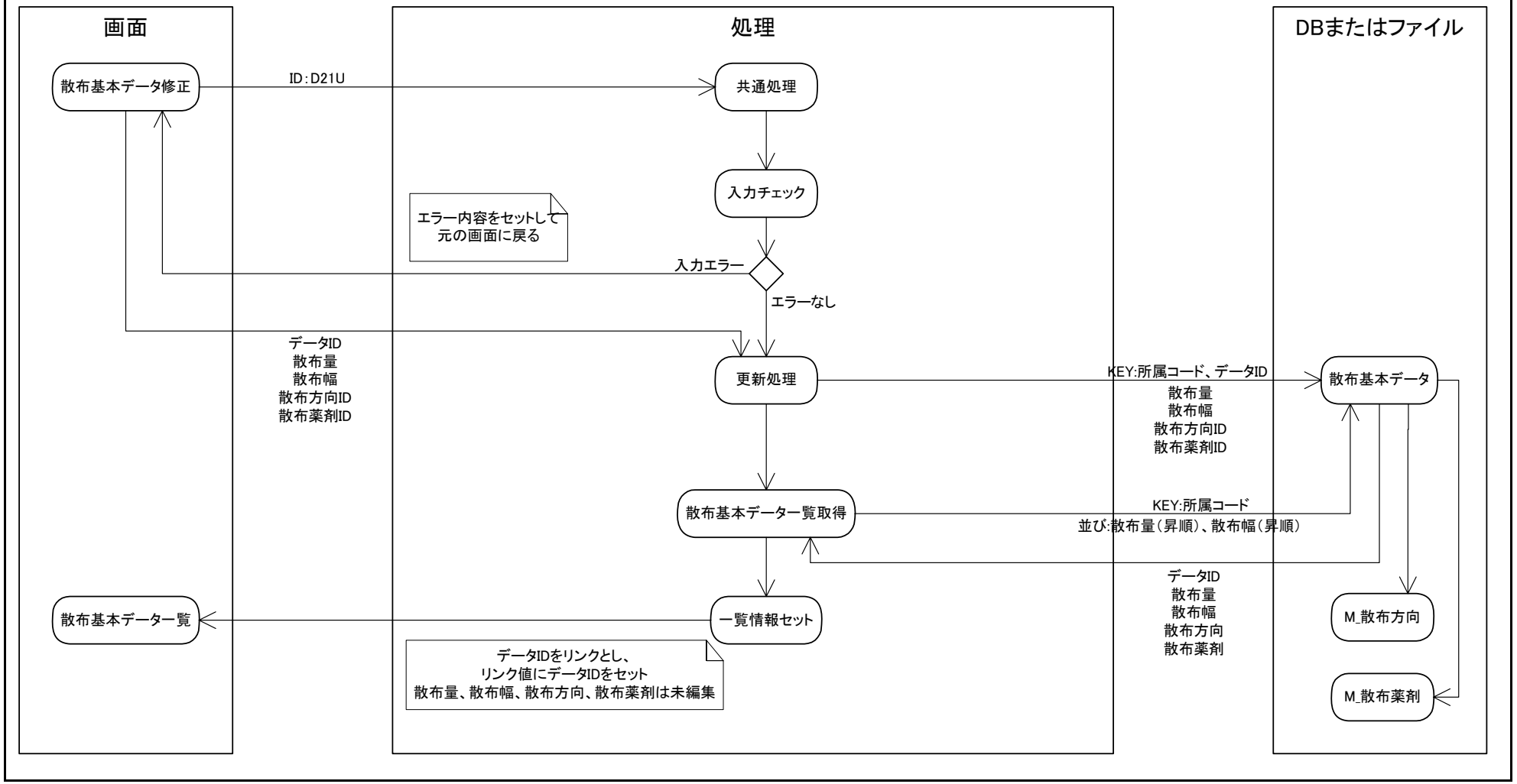


表 3-137　処理フロー

遷移元	散布基本データ修正	アクション	ボタン：複製	遷移先	散布基本データ登録

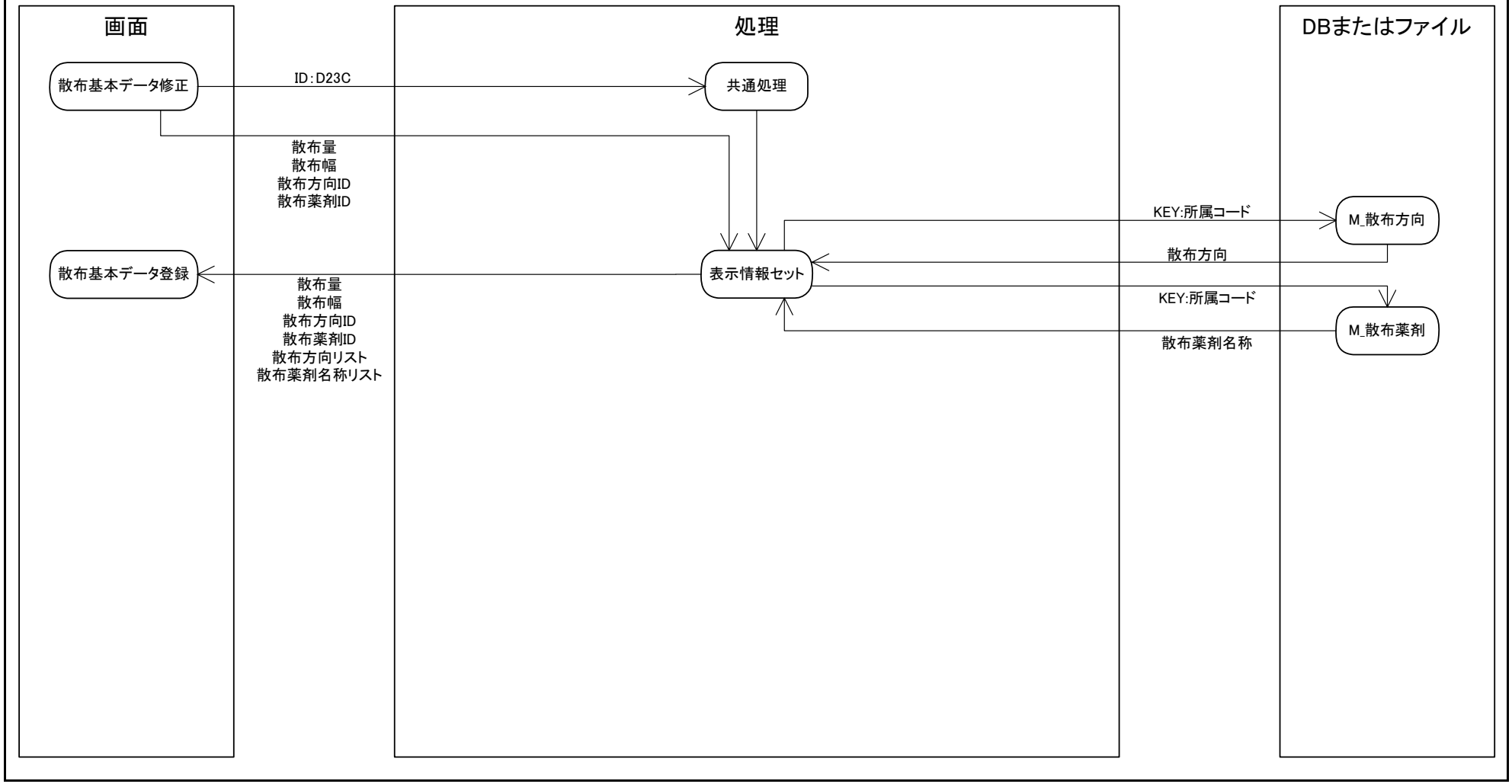


表 3-138　処理フロー

遷移元	散布基本データ修正	アクション	ボタン：削除	遷移先	散布基本データ一覧

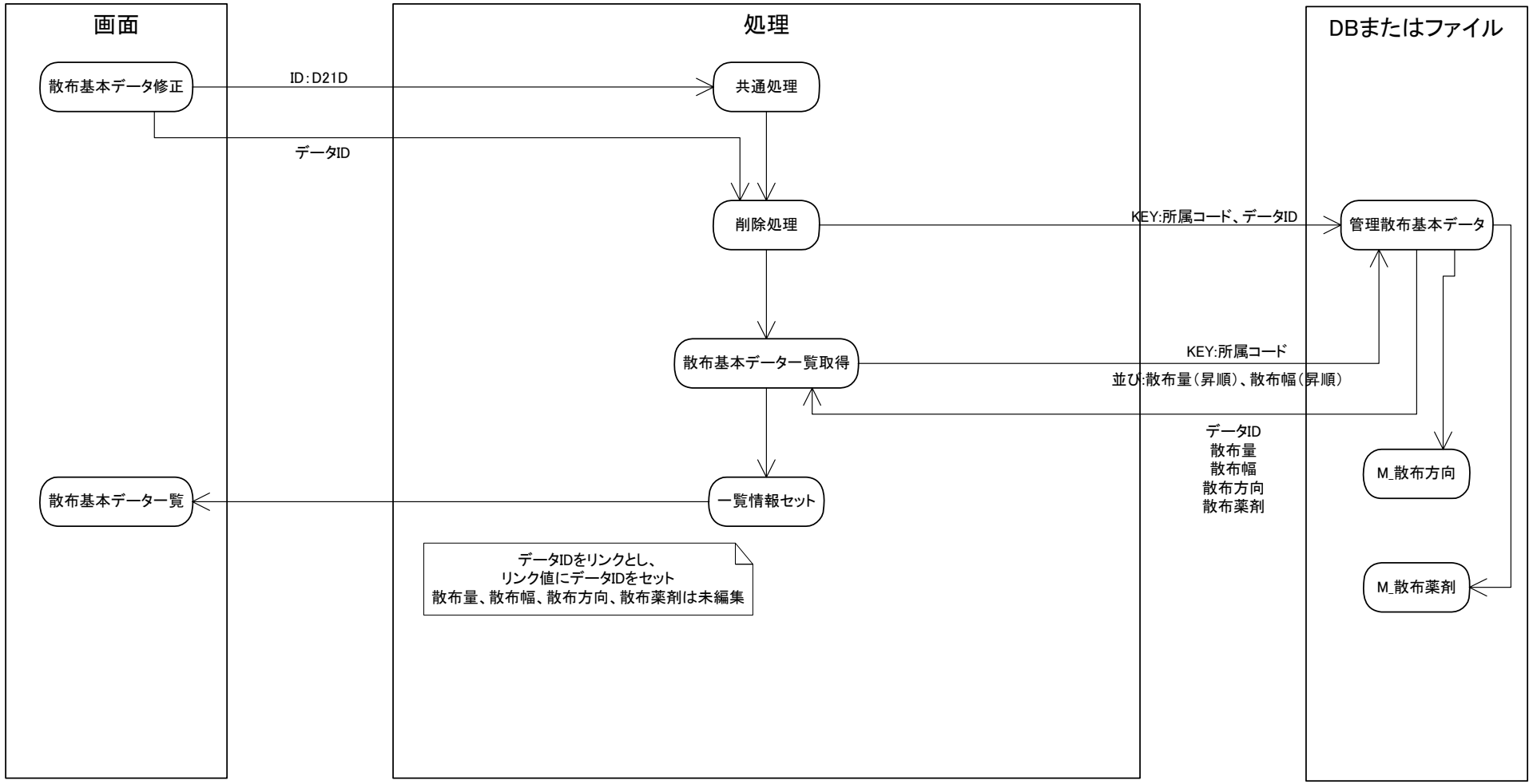


表 3-139　処理フロー

遷移元	散布基本データ修正	アクション	ボタン：戻る	遷移先	散布基本データ一覧

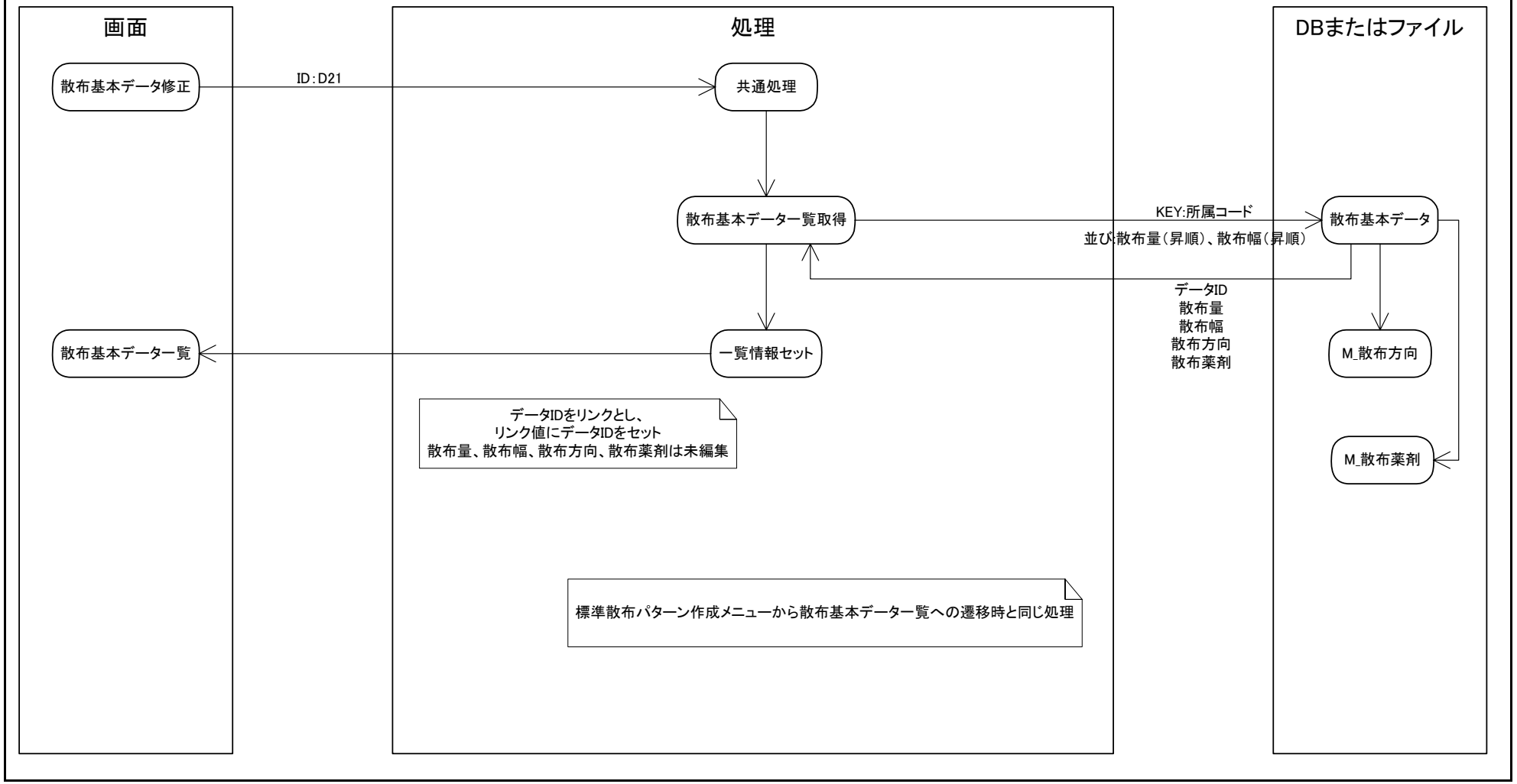


表 3-140　処理フロー

遷移元	散布基本データ登録	アクション	ボタン：登録	遷移先	散布基本データ一覧

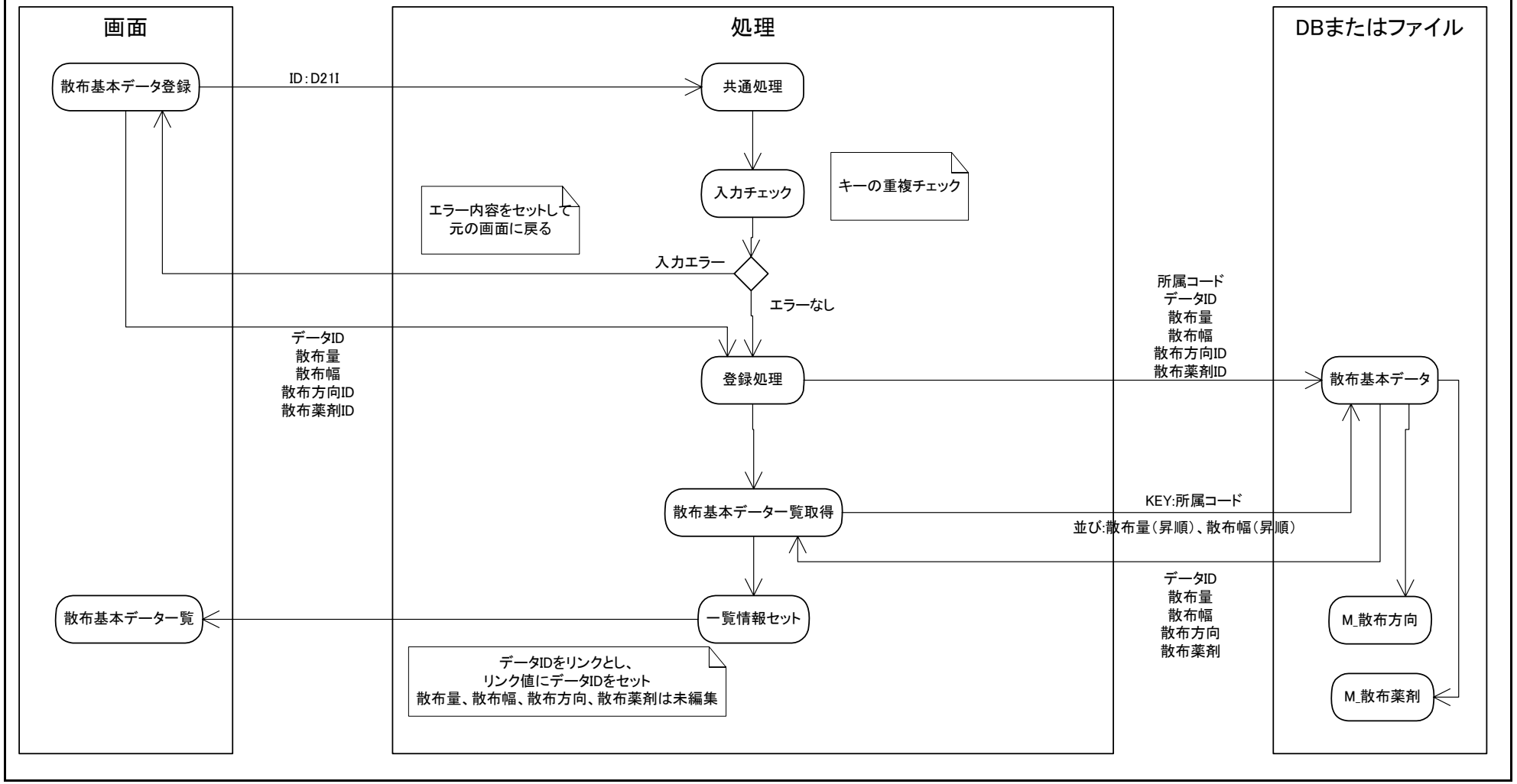


表 3-141　処理フロー

遷移元	散布基本データ登録	アクション	ボタン：戻る	遷移先	散布基本データ一覧

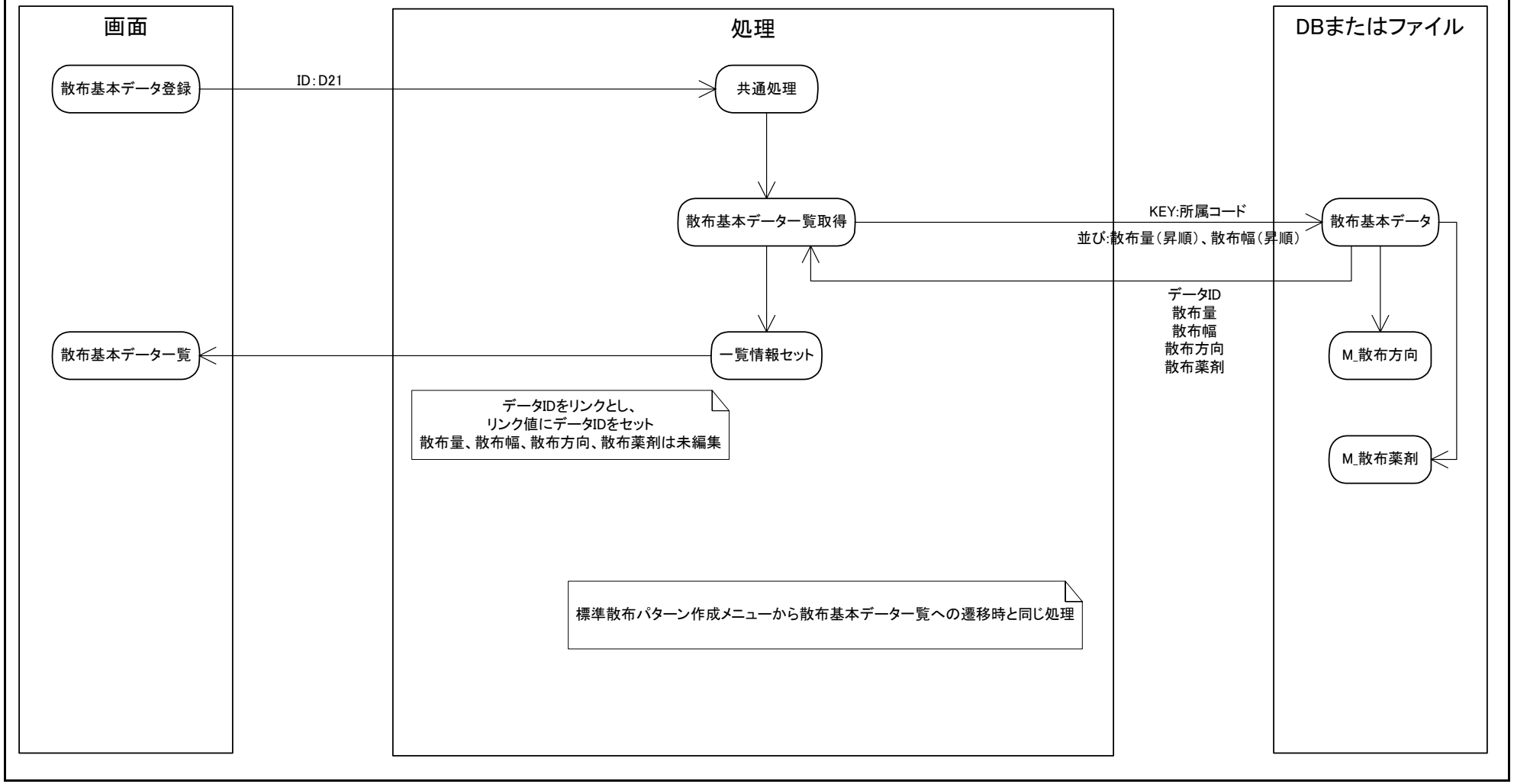


表 3-142　処理フロー

遷移元	散布標準パターン作成メニュー	アクション	リンク：標準散布パターン作成	遷移先	標準散布パターン一覧

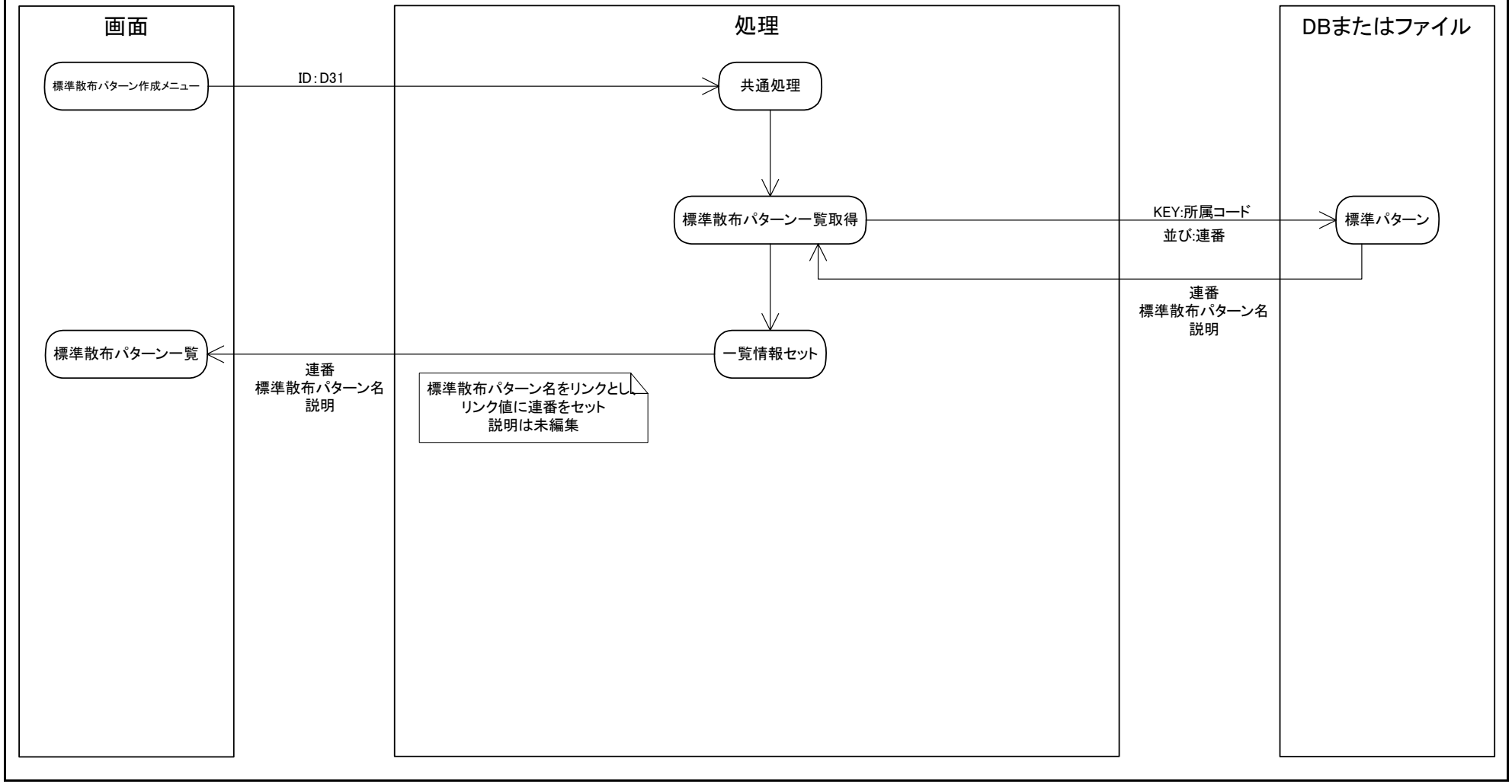


表 3-143　処理フロー

遷移元	標準散布パターン一覧	アクション	リンク：標準散布パターン名	遷移先	標準散布パターン修正

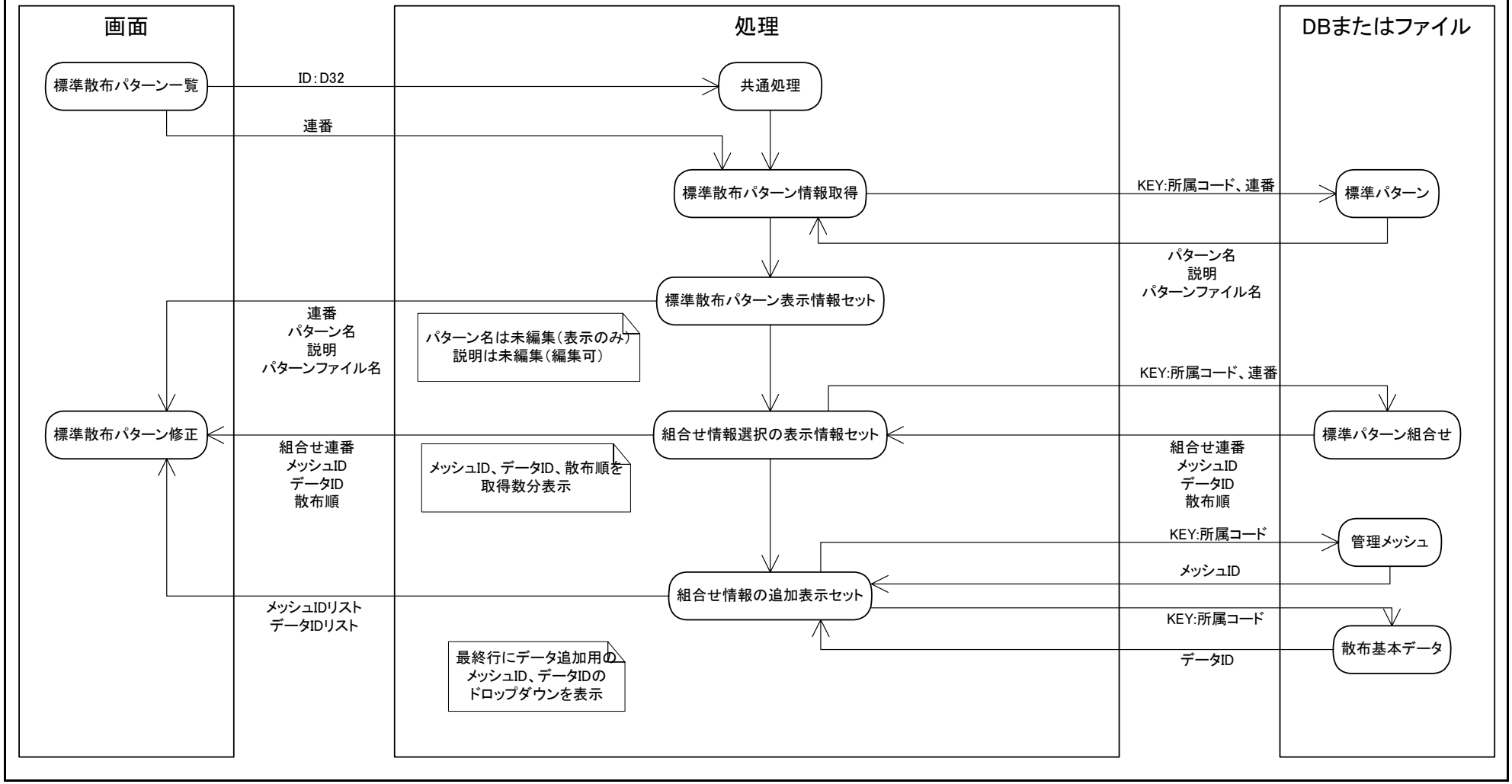


表 3-144　処理フロー

遷移元	標準散布パターン一覧	アクション	ボタン：新規	遷移先	標準散布パターン新規登録

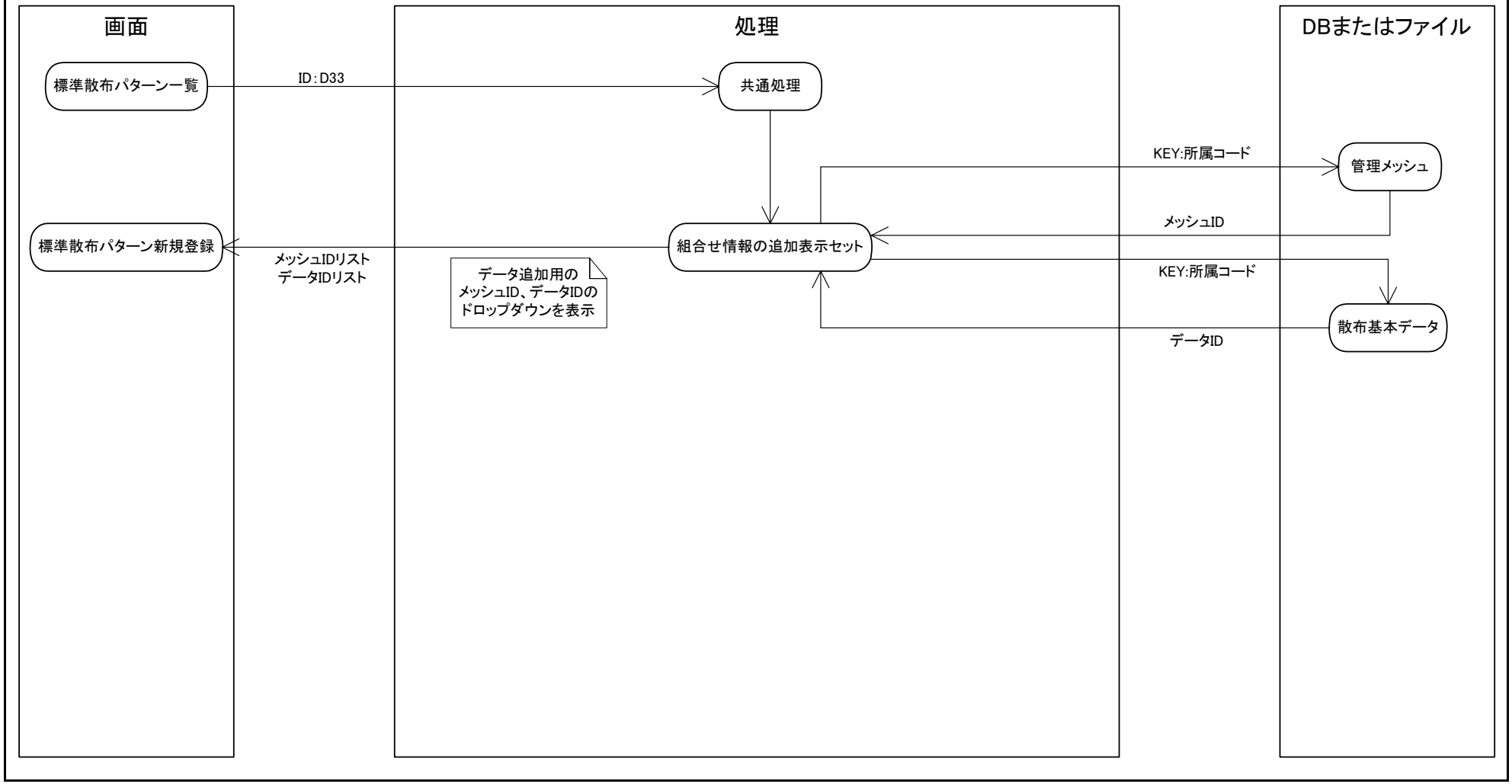


表 3-145　処理フロー

遷移元	標準散布パターン一覧	アクション	ボタン：戻る	遷移先	標準散布パターン作成メニュー

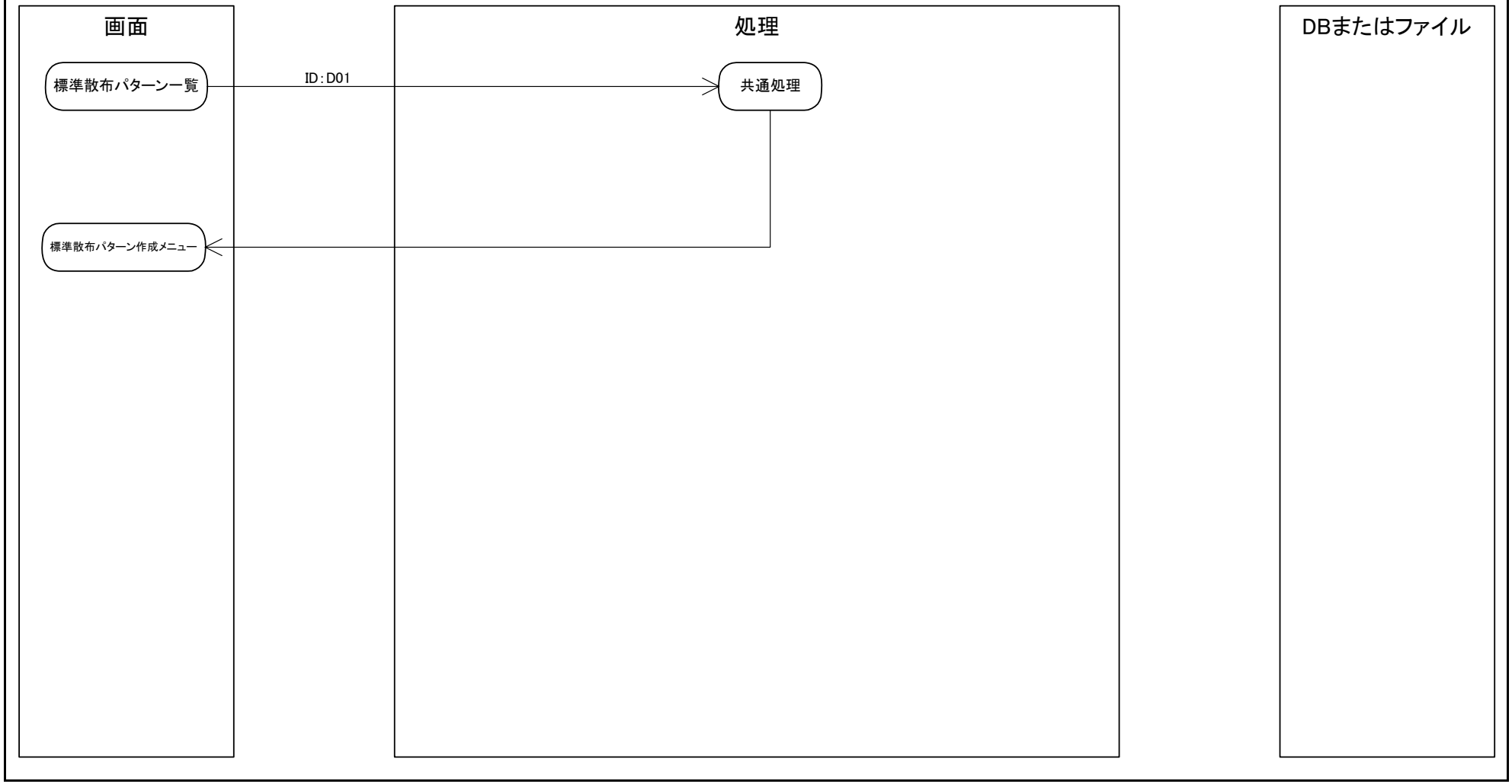


表 3-146　処理フロー

遷移元	標準散布パターン一覧	アクション	ボタン：メニューへ	遷移先	トップメニュー

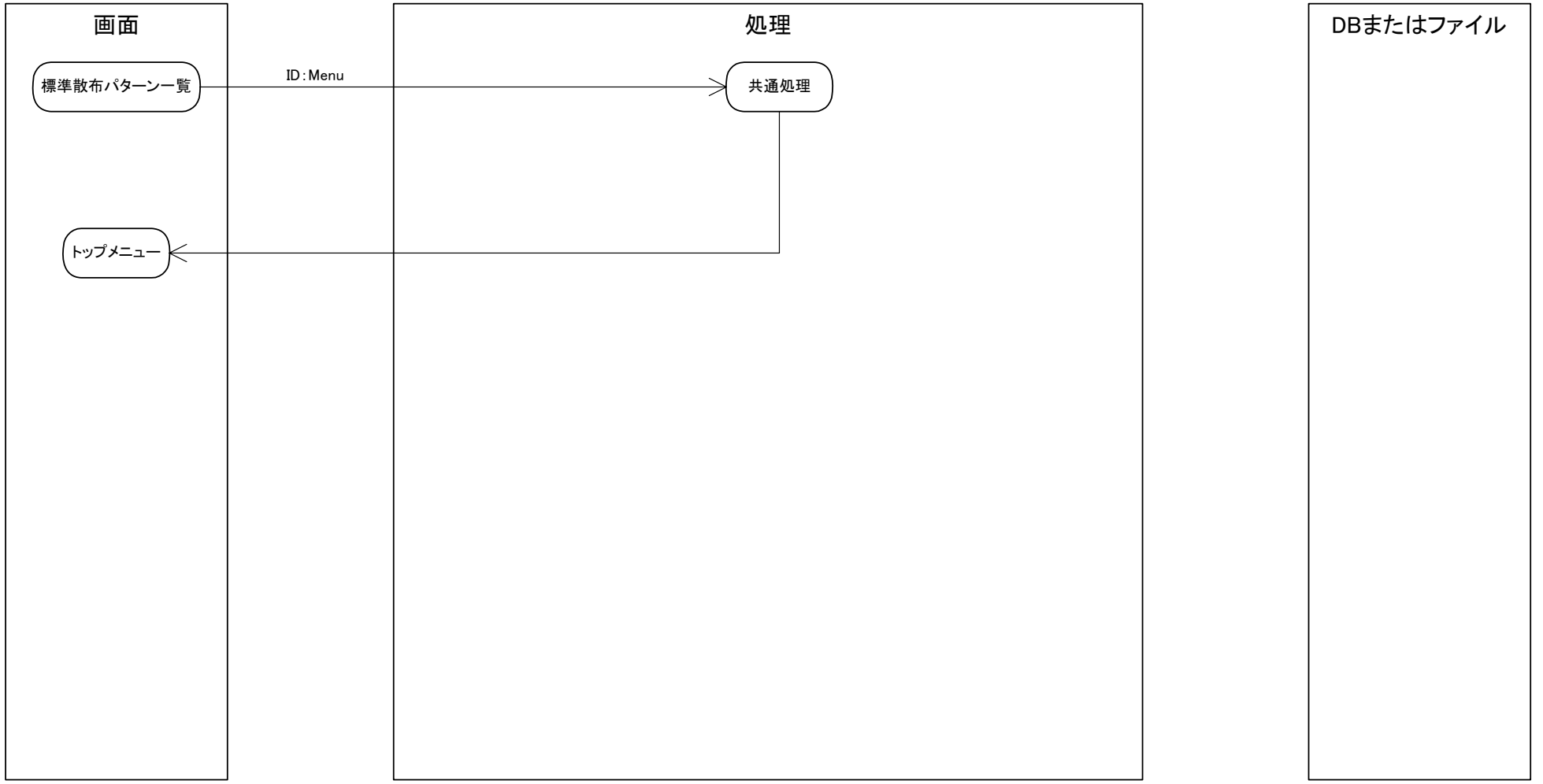


表 3-147　処理フロー

遷移元	標準散布パターン修正	アクション	ボタン：修正	遷移先	標準散布パターン一覧

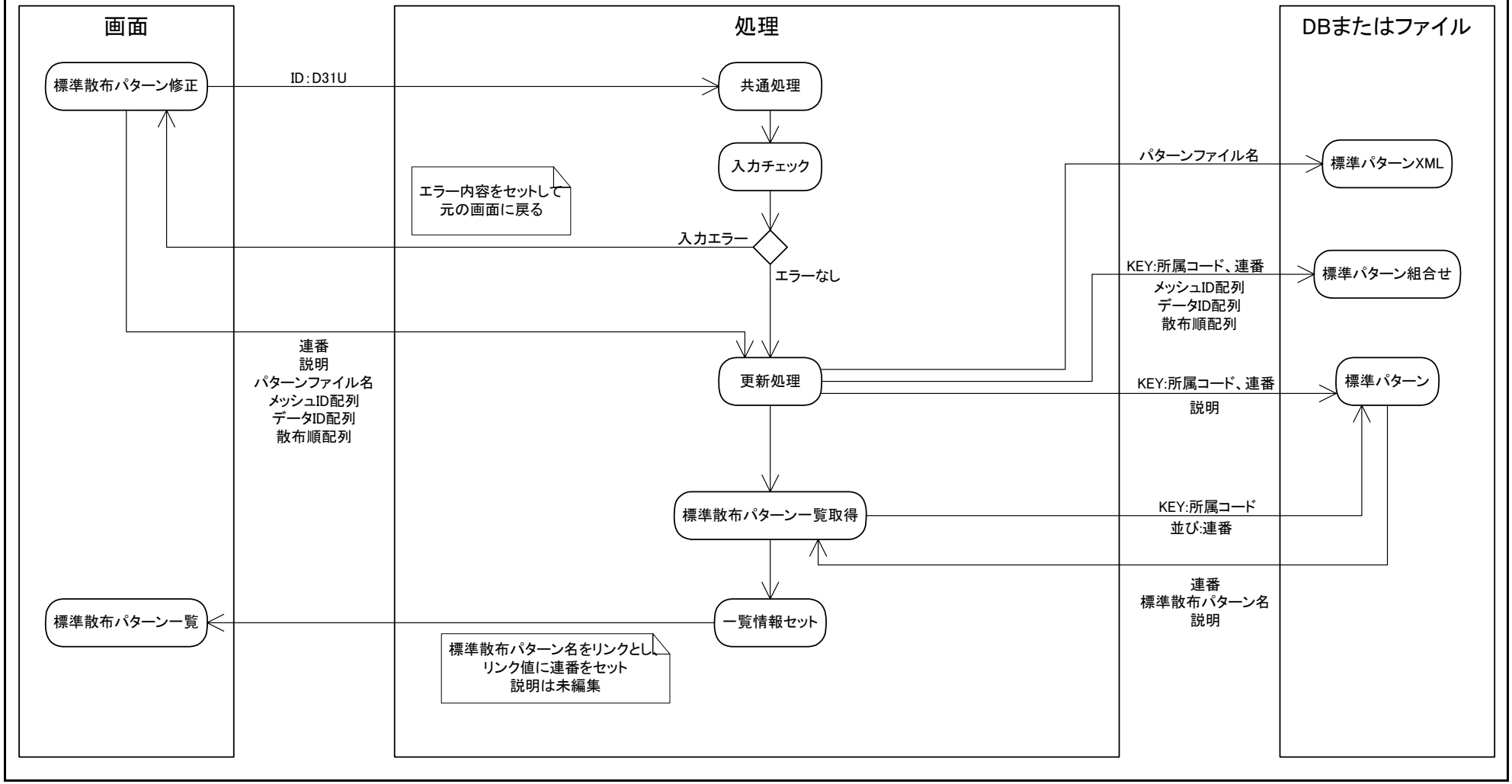


表 3-148　処理フロー

遷移元	標準散布パターン修正	アクション	ボタン：複製	遷移先	標準散布パターン登録

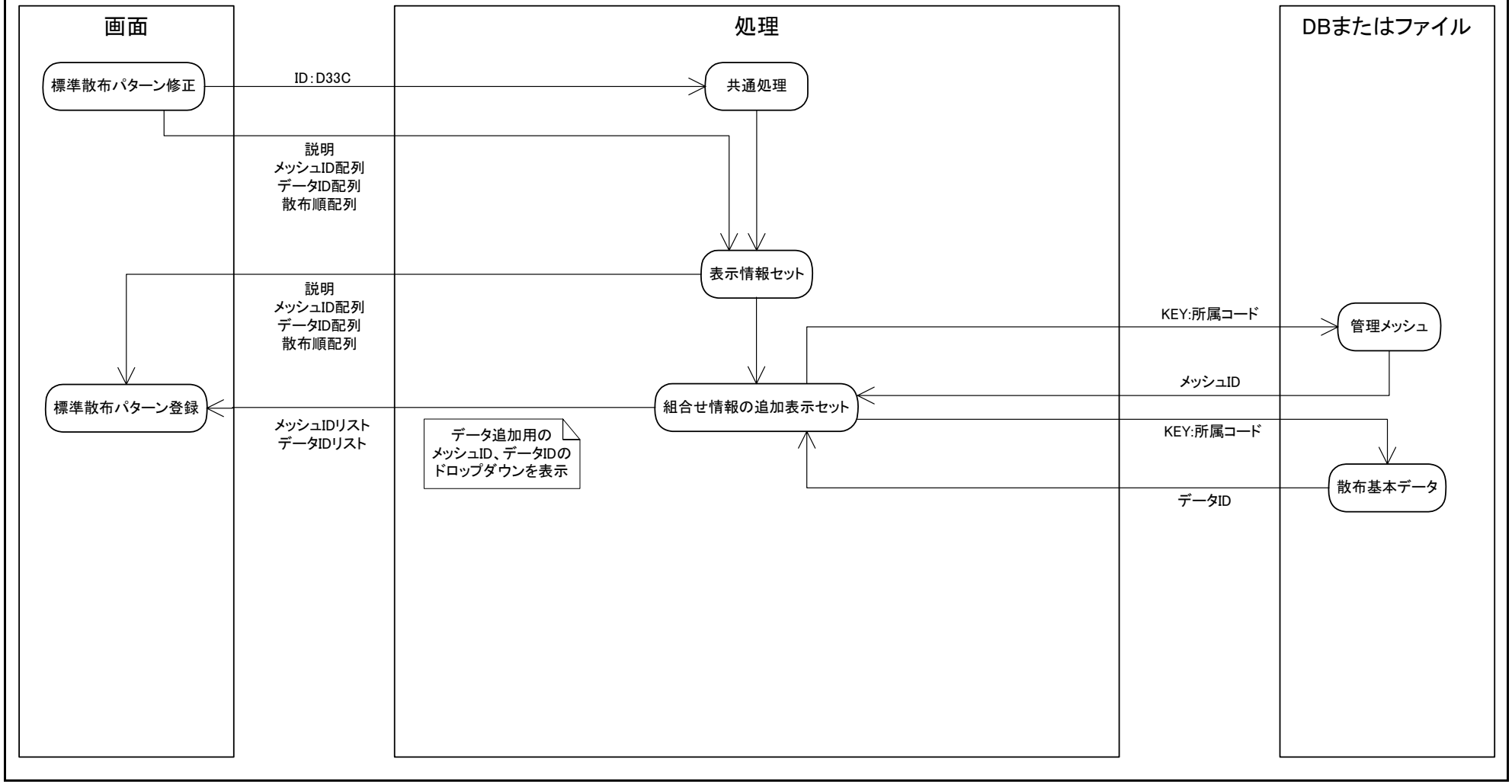


表 3-149　処理フロー

遷移元	標準散布パターン修正	アクション	ボタン：削除	遷移先	標準散布パターン一覧

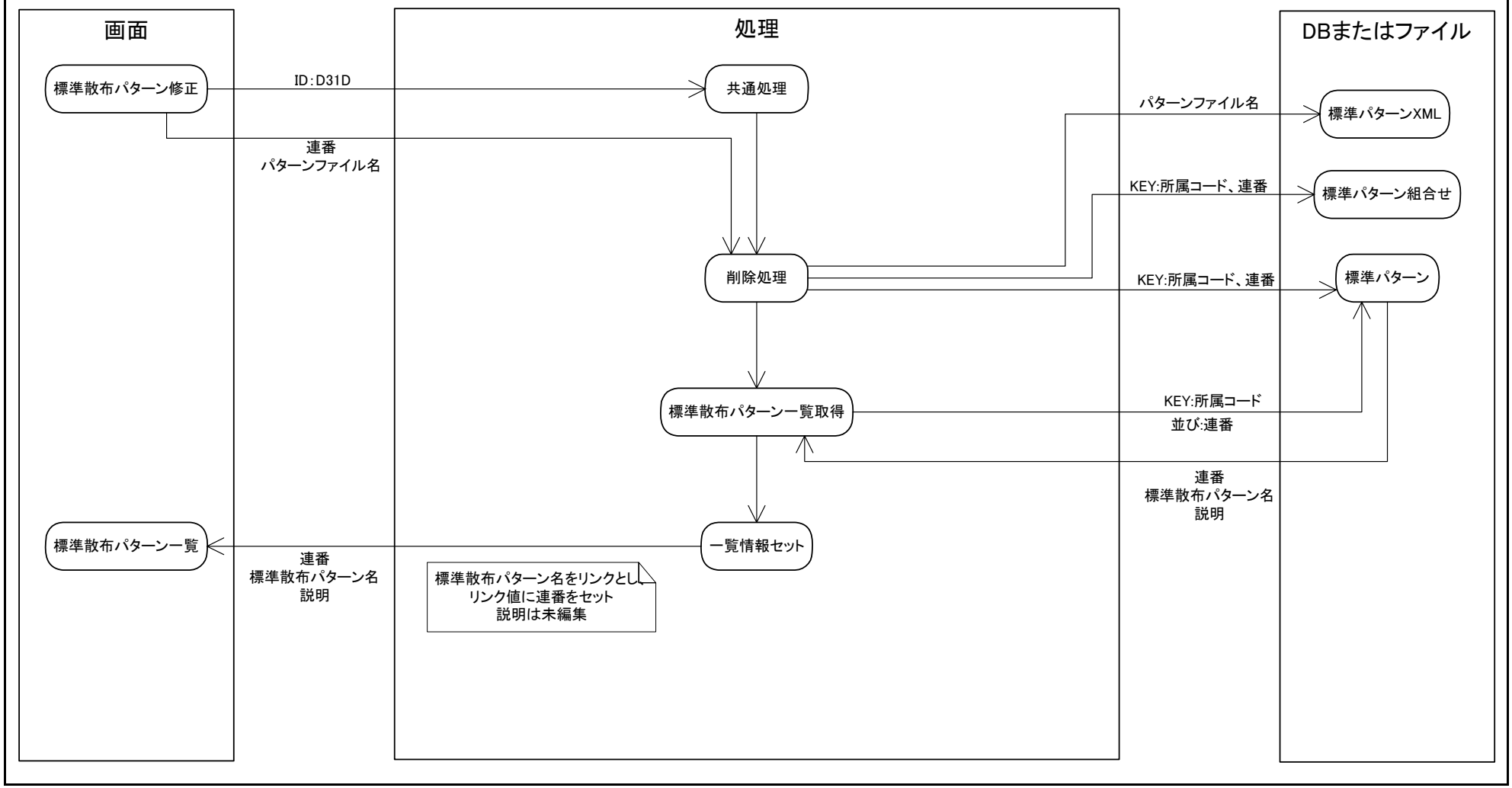


表 3-150　処理フロー

遷移元	標準散布パターン修正	アクション	ボタン：戻る	遷移先	標準散布パターン一覧

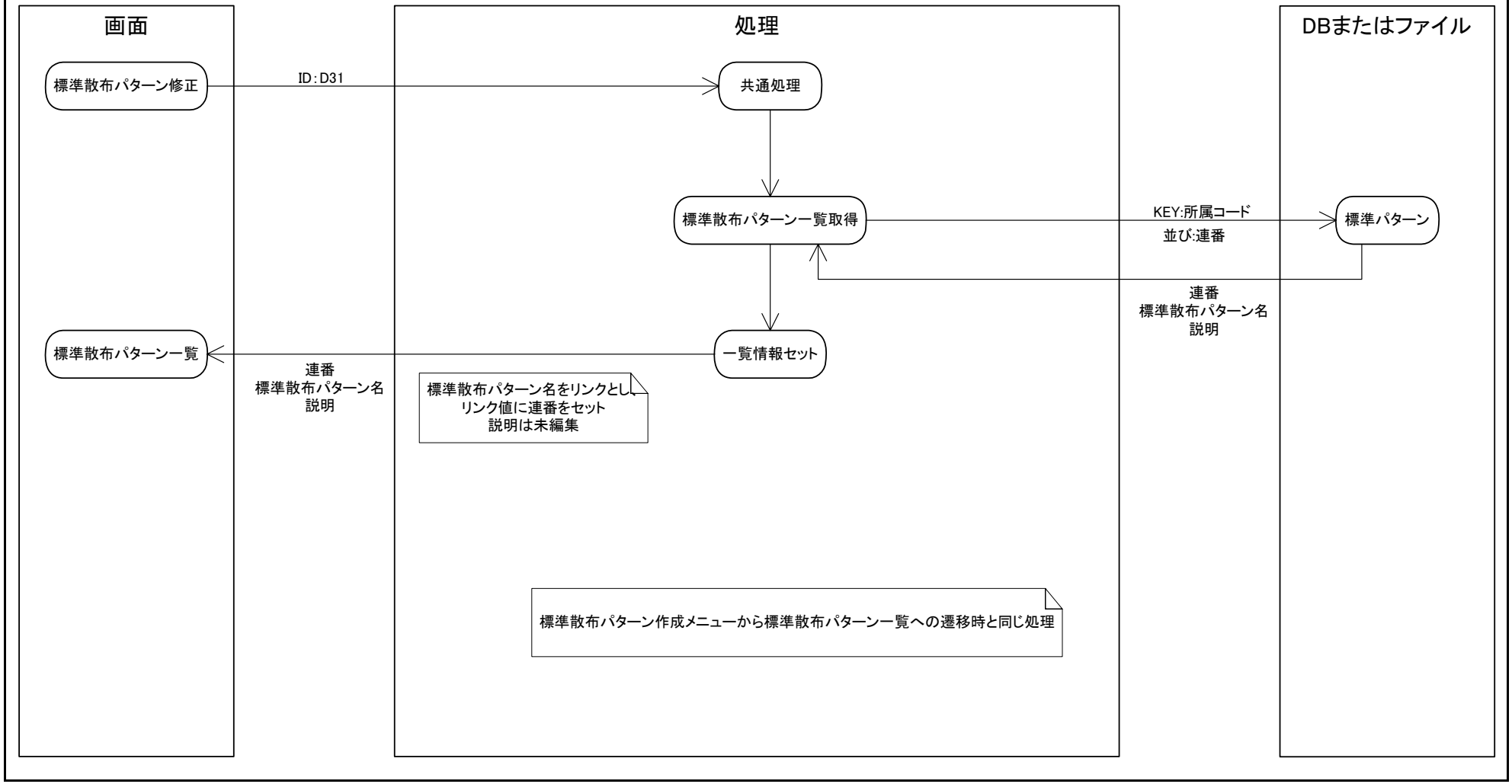


表 3-151　処理フロー

遷移元	標準散布パターン修正	アクション	ボタン：修正・削除	遷移先	管理メッシュ・散布基本データの修正・削除

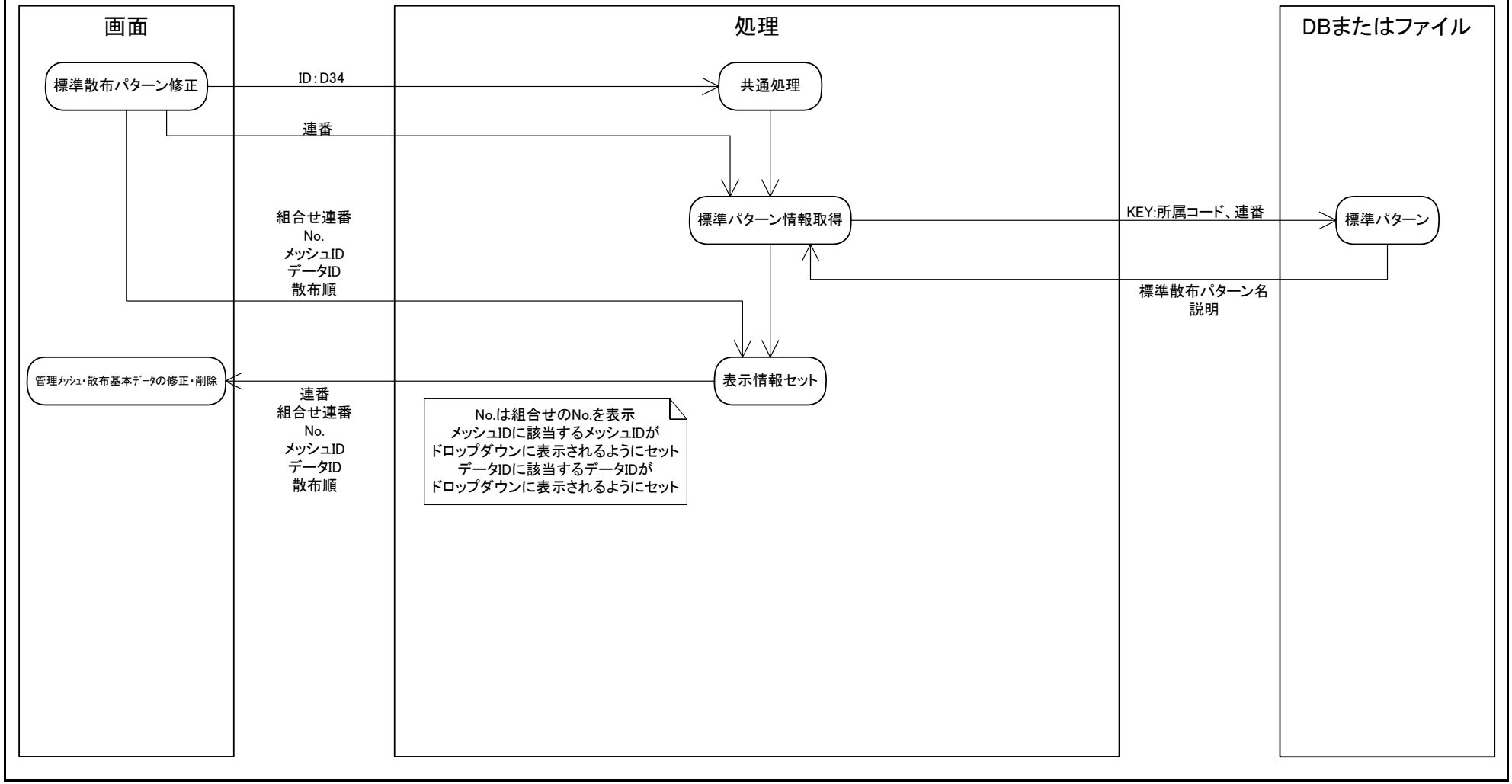


表 3-152　処理フロー

遷移元	標準散布パターン修正	アクション	ボタン：追加	遷移先	標準散布パターン修正

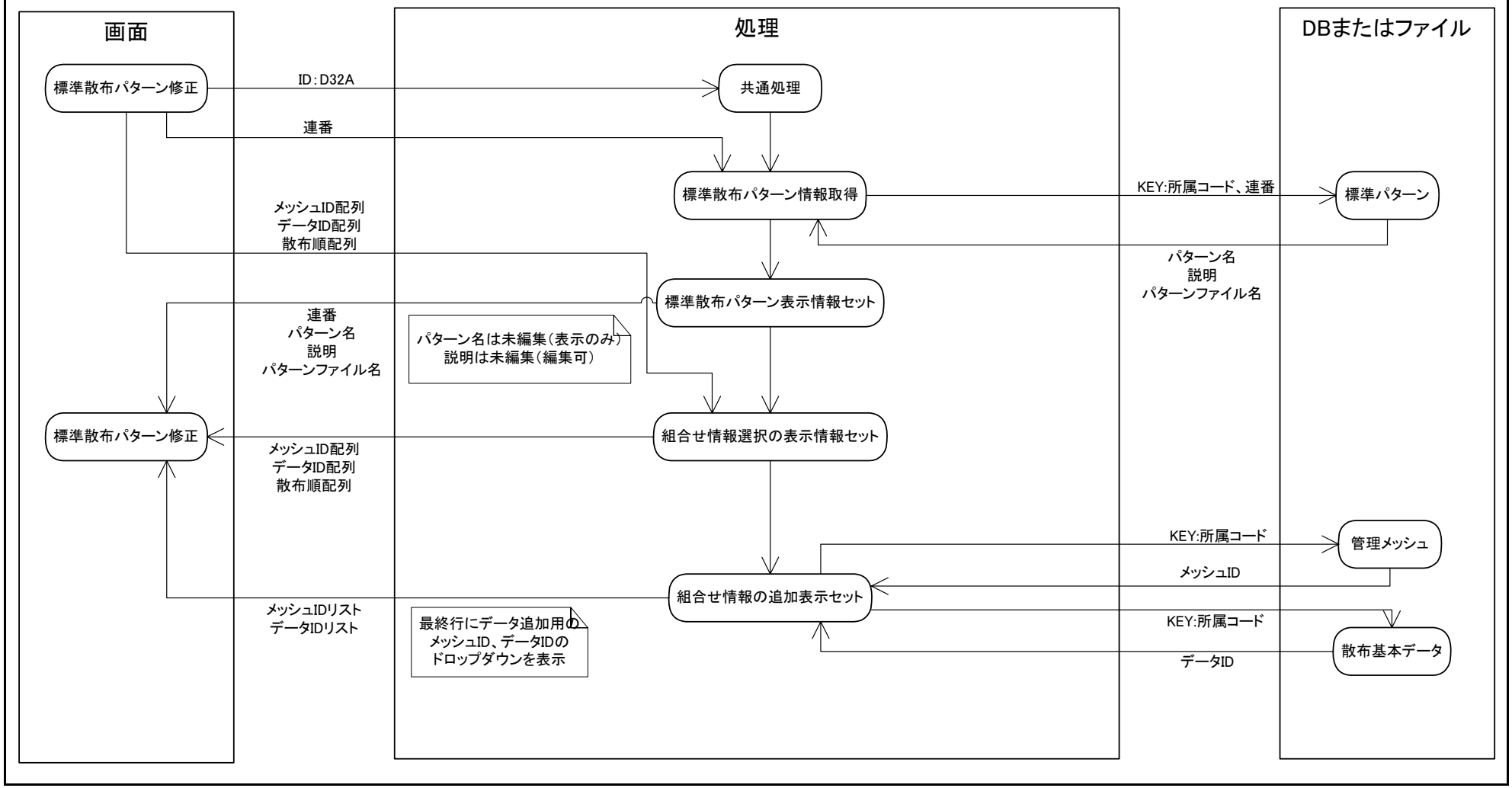


表 3-153　処理フロー

遷移元	標準散布パターン登録	アクション	ボタン：登録	遷移先	標準散布パターン一覧

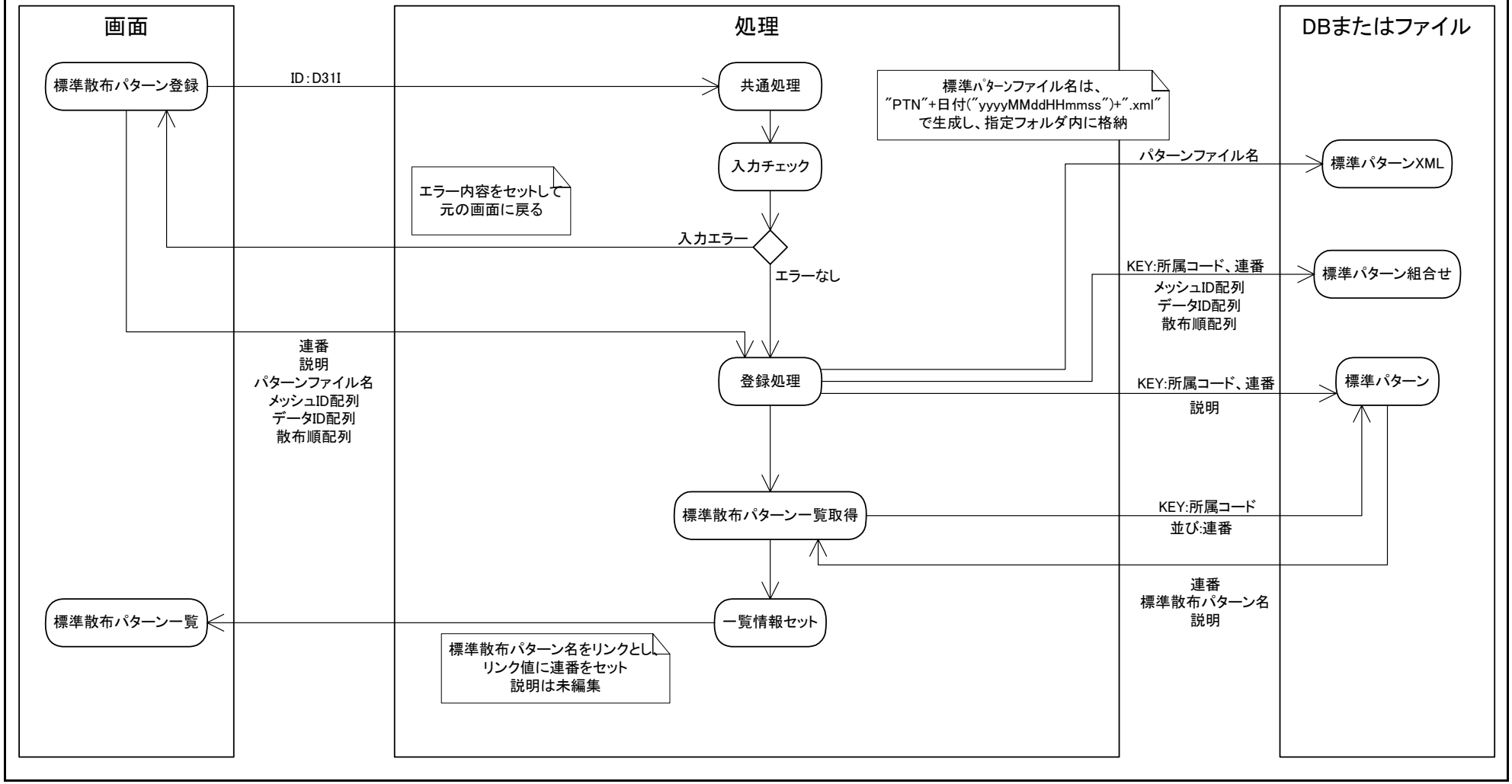


表 3-154　処理フロー

遷移元	標準散布パターン登録	アクション	ボタン：戻る	遷移先	標準散布パターン一覧

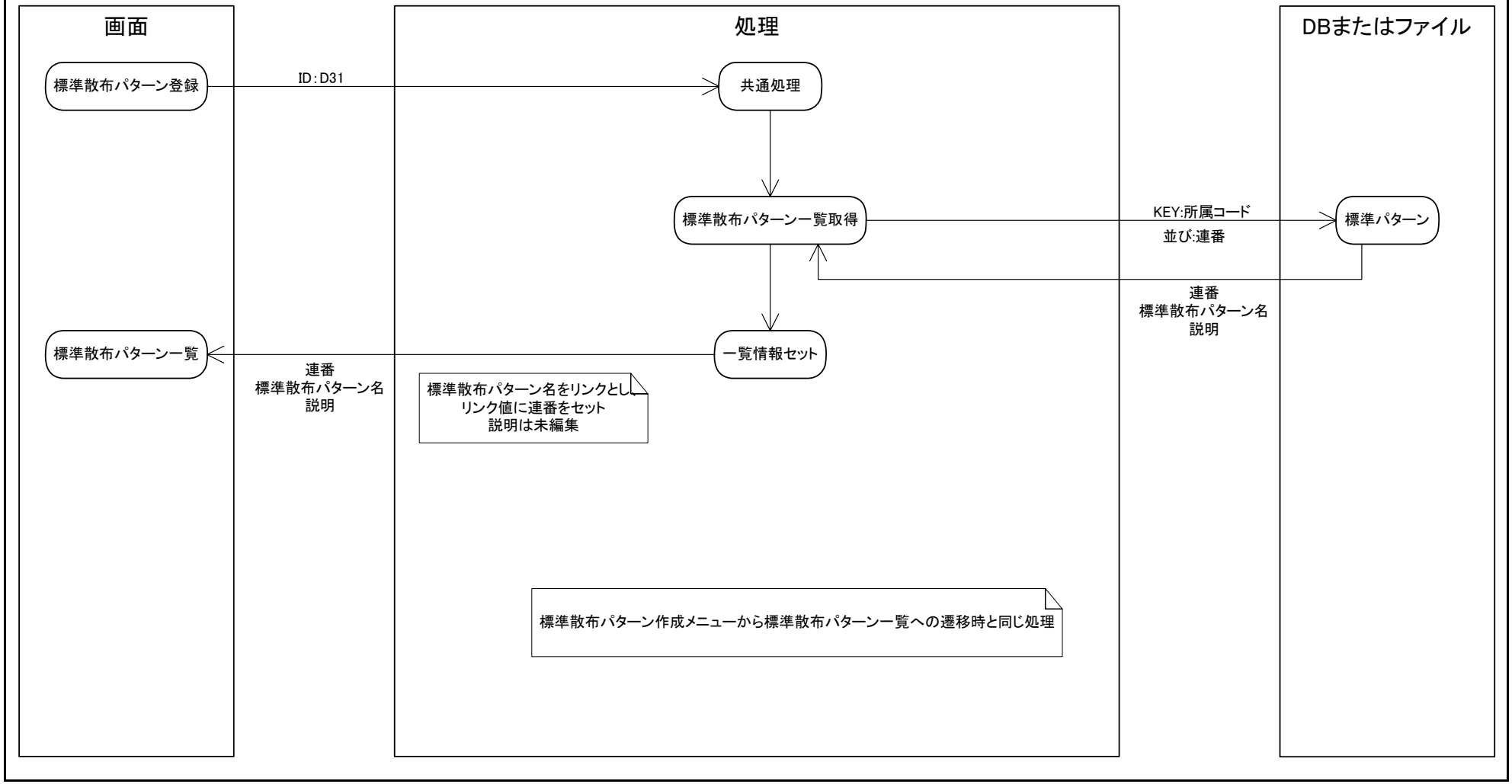


表 3-155　処理フロー

遷移元	標準散布パターン登録	アクション	ボタン：修正・削除	遷移先	管理メッシュ・散布基本データの修正・削除

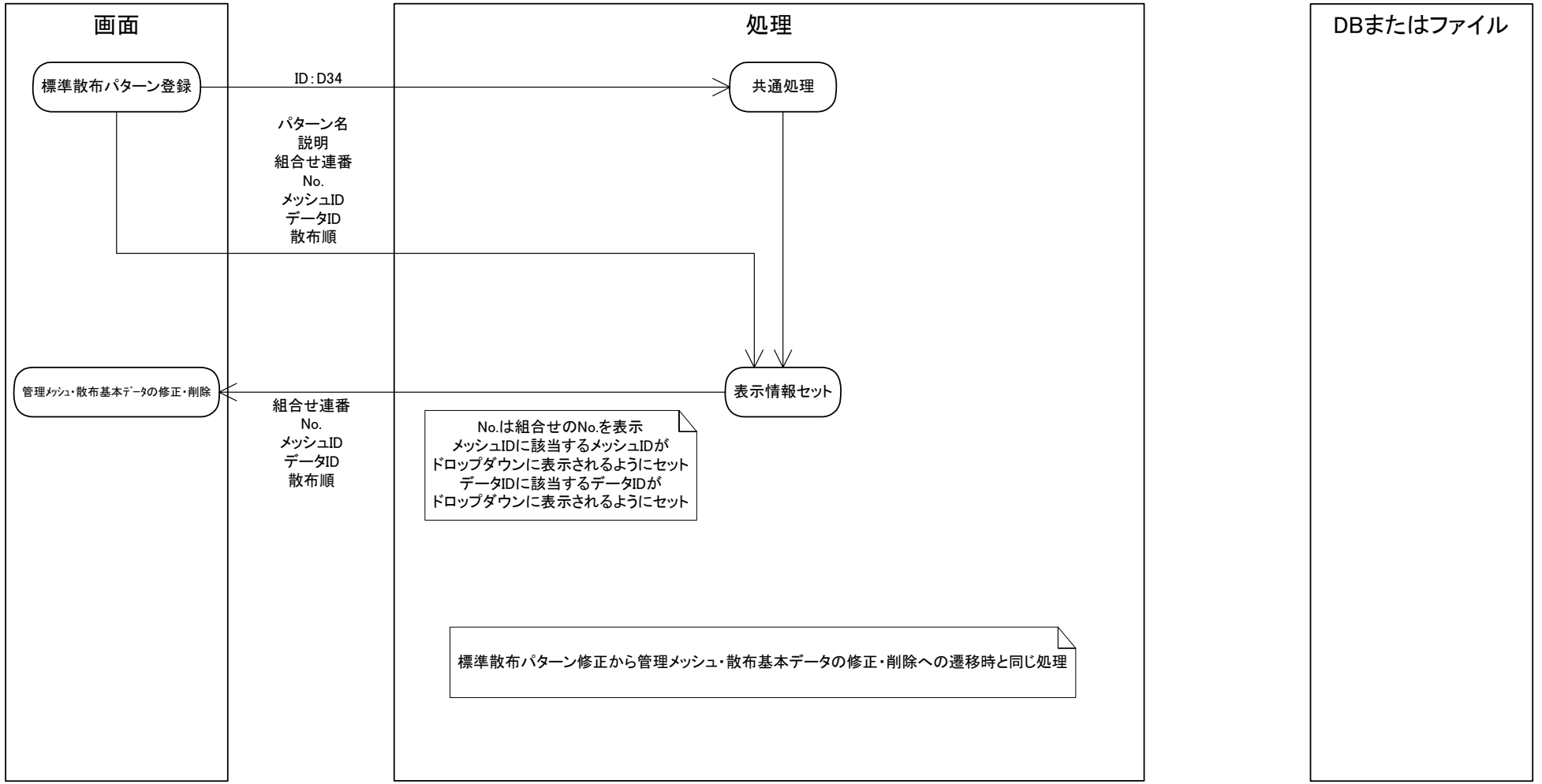


表 3-156　処理フロー

遷移元	標準散布パターン登録	アクション	ボタン：追加	遷移先	標準散布パターン登録

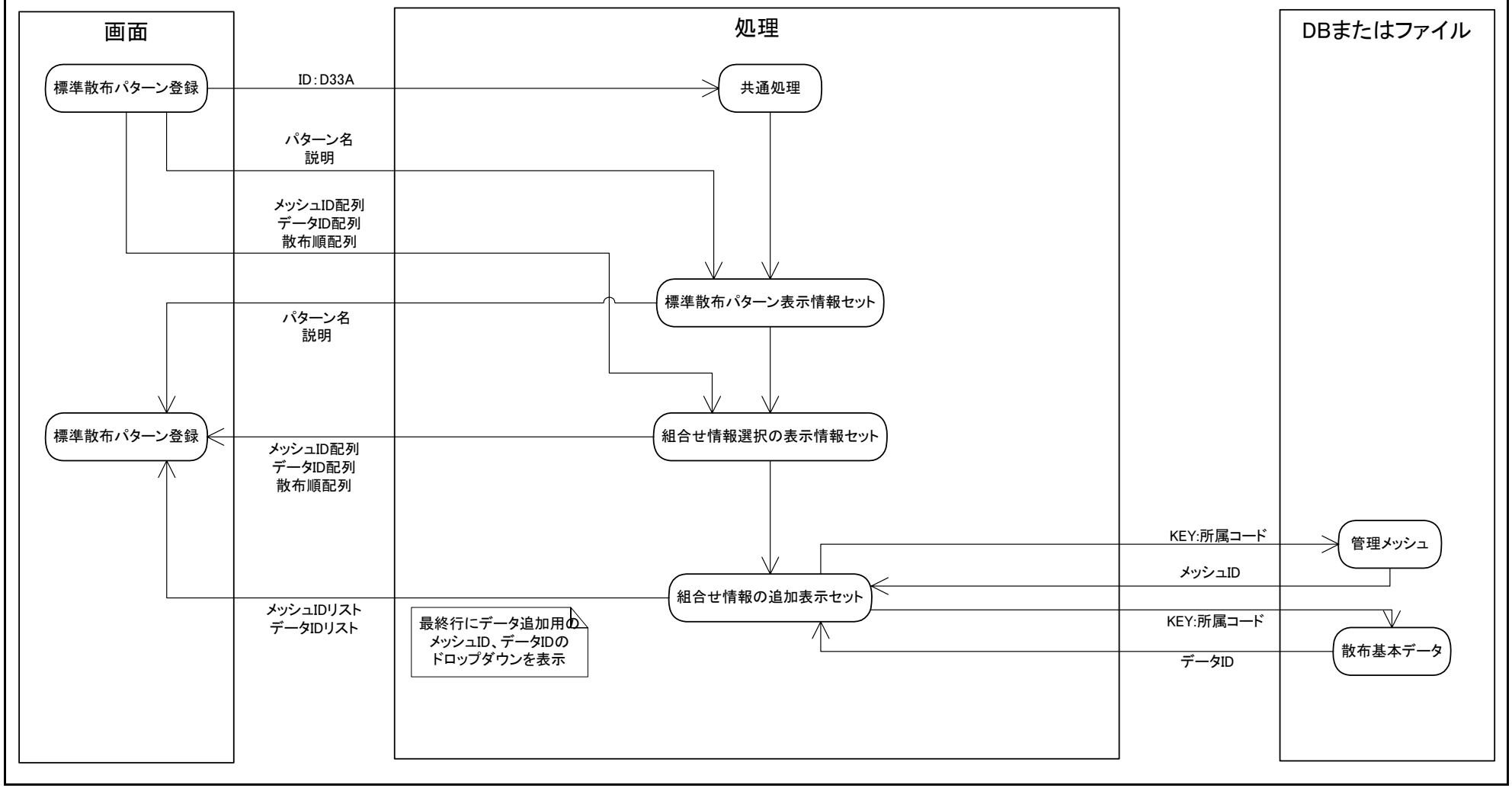


表 3-157　処理フロー

遷移元	管理メッシュ・散布基本データの修正・削除	アクション	ボタン：修正	遷移先	標準散布パターン修正

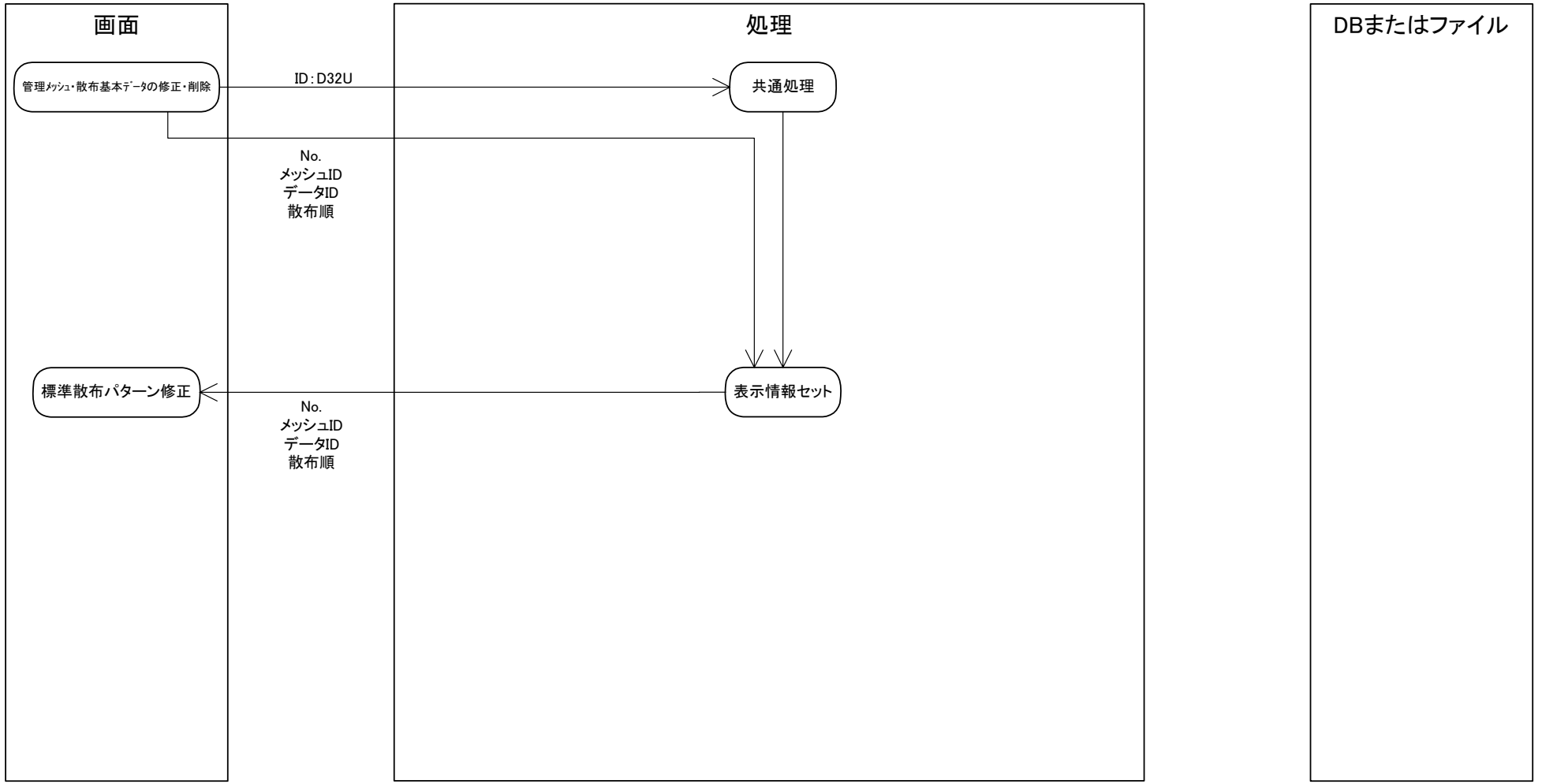


表 3-158　処理フロー

遷移元	管理メッシュ・散布基本データの修正・削除	アクション	ボタン：修正	遷移先	標準散布パターン登録

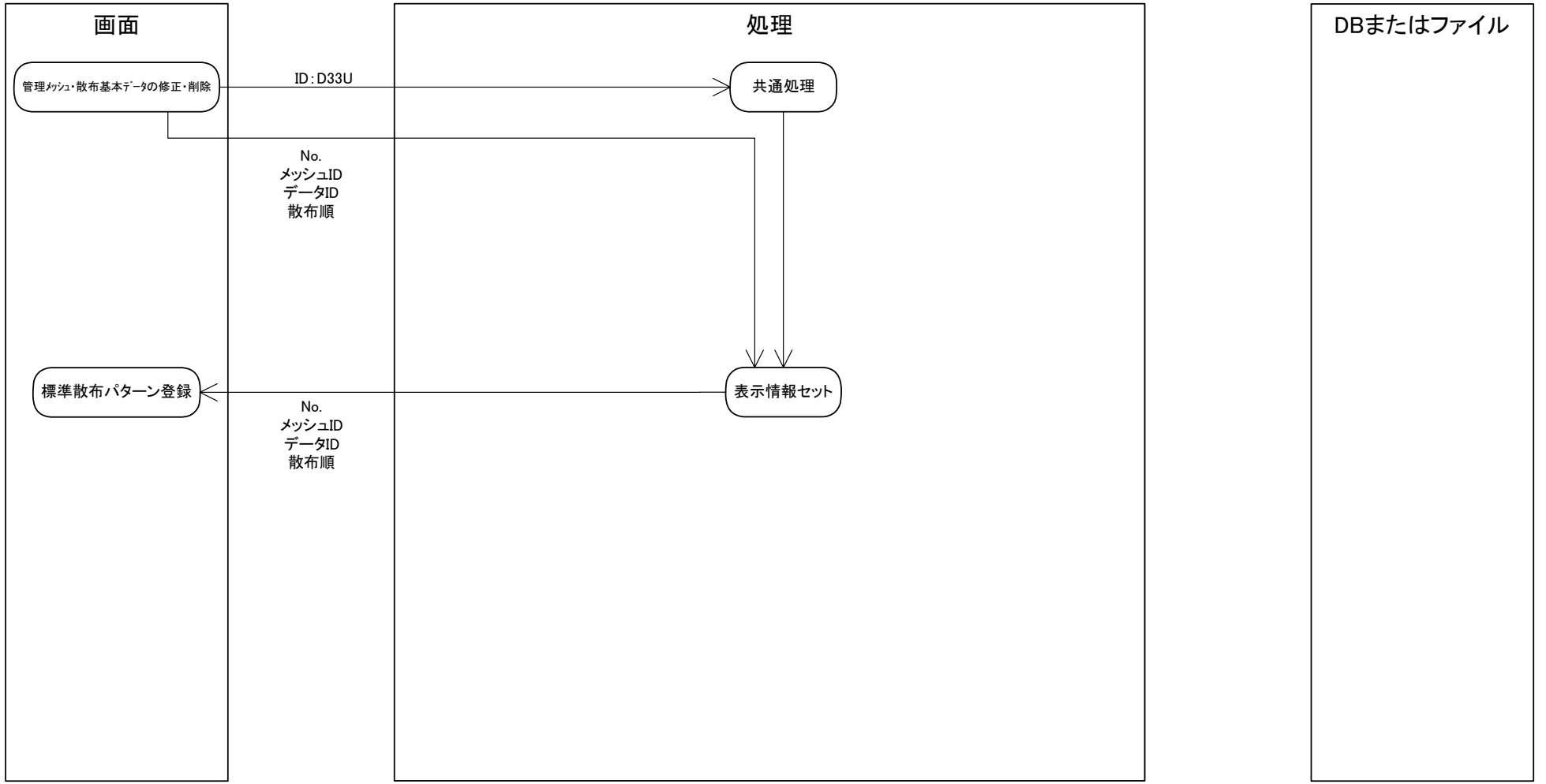


表 3-159　処理フロー

遷移元	管理メッシュ・散布基本データの修正・削除	アクション	ボタン：削除	遷移先	標準散布パターン修正

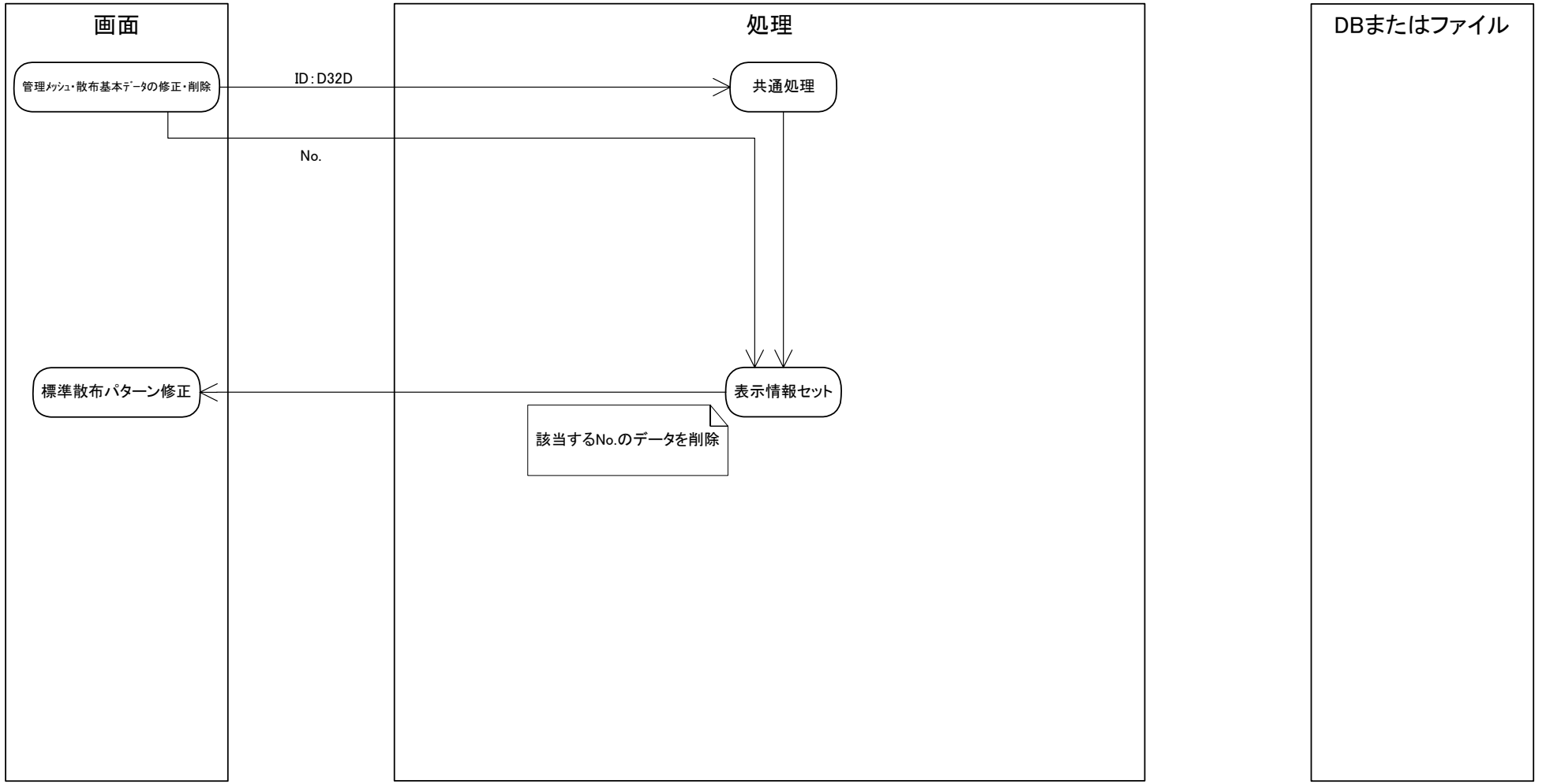


表 3-160　処理フロー

遷移元	管理メッシュ・散布基本データの修正・削除	アクション	ボタン：削除	遷移先	標準散布パターン登録

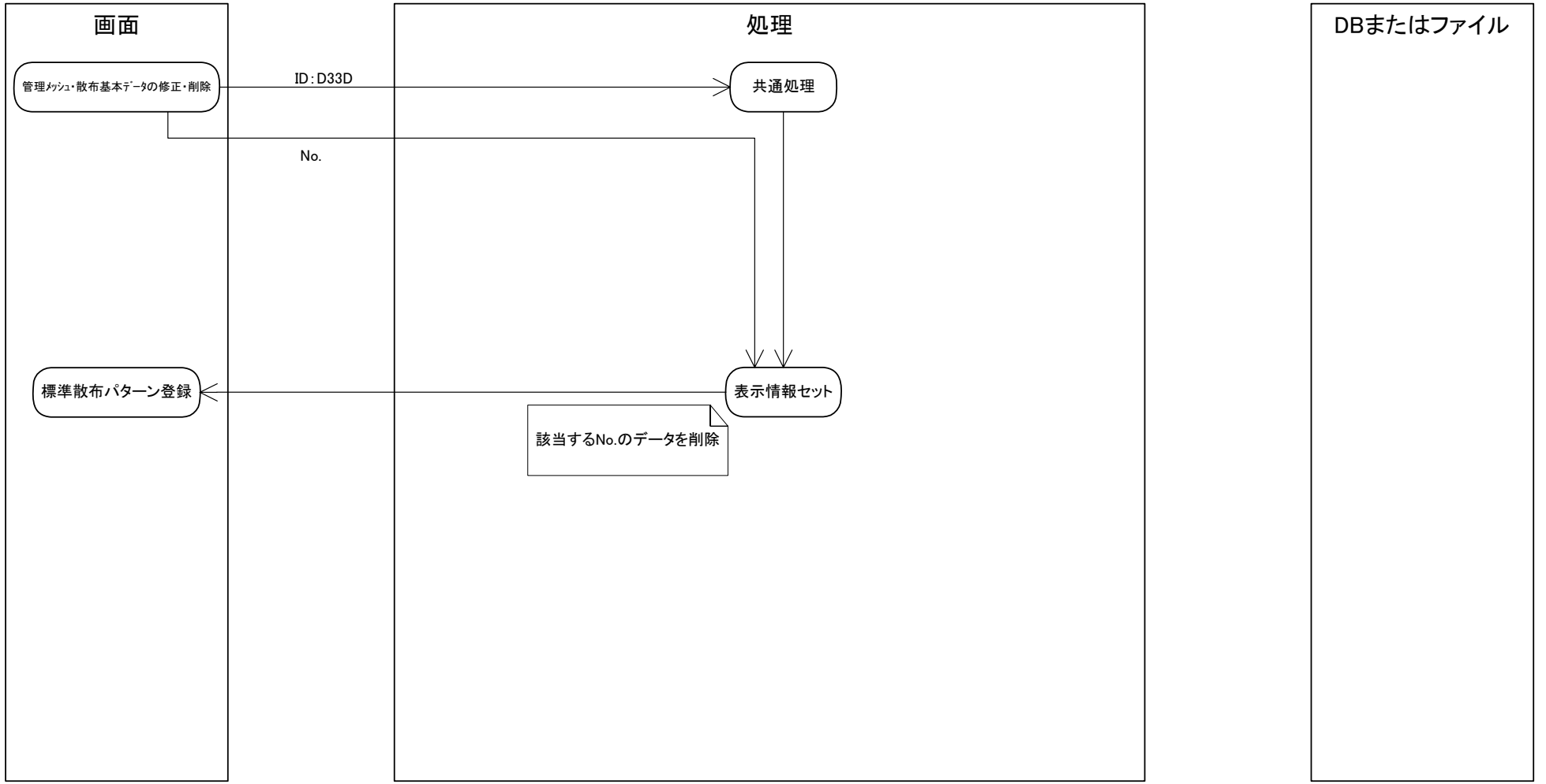


表 3-161　処理フロー

遷移元	管理メッシュ・散布基本データの修正・削除	アクション	ボタン：戻る	遷移先	標準散布パターン修正

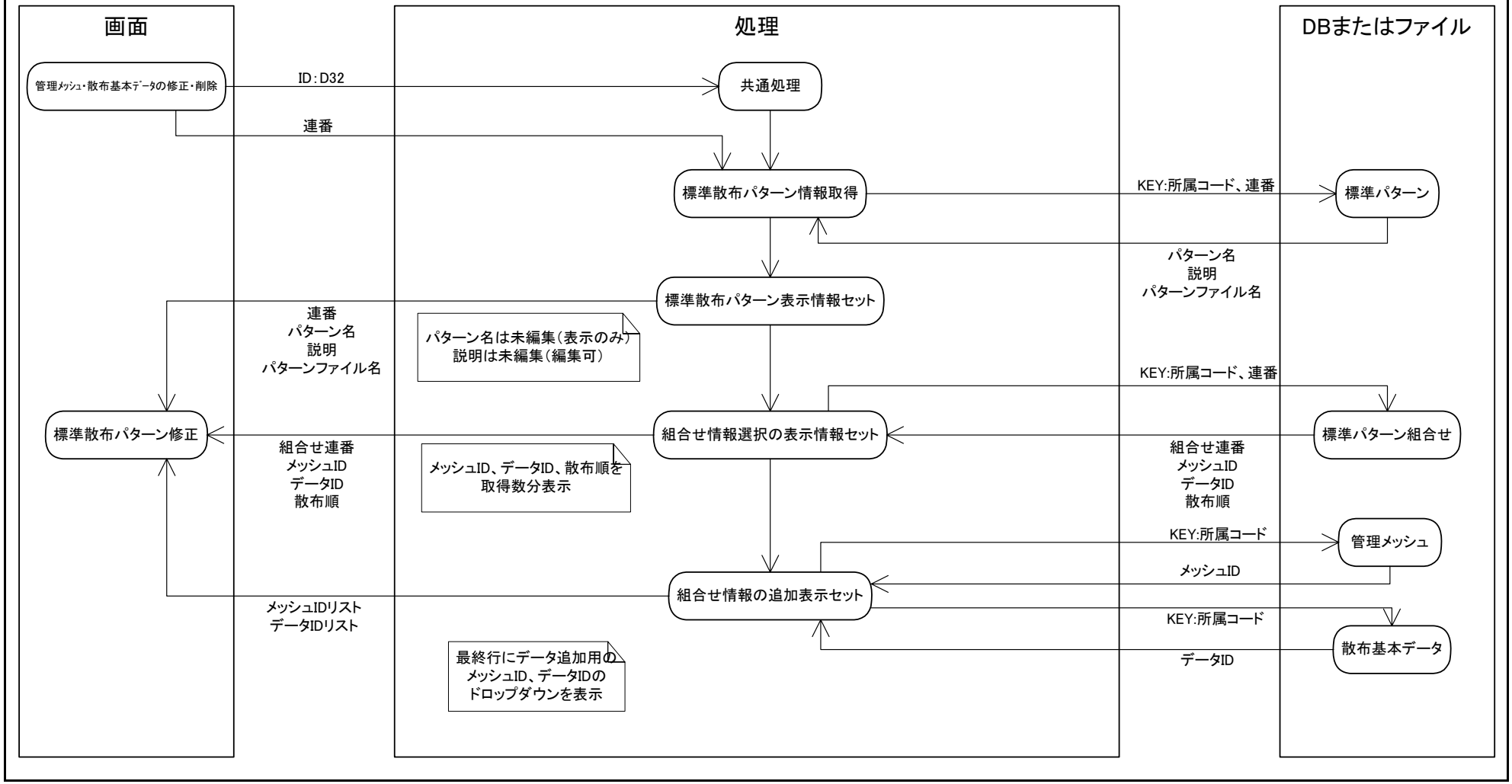


表 3-162　処理フロー

遷移元	管理メッシュ・散布基本データの修正・削除	アクション	ボタン：戻る	遷移先	標準散布パターン登録

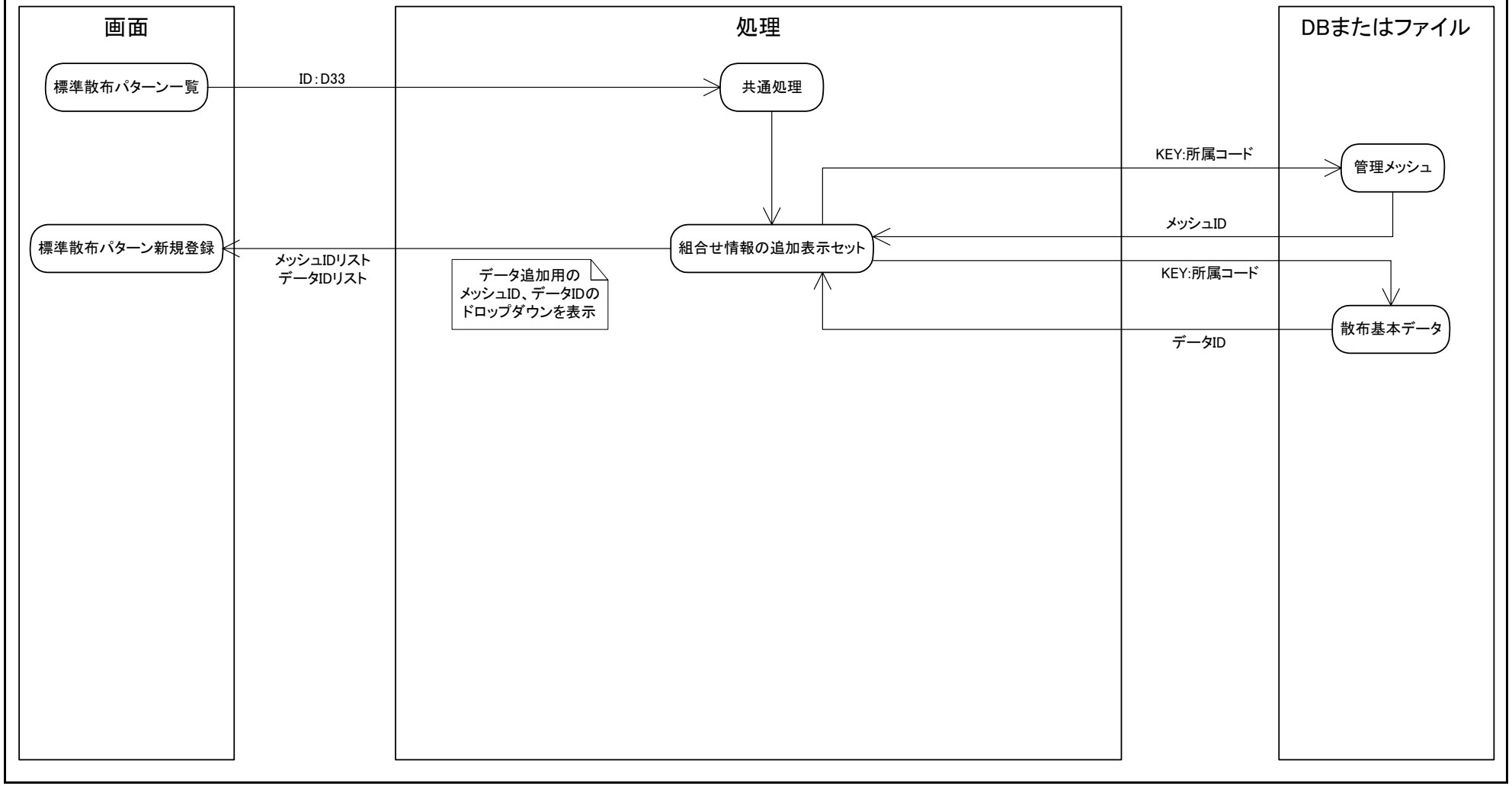


表 3-163　処理フロー

遷移元	標準散布パターン作成メニュー	アクション	リンク：標準散布パターン確認	遷移先	標準散布パターン一覧

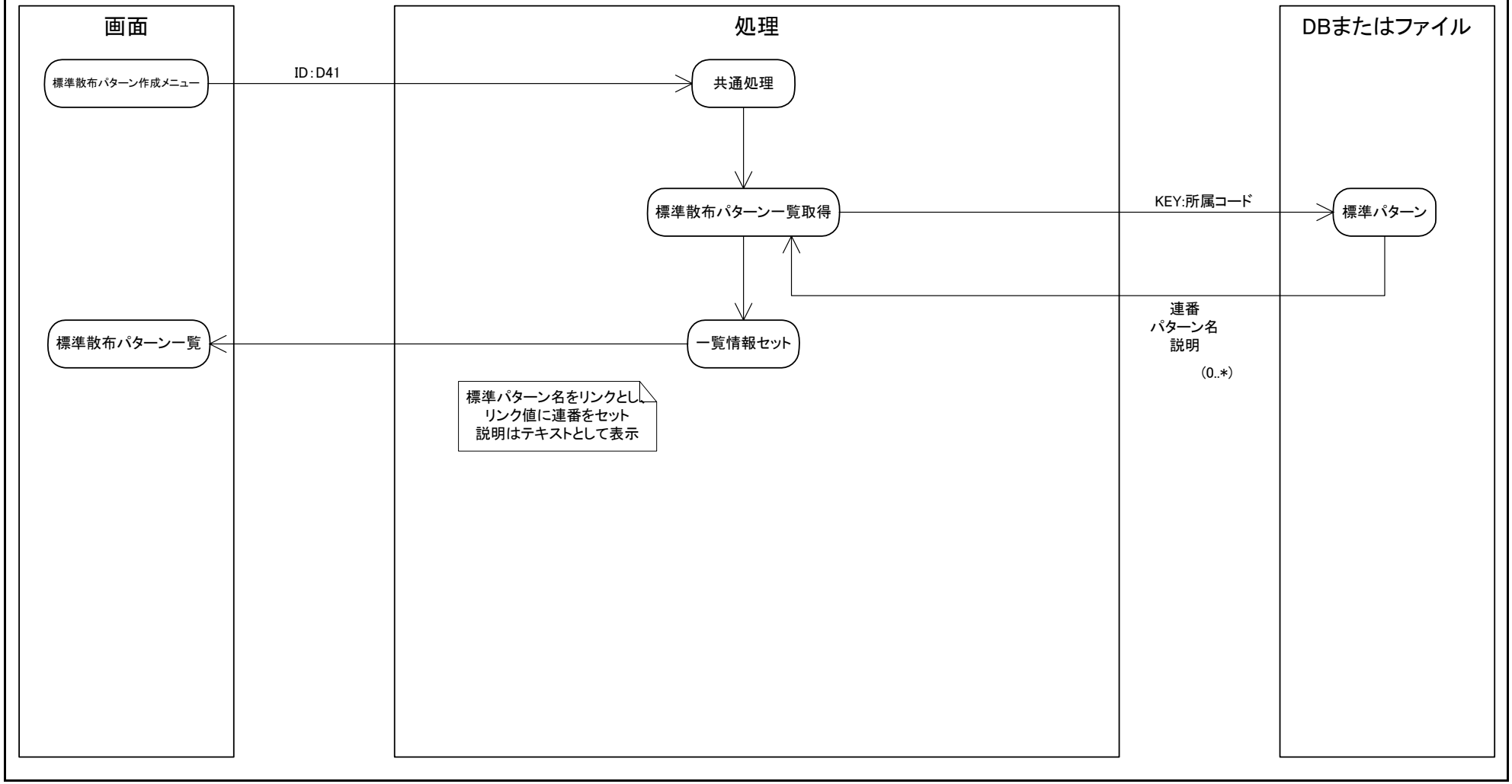


表 3-164　処理フロー

遷移元	標準散布パターン一覧	アクション	ボタン：戻る	遷移先	標準散布パターン作成メニュー

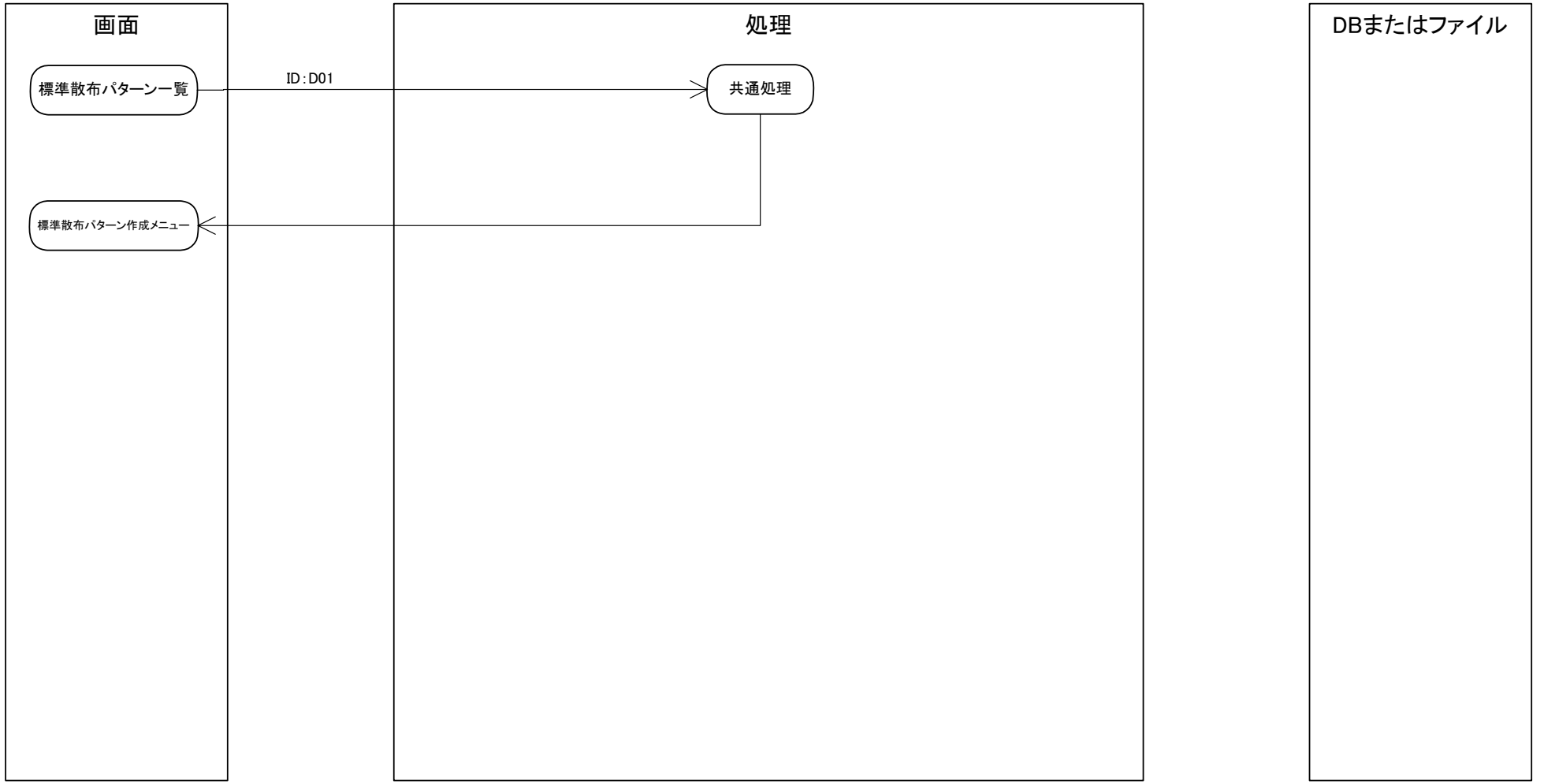


表 3-165　処理フロー

遷移元	標準散布パターン一覧	アクション	リンク：標準散布パターン名	遷移先	標準散布パターン確認

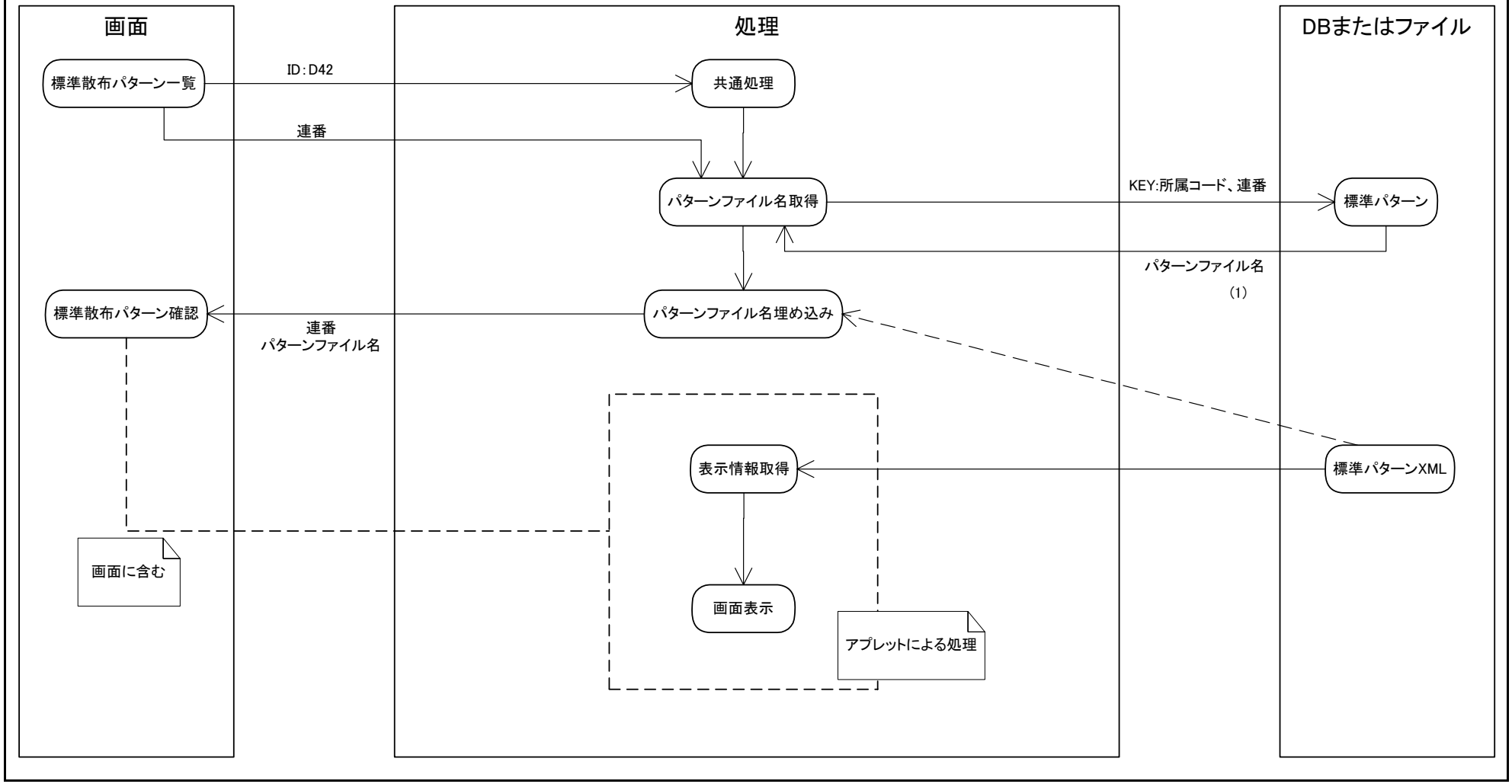


表 3-166　処理フロー

遷移元	標準散布パターン確認	アクション	ボタン：戻る	遷移先	標準散布パターン一覧

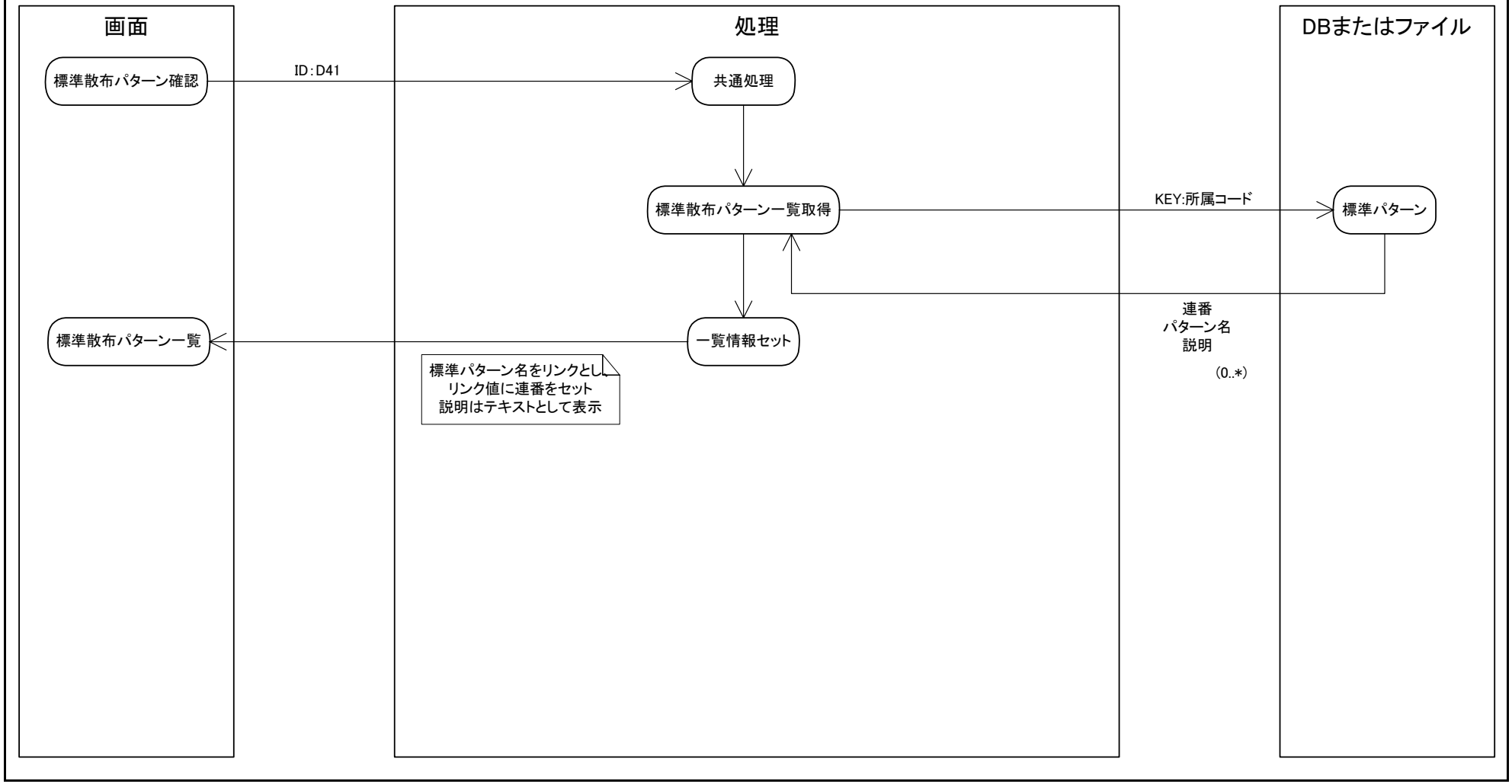


表 3-167　処理フロー

遷移元	標準散布パターン確認	アクション	ボタン：メニューへ	遷移先	トップメニュー

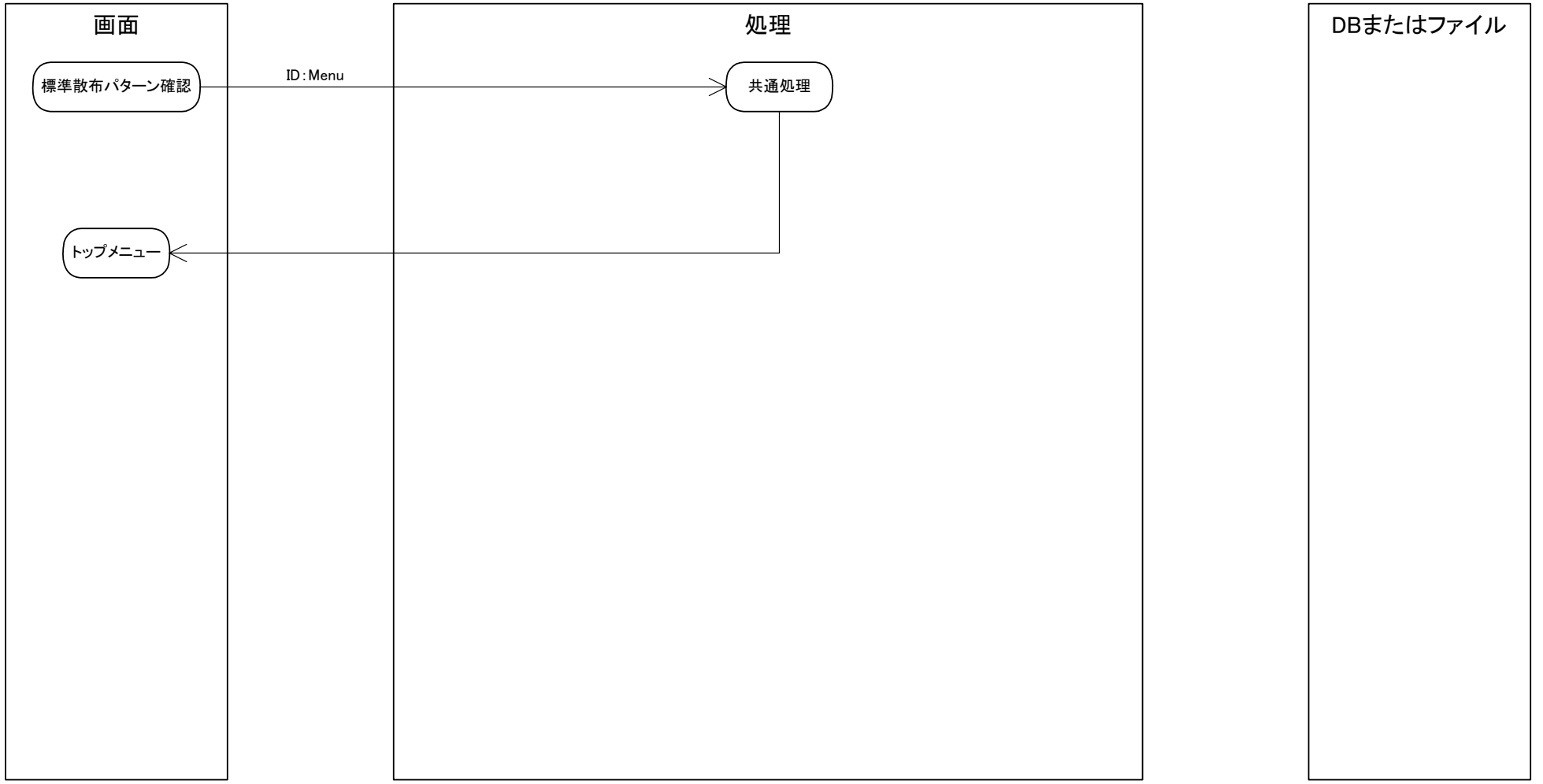


表 3-168　処理フロー

遷移元	トップメニュー	アクション	ボタン：作業者名リスト作成	遷移先	作業者名リスト一覧

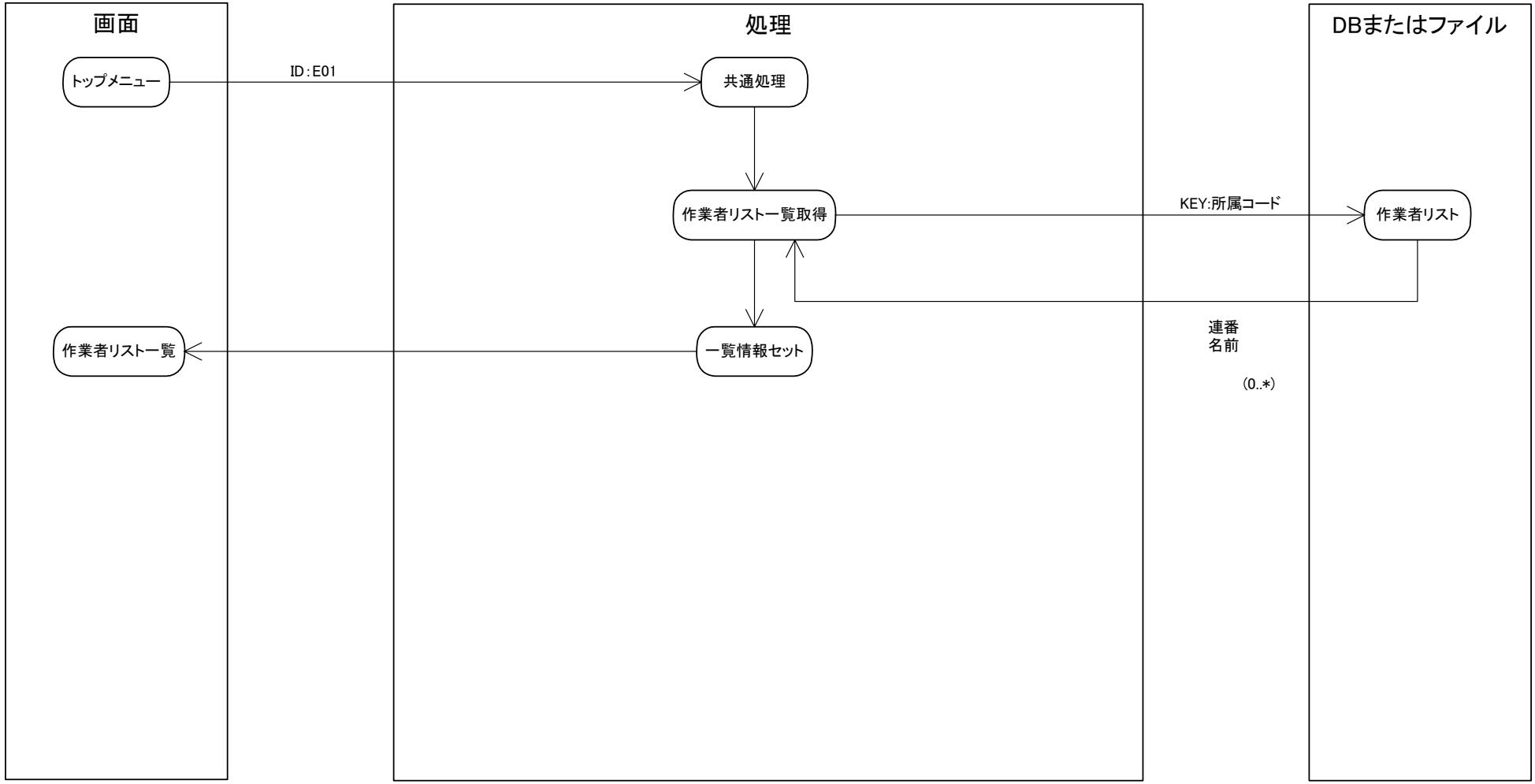


表 3-169　処理フロー

遷移元	作業者名リスト一覧	アクション	ボタン：修正・削除	遷移先	作業者名修正

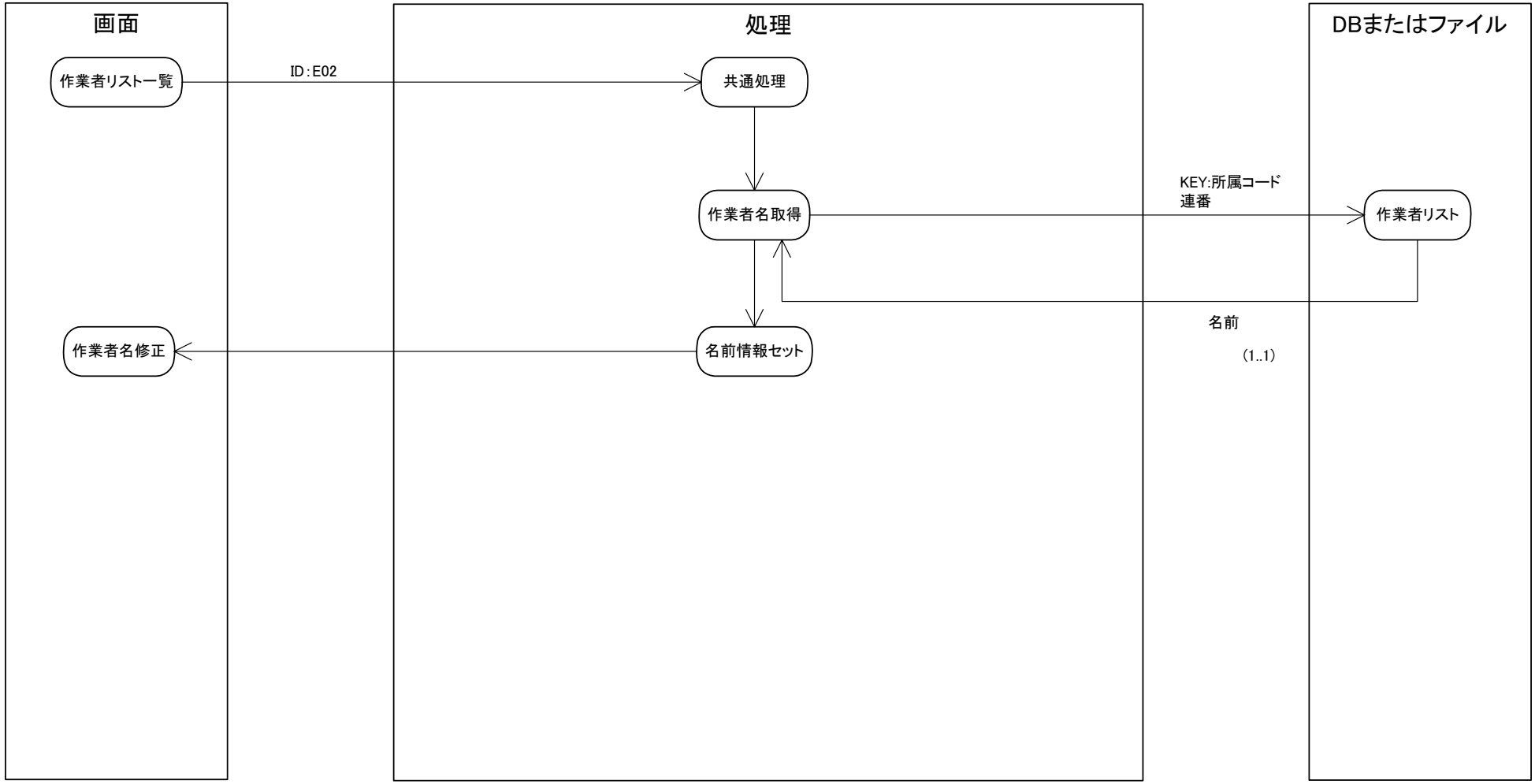


表 3-170　処理フロー

遷移元	標準ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	アクション	マウスクリック（その１）	遷移先	標準ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ

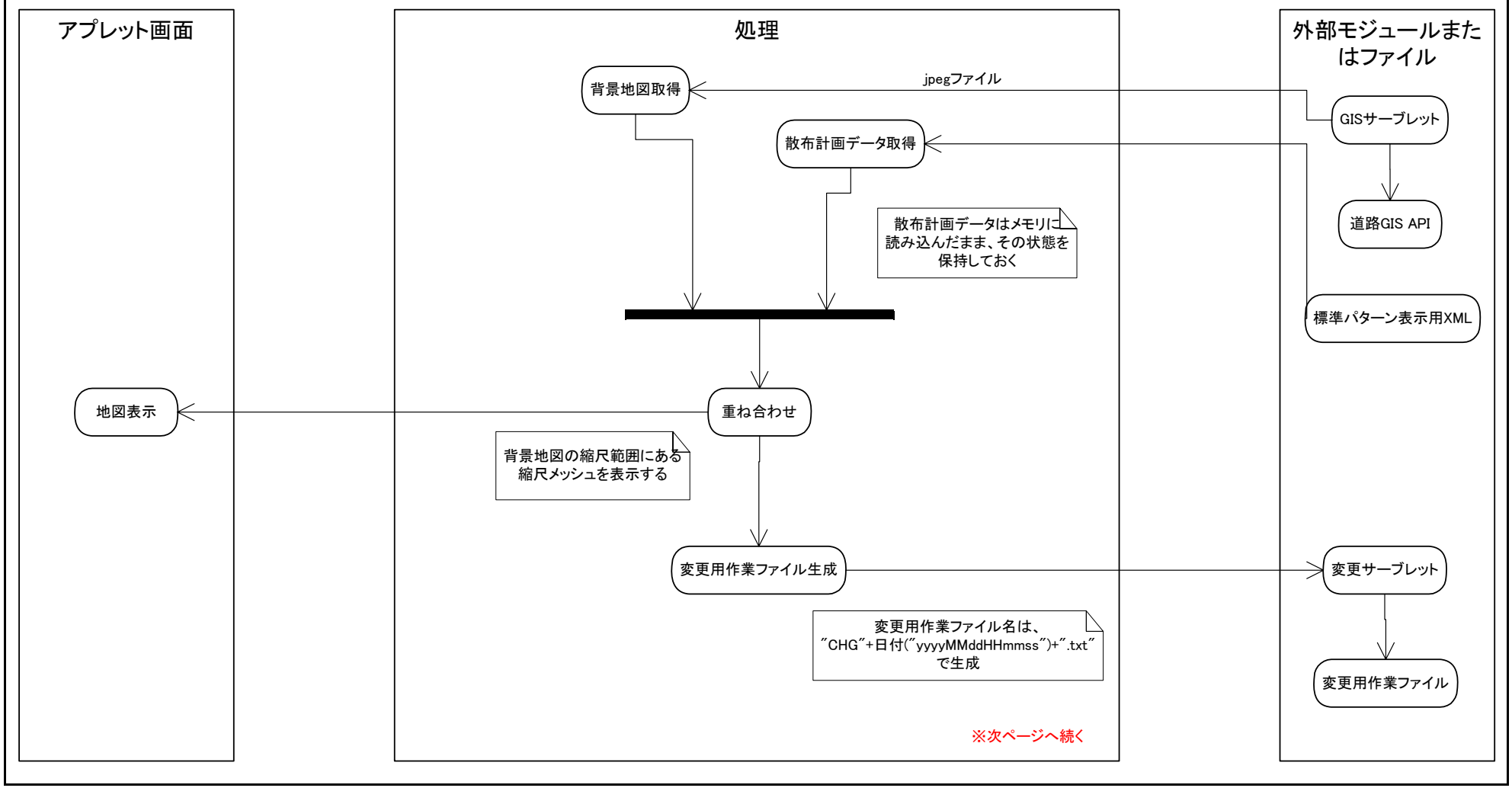


表 3-171　処理フロー

遷移元	標準ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	アクション	マウスクリック（その２）	遷移先	標準ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ

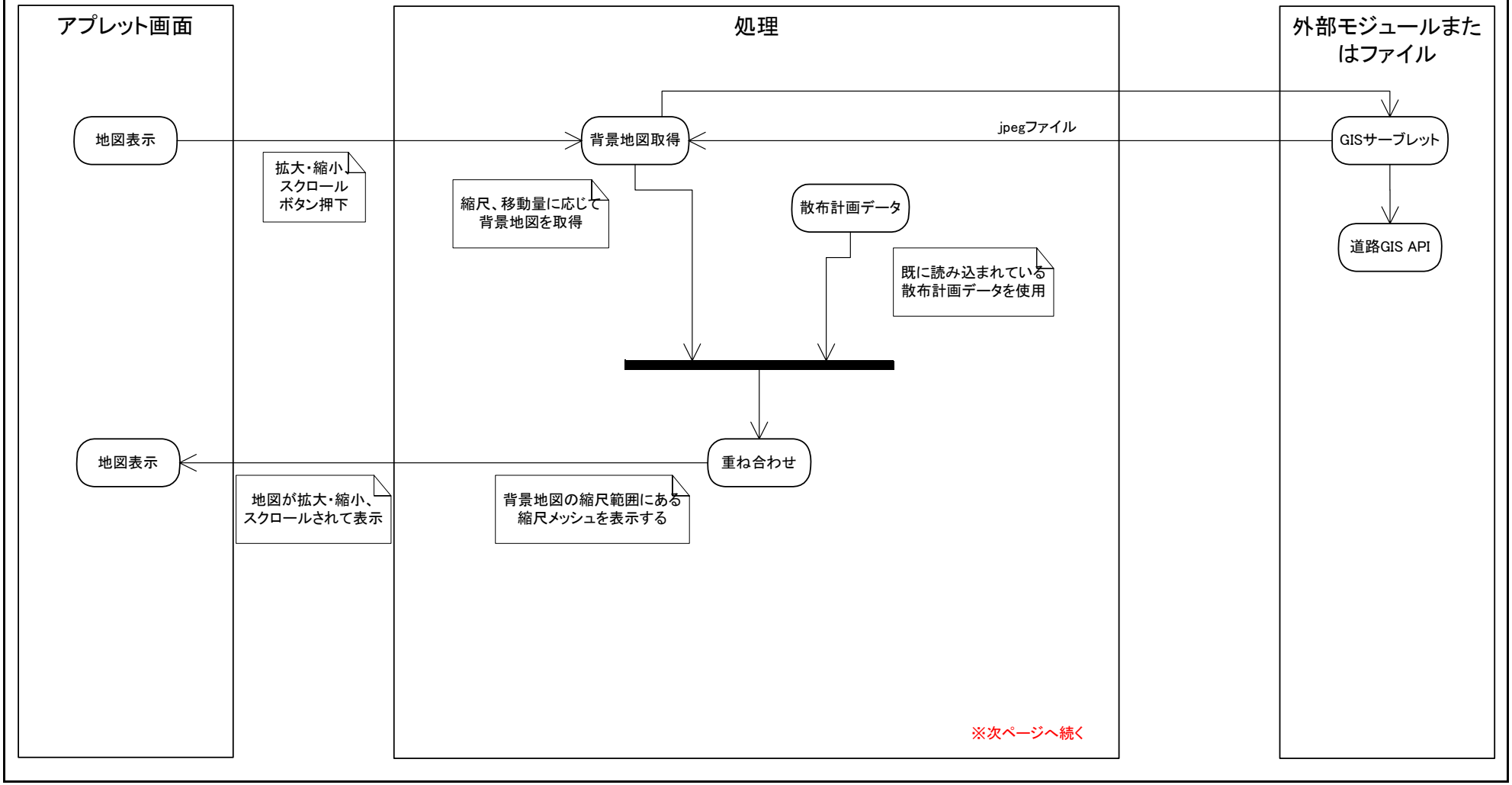


表 3-172　処理フロー

遷移元	標準ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	アクション	マウスクリック（その３）	遷移先	標準ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ

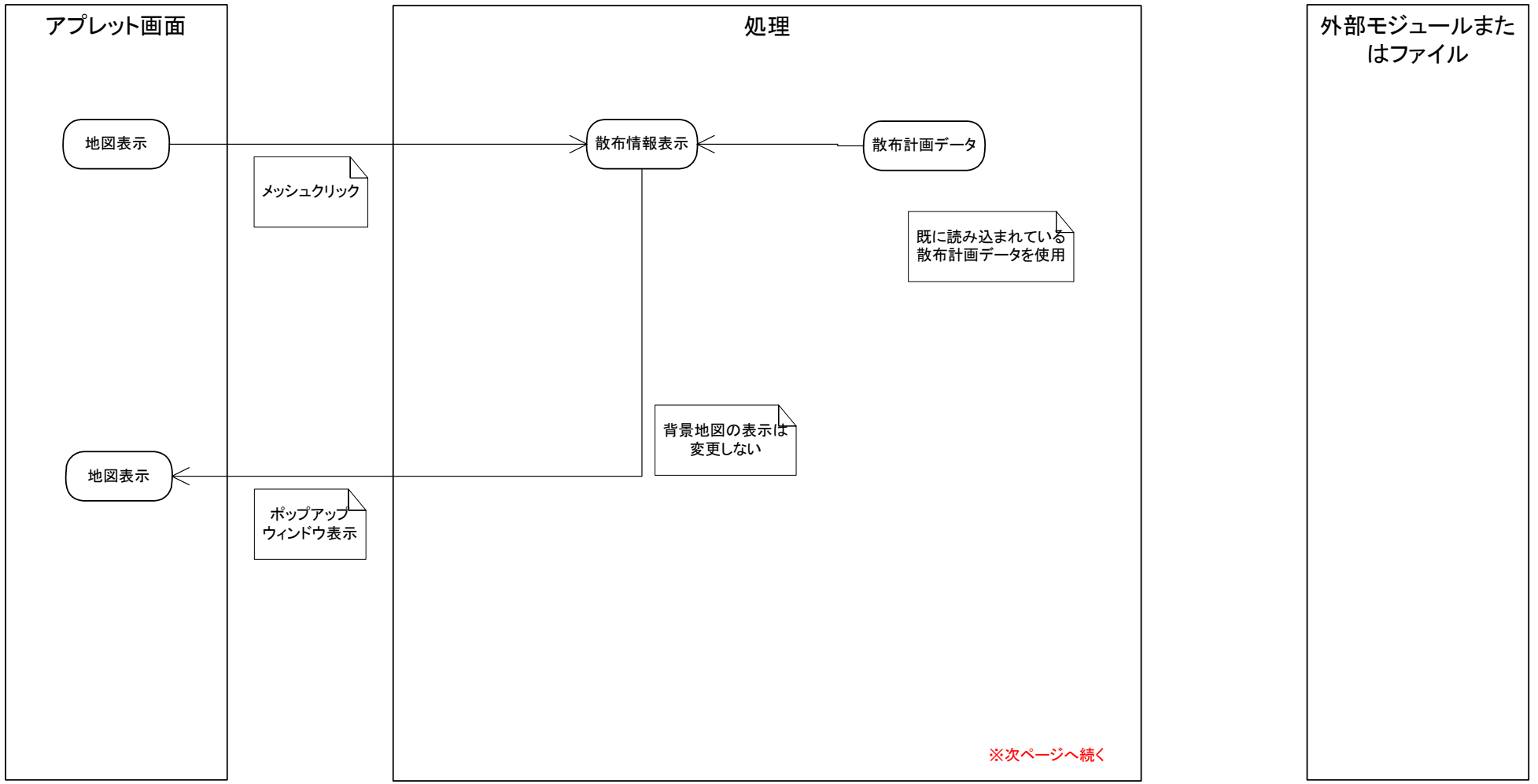


表 3-173　処理フロー

遷移元	標準ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	アクション	マウスクリック（その４）	遷移先	標準ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ

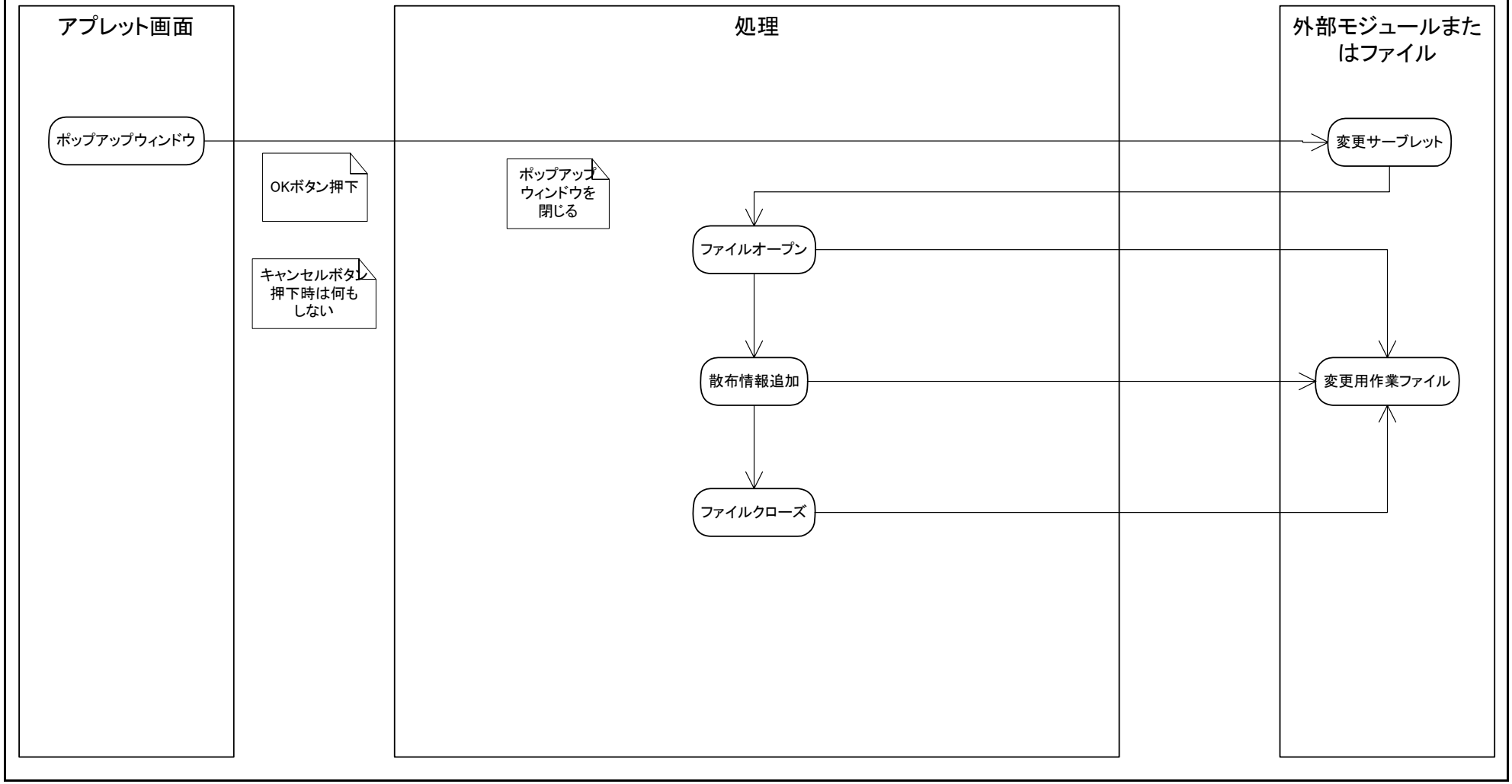


表 3-174　処理フロー

遷移元	散布履歴表示ｱﾌﾟﾚｯﾄ	アクション	マウスクリック（その１）	遷移先	散布履歴表示ｱﾌﾟﾚｯﾄ

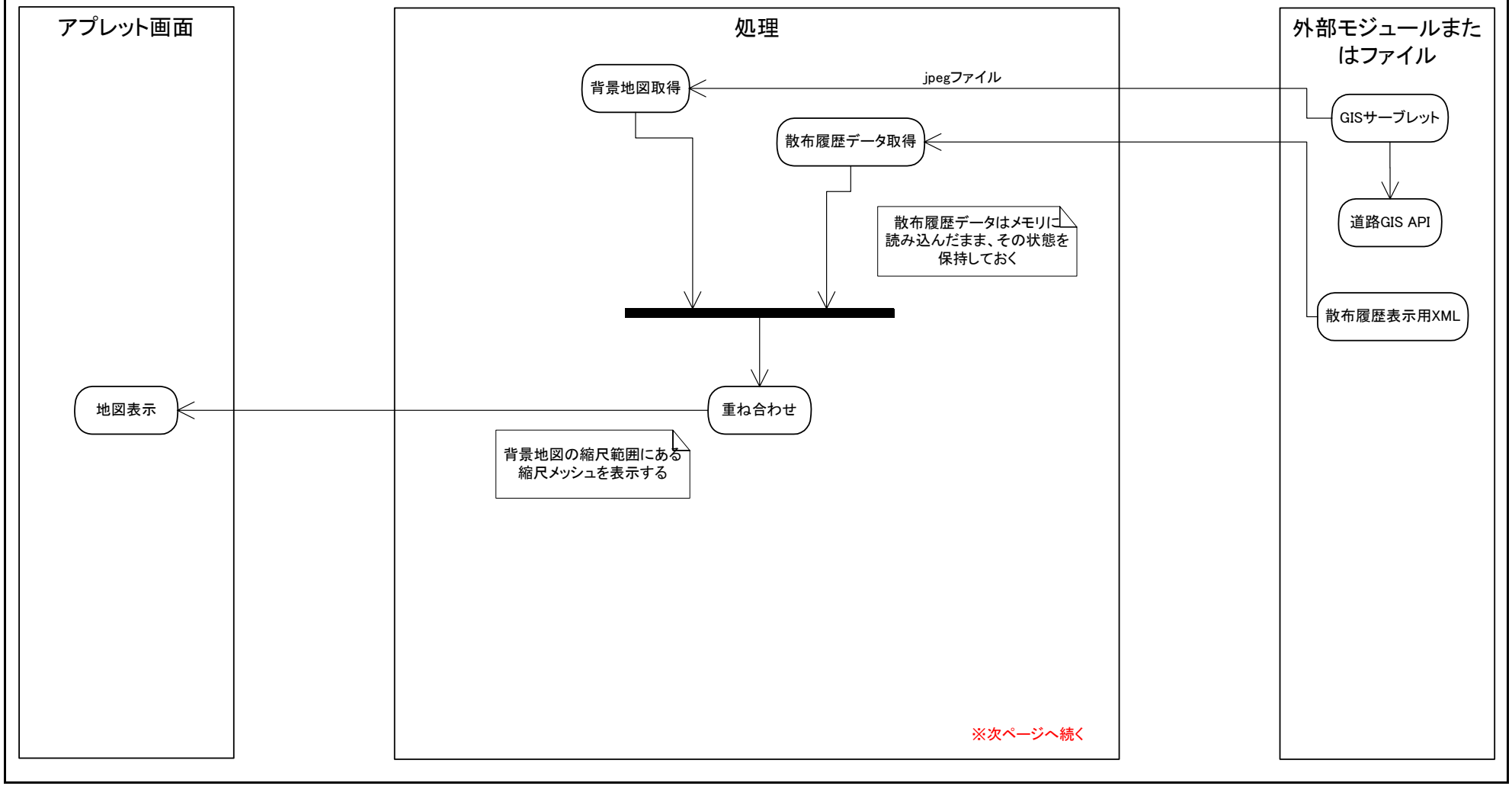


表 3-175　処理フロー

遷移元	散布履歴表示ｱﾌﾟﾚｯﾄ	アクション	マウスクリック（その２）	遷移先	散布履歴表示ｱﾌﾟﾚｯﾄ

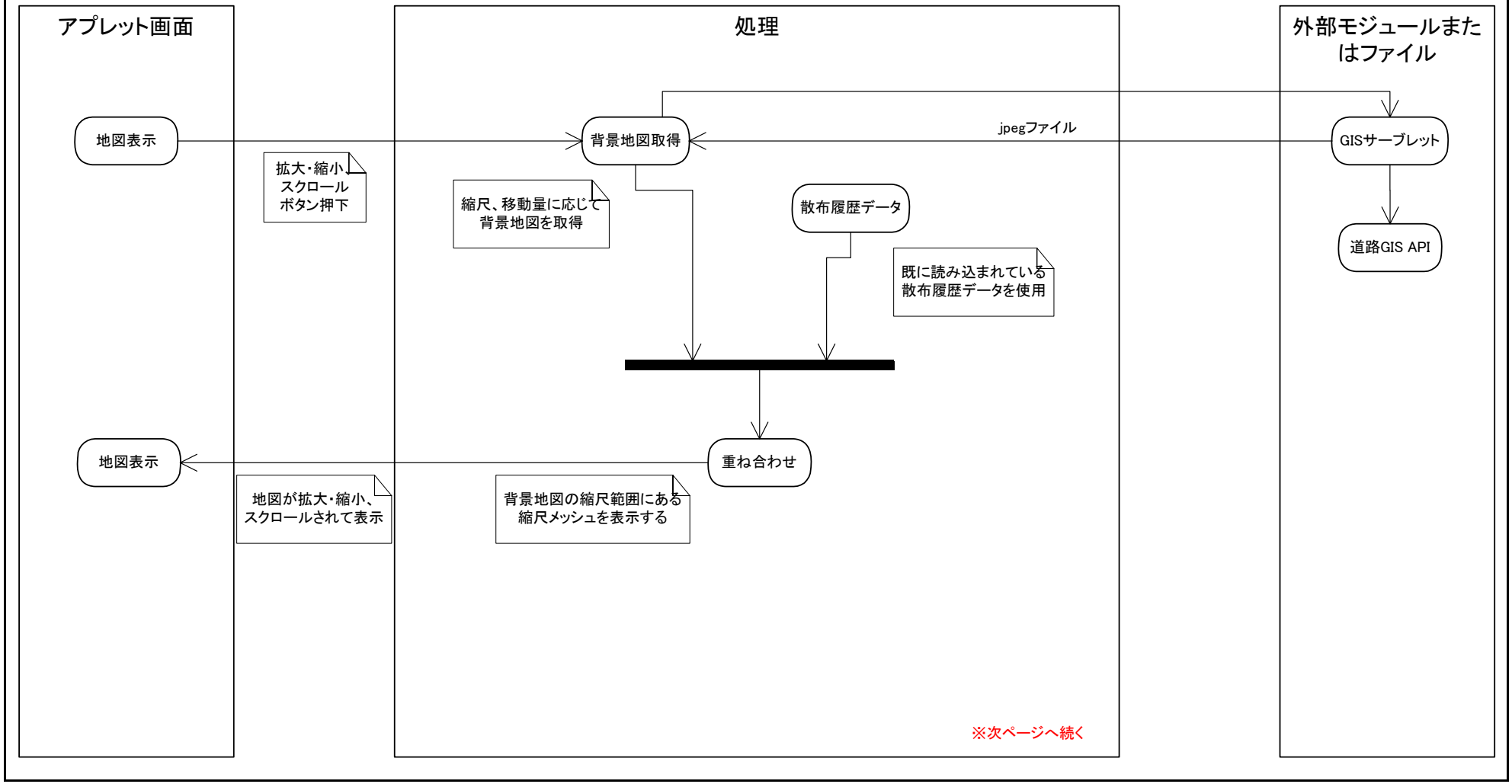


表 3-176　処理フロー

遷移元	散布履歴表示ｱﾌﾟﾚｯﾄ	アクション	マウスクリック（その３）	遷移先	散布履歴表示ｱﾌﾟﾚｯﾄ

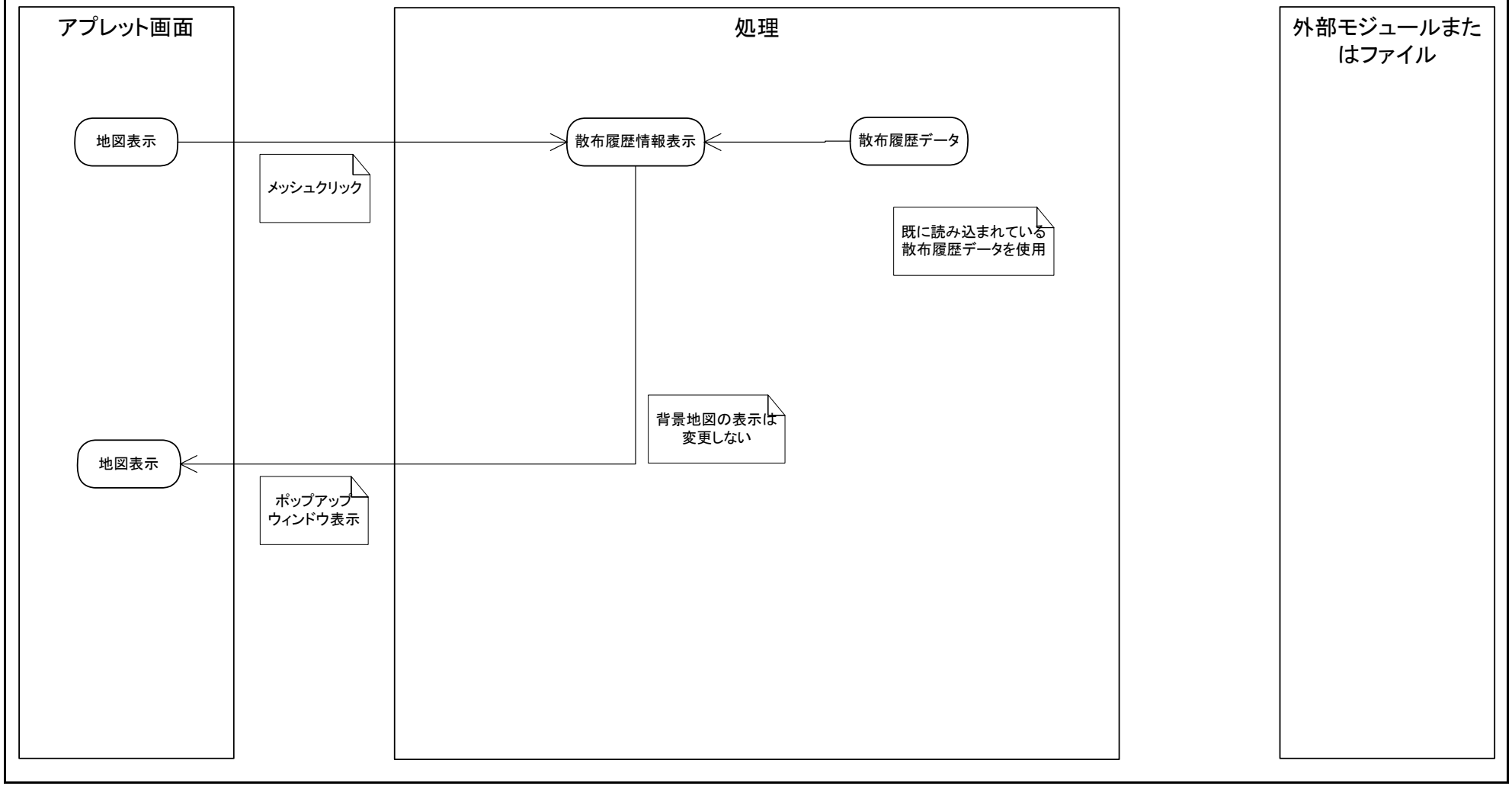


表 3-177　処理フロー

遷移元	標準散布ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	アクション	マウスクリック（その１）	遷移先	標準散布ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ

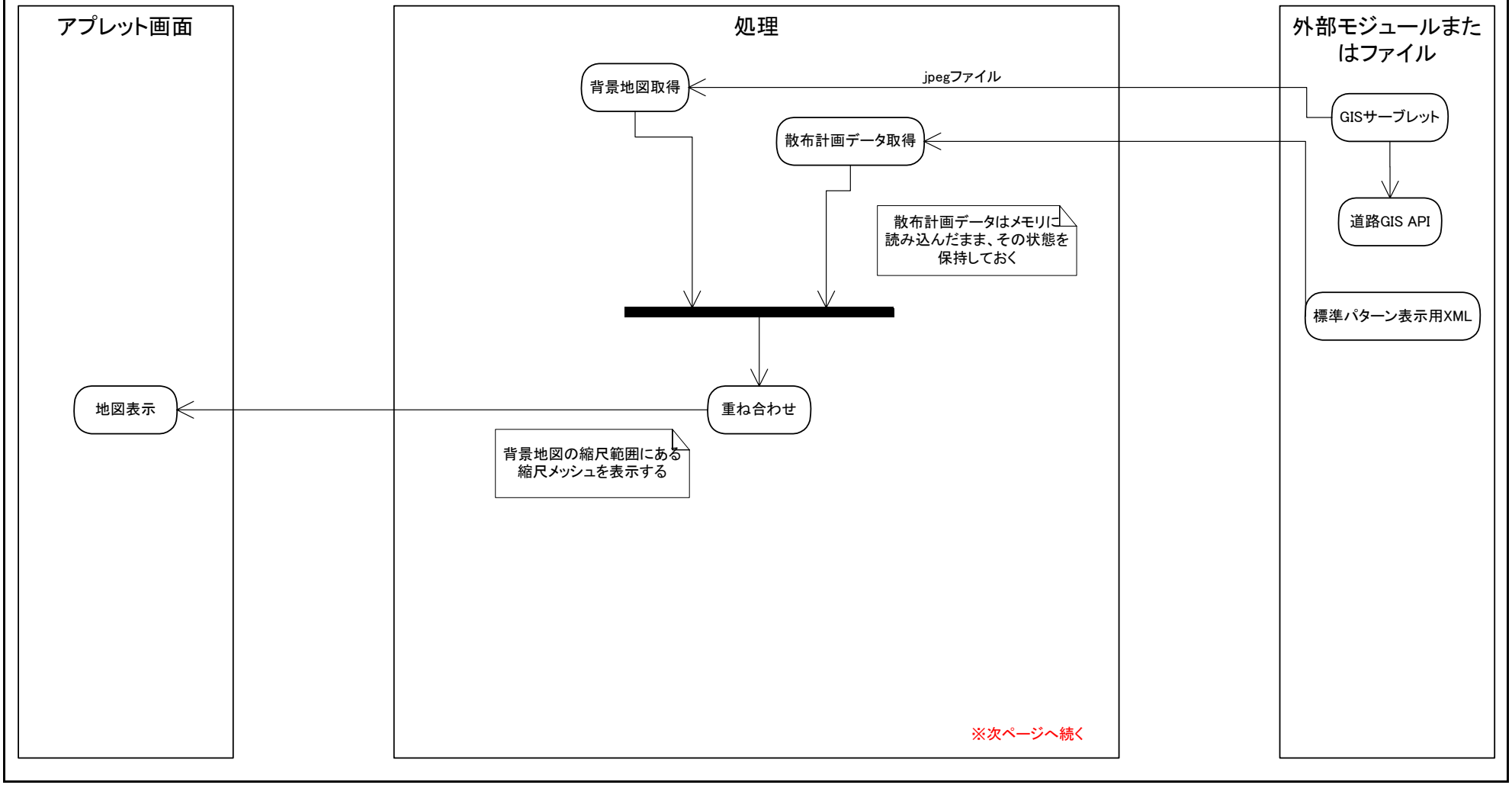


表 3-178　処理フロー

遷移元	標準散布ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	アクション	マウスクリック（その２）	遷移先	標準散布ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ

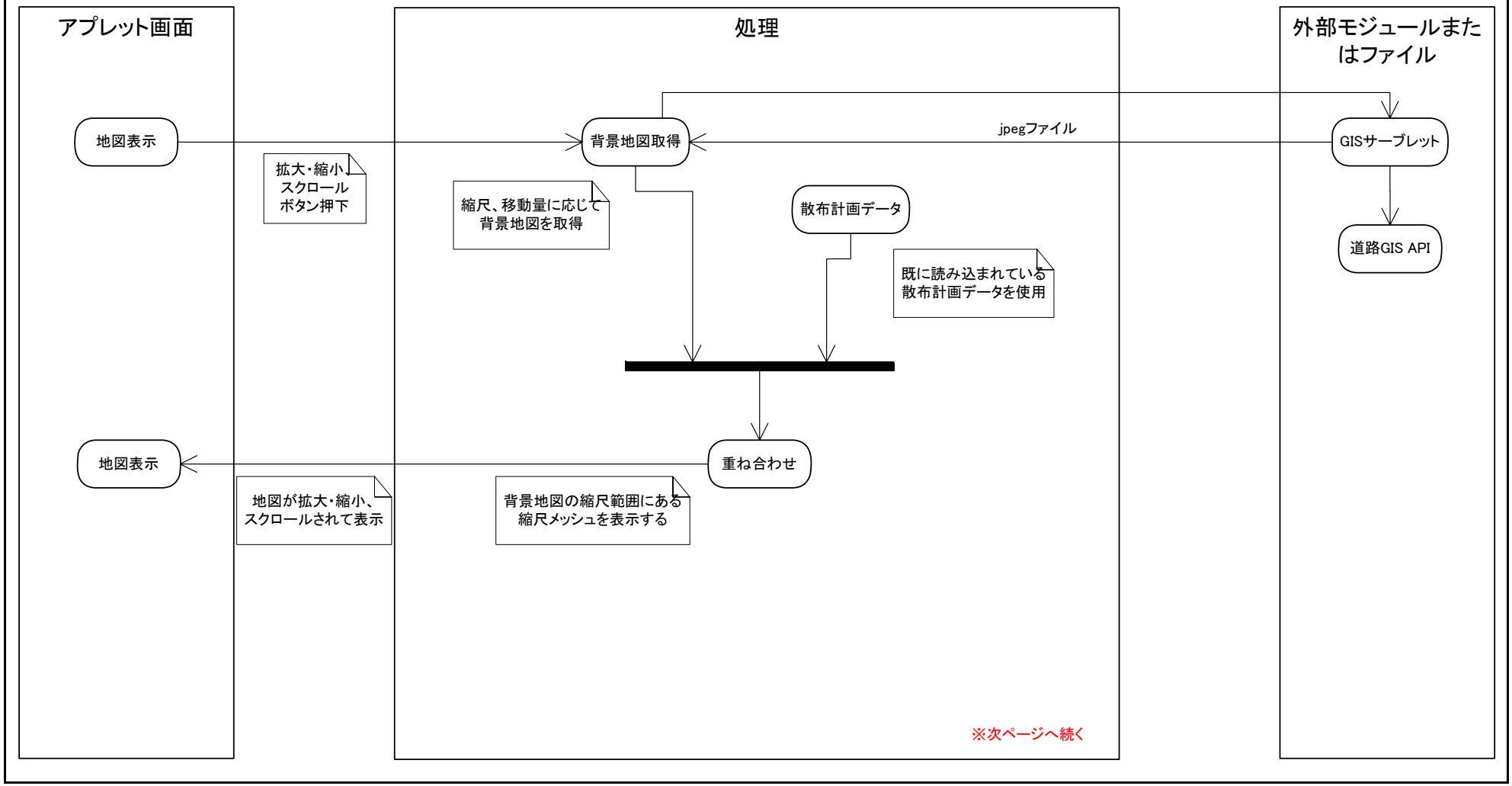
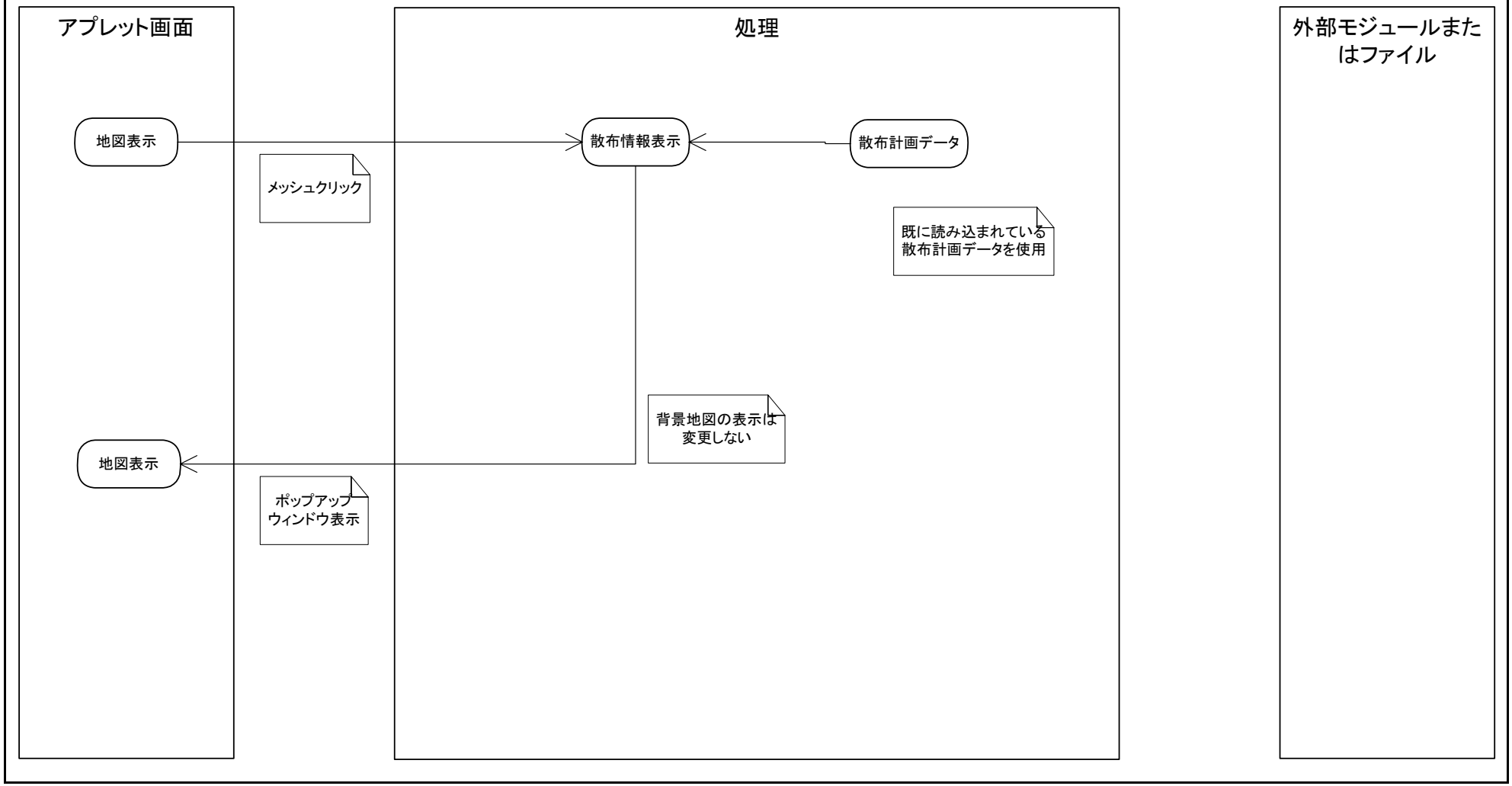


表 3-179　処理フロー

遷移元	標準散布ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ	アクション	マウスクリック（その３）	遷移先	標準散布ﾊﾟﾀｰﾝ確認ｱﾌﾟﾚｯﾄ



(7)　入力チェック仕様

入力チェック仕様を以下に示す。

表 3-180　入力チェック

画面名	散布作業記録ファイル選択			
項目名	ｸﾗｲｱﾝﾄ	ｻｰﾊﾞ	チェック内容	エラーメッセージ
参照ファイル名	○		必須入力	ファイルが選択されていません。ファイルを選択してください。
参照ファイル名		○	存在しないファイルもしくは空ファイル	ファイルが存在しないか、もしくはデータがありません。
参照ファイル名		○	データ部がないとき	データがありません。
参照ファイル名		○	散布作業記録フォーマット以外	指定されたファイルを読込めません。

表 3-181　入力チェック

画面名	巡回記録ファイル選択			
項目名	ｸﾗｲｱﾝﾄ	ｻｰﾊﾞ	チェック内容	エラーメッセージ
参照ファイル名	○		必須入力	ファイルが選択されていません。ファイルを選択してください。
参照ファイル名		○	存在しないファイルもしくは空ファイル	ファイルが存在しないか、もしくはデータがありません。
参照ファイル名		○	データ部がないとき	データがありません。
参照ファイル名		○	散布作業記録フォーマット以外	指定されたファイルを読込めません。

表 3-182　入力チェック

画面名	管理メッシュ新規登録			
項目名	ｸﾗｲｱﾝﾄ	ｻｰﾊﾞ	チェック内容	エラーメッセージ
メッシュ ID	○		必須入力	メッシュ ID が入力されていません。
路線名	○		必須入力	路線名が入力されていません。
開始 KP	○		必須入力	開始 KP が入力されていません。
終了 KP	○		必須入力	終了 KP が入力されていません。
メッシュ ID	○		最大桁(32 桁)	（入力制限）
路線名	○		最大桁(10 桁)	（入力制限）
開始 KP	○		数値	開始 KP は数値を入力してください。
終了 KP	○		数値	終了 KP は数値を入力してください。
メッシュ ID		○	重複チェック	このメッシュ ID は既に登録済みです。違うメッシュ ID を入力してください。
開始 KP		○	存在チェック	この車線にはこの開始 KP が存在しません。正しい開始 KP を入力してください。
終了 KP		○	存在チェック	この車線にはこの終了 KP が存在しません。正しい終了 KP を入力してください。
開始 KP／終了 KP	○		上り車線	開始 KP＞＝終了 KP の範囲で入力してください。
	○		下り車線	開始 KP＜＝終了 KP の範囲で入力してください。

表 3-183　入力チェック

画面名	管理メッシュ修正			
項目名	ｸﾗｲｱﾝﾄ	ｻｰﾊﾞ	チェック内容	エラーメッセージ
路線名	○		必須入力	路線名が入力されていません。
開始 KP	○		必須入力	開始 KP が入力されていません。
終了 KP	○		必須入力	終了 KP が入力されていません。
路線名	○		最大桁(10 桁)	（入力制限）
開始 KP	○		数値	開始 KP は数値を入力してください。
終了 KP	○		数値	終了 KP は数値を入力してください。
開始 KP		○	存在チェック	この車線にはこの開始 KP が存在しません。正しい開始 KP を入力してください。
終了 KP		○	存在チェック	この車線にはこの終了 KP が存在しません。正しい終了 KP を入力してください。
開始 KP／終了 KP	○		上り車線	開始 KP＞＝終了 KP の範囲で入力してください。
	○		下り車線	開始 KP＜＝終了 KP の範囲で入力してください。
削除		○	参照整合性	標準パターンで選択中のため、削除することができません。

表 3-184　入力チェック

画面名	散布基本データ新規登録			
項目名	ｸﾗｲｱﾝﾄ	ｻｰﾊﾞ	チェック内容	エラーメッセージ
データ ID	○		必須入力	データ ID が入力されていません。
データ ID	○		最大桁(32 桁)	（入力制限）
データ ID		○	重複チェック	このデータ ID は既に登録済みです。違うデータ ID を入力してください。

表 3-185　入力チェック

画面名	散布基本データ修正			
項目名	ｸﾗｲｱﾝﾄ	ｻｰﾊﾞ	チェック内容	エラーメッセージ
削除		○	参照整合性	標準パターンで選択中のため、削除することができません。

表 3-186　入力チェック

画面名	標準散布パターン新規登録			
項目名	ｸﾗｲｱﾝﾄ	ｻｰﾊﾞ	チェック内容	エラーメッセージ
標準散布パターン名	○		必須入力	標準散布パターン名が入力されていません。
説明	○		必須入力	説明が入力されていません。
標準散布パターン名	○		最大桁(40 桁)	（入力制限）
説明	○		最大桁(300 桁)	（入力制限）
管理メッシュ・散布基本データ	○		データが１つ以上あるか	管理メッシュ・散布基本データを作成してください。
標準散布パターン名		○	先頭文字が『.』は不可	標準散布パターン名の先頭に『.』は入力できません。
標準散布パターン名		○	ファイル名として無効な文字はエラー	標準散布パターン名に『 ¥¥ / : ; * ? ¥" < > | 』は入力できません。

表 3-187　入力チェック

画面名	標準散布パターン修正			
項目名	ｸﾗｲｱﾝﾄ	ｻｰﾊﾞ	チェック内容	エラーメッセージ
説明	○		必須入力	説明が入力されていません。
説明	○		最大桁(300 桁)	（入力制限）
管理メッシュ･散布基本データ	○		データが１つ以上あるか	管理メッシュ・散布基本データを作成してください。

3.3. 計画支援システムの実装設計

(1) 適用範囲

本節は、計画支援システムにおける実装設計についてのみ記載するものとする。

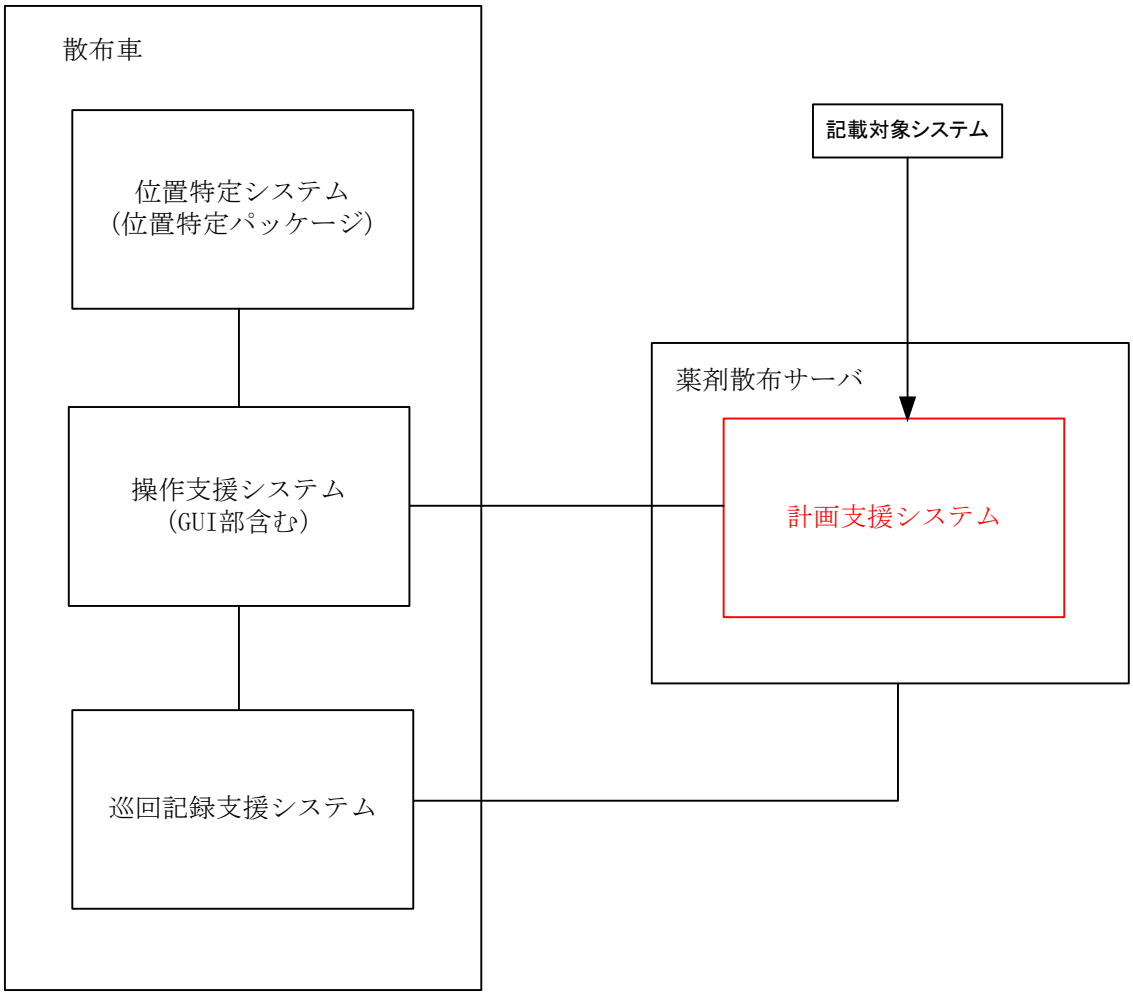


図 3-44　記載範囲

(2) ソフトウェア構成

ソフトウェアの基本構成を以下に示す。

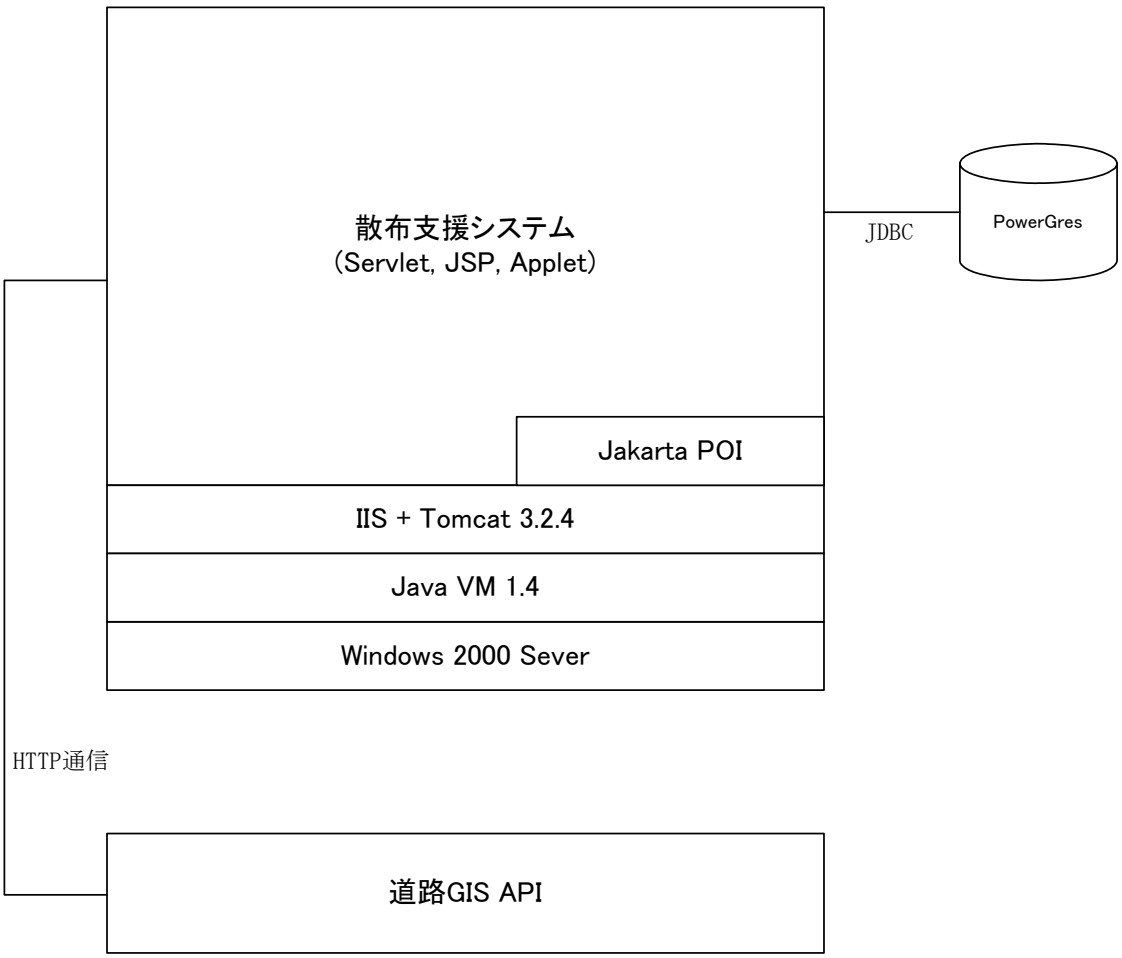


図 2-100　ソフトウェア構成

(3) モジュール設計

a） パッケージ一覧

■ １　サーブレット

パッケージ	
Snowmelt	共通サーブレット
Snowmelt.common	共通処理
Snowmelt. edit	標準パターン編集
snowmelt. entry	散布履歴、巡回記録登録
snowmelt. history	散布履歴、巡回記録表示
snowmelt. pattern	標準パターン出力
snowmelt. workman	作業者名編集

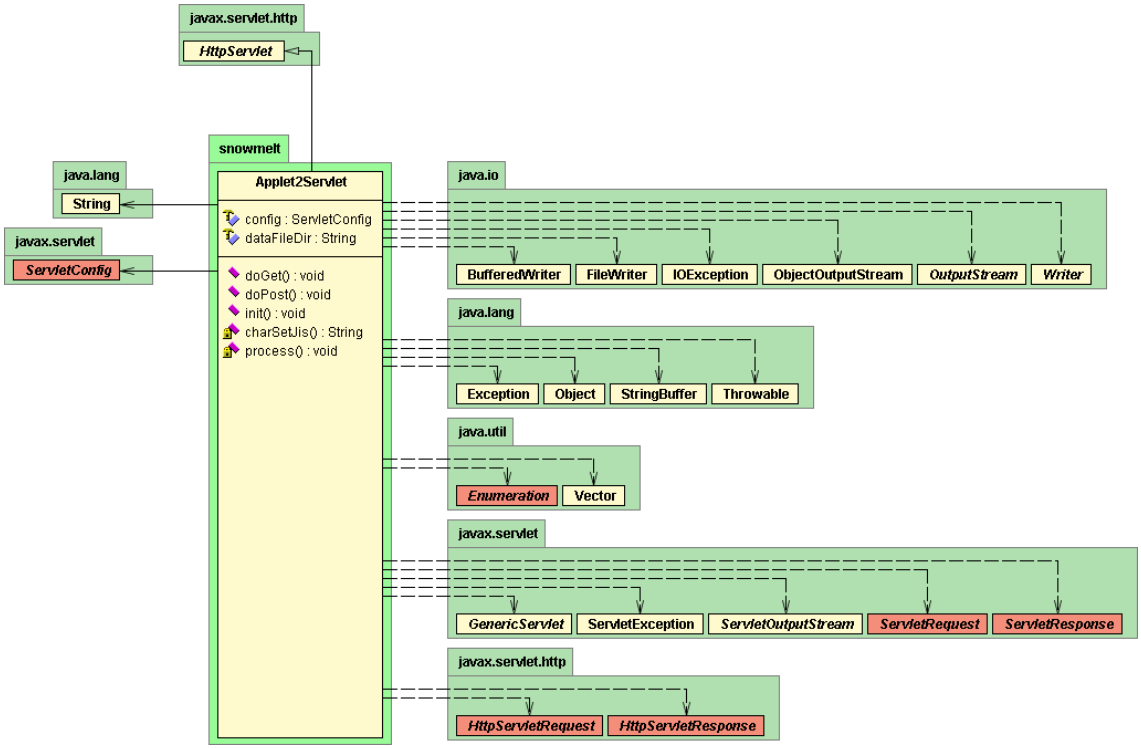
■ ２　アップレット

パッケージ	
common	共通処理
pattern	標準パターン出力
sprinkle	散布履歴表示
stdpattern	標準パターン編集

b） パッケージ snowmelt

クラスの概要	
Applet2Servlet	アプレットからのデータを受信してファイルに書き出すサーブレット
CallAPI	道路 GIS API を呼び出すためのクラス
DLServlet	ファイルダウンロードサーブレット
RgisapiJoinServlet	アプレットと道路 GIS_API サーブレットの受け渡しを行うサーブレット
SMCertifyServlet	ユーザー認証 サーブレット
SMServlet	メインサーブレット

■ 1　クラス Applet2Servlet



java.lang.Object

  |

  +--javax.servlet.GenericServlet

        |

        +--javax.servlet.http.HttpServlet

              |

              +--**snowmelt.Applet2Servlet**

すべての実装インタフェース:

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

---

public class **Applet2Servlet**

extends javax.servlet.http.HttpServlet

アプレットからのデータを受信してファイルに書き出すサーブレット

フィールドの概要

（パッケージプライベート） javax.servlet.ServletConfig	**config** サーブレット設定クラス

（パッケージプライベート） java.lang.String	**dataFileDir** データ格納フォルダ名

コンストラクタの概要

Applet2Servlet()

メソッドの概要

private java.lang.String	**charSetJis**(java.lang.String PrmData) 文字列を S-JIS に変換する
void	**doGet**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) ブラウザからの呼び出し
void	**doPost**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) ブラウザからの呼び出し
void	**init**(javax.servlet.ServletConfig conf) サーブレット init()メソッド
private void	**process**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) サーブレット

フィールドの詳細

config

javax.servlet.ServletConfig **config**

サーブレット設定クラス

---

dataFileDir

java.lang.String **dataFileDir**

データ格納フォルダ名

コンストラクタの詳細

Applet2Servlet

public **Applet2Servlet**()

メソッドの詳細

init

public void **init**(javax.servlet.ServletConfig conf)
throws javax.servlet.ServletException

サーブレット init()メソッド

オーバーライド:

クラス javax.servlet.GenericServlet 内の init

パラメータ:

conf - ServletConfig クラス

例外:

javax.servlet.ServletException -

---

doGet

```
public void doGet(javax.servlet.http.HttpServletRequest request,
                  javax.servlet.http.HttpServletResponse response)
           throws javax.servlet.ServletException,
                  java.io.IOException
```

ブラウザからの呼び出し

オーバーライド:

クラス javax.servlet.http.HttpServlet 内の doGet

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

doPost

```
public void doPost(javax.servlet.http.HttpServletRequest request,
                   javax.servlet.http.HttpServletResponse response)
            throws javax.servlet.ServletException,
                   java.io.IOException
```

ブラウザからの呼び出し

オーバーライド:

クラス javax.servlet.http.HttpServlet 内の doPost

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

process

```
private void process(javax.servlet.http.HttpServletRequest request,
```

javax.servlet.http.HttpServletResponse response)
throws javax.servlet.ServletException,
java.io.IOException

サーブレット

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

charSetJis

private java.lang.String **charSetJis**(java.lang.String PrmData)

文字列を S-JIS に変換する

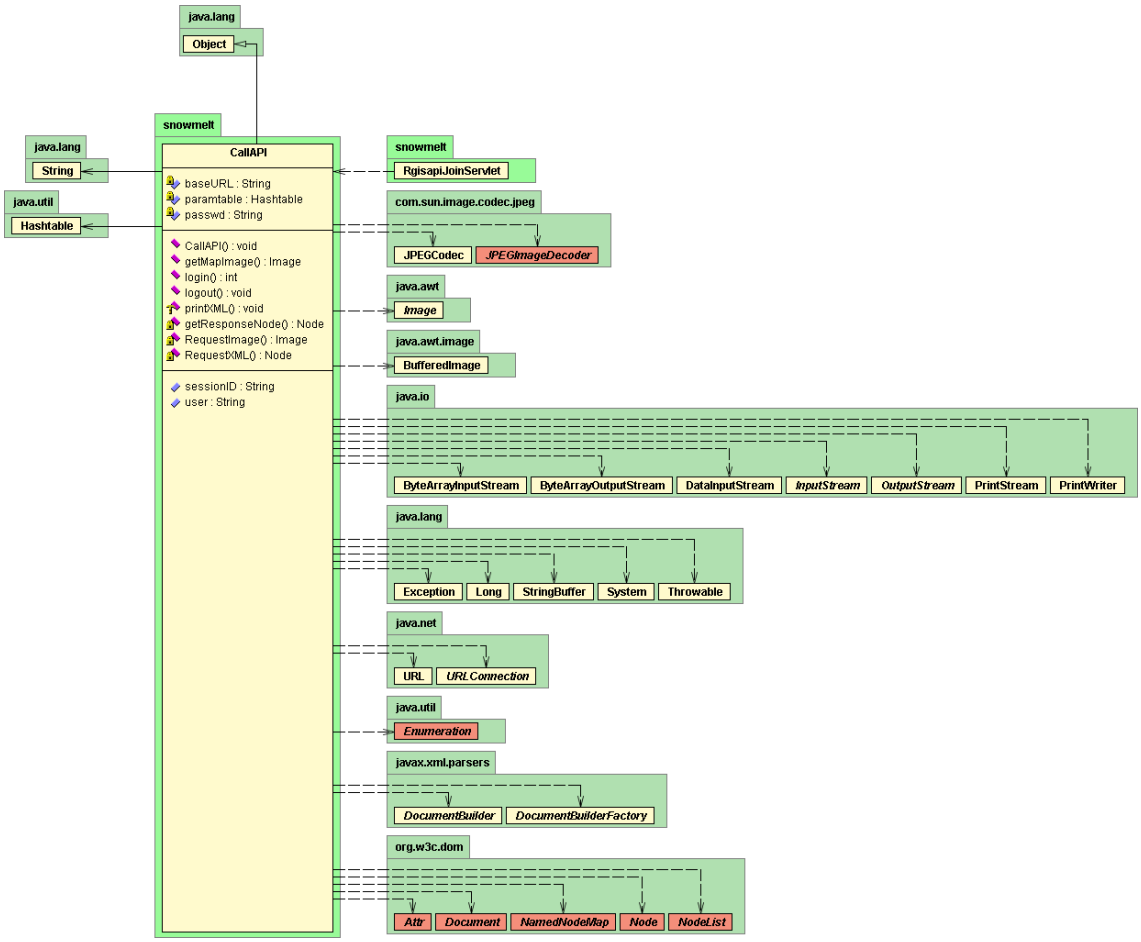
パラメータ:

PrmData - 変換前文字列

戻り値:

変換後文字列

■ 2　クラス CallAPI



java.lang.Object

　|

　+--**snowmelt.CallAPI**

---

public class **CallAPI**

extends java.lang.Object

道路 GIS API を呼び出すためのクラス

フィールドの概要

private　java.lang.String	**baseURL** 道路 GIS_API の URL
private　java.util.Hashtable	**paramtable**

	セッション管理パラメータ
private java.lang.String	**passwd** パスワード
private java.lang.String	**sessionID** セッション ID
private java.lang.String	**user** ユーザーID

コンストラクタの概要

CallAPI(java.lang.String baseurl, java.lang.String usr, java.lang.String pass) コンストラクタ

メソッドの概要

java.awt.Image	**getMapImage**(java.lang.String north, java.lang.String west, java.lang.String south, java.lang.String east, java.lang.String xsize, java.lang.String ysize, java.lang.String scale) サーバから地図イメージを取得
private snowmelt.Node	**getResponseNode**(snowmelt.Node response_node) サーバーのレスポンスの変換を行う
java.lang.String	**getSessionID**() セッション ID の取得
java.lang.String	**getUser**() ユーザーID の取得
int	**login**() セッションを開始する

void	**logout**() セッションの破棄
(パッケージプライベート) void	**printXML**(snowmelt.Node node)
private java.awt.Image	**RequestImage**(java.lang.String api, java.lang.String arg) サーバーAPI 呼び出し イメージデータを受け取る
private snowmelt.Node	**RequestXML**(java.lang.String api, java.lang.String arg) サーバーAPI 呼び出し XML 形式の返答を受け取る
void	**setSessionID**(java.lang.String sid) セッション ID の設定
void	**setUser**(java.lang.String usr) ユーザ ID の設定

フィールドの詳細

baseURL

private java.lang.String **baseURL**

道路 GIS_API の URL

---

user

private java.lang.String **user**

ユーザーID

---

passwd

private java.lang.String **passwd**

パスワード

---

sessionID

private java.lang.String **sessionID**

セッション ID

---

paramtable

private java.util.Hashtable **paramtable**

セッション管理パラメータ

コンストラクタの詳細

CallAPI

public **CallAPI**(java.lang.String baseurl,
java.lang.String usr,
java.lang.String pass)

コンストラクタ

パラメータ:

baseurl - 道路 GIS サーバーの URL

user - ユーザーＩＤ

pass - パスワード

メソッドの詳細

login

public int **login**()

セッションを開始する

パラメータ:

user - ユーザ名

pass - パスワード

戻り値:

0:エラー無し 1:認証エラー 2:サーバー通信エラー

---

getMapImage

public java.awt.Image **getMapImage**(java.lang.String north,
java.lang.String west,
java.lang.String south,
java.lang.String east,
java.lang.String xsize,
java.lang.String ysize,
java.lang.String scale)

サーバから地図イメージを取得

パラメータ:

north - 北緯

west - 西経

south - 南緯

east - 東経

xsize - 画面 X サイズ

ysize - 画面 Y サイズ

scale - 縮尺

戻り値:

JPEG 画像ファイル

---

logout

public void **logout**()

セッションの破棄

RequestXML

private snowmelt.Node **RequestXML**(java.lang.String api,
java.lang.String arg)

サーバーAPI 呼び出し XML 形式の返答を受け取る

パラメータ:

api - Servlet サービス文字列

arg - パラメータ文字列

戻り値:

XML データ

---

getResponseNode

private snowmelt.Node **getResponseNode**(snowmelt.Node response_node)

サーバーのレスポンスの変換を行う

パラメータ:

response_node - 返還前ノード

戻り値:

変換後ノード

RequestImage

private java.awt.Image **RequestImage**(java.lang.String api,
java.lang.String arg)

サーバーAPI 呼び出し イメージデータを受け取る

パラメータ:

api - Servlet サービス文字列

arg - パラメータ文字列

戻り値:

イメージデータ

---

getUser

public java.lang.String **getUser**()

ユーザーID の取得

戻り値:

ユーザ ID

---

getSessionID

public java.lang.String **getSessionID**()

セッション ID の取得

戻り値:

セッション ID

---

setUser

public void **setUser**(java.lang.String usr)

ユーザ ID の設定

パラメータ:

usr - ユーザ ID

---

setSessionID

public void **setSessionID**(java.lang.String sid)

セッション ID の設定

パラメータ:

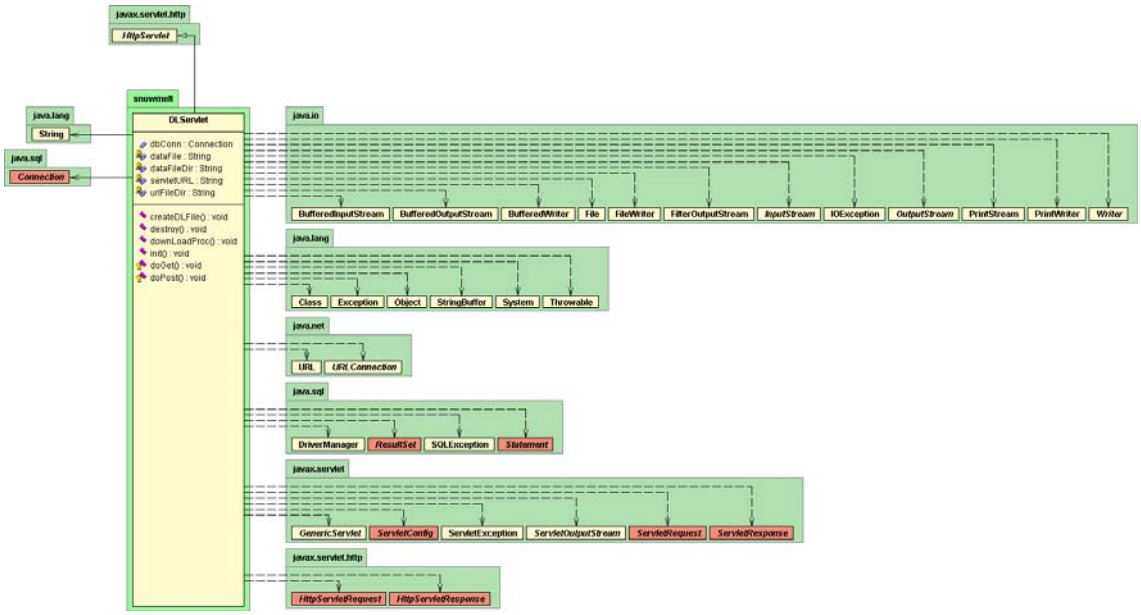
sid - セッション ID

---

printXML

void **printXML**(snowmelt.Node node)

■ 3　クラス DLServlet



java.lang.Object

　|

　+--javax.servlet.GenericServlet

　　|

　　+--javax.servlet.http.HttpServlet

　　　|

　　　+--**snowmelt.DLServlet**

すべての実装インタフェース:

　　java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

---

public class **DLServlet**

extends javax.servlet.http.HttpServlet

ファイルダウンロードサーブレット

フィールドの概要

private　java.lang.String	**dataFile** 　　格納データファイル名

private java.lang.String	**dataFileDir** 格納データファイルのディレクトリ名
java.sql.Connection	**dbConn** データベースコネクション
private java.lang.String	**servletURL** サーブレット URL
private java.lang.String	**urlFileDir** サーブレット URL のディレクトリ名

コンストラクタの概要

DLServlet()

メソッドの概要

void	**createDLFile**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) 作業者リスト作成
void	**destroy**() 破棄処理
protected void	**doGet**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) ブラウザからの呼び出し
protected void	**doPost**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) ブラウザからの呼び出し
void	**downLoadProc**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response)

	ダウンロード処理
void	init(javax.servlet.ServletConfig config) 初期化処理

フィールドの詳細

dbConn

public java.sql.Connection **dbConn**

データベースコネクション

---

servletURL

private java.lang.String **servletURL**

サーブレット URL

---

urlFileDir

private java.lang.String **urlFileDir**

サーブレット URL のディレクトリ名

---

dataFileDir

private java.lang.String **dataFileDir**

格納データファイルのディレクトリ名

---

dataFile

private java.lang.String **dataFile**

格納データファイル名

コンストラクタの詳細

DLServlet

public **DLServlet**()

メソッドの詳細

init

public void **init**(javax.servlet.ServletConfig config)
throws javax.servlet.ServletException

初期化処理

オーバーライド:

クラス javax.servlet.GenericServlet 内の init

パラメータ:

config - ServletConfig クラス

例外:

javax.servlet.ServletException -

---

destroy

public void **destroy**()

破棄処理

オーバーライド:

クラス javax.servlet.GenericServlet 内の destroy

---

doGet

protected void **doGet**(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws javax.servlet.ServletException,
java.io.IOException

ブラウザからの呼び出し

オーバーライド:

クラス javax.servlet.http.HttpServlet 内の doGet

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

doPost

```
protected void doPost(javax.servlet.http.HttpServletRequest request,
                      javax.servlet.http.HttpServletResponse response)
               throws javax.servlet.ServletException,
                      java.io.IOException
```

ブラウザからの呼び出し

オーバーライド:

クラス javax.servlet.http.HttpServlet 内の doPost

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

createDLFile

```
public void createDLFile(javax.servlet.http.HttpServletRequest request,
                         javax.servlet.http.HttpServletResponse response)
                  throws javax.servlet.ServletException,
                         java.io.IOException
```

作業者リスト作成

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

downLoadProc

```
public void downLoadProc(javax.servlet.http.HttpServletRequest request,
                         javax.servlet.http.HttpServletResponse response)
                  throws javax.servlet.ServletException,
                         java.io.IOException
```

ダウンロード処理

パラメータ:

request - HttpServletRequest クラス

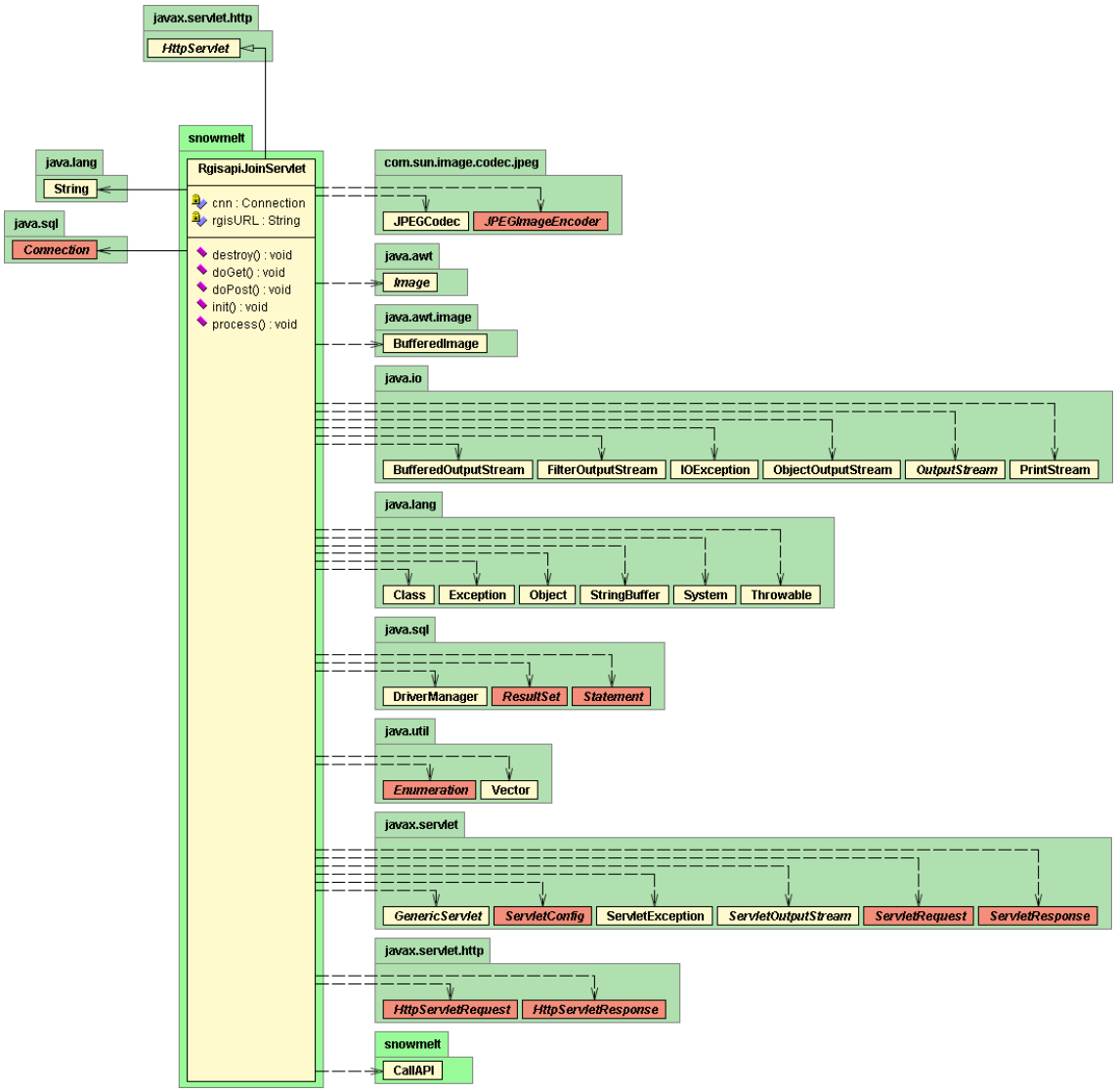
response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

■ 4　クラス RgisapiJoinServlet



java.lang.Object

  |

  +--javax.servlet.GenericServlet

　　|

　　+--javax.servlet.http.HttpServlet

　　　　|

　　　　+--**snowmelt.RgisapiJoinServlet**

すべての実装インタフェース:

　　java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

---

public class **RgisapiJoinServlet**

extends javax.servlet.http.HttpServlet

アプレットと道路 GIS_API サーブレットの受け渡しを行うサーブレット

フィールドの概要

private java.sql.Connection	**cnn** DB コネクション
private java.lang.String	**rgisURL** 道路 GIS サーバーURL

コンストラクタの概要

RgisapiJoinServlet()

メソッドの概要

void	**destroy**() destroy メソッド
void	**doGet**(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse res) ブラウザからの呼び出し
void	**doPost**(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse res) ブラウザからの呼び出し
void	**init**(javax.servlet.ServletConfig config) init メソッド
void	**process**(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse res) サーブレット

フィールドの詳細

rgisURL

private java.lang.String **rgisURL**

道路 GIS サーバーURL

---

cnn

private java.sql.Connection **cnn**

DB コネクション

コンストラクタの詳細

RgisapiJoinServlet

public **RgisapiJoinServlet**()

メソッドの詳細

init

public void **init**(javax.servlet.ServletConfig config)
throws javax.servlet.ServletException

init メソッド

オーバーライド:

クラス javax.servlet.GenericServlet 内の init

パラメータ:

config - ServletConfig クラス

---

doGet

public void **doGet**(javax.servlet.http.HttpServletRequest req,
javax.servlet.http.HttpServletResponse res)
throws java.io.IOException,
javax.servlet.ServletException

ブラウザからの呼び出し

オーバーライド:

クラス javax.servlet.http.HttpServlet 内の doGet

パラメータ:

req - HttpServletRequest クラス

res - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

doPost

public void **doPost**(javax.servlet.http.HttpServletRequest req,
javax.servlet.http.HttpServletResponse res)
throws java.io.IOException,
javax.servlet.ServletException

ブラウザからの呼び出し

オーバーライド:

クラス javax.servlet.http.HttpServlet 内の doPost

パラメータ:

req - HttpServletRequest クラス

res - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

process

public void **process**(javax.servlet.http.HttpServletRequest req,
javax.servlet.http.HttpServletResponse res)
throws java.io.IOException,

javax.servlet.ServletException

サーブレット

パラメータ:

req - HttpServletRequest クラス

res - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

destroy

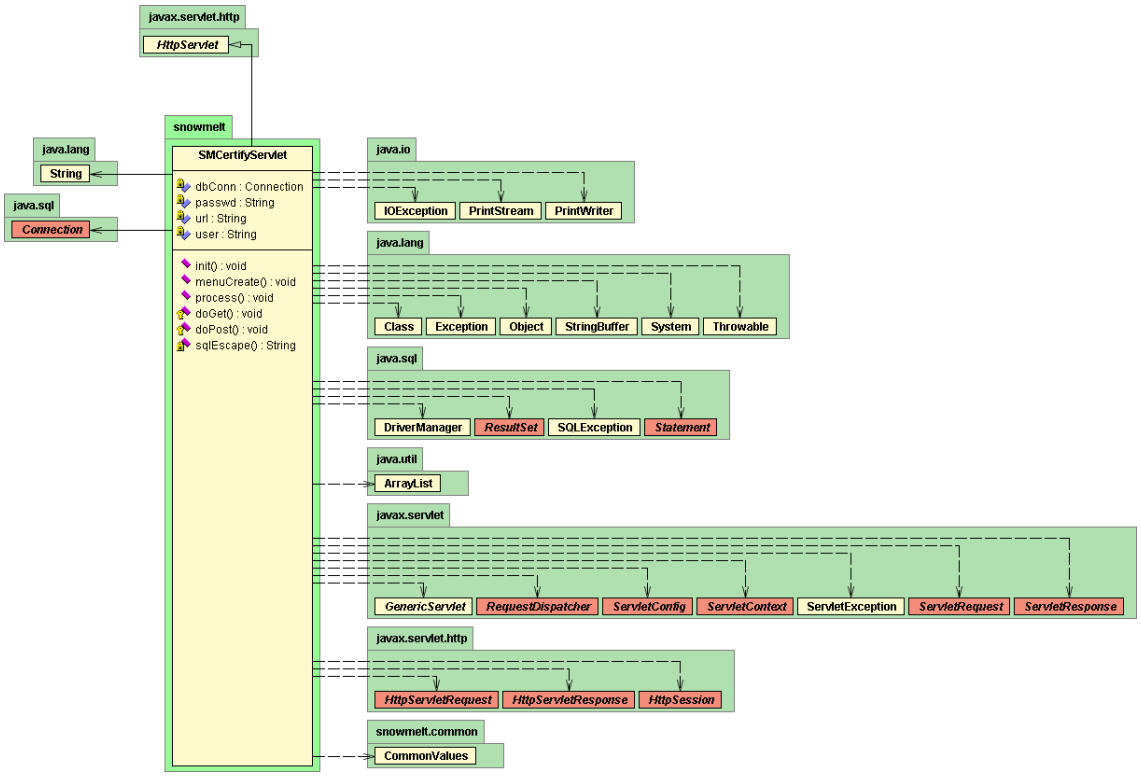
public void **destroy**()

destroy メソッド

オーバーライド:

クラス javax.servlet.GenericServlet 内の destroy

■ ５　クラス SMCertifyServlet




```
java.lang.Object
  |
  +--javax.servlet.GenericServlet
        |
        +--javax.servlet.http.HttpServlet
              |
              +--snowmelt.SMCertifyServlet
```

すべての実装インタフェース:

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

---

public class **SMCertifyServlet**

extends javax.servlet.http.HttpServlet

ユーザー認証 サーブレット

フィールドの概要

private java.sql.Connection	**dbConn** データベースコネクション
private java.lang.String	**passwd** パスワード
private java.lang.String	**url** ＵＲＬ
private java.lang.String	**user** ユーザーＩＤ

コンストラクタの概要

SMCertifyServlet()

メソッドの概要

protected void	**doGet**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) ブラウザからの呼び出し
protected void	**doPost**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) ブラウザからの呼び出し
void	**init**(javax.servlet.ServletConfig config) 初期化処理
void	**menuCreate**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response, java.util.ArrayList blongCodeArray, java.util.ArrayList jimusyoMeiArray, java.util.ArrayList koukuMeiArray)

	動的メインメニュー作成
void	**process**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) ユーザー認証
private java.lang.String	**sqlEscape**(java.lang.String strEsc) SQL エンコード

フィールドの詳細

dbConn

private java.sql.Connection **dbConn**

データベースコネクション

---

url

private java.lang.String **url**

ＵＲＬ

---

user

private java.lang.String **user**

ユーザーＩＤ

---

passwd

private java.lang.String **passwd**

パスワード

コンストラクタの詳細

SMCertifyServlet

public **SMCertifyServlet**()

メソッドの詳細

init

public void **init**(javax.servlet.ServletConfig config)
throws javax.servlet.ServletException

初期化処理

オーバーライド:

クラス javax.servlet.GenericServlet 内の init

パラメータ:

config - ServletConfig クラス

例外:

javax.servlet.ServletException -

---

doGet

protected void **doGet**(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws javax.servlet.ServletException,
java.io.IOException

ブラウザからの呼び出し

オーバーライド:

クラス javax.servlet.http.HttpServlet 内の doGet

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

doPost

```
protected void doPost(javax.servlet.http.HttpServletRequest request,
                      javax.servlet.http.HttpServletResponse response)
               throws javax.servlet.ServletException,
                      java.io.IOException
```

ブラウザからの呼び出し

オーバーライド:

クラス javax.servlet.http.HttpServlet 内の doPost

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

process

```
public void process(javax.servlet.http.HttpServletRequest request,
                    javax.servlet.http.HttpServletResponse response)
             throws javax.servlet.ServletException,
                    java.io.IOException
```

ユーザー認証

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

menuCreate

```
public void menuCreate(javax.servlet.http.HttpServletRequest request,
                       javax.servlet.http.HttpServletResponse response,
                       java.util.ArrayList blongCodeArray,
```

java.util.ArrayList jimusyoMeiArray,
java.util.ArrayList koukuMeiArray)
throws javax.servlet.ServletException,
java.io.IOException

動的メインメニュー作成

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

blongCodeArray - 所属コード配列

jimusyoMeiArray - 事務所名配列

koukuMeiArray - 工区名配列

例外:

javax.servlet.ServletException -

java.io.IOException -

---

sqlEscape

private java.lang.String **sqlEscape**(java.lang.String strEsc)

SQL エンコード

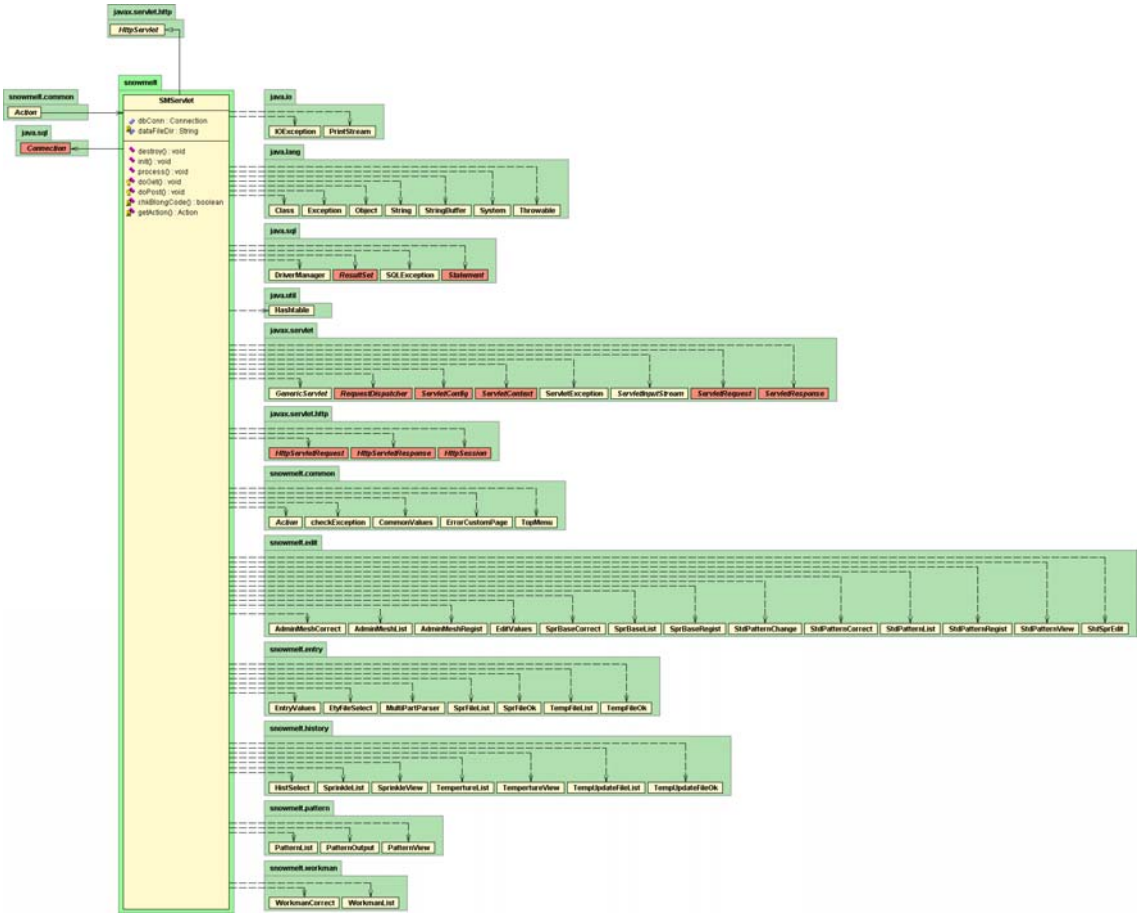
パラメータ:

strEsc - SQL

戻り値:

エンコードされた SQL

■ ６　クラス SMServlet



java.lang.Object
  |
  +--javax.servlet.GenericServlet
        |
        +--javax.servlet.http.HttpServlet
              |
              +--**snowmelt.SMServlet**

すべての実装インタフェース:

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

---

public class **SMServlet**
extends javax.servlet.http.HttpServlet

メインサーブレット

フィールドの概要

private java.lang.String	**dataFileDir** データ格納フォルダ
java.sql.Connection	**dbConn** データベースコネクション

コンストラクタの概要

SMServlet()

メソッドの概要

private boolean	**chkBlongCode**(java.sql.Connection dbConn, CommonValues commonValues, javax.servlet.http.HttpServletRequest request) 所属コード存在チェック＆GIS Server へのユーザーＩＤ登録
void	**destroy**() 破棄処理
protected void	**doGet**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) ブラウザからの呼び出し
protected void	**doPost**(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) ブラウザからの呼び出し
private Action	**getAction**(java.lang.String id, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response,

	Action actionRev, CommonValues commonValues, EditValues editValues, EntryValues entryValues, MultiPartParser tmpParser) URL から目的の Action を選択する
void	init(javax.servlet.ServletConfig config) 初期化処理
void	process(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response, java.sql.Connection dbConn) サーブレット

フィールドの詳細

dbConn

public java.sql.Connection **dbConn**

データベースコネクション

---

dataFileDir

private java.lang.String **dataFileDir**

データ格納フォルダ

コンストラクタの詳細

SMServlet

public **SMServlet**()

メソッドの詳細

init

public void **init**(javax.servlet.ServletConfig config)
throws javax.servlet.ServletException

初期化処理

オーバーライド:

クラス javax.servlet.GenericServlet 内の init

パラメータ:

config - ServletConfig クラス

例外:

javax.servlet.ServletException -

---

destroy

public void **destroy**()

破棄処理

オーバーライド:

クラス javax.servlet.GenericServlet 内の destroy

---

doGet

protected void **doGet**(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws javax.servlet.ServletException,
java.io.IOException

ブラウザからの呼び出し

オーバーライド:

クラス javax.servlet.http.HttpServlet 内の doGet

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

doPost

```
protected void doPost(javax.servlet.http.HttpServletRequest request,
                      javax.servlet.http.HttpServletResponse response)
               throws javax.servlet.ServletException,
                      java.io.IOException
```

ブラウザからの呼び出し

オーバーライド:

クラス javax.servlet.http.HttpServlet 内の doPost

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

process

```
public void process(javax.servlet.http.HttpServletRequest request,
                    javax.servlet.http.HttpServletResponse response,
                    java.sql.Connection dbConn)
             throws javax.servlet.ServletException,
                    java.io.IOException
```

サーブレット

パラメータ:

request - HttpServletRequest クラス

response - HttpServletResponse クラス

dbConn - Connection クラス

例外:

javax.servlet.ServletException -

java.io.IOException -

---

chkBlongCode

```
private boolean chkBlongCode(java.sql.Connection dbConn,
                             CommonValues commonValues,
                             javax.servlet.http.HttpServletRequest request)
```

所属コード存在チェック＆GIS Server へのユーザーＩＤ登録

パラメータ:

dbConn - Connection クラス

commonValues - CommonValues クラス

request - HttpServletRequest クラス

戻り値:

---

getAction

```
private Action getAction(java.lang.String id,
                         javax.servlet.http.HttpServletRequest request,
                         javax.servlet.http.HttpServletResponse response,
                         Action actionRev,
                         CommonValues commonValues,
                         EditValues editValues,
                         EntryValues entryValues,
                         MultiPartParser tmpParser)
                  throws checkException
```

URL から目的の Action を選択する

パラメータ:

id - アクションＩＤ

request - HttpServletRequest クラス

response - HttpServletResponse クラス

actionRev - Action クラス

commonValues - CommonValues クラス

editValues - EditValues クラス

entryValues - EntryValues クラス

tmpParser - MultiPartParser クラス

戻り値:

Action クラス

例外:

checkException -

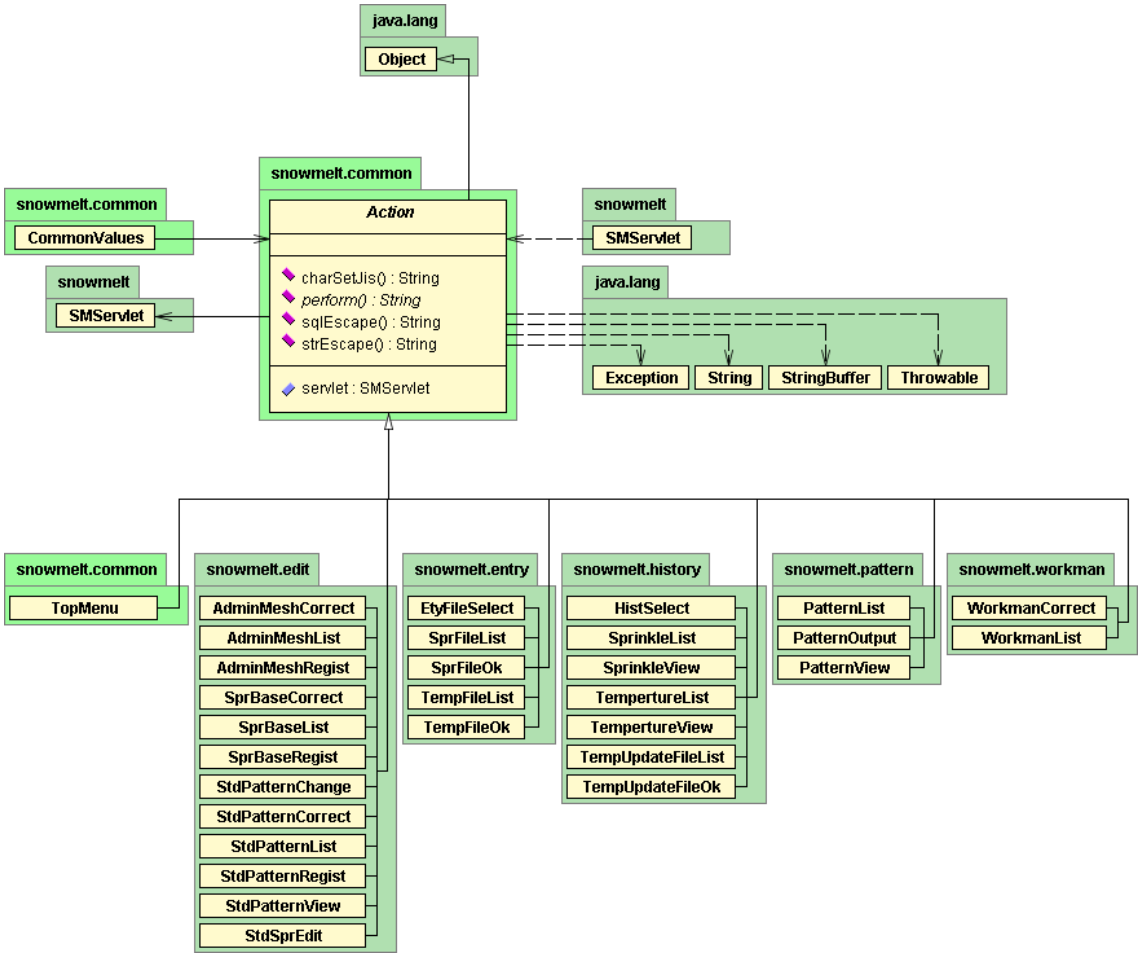
---

c） パッケージ snowmelt.common

クラスの概要	
Action	車載用高精度位置特定実験システム抽象クラス
CommonValues	共通変数格納クラス
ErrorCustomPage	カスタムエラーページ　クラス
TopMenu	Ｔｏｐメニュークラス

例外の概要	
checkException	例外処理クラス

■ 1　クラス Action



java.lang.Object

  |

  +--**snowmelt.common.Action**

直系の既知のサブクラス:

AdminMeshCorrect, AdminMeshList, AdminMeshRegist, EtyFileSelect, HistSelect, PatternList, PatternOutput, PatternView, SprBaseCorrect, SprBaseList, SprBaseRegist, SprFileList, SprFileOk, SprinkleList, SprinkleView, StdPatternChange, StdPatternCorrect, StdPatternList, StdPatternRegist, StdPatternView, StdSprEdit, TempertureList, TempertureView, TempFileList, TempFileOk, TempUpdateFileList, TempUpdateFileOk, TopMenu, WorkmanCorrect, WorkmanList

---

public abstract class **Action**

extends java.lang.Object

車載用高精度位置特定実験システム抽象クラス

フィールドの概要

private SMServlet	**servlet** 共通サーブレット

コンストラクタの概要

Action()

メソッドの概要

java.lang.String	**charSetJis**(java.lang.String PrmData) JSP からの２バイト文字をエンコード
SMServlet	**getServlet**() サーブレットゲッター
abstract java.lang.String	**perform**() 遷移先ｊｓｐ名
void	**setServlet**(SMServlet servlet) サーブレットセッター
java.lang.String	**sqlEscape**(java.lang.String strEsc) SQL エンコード
java.lang.String	**strEscape**(java.lang.String strEsc) HTML エンコード

フィールドの詳細

servlet

private SMServlet **servlet**

共通サーブレット

コンストラクタの詳細

Action

public **Action**()

メソッドの詳細

setServlet

public void **setServlet**(SMServlet servlet)

サーブレットセッター

パラメータ:

servlet - サーブレット

---

getServlet

public SMServlet **getServlet**()

サーブレットゲッター

戻り値:

サーブレット

---

charSetJis

public java.lang.String **charSetJis**(java.lang.String PrmData)

JSP からの２バイト文字をエンコード

パラメータ:

PrmData - JSP からの２バイト文字

戻り値:

エンコード文字

---

strEscape

public java.lang.String **strEscape**(java.lang.String strEsc)

HTML エンコード

パラメータ:

strEsc - HTML 文字

戻り値:

エンコード文字

---

sqlEscape

public java.lang.String **sqlEscape**(java.lang.String strEsc)

SQL エンコード

パラメータ:

strEsc - SQL 文字

戻り値:

エンコード文字

---

perform

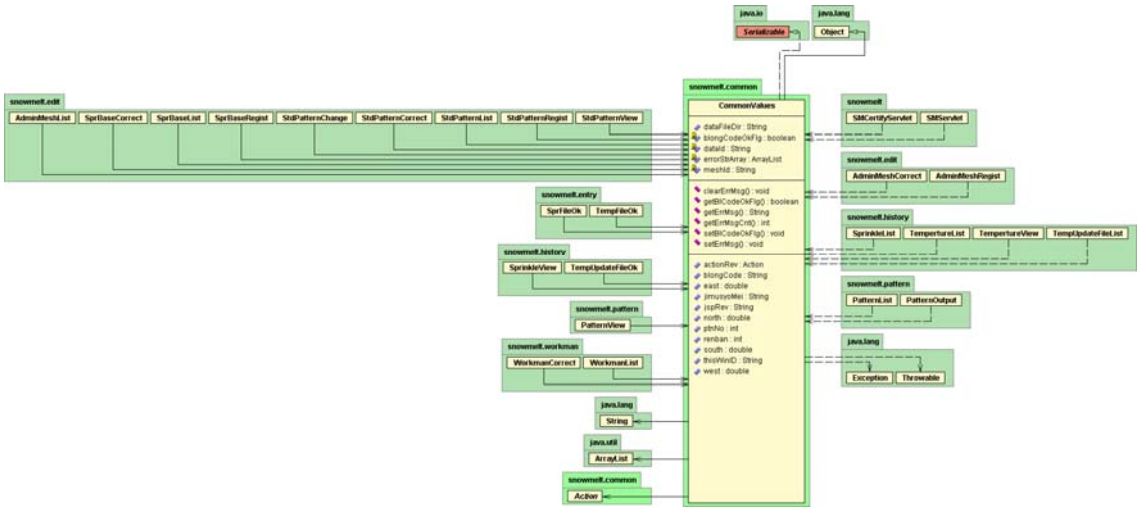
public abstract java.lang.String **perform**()

遷移先ｊｓｐ名

戻り値:

遷移先ｊｓｐ名

■ 2　クラス CommonValues



java.lang.Object

　|

　+--**snowmelt.common.CommonValues**

すべての実装インタフェース:

　　java.io.Serializable

---

public class **CommonValues**

extends java.lang.Object

implements java.io.Serializable

共通変数格納クラス

フィールドの概要

private　Action	**actionRev** 前 action
private　java.lang.String	**blongCode** 所属コード
private　boolean	**blongCodeOkFlg** 所属コード認証フラグ
java.lang.String	**dataFileDir**

	データフォルダ
private java.lang.String	**dataId** データＩＤ
private double	**east** 西経
private java.util.ArrayList	**errorStrArray** エラーメッセージ
private java.lang.String	**jimusyoMei** 事務所名
private java.lang.String	**jspRev** 前 jsp ページ
private java.lang.String	**meshId** メッシュＩＤ
private double	**north** 北緯
private int	**ptnNo** パターン番号
private int	**renban** 連番
private double	**south** 南緯
private java.lang.String	**thisWinID** 呼び出し元の親 ID
private double	**west** 東経

コンストラクタの概要

CommonValues()

メソッドの概要

void	**clearErrMsg**() エラーメッセージクリア
Action	**getActionRev**() 前 action ゲッター
boolean	**getBlCodeOkFlg**() 所属コード認証フラグゲッター
java.lang.String	**getBlongCode**() 所属コードゲッター
double	**getEast**() 西経ゲッター
java.lang.String	**getErrMsg**(int tblIdx) エラーメッセージゲッター
int	**getErrMsgCnt**() エラーメッセージ要素数ゲッター
java.lang.String	**getJimusyoMei**() 所属名ゲッター
java.lang.String	**getJspRev**() 前 jsp ページゲッター
double	**getNorth**() 北緯ゲッター
int	**getPtnNo**() パターン番号ゲッター
int	**getRenban**() 連番ゲッター
double	**getSouth**() 南緯ゲッター
java.lang.String	**getThisWinID**()

	現在 ID ゲッター
double	**getWest**() 東経ゲッター
void	**setActionRev**(Action PrmData) 前 action セッター
void	**setBlCodeOkFlg**(boolean PrmFlg) 所属コード認証フラグセッター
void	**setBlongCode**(java.lang.String PrmData) 所属コードセッター
void	**setEast**(double xyPos) 西経セッター
void	**setErrMsg**(java.lang.String PrmData) エラーメッセージセッター
void	**setJimusyoMei**(java.lang.String prmData) 所属名セッター
void	**setJspRev**(java.lang.String PrmData) 前 jsp ページセッター
void	**setNorth**(double xyPos) 北緯セッター
void	**setPtnNo**(int PrmData) パターン番号セッター
void	**setRenban**(int PrmData) 連番セッター
void	**setSouth**(double xyPos) 南緯セッター
void	**setThisWinID**(java.lang.String PrmData) 現在 ID セッター
void	**setWest**(double xyPos) 東経セッター

フィールドの詳細

dataFileDir

public java.lang.String **dataFileDir**

データフォルダ

---

blongCodeOkFlg

private boolean **blongCodeOkFlg**

所属コード認証フラグ

---

blongCode

private java.lang.String **blongCode**

所属コード

---

jimusyoMei

private java.lang.String **jimusyoMei**

事務所名

---

thisWinID

private java.lang.String **thisWinID**

呼び出し元の親 ID

---

ptnNo

private int **ptnNo**

パターン番号

---

renban

private int **renban**

連番

---

meshId

private java.lang.String **meshId**

メッシュＩＤ

---

dataId

private java.lang.String **dataId**

データＩＤ

---

errorStrArray

private java.util.ArrayList **errorStrArray**

エラーメッセージ

---

actionRev

private Action **actionRev**

前 action

---

jspRev

private java.lang.String **jspRev**

前 jsp ページ

---

west

private double **west**

東経

east

private double **east**

西経

north

private double **north**

北緯

south

private double **south**

南緯

コンストラクタの詳細

CommonValues

public **CommonValues**()

メソッドの詳細

getBlCodeOkFlg

public boolean **getBlCodeOkFlg**()

所属コード認証フラグゲッター

戻り値:

所属コード認証フラグ

setBlCodeOkFlg

public void **setBlCodeOkFlg**(boolean PrmFlg)

所属コード認証フラグセッター

パラメータ:

PrmFlg - 所属コード認証フラグ

---

getErrMsg

public java.lang.String **getErrMsg**(int tblIdx)

エラーメッセージゲッター

パラメータ:

tblIdx - インデックス

戻り値:

エラーメッセージ

---

setErrMsg

public void **setErrMsg**(java.lang.String PrmData)

エラーメッセージセッター

パラメータ:

PrmData - エラーメッセージ

---

clearErrMsg

public void **clearErrMsg**()

エラーメッセージクリア

---

getErrMsgCnt

public int **getErrMsgCnt**()

エラーメッセージ要素数ゲッター

戻り値:

エラーメッセージ要素数

---

getBlongCode

public java.lang.String **getBlongCode**()

所属コードゲッター

戻り値:

所属コード

---

setBlongCode

public void **setBlongCode**(java.lang.String PrmData)

所属コードセッター

パラメータ:

PrmData - 所属コード

---

getJimusyoMei

public java.lang.String **getJimusyoMei**()

所属名ゲッター

戻り値:

所属名

---

setJimusyoMei

public void **setJimusyoMei**(java.lang.String prmData)

所属名セッター

パラメータ:

prmData - 所属名

---

getThisWinID

public java.lang.String **getThisWinID**()

現在 ID ゲッター

---

setThisWinID

public void **setThisWinID**(java.lang.String PrmData)

現在 ID セッター

パラメータ:

PrmData - 現在 ID

---

getPtnNo

public int **getPtnNo**()

パターン番号ゲッター

戻り値:

パターン番号

---

setPtnNo

public void **setPtnNo**(int PrmData)

パターン番号セッター

パラメータ:

PrmData - パターン番号

---

getRenban

public int **getRenban**()

連番ゲッター

戻り値:

連番

---

setRenban

public void **setRenban**(int PrmData)

連番セッター

パラメータ:

PrmData - 連番

---

getActionRev

public Action **getActionRev**()

前 action ゲッター

戻り値:

前 action

---

setActionRev

public void **setActionRev**(Action PrmData)

前 action セッター

パラメータ:

PrmData -

---

getJspRev

public java.lang.String **getJspRev**()

前 jsp ページゲッター

戻り値:

前 jsp ページ

---

setJspRev

public void **setJspRev**(java.lang.String PrmData)

前 jsp ページセッター

パラメータ:

PrmData - 前 jsp ページ

---

getWest

public double **getWest**()

東経ゲッター

戻り値:

東経

---

setWest

public void **setWest**(double xyPos)

東経セッター

パラメータ:

xyPos - 東経

---

getEast

public double **getEast**()

西経ゲッター

戻り値:

西経

---

setEast

public void **setEast**(double xyPos)

西経セッター

パラメータ:

xyPos - 西経

---

getNorth

public double **getNorth**()

北緯ゲッター

戻り値:

北緯

---

setNorth

public void **setNorth**(double xyPos)

北緯セッター

パラメータ:

xyPos - 北緯

---

getSouth

public double **getSouth**()

南緯ゲッター

戻り値:

南緯

---

setSouth

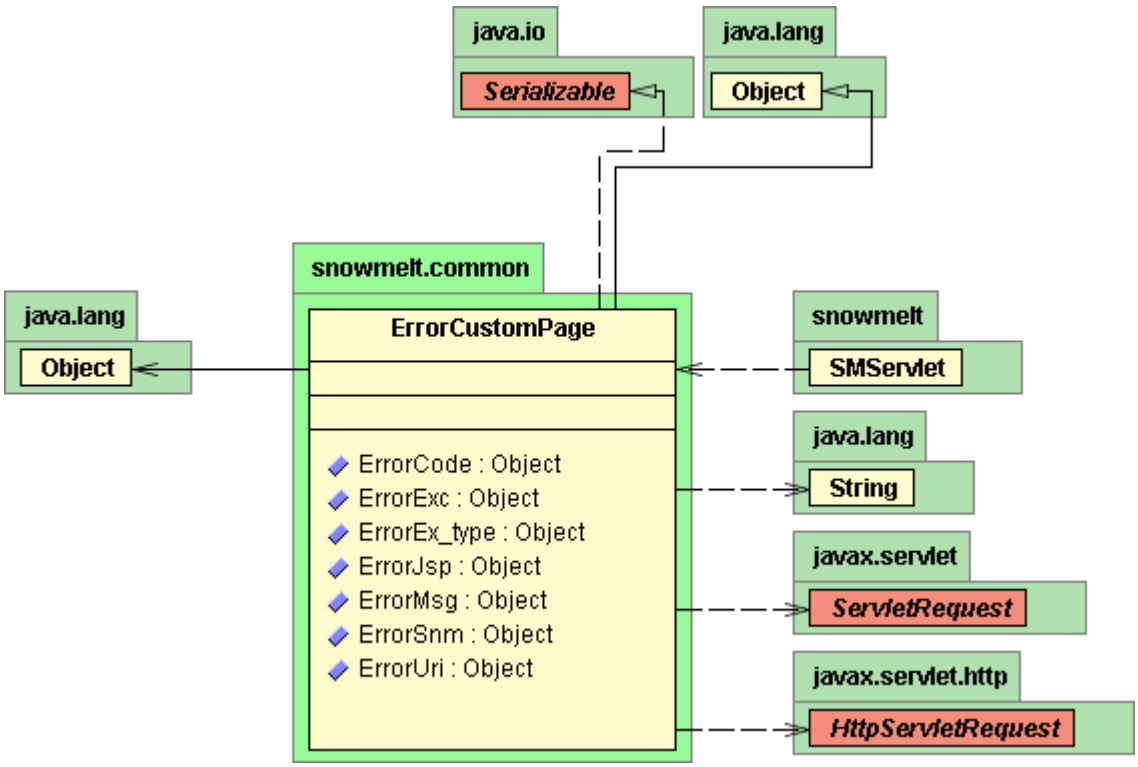
public void **setSouth**(double xyPos)

南緯セッター

パラメータ:

xyPos - 南緯

■ 3　クラス ErrorCustomPage



java.lang.Object

　|

　+--**snowmelt.common.ErrorCustomPage**

すべての実装インタフェース:

　　java.io.Serializable

---

public class **ErrorCustomPage**

extends java.lang.Object

implements java.io.Serializable

カスタムエラーページ　クラス

フィールドの概要	
private　java.lang.Object	**ErrorCode** 　　ステータスコード
private　java.lang.Object	**ErrorEx_type**

	エクセプションタイプ
private java.lang.Object	**ErrorExc** エクセプション
private java.lang.Object	**ErrorJsp** ＪＳＰ
private java.lang.Object	**ErrorMsg** エラーメッセージ
private java.lang.Object	**ErrorSnm** サーブレット
private java.lang.Object	**ErrorUri** リクエストＵＲＩ

コンストラクタの概要

ErrorCustomPage()

メソッドの概要

java.lang.Object	**getErrorCode**(javax.servlet.http.HttpServletRequest request) ステータスコードゲッター
java.lang.Object	**getErrorEx_type**(javax.servlet.http.HttpServletRequest request) エクセプションタイプゲッター
java.lang.Object	**getErrorExc**(javax.servlet.http.HttpServletRequest request) エクセプションゲッター
java.lang.Object	**getErrorJsp**(

	javax.servlet.http.HttpServletRequest request) 　　ＪＳＰゲッター
java.lang.Object	**getErrorMsg**(javax.servlet.http.HttpServletRequest request) 　　エラーメッセージゲッター
java.lang.Object	**getErrorSnm**(javax.servlet.http.HttpServletRequest request) 　　サーブレットゲッター
java.lang.Object	**getErrorUri**(javax.servlet.http.HttpServletRequest request) 　　リクエストＵＲＩゲッター

フィールドの詳細

ErrorCode

private java.lang.Object **ErrorCode**

　　ステータスコード

ErrorEx_type

private java.lang.Object **ErrorEx_type**

　　エクセプションタイプ

ErrorMsg

private java.lang.Object **ErrorMsg**

　　エラーメッセージ

ErrorExc

private java.lang.Object **ErrorExc**

エクセプション

ErrorUri

private java.lang.Object **ErrorUri**

リクエストＵＲＩ

ErrorSnm

private java.lang.Object **ErrorSnm**

サーブレット

ErrorJsp

private java.lang.Object **ErrorJsp**

ＪＳＰ

コンストラクタの詳細

ErrorCustomPage

public **ErrorCustomPage**()

メソッドの詳細

getErrorCode

public java.lang.Object **getErrorCode**(javax.servlet.http.HttpServletRequest request)

ステータスコードゲッター

パラメータ:

request - HttpServletRequest

戻り値:

ステータスコード

---

getErrorEx_type

public java.lang.Object **getErrorEx_type**(javax.servlet.http.HttpServletRequest request)

エクセプションタイプゲッター

パラメータ:

request - HttpServletRequest

戻り値:

エクセプションタイプ

---

getErrorMsg

public java.lang.Object **getErrorMsg**(javax.servlet.http.HttpServletRequest request)

エラーメッセージゲッター

パラメータ:

request - HttpServletRequest

戻り値:

エラーメッセージ

---

getErrorExc

public java.lang.Object **getErrorExc**(javax.servlet.http.HttpServletRequest request)

エクセプションゲッター

パラメータ:

request - HttpServletRequest

戻り値:

エクセプション

---

getErrorUri

public java.lang.Object **getErrorUri**(javax.servlet.http.HttpServletRequest request)

リクエストＵＲＩゲッター

パラメータ:

request - HttpServletRequest

戻り値:

リクエストＵＲＩ

---

getErrorSnm

public java.lang.Object **getErrorSnm**(javax.servlet.http.HttpServletRequest request)

サーブレットゲッター

パラメータ:

request - HttpServletRequest

戻り値:

サーブレット名

---

getErrorJsp

public java.lang.Object **getErrorJsp**(javax.servlet.http.HttpServletRequest request)

ＪＳＰゲッター

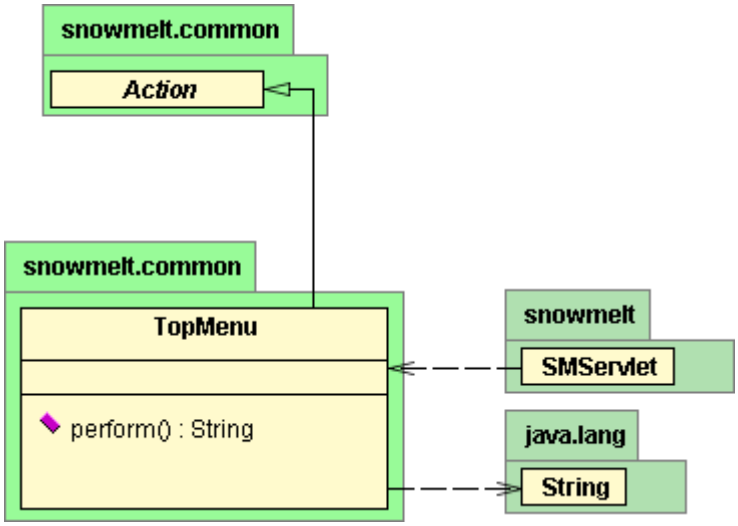
パラメータ:

request - HttpServletRequest

戻り値:

ＪＳＰ

■ 4　クラス TopMenu




```
java.lang.Object
  |
  +--snowmelt.common.Action
        |
        +--snowmelt.common.TopMenu
```

---

public class **TopMenu**

extends Action

Ｔｏｐメニュークラス

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

TopMenu ()

メソッドの概要

java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

コンストラクタの詳細

TopMenu

public **TopMenu**()

メソッドの詳細

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

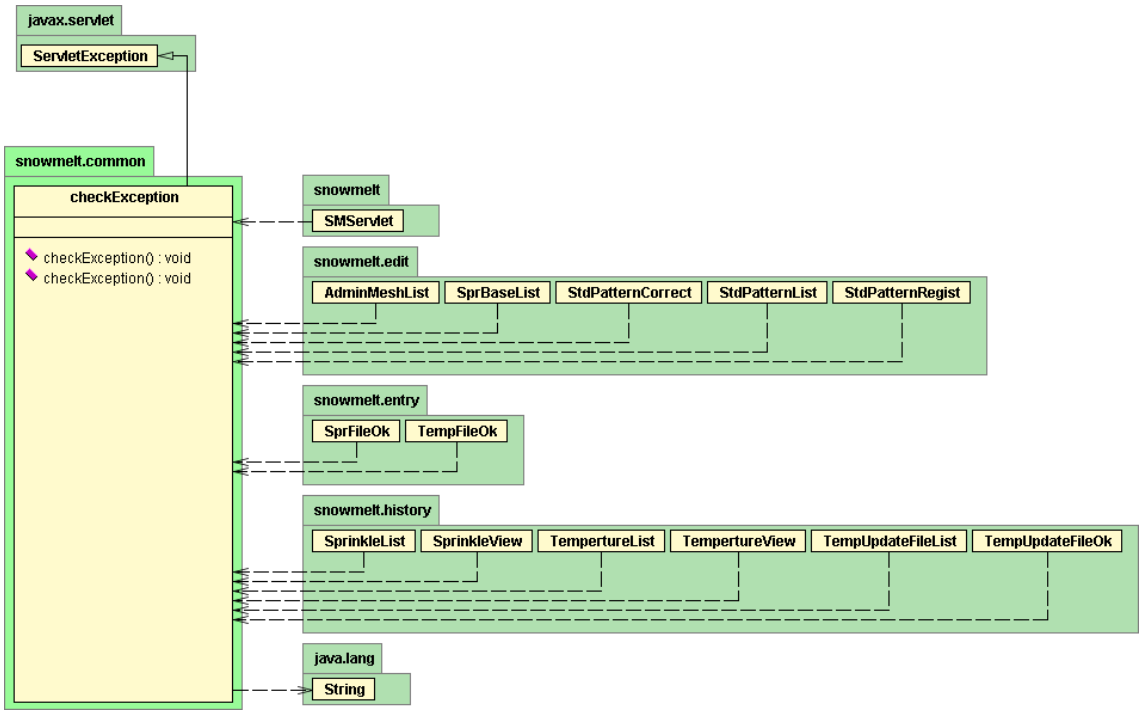
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/Menu.jsp"

■ ５　クラス checkException



java.lang.Object

  |

  +--java.lang.Throwable

        |

        +--java.lang.Exception

              |

              +--javax.servlet.ServletException

                    |

                    +--**snowmelt.common.checkException**

すべての実装インタフェース:

java.io.Serializable

---

public class **checkException**

extends javax.servlet.ServletException

例外処理クラス

コンストラクタの概要

checkException() コンストラクタ
checkException(java.lang.String msg) コンストラクタ

コンストラクタの詳細

checkException

public **checkException**()

コンストラクタ

---

checkException

public **checkException**(java.lang.String msg)

コンストラクタ

パラメータ:

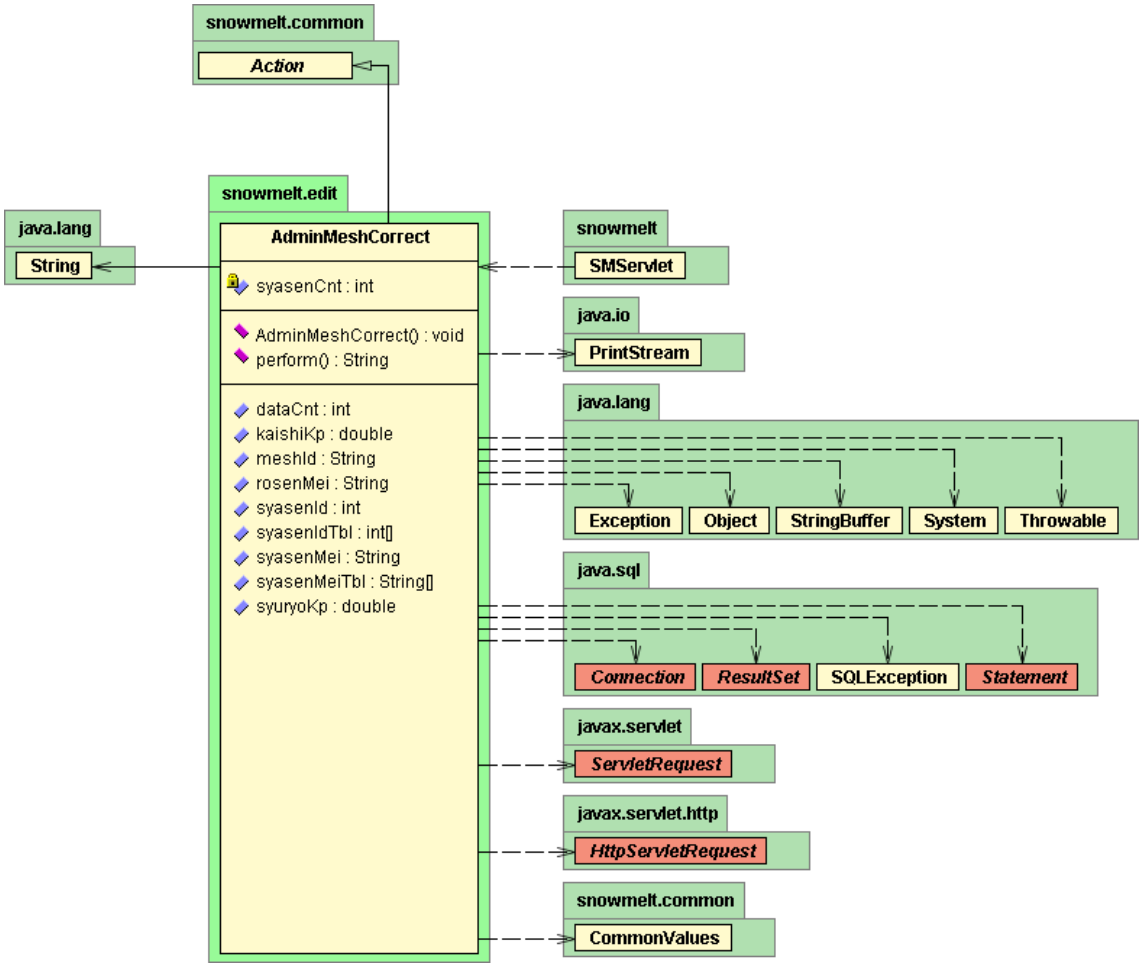
msg - 例外メッセージ

---

d） パッケージ snowmelt.edit

クラスの概要	
AdminMeshCorrect	管理メッシュ修正クラス
AdminMeshList	管理メッシュ一覧クラス
AdminMeshRegist	管理メッシュ新規登録クラス
EditValues	標準散布パターン作成　共通変数格納クラス
SprBaseCorrect	散布基本データ修正クラス
SprBaseList	散布基本データ修正クラス
SprBaseRegist	散布基本データ新規登録クラス
StdPatternChange	管理メッシュ・散布基本データの修正・削除クラス
StdPatternCorrect	標準散布パターン修正クラス
StdPatternList	標準散布パターン一覧クラス
StdPatternRegist	標準散布パターン新規登録クラス
StdPatternView	標準散布パターン表示クラス
StdSprEdit	標準散布パターン作成メニュークラス

■ 1　クラス AdminMeshCorrect



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.edit.AdminMeshCorrect**

---

public class **AdminMeshCorrect**

extends Action

管理メッシュ修正クラス

フィールドの概要

private int	**dataCnt**

	データ数
private double	**kaishiKp** 開始ＫＰ
private java.lang.String	**meshId** メッシュＩＤ
private java.lang.String	**rosenMei** 路線名
private int	**syasenCnt** 車線数
private int	**syasenId** 車線ＩＤ
private int[]	**syasenIdTbl** 車線ＩＤ配列
private java.lang.String	**syasenMei** 車線名
private java.lang.String[]	**syasenMeiTbl** 車線名配列
private double	**syuryoKp** 終了ＫＰ

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

AdminMeshCorrect(java.sql.Connection dbConn, CommonValues commonValues, javax.servlet.http.HttpServletRequest request) コンストラクタ

メソッドの概要

int	**getDataCnt**() 車線名データ数を返す
double	**getKaishiKp**() 開始ＫＰを返す
java.lang.String	**getMeshId**() メッシュＩＤを返す
java.lang.String	**getRosenMei**() 路線名を返す
int	**getSyasenId**() 車線ＩＤを返す
int	**getSyasenIdTbl**(int dataIndex) 車線ＩＤを返す
java.lang.String	**getSyasenMei**() 車線名を返す
java.lang.String	**getSyasenMeiTbl**(int dataIndex) 車線名を返す
double	**getSyuryoKp**() 終了ＫＰを返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

syasenIdTbl

private int[] syasenIdTbl

車線ＩＤ配列

---

syasenMeiTbl

private java.lang.String[] syasenMeiTbl

車線名配列

---

syasenCnt

private int syasenCnt

車線数

---

dataCnt

private int dataCnt

データ数

---

meshId

private java.lang.String meshId

メッシュＩＤ

---

rosenMei

private java.lang.String rosenMei

路線名

---

syasenId

private int **syasenId**

車線ＩＤ

---

syasenMei

private java.lang.String **syasenMei**

車線名

---

kaishiKp

private double **kaishiKp**

開始ＫＰ

---

syuryoKp

private double **syuryoKp**

終了ＫＰ

コンストラクタの詳細

AdminMeshCorrect

public **AdminMeshCorrect**(java.sql.Connection dbConn,
CommonValues commonValues,
javax.servlet.http.HttpServletRequest request)

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonValues - CommonValues クラス

request - HttpServletRequest クラス

メソッドの詳細

getMeshId

public java.lang.String **getMeshId**()

メッシュＩＤを返す

戻り値:

メッシュＩＤ

---

getRosenMei

public java.lang.String **getRosenMei**()

路線名を返す

戻り値:

路線名

---

getSyasenId

public int **getSyasenId**()

車線ＩＤを返す

戻り値:

車線ＩＤ

---

getSyasenMei

public java.lang.String **getSyasenMei**()

車線名を返す

戻り値:

車線名

---

getKaishiKp

public double **getKaishiKp**()

開始ＫＰを返す

戻り値:

開始ＫＰ

---

getSyuryoKp

public double **getSyuryoKp**()

終了ＫＰを返す

戻り値:

終了ＫＰ

---

getSyasenIdTbl

public int **getSyasenIdTbl**(int dataIndex)

車線ＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

車線ＩＤ

---

getSyasenMeiTbl

public java.lang.String **getSyasenMeiTbl**(int dataIndex)

車線名を返す

パラメータ:

dataIndex - インデックス

戻り値:

車線名

---

getDataCnt

public int **getDataCnt**()

車線名データ数を返す

戻り値:

車線名データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

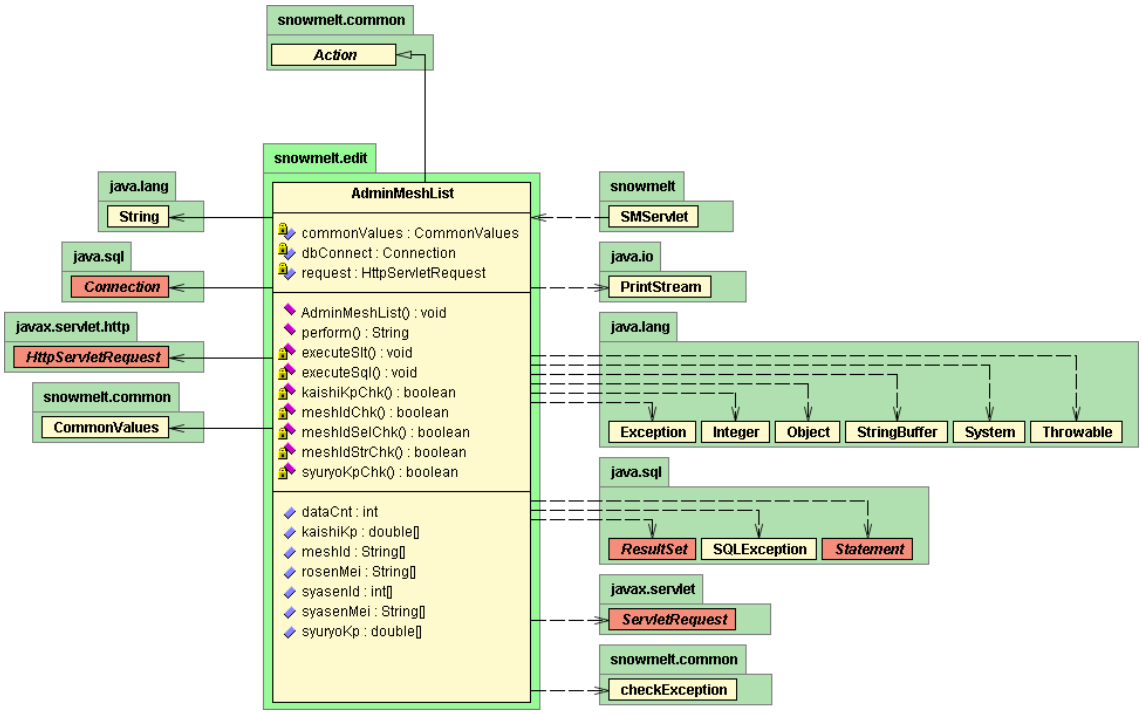
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/D12.jsp"

■ 2　クラス AdminMeshList



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.edit.AdminMeshList**

---

public class **AdminMeshList**

extends Action

管理メッシュ一覧クラス

フィールドの概要	
private CommonValues	**commonValues** 共通変数クラスのインスタンス
private int	**dataCnt** データ数
private java.sql.Connection	**dbConnect**

	Connection クラス
private double[]	**kaishiKp** 開始ＫＰ
private java.lang.String[]	**meshId** メッシュＩＤ
private javax.servlet.http.HttpServletRequest	**request** HttpServletRequest クラス
private java.lang.String[]	**rosenMei** 路線名
private int[]	**syasenId** 車線ＩＤ
private java.lang.String[]	**syasenMei** 車線名
private double[]	**syuryoKp** 終了ＫＰ

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

AdminMeshList(java.lang.String id, java.sql.Connection dbConn, CommonValues commonVal, javax.servlet.http.HttpServletRequest req)
コンストラクタ

メソッドの概要

private void	**executeSlt**() データ取得
private void	**executeSql**(java.lang.String id) データ更新
int	**getDataCnt**() データ数を返す
double	**getKaishiKp**(int dataIndex) 開始ＫＰを返す
java.lang.String	**getMeshId**(int dataIndex) メッシュＩＤを返す
java.lang.String	**getRosenMei**(int dataIndex) 路線名を返す
int	**getSyasenId**(int dataIndex) 車線ＩＤを返す
java.lang.String	**getSyasenMei**(int dataIndex) 車線名を返す
double	**getSyuryoKp**(int dataIndex) 終了ＫＰを返す
private boolean	**kaishiKpChk**(java.lang.String blongCode, int syasenId, java.lang.String kaishiKp) 開始ＫＰ存在チェック
private boolean	**meshIdChk**(java.lang.String blongCode, java.lang.String meshId) メッシュＩＤ重複チェック
private boolean	**meshIdSelChk**(java.lang.String blongCode, java.lang.String meshId) 参照整合性チェック
private boolean	**meshIdStrChk**(java.lang.String meshId) メッシュＩＤ文字チェック
java.lang.String	**perform**() 遷移先ｊｓｐ名

private boolean	**syuryoKpChk**(java.lang.String blongCode, int syasenId, java.lang.String kaishiKp, java.lang.String syuryoKp) 終了ＫＰ存在チェック

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

dbConnect

private java.sql.Connection **dbConnect**

Connection クラス

---

request

private javax.servlet.http.HttpServletRequest **request**

HttpServletRequest クラス

---

dataCnt

private int **dataCnt**

データ数

---

meshId

private java.lang.String[] **meshId**

メッシュＩＤ

---

rosenMei

private java.lang.String[] **rosenMei**

路線名

---

syasenId

private int[] **syasenId**

車線ＩＤ

---

syasenMei

private java.lang.String[] **syasenMei**

車線名

---

kaishiKp

private double[] **kaishiKp**

開始ＫＰ

---

syuryoKp

private double[] **syuryoKp**

終了ＫＰ

コンストラクタの詳細

AdminMeshList

public **AdminMeshList**(java.lang.String id,
java.sql.Connection dbConn,
CommonValues commonVal,
javax.servlet.http.HttpServletRequest req)
throws checkException

コンストラクタ

パラメータ:

id - アクションＩＤ

dbConn - Connection クラス

commonVal - CommonValues クラス

req - HttpServletRequest クラス

例外:

checkException -

メソッドの詳細

executeSql

private void **executeSql**(java.lang.String id)
throws checkException

データ更新

パラメータ:

id - アクションＩＤ

例外:

checkException -

---

meshIdStrChk

private boolean **meshIdStrChk**(java.lang.String meshId)

メッシュＩＤ文字チェック

パラメータ:

prmStr - メッシュＩＤ

戻り値:

true:OK false:NG

---

meshIdChk

private boolean **meshIdChk**(java.lang.String blongCode,
java.lang.String meshId)

メッシュＩＤ重複チェック

パラメータ:

blongCode - 所属コード

meshId - メッシュＩＤ

戻り値:

true:重複なし false:重複あり

---

kaishiKpChk

private boolean **kaishiKpChk**(java.lang.String blongCode,
int syasenId,
java.lang.String kaishiKp)

開始ＫＰ存在チェック

パラメータ:

blongCode - 所属コード

syasenId - 車線ＩＤ

kaishiKp - 開始ＫＰ

戻り値:

true:存在 false:存在しない

---

syuryoKpChk

private boolean **syuryoKpChk**(java.lang.String blongCode,
int syasenId,
java.lang.String kaishiKp,
java.lang.String syuryoKp)

終了ＫＰ存在チェック

パラメータ:

blongCode - 所属コード

syasenId - 車線ＩＤ

kaishiKp - 開始ＫＰ

syuryoKp - 終了ＫＰ

戻り値:

true:存在 false:存在しない

---

meshIdSelChk

private boolean **meshIdSelChk**(java.lang.String blongCode,
java.lang.String meshId)

参照整合性チェック

パラメータ:

blongCode - 所属コード

meshId - メッシュＩＤ

戻り値:

true:削除 OK false:削除 NG

---

executeSlt

private void **executeSlt**()

データ取得

---

getMeshId

public java.lang.String **getMeshId**(int dataIndex)

メッシュＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

メッシュＩＤ

---

getRosenMei

public java.lang.String **getRosenMei**(int dataIndex)

路線名を返す

パラメータ:

dataIndex - インデックス

戻り値:

路線名

---

getSyasenId

public int **getSyasenId**(int dataIndex)

車線ＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

車線ＩＤ

---

getSyasenMei

public java.lang.String **getSyasenMei**(int dataIndex)

車線名を返す

パラメータ:

dataIndex - インデックス

戻り値:

車線名

---

getKaishiKp

public double **getKaishiKp**(int dataIndex)

開始ＫＰを返す

パラメータ:

dataIndex - インデックス

戻り値:

開始ＫＰ

---

getSyuryoKp

public double **getSyuryoKp**(int dataIndex)

終了ＫＰを返す

パラメータ:

dataIndex - インデックス

戻り値:

終了ＫＰ

---

getDataCnt

public int **getDataCnt**()

データ数を返す

戻り値:

データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

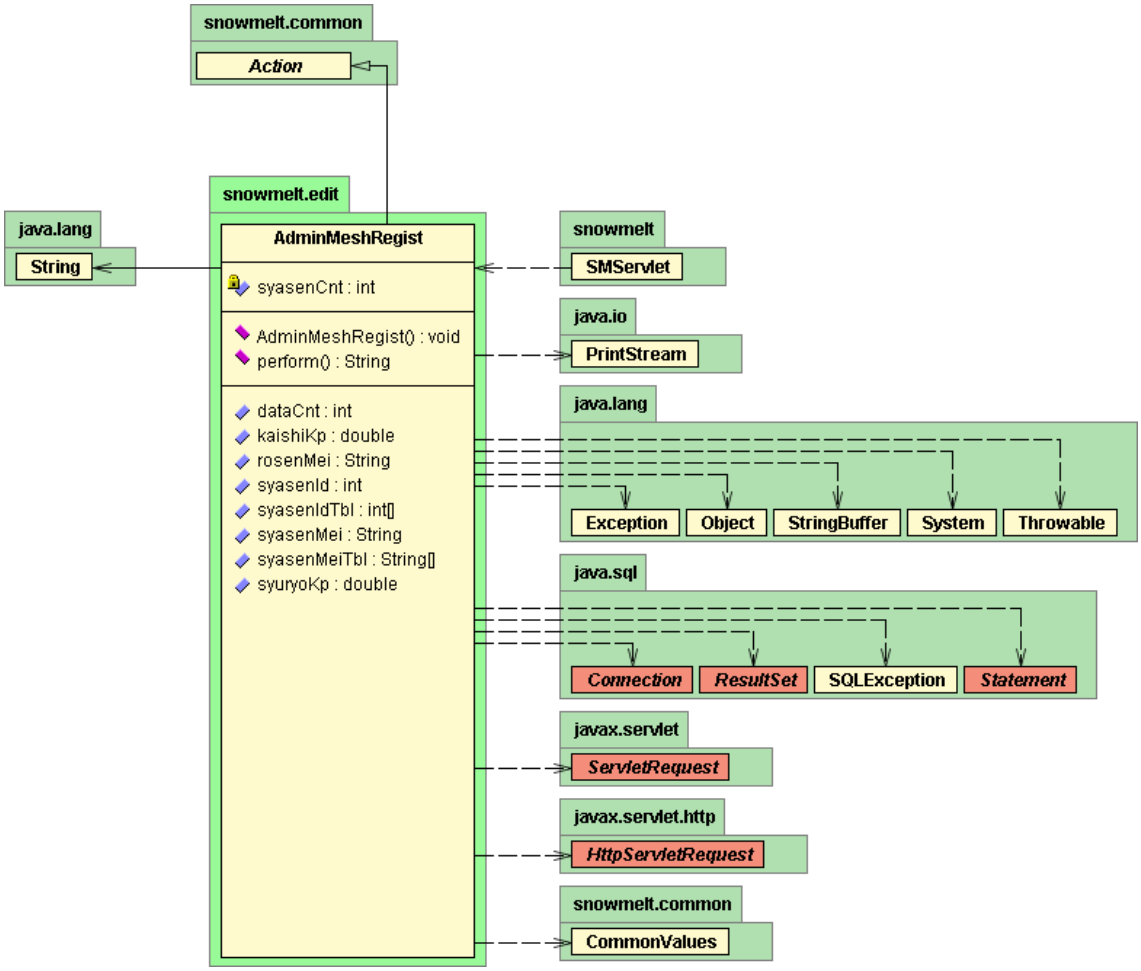
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/D11.jsp"

■ 3　クラス AdminMeshRegist



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.edit.AdminMeshRegist**

---

public class **AdminMeshRegist**

extends Action

管理メッシュ新規登録クラス

フィールドの概要

private int	**dataCnt**

	データ数
private double	**kaishiKp** 開始ＫＰ
private java.lang.String	**rosenMei** 路線名
private int	**syasenCnt** 車線数
private int	**syasenId** 車線ＩＤ
private int[]	**syasenIdTbl** 車線ＩＤ配列
private java.lang.String	**syasenMei** 車線名
private java.lang.String[]	**syasenMeiTbl** 車線名配列
private double	**syuryoKp** 終了ＫＰ

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

AdminMeshRegist(java.lang.String id, java.sql.Connection dbConn, CommonValues commonValues, javax.servlet.http.HttpServletRequest request)
コンストラクタ

メソッドの概要

int	**getDataCnt**() 車線名データ数を返す
java.lang.String	**getKaishiKp**() 開始ＫＰを返す
java.lang.String	**getRosenMei**() 路線名を返す
int	**getSyasenId**() 車線ＩＤを返す
int	**getSyasenIdTbl**(int dataIndex) 車線ＩＤを返す
java.lang.String	**getSyasenMei**() 車線名を返す
java.lang.String	**getSyasenMeiTbl**(int dataIndex) 車線名を返す
java.lang.String	**getSyuryoKp**() 終了ＫＰを返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

syasenIdTbl

private int[] **syasenIdTbl**

車線ＩＤ配列

---

syasenMeiTbl

private java.lang.String[] **syasenMeiTbl**

車線名配列

---

syasenCnt

private int **syasenCnt**

車線数

---

dataCnt

private int **dataCnt**

データ数

---

rosenMei

private java.lang.String **rosenMei**

路線名

---

syasenId

private int **syasenId**

車線ＩＤ

---

syasenMei

private java.lang.String **syasenMei**

車線名

---

kaishiKp

private double **kaishiKp**

開始ＫＰ

---

syuryoKp

private double **syuryoKp**

終了ＫＰ

コンストラクタの詳細

AdminMeshRegist

public **AdminMeshRegist**(java.lang.String id,
java.sql.Connection dbConn,
CommonValues commonValues,
javax.servlet.http.HttpServletRequest request)

コンストラクタ

パラメータ:

id - アクションＩＤ

dbConn - Connection クラス

commonValues - CommonValues クラス

request - HttpServletRequest クラス

メソッドの詳細

getRosenMei

public java.lang.String **getRosenMei**()

路線名を返す

戻り値:

路線名

---

getSyasenId

public int **getSyasenId**()

車線ＩＤを返す

戻り値:

車線ＩＤ

---

getSyasenMei

public java.lang.String **getSyasenMei**()

車線名を返す

戻り値:

車線名

---

getKaishiKp

public java.lang.String **getKaishiKp**()

開始ＫＰを返す

戻り値:

開始ＫＰ

---

getSyuryoKp

public java.lang.String **getSyuryoKp**()

終了ＫＰを返す

戻り値:

終了ＫＰ

---

getSyasenIdTbl

public int **getSyasenIdTbl**(int dataIndex)

車線ＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

車線ＩＤ

---

getSyasenMeiTbl

public java.lang.String **getSyasenMeiTbl**(int dataIndex)

車線名を返す

パラメータ:

dataIndex - インデックス

戻り値:

車線名

---

getDataCnt

public int **getDataCnt**()

車線名データ数を返す

戻り値:

車線名データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

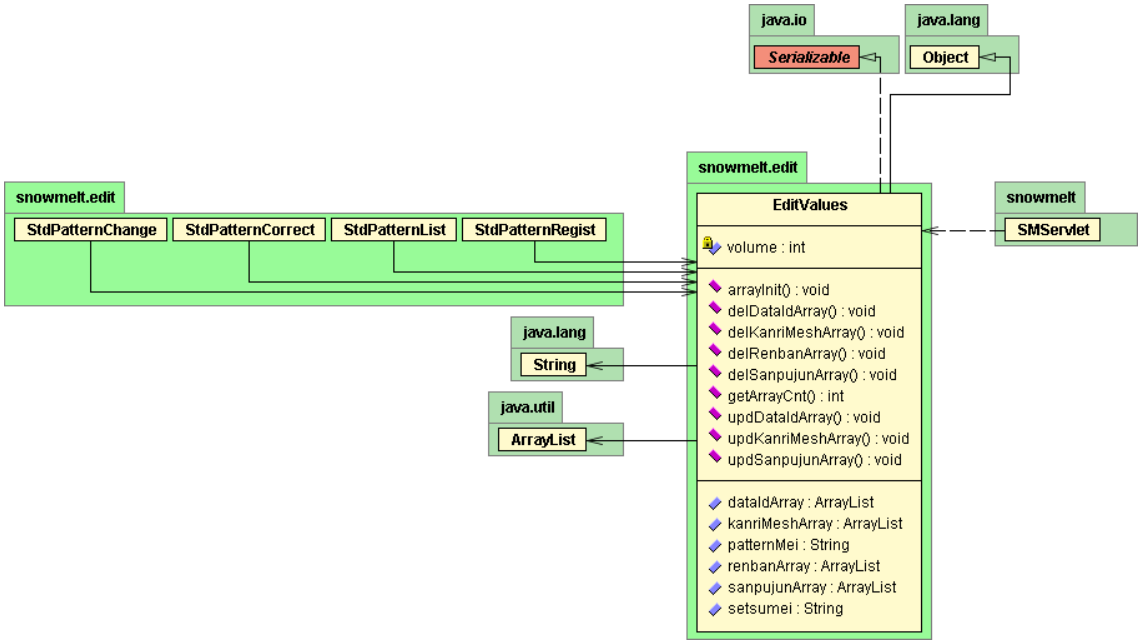
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/D13.jsp"

■ 4　クラス EditValues



java.lang.Object

　|

　+--**snowmelt.edit.EditValues**

すべての実装インタフェース:

java.io.Serializable

---

public class **EditValues**

extends java.lang.Object

implements java.io.Serializable

標準散布パターン作成　共通変数格納クラス

フィールドの概要	
private　java.util.ArrayList	**dataIdArray** データＩＤ
private　java.util.ArrayList	**kanriMeshArray** 管理メッシュ
private　java.lang.String	**patternMei** パターン名

private java.util.ArrayList	**renbanArray** 連番
private java.util.ArrayList	**sanpujunArray** 散布順
private java.lang.String	**setsumei** 説明
private int	**volume** 総散布量

コンストラクタの概要

EditValues()	

メソッドの概要

void	**arrayInit**() 配列初期化
void	**delDataIdArray**(int tblIdx) データＩＤ削除
void	**delKanriMeshArray**(int tblIdx) 管理メッシュ削除
void	**delRenbanArray**(int tblIdx) 連番削除
void	**delSanpujunArray**(int tblIdx) 散布順削除
int	**getArrayCnt**() 配列要素数を返す

java.lang.String	**getDataIdArray**(int tblIdx) データＩＤゲッター
java.lang.String	**getKanriMeshArray**(int tblIdx) 管理メッシュゲッター
java.lang.String	**getPatternMei**() パターン名ゲッター
java.lang.String	**getRenbanArray**(int tblIdx) 連番ゲッター
java.lang.String	**getSanpujunArray**(int tblIdx) 散布順ゲッター
java.lang.String	**getSetsumei**() 説明ゲッター
void	**setDataIdArray**(java.lang.String PrmData) データＩＤセッター
void	**setKanriMeshArray**(java.lang.String PrmData) 管理メッシュセッター
void	**setPatternMei**(java.lang.String PrmData) パターン名セッター
void	**setRenbanArray**(java.lang.String PrmData) 連番セッター
void	**setSanpujunArray**(java.lang.String PrmData) 散布順セッター
void	**setSetsumei**(java.lang.String PrmData) 説明セッター
void	**updDataIdArray**(int tblIdx, java.lang.String PrmData) データＩＤ変更
void	**updKanriMeshArray**(int tblIdx, java.lang.String PrmData) 管理メッシュ変更
void	**updSanpujunArray**(int tblIdx, java.lang.String PrmData) 散布順変更

フィールドの詳細

patternMei

private java.lang.String **patternMei**

パターン名

---

setsumei

private java.lang.String **setsumei**

説明

---

volume

private int **volume**

総散布量

---

renbanArray

private java.util.ArrayList **renbanArray**

連番

---

kanriMeshArray

private java.util.ArrayList **kanriMeshArray**

管理メッシュ

---

dataIdArray

private java.util.ArrayList **dataIdArray**

データＩＤ

---

sanpujunArray

private java.util.ArrayList **sanpujunArray**

散布順

コンストラクタの詳細

EditValues

public **EditValues**()

メソッドの詳細

arrayInit

public void **arrayInit**()

配列初期化

---

getArrayCnt

public int **getArrayCnt**()

配列要素数を返す

戻り値:

配列要素数

---

getPatternMei

public java.lang.String **getPatternMei**()

パターン名ゲッター

戻り値:

パターン名

---

setPatternMei

public void **setPatternMei**(java.lang.String PrmData)

パターン名セッター

パラメータ:

PrmData - パターン名

---

getSetsumei

public java.lang.String **getSetsumei**()

説明ゲッター

戻り値:

説明

---

setSetsumei

public void **setSetsumei**(java.lang.String PrmData)

説明セッター

パラメータ:

PrmData - 説明

---

getRenbanArray

public java.lang.String **getRenbanArray**(int tblIdx)

連番ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

連番

---

setRenbanArray

public void **setRenbanArray**(java.lang.String PrmData)

連番セッター

パラメータ:

PrmData - 連番

delRenbanArray

public void **delRenbanArray**(int tblIdx)

連番削除

パラメータ:

tblIdx - 連番

---

getKanriMeshArray

public java.lang.String **getKanriMeshArray**(int tblIdx)

管理メッシュゲッター

パラメータ:

tblIdx - インデックス

戻り値:

管理メッシュ

---

setKanriMeshArray

public void **setKanriMeshArray**(java.lang.String PrmData)

管理メッシュセッター

パラメータ:

PrmData - 管理メッシュ

---

updKanriMeshArray

public void **updKanriMeshArray**(int tblIdx,
java.lang.String PrmData)

管理メッシュ変更

パラメータ:

tblIdx - インデックス

PrmData - 管理メッシュ

---

delKanriMeshArray

public void **delKanriMeshArray**(int tblIdx)

管理メッシュ削除

パラメータ:

tblIdx - インデックス

---

getDataIdArray

public java.lang.String **getDataIdArray**(int tblIdx)

データＩＤゲッター

パラメータ:

tblIdx - インデックス

戻り値:

データＩＤ

---

setDataIdArray

public void **setDataIdArray**(java.lang.String PrmData)

データＩＤセッター

パラメータ:

PrmData - データＩＤ

---

updDataIdArray

public void **updDataIdArray**(int tblIdx,
java.lang.String PrmData)

データＩＤ変更

パラメータ:

tblIdx - インデックス

PrmData - データＩＤ

---

delDataIdArray

public void **delDataIdArray**(int tblIdx)

データＩＤ削除

パラメータ:

tblIdx - インデックス

---

getSanpujunArray

public java.lang.String **getSanpujunArray**(int tblIdx)

散布順ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

散布順

---

setSanpujunArray

public void **setSanpujunArray**(java.lang.String PrmData)

散布順セッター

パラメータ:

PrmData - 散布順

---

updSanpujunArray

public void **updSanpujunArray**(int tblIdx,
java.lang.String PrmData)

散布順変更

パラメータ:

tblIdx - インデックス

PrmData - 散布順

---

delSanpujunArray

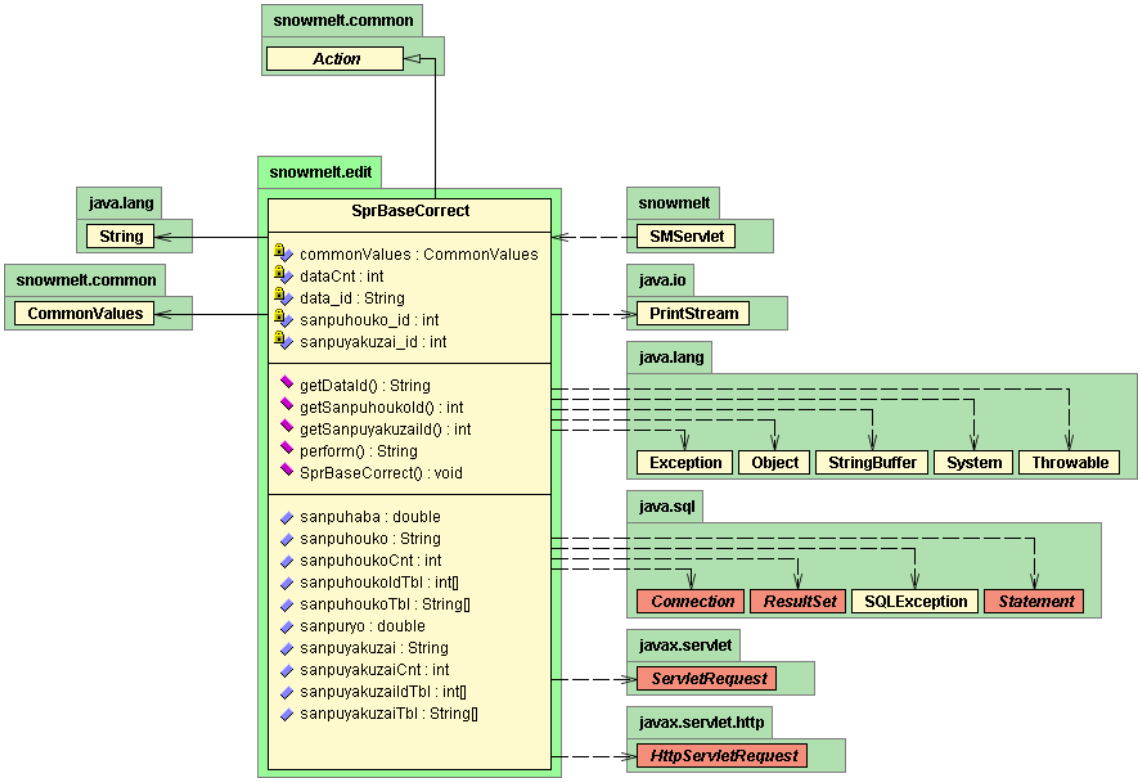
public void **delSanpujunArray**(int tblIdx)

散布順削除

パラメータ:

`tblIdx` - インデックス

■ ５　クラス SprBaseCorrect



java.lang.Object

　|

　+--snowmelt.common.Action

　　　|

　　　+--**snowmelt.edit.SprBaseCorrect**

---

public class **SprBaseCorrect**

extends Action

散布基本データ修正クラス

フィールドの概要	
private　CommonValues	**commonValues** 共通変数クラスのインスタンス
private　java.lang.String	**data_id** データＩＤ

private int	**dataCnt** データ数
private double	**sanpuhaba** 散布幅
private java.lang.String	**sanpuhouko** 散布方向
private int	**sanpuhouko_id** 散布方向ＩＤ
private int	**sanpuhoukoCnt** 散布方向数
private int[]	**sanpuhoukoIdTbl** 散布方向ＩＤ配列
private java.lang.String[]	**sanpuhoukoTbl** 散布方向配列
private double	**sanpuryo** 散布量
private java.lang.String	**sanpuyakuzai** 散布薬剤
private int	**sanpuyakuzai_id** 散布薬剤ＩＤ
private int	**sanpuyakuzaiCnt** 散布薬剤数
private int[]	**sanpuyakuzaiIdTbl** 散布薬剤ＩＤ配列
private java.lang.String[]	**sanpuyakuzaiTbl** 散布薬剤配列

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

SprBaseCorrect(java.sql.Connection dbConn, CommonValues commonValues, javax.servlet.http.HttpServletRequest request) コンストラクタ

メソッドの概要

java.lang.String	**getDataId**() データＩＤを返す
double	**getSanpuhaba**() 散布幅を返す
java.lang.String	**getSanpuhouko**() 散布方向を返す
int	**getSanpuhoukoCnt**() 散布方向データ数を返す
int	**getSanpuhoukoId**() 散布方向ＩＤを返す
int	**getSanpuhoukoIdTbl**(int dataIndex) 散布方向ＩＤを返す
java.lang.String	**getSanpuhoukoTbl**(int dataIndex) 散布方向名を返す
double	**getSanpuryo**() 散布量を返す
java.lang.String	**getSanpuyakuzai**() 散布薬剤を返す
int	**getSanpuyakuzaiCnt**() 散布薬剤データ数を返す

int	**getSanpuyakuzaiId**() 散布薬剤ＩＤを返す
int	**getSanpuyakuzaiIdTbl**(int dataIndex) 散布薬剤ＩＤを返す
java.lang.String	**getSanpuyakuzaiTbl**(int dataIndex) 散布薬剤名を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

sanpuhoukoCnt

private int **sanpuhoukoCnt**

散布方向数

---

sanpuhoukoIdTbl

private int[] **sanpuhoukoIdTbl**

散布方向ＩＤ配列

---

sanpuhoukoTbl

private java.lang.String[] **sanpuhoukoTbl**

散布方向配列

---

sanpuyakuzaiCnt

private int **sanpuyakuzaiCnt**

散布薬剤数

---

sanpuyakuzaiIdTbl

private int[] **sanpuyakuzaiIdTbl**

散布薬剤ＩＤ配列

---

sanpuyakuzaiTbl

private java.lang.String[] **sanpuyakuzaiTbl**

散布薬剤配列

---

dataCnt

private int **dataCnt**

データ数

---

data_id

private java.lang.String **data_id**

データＩＤ

---

sanpuryo

private double **sanpuryo**

散布量

sanpuhaba

private double **sanpuhaba**

散布幅

---

sanpuhouko_id

private int **sanpuhouko_id**

散布方向ＩＤ

---

sanpuhouko

private java.lang.String **sanpuhouko**

散布方向

---

sanpuyakuzai_id

private int **sanpuyakuzai_id**

散布薬剤ＩＤ

---

sanpuyakuzai

private java.lang.String **sanpuyakuzai**

散布薬剤

コンストラクタの詳細

SprBaseCorrect

public **SprBaseCorrect**(java.sql.Connection dbConn,
CommonValues commonValues,
javax.servlet.http.HttpServletRequest request)

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonValues - CommonValues クラス

request - HttpServletRequest クラス

メソッドの詳細

getDataId

public java.lang.String **getDataId**()

データＩＤを返す

戻り値:

データＩＤ

---

getSanpuryo

public double **getSanpuryo**()

散布量を返す

戻り値:

散布量

---

getSanpuhaba

public double **getSanpuhaba**()

散布幅を返す

戻り値:

散布幅

---

getSanpuhoukoId

public int **getSanpuhoukoId**()

散布方向ＩＤを返す

戻り値:

散布方向ＩＤ

---

getSanpuhouko

public java.lang.String **getSanpuhouko**()

散布方向を返す

戻り値:

散布方向

---

getSanpuyakuzaiId

public int **getSanpuyakuzaiId**()

散布薬剤ＩＤを返す

戻り値:

散布薬剤ＩＤ

---

getSanpuyakuzai

public java.lang.String **getSanpuyakuzai**()

散布薬剤を返す

戻り値:

散布薬剤

---

getSanpuhoukoIdTbl

public int **getSanpuhoukoIdTbl**(int dataIndex)

散布方向ＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

散布方向ＩＤ

---

getSanpuhoukoTbl

public java.lang.String **getSanpuhoukoTbl**(int dataIndex)

散布方向名を返す

パラメータ:

dataIndex - インデックス

戻り値:

散布方向名

---

getSanpuhoukoCnt

public int **getSanpuhoukoCnt**()

散布方向データ数を返す

戻り値:

散布方向データ数

---

getSanpuyakuzaiIdTbl

public int **getSanpuyakuzaiIdTbl**(int dataIndex)

散布薬剤ＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

散布薬剤ＩＤ

---

getSanpuyakuzaiTbl

public java.lang.String **getSanpuyakuzaiTbl**(int dataIndex)

散布薬剤名を返す

パラメータ:

dataIndex - インデックス

戻り値:

散布薬剤名

---

getSanpuyakuzaiCnt

public int **getSanpuyakuzaiCnt**()

散布薬剤データ数を返す

戻り値:

散布薬剤データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

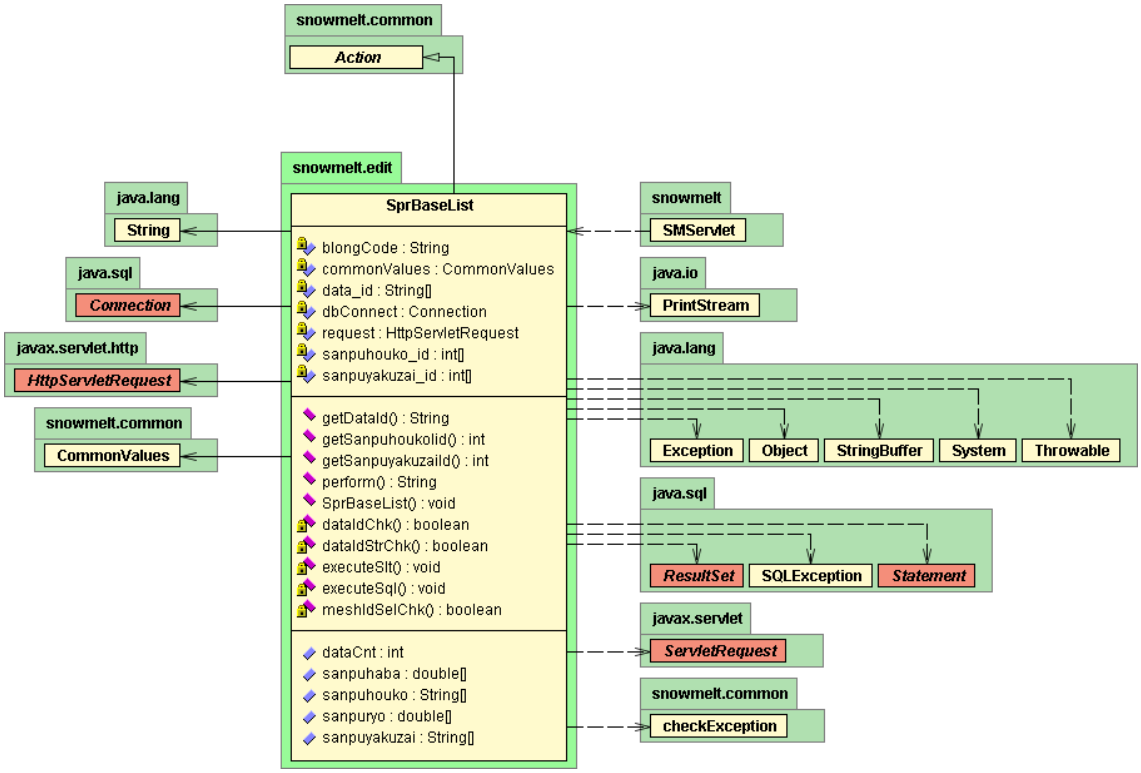
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/D22.jsp"

■ ６　クラス SprBaseList



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.edit.SprBaseList**

---

public class **SprBaseList**

extends Action

散布基本データ修正クラス

フィールドの概要

private　java.lang.String	**blongCode** 所属コード
private　CommonValues	**commonValues** 共通変数クラスのインスタンス

private java.lang.String[]	**data_id** データＩＤ
private int	**dataCnt** データ数
private java.sql.Connection	**dbConnect** Connection クラス
private javax.servlet.http.HttpServletRequest	**request** HttpServletRequest クラス
private double[]	**sanpuhaba** 散布幅
private java.lang.String[]	**sanpuhouko** 散布方向
private int[]	**sanpuhouko_id** 散布方向ＩＤ
private double[]	**sanpuryo** 散布量
private java.lang.String[]	**sanpuyakuzai** 散布薬剤
private int[]	**sanpuyakuzai_id** 散布薬剤ＩＤ

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

SprBaseList(java.lang.String id, java.sql.Connection dbConn, CommonValues commonVal, javax.servlet.http.HttpServletRequest req)

コンストラクタ

メソッドの概要

private boolean	**dataIdChk**(java.lang.String dataId) データＩＤ重複チェック
private boolean	**dataIdStrChk**(java.lang.String dataId) データＩＤ
private void	**executeSlt**() データ取得
private void	**executeSql**(java.lang.String id) データ更新
int	**getDataCnt**() データ数を返す
java.lang.String	**getDataId**(int dataIndex) データＩＤを返す
double	**getSanpuhaba**(int dataIndex) 散布幅を返す
java.lang.String	**getSanpuhouko**(int dataIndex) 散布方向を返す
int	**getSanpuhoukoId**(int dataIndex) 散布方向ＩＤを返す
double	**getSanpuryo**(int dataIndex) 散布量を返す
java.lang.String	**getSanpuyakuzai**(int dataIndex) 散布薬剤を返す
int	**getSanpuyakuzaiId**(int dataIndex) 散布薬剤ＩＤを返す
private boolean	**meshIdSelChk**(java.lang.String blongCode, java.lang.String dataId)

	参照整合性チェック
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

dbConnect

private java.sql.Connection **dbConnect**

Connection クラス

---

request

private javax.servlet.http.HttpServletRequest **request**

HttpServletRequest クラス

---

blongCode

private java.lang.String **blongCode**

所属コード

---

dataCnt

private int **dataCnt**

　　データ数

---

data_id

private java.lang.String[] **data_id**

　　データＩＤ

---

sanpuryo

private double[] **sanpuryo**

　　散布量

---

sanpuhaba

private double[] **sanpuhaba**

　　散布幅

---

sanpuhouko_id

private int[] **sanpuhouko_id**

　　散布方向ＩＤ

---

sanpuhouko

private java.lang.String[] **sanpuhouko**

　　散布方向

---

sanpuyakuzai_id

private int[] **sanpuyakuzai_id**

　　散布薬剤ＩＤ

sanpuyakuzai

private java.lang.String[] **sanpuyakuzai**

散布薬剤

コンストラクタの詳細

SprBaseList

public **SprBaseList**(java.lang.String id,
java.sql.Connection dbConn,
CommonValues commonVal,
javax.servlet.http.HttpServletRequest req)
throws checkException

コンストラクタ

パラメータ:

id - アクションＩＤ

dbConn - Connection クラス

commonVal - CommonValues クラス

req - HttpServletRequest クラス

例外:

checkException -

メソッドの詳細

executeSql

private void **executeSql**(java.lang.String id)
throws checkException

データ更新

パラメータ:

id - アクションＩＤ

例外:

checkException -

---

dataIdStrChk

private boolean **dataIdStrChk**(java.lang.String dataId)

データＩＤ

パラメータ:

prmStr - データＩＤ

戻り値:

true:エラーなし false:エラーあり

---

dataIdChk

private boolean **dataIdChk**(java.lang.String dataId)

データＩＤ重複チェック

パラメータ:

dataId - データＩＤ

戻り値:

true:重複 false:重複なし

---

meshIdSelChk

private boolean **meshIdSelChk**(java.lang.String blongCode,
java.lang.String dataId)

参照整合性チェック

パラメータ:

blongCode - 所属コード

dataId - データＩＤ

戻り値:

true:削除 OK false:削除 NG

---

executeSlt

private void **executeSlt**()

データ取得

---

getDataId

public java.lang.String **getDataId**(int dataIndex)

データＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

データＩＤ

---

getSanpuryo

public double **getSanpuryo**(int dataIndex)

散布量を返す

パラメータ:

dataIndex - インデックス

戻り値:

散布量

---

getSanpuhaba

public double **getSanpuhaba**(int dataIndex)

散布幅を返す

パラメータ:

dataIndex - インデックス

戻り値:

散布幅

---

getSanpuhoukoIid

public int **getSanpuhoukolid**(int dataIndex)

散布方向ＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

散布方向ＩＤ

---

getSanpuhouko

public java.lang.String **getSanpuhouko**(int dataIndex)

散布方向を返す

パラメータ:

dataIndex - インデックス

戻り値:

散布方向

---

getSanpuyakuzaiId

public int **getSanpuyakuzaiId**(int dataIndex)

散布薬剤ＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

散布薬剤ＩＤ

---

getSanpuyakuzai

public java.lang.String **getSanpuyakuzai**(int dataIndex)

散布薬剤を返す

パラメータ:

dataIndex - インデックス

戻り値:

散布薬剤

---

getDataCnt

public int **getDataCnt**()

データ数を返す

戻り値:

データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

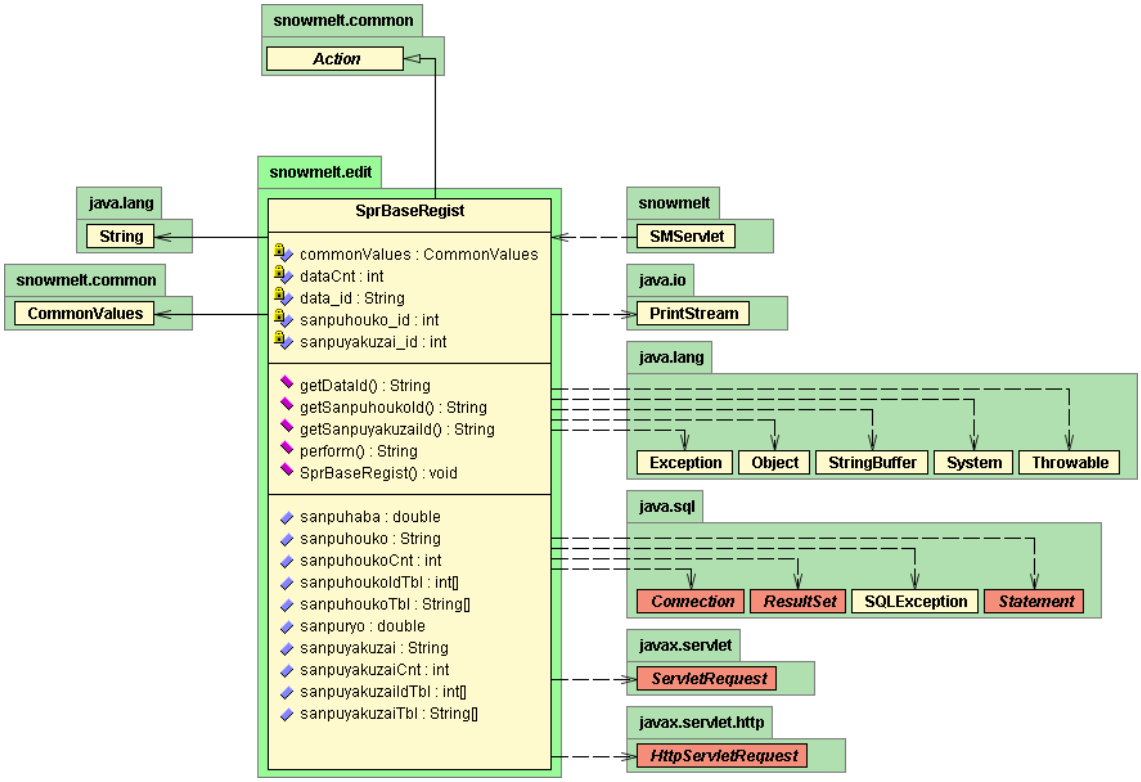
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/D21.jsp"

■ 7　クラス SprBaseRegist



java.lang.Object

　|

　+--snowmelt.common.Action

　　|

　　+--**snowmelt.edit.SprBaseRegist**

---

public class **SprBaseRegist**

extends Action

散布基本データ新規登録クラス

フィールドの概要	
private　CommonValues	**commonValues** 共通変数クラスのインスタンス
private　java.lang.String	**data_id** データＩＤ

private int	**dataCnt** データ数
private double	**sanpuhaba** 散布幅
private java.lang.String	**sanpuhouko** 散布方向
private int	**sanpuhouko_id** 散布方向ＩＤ
private int	**sanpuhoukoCnt** 散布方向数
private int[]	**sanpuhoukoIdTbl** 散布方向ＩＤ配列
private java.lang.String[]	**sanpuhoukoTbl** 散布方向配列
private double	**sanpuryo** 散布量
private java.lang.String	**sanpuyakuzai** 散布薬剤
private int	**sanpuyakuzai_id** 散布薬剤ＩＤ
private int	**sanpuyakuzaiCnt** 散布薬剤数
private int[]	**sanpuyakuzaiIdTbl** 散布薬剤ＩＤ配列
private java.lang.String[]	**sanpuyakuzaiTbl** 散布薬剤配列

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

SprBaseRegist(java.lang.String id, java.sql.Connection dbConn, CommonValues commonValues, javax.servlet.http.HttpServletRequest request) コンストラクタ

メソッドの概要

java.lang.String	**getDataId**() データＩＤを返す
java.lang.String	**getSanpuhaba**() 散布幅を返す
java.lang.String	**getSanpuhouko**() 散布方向を返す
int	**getSanpuhoukoCnt**() 全散布方向データ数を返す
java.lang.String	**getSanpuhoukoId**() 散布方向ＩＤを返す
java.lang.String	**getSanpuhoukoIdTbl**(int dataIndex) 全散布方向ＩＤを返す
java.lang.String	**getSanpuhoukoTbl**(int dataIndex) 全散布方向名を返す
java.lang.String	**getSanpuryo**() 散布量を返す
java.lang.String	**getSanpuyakuzai**() 散布薬剤を返す
int	**getSanpuyakuzaiCnt**() 全散布薬剤データ数を返す

java.lang.String	**getSanpuyakuzaiId**() 散布薬剤ＩＤを返す
java.lang.String	**getSanpuyakuzaiIdTbl**(int dataIndex) 全散布薬剤ＩＤを返す
java.lang.String	**getSanpuyakuzaiTbl**(int dataIndex) 全散布薬剤名を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

sanpuhoukoCnt

private int **sanpuhoukoCnt**

散布方向数

---

sanpuhoukoIdTbl

private int[] **sanpuhoukoIdTbl**

散布方向ＩＤ配列

---

sanpuhoukoTbl

private java.lang.String[] **sanpuhoukoTbl**

散布方向配列

---

sanpuyakuzaiCnt

private int **sanpuyakuzaiCnt**

散布薬剤数

---

sanpuyakuzaiIdTbl

private int[] **sanpuyakuzaiIdTbl**

散布薬剤ＩＤ配列

---

sanpuyakuzaiTbl

private java.lang.String[] **sanpuyakuzaiTbl**

散布薬剤配列

---

dataCnt

private int **dataCnt**

データ数

---

data_id

private java.lang.String **data_id**

データＩＤ

---

sanpuryo

private double **sanpuryo**

散布量

sanpuhaba

private double **sanpuhaba**

散布幅

---

sanpuhouko_id

private int **sanpuhouko_id**

散布方向ＩＤ

---

sanpuhouko

private java.lang.String **sanpuhouko**

散布方向

---

sanpuyakuzai_id

private int **sanpuyakuzai_id**

散布薬剤ＩＤ

---

sanpuyakuzai

private java.lang.String **sanpuyakuzai**

散布薬剤

コンストラクタの詳細

SprBaseRegist

public **SprBaseRegist**(java.lang.String id,
java.sql.Connection dbConn,
CommonValues commonValues,

javax.servlet.http.HttpServletRequest request)

コンストラクタ

パラメータ:

id - アクションＩＤ

dbConn - Connection クラス

commonValues - CommonValues クラス

request - HttpServletRequest クラス

メソッドの詳細

getDataId

public java.lang.String **getDataId**()

データＩＤを返す

戻り値:

データＩＤ

---

getSanpuryo

public java.lang.String **getSanpuryo**()

散布量を返す

戻り値:

散布量

---

getSanpuhaba

public java.lang.String **getSanpuhaba**()

散布幅を返す

戻り値:

散布幅

---

getSanpuhoukoId

public java.lang.String **getSanpuhoukoId**()

散布方向ＩＤを返す

戻り値:

散布方向ＩＤ

---

getSanpuhouko

public java.lang.String **getSanpuhouko**()

散布方向を返す

戻り値:

散布方向

---

getSanpuyakuzaiId

public java.lang.String **getSanpuyakuzaiId**()

散布薬剤ＩＤを返す

戻り値:

散布薬剤ＩＤ

---

getSanpuyakuzai

public java.lang.String **getSanpuyakuzai**()

散布薬剤を返す

戻り値:

散布薬剤

---

getSanpuhoukoIdTbl

public java.lang.String **getSanpuhoukoIdTbl**(int dataIndex)

全散布方向ＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

全散布方向ＩＤ

---

getSanpuhoukoTbl

public java.lang.String **getSanpuhoukoTbl**(int dataIndex)

全散布方向名を返す

パラメータ:

dataIndex - インデックス

戻り値:

全散布方向名

---

getSanpuhoukoCnt

public int **getSanpuhoukoCnt**()

全散布方向データ数を返す

戻り値:

全散布方向データ数

---

getSanpuyakuzaiIdTbl

public java.lang.String **getSanpuyakuzaiIdTbl**(int dataIndex)

全散布薬剤ＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

全散布薬剤ＩＤ

---

getSanpuyakuzaiTbl

public java.lang.String **getSanpuyakuzaiTbl**(int dataIndex)

全散布薬剤名を返す

パラメータ:

dataIndex - インデックス

戻り値:

全散布薬剤名

---

getSanpuyakuzaiCnt

public int **getSanpuyakuzaiCnt**()

全散布薬剤データ数を返す

戻り値:

全散布薬剤データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

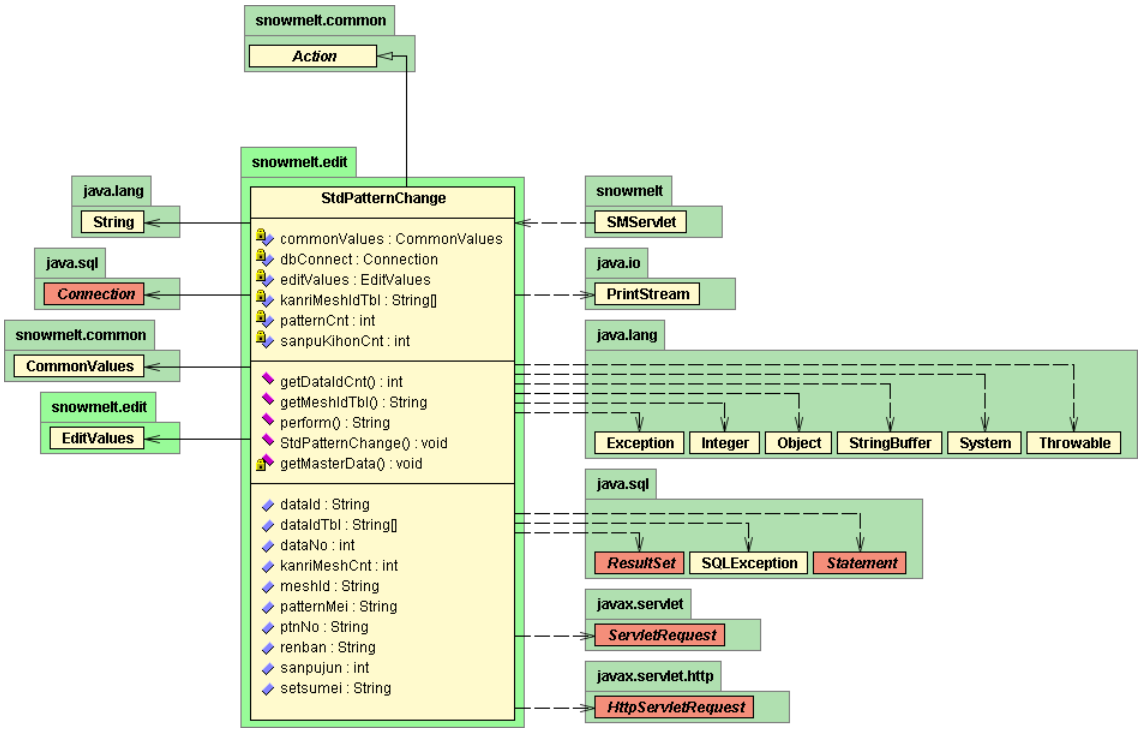
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/D23.jsp"

■ 8　クラス StdPatternChange



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.edit.StdPatternChange**

---

public class **StdPatternChange**

extends Action

管理メッシュ・散布基本データの修正・削除クラス

フィールドの概要	
private CommonValues	**commonValues** 共通変数クラスのインスタンス
private java.lang.String	**dataId** データ ID
private java.lang.String[]	**dataIdTbl**

	データ ID 配列
private int	**dataNo** データ番号
private java.sql.Connection	**dbConnect** Connection クラス
private EditValues	**editValues** 入力値維持クラスのインスタンス
private int	**kanriMeshCnt** 管理メッシュ取得用変数
private java.lang.String[]	**kanriMeshIdTbl** 管理メッシュ ID 配列
private java.lang.String	**meshId** メッシュ ID
private int	**patternCnt** 標準パターン数
private java.lang.String	**patternMei** 標準パターン名
private java.lang.String	**ptnNo** 標準パターン番号
private java.lang.String	**renban** 連番
private int	**sanpujun** 散布順
private int	**sanpuKihonCnt** 散布基本データＩＤ取得用変数
private java.lang.String	**setsumei** 標準パターン説明

クラス snowmelt.common.Action から継承したフィールド

servlet

コンストラクタの概要

StdPatternChange(java.sql.Connection dbConn, CommonValues commonVal, EditValues editVal, javax.servlet.http.HttpServletRequest request)

コンストラクタ

メソッドの概要

java.lang.String	**getDataId**() データ ID を返す
int	**getDataIdCnt**() 全散布基本データＩＤ数を返す
java.lang.String	**getDataIdTbl**(int dataIndex) 散布基本データＩＤを返す
java.lang.String	**getDataNo**() 選択データ番号を返す
int	**getKanriMeshCnt**() 全管理メッシュＩＤ数を返す
private void	**getMasterData**() マスターデータ取得
java.lang.String	**getMeshId**() メッシュ ID を返す
java.lang.String	**getMeshIdTbl**(int dataIndex) 管理メッシュＩＤ返す
java.lang.String	**getPatternMei**() 標準パターン名を返す

java.lang.String	**getPtnNo**() パターン番号を返す
java.lang.String	**getRenban**() 連番を返す
int	**getSanpujun**() 散布順を返す
java.lang.String	**getSetsumei**() 標準パターン説明を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

editValues

private EditValues **editValues**

入力値維持クラスのインスタンス

---

dbConnect

private java.sql.Connection **dbConnect**

---

kanriMeshCnt

private int **kanriMeshCnt**

管理メッシュ取得用変数

---

kanriMeshIdTbl

private java.lang.String[] **kanriMeshIdTbl**

管理メッシュ ID 配列

---

sanpuKihonCnt

private int **sanpuKihonCnt**

散布基本データＩＤ取得用変数

---

dataIdTbl

private java.lang.String[] **dataIdTbl**

データ ID 配列

---

patternCnt

private int **patternCnt**

標準パターン数

---

patternMei

private java.lang.String **patternMei**

標準パターン名

---

setsumei

private java.lang.String **setsumei**

標準パターン説明

---

ptnNo

private java.lang.String **ptnNo**

標準パターン番号

---

renban

private java.lang.String **renban**

連番

---

meshId

private java.lang.String **meshId**

メッシュ ID

---

dataId

private java.lang.String **dataId**

データ ID

---

sanpujun

private int **sanpujun**

散布順

---

dataNo

private int **dataNo**

データ番号

コンストラクタの詳細

StdPatternChange

public **StdPatternChange**(java.sql.Connection dbConn,
CommonValues commonVal,
EditValues editVal,
javax.servlet.http.HttpServletRequest request)

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonVal - CommonValues クラス

editVal - EditValues クラス

request - HttpServletRequest クラス

メソッドの詳細

getMasterData

private void **getMasterData**()

マスターデータ取得

---

getPatternMei

public java.lang.String **getPatternMei**()

標準パターン名を返す

---

getSetsumei

public java.lang.String **getSetsumei**()

標準パターン説明を返す

戻り値:

標準パターン説明

---

getPtnNo

public java.lang.String **getPtnNo**()

パターン番号を返す

戻り値:

パターン番号

---

getRenban

public java.lang.String **getRenban**()

連番を返す

戻り値:

連番

---

getMeshId

public java.lang.String **getMeshId**()

メッシュ ID を返す

戻り値:

メッシュ ID

---

getDataId

public java.lang.String **getDataId**()

データ ID を返す

戻り値:

データ ID

---

getSanpujun

public int **getSanpujun**()

散布順を返す

---

getDataNo

public java.lang.String **getDataNo**()

選択データ番号を返す

戻り値:

選択データ番号

---

getMeshIdTbl

public java.lang.String **getMeshIdTbl**(int dataIndex)

管理メッシュＩＤ返す

パラメータ:

dataIndex - インデックス

戻り値:

管理メッシュＩＤ

---

getKanriMeshCnt

public int **getKanriMeshCnt**()

全管理メッシュＩＤ数を返す

戻り値:

全管理メッシュＩＤ数

---

getDataIdTbl

public java.lang.String **getDataIdTbl**(int dataIndex)

散布基本データＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

散布基本データＩＤ

---

getDataIdCnt

public int **getDataIdCnt**()

全散布基本データＩＤ数を返す

戻り値:

全散布基本データＩＤ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

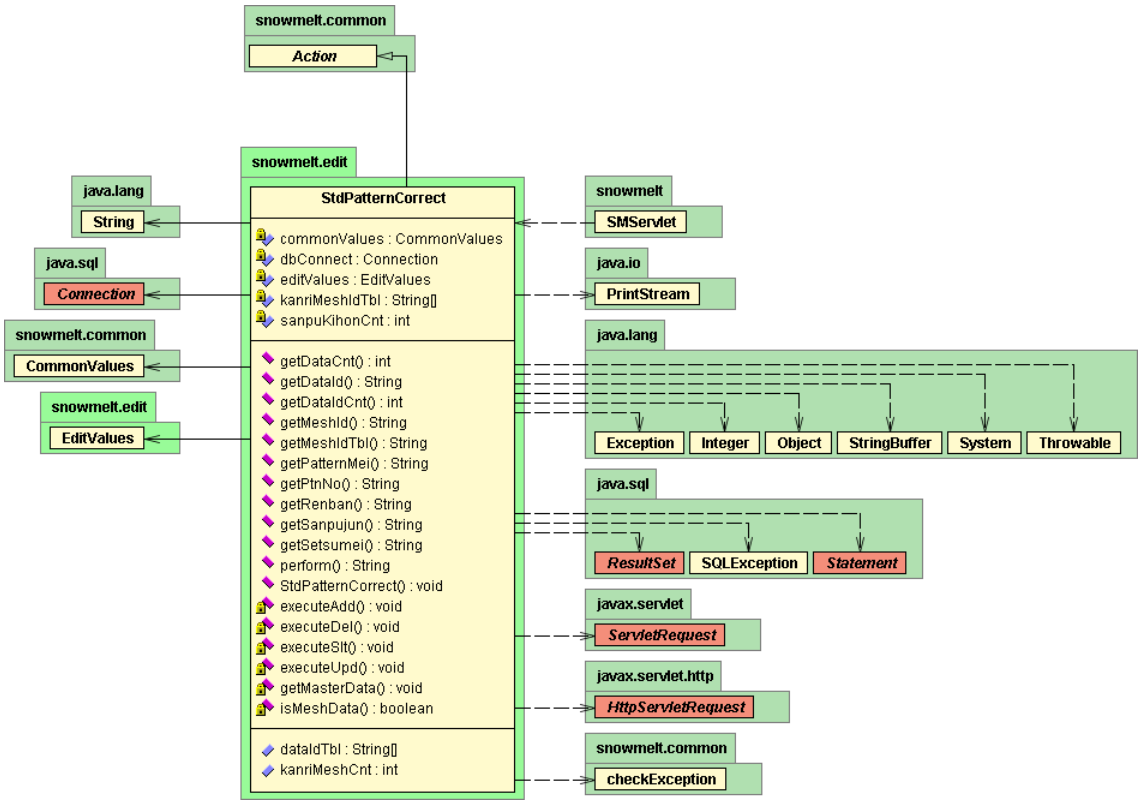
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/D34.jsp"

■ ９　クラス StdPatternCorrect



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.edit.StdPatternCorrect**

---

public class **StdPatternCorrect**

extends Action

標準散布パターン修正クラス

フィールドの概要	
private　CommonValues	**commonValues** 共通変数クラスのインスタンス
private　java.lang.String[]	**dataIdTbl** データ ID 配列

private java.sql.Connection	**dbConnect** Connection クラス
private EditValues	**editValues** 入力値維持クラスのインスタンス
private int	**kanriMeshCnt** 管理メッシュ取得用変数
private java.lang.String[]	**kanriMeshIdTbl** 管理メッシュ ID 配列
private int	**sanpuKihonCnt** 散布基本データＩＤ取得用変数

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

StdPatternCorrect(java.lang.String id, java.sql.Connection dbConn, CommonValues commonVal, EditValues editVal, javax.servlet.http.HttpServletRequest request)
コンストラクタ

メソッドの概要

private void	**executeAdd**(javax.servlet.http.HttpServletRequest request) データ追加処理
private void	**executeDel**(javax.servlet.http.HttpServletRequest request) データ削除処理

private void	**executeSlt**() データ取得
private void	**executeUpd**(javax.servlet.http.HttpServletRequest request) データ更新処理
int	**getDataCnt**() 標準パターン組合せ数を返す
java.lang.String	**getDataId**(int dataIndex) データＩＤを返す
int	**getDataIdCnt**() 全散布基本データＩＤ数を返す
java.lang.String	**getDataIdTbl**(int dataIndex) 全散布基本データＩＤ返す
int	**getKanriMeshCnt**() 全管理メッシュＩＤ数を返す
private void	**getMasterData**() マスターデータ取得
java.lang.String	**getMeshId**(int dataIndex) メッシュを返す
java.lang.String	**getMeshIdTbl**(int dataIndex) 全管理メッシュＩＤ返す
java.lang.String	**getPatternMei**() 標準パターン名を返す
java.lang.String	**getPtnNo**() パターン番号を返す
java.lang.String	**getRenban**(int dataIndex) 連番を返す
java.lang.String	**getSanpujun**(int dataIndex) 散布順を返す
java.lang.String	**getSetsumei**() 標準パターン説明を返す
private boolean	**isMeshData**(java.lang.String pramKanriMesh,

	java.lang.String pramSpjyn, int renban) 散布順重複エラーチェック
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

editValues

private EditValues **editValues**

入力値維持クラスのインスタンス

---

dbConnect

private java.sql.Connection **dbConnect**

Connection クラス

---

kanriMeshCnt

private int **kanriMeshCnt**

管理メッシュ取得用変数

---

kanriMeshIdTbl

private java.lang.String[] **kanriMeshIdTbl**

管理メッシュ ID 配列

---

sanpuKihonCnt

private int **sanpuKihonCnt**

散布基本データＩＤ取得用変数

---

dataIdTbl

private java.lang.String[] **dataIdTbl**

データ ID 配列

コンストラクタの詳細

StdPatternCorrect

public **StdPatternCorrect**(java.lang.String id,
java.sql.Connection dbConn,
CommonValues commonVal,
EditValues editVal,
javax.servlet.http.HttpServletRequest request)
throws checkException

コンストラクタ

パラメータ:

id - アクション ID

dbConn - Connection クラス

commonVal - CommonValues クラス

editVal - EditValues クラス

request - HttpServletRequest クラス

メソッドの詳細

executeUpd

private void **executeUpd**(javax.servlet.http.HttpServletRequest request)
throws checkException

データ更新処理

パラメータ:

request - HttpServletRequest クラス

---

executeDel

private void **executeDel**(javax.servlet.http.HttpServletRequest request)

データ削除処理

パラメータ:

request - HttpServletRequest クラス

---

executeAdd

private void **executeAdd**(javax.servlet.http.HttpServletRequest request)
throws checkException

データ追加処理

パラメータ:

request - HttpServletRequest クラス

---

isMeshData

private boolean **isMeshData**(java.lang.String pramKanriMesh,
java.lang.String pramSpjyn,
int renban)

散布順重複エラーチェック

パラメータ:

KanriMesh -- 管理メッシュ

Spjyn -- 散布順

戻り値:

true:重複 false:重複なし

---

getMasterData

private void **getMasterData**()

マスターデータ取得

---

executeSlt

private void **executeSlt**()

データ取得

---

getPatternMei

public java.lang.String **getPatternMei**()

標準パターン名を返す

戻り値:

標準パターン名

---

getSetsumei

public java.lang.String **getSetsumei**()

標準パターン説明を返す

戻り値:

標準パターン説明

---

getPtnNo

public java.lang.String **getPtnNo**()

パターン番号を返す

戻り値:

パターン番号

---

getRenban

public java.lang.String **getRenban**(int dataIndex)

連番を返す

パラメータ:

dataIndex - インデックス

戻り値:

連番

---

getMeshId

public java.lang.String **getMeshId**(int dataIndex)

メッシュを返す

パラメータ:

dataIndex - インデックス

戻り値:

メッシュ ID

---

getDataId

public java.lang.String **getDataId**(int dataIndex)

データＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

データＩＤ

---

getSanpujun

public java.lang.String **getSanpujun**(int dataIndex)

散布順を返す

パラメータ:

dataIndex - インデックス

戻り値:

散布順

getDataCnt

public int **getDataCnt**()

標準パターン組合せ数を返す

戻り値:

標準パターン組合せ数

---

getMeshIdTbl

public java.lang.String **getMeshIdTbl**(int dataIndex)

全管理メッシュＩＤ返す

パラメータ:

dataIndex - インデックス

戻り値:

全管理メッシュＩＤ

---

getKanriMeshCnt

public int **getKanriMeshCnt**()

全管理メッシュＩＤ数を返す

戻り値:

全管理メッシュＩＤ数

---

getDataIdTbl

public java.lang.String **getDataIdTbl**(int dataIndex)

全散布基本データＩＤ返す

パラメータ:

dataIndex - インデックス

戻り値:

全散布基本データＩＤ

---

getDataIdCnt

public int **getDataIdCnt**()

全散布基本データＩＤ数を返す

戻り値:

全散布基本データＩＤ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

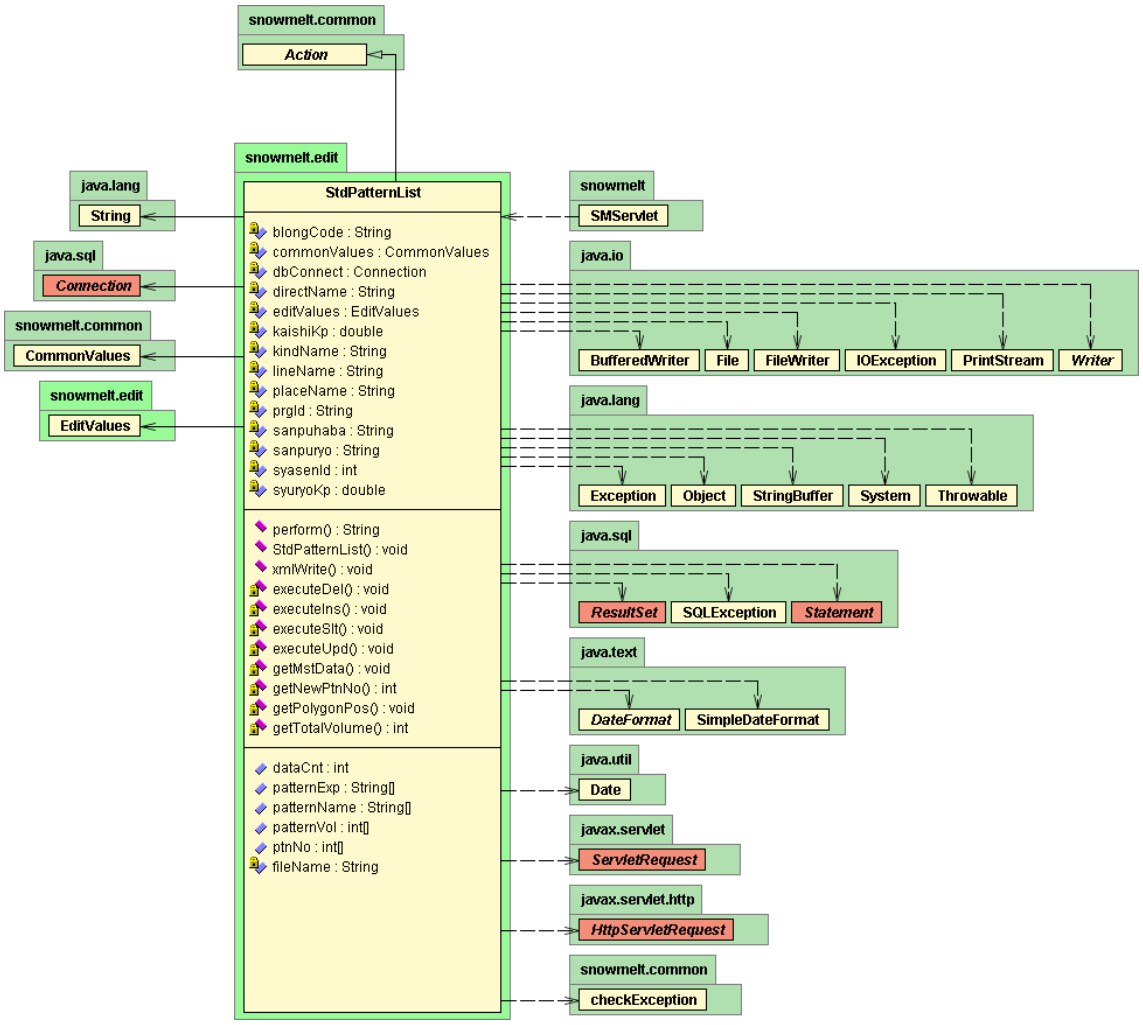
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/D32.jsp"

■ １０　クラス StdPatternList




```
java.lang.Object
  |
  +--snowmelt.common.Action
        |
        +--snowmelt.edit.StdPatternList
```

---

public class **StdPatternList**

extends Action

標準散布パターン一覧クラス

フィールドの概要

private java.lang.String	**blongCode** 所属コード
private CommonValues	**commonValues** 共通変数クラスのインスタンス
private int	**dataCnt** データ数
private java.sql.Connection	**dbConnect** Connection クラス
private java.lang.String	**directName** 散布方向名称
private EditValues	**editValues** 入力値維持クラスのインスタンス
private java.lang.String	**fileName** 作業記録保存ファイル名
private double	**kaishiKp** 開始ＫＰ
private java.lang.String	**kindName** 薬剤名称
private java.lang.String	**lineName** 車線名
private java.lang.String[]	**patternExp** パターン説明配列
private java.lang.String[]	**patternName** パターン名配列
private int[]	**patternVol** 総散布量配列
private java.lang.String	**placeName** 散布箇所名称
private java.lang.String	**prgId** アクション ID
private int[]	**ptnNo**

	パターン番号配列
private java.lang.String	**sanpuhaba** 散布幅
private java.lang.String	**sanpuryo** 散布量
private int	**syasenId** 車線ＩＤ
private double	**syuryoKp** 終了ＫＰ

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

StdPatternList(java.lang.String id, java.sql.Connection dbConn, CommonValues commonVal, EditValues editVal, javax.servlet.http.HttpServletRequest request) コンストラクタ

メソッドの概要

private void	**executeDel**() データ削除
private void	**executeIns**() データ追加
private void	**executeSlt**()

	データ取得
private void	**executeUpd**() データ更新
int	**getDataCnt**() データ数を返す
private java.lang.String	**getFileName**() 受け渡した文字列（ファイル名）に現在の日時をつけて返す
private void	**getMstData**(int dataId) マスターＤＢより名称を取得する
private int	**getNewPtnNo**() 新しいパターン番号取得
java.lang.String	**getPatternExp**(int dataIndex) パターン説明を返す
java.lang.String	**getPatternName**(int dataIndex) パターン名を返す
int	**getPatternVol**(int dataIndex) 総散布量を返す
private void	**getPolygonPos**(int syukusyakuId, java.io.BufferedWriter workWriter) ポリゴンの座標を返す
int	**getPtnNo**(int dataIndex) パターン番号を返す
private int	**getTotalVolume**(java.sql.Statement stmt, java.lang.String blongCode, int ptn_no) 散布パターンの合計散布量（kg）を計算する
java.lang.String	**perform**() 遷移先ｊｓｐ名
void	**xmlWrite**() XML ファイルに書き込み

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues commonValues

共通変数クラスのインスタンス

---

editValues

private EditValues editValues

入力値維持クラスのインスタンス

---

dbConnect

private java.sql.Connection dbConnect

Connection クラス

---

dataCnt

private int dataCnt

データ数

---

ptnNo

private int[] ptnNo

パターン番号配列

---

patternName

private java.lang.String[] **patternName**

パターン名配列

---

patternExp

private java.lang.String[] **patternExp**

パターン説明配列

---

patternVol

private int[] **patternVol**

総散布量配列

---

prgId

private java.lang.String **prgId**

アクション ID

---

blongCode

private java.lang.String **blongCode**

所属コード

---

fileName

private java.lang.String **fileName**

作業記録保存ファイル名

---

syasenId

private int **syasenId**

車線ＩＤ

---

lineName

private java.lang.String **lineName**

車線名

---

placeName

private java.lang.String **placeName**

散布箇所名称

---

sanpuryo

private java.lang.String **sanpuryo**

散布量

---

sanpuhaba

private java.lang.String **sanpuhaba**

散布幅

---

directName

private java.lang.String **directName**

散布方向名称

---

kindName

private java.lang.String **kindName**

薬剤名称

---

kaishiKp

private double **kaishiKp**

開始ＫＰ

syuryoKp

private double **syuryoKp**

終了ＫＰ

コンストラクタの詳細

StdPatternList

public **StdPatternList**(java.lang.String id,
java.sql.Connection dbConn,
CommonValues commonVal,
EditValues editVal,
javax.servlet.http.HttpServletRequest request)
throws checkException

コンストラクタ

パラメータ:

id - アクション ID

dbConn - Connection クラス

commonVal - CommonValues クラス

editVal - EditValues クラス

request - HttpServletRequest クラス

メソッドの詳細

executeUpd

private void **executeUpd**()

データ更新

executeIns

private void **executeIns**()

throws checkException

データ追加

---

executeDel

private void **executeDel**()

データ削除

---

getNewPtnNo

private int **getNewPtnNo**()

新しいパターン番号取得

戻り値:

新しいパターン番号

---

executeSlt

private void **executeSlt**()

データ取得

---

xmlWrite

public void **xmlWrite**()

XML ファイルに書き込み

---

getFileName

private java.lang.String **getFileName**()

受け渡した文字列（ファイル名）に現在の日時をつけて返す

戻り値:

"ptn" + 現在の日時 + ".xml"

---

getPolygonPos

private void **getPolygonPos**(int syukusyakuId,
java.io.BufferedWriter workWriter)

ポリゴンの座標を返す

パラメータ:

syukusyakuId - 縮尺 ID

workWriter - BufferedWriter クラス

---

getMstData

private void **getMstData**(int dataId)

マスターＤＢより名称を取得する

パラメータ:

dataId - データ ID

---

getPtnNo

public int **getPtnNo**(int dataIndex)

パターン番号を返す

パラメータ:

dataIndex - インデックス

戻り値:

パターン番号

---

getPatternName

public java.lang.String **getPatternName**(int dataIndex)

パターン名を返す

パラメータ:

dataIndex - インデックス

戻り値:

パターン名

---

getPatternExp

public java.lang.String **getPatternExp**(int dataIndex)

パターン説明を返す

パラメータ:

dataIndex - インデックス

戻り値:

パターン説明

---

getPatternVol

public int **getPatternVol**(int dataIndex)

総散布量を返す

パラメータ:

dataIndex - インデックス

戻り値:

総散布量

---

getDataCnt

public int **getDataCnt**()

データ数を返す

戻り値:

データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/D31.jsp" || "/jsp/D41.jsp" || "/jsp/Menu.jsp"

---

getTotalVolume

```
private int getTotalVolume(java.sql.Statement stmt,
                           java.lang.String blongCode,
                           int ptn_no)
```

散布パターンの合計散布量（kg）を計算する

パラメータ:

stmt - Statement クラス

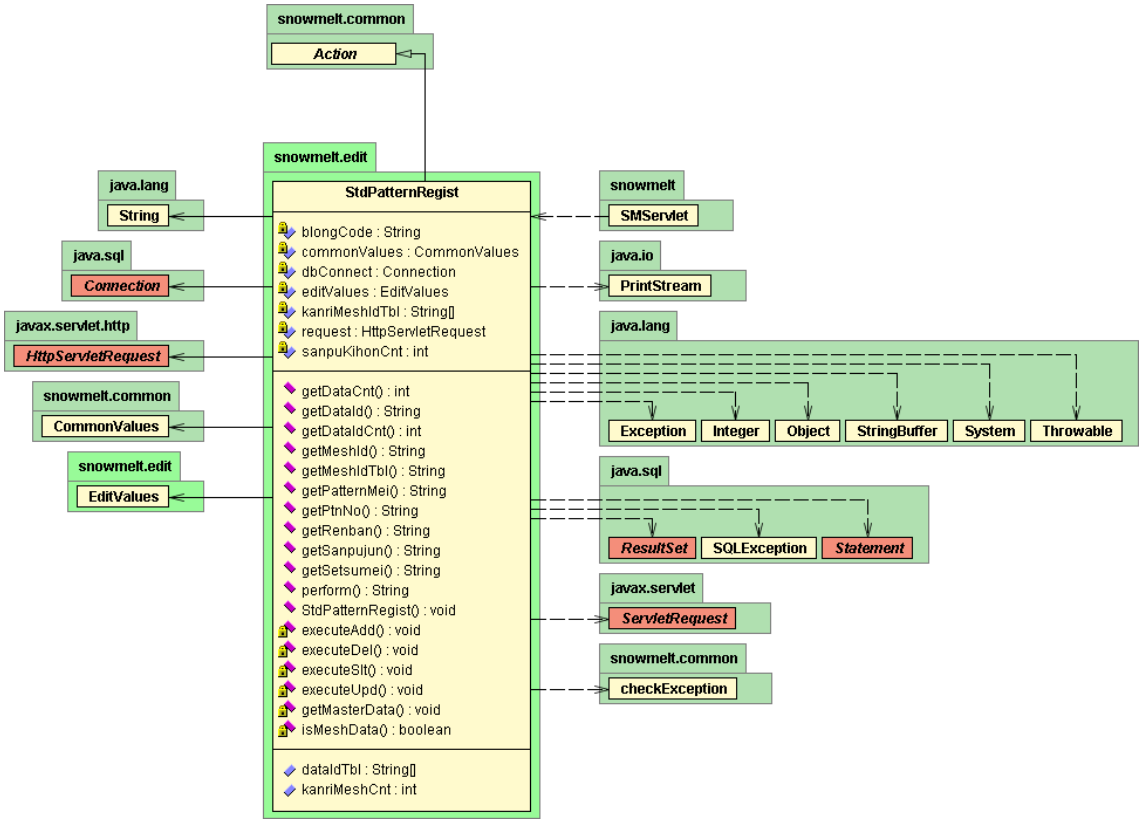
blongCode - 所属コード

ptn_no - パターン番号

戻り値:

合計散布量

■ １１　クラス StdPatternRegist



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.edit.StdPatternRegist**

---

public class **StdPatternRegist**

extends Action

標準散布パターン新規登録クラス

フィールドの概要

private java.lang.String	**blongCode** 所属コード
private CommonValues	**commonValues**

	共通変数クラスのインスタンス
private java.lang.String[]	**dataIdTbl** データ ID 配列
private java.sql.Connection	**dbConnect** Connection クラス
private EditValues	**editValues** 入力値維持クラスのインスタンス
private int	**kanriMeshCnt** 管理メッシュ取得用変数
private java.lang.String[]	**kanriMeshIdTbl** 管理メッシュ配列
private javax.servlet.http.HttpServletRequest	**request** HttpServletRequest クラス
private int	**sanpuKihonCnt** 散布基本データＩＤ取得用変数

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

StdPatternRegist(java.lang.String id, java.sql.Connection dbConn, CommonValues commonVal, EditValues editVal, javax.servlet.http.HttpServletRequest req)
コンストラクタ

メソッドの概要

private void	**executeAdd**() データ追加処理
private void	**executeDel**() データ削除処理
private void	**executeSlt**() データ取得
private void	**executeUpd**() データ更新処理
int	**getDataCnt**() 標準パターン組合せ数を返す
java.lang.String	**getDataId**(int dataIndex) データＩＤを返す
int	**getDataIdCnt**() 全散布基本データＩＤ数を返す
java.lang.String	**getDataIdTbl**(int dataIndex) 散布基本データＩＤ返す
int	**getKanriMeshCnt**() 全管理メッシュＩＤ数を返す
private void	**getMasterData**() マスターデータ取得
java.lang.String	**getMeshId**(int dataIndex) メッシュを返す
java.lang.String	**getMeshIdTbl**(int dataIndex) 管理メッシュＩＤを返す
java.lang.String	**getPatternMei**() 標準パターン名を返す
java.lang.String	**getPtnNo**() パターン番号を返す
java.lang.String	**getRenban**(int dataIndex)

	連番を返す
java.lang.String	**getSanpujun**(int dataIndex) 散布順を返す
java.lang.String	**getSetsumei**() 標準パターン説明を返す
private boolean	**isMeshData**(java.lang.String pramKanriMesh, java.lang.String pramSpjyn, int renban) 散布順重複エラーチェック
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

editValues

private EditValues **editValues**

入力値維持クラスのインスタンス

---

dbConnect

private java.sql.Connection **dbConnect**

Connection クラス

---

request

private javax.servlet.http.HttpServletRequest **request**

HttpServletRequest クラス

---

blongCode

private java.lang.String **blongCode**

所属コード

---

kanriMeshCnt

private int **kanriMeshCnt**

管理メッシュ取得用変数

---

kanriMeshIdTbl

private java.lang.String[] **kanriMeshIdTbl**

管理メッシュ配列

---

sanpuKihonCnt

private int **sanpuKihonCnt**

散布基本データＩＤ取得用変数

---

dataIdTbl

private java.lang.String[] **dataIdTbl**

データ ID 配列

コンストラクタの詳細

StdPatternRegist

public **StdPatternRegist**(java.lang.String id,
java.sql.Connection dbConn,
CommonValues commonVal,
EditValues editVal,
javax.servlet.http.HttpServletRequest req)
throws checkException

コンストラクタ

パラメータ:

id - アクション ID

dbConn - Connection クラス

commonVal - CommonValues クラス

editVal - EditValues クラス

req - HttpServletRequest クラス

メソッドの詳細

executeUpd

private void **executeUpd**()
throws checkException

データ更新処理

---

executeDel

private void **executeDel**()

データ削除処理

---

executeAdd

private void **executeAdd**()
throws checkException

データ追加処理

---

isMeshData

private boolean **isMeshData**(java.lang.String pramKanriMesh,
java.lang.String pramSpjyn,
int renban)

散布順重複エラーチェック

パラメータ:

KanriMesh - - 管理メッシュ

Spjyn - - 散布順

戻り値:

true:重複 false:重複なし

---

getMasterData

private void **getMasterData**()

マスターデータ取得

---

executeSlt

private void **executeSlt**()

データ取得

---

getPatternMei

public java.lang.String **getPatternMei**()

標準パターン名を返す

戻り値:

標準パターン名

---

getSetsumei

public java.lang.String **getSetsumei**()

標準パターン説明を返す

戻り値:

標準パターン説明

---

getPtnNo

public java.lang.String **getPtnNo**()

パターン番号を返す

戻り値:

パターン番号

---

getRenban

public java.lang.String **getRenban**(int dataIndex)

連番を返す

パラメータ:

dataIndex - インデックス

戻り値:

連番

---

getMeshId

public java.lang.String **getMeshId**(int dataIndex)

メッシュを返す

パラメータ:

dataIndex - インデックス

戻り値:

メッシュ

---

getDataId

public java.lang.String **getDataId**(int dataIndex)

データＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

データＩＤ

---

getSanpujun

public java.lang.String **getSanpujun**(int dataIndex)

散布順を返す

パラメータ:

dataIndex - インデックス

戻り値:

散布順

---

getDataCnt

public int **getDataCnt**()

標準パターン組合せ数を返す

戻り値:

標準パターン組合せ数

---

getMeshIdTbl

public java.lang.String **getMeshIdTbl**(int dataIndex)

管理メッシュＩＤを返す

パラメータ:

dataIndex - インデックス

戻り値:

管理メッシュＩＤ

---

getKanriMeshCnt

public int **getKanriMeshCnt**()

全管理メッシュＩＤ数を返す

戻り値:

全管理メッシュＩＤ数

---

getDataIdTbl

public java.lang.String **getDataIdTbl**(int dataIndex)

散布基本データＩＤ返す

パラメータ:

dataIndex - インデックス

戻り値:

散布基本データＩＤ

---

getDataIdCnt

public int **getDataIdCnt**()

全散布基本データＩＤ数を返す

戻り値:

全散布基本データＩＤ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

オーバーライド:

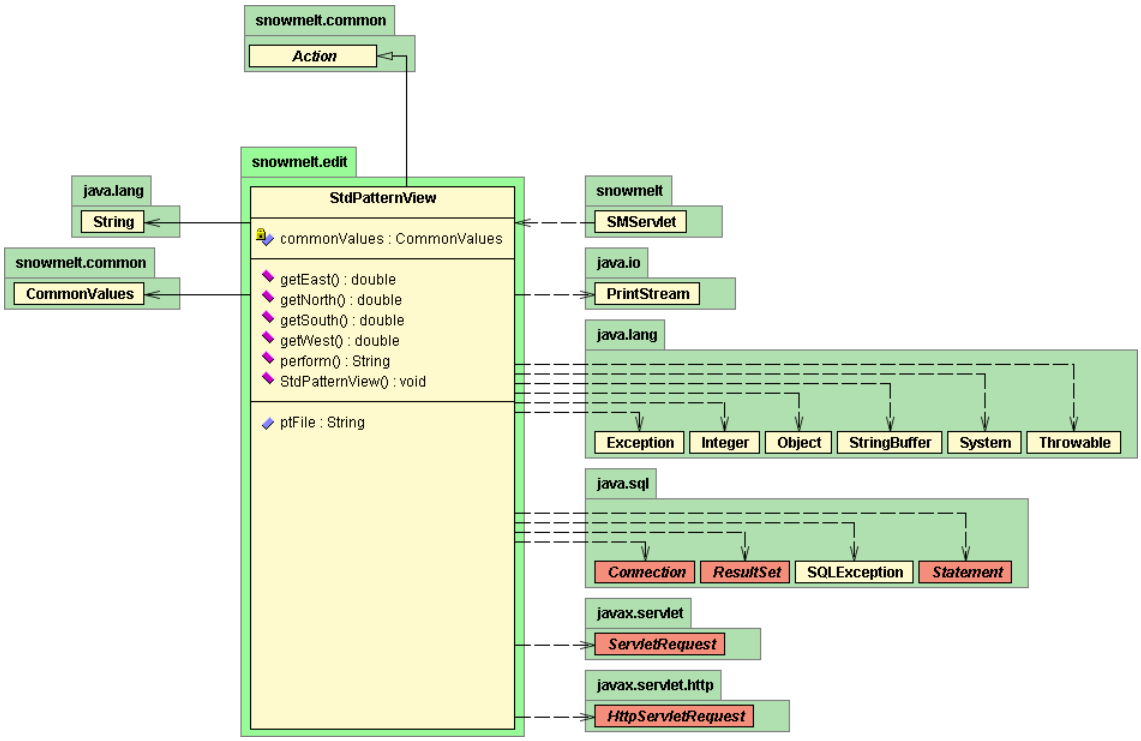
クラス Action 内の perform

クラス snowmelt.common.Action からコピーされたタグ:

戻り値:

遷移先ｊｓｐ名

■ １２　クラス StdPatternView



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.edit.StdPatternView**

---

public class **StdPatternView**

extends Action

標準散布パターン表示クラス

フィールドの概要

private CommonValues	**commonValues** 共通変数クラスのインスタンス
private java.lang.String	**ptFile** 標準パターンファイル名

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

StdPatternView(java.sql.Connection dbConn, CommonValues commonVal, javax.servlet.http.HttpServletRequest request) コンストラクタ

メソッドの概要

double	**getEast**() 西経を返す
double	**getNorth**() 北緯を返す
java.lang.String	**getPtFile**() パターンファイル名
double	**getSouth**() 南緯を返す
double	**getWest**() 東経を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

ptFile

private java.lang.String **ptFile**

標準パターンファイル名

コンストラクタの詳細

StdPatternView

public **StdPatternView**(java.sql.Connection dbConn,
CommonValues commonVal,
javax.servlet.http.HttpServletRequest request)

コンストラクタ

パラメータ:

dbConn - Connection

commonVal - CommonValues

request – HttpServletRequest

メソッドの詳細

getWest

public double **getWest**()

東経を返す

戻り値:

東経

---

getEast

public double **getEast**()

西経を返す

戻り値:

西経

---

getNorth

public double **getNorth**()

北緯を返す

戻り値:

北緯

---

getSouth

public double **getSouth**()

南緯を返す

戻り値:

南緯

---

getPtFile

public java.lang.String **getPtFile**()

パターンファイル名

戻り値:

パターンファイル名

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

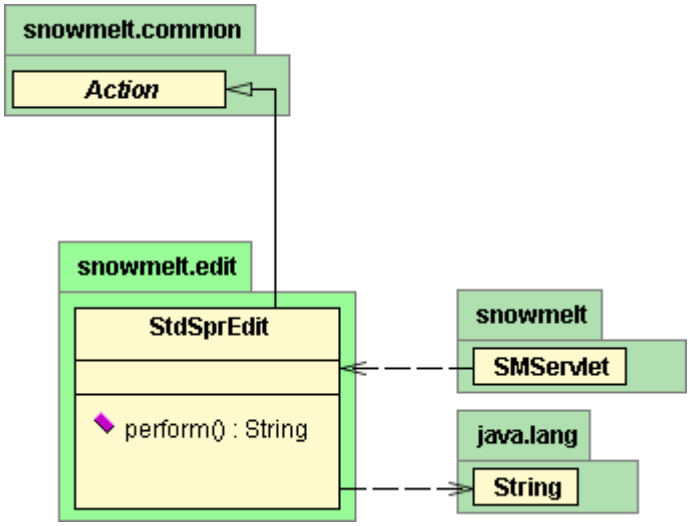
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/D42.jsp"

■ １３　クラス StdSprEdit



java.lang.Object
  |
  +--snowmelt.common.Action
        |
        +--**snowmelt.edit.StdSprEdit**

---

public class **StdSprEdit**
extends Action

標準散布パターン作成メニュークラス

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

StdSprEdit()

メソッドの概要

java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

コンストラクタの詳細

StdSprEdit

public **StdSprEdit**()

メソッドの詳細

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

オーバーライド:

クラス Action 内の perform

戻り値:

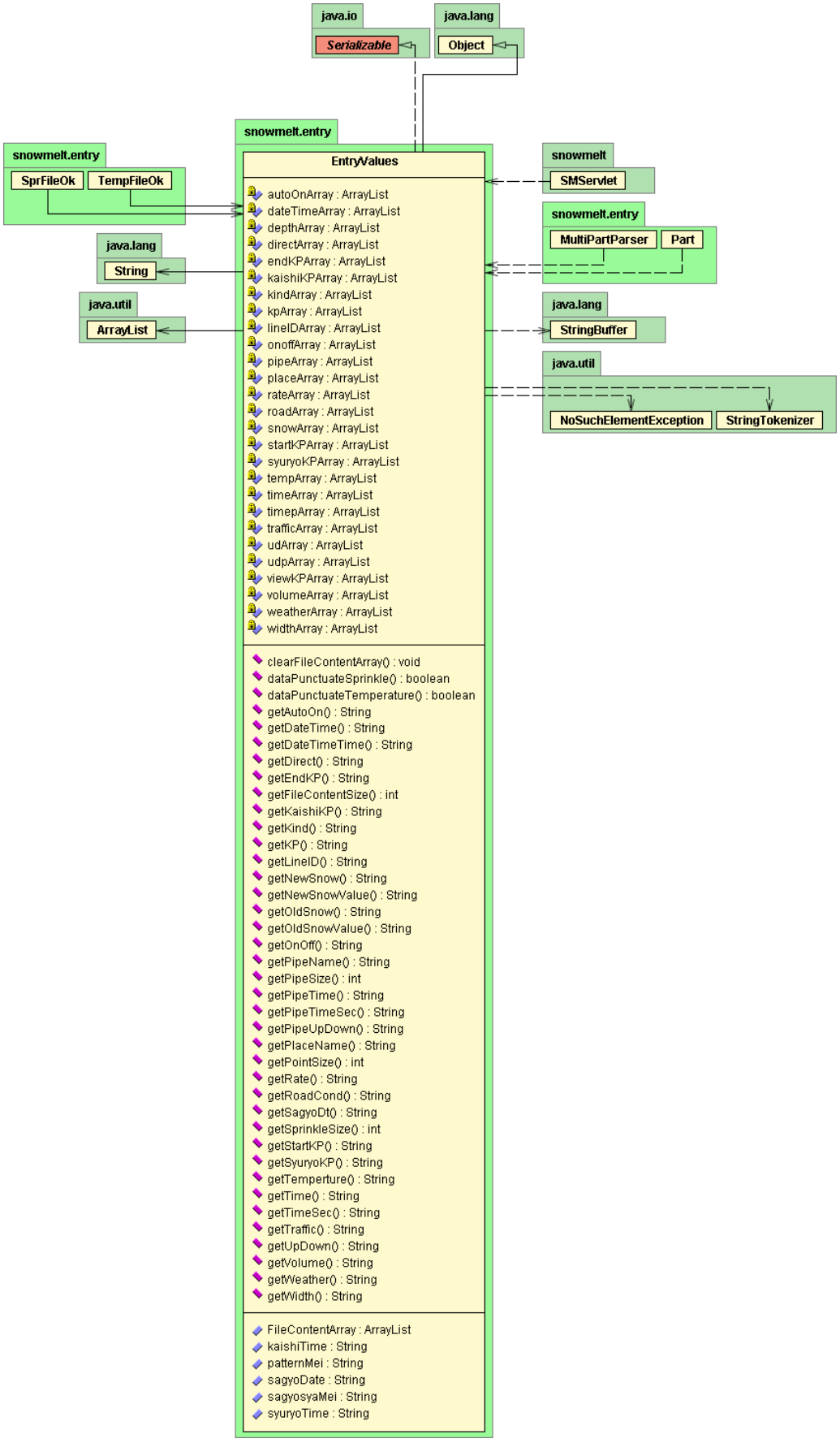
"/jsp/D01.jsp"

---

e） パッケージ snowmelt.entry

クラスの概要	
EntryValues	登録ファイル種別選択　共通変数格納クラス
EtyFileSelect	登録ファイル種別選択
MultiPartParser	送信データ解析クラス
Part	送信の個々のパートを表すクラス
SprFileList	散布作業記録ファイル選択
SprFileOk	散布作業記録ファイル登録完了クラス
TempFileList	巡回記録ファイル選択
TempFileOk	巡回記録ファイル登録完了

■ 1　クラス EntryValues



java.lang.Object

  |

  +--**snowmelt.entry.EntryValues**

すべての実装インタフェース:

java.io.Serializable

---

public class **EntryValues**

extends java.lang.Object

implements java.io.Serializable

登録ファイル種別選択　共通変数格納クラス

フィールドの概要

private　java.util.ArrayList	**autoOnArray** 散布結果データ部 自動・手動
private　java.util.ArrayList	**dateTimeArray** 散布結果データ部 日時
private　java.util.ArrayList	**depthArray** 巡回記録データ部（地点） 積雪深
private　java.util.ArrayList	**directArray** 散布結果データ部 散布方向
private　java.util.ArrayList	**endKPArray** 散布結果データ部 終了 KP
private　java.util.ArrayList	**FileContentArray**
private　java.util.ArrayList	**kaishiKPArray** 巡回記録データ部（地点） 開始 KP
private　java.lang.String	**kaishiTime** ヘッダー部 開始時間
private　java.util.ArrayList	**kindArray** 散布結果データ部 散布薬剤

private java.util.ArrayList	**kpArray** 巡回記録データ部（地点） KP
private java.util.ArrayList	**lineIDArray** 散布結果データ部 車線 ID
private java.util.ArrayList	**onoffArray** 巡回記録データ部（消雪） 消雪パイプ稼動状況
private java.lang.String	**patternMei** ヘッダー部 パターン名
private java.util.ArrayList	**pipeArray** 巡回記録データ部（消雪） 消雪パイプ箇所名
private java.util.ArrayList	**placeArray** 巡回記録データ部（地点） 地点名
private java.util.ArrayList	**rateArray** 散布結果データ部 散布割合
private java.util.ArrayList	**roadArray** 巡回記録データ部（地点） 路面状況
private java.lang.String	**sagyoDate** ヘッダー部 作業日
private java.lang.String	**sagyosyaMei** ヘッダー部 作業者名
private java.util.ArrayList	**snowArray** 巡回記録データ部（地点） 雪種別
private java.util.ArrayList	**startKPArray** 散布結果データ部 開始 KP
private java.util.ArrayList	**syuryoKPArray** 巡回記録データ部（地点） 終了 KP
private java.lang.String	**syuryoTime** ヘッダー部 終了時間
private java.util.ArrayList	**tempArray** 巡回記録データ部（地点） 気温
private java.util.ArrayList	**timeArray**

	巡回記録データ部（地点） 観測時間
private java.util.ArrayList	**timepArray** 巡回記録データ部（消雪） 観測時間
private java.util.ArrayList	**trafficArray** 巡回記録データ部（地点） 交通状況
private java.util.ArrayList	**udArray** 巡回記録データ部（地点） 上り下り種別
private java.util.ArrayList	**udpArray** 巡回記録データ部（消雪） 上り下り種別
private java.util.ArrayList	**viewKPArray** 巡回記録データ部（地点） 表示 KP
private java.util.ArrayList	**volumeArray** 散布結果データ部 散布密度
private java.util.ArrayList	**weatherArray** 巡回記録データ部（地点） 天候
private java.util.ArrayList	**widthArray** 散布結果データ部 散布幅

コンストラクタの概要

EntryValues()	

メソッドの概要

void	**clearFileContentArray**() ファイルコンテントクリア
boolean	**dataPunctuateSprinkle**()

	","で区切られたデータを振分ける（散布結果）
boolean	**dataPunctuateTemperature**() ","で区切られたデータを振分ける（巡回記録）
java.lang.String	**getAutoOn**(int tblIdx) 自動・手動ゲッター
java.lang.String	**getDateTime**(int tblIdx) 散布日時ゲッター
java.lang.String	**getDateTimeTime**(int tblIdx) 散布時刻ゲッター
java.lang.String	**getDirect**(int tblIdx) 散布方向ゲッター
java.lang.String	**getEndKP**(int tblIdx) 終了 KP ゲッター
java.lang.String	**getFileContentArray**(int tblIdx) ファイルコンテントゲッター
int	**getFileContentSize**() ファイルコンテントサイズゲッター
java.lang.String	**getKaishiKP**(int tblIdx) 開始 KP ゲッター
java.lang.String	**getKaishiTime**() 開始時間ゲッター
java.lang.String	**getKind**(int tblIdx) 散布薬剤ゲッター
java.lang.String	**getKP**(int tblIdx) KP ゲッター
java.lang.String	**getLineID**(int tblIdx) 車線 ID ゲッター
java.lang.String	**getNewSnow**(int tblIdx) 新雪深ゲッター
java.lang.String	**getNewSnowValue**(int tblIdx) 新雪深（値のみ）ゲッター

java.lang.String	**getOldSnow**(int tblIdx) 圧雪深ゲッター
java.lang.String	**getOldSnowValue**(int tblIdx) 圧雪深（値のみ）ゲッター
java.lang.String	**getOnOff**(int tblIdx) 消雪パイプ稼動状況ゲッター
java.lang.String	**getPatternMei**() パターン名ゲッター
java.lang.String	**getPipeName**(int tblIdx) 消雪パイプ箇所名ゲッター
int	**getPipeSize**() 消雪パイプ数の取得
java.lang.String	**getPipeTime**(int tblIdx) 観測時間ゲッター
java.lang.String	**getPipeTimeSec**(int tblIdx) 観測時間ゲッター
java.lang.String	**getPipeUpDown**(int tblIdx) 消雪パイプ上り・下り種別
java.lang.String	**getPlaceName**(int tblIdx) 地点箇所名ゲッター
int	**getPointSize**() 地点数の取得
java.lang.String	**getRate**(int tblIdx) 散布割合ゲッター
java.lang.String	**getRoadCond**(int tblIdx) 路面状況ゲッター
java.lang.String	**getSagyoDate**() 作業日ゲッター
java.lang.String	**getSagyoDt**() 作業日＋時間ゲッター
java.lang.String	**getSagyosyaMei**()

	作業者名ゲッター
int	**getSprinkleSize**() 散布箇所数ゲッター
java.lang.String	**getStartKP**(int tblIdx) 開始 KP ゲッター
java.lang.String	**getSyuryoKP**(int tblIdx) 終了 KP ゲッター
java.lang.String	**getSyuryoTime**() 終了時間ゲッター
java.lang.String	**getTemperture**(int tblIdx) 気温ゲッター
java.lang.String	**getTime**(int tblIdx) 観測時間ゲッター
java.lang.String	**getTimeSec**(int tblIdx) 観測時間ゲッター
java.lang.String	**getTraffic**(int tblIdx) 交通状況ゲッター
java.lang.String	**getUpDown**(int tblIdx) 上り・下り種別
java.lang.String	**getVolume**(int tblIdx) 散布密度ゲッター
java.lang.String	**getWeather**(int tblIdx) 天候ゲッター
java.lang.String	**getWidth**(int tblIdx) 散布幅ゲッター
void	**setFileContentArray**(java.lang.String PrmData) ファイルコンテントセッター

フィールドの詳細

FileContentArray

private java.util.ArrayList FileContentArray

---

sagyoDate

private java.lang.String sagyoDate

ヘッダー部 作業日

---

kaishiTime

private java.lang.String kaishiTime

ヘッダー部 開始時間

---

syuryoTime

private java.lang.String syuryoTime

ヘッダー部 終了時間

---

sagyosyaMei

private java.lang.String sagyosyaMei

ヘッダー部 作業者名

---

patternMei

private java.lang.String patternMei

ヘッダー部 パターン名

---

dateTimeArray

private java.util.ArrayList dateTimeArray

散布結果データ部 日時

---

autoOnArray

private java.util.ArrayList autoOnArray

散布結果データ部 自動・手動

---

lineIDArray

private java.util.ArrayList lineIDArray

散布結果データ部 車線 ID

---

startKPArray

private java.util.ArrayList startKPArray

散布結果データ部 開始 KP

---

endKPArray

private java.util.ArrayList endKPArray

散布結果データ部 終了 KP

---

volumeArray

private java.util.ArrayList volumeArray

散布結果データ部 散布密度

---

widthArray

private java.util.ArrayList widthArray

散布結果データ部 散布幅

---

directArray

private java.util.ArrayList directArray

散布結果データ部 散布方向

rateArray

private java.util.ArrayList rateArray

散布結果データ部 散布割合

---

kindArray

private java.util.ArrayList kindArray

散布結果データ部 散布薬剤

---

kpArray

private java.util.ArrayList kpArray

巡回記録データ部（地点） KP

---

placeArray

private java.util.ArrayList placeArray

巡回記録データ部（地点） 地点名

---

udArray

private java.util.ArrayList udArray

巡回記録データ部（地点） 上り下り種別

---

timeArray

private java.util.ArrayList timeArray

巡回記録データ部（地点） 観測時間

---

viewKPArray

private java.util.ArrayList **viewKPArray**

巡回記録データ部（地点） 表示 KP

---

kaishiKPArray

private java.util.ArrayList **kaishiKPArray**

巡回記録データ部（地点） 開始 KP

---

syuryoKPArray

private java.util.ArrayList **syuryoKPArray**

巡回記録データ部（地点） 終了 KP

---

tempArray

private java.util.ArrayList **tempArray**

巡回記録データ部（地点） 気温

---

weatherArray

private java.util.ArrayList **weatherArray**

巡回記録データ部（地点） 天候

---

roadArray

private java.util.ArrayList **roadArray**

巡回記録データ部（地点） 路面状況

---

trafficArray

private java.util.ArrayList **trafficArray**

巡回記録データ部（地点） 交通状況

---

snowArray

private java.util.ArrayList snowArray

巡回記録データ部（地点）　雪種別

---

depthArray

private java.util.ArrayList depthArray

巡回記録データ部（地点）　積雪深

---

pipeArray

private java.util.ArrayList pipeArray

巡回記録データ部（消雪）　消雪パイプ箇所名

---

udpArray

private java.util.ArrayList udpArray

巡回記録データ部（消雪）　上り下り種別

---

timepArray

private java.util.ArrayList timepArray

巡回記録データ部（消雪）　観測時間

---

onoffArray

private java.util.ArrayList onoffArray

巡回記録データ部（消雪）　消雪パイプ稼動状況

コンストラクタの詳細

EntryValues

public **EntryValues**()

メソッドの詳細

clearFileContentArray

public void **clearFileContentArray**()

ファイルコンテントクリア

---

getFileContentSize

public int **getFileContentSize**()

ファイルコンテントサイズゲッター

---

dataPunctuateSprinkle

public boolean **dataPunctuateSprinkle**()

","で区切られたデータを振分ける（散布結果）

戻り値:

true:OK false:転送ファイルが違う

---

dataPunctuateTemperature

public boolean **dataPunctuateTemperature**()

","で区切られたデータを振分ける（巡回記録）

戻り値:

true:OK false:転送ファイルが違う

---

getFileContentArray

public java.lang.String **getFileContentArray**(int tblIdx)

ファイルコンテントゲッター

パラメータ:

tblIdx - インデックス

戻り値:

ファイルコンテント

---

setFileContentArray

public void **setFileContentArray**(java.lang.String PrmData)

ファイルコンテントセッター

パラメータ:

PrmData - ファイルコンテント

---

getSagyoDt

public java.lang.String **getSagyoDt**()

作業日＋時間ゲッター

戻り値:

作業日＋時間

---

getSagyoDate

public java.lang.String **getSagyoDate**()

作業日ゲッター

戻り値:

作業日

---

getKaishiTime

public java.lang.String **getKaishiTime**()

開始時間ゲッター

戻り値:

開始時間

---

getSyuryoTime

public java.lang.String **getSyuryoTime**()

終了時間ゲッター

戻り値:

終了時間

---

getSagyosyaMei

public java.lang.String **getSagyosyaMei**()

作業者名ゲッター

戻り値:

作業者名

---

getPatternMei

public java.lang.String **getPatternMei**()

パターン名ゲッター

戻り値:

パターン名

---

getDateTime

public java.lang.String **getDateTime**(int tblIdx)

散布日時ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

散布日時

---

getDateTimeTime

public java.lang.String **getDateTimeTime**(int tblIdx)

散布時刻ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

散布時刻

---

getAutoOn

public java.lang.String **getAutoOn**(int tblIdx)

自動・手動ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

自動・手動

---

getLineID

public java.lang.String **getLineID**(int tblIdx)

車線 ID ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

車線 ID

---

getStartKP

public java.lang.String **getStartKP**(int tblIdx)

開始 KP ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

開始 KP

---

getEndKP

public java.lang.String **getEndKP**(int tblIdx)

終了 KP ゲッター

パラメータ:

tbllidx - インデックス

戻り値:

終了 KP

---

getVolume

public java.lang.String **getVolume**(int tbllidx)

散布密度ゲッター

パラメータ:

tbllidx - インデックス

戻り値:

散布密度

---

getWidth

public java.lang.String **getWidth**(int tbllidx)

散布幅ゲッター

パラメータ:

tbllidx - インデックス

戻り値:

散布幅

---

getDirect

public java.lang.String **getDirect**(int tbllidx)

散布方向ゲッター

パラメータ:

tbllidx - インデックス

戻り値:

散布方向

---

getRate

public java.lang.String **getRate**(int tblIdx)

散布割合ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

散布割合

---

getKind

public java.lang.String **getKind**(int tblIdx)

散布薬剤ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

散布薬剤

---

getSprinkleSize

public int **getSprinkleSize**()

散布箇所数ゲッター

戻り値:

散布箇所数

---

getKP

public java.lang.String **getKP**(int tblIdx)

KP ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

KP

---

getPlaceName

public java.lang.String **getPlaceName**(int tblIdx)

地点箇所名ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

地点箇所名

---

getUpDown

public java.lang.String **getUpDown**(int tblIdx)

上り・下り種別

パラメータ:

tblIdx - インデックス

戻り値:

上り・下り

---

getTime

public java.lang.String **getTime**(int tblIdx)

観測時間ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

観測時間

---

getTimeSec

public java.lang.String **getTimeSec**(int tblIdx)

観測時間ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

観測時間

---

getTemperture

public java.lang.String **getTemperture**(int tblIdx)

気温ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

気温

---

getWeather

public java.lang.String **getWeather**(int tblIdx)

天候ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

天候

---

getRoadCond

public java.lang.String **getRoadCond**(int tblIdx)

路面状況ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

路面状況

---

getTraffic

public java.lang.String **getTraffic**(int tblIdx)

交通状況ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

交通状況

getNewSnow

public java.lang.String **getNewSnow**(int tblIdx)

新雪深ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

新雪深

---

getOldSnow

public java.lang.String **getOldSnow**(int tblIdx)

圧雪深ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

圧雪深

---

getNewSnowValue

public java.lang.String **getNewSnowValue**(int tblIdx)

新雪深（値のみ）ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

新雪深（値のみ）

---

getOldSnowValue

public java.lang.String **getOldSnowValue**(int tblIdx)

圧雪深（値のみ）ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

圧雪深（値のみ）

---

getKaishiKP

public java.lang.String **getKaishiKP**(int tblIdx)

開始 KP ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

開始 KP

---

getSyuryoKP

public java.lang.String **getSyuryoKP**(int tblIdx)

終了 KP ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

終了 KP

---

getPipeName

public java.lang.String **getPipeName**(int tblIdx)

消雪パイプ箇所名ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

消雪パイプ箇所名

---

getPipeUpDown

public java.lang.String **getPipeUpDown**(int tblIdx)

消雪パイプ上り・下り種別

パラメータ:

tblIdx - インデックス

戻り値:

消雪パイプ上り・下り

---

getPipeTime

public java.lang.String **getPipeTime**(int tblIdx)

観測時間ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

観測時間

---

getPipeTimeSec

public java.lang.String **getPipeTimeSec**(int tblIdx)

観測時間ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

観測時間

---

getOnOff

public java.lang.String **getOnOff**(int tblIdx)

消雪パイプ稼動状況ゲッター

パラメータ:

tblIdx - インデックス

戻り値:

消雪パイプ稼動状況

---

getPointSize

public int **getPointSize**()

地点数の取得

戻り値:

地点数

---

getPipeSize

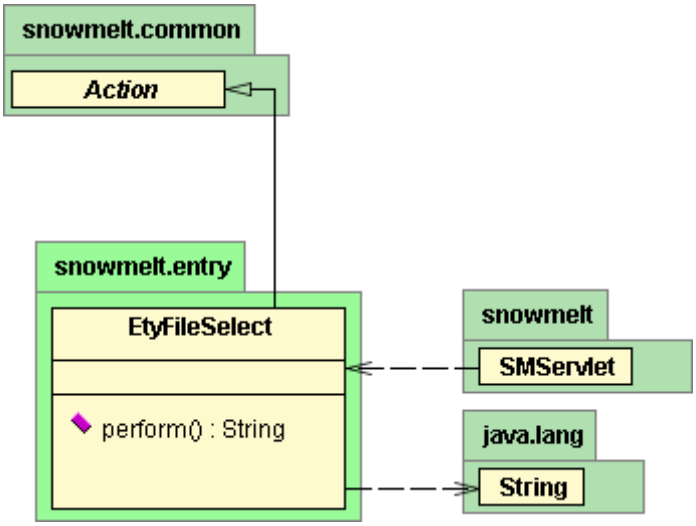
public int **getPipeSize**()

消雪パイプ数の取得

戻り値:

消雪パイプ数

■ 2　クラス EtyFileSelect




```
java.lang.Object
  |
  +--snowmelt.common.Action
        |
        +--snowmelt.entry.EtyFileSelect
```

---

public class **EtyFileSelect**

extends Action

登録ファイル種別選択

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

EtyFileSelect()

メソッドの概要

java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

コンストラクタの詳細

EtyFileSelect

public **EtyFileSelect**()

メソッドの詳細

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

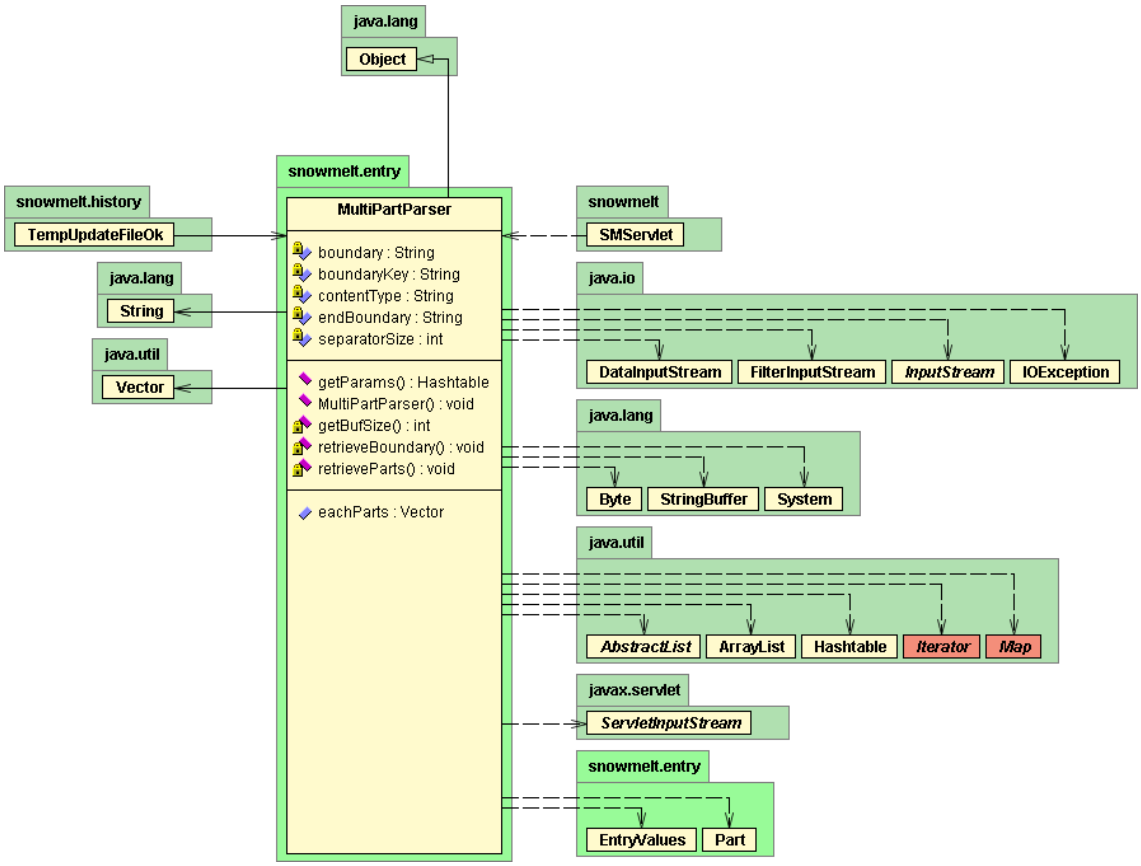
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/C01.jsp"

■ 3　クラス MultiPartParser



java.lang.Object

  |

  +--**snowmelt.entry.MultiPartParser**

---

public class **MultiPartParser**

extends java.lang.Object

送信データ解析クラス

フィールドの概要	
private　java.lang.String	**boundary** 開始バウンダリ
private　java.lang.String	**boundaryKey** バウンダリ文字列
private　java.lang.String	**contentType**

	コンテントタイプ
private java.util.Vector	**eachParts** 各パート
private java.lang.String	**endBoundary** 終了バウンダリ
private int	**separatorSize** セパレータサイズ

コンストラクタの概要

MultiPartParser(EntryValues entryValues, java.lang.String contentType, int contentLength, javax.servlet.ServletInputStream istream) コンストラクタ

メソッドの概要

private int	**getBufSize**(int size) バッファサイズの取得
java.util.Vector	**getEachParts**() 個々のパート取り出し
java.util.Hashtable	**getParams**() 取得パラメータを返す
private void	**retrieveBoundary**(java.lang.String contentType) バウンダリ文字列の検索
private void	**retrieveParts**(EntryValues entryValues, int contentLength, javax.servlet.ServletInputStream istream) 受け取りデータを個々の Part にパース（分解）する

フィールドの詳細

boundaryKey

private final java.lang.String boundaryKey

バウンダリ文字列

---

separatorSize

private final int separatorSize

セパレータサイズ

---

boundary

private java.lang.String boundary

開始バウンダリ

---

endBoundary

private java.lang.String endBoundary

終了バウンダリ

---

eachParts

private java.util.Vector eachParts

各パート

---

contentType

private java.lang.String contentType

コンテントタイプ

コンストラクタの詳細

MultiPartParser

public **MultiPartParser**(EntryValues entryValues,
java.lang.String contentType,
int contentLength,
javax.servlet.ServletInputStream istream)
throws java.io.IOException

コンストラクタ

パラメータ:

entryValues - EntryValues クラス

contentType - コンテンツタイプ

contentLength - コンテンツの文字列長

istream - 読み込みもとストリーム

メソッドの詳細

getEachParts

public java.util.Vector **getEachParts**()

個々のパート取り出し

戻り値:

分割した個々のパートの配列を返す

---

retrieveBoundary

private void **retrieveBoundary**(java.lang.String contentType)

バウンダリ文字列の検索

パラメータ:

contentType - コンテンツタイプ

---

retrieveParts

private void **retrieveParts**(EntryValues entryValues,
int contentLength,
javax.servlet.ServletInputStream istream)
throws java.io.IOException

受け取りデータを個々の Part にパース（分解）する

パラメータ:

contentLength - コンテンツ長

istream - データ読み込みもと Stream

---

getBufSize

private int **getBufSize**(int size)

バッファサイズの取得

パラメータ:

size - データサイズ

戻り値:

必要バッファサイズ

---

getParams

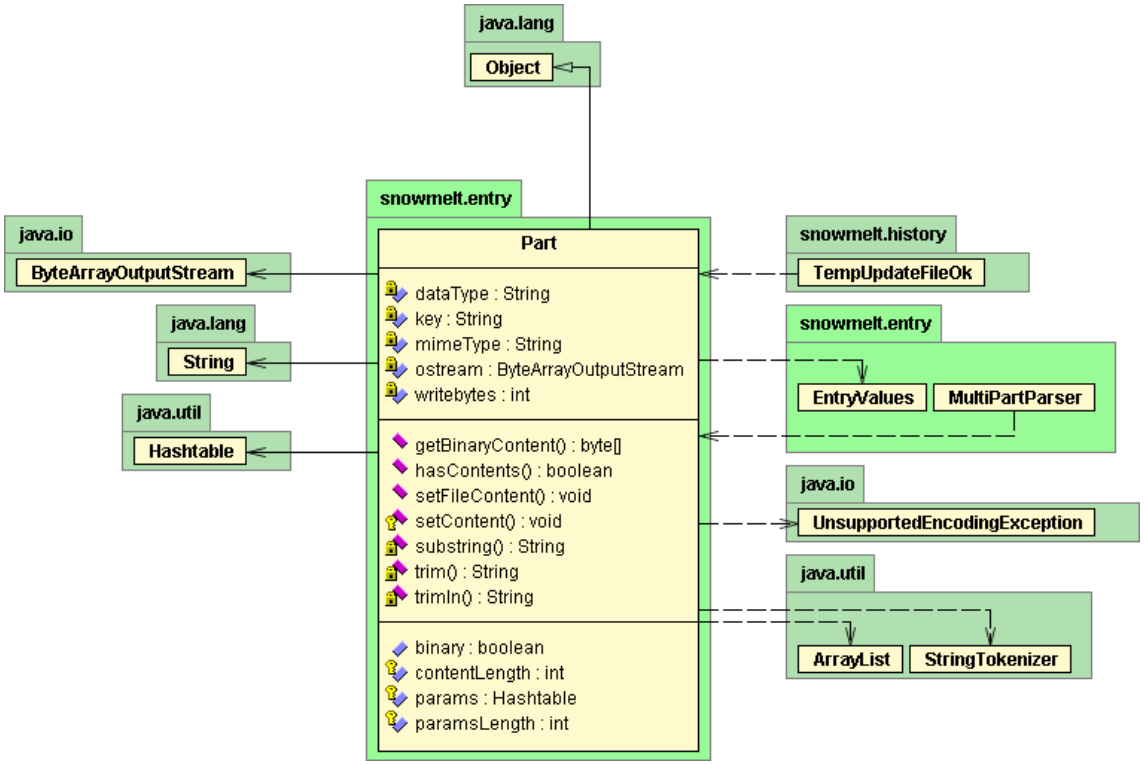
public java.util.Hashtable **getParams**()

取得パラメータを返す

戻り値:

Hashtable 取得パラメータ

■ 4　クラス Part



java.lang.Object

　|

　+--**snowmelt.entry.Part**

---

public class **Part**

extends java.lang.Object

送信の個々のパートを表すクラス

フィールドの概要

private boolean	**binary** バイナリ判定フラグ
private int	**contentLength** コンテントレングス
private java.lang.String	**dataType** コンテントディスポジション文字

private java.lang.String	**key** ハッシュテーブル用キー
private java.lang.String	**mimeType** コンテントタイプ文字
private java.io.ByteArrayOutputStream	**ostream** ByteArrayOutputStream クラス
private java.util.Hashtable	**params** ハッシュテーブル
private int	**paramsLength** パラメータレングス
private int	**writebytes** 書き込みバイト数

コンストラクタの概要

Part()	

メソッドの概要

byte[]	**getBinaryContent**() バイナリコンテンツの取得
protected int	**getContentLength**() コンテンツ長の取得
java.util.Hashtable	**getParams**() 送られてきたHTTP Parameterを表すHashtableを取得する。
protected int	**getParamsLength**() パラメター文字列長の取得

boolean	**hasContents**() コンテンツがあるかどうかの判定
boolean	**isBinary**() このパートが持っているコンテンツがバイナリコンテンツかどうかの判定
protected void	**setContent**(byte[] buf) コンテンツの設定
protected void	**setContentLength**(int contentLength) コンテンツ長の設定
void	**setFileContent**(EntryValues entryValues, java.util.ArrayList fileContents) 共通領域にファイルコンテンツを格納
protected void	**setParams**(java.lang.String paramString) パラメータの設定
protected void	**setParamsLength**(int paramsLength) パラメータ文字列長の設定
private java.lang.String	**substring**(java.lang.String str, java.lang.String key) 先頭文字の削除
private java.lang.String	**trim**(java.lang.String str) 改行文字と最後の１文字の削除
private java.lang.String	**trimIn**(java.lang.String str) 改行文字の削除

フィールドの詳細

dataType

private final java.lang.String **dataType**

コンテントディスポジション文字

---

mimeType

private final java.lang.String **mimeType**

コンテントタイプ文字

---

binary

private boolean **binary**

バイナリ判定フラグ

---

paramsLength

private int **paramsLength**

パラメータレングス

---

contentLength

private int **contentLength**

コンテントレングス

---

writebytes

private int **writebytes**

書き込みバイト数

---

key

private java.lang.String **key**

ハッシュテーブル用キー

---

params

private java.util.Hashtable **params**

ハッシュテーブル

ostream

private java.io.ByteArrayOutputStream **ostream**

ByteArrayOutputStream クラス

コンストラクタの詳細

Part

public **Part**()

メソッドの詳細

getParams

public java.util.Hashtable **getParams**()

送られてきた HTTP Parameter を表す Hashtable を取得する。

戻り値:

Hashtable HTTP Parameter を表す Hashtable

---

hasContents

public boolean **hasContents**()

コンテンツがあるかどうかの判定

戻り値:

コンテンツがある場合は、true,それ以外は false

---

isBinary

public boolean **isBinary**()

このパートが持っているコンテンツがバイナリコンテンツかどうかの判定

戻り値:

持っているコンテンツがバイナリコンテンツの場合は、true,それ以外は false

---

setParams

protected void **setParams**(java.lang.String paramString)

パラメータの設定

パラメータ:

String - paramString 設定するパラメータ文字列

---

setParamsLength

protected void **setParamsLength**(int paramsLength)

パラメータ文字列長の設定

パラメータ:

int - paramLength パラメータ文字列長

---

getParamsLength

protected int **getParamsLength**()

パラメター文字列長の取得

戻り値:

int パラメータ文字列長

---

setContentLength

protected void **setContentLength**(int contentLength)

コンテンツ長の設定

パラメータ:

int - contentLength コンテンツ長

---

getContentLength

protected int **getContentLength**()

コンテンツ長の取得

戻り値:

int コンテンツ長

---

substring

private java.lang.String **substring**(java.lang.String str,
java.lang.String key)

先頭文字の削除

パラメータ:

str - 変換元文字

key - 先頭文字

戻り値:

変換後文字

---

trim

private java.lang.String **trim**(java.lang.String str)

改行文字と最後の１文字の削除

パラメータ:

str - 変換前文字

戻り値:

変換後文字

---

trimln

private java.lang.String **trimln**(java.lang.String str)

改行文字の削除

パラメータ:

str - 変換前文字

戻り値:

変換後文字

---

setContent

protected void **setContent**(byte[] buf)

throws java.io.UnsupportedEncodingException

コンテンツの設定

パラメータ:

byte[] - buf コンテンツ

---

setFileContent

public void **setFileContent**(EntryValues entryValues,

java.util.ArrayList fileContents)

共通領域にファイルコンテンツを格納

パラメータ:

entryValues - 共通領域

fileContents - ファイルコンテンツ

---

getBinaryContent

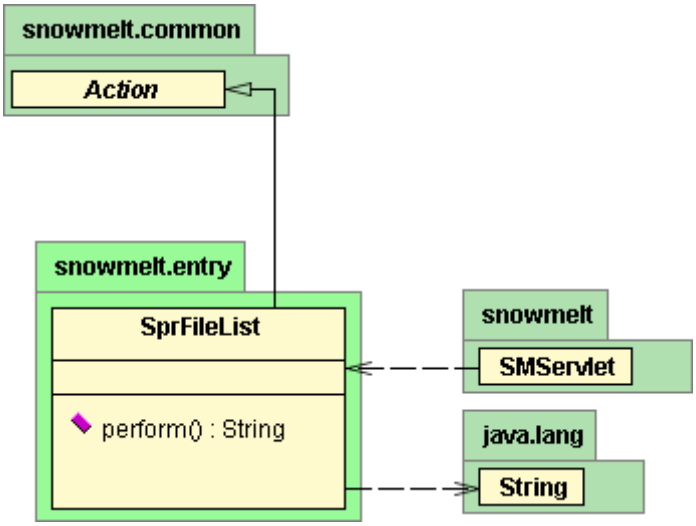
public byte[] **getBinaryContent**()

バイナリコンテンツの取得

戻り値:

byte[] バイナリコンテンツのバイト配列

■ ５　クラス SprFileList




```
java.lang.Object
  |
  +--snowmelt.common.Action
        |
        +--snowmelt.entry.SprFileList
```

---

public class **SprFileList**

extends Action

散布作業記録ファイル選択

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

SprFileList()

メソッドの概要

java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

コンストラクタの詳細

SprFileList

public **SprFileList**()

メソッドの詳細

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

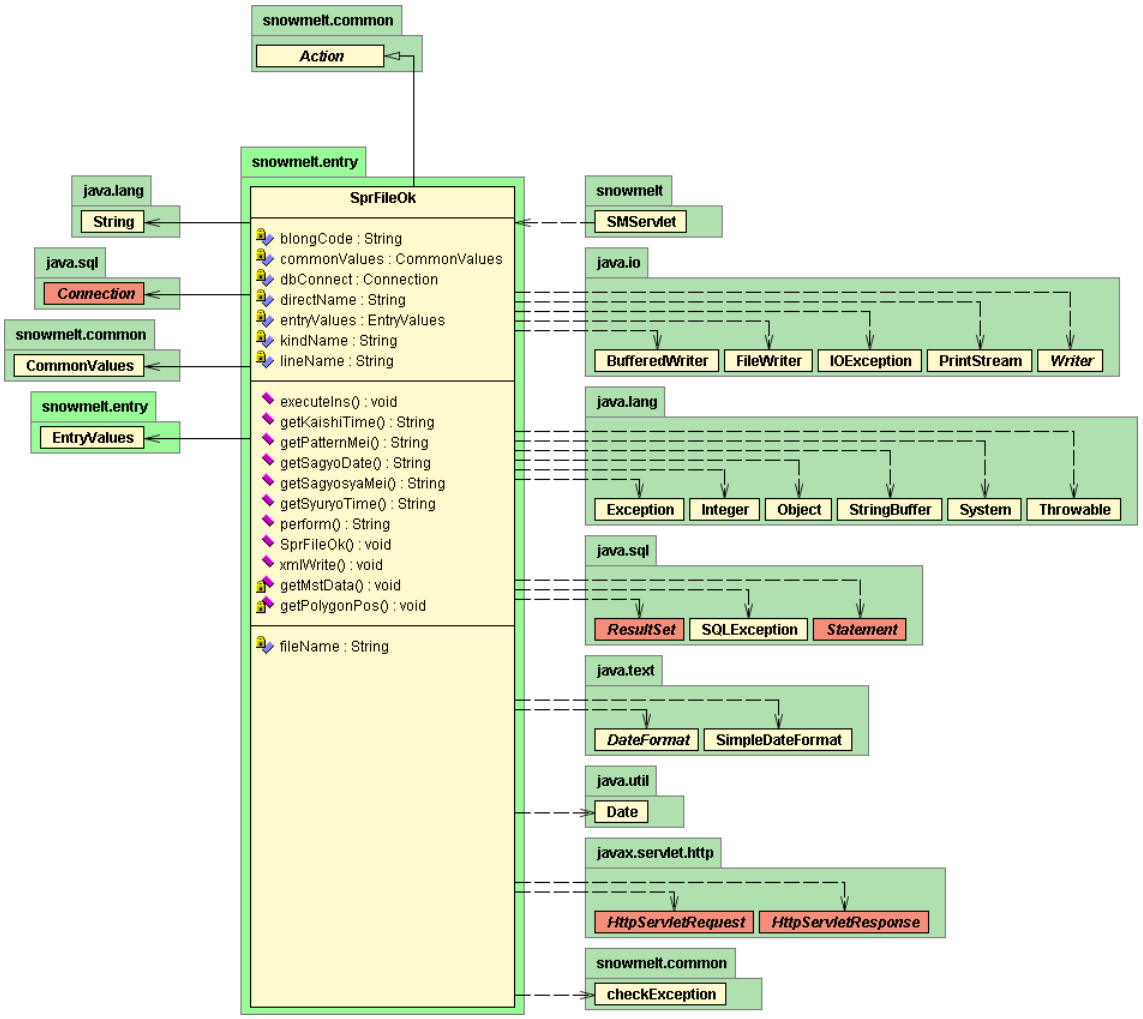
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/C11.jsp"

■ ６　クラス SprFileOk



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.entry.SprFileOk**

---

public class **SprFileOk**

extends Action

散布作業記録ファイル登録完了クラス

フィールドの概要

private java.lang.String	**blongCode** 所属コード
private CommonValues	**commonValues** 共通変数クラスのインスタンス
private java.sql.Connection	**dbConnect** Connection クラス
private java.lang.String	**directName** 散布方向名称
private EntryValues	**entryValues** 入力値維持クラスのインスタンス
private java.lang.String	**fileName** 作業記録保存ファイル名
private java.lang.String	**kindName** 薬剤名称
private java.lang.String	**lineName** 車線名

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

SprFileOk(java.sql.Connection dbConn, CommonValues commonVal, EntryValues entryVal, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response)
コンストラクタ

メソッドの概要

void	**executeIns**() データ更新処理
private java.lang.String	**getFileName**() 受け渡した文字列（ファイル名）に現在の日時をつけて返す
java.lang.String	**getKaishiTime**() 作業開始時間を返す
private void	**getMstData**(int datald) マスターＤＢより名称を取得する
java.lang.String	**getPatternMei**() 散布パターン名を返す
private void	**getPolygonPos**(int syukusyakuld, int datald, java.io.BufferedWriter workWriter) ポリゴンの座標を返す
java.lang.String	**getSagyoDate**() 作業日を返す
java.lang.String	**getSagyosyaMei**() 作業者名を返す
java.lang.String	**getSyuryoTime**() 作業終了時間を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名
void	**xmlWrite**() XML ファイルに書き込み

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues commonValues

共通変数クラスのインスタンス

---

entryValues

private EntryValues entryValues

入力値維持クラスのインスタンス

---

dbConnect

private java.sql.Connection dbConnect

Connection クラス

---

blongCode

private java.lang.String blongCode

所属コード

---

fileName

private java.lang.String fileName

作業記録保存ファイル名

---

lineName

private java.lang.String lineName

車線名

---

directName

private java.lang.String **directName**

散布方向名称

---

kindName

private java.lang.String **kindName**

薬剤名称

コンストラクタの詳細

SprFileOk

public **SprFileOk**(java.sql.Connection dbConn,
CommonValues commonVal,
EntryValues entryVal,
javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws checkException

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonVal - CommonValues クラス

entryVal - EntryValues クラス

request - HttpServletRequest クラス

responseHttpServletResponse クラス -

例外:

checkException -

メソッドの詳細

executeIns

public void **executeIns**()
throws checkException

データ更新処理

例外:

checkException -

---

xmlWrite

public void **xmlWrite**()

XML ファイルに書き込み

---

getFileName

private java.lang.String **getFileName**()

受け渡した文字列（ファイル名）に現在の日時をつけて返す

戻り値:

"spr" + 現在の日時 + ".xml"

---

getPolygonPos

private void **getPolygonPos**(int syukusyakuId,
int dataId,
java.io.BufferedWriter workWriter)

ポリゴンの座標を返す

パラメータ:

syukusyakuId - 縮尺ＩＤ

dataId - データＩＤ

workWriter - BufferedWriter クラス

---

getMstData

private void **getMstData**(int dataId)

マスターＤＢより名称を取得する

パラメータ:

dataId - データＩＤ

---

getSagyoDate

public java.lang.String **getSagyoDate**()

作業日を返す

戻り値:

作業日

---

getKaishiTime

public java.lang.String **getKaishiTime**()

作業開始時間を返す

戻り値:

作業開始時間

---

getSyuryoTime

public java.lang.String **getSyuryoTime**()

作業終了時間を返す

戻り値:

作業終了時間

---

getSagyosyaMei

public java.lang.String **getSagyosyaMei**()

作業者名を返す

戻り値:

作業者名

---

getPatternMei

public java.lang.String **getPatternMei**()

散布パターン名を返す

戻り値:

散布パターン名

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

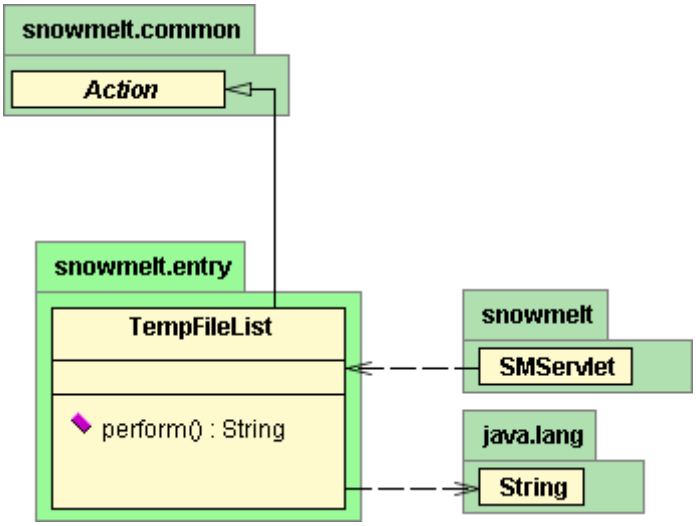
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/C12.jsp"

■ 7　クラス TempFileList



java.lang.Object
  |
  +--snowmelt.common.Action
        |
        +--**snowmelt.entry.TempFileList**

---

public class **TempFileList**
extends Action

巡回記録ファイル選択

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

TempFileList()

メソッドの概要

java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

コンストラクタの詳細

TempFileList

public **TempFileList**()

メソッドの詳細

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

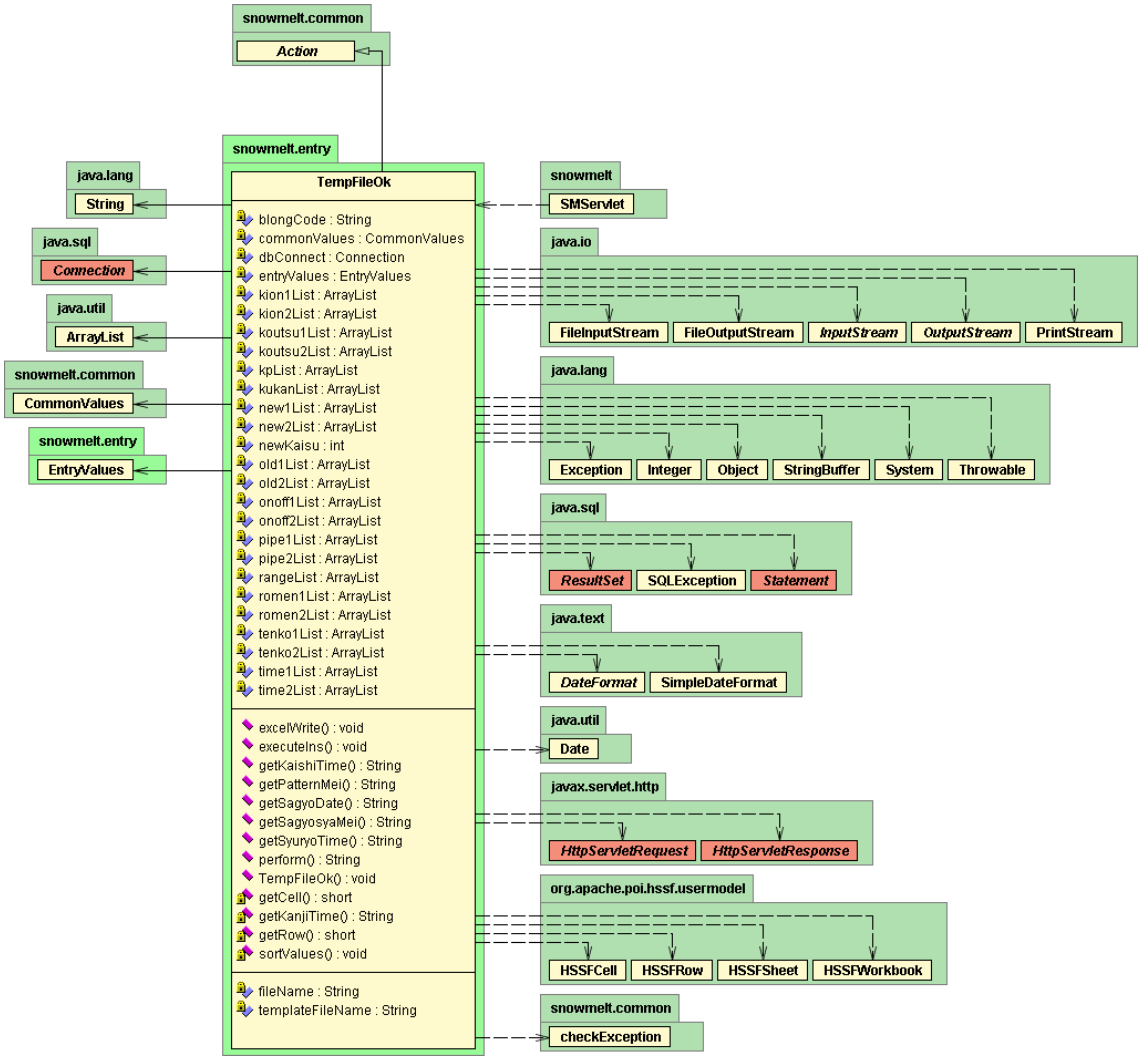
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/C21.jsp"

■ ８　クラス TempFileOk




```
java.lang.Object
  |
  +--snowmelt.common.Action
        |
        +--snowmelt.entry.TempFileOk
```

---

public class **TempFileOk**

extends Action

巡回記録ファイル登録完了

フィールドの概要

private java.lang.String	**blongCode** 所属コード
private CommonValues	**commonValues** 共通変数クラスのインスタンス
private java.sql.Connection	**dbConnect** Connection クラス
private EntryValues	**entryValues** 入力値維持クラスのインスタンス
private java.lang.String	**fileName** 巡回記録保存 Excel ファイル名
private java.util.ArrayList	**kion1List** 巡回記録格納リスト（行き気温）
private java.util.ArrayList	**kion2List** 巡回記録格納リスト（帰り気温）
private java.util.ArrayList	**koutsu1List** 巡回記録格納リスト（行き交通状況）
private java.util.ArrayList	**koutsu2List** 巡回記録格納リスト（帰り交通状況）
private java.util.ArrayList	**kpList** 巡回記録格納リスト（ＫＰ）
private java.util.ArrayList	**kukanList** 巡回記録格納リスト（区間名）
private java.util.ArrayList	**new1List** 巡回記録格納リスト（行き新雪深）
private java.util.ArrayList	**new2List** 巡回記録格納リスト（帰り新雪深）
private int	**newKaisu** 巡回回数
private java.util.ArrayList	**old1List**

	巡回記録格納リスト（行き圧雪深）
private java.util.ArrayList	old2List 巡回記録格納リスト（帰り圧雪深）
private java.util.ArrayList	onoff1List 巡回記録格納リスト（行き消雪パイプ稼動状況）
private java.util.ArrayList	onoff2List 巡回記録格納リスト（帰り消雪パイプ稼動状況）
private java.util.ArrayList	pipe1List 巡回記録格納リスト（行き消雪パイプ名）
private java.util.ArrayList	pipe2List 巡回記録格納リスト（帰り消雪パイプ名）
private java.util.ArrayList	rangeList 巡回記録格納リスト（範囲）
private java.util.ArrayList	romen1List 巡回記録格納リスト（行き路面状況）
private java.util.ArrayList	romen2List 巡回記録格納リスト（帰り路面状況）
private java.lang.String	templateFileName Excel テンプレートファイル名
private java.util.ArrayList	tenko1List 巡回記録格納リスト（行き天候）
private java.util.ArrayList	tenko2List 巡回記録格納リスト（帰り天候）
private java.util.ArrayList	time1List 巡回記録格納リスト（行き時刻）
private java.util.ArrayList	time2List 巡回記録格納リスト（帰り自国）

クラス snowmelt.common.Action から継承したフィールド

servlet

コンストラクタの概要

TempFileOk(java.sql.Connection dbConn, CommonValues commonVal, EntryValues entryVal, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response)

コンストラクタ

メソッドの概要

void	**excelWrite**() Excel ファイルに書き込み
void	**executeIns**() データ更新処理
private short	**getCell**(java.lang.String cell) Excel のセル番号文字列（ex."AB16"）を与えて、列番号を返す
private java.lang.String	**getFileName**() 受け渡した文字列（ファイル名）に現在の日時をつけて返す
java.lang.String	**getKaishiTime**() 作業開始時間を返す
private java.lang.String	**getKanjiTime**(java.lang.String str) 時間(09:07)を 9 時 07 分に変換する
java.lang.String	**getPatternMei**() 散布パターン名を返す
private short	**getRow**(java.lang.String cell) Excel のセル番号文字列（ex."AB16"）を与えて、行番号

	を返す
java.lang.String	**getSagyoDate**() 作業日を返す
java.lang.String	**getSagyosyaMei**() 作業者名を返す
java.lang.String	**getSyuryoTime**() 作業終了時間を返す
private java.lang.String	**getTemplateFileName**() テンプレート Excel ファイル名を DB から取得する
java.lang.String	**perform**() 遷移先ｊｓｐ名
private void	**sortValues**() 行き・帰りのデータを１つにまとめる

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

entryValues

private EntryValues **entryValues**

入力値維持クラスのインスタンス

---

dbConnect

private java.sql.Connection dbConnect

Connection クラス

---

blongCode

private java.lang.String blongCode

所属コード

---

templateFileName

private java.lang.String templateFileName

Excel テンプレートファイル名

---

fileName

private java.lang.String fileName

巡回記録保存 Excel ファイル名

---

newKaisu

private int newKaisu

巡回回数

---

kukanList

private java.util.ArrayList kukanList

巡回記録格納リスト（区間名）

---

kpList

private java.util.ArrayList kpList

巡回記録格納リスト（ＫＰ）

rangeList

private java.util.ArrayList rangeList

巡回記録格納リスト（範囲）

---

time1List

private java.util.ArrayList time1List

巡回記録格納リスト（行き時刻）

---

time2List

private java.util.ArrayList time2List

巡回記録格納リスト（帰り自国）

---

kion1List

private java.util.ArrayList kion1List

巡回記録格納リスト（行き気温）

---

kion2List

private java.util.ArrayList kion2List

巡回記録格納リスト（帰り気温）

---

tenko1List

private java.util.ArrayList tenko1List

巡回記録格納リスト（行き天候）

---

tenko2List

private java.util.ArrayList **tenko2List**

巡回記録格納リスト（帰り天候）

---

romen1List

private java.util.ArrayList **romen1List**

巡回記録格納リスト（行き路面状況）

---

romen2List

private java.util.ArrayList **romen2List**

巡回記録格納リスト（帰り路面状況）

---

koutsu1List

private java.util.ArrayList **koutsu1List**

巡回記録格納リスト（行き交通状況）

---

koutsu2List

private java.util.ArrayList **koutsu2List**

巡回記録格納リスト（帰り交通状況）

---

new1List

private java.util.ArrayList **new1List**

巡回記録格納リスト（行き新雪深）

---

new2List

private java.util.ArrayList **new2List**

巡回記録格納リスト（帰り新雪深）

---

old1List

private java.util.ArrayList old1List

巡回記録格納リスト（行き圧雪深）

---

old2List

private java.util.ArrayList old2List

巡回記録格納リスト（帰り圧雪深）

---

pipe1List

private java.util.ArrayList pipe1List

巡回記録格納リスト（行き消雪パイプ名）

---

pipe2List

private java.util.ArrayList pipe2List

巡回記録格納リスト（帰り消雪パイプ名）

---

onoff1List

private java.util.ArrayList onoff1List

巡回記録格納リスト（行き消雪パイプ稼動状況）

---

onoff2List

private java.util.ArrayList onoff2List

巡回記録格納リスト（帰り消雪パイプ稼動状況）

コンストラクタの詳細

TempFileOk

public **TempFileOk**(java.sql.Connection dbConn,
CommonValues commonVal,
EntryValues entryVal,
javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws checkException

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonVal - CommonValues クラス

entryVal - EntryValues クラス

request - HttpServletRequest クラス

response - HttpServletResponse クラス

例外:

checkException -

メソッドの詳細

executeIns

public void **executeIns**()
throws checkException

データ更新処理

例外:

checkException -

---

excelWrite

public void **excelWrite**()

Excel ファイルに書き込み

---

sortValues

private void **sortValues**()

行き・帰りのデータを１つにまとめる

---

getKanjiTime

private java.lang.String **getKanjiTime**(java.lang.String str)

時間(09:07)を 9 時 07 分に変換する

パラメータ:

str - 時間(09:07)

戻り値:

変換後時間(9 時 07 分)

---

getCell

private short **getCell**(java.lang.String cell)

Excel のセル番号文字列（ex."AB16"）を与えて、列番号を返す

パラメータ:

cell - Excel のセル番号文字列

戻り値:

列番号

---

getRow

private short **getRow**(java.lang.String cell)

Excel のセル番号文字列（ex."AB16"）を与えて、行番号を返す

パラメータ:

cell - Excel のセル番号文字列

戻り値:

行番号

---

getTemplateFileName

private java.lang.String **getTemplateFileName**()

テンプレート Excel ファイル名を DB から取得する

戻り値:

テンプレート Excel ファイル名

---

getFileName

private java.lang.String **getFileName**()

受け渡した文字列（ファイル名）に現在の日時をつけて返す

戻り値:

テンプレート Excel + 現在の日時 + ".xls"

---

getSagyoDate

public java.lang.String **getSagyoDate**()

作業日を返す

戻り値:

作業日

---

getKaishiTime

public java.lang.String **getKaishiTime**()

作業開始時間を返す

戻り値:

作業開始時間

---

getSyuryoTime

public java.lang.String **getSyuryoTime**()

作業終了時間を返す

戻り値:

作業終了時間

---

getSagyosyaMei

public java.lang.String **getSagyosyaMei**()

作業者名を返す

戻り値:

作業者名

---

getPatternMei

public java.lang.String **getPatternMei**()

散布パターン名を返す

戻り値:

散布パターン名

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

オーバーライド:

クラス Action 内の perform

戻り値:

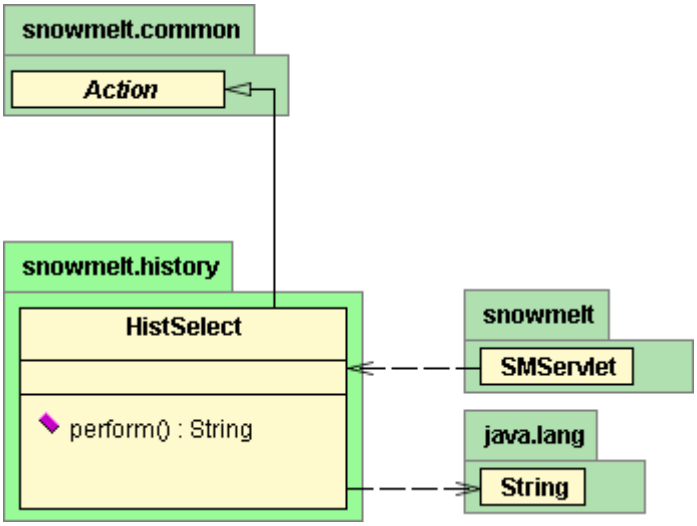
"/jsp/C22.jsp"

---

f） パッケージ snowmelt.history

クラスの概要	
HistSelect	履歴選択クラス
SprinkleList	散布履歴日時選択クラス
SprinkleView	散布履歴表示クラス
TempertureList	巡回記録表示履歴選択クラス
TempertureView	巡回記録表示履歴選択クラス
TempUpdateFileList	巡回記録ファイル修正版選択
TempUpdateFileOk	巡回記録ファイル選択

■ 1　クラス HistSelect



java.lang.Object
  |
  +--snowmelt.common.Action
        |
        +--**snowmelt.history.HistSelect**

---

public class **HistSelect**
extends Action

履歴選択クラス

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

HistSelect()

メソッドの概要

java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

コンストラクタの詳細

HistSelect

public **HistSelect**()

メソッドの詳細

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

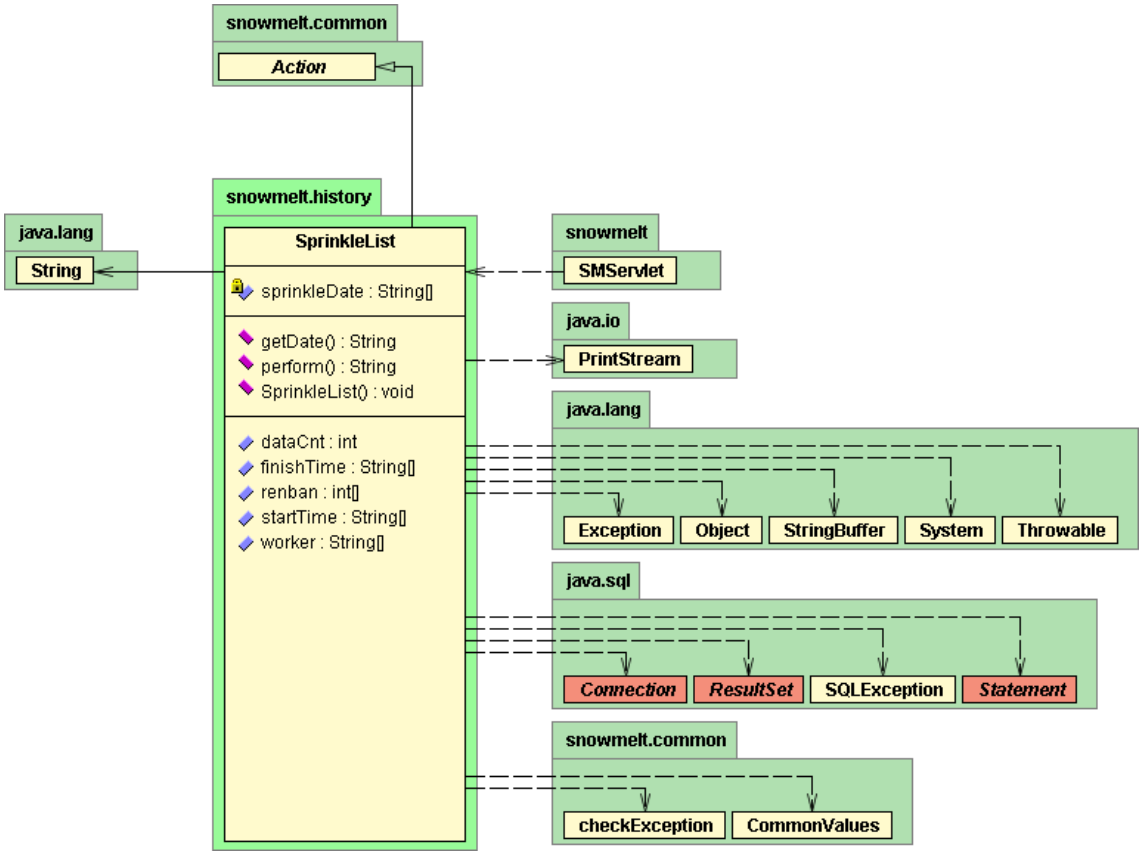
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/B01.jsp"

■ 2　クラス SprinkleList



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.history.SprinkleList**

---

public class **SprinkleList**

extends Action

散布履歴日時選択クラス

フィールドの概要

private　int	**dataCnt** データ数
private　java.lang.String[]	**finishTime**

	終了時間配列
private int[]	**renban** 連番配列
private java.lang.String[]	**sprinkleDate** 作業日配列
private java.lang.String[]	**startTime** 開始時間配列
private java.lang.String[]	**worker** 作業者名配列

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

SprinkleList(java.sql.Connection dbConn, CommonValues commonValues) コンストラクタ

メソッドの概要

int	**getDataCnt**() データ数を返す
java.lang.String	**getDate**(int dataIndex) 作業日付を返す
java.lang.String	**getFinishTime**(int dataIndex) 作業終了時間を返す
int	**getRenban**(int dataIndex)

	連番を返す
java.lang.String	**getStartTime**(int dataIndex) 作業開始時間を返す
java.lang.String	**getWorker**(int dataIndex) 作業者名を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

dataCnt

private int **dataCnt**

データ数

---

renban

private int[] **renban**

連番配列

---

sprinkleDate

private java.lang.String[] **sprinkleDate**

作業日配列

---

startTime

private java.lang.String[] **startTime**

開始時間配列

---

finishTime

private java.lang.String[] **finishTime**

終了時間配列

---

worker

private java.lang.String[] **worker**

作業者名配列

コンストラクタの詳細

SprinkleList

public **SprinkleList**(java.sql.Connection dbConn,
CommonValues commonValues)
throws checkException

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonValues - CommonValues クラス

メソッドの詳細

getRenban

public int **getRenban**(int dataIndex)

連番を返す

パラメータ:

dataIndex - インデックス

戻り値:

連番

---

getDate

public java.lang.String **getDate**(int dataIndex)

作業日付を返す

パラメータ:

dataIndex - インデックス

戻り値:

作業日付

---

getStartTime

public java.lang.String **getStartTime**(int dataIndex)

作業開始時間を返す

パラメータ:

dataIndex - インデックス

戻り値:

作業開始時間

---

getFinishTime

public java.lang.String **getFinishTime**(int dataIndex)

作業終了時間を返す

パラメータ:

dataIndex - インデックス

戻り値:

作業終了時間

---

getWorker

public java.lang.String **getWorker**(int dataIndex)

作業者名を返す

パラメータ:

dataIndex - インデックス

戻り値:

作業者名

---

getDataCnt

public int **getDataCnt**()

データ数を返す

戻り値:

データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

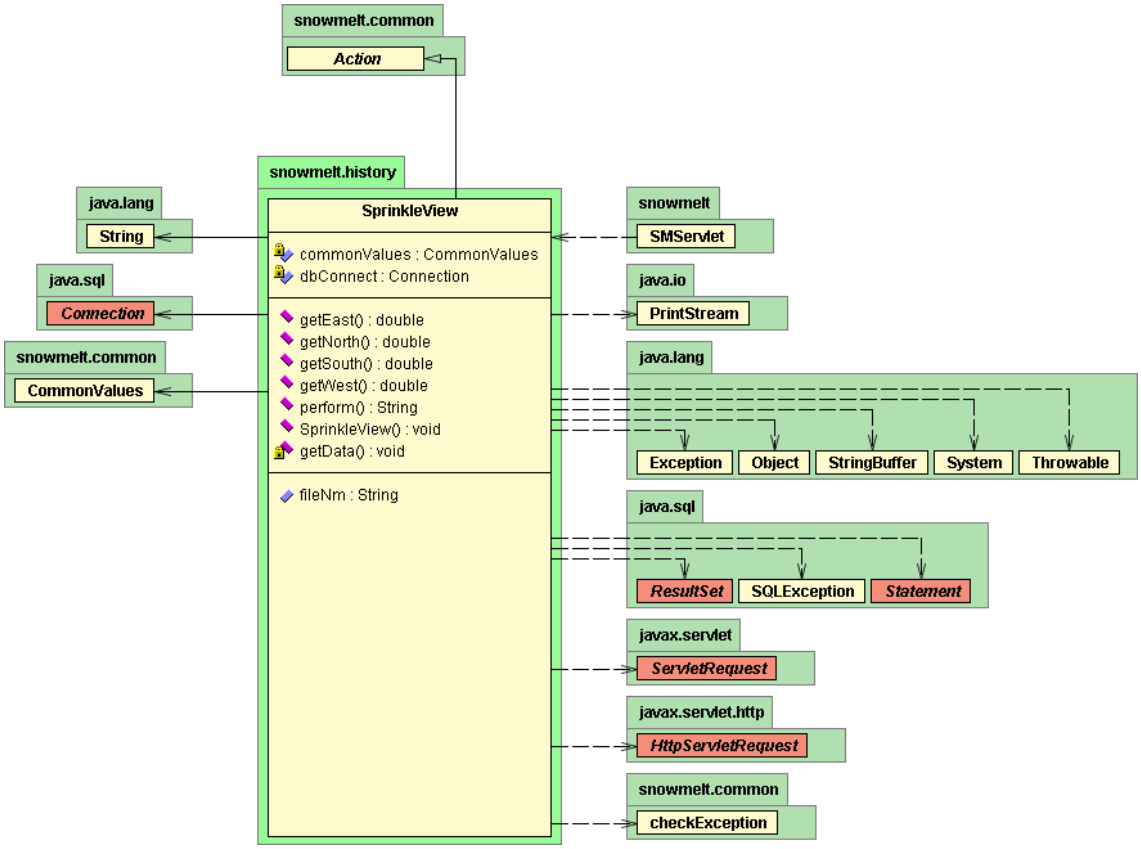
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/B11.jsp"

■ 3　クラス SprinkleView



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.history.SprinkleView**

---

public class **SprinkleView**

extends Action

散布履歴表示クラス

フィールドの概要

private　CommonValues	**commonValues** 共通変数クラスのインスタンス
private　java.sql.Connection	**dbConnect**

	Connection クラス
private java.lang.String	**fileNm** 履歴ファイル名

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

SprinkleView(java.lang.String id, java.sql.Connection dbConn, CommonValues commonVal, javax.servlet.http.HttpServletRequest request)
コンストラクタ

メソッドの概要

private void	**getData**(java.lang.String sSqlStr) データ取得
double	**getEast**() 西経を返す
java.lang.String	**getFileNm**() ファイル名を返す
double	**getNorth**() 北緯を返す
double	**getSouth**() 南緯を返す
double	**getWest**() 東経を返す

java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

dbConnect

private java.sql.Connection **dbConnect**

Connection クラス

---

fileNm

private java.lang.String **fileNm**

履歴ファイル名

コンストラクタの詳細

SprinkleView

public **SprinkleView**(java.lang.String id,
java.sql.Connection dbConn,
CommonValues commonVal,

javax.servlet.http.HttpServletRequest request)
throws checkException

コンストラクタ

パラメータ:

id - アクションＩＤ

dbConn - Connection クラス

commonVal - CommonValues クラス

request - HttpServletRequest クラス

例外:

checkException -

メソッドの詳細

getData

private void **getData**(java.lang.String sSqlStr)
throws checkException

データ取得

パラメータ:

sSqlStr - ＳＱＬ

---

getWest

public double **getWest**()

東経を返す

戻り値:

東経

---

getEast

public double **getEast**()

西経を返す

戻り値:

西経

getNorth

public double **getNorth**()

北緯を返す

戻り値:

北緯

---

getSouth

public double **getSouth**()

南緯を返す

戻り値:

南緯

---

getFileNm

public java.lang.String **getFileNm**()

ファイル名を返す

戻り値:

ファイル名

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

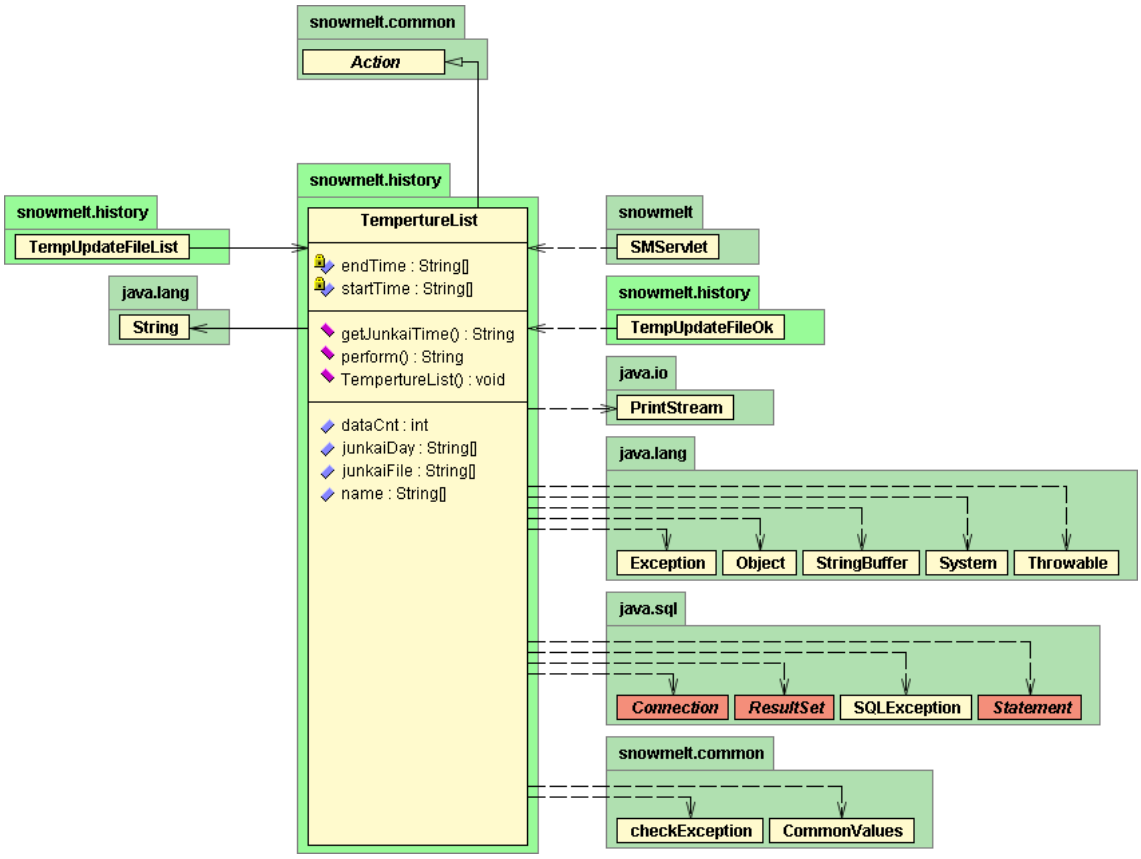
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/B12.jsp"

■ ４　クラス TempertureList



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.history.TempertureList**

---

public class **TempertureList**

extends Action

巡回記録表示履歴選択クラス

フィールドの概要

private int	**dataCnt** データ数
private java.lang.String[]	**endTime**

	終了時刻配列
private java.lang.String[]	**junkaiDay** 巡回日配列
private java.lang.String[]	**junkaiFile** 巡回記録ファイル名配列
private java.lang.String[]	**name** 巡回者名配列
private java.lang.String[]	**startTime** 開始時刻配列

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

TempertureList(java.sql.Connection dbConn, CommonValues commonValues)
コンストラクタ

メソッドの概要

int	**getDataCnt**() データ数を返す
java.lang.String	**getJunkaiDay**(int dataIndex) 巡回日を返す
java.lang.String	**getJunkaiFile**(int dataIndex) 帳票ファイル名を返す
java.lang.String	**getJunkaiTime**(int dataIndex)

	巡回時間を返す
java.lang.String	**getName**(int dataIndex) 運転者名を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

dataCnt

private int **dataCnt**

データ数

---

junkaiDay

private java.lang.String[] **junkaiDay**

巡回日配列

---

startTime

private java.lang.String[] **startTime**

開始時刻配列

---

endTime

private java.lang.String[] **endTime**

終了時刻配列

---

name

private java.lang.String[] **name**

巡回者名配列

---

junkaiFile

private java.lang.String[] **junkaiFile**

巡回記録ファイル名配列

コンストラクタの詳細

TempertureList

public **TempertureList**(java.sql.Connection dbConn,
CommonValues commonValues)
throws checkException

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonValues - CommonValues クラス

例外:

checkException -

メソッドの詳細

getJunkaiDay

public java.lang.String **getJunkaiDay**(int dataIndex)

巡回日を返す

パラメータ:

dataIndex - インデックス

戻り値:

巡回日

---

getJunkaiTime

public java.lang.String **getJunkaiTime**(int dataIndex)

巡回時間を返す

パラメータ:

dataIndex - インデックス

戻り値:

巡回時間

---

getName

public java.lang.String **getName**(int dataIndex)

運転者名を返す

パラメータ:

dataIndex - インデックス

戻り値:

運転者名

---

getJunkaiFile

public java.lang.String **getJunkaiFile**(int dataIndex)

帳票ファイル名を返す

パラメータ:

dataIndex - インデックス

戻り値:

帳票ファイル名

---

getDataCnt

public int **getDataCnt**()

データ数を返す

戻り値:

データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

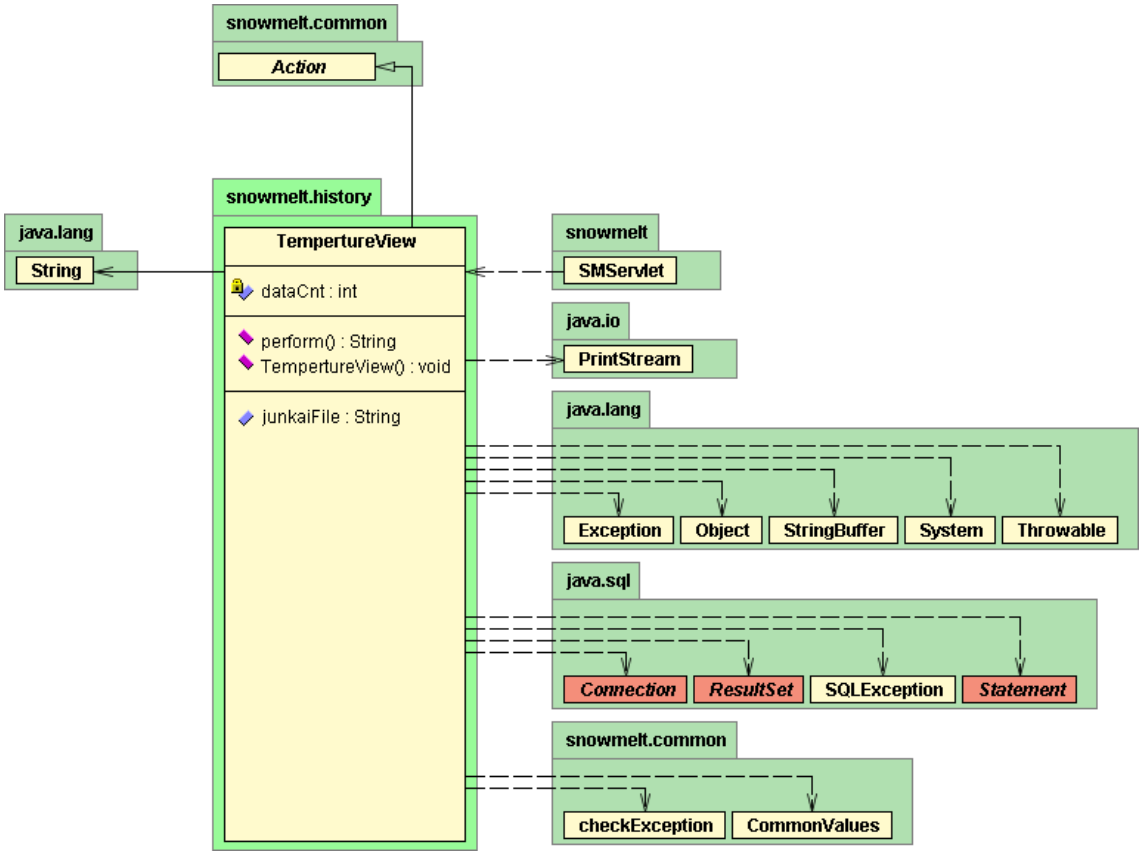
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/B21.jsp"

■ ５　クラス TempertureView



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.history.TempertureView**

---

public class **TempertureView**

extends Action

巡回記録表示履歴選択クラス

フィールドの概要

private int	**dataCnt** データ数
private java.lang.String	**junkaiFile**

	巡回ファイル名

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

TempertureView(java.sql.Connection dbConn, CommonValues commonValues) コンストラクタ

メソッドの概要

java.lang.String	**getJunkaiFile**() 帳票ファイル名を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

dataCnt

private int **dataCnt**

データ数

---

junkaiFile

private java.lang.String **junkaiFile**

巡回ファイル名

コンストラクタの詳細

TempertureView

public **TempertureView**(java.sql.Connection dbConn,
CommonValues commonValues)
throws checkException

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonValues - CommonValues クラス

例外:

checkException -

メソッドの詳細

getJunkaiFile

public java.lang.String **getJunkaiFile**()

帳票ファイル名を返す

戻り値:

帳票ファイル名

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

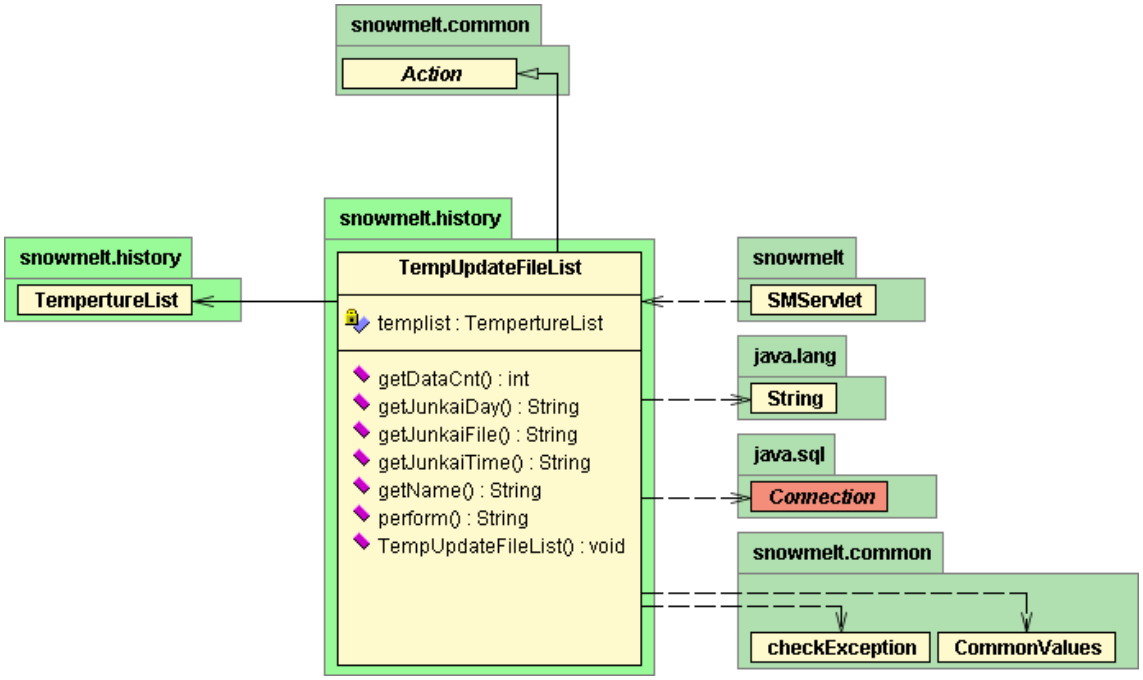
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/B22.jsp"

■ ６　クラス TempUpdateFileList



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.history.TempUpdateFileList**

---

public class **TempUpdateFileList**

extends Action

巡回記録ファイル修正版選択

フィールドの概要

private　TempertureList	**templist** TempertureList クラス

クラス **snowmelt.common.Action** から継承したフィールド

servlet

コンストラクタの概要

TempUpdateFileList(java.sql.Connection dbConn, CommonValues commonValues)

コンストラクタ

メソッドの概要

int	**getDataCnt**() データ数を返す
java.lang.String	**getJunkaiDay**(int dataIndex) 巡回日を返す
java.lang.String	**getJunkaiFile**(int dataIndex) 帳票ファイル名を返す
java.lang.String	**getJunkaiTime**(int dataIndex) 巡回時間を返す
java.lang.String	**getName**(int dataIndex) 運転者名を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド

charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

templist

private TempertureList **templist**

TempertureList クラス

コンストラクタの詳細

TempUpdateFileList

public **TempUpdateFileList**(java.sql.Connection dbConn,
CommonValues commonValues)
throws checkException

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonValues - CommonValues クラス

例外:

checkException -

メソッドの詳細

getJunkaiDay

public java.lang.String **getJunkaiDay**(int dataIndex)

巡回日を返す

パラメータ:

dataIndex - インデックス

戻り値:

巡回日

---

getJunkaiTime

public java.lang.String **getJunkaiTime**(int dataIndex)

巡回時間を返す

パラメータ:

dataIndex - インデックス

戻り値:

巡回時間

---

getName

public java.lang.String **getName**(int dataIndex)

運転者名を返す

パラメータ:

dataIndex - インデックス

戻り値:

運転者名

---

getJunkaiFile

public java.lang.String **getJunkaiFile**(int dataIndex)

帳票ファイル名を返す

パラメータ:

dataIndex - インデックス

戻り値:

帳票ファイル名

---

getDataCnt

public int **getDataCnt**()

データ数を返す

戻り値:

データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

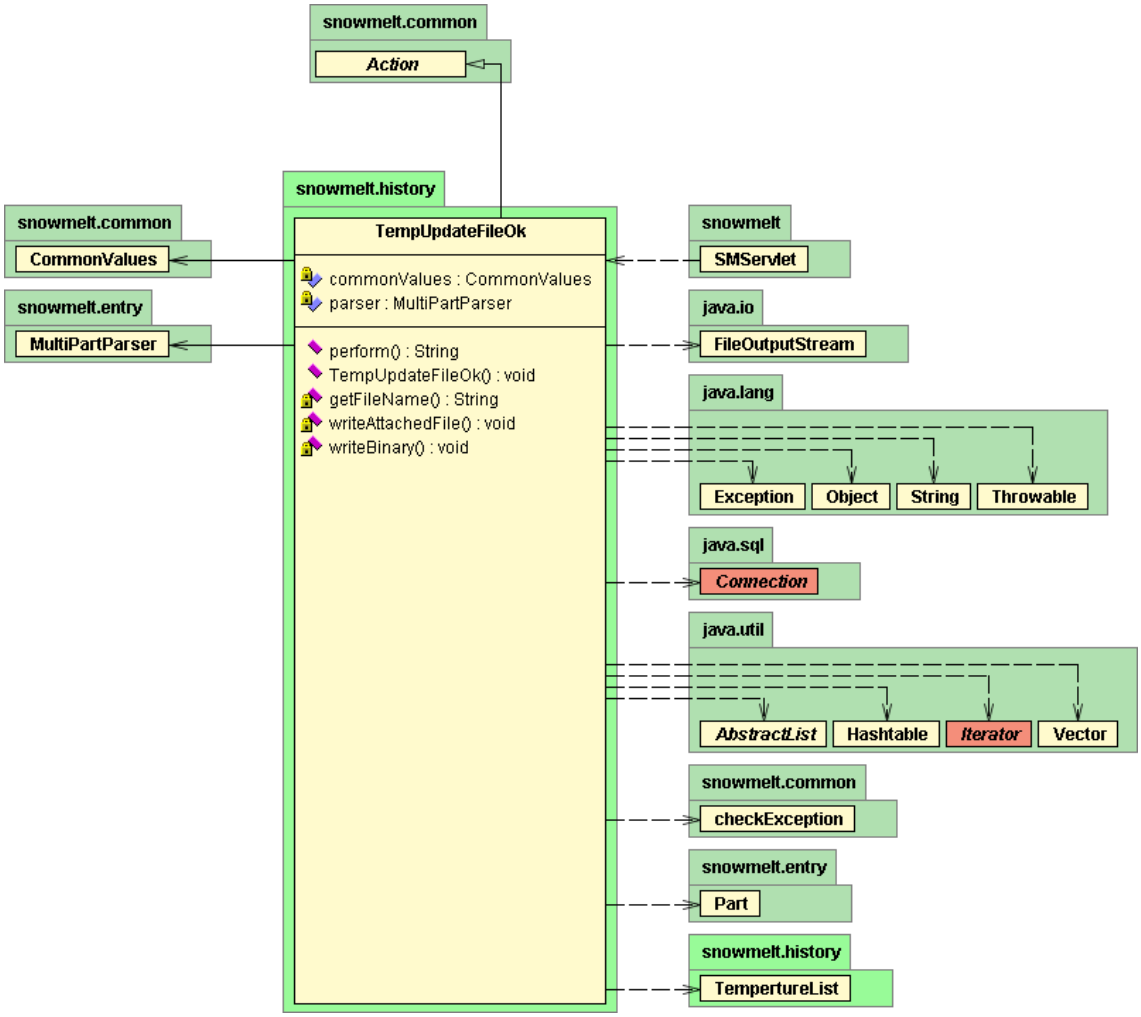
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/B31.jsp"

■ 7　クラス TempUpdateFileOk



java.lang.Object

  |

  +--snowmelt.common.Action

　　|

　　+--**snowmelt.history.TempUpdateFileOk**

---

public class **TempUpdateFileOk**

extends Action

巡回記録ファイル選択

フィールドの概要

private CommonValues	**commonValues** 共通変数クラスのインスタンス
private MultiPartParser	**parser** MultiPartParser クラスのインスタンス

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

TempUpdateFileOk(java.sql.Connection dbConn, CommonValues commonVal, MultiPartParser tmpParser)
コンストラクタ

メソッドの概要

private java.lang.String	**getFileName**(java.lang.String fname) クライアント側の完全ファイル名からファイル名だけを抽出する
java.lang.String	**perform**() 遷移先ｊｓｐ名
private void	**writeAttachedFile**(java.lang.String filename) 添付ファイルの書き込み
private void	**writeBinary**(Part part, java.lang.String outfilename) ファイルの書き出し

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

parser

private MultiPartParser **parser**

MultiPartParser クラスのインスタンス

---

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

コンストラクタの詳細

TempUpdateFileOk

public **TempUpdateFileOk**(java.sql.Connection dbConn,
CommonValues commonVal,
MultiPartParser tmpParser)
throws checkException

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonVal - CommonValues クラス

tmpParser - MultiPartParser クラス

例外:

checkException -

メソッドの詳細

writeAttachedFile

private void **writeAttachedFile**(java.lang.String filename)

添付ファイルの書き込み

パラメータ:

filename - 添付ファイル名

---

writeBinary

private void **writeBinary**(Part part,
java.lang.String outfilename)

ファイルの書き出し

パラメータ:

part - Part クラス

outfilename - 出力ファイル名

---

getFileName

private java.lang.String **getFileName**(java.lang.String fname)

クライアント側の完全ファイル名からファイル名だけを抽出する

パラメータ:

fname - クライアント側完全ファイル名

戻り値:

ファイル名

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

オーバーライド:

クラス Action 内の perform

戻り値:

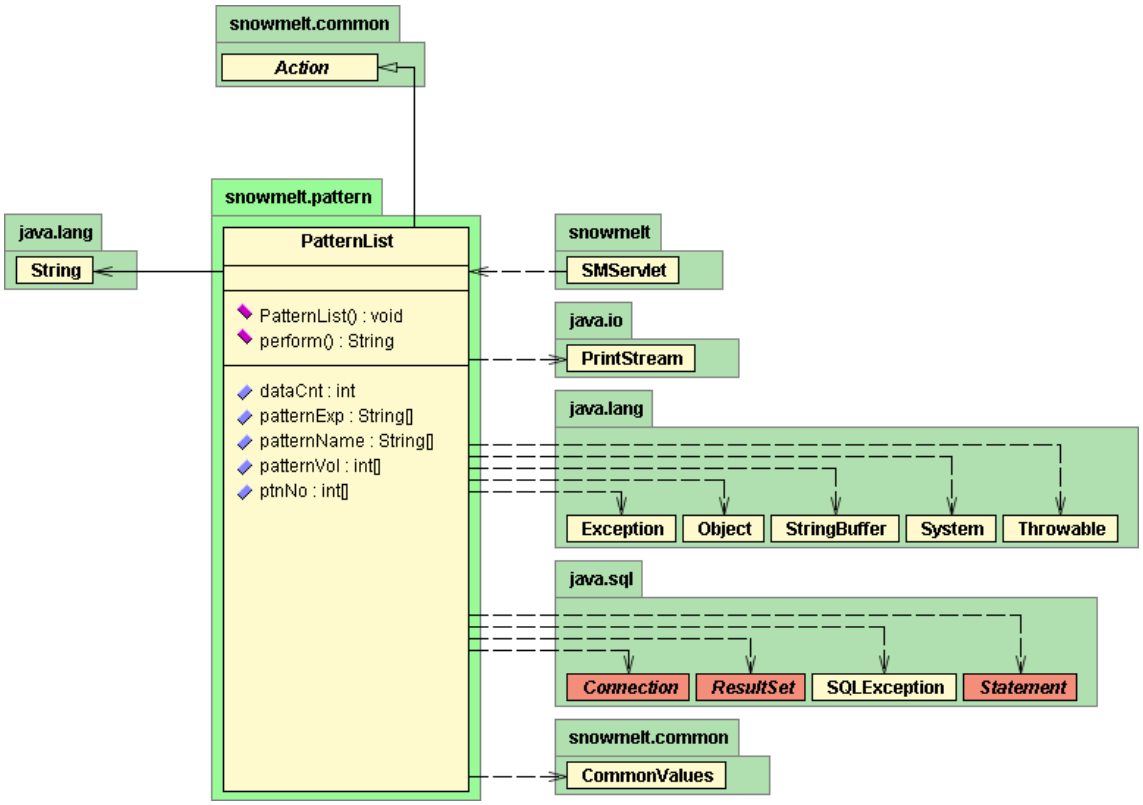
"/jsp/B32.jsp"

---

g） パッケージ snowmelt.pattern

クラスの概要	
PatternList	標準パターン一覧クラス
PatternOutput	散布計画出力クラス
PatternView	標準パターン確認クラス

■ 1　クラス PatternList



java.lang.Object

　|

　+--snowmelt.common.Action

　　|

　　+--**snowmelt.pattern.PatternList**

---

public class **PatternList**

extends Action

標準パターン一覧クラス

フィールドの概要	
private int	**dataCnt** データ数
private java.lang.String[]	**patternExp** パターン説明

private java.lang.String[]	**patternName** パターン名
private int[]	**patternVol** 総散布量
private int[]	**ptnNo** パターン番号

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

PatternList(java.sql.Connection dbConn, CommonValues commonValues) コンストラクタ

メソッドの概要

int	**getDataCnt**() データ数を返す
java.lang.String	**getPatternExp**(int dataIndex) パターン説明を返す
java.lang.String	**getPatternName**(int dataIndex) パターン名を返す
int	**getPatternVol**(int dataIndex) 総散布量を返す
int	**getPtnNo**(int dataIndex) パターン番号を返す

java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

dataCnt

private int **dataCnt**

データ数

---

ptnNo

private int[] **ptnNo**

パターン番号

---

patternName

private java.lang.String[] **patternName**

パターン名

---

patternExp

private java.lang.String[] **patternExp**

パターン説明

---

patternVol

private int[] **patternVol**

総散布量

コンストラクタの詳細

PatternList

public **PatternList**(java.sql.Connection dbConn,
CommonValues commonValues)

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonValues - CommonValues クラス

メソッドの詳細

getPtnNo

public int **getPtnNo**(int dataIndex)

パターン番号を返す

パラメータ:

dataIndex - インデックス

戻り値:

パターン番号

---

getPatternName

public java.lang.String **getPatternName**(int dataIndex)

パターン名を返す

パラメータ:

dataIndex - インデックス

戻り値:

パターン名

---

getPatternExp

public java.lang.String **getPatternExp**(int dataIndex)

パターン説明を返す

パラメータ:

dataIndex - インデックス

戻り値:

パターン説明

---

getPatternVol

public int **getPatternVol**(int dataIndex)

総散布量を返す

パラメータ:

dataIndex - インデックス

戻り値:

総散布量

---

getDataCnt

public int **getDataCnt**()

データ数を返す

戻り値:

データ数

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

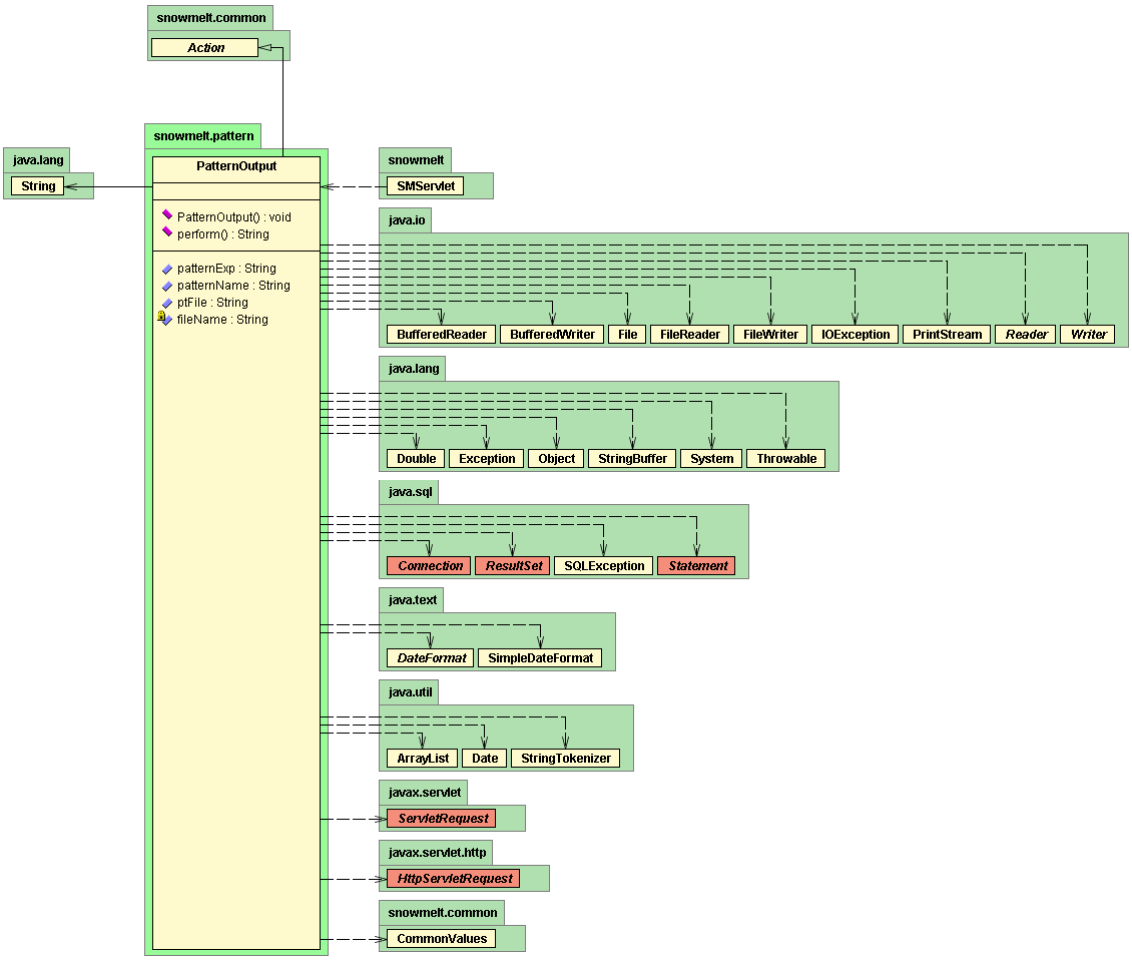
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/A01.jsp"

■ ２　クラス PatternOutput



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.pattern.PatternOutput**

---

public class **PatternOutput**

extends Action

散布計画出力クラス

フィールドの概要

private　java.lang.String	**fileName**

	作業記録保存ワークファイル名
private java.lang.String	**patternExp** パターン説明
private java.lang.String	**patternName** パターン名
private java.lang.String	**ptFile** パターンファイル名

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

PatternOutput(java.sql.Connection dbConn, CommonValues commonValues, javax.servlet.http.HttpServletRequest request)
コンストラクタ

メソッドの概要

private java.lang.String	**getFileName**() 受け渡した文字列（ファイル名）に現在の日時をつけて返す
java.lang.String	**getPatternExp**() パターン説明を返す
java.lang.String	**getPatternName**() パターン名を返す
java.lang.String	**getPtFile**()

	パターンファイル名を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

patternName

private java.lang.String **patternName**

パターン名

---

patternExp

private java.lang.String **patternExp**

パターン説明

---

ptFile

private java.lang.String **ptFile**

パターンファイル名

---

fileName

private java.lang.String **fileName**

作業記録保存ワークファイル名

コンストラクタの詳細

PatternOutput

public **PatternOutput**(java.sql.Connection dbConn,
CommonValues commonValues,
javax.servlet.http.HttpServletRequest request)

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonValues - CommonValues クラス

request - HttpServletRequest クラス

メソッドの詳細

getFileName

private java.lang.String **getFileName**()

受け渡した文字列（ファイル名）に現在の日時をつけて返す

戻り値:

"ptn" + 現在の日時 + ".csv"

---

getPatternName

public java.lang.String **getPatternName**()

パターン名を返す

戻り値:

パターン名

---

getPatternExp

public java.lang.String **getPatternExp**()

パターン説明を返す

戻り値:

パターン説明

---

getPtFile

public java.lang.String **getPtFile**()

パターンファイル名を返す

戻り値:

パターンファイル名

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

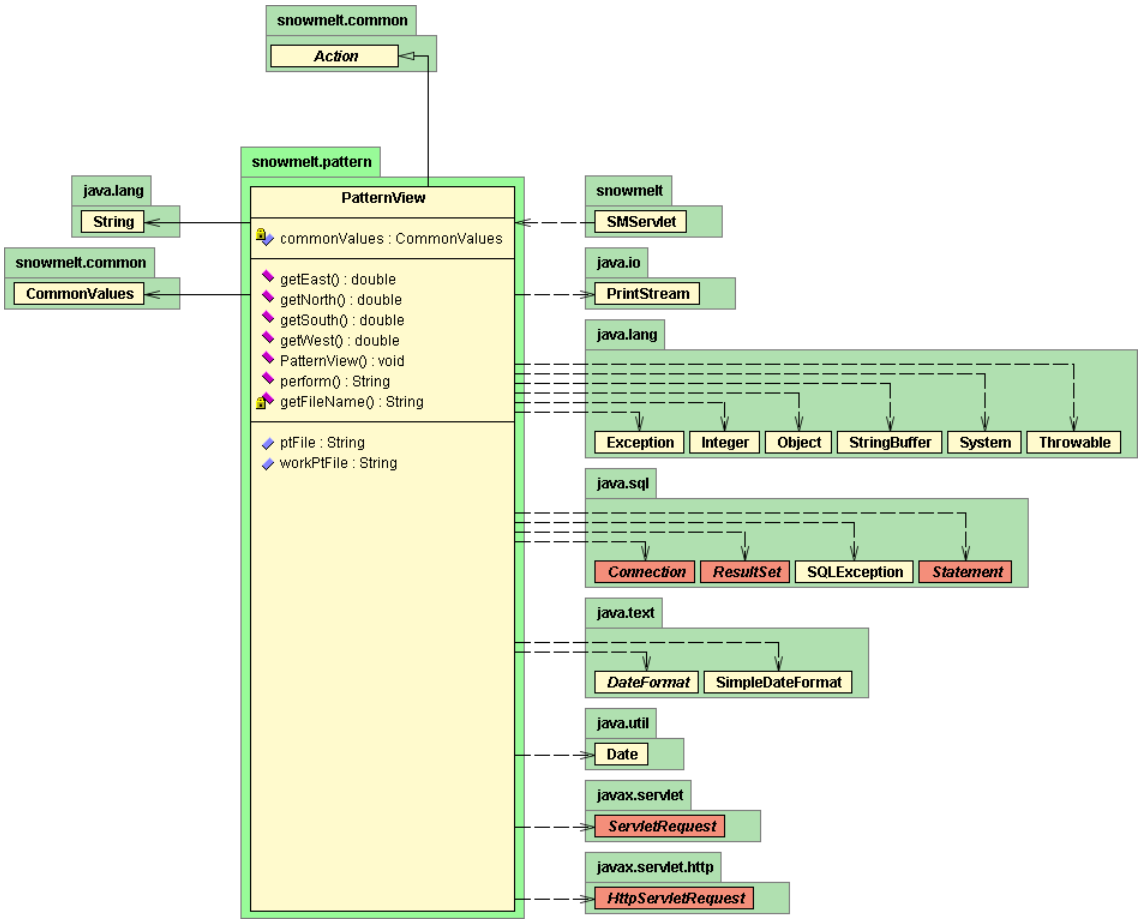
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/A03.jsp"

■ 3　クラス PatternView



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.pattern.PatternView**

---

public class **PatternView**

extends Action

標準パターン確認クラス

フィールドの概要

private CommonValues	**commonValues** 共通変数クラスのインスタンス

private java.lang.String	**ptFile** 作業記録保存ワークファイル名
private java.lang.String	**workPtFile** 一時作業ファイル名

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

PatternView(java.sql.Connection dbConn, CommonValues commonVal, javax.servlet.http.HttpServletRequest request)
コンストラクタ

メソッドの概要

double	**getEast**() 西経を返す
private java.lang.String	**getFileName**(java.lang.String PtnName) 受け渡した文字列（ファイル名）に現在の日時をつけて返す
double	**getNorth**() 北緯を返す
java.lang.String	**getPtFile**() ファイル名を返す
double	**getSouth**() 南緯を返す

double	**getWest**() 東経を返す
java.lang.String	**getWorkPtFile**() 一時作業ファイル名を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

ptFile

private java.lang.String **ptFile**

作業記録保存ワークファイル名

---

workPtFile

private java.lang.String **workPtFile**

一時作業ファイル名

コンストラクタの詳細

PatternView

public **PatternView**(java.sql.Connection dbConn,
CommonValues commonVal,
javax.servlet.http.HttpServletRequest request)

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonVal - CommonValues クラス

request - HttpServletRequest クラス

メソッドの詳細

getFileName

private java.lang.String **getFileName**(java.lang.String PtnName)

受け渡した文字列（ファイル名）に現在の日時をつけて返す

パラメータ:

PtnName - 標準パターン名

戻り値:

標準パターン名 ＋ 現在の日時 ＋".csv"

---

getWest

public double **getWest**()

東経を返す

戻り値:

東経

---

getEast

public double **getEast**()

西経を返す

戻り値:

西経

getNorth

public double **getNorth**()

北緯を返す

戻り値:

北緯

---

getSouth

public double **getSouth**()

南緯を返す

戻り値:

南緯

---

getPtFile

public java.lang.String **getPtFile**()

ファイル名を返す

戻り値:

ファイル名

---

getWorkPtFile

public java.lang.String **getWorkPtFile**()

一時作業ファイル名を返す

戻り値:

一時作業ファイル名

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

オーバーライド:

クラス Action 内の perform

戻り値:

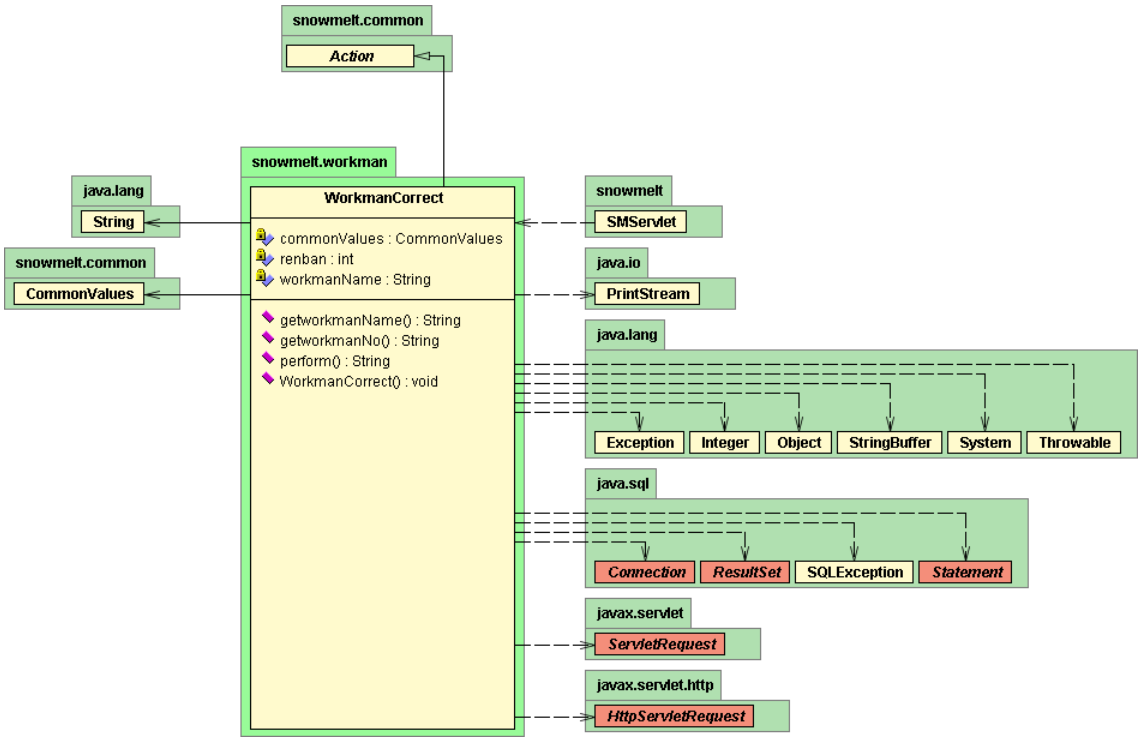
"/jsp/A02.jsp"

---

h） パッケージ snowmelt.workman

クラスの概要	
WorkmanCorrect	実験システム作業者名リスト作成　クラス
WorkmanList	実験システム作業者名リスト作成　クラス

■ 1　クラス WorkmanCorrect



java.lang.Object

  |

  +--snowmelt.common.Action

        |

        +--**snowmelt.workman.WorkmanCorrect**

---

public class **WorkmanCorrect**

extends Action

実験システム作業者名リスト作成　クラス

フィールドの概要	
private　CommonValues	**commonValues** 共通変数クラスのインスタンス
private　int	**renban** 連番
private　java.lang.String	**workmanName**

	作業者名

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

WorkmanCorrect(java.sql.Connection dbConn, CommonValues commonVal, javax.servlet.http.HttpServletRequest request) コンストラクタ

メソッドの概要

java.lang.String	**getworkmanName**() 作業者名を返す
java.lang.String	**getworkmanNo**() 作業者番号を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

renban

private int **renban**

連番

---

workmanName

private java.lang.String **workmanName**

作業者名

コンストラクタの詳細

WorkmanCorrect

public **WorkmanCorrect**(java.sql.Connection dbConn,
CommonValues commonVal,
javax.servlet.http.HttpServletRequest request)

コンストラクタ

パラメータ:

dbConn - Connection クラス

commonVal - CommonValues クラス

request - HttpServletRequest クラス

メソッドの詳細

getworkmanNo

public java.lang.String **getworkmanNo**()

作業者番号を返す

戻り値:

作業者番号

---

getworkmanName

public java.lang.String **getworkmanName**()

作業者名を返す

戻り値:

作業者名

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

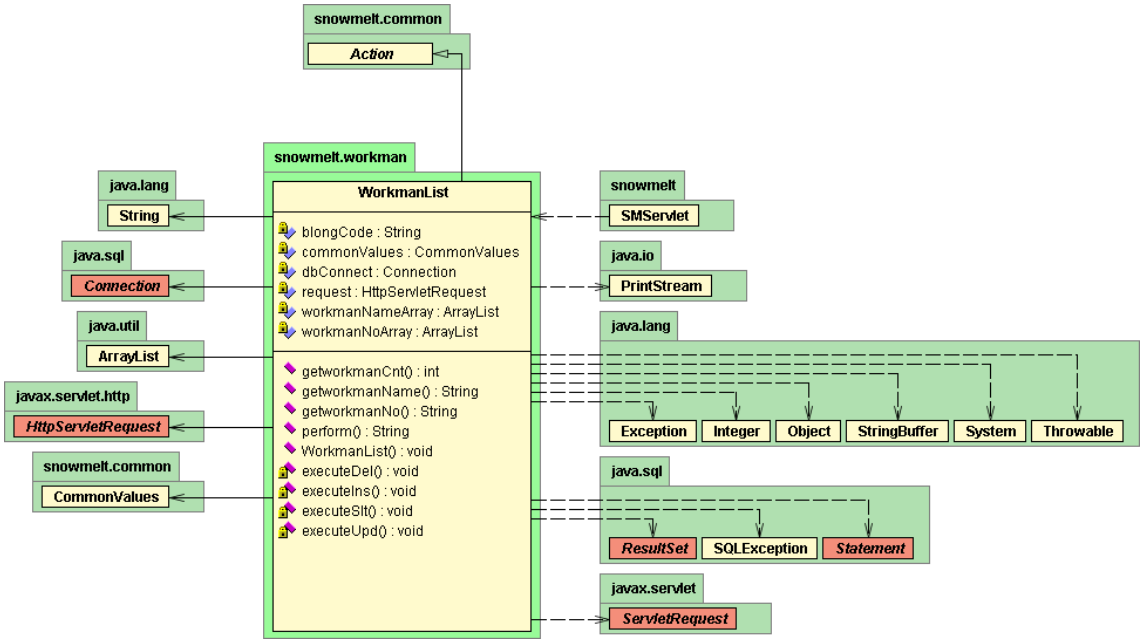
オーバーライド:

クラス Action 内の perform

戻り値:

"/jsp/E02.jsp"

■ 2　クラス WorkmanList



java.lang.Object

|

+--snowmelt.common.Action

|

+--**snowmelt.workman.WorkmanList**

---

public class **WorkmanList**

extends Action

実験システム作業者名リスト作成　クラス

フィールドの概要

private　java.lang.String	**blongCode** 所属コード
private　CommonValues	**commonValues** 共通変数クラスのインスタンス
private　java.sql.Connection	**dbConnect** Connection クラス

private javax.servlet.http.HttpServletRequest	**request** HttpServletRequest クラス
private java.util.ArrayList	**workmanNameArray** 作業者名配列
private java.util.ArrayList	**workmanNoArray** 作業者番号配列

クラス snowmelt.common.Action から継承したフィールド
servlet

コンストラクタの概要

WorkmanList(java.lang.String id, java.sql.Connection dbConn, CommonValues commonVal, javax.servlet.http.HttpServletRequest req)
コンストラクタ

メソッドの概要

private void	**executeDel**() データ削除
private void	**executeIns**() データ追加
private void	**executeSlt**() データ取得
private void	**executeUpd**() データ更新
int	**getworkmanCnt**()

	作業者名数を返す
java.lang.String	**getworkmanName**(int dataIndex) 作業者名を返す
java.lang.String	**getworkmanNo**(int dataIndex) 作業者番号を返す
java.lang.String	**perform**() 遷移先ｊｓｐ名

クラス snowmelt.common.Action から継承したメソッド
charSetJis, getServlet, setServlet, sqlEscape, strEscape

フィールドの詳細

commonValues

private CommonValues **commonValues**

共通変数クラスのインスタンス

---

dbConnect

private java.sql.Connection **dbConnect**

Connection クラス

---

request

private javax.servlet.http.HttpServletRequest **request**

HttpServletRequest クラス

---

blongCode

private java.lang.String **blongCode**

所属コード

---

workmanNoArray

private java.util.ArrayList **workmanNoArray**

作業者番号配列

---

workmanNameArray

private java.util.ArrayList **workmanNameArray**

作業者名配列

コンストラクタの詳細

WorkmanList

public **WorkmanList**(java.lang.String id,
java.sql.Connection dbConn,
CommonValues commonVal,
javax.servlet.http.HttpServletRequest req)

コンストラクタ

パラメータ:

id - アクションＩＤ

dbConn - Connection クラス

commonVal - CommonValues クラス

req - HttpServletRequest クラス

メソッドの詳細

executeIns

private void **executeIns**()

データ追加

---

executeUpd

private void **executeUpd**()

データ更新

---

executeDel

private void **executeDel**()

データ削除

---

executeSlt

private void **executeSlt**()

データ取得

---

getworkmanCnt

public int **getworkmanCnt**()

作業者名数を返す

戻り値:

作業者名数

---

getworkmanNo

public java.lang.String **getworkmanNo**(int dataIndex)

作業者番号を返す

パラメータ:

dataIndex - インデックス

戻り値:

作業者番号

---

getworkmanName

public java.lang.String **getworkmanName**(int dataIndex)

作業者名を返す

パラメータ:

dataIndex - インデックス

戻り値:

作業者名

---

perform

public java.lang.String **perform**()

遷移先ｊｓｐ名

オーバーライド:

クラス Action 内の perform

戻り値:

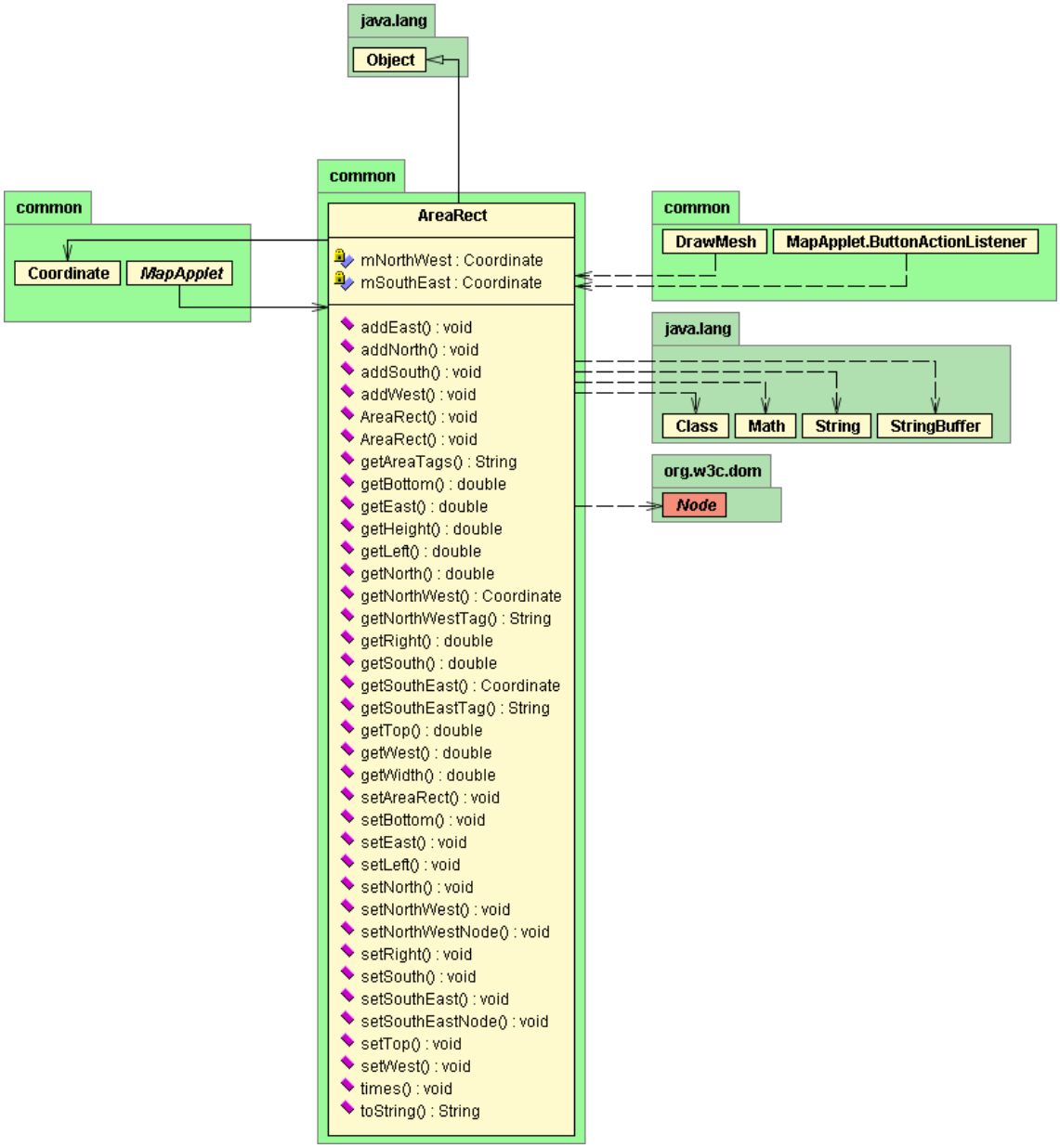
"/jsp/E01.jsp"

---

i） パッケージ common

クラスの概要	
AreaRect	範囲管理クラス
Coordinate	位置情報クラス 緯度経度値を保持
DoublePoint	Double 型 Point クラス
DrawMesh	メッシュ描画クラス
MapApplet	背景地図表示アプレット
MapMesh	共通メッシュ
MeshPolygon	メッシュポリゴンクラス

■ 1　クラス AreaRect



java.lang.Object

　|

　+--**common.AreaRect**

---

public class **AreaRect**

extends java.lang.Object

範囲管理クラス

フィールドの概要

private Coordinate	**mNorthWest** 上・左位置
private Coordinate	**mSouthEast** 下・右位置

コンストラクタの概要

AreaRect() コンストラクタ
AreaRect(AreaRect ar) コンストラクタ

メソッドの概要

void	**addEast**(double addValue) 右座標加算
void	**addNorth**(double addValue) 上座標加算
void	**addSouth**(double addValue) 下座標加算
void	**addWest**(double addValue) 左座標加算
java.lang.String	**getAreaTags**() 各要素を XML 形式の文字列で返す
double	**getBottom**() 下座標取得

double	**getEast**() 右座標取得
double	**getHeight**() 高さ取得
double	**getLeft**() 左座標取得
double	**getNorth**() 上座標取得
Coordinate	**getNorthWest**() 左上座標取得
java.lang.String	**getNorthWestTag**() 右下要素を XML 形式の文字列で返す
double	**getRight**() 右座標取得
double	**getSouth**() 下座標取得
Coordinate	**getSouthEast**() 右下座標取得
java.lang.String	**getSouthEastTag**() 左上要素を XML 形式の文字列で返す
double	**getTop**() 上座標取得
double	**getWest**() 左座標取得
double	**getWidth**() 幅取得
void	**setAreaRect**(AreaRect newValue) 範囲設定
void	**setBottom**(double newBottom) 下座標設定
void	**setEast**(double newEast)

	右座標設定
void	**setLeft**(double newLeft) 左座標設定
void	**setNorth**(double newNorth) 上座標設定
void	**setNorthWest**(Coordinate newValue) 左上座標設定
void	**setNorthWestNode**(org.w3c.dom.Node nwNode) XML 形式の文字列を解釈し左上座標を設定する
void	**setRight**(double newRight) 右座標設定
void	**setSouth**(double newSouth) 下座標設定
void	**setSouthEast**(Coordinate newValue) 右下座標設定
void	**setSouthEastNode**(org.w3c.dom.Node seNode) XML 形式の文字列を解釈し右下座標を設定する
void	**setTop**(double newTop) 上座標設定
void	**setWest**(double newWest) 左座標設定
void	**times**(double scale) mNorthWest の位置はそのままに、縦横の長さを scale 倍する。
java.lang.String	**toString**() 各要素の String 表現を保持している、AreaRect の文字列表現を返します。

フィールドの詳細

mNorthWest

private Coordinate **mNorthWest**

上・左位置

---

mSouthEast

private Coordinate **mSouthEast**

下・右位置

コンストラクタの詳細

AreaRect

public **AreaRect**()

コンストラクタ

AreaRect

public **AreaRect**(AreaRect ar)

コンストラクタ

パラメータ:

ar - 位置情報

メソッドの詳細

getWest

public double **getWest**()

左座標取得

戻り値:

位置情報

---

setWest

public void **setWest**(double newWest)

左座標設定

パラメータ:

newWest - 座標位置

---

getNorth

public double **getNorth**()

上座標取得

戻り値:

位置情報

---

setNorth

public void **setNorth**(double newNorth)

上座標設定

パラメータ:

newNorth - 座標位置

---

getEast

public double **getEast**()

右座標取得

戻り値:

位置情報

---

setEast

public void **setEast**(double newEast)

右座標設定

パラメータ:

newEast - 座標位置

---

getSouth

public double **getSouth**()

下座標取得

戻り値:

位置情報

---

setSouth

public void **setSouth**(double newSouth)

下座標設定

パラメータ:

newSouth - 座標位置

---

addWest

public void **addWest**(double addValue)

左座標加算

パラメータ:

addValue - 加算値

---

addNorth

public void **addNorth**(double addValue)

上座標加算

パラメータ:

addValue - 加算値

---

addEast

public void **addEast**(double addValue)

右座標加算

パラメータ:

addValue - 加算値

---

addSouth

public void **addSouth**(double addValue)

下座標加算

パラメータ:

addValue - 加算値

---

getLeft

public double **getLeft**()

左座標取得

戻り値:

位置情報

---

setLeft

public void **setLeft**(double newLeft)

左座標設定

パラメータ:

newWest - 座標位置

---

getTop

public double **getTop**()

上座標取得

戻り値:

位置情報

---

setTop

public void **setTop**(double newTop)

上座標設定

パラメータ:

newNorth - 座標位置

---

getRight

public double **getRight**()

右座標取得

戻り値:

位置情報

---

setRight

public void **setRight**(double newRight)

右座標設定

パラメータ:

newEast - 座標位置

---

getBottom

public double **getBottom**()

下座標取得

戻り値:

位置情報

---

setBottom

public void **setBottom**(double newBottom)

下座標設定

パラメータ:

newSouth - 座標位置

---

getWidth

public double **getWidth**()

幅取得

戻り値:

幅

---

getHeight

public double **getHeight**()

高さ取得

戻り値:

高さ

---

getNorthWest

public Coordinate **getNorthWest**()

左上座標取得

戻り値:

座標

---

setNorthWest

public void **setNorthWest**(Coordinate newValue)

左上座標設定

パラメータ:

newValue - 左上座標

---

getSouthEast

public Coordinate **getSouthEast**()

右下座標取得

戻り値:

座標

---

setSouthEast

public void **setSouthEast**(Coordinate newValue)

右下座標設定

パラメータ:

newValue - 右下座標

---

setAreaRect

public void **setAreaRect**(AreaRect newValue)

範囲設定

パラメータ:

範囲 -

---

setNorthWestNode

public void **setNorthWestNode**(org.w3c.dom.Node nwNode)

XML 形式の文字列を解釈し左上座標を設定する

パラメータ:

nwNode - ノード

---

setSouthEastNode

public void **setSouthEastNode**(org.w3c.dom.Node seNode)

XML 形式の文字列を解釈し右下座標を設定する

パラメータ:

nwNode - ノード

---

getNorthWestTag

public java.lang.String **getNorthWestTag**()

右下要素を XML 形式の文字列で返す

戻り値:

XML 文字列

---

getSouthEastTag

public java.lang.String **getSouthEastTag**()

左上要素を XML 形式の文字列で返す

戻り値:

XML 文字列

---

getAreaTags

public java.lang.String **getAreaTags**()

各要素を XML 形式の文字列で返す

戻り値:

XML 文字列

---

toString

public java.lang.String **toString**()

各要素の String 表現を保持している、AreaRect の文字列表現を返します。

オーバーライド:

クラス java.lang.Object 内の toString

戻り値:

このコレクションの文字列表現

---

times

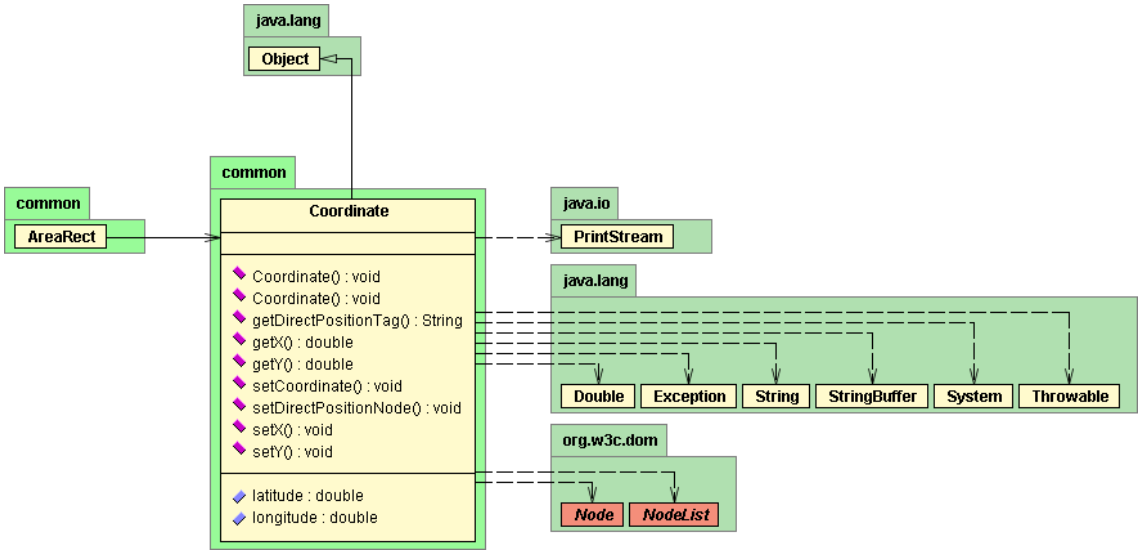
public void **times**(double scale)

mNorthWest の位置はそのままに、縦横の長さを scale 倍する。 結果、mSouthEast の位置が移動する。(日付変更線をまたがる場合は使えない)

パラメータ:

scale - 倍数

■ 2　クラス Coordinate



java.lang.Object

　|

　+--**common.Coordinate**

---

public class **Coordinate**

extends java.lang.Object

位置情報クラス 緯度経度値を保持

フィールドの概要

double	**latitude** 緯度
double	**longitude** 経度

コンストラクタの概要

Coordinate()

コンストラクタ
Coordinate(Coordinate coord) コンストラクタ

メソッドの概要

java.lang.String	**getDirectPositionTag**() 各要素を XML 形式の文字列で返す
double	**getLatitude**() 緯度座標取得
double	**getLongitude**() 経度座標取得
double	**getX**() X 座標取得
double	**getY**() Y 座標取得
void	**setCoordinate**(Coordinate newValue) 座標設定
void	**setDirectPositionNode**(org.w3c.dom.Node dpNode) Node を解釈し緯度経度座標を設定する
void	**setLatitude**(double newY) 緯度座標設定
void	**setLongitude**(double newX) 経度座標設定
void	**setX**(double newX) X 座標設定
void	**setY**(double newY) Y 座標設定

フィールドの詳細

latitude

public double **latitude**

緯度

---

longitude

public double **longitude**

経度

コンストラクタの詳細

Coordinate

public **Coordinate**()

コンストラクタ

Coordinate

public **Coordinate**(Coordinate coord)

コンストラクタ

パラメータ:

coord - 位置

メソッドの詳細

getX

public double **getX**()

X 座標取得

戻り値:

X 座標

---

setX

public void **setX**(double newX)

X 座標設定

パラメータ:

newX - X 座標

---

getY

public double **getY**()

Y 座標取得

戻り値:

Y 座標

---

setY

public void **setY**(double newY)

Y 座標設定

パラメータ:

newY - Y 座標

---

getLongitude

public double **getLongitude**()

経度座標取得

戻り値:

経度座標

---

setLongitude

public void **setLongitude**(double newX)

経度座標設定

パラメータ:

newX - 経度座標

---

getLatitude

public double **getLatitude**()

緯度座標取得

戻り値:

緯度座標

---

setLatitude

public void **setLatitude**(double newY)

緯度座標設定

パラメータ:

newY - 緯度座標

---

setCoordinate

public void **setCoordinate**(Coordinate newValue)

座標設定

パラメータ:

newVlaue - 緯度経度座標

---

getDirectPositionTag

public java.lang.String **getDirectPositionTag**()

各要素を XML 形式の文字列で返す

戻り値:

XML 文字列

---

setDirectPositionNode

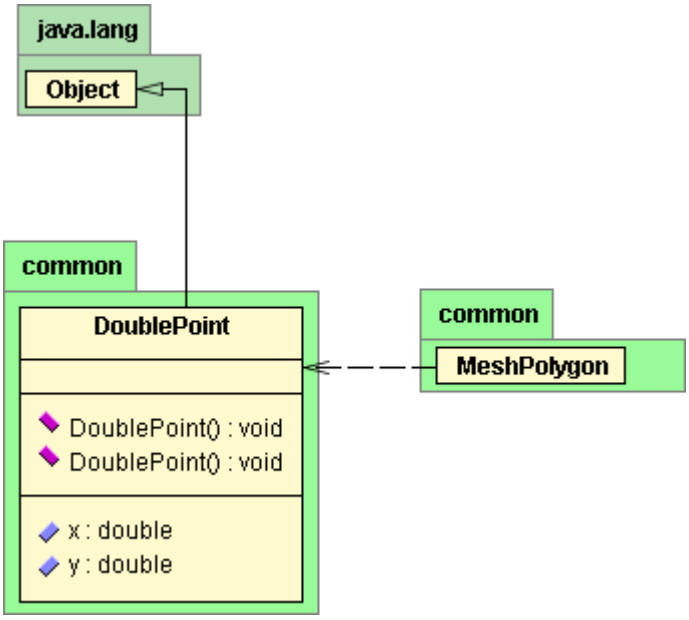
public void **setDirectPositionNode**(org.w3c.dom.Node dpNode)

Node を解釈し緯度経度座標を設定する

パラメータ:

dpNode - ノード

■ 3　クラス DoublePoint



java.lang.Object

　|

　+--**common.DoublePoint**

---

public class **DoublePoint**

extends java.lang.Object

Double 型 Point クラス

フィールドの概要

private　double	**x** x 座標
private　double	**y** y 座標

コンストラクタの概要

DoublePoint() コンストラクタ
DoublePoint(double dx, double dy) コンストラクタ

メソッドの概要

double	**getX**() x 座標の取得
double	**getY**() y 座標の取得
void	**setX**(double dx) x 座標のセット
void	**setY**(double dy) y 座標のセット

フィールドの詳細

x

private double **x**

x 座標

---

y

private double **y**

y 座標

コンストラクタの詳細

DoublePoint

public **DoublePoint**()

コンストラクタ

DoublePoint

public **DoublePoint**(double dx,
double dy)

コンストラクタ

パラメータ:

dx - ｘ座標

dy - ｙ座標

メソッドの詳細

setX

public void **setX**(double dx)

ｘ座標のセット

パラメータ:

dx - ｘ座標

---

setY

public void **setY**(double dy)

ｙ座標のセット

パラメータ:

dy - ｙ座標

---

getX

public double **getX**()

ｘ座標の取得

戻り値:

x 座標

---

getY

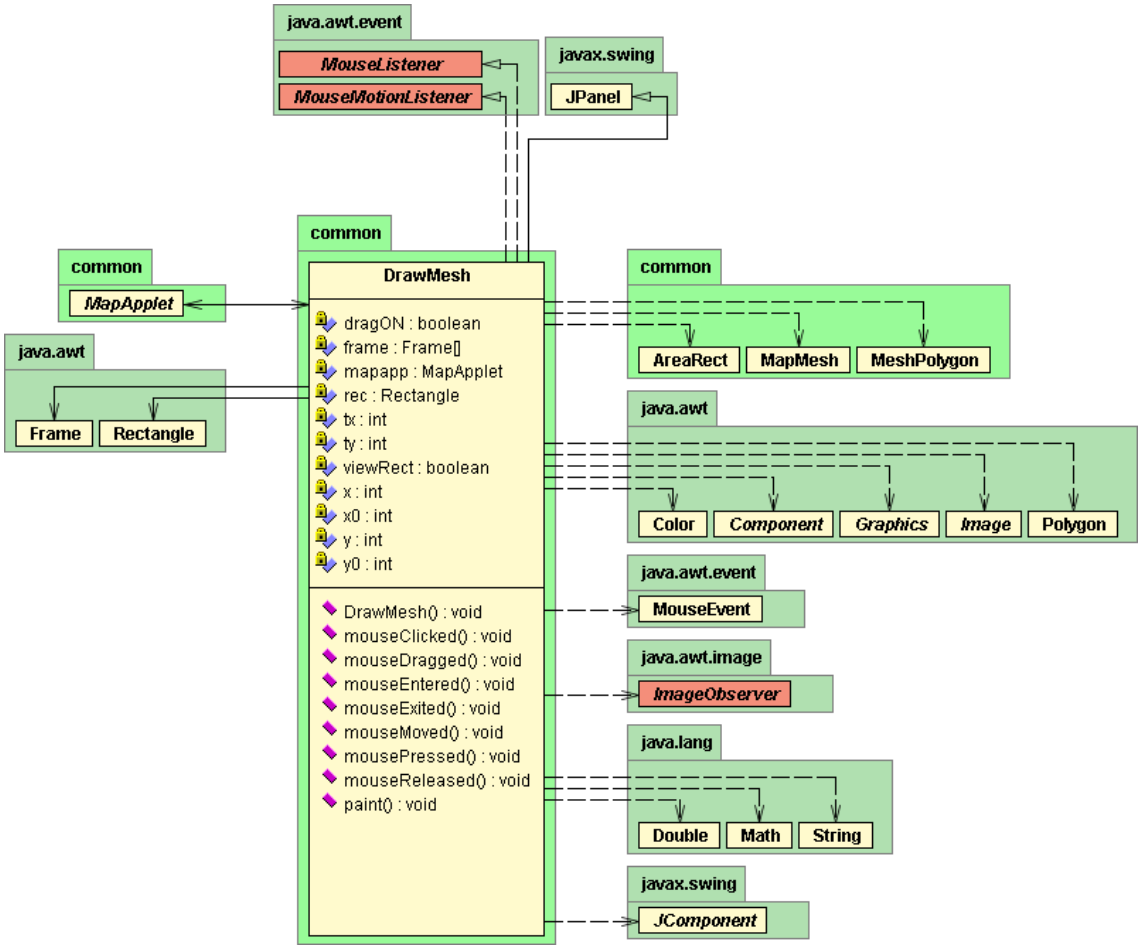
public double **getY**()

y 座標の取得

戻り値:

y 座標

■ 4　クラス DrawMesh




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--javax.swing.JComponent
                    |
                    +--javax.swing.JPanel
                          |
                          +--common.DrawMesh
```

すべての実装インタフェース:

javax.accessibility.Accessible, java.util.EventListener, java.awt.image.ImageObserver, java.awt.MenuContainer, java.awt.event.MouseListener, java.awt.event.MouseMotionListener, java.io.Serializable

public class **DrawMesh**

extends javax.swing.JPanel

implements java.awt.event.MouseListener, java.awt.event.MouseMotionListener

メッシュ描画クラス

フィールドの概要

private boolean	**dragON** マウスクリックかドラッグかの判定フラグ true:ドラッグ時
private java.awt.Frame[]	**frame** ダイアログ用フレームクラス
private MapApplet	**mapapp** MapApplet クラス
private java.awt.Rectangle	**rec** 範囲指定の四角
private int	**tx** マウスドラッグ時の X 方向ドラッグ量
private int	**ty** マウスドラッグ時の Y 方向ドラッグ量
private boolean	**viewRect** 範囲拡大時の選択矩形描画フラグ true:描画
private int	**x** マウスクリック時の X 座標
private int	**x0** 矩形選択時のマウス原点 X 座標
private int	**y** マウスクリック時の Y 座標
private int	**y0** 矩形選択時のマウス原点 Y 座標

コンストラクタの概要

DrawMesh(MapApplet ma) コンストラクタ	

メソッドの概要

void	**mouseClicked**(java.awt.event.MouseEvent e) メッシュをクリックした時の処理
void	**mouseDragged**(java.awt.event.MouseEvent e) マウスをドラッグした時の処理
void	**mouseEntered**(java.awt.event.MouseEvent e) マウスポインタが地図の中に入ってきた時の処理
void	**mouseExited**(java.awt.event.MouseEvent e) マウスポインタが地図の外にでた時の処理
void	**mouseMoved**(java.awt.event.MouseEvent e) マウスを動かした時の処理
void	**mousePressed**(java.awt.event.MouseEvent e) クリックした時の処理
void	**mouseReleased**(java.awt.event.MouseEvent e) マウスを離した時の処理
void	**paint**(java.awt.Graphics g) 散布メッシュの描画

フィールドの詳細

mapapp

private MapApplet **mapapp**

MapApplet クラス

---

x

private int **x**

マウスクリック時の X 座標

---

y

private int **y**

マウスクリック時の Y 座標

---

tx

private int **tx**

マウスドラッグ時の X 方向ドラッグ量

---

ty

private int **ty**

マウスドラッグ時の Y 方向ドラッグ量

---

rec

private java.awt.Rectangle **rec**

範囲指定の四角

---

viewRect

private boolean **viewRect**

範囲拡大時の選択矩形描画フラグ true:描画

---

dragON

private boolean **dragON**

マウスクリックかドラッグかの判定フラグ true:ドラッグ時

---

x0

private int **x0**

矩形選択時のマウス原点 X 座標

---

y0

private int **y0**

矩形選択時のマウス原点 Y 座標

---

frame

private java.awt.Frame[] **frame**

ダイアログ用フレームクラス

コンストラクタの詳細

DrawMesh

public **DrawMesh**(MapApplet ma)

コンストラクタ

パラメータ:

ma - MapApplet クラス

メソッドの詳細

paint

public void **paint**(java.awt.Graphics g)

散布メッシュの描画

オーバーライド:

クラス javax.swing.JComponent 内の paint

パラメータ:

g -

---

mouseClicked

public void **mouseClicked**(java.awt.event.MouseEvent e)

メッシュをクリックした時の処理

定義:

インタフェース java.awt.event.MouseListener 内の mouseClicked

パラメータ:

e -

---

mouseEntered

public void **mouseEntered**(java.awt.event.MouseEvent e)

マウスポインタが地図の中に入ってきた時の処理

定義:

インタフェース java.awt.event.MouseListener 内の mouseEntered

---

mouseExited

public void **mouseExited**(java.awt.event.MouseEvent e)

マウスポインタが地図の外にでた時の処理

定義:

インタフェース java.awt.event.MouseListener 内の mouseExited

---

mousePressed

public void **mousePressed**(java.awt.event.MouseEvent e)

クリックした時の処理

定義:

インタフェース java.awt.event.MouseListener 内の mousePressed

---

mouseReleased

public void **mouseReleased**(java.awt.event.MouseEvent e)

マウスを離した時の処理

定義:

インタフェース java.awt.event.MouseListener 内の mouseReleased

---

mouseDragged

public void **mouseDragged**(java.awt.event.MouseEvent e)

マウスをドラッグした時の処理

定義:

インタフェース java.awt.event.MouseMotionListener 内の mouseDragged

---

mouseMoved

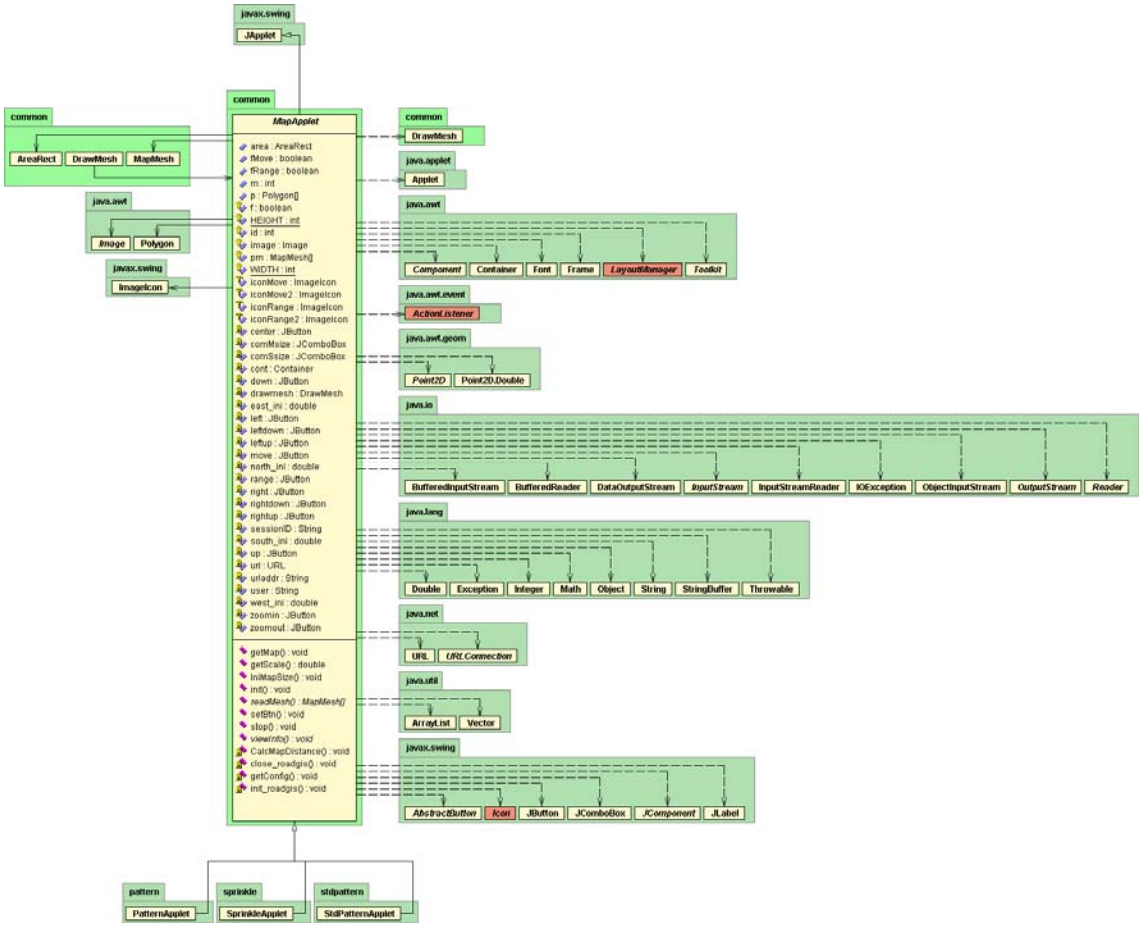
public void **mouseMoved**(java.awt.event.MouseEvent e)

マウスを動かした時の処理

定義:

インタフェース java.awt.event.MouseMotionListener 内の mouseMoved

■ ５　クラス MapApplet



java.lang.Object

  |

  +--java.awt.Component

        |

        +--java.awt.Container

              |

              +--java.awt.Panel

                    |

                    +--java.applet.Applet

                          |

                          +--javax.swing.JApplet

                                |

                                +--**common.MapApplet**

すべての実装インタフェース:

javax.accessibility.Accessible, java.awt.image.ImageObserver,

java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable

直系の既知のサブクラス:

PatternApplet, SprinkleApplet, StdPatternApplet

---

public abstract class **MapApplet**

extends javax.swing.JApplet

背景地図表示アプレット

内部クラスの概要

（パッケージプライベート） class	**MapApplet.ButtonActionListener** ボタンの処理クラス

フィールドの概要

AreaRect	**area** 道路 GIS 画像描画範囲
private javax.swing.JButton	**center** 中心スクロールボタン
private javax.swing.JComboBox	**comMsize** 移動量コンボボックス
private javax.swing.JComboBox	**comSsize** 拡大・縮小率コンボボックス
private java.awt.Container	**cont** コンテントペイン
private javax.swing.JButton	**down** 下スクロールボタン
private DrawMesh	**drawmesh** メッシュ描画クラス

private double	**east_ini** 右側経度初期値
protected boolean	**f** データ取得エラー判定フラグ
boolean	**fMove** 移動ボタン選択フラグ true:ON false:OFF
boolean	**fRange** 範囲指定ボタン選択フラグ true:ON false:OFF
protected static int	**HEIGHT** アプレットの表示高さ
(パッケージプライベート) javax.swing.ImageIcon	**iconMove**
(パッケージプライベート) javax.swing.ImageIcon	**iconMove2**
(パッケージプライベート) javax.swing.ImageIcon	**iconRange** 中間サーブレットへ渡すキー
(パッケージプライベート) javax.swing.ImageIcon	**iconRange2**
protected int	**id** 縮尺 ID
protected java.awt.Image	**image** 地図画像
private javax.swing.JButton	**left** 左スクロールボタン
private javax.swing.JButton	**leftdown** 左下スクロールボタン
private javax.swing.JButton	**leftup** 左上スクロールボタン
int	**m** メッシュ数
private javax.swing.JButton	**move**

	移動ボタン
private double	**north_ini** 上側緯度初期値
java.awt.Polygon[]	**p** メッシュ描画用ポリゴンクラス
protected MapMesh[]	**pm** メッシュクラス
private javax.swing.JButton	**range** 範囲指定ボタン
private javax.swing.JButton	**right** 右スクロールボタン
private javax.swing.JButton	**rightdown** 右下スクロールボタン
private javax.swing.JButton	**rightup** 右上スクロールボタン
private java.lang.String	**sessionID** 道路 GIS サーバー用セッション ID
private double	**south_ini** 下側緯度初期値
private javax.swing.JButton	**up** 上スクロールボタン
private java.net.URL	**url** 中間サーバーURL
private java.lang.String	**urladdr** 道路 GIS 中継サーブレット URL http://hostURL/SnowMelt/servlet/ snowmelt.RgisapiJoinServlet
private java.lang.String	**user** 道路 GIS サーバーユーザー名
private double	**west_ini** 左側経度初期値

protected static int	**WIDTH** アプレットの表示幅
private javax.swing.JButton	**zoomin** 拡大ボタン
private javax.swing.JButton	**zoomout** 縮小ボタン

コンストラクタの概要

MapApplet()

メソッドの概要

private void	**CalcMapDistance**(AreaRect location, java.awt.geom.Point2D.Double dPoint) ヒュベニの距離計算式より表示範囲の大きさを計算
private void	**close_roadgis**() 道路 GIS の終了処理
private void	**getConfig**() 設定ファイル読込
void	**getMap**() 道路 GIS から地図画像を取得し、描画する
double	**getScale**(AreaRect location) 現在の縮尺を取得
void	**IniMapSize**() 初期の表示範囲を画面の縦横比に合わせる
private void	**init_roadgis**() 道路 GIS の初期化処理

void	**init**() アプレットの初期化
abstract MapMesh[]	**readMesh**() メッシュ情報 XML の読込メソッド
void	**setBtn**() ボタンの配置
void	**stop**() stop()メソッド
abstract void	**viewInfo**(java.awt.Frame frame, int k, int j, int gx, int gy) メッシュ情報を表示するウィンドウを生成する

フィールドの詳細

url

private java.net.URL **url**

中間サーバーURL

---

user

private java.lang.String **user**

道路 GIS サーバーユーザー名

---

sessionID

private java.lang.String **sessionID**

道路 GIS サーバー用セッション ID

---

image

protected java.awt.Image **image**

地図画像

---

f

protected boolean f

データ取得エラー判定フラグ

WIDTH

protected static final int WIDTH

アプレットの表示幅

HEIGHT

protected static final int HEIGHT

アプレットの表示高さ

---

leftup

private javax.swing.JButton leftup

左上スクロールボタン

---

up

private javax.swing.JButton up

上スクロールボタン

---

rightup

private javax.swing.JButton rightup

右上スクロールボタン

---

left

private javax.swing.JButton **left**

左スクロールボタン

---

center

private javax.swing.JButton **center**

中心スクロールボタン

---

right

private javax.swing.JButton **right**

右スクロールボタン

---

leftdown

private javax.swing.JButton **leftdown**

左下スクロールボタン

---

down

private javax.swing.JButton **down**

下スクロールボタン

---

rightdown

private javax.swing.JButton **rightdown**

右下スクロールボタン

---

move

private javax.swing.JButton **move**

移動ボタン

---

fMove

public boolean **fMove**

移動ボタン選択フラグ true:ON false:OFF

---

range

private javax.swing.JButton **range**

範囲指定ボタン

---

fRange

public boolean **fRange**

範囲指定ボタン選択フラグ true:ON false:OFF

---

zoomin

private javax.swing.JButton **zoomin**

拡大ボタン

---

zoomout

private javax.swing.JButton **zoomout**

縮小ボタン

---

comSsize

private javax.swing.JComboBox **comSsize**

拡大・縮小率コンボボックス

---

comMsize

private javax.swing.JComboBox **comMsize**

移動量コンボボックス

m

public int **m**

メッシュ数

---

pm

protected MapMesh[] **pm**

メッシュクラス

---

p

public java.awt.Polygon[] **p**

メッシュ描画用ポリゴンクラス

---

area

public AreaRect **area**

道路 GIS 画像描画範囲

---

id

protected int **id**

縮尺 ID

---

cont

private java.awt.Container **cont**

コンテントペイン

---

drawmesh

private DrawMesh **drawmesh**

メッシュ描画クラス

---

west_ini

private double **west_ini**

左側経度初期値

---

east_ini

private double **east_ini**

右側経度初期値

---

north_ini

private double **north_ini**

上側緯度初期値

---

south_ini

private double **south_ini**

下側緯度初期値

---

urladdr

private java.lang.String **urladdr**

道路GIS中継サーブレットURL

http://hostURL/SnowMelt/servlet/snowmelt.RgisapiJoinServlet

---

iconRange

javax.swing.ImageIcon **iconRange**

中間サーブレットへ渡すキー

---

iconRange2

javax.swing.ImageIcon **iconRange2**

---

iconMove

javax.swing.ImageIcon **iconMove**

---

iconMove2

javax.swing.ImageIcon **iconMove2**

コンストラクタの詳細

MapApplet

public **MapApplet**()

メソッドの詳細

init

public void **init**()

アプレットの初期化

オーバーライド:

クラス java.applet.Applet 内の init

---

stop

public void **stop**()

stop()メソッド

オーバーライド:

クラス java.applet.Applet 内の stop

---

getConfig

private void **getConfig**()

設定ファイル読込

---

setBtn

public void **setBtn**()

ボタンの配置

---

init_roadgis

private void **init_roadgis**()

道路 GIS の初期化処理

---

getMap

public void **getMap**()

道路 GIS から地図画像を取得し、描画する

---

close_roadgis

private void **close_roadgis**()

道路 GIS の終了処理

---

getScale

public double **getScale**(AreaRect location)

現在の縮尺を取得

パラメータ:

location - 表示範囲

戻り値:

縮尺

CalcMapDistance

private void **CalcMapDistance**(AreaRect location,
java.awt.geom.Point2D.Double dPoint)

ヒュベニの距離計算式より表示範囲の大きさを計算

パラメータ:

location -

dPoint -

IniMapSize

public void **IniMapSize**()

初期の表示範囲を画面の縦横比に合わせる

---

viewInfo

public abstract void **viewInfo**(java.awt.Frame frame,
int k,
int j,
int gx,
int gy)

メッシュ情報を表示するウィンドウを生成する

パラメータ:

j - メッシュ配列番号

gx - マウスクリック X ポイント

gy - マウスクリック Y ポイント

---

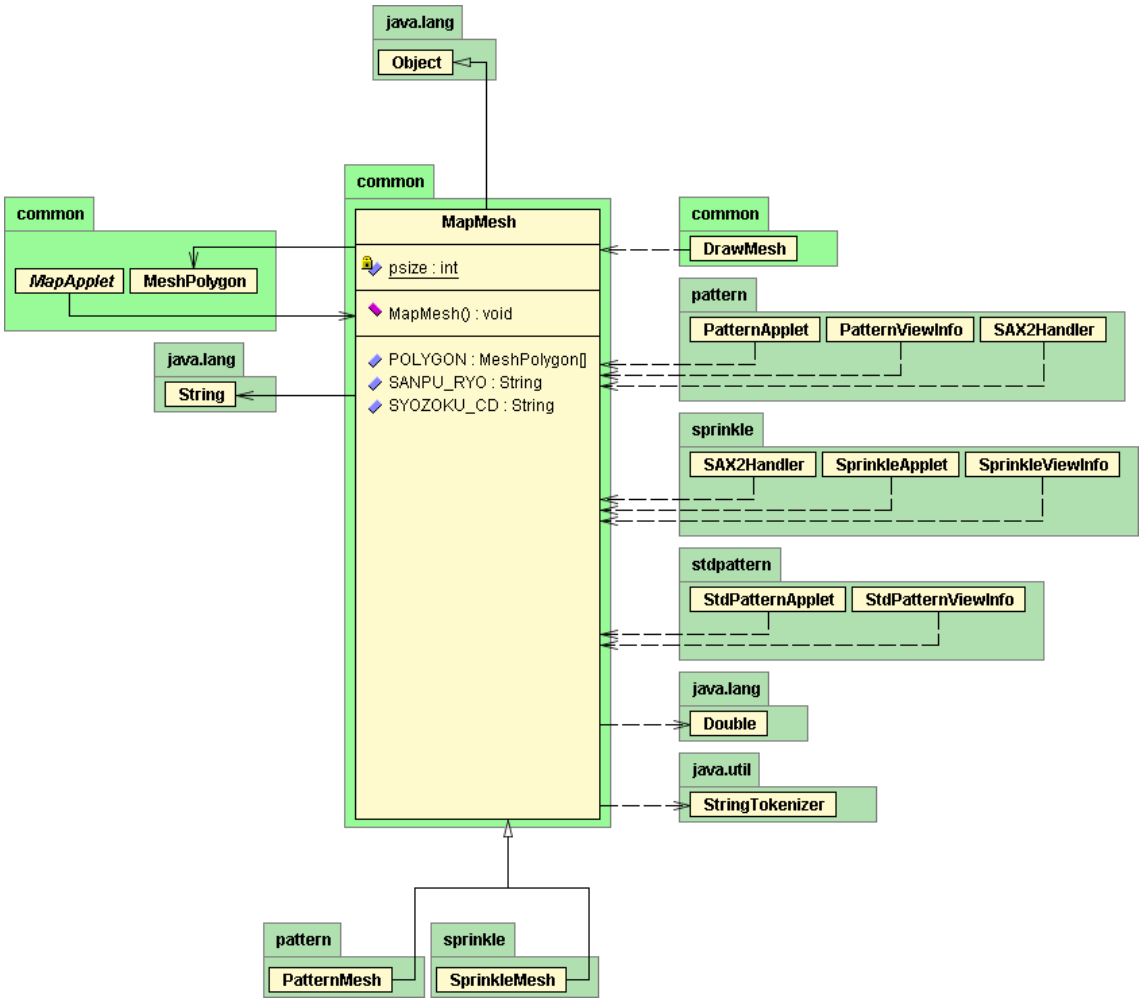
readMesh

public abstract MapMesh[] **readMesh**()

メッシュ情報 XML の読込メソッド

戻り値:

■ ６　クラス MapMesh



java.lang.Object

　|

　+--**common.MapMesh**

直系の既知のサブクラス:

PatternMesh, SprinkleMesh

---

public class **MapMesh**

extends java.lang.Object

共通メッシュ

フィールドの概要

private MeshPolygon[]	**POLYGON** 縮尺ポリゴン
private static int	**psize** 縮尺ポリゴン数
private java.lang.String	**SANPU_RYO** 散布量
private java.lang.String	**SYOZOKU_CD** 所属コード

コンストラクタの概要

MapMesh() コンストラクタ

メソッドの概要

MeshPolygon	**getPOLYGON**(int i) 縮尺ポリゴンの取得
java.lang.String	**getSANPU_RYO**() 散布量の取得
java.lang.String	**getSYOZOKU_CD**() 所属コードの取得
void	**setPOLYGON**(int i, java.lang.String val) 縮尺ポリゴンのセット
void	**setSANPU_RYO**(java.lang.String val) 散布量のセット
void	**setSYOZOKU_CD**(java.lang.String val) 所属コードのセット

フィールドの詳細

psize

private static final int psize

縮尺ポリゴン数

SYOZOKU_CD

private java.lang.String SYOZOKU_CD

所属コード

SANPU_RYO

private java.lang.String SANPU_RYO

散布量

POLYGON

private MeshPolygon[] POLYGON

縮尺ポリゴン

コンストラクタの詳細

MapMesh

public MapMesh()

コンストラクタ

メソッドの詳細

setSYOZOKU_CD

public void **setSYOZOKU_CD**(java.lang.String val)

所属コードのセット

パラメータ:

val - 所属コード

---

setSANPU_RYO

public void **setSANPU_RYO**(java.lang.String val)

散布量のセット

パラメータ:

val - 散布量

---

setPOLYGON

public void **setPOLYGON**(int i,
java.lang.String val)

縮尺ポリゴンのセット

パラメータ:

i - 縮尺 ID

val - 縮尺ポリゴン座標の書かれた文字列

---

getSYOZOKU_CD

public java.lang.String **getSYOZOKU_CD**()

所属コードの取得

戻り値:

所属コード

---

getSANPU_RYO

public java.lang.String **getSANPU_RYO**()

散布量の取得

戻り値:

散布量

---

getPOLYGON

public MeshPolygon **getPOLYGON**(int i)

縮尺ポリゴンの取得

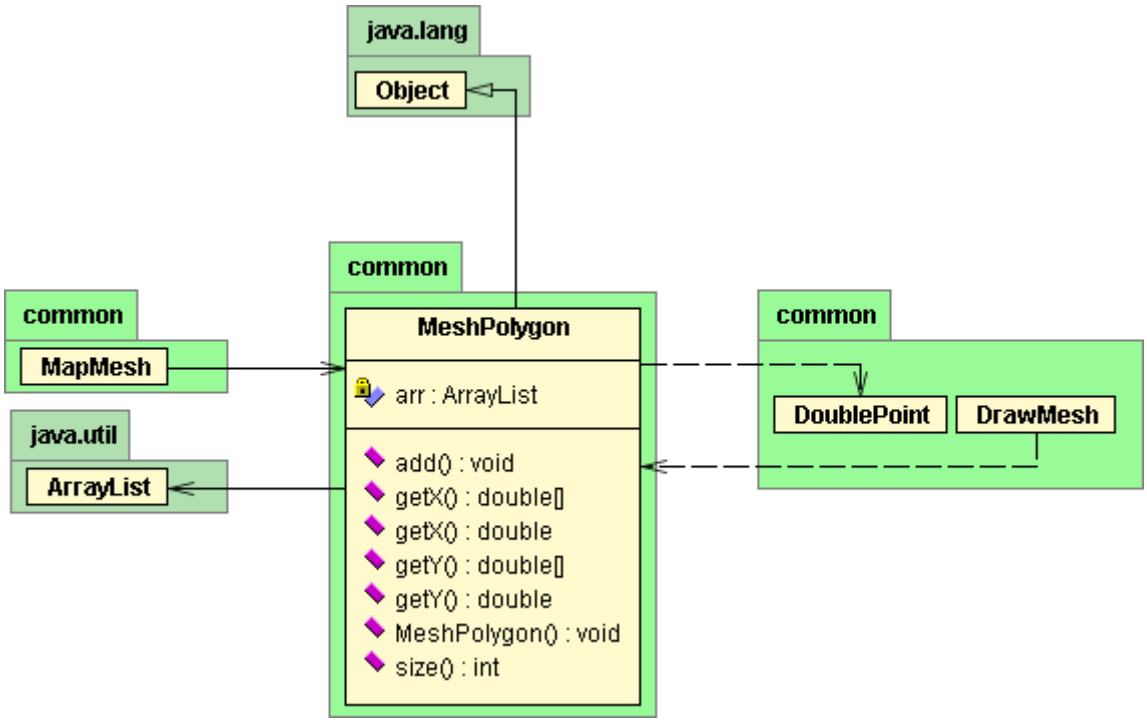
パラメータ:

i - 縮尺 ID

戻り値:

縮尺１ポリゴン

■ 7　クラス MeshPolygon



java.lang.Object

　|

　+--**common.MeshPolygon**

---

public class **MeshPolygon**

extends java.lang.Object

メッシュポリゴンクラス

フィールドの概要

private　java.util.ArrayList	**arr** ポリゴンの点格納用リスト

コンストラクタの概要

MeshPolygon() コンストラクタ

メソッドの概要

void	**add**(double dx, double dy) ポリゴンに点を追加する
double[]	**getX**() ポリゴンのｘ座標配列を返す
double	**getX**(int i) ポリゴンの指定した番号の点のｘ座標を返す
double[]	**getY**() ポリゴンのｙ座標配列を返す
double	**getY**(int i) ポリゴンの指定した番号の点のｙ座標を返す
int	**size**() ポリゴンの節点数を返す

フィールドの詳細

arr

private java.util.ArrayList **arr**

ポリゴンの点格納用リスト

コンストラクタの詳細

MeshPolygon

public **MeshPolygon**()

コンストラクタ

メソッドの詳細

add

public void **add**(double dx,
double dy)

ポリゴンに点を追加する

パラメータ:

dx - 点の X 座標

dy - 点の Y 座標

---

size

public int **size**()

ポリゴンの節点数を返す

戻り値:

ポリゴンの節点数

---

getX

public double[] **getX**()

ポリゴンの x 座標配列を返す

戻り値:

ポリゴンの x 座標配列

---

getY

public double[] **getY**()

ポリゴンの y 座標配列を返す

戻り値:

ポリゴンの y 座標配列

---

getX

public double **getX**(int i)

ポリゴンの指定した番号の点のｘ座標を返す

パラメータ:

i - 指定番号

戻り値:

指定した番号の点のｘ座標

---

getY

public double **getY**(int i)

ポリゴンの指定した番号の点のｙ座標を返す

パラメータ:

i - 指定番号

戻り値:

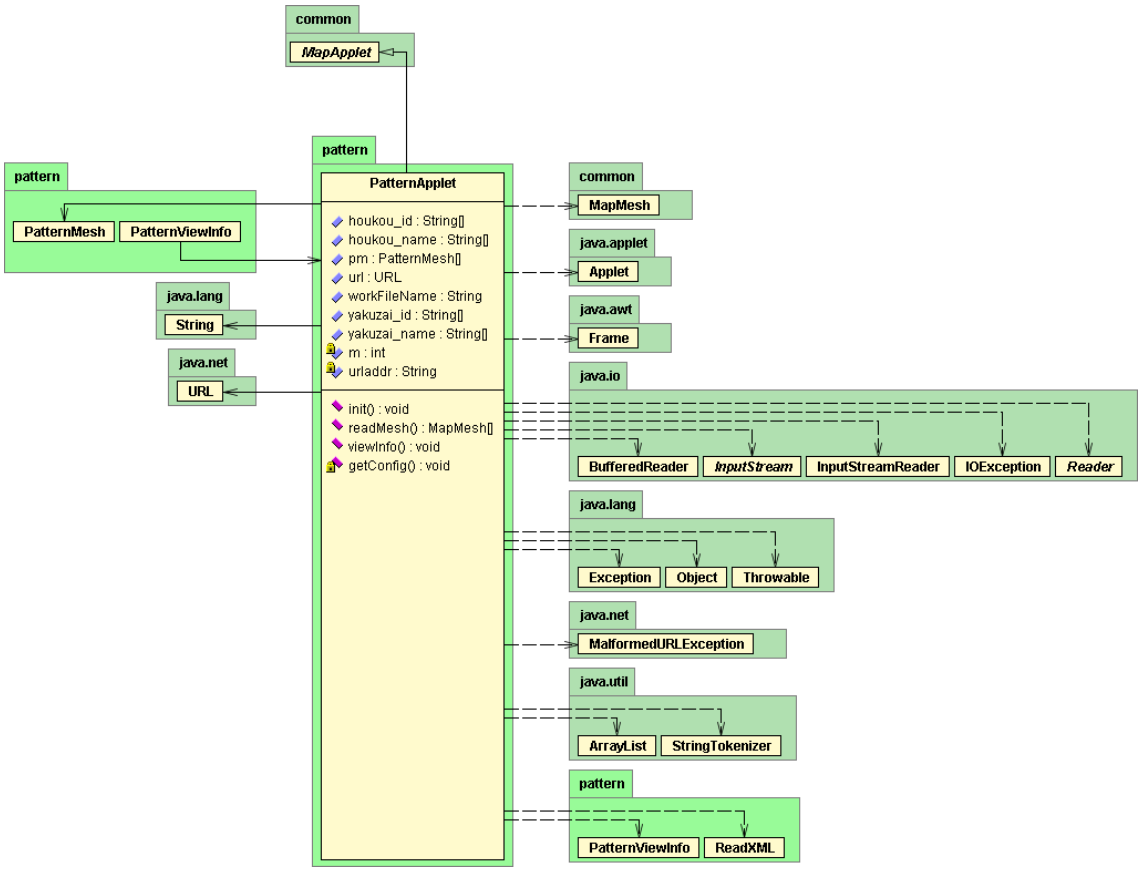
指定した番号の点のｙ座標

---

j） パッケージ pattern

クラスの概要	
PatternApplet	標準パターン確認画面
PatternMesh	標準パターンメッシュ情報
PatternViewInfo	標準パターン作成用メッシュダイアログ
ReadXML	設定ファイル（XML）を読み込むクラス
SAX2Handler	標準パターン表示用 XML 読込クラス

■ １　クラス PatternApplet



java.lang.Object

|

+--java.awt.Component

|

+--java.awt.Container

|

+--java.awt.Panel

|

+--java.applet.Applet

|

+--javax.swing.JApplet

|

+--common.MapApplet

|

+--**pattern.PatternApplet**

すべての実装インタフェース:

javax.accessibility.Accessible, java.awt.image.ImageObserver,

java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable

---

public class **PatternApplet**

extends MapApplet

標準パターン確認画面

クラス common.MapApplet から継承した内部クラス
MapApplet.ButtonActionListener

フィールドの概要

java.lang.String[]	**houkou_id** 散布方向値
java.lang.String[]	**houkou_name** 散布方向表示名
private int	**m** メッシュ数
PatternMesh[]	**pm** メッシュクラス
java.net.URL	**url** データ取得サーブレット
private java.lang.String	**urladdr** データ取得サーブレット URL http://hostURL/SnowMelt/servlet/ snowmelt.Applet2Servlet
java.lang.String	**workFileName** サーブレット側変更用作業ファイル名
java.lang.String[]	**yakuzai_id** 薬剤種類値

java.lang.String[]	**yakuzai_name** 薬剤種類表示名

クラス common.MapApplet から継承したフィールド
area, center, comMsize, comSsize, cont, down, drawmesh, east_ini, f, fMove, fRange, HEIGHT, iconMove, iconMove2, iconRange, iconRange2, id, image, left, leftdown, leftup, move, north_ini, p, range, right, rightdown, rightup, sessionID, south_ini, up, user, west_ini, WIDTH, zoomin, zoomout

コンストラクタの概要

PatternApplet()

メソッドの概要

private void	**getConfig**() 設定ファイル読込
void	**init**() アプレットの初期化
MapMesh[]	**readMesh**() 散布計画 XML ファイル読込 MapApplet クラスで使用
void	**viewInfo**(java.awt.Frame frame, int k, int j, int lx, int ly) 「散布計画」ダイアログの表示

クラス common.MapApplet から継承したメソッド

CalcMapDistance, close_roadgis, getMap, getScale, IniMapSize, init_roadgis, setBtn, stop

フィールドの詳細

m

private int m

メッシュ数

---

pm

public PatternMesh[] pm

メッシュクラス

---

workFileName

public java.lang.String workFileName

サーブレット側変更用作業ファイル名

---

url

public java.net.URL url

データ取得サーブレット

---

urladdr

private java.lang.String urladdr

データ取得サーブレット URL

http://hostURL/SnowMelt/servlet/snowmelt.Applet2Servlet

---

houkou_id

public java.lang.String[] **houkou_id**

散布方向値

---

houkou_name

public java.lang.String[] **houkou_name**

散布方向表示名

---

yakuzai_id

public java.lang.String[] **yakuzai_id**

薬剤種類値

---

yakuzai_name

public java.lang.String[] **yakuzai_name**

薬剤種類表示名

コンストラクタの詳細

PatternApplet

public **PatternApplet**()

メソッドの詳細

init

public void **init**()

アプレットの初期化

オーバーライド:

クラス MapApplet 内の init

---

getConfig

private void **getConfig**()

設定ファイル読込

オーバーライド:

クラス MapApplet 内の getConfig

---

readMesh

public MapMesh[] **readMesh**()

散布計画 XML ファイル読込 MapApplet クラスで使用

オーバーライド:

クラス MapApplet 内の readMesh

戻り値:

MapMesh[] MapMesh クラスの配列

---

viewInfo

public void **viewInfo**(java.awt.Frame frame,
int k,
int j,
int lx,
int ly)

「散布計画」ダイアログの表示

オーバーライド:

クラス MapApplet 内の viewInfo

パラメータ:

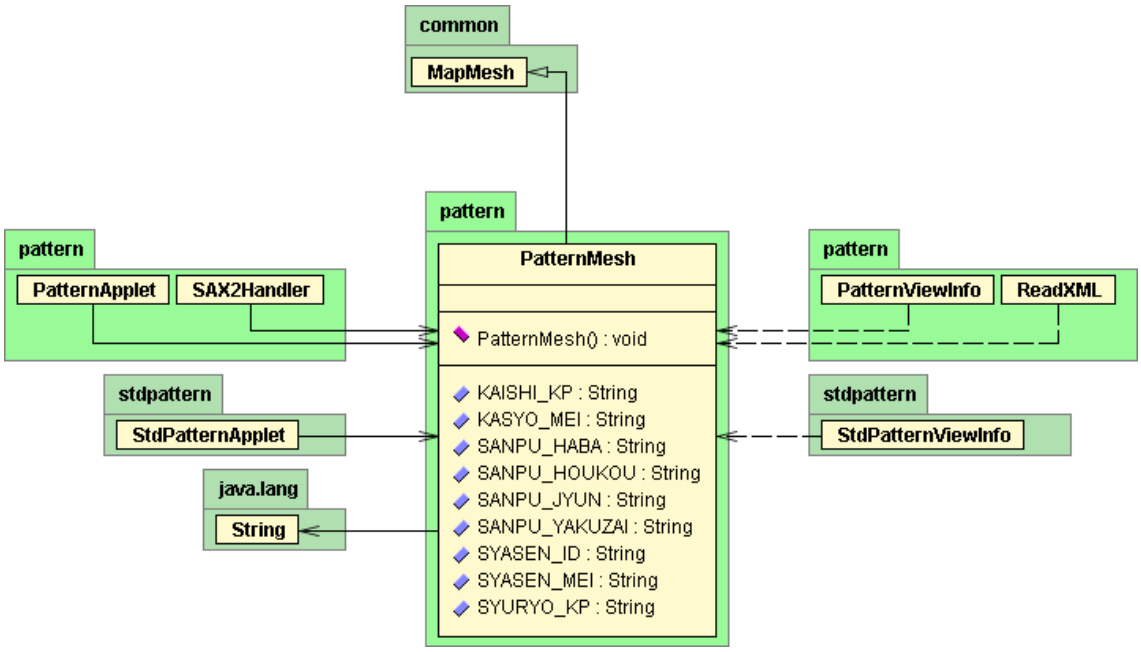
k - カウンタ（重なっている場合の連番）

j - メッシュ番号

lx - X 方向ダイアログ表示位置

ly - Y 方向ダイアログ表示位置

■ 2　クラス PatternMesh



java.lang.Object

　|

　+--common.MapMesh

　　　|

　　　+--**pattern.PatternMesh**

---

public class **PatternMesh**

extends MapMesh

標準パターンメッシュ情報

フィールドの概要	
private　java.lang.String	**KAISHI_KP** 開始 KP
private　java.lang.String	**KASYO_MEI** 箇所名
private　java.lang.String	**SANPU_HABA** 散布幅

private java.lang.String	**SANPU_HOUKOU** 散布方向
private java.lang.String	**SANPU_JYUN** 散布順
private java.lang.String	**SANPU_YAKUZAI** 薬剤種類
private java.lang.String	**SYASEN_ID** 車線 ID
private java.lang.String	**SYASEN_MEI** 車線名
private java.lang.String	**SYURYO_KP** 終了 KP

クラス common.MapMesh から継承したフィールド
POLYGON, psize, SANPU_RYO, SYOZOKU_CD

コンストラクタの概要

PatternMesh() コンストラクタ	

メソッドの概要

java.lang.String	**getKAISHI_KP**() 開始 KP の取得
java.lang.String	**getKASYO_MEI**() 箇所名の取得

java.lang.String	**getSANPU_HABA**() 散布幅の取得
java.lang.String	**getSANPU_HOUKOU**() 散布方向の取得
java.lang.String	**getSANPU_JYUN**() 散布順の取得
java.lang.String	**getSANPU_YAKUZAI**() 薬剤種類の取得
java.lang.String	**getSYASEN_ID**() 車線 ID の取得
java.lang.String	**getSYASEN_MEI**() 車線名の取得
java.lang.String	**getSYURYO_KP**() 終了 KP の取得
void	**setKAISHI_KP**(java.lang.String val) 開始 KP のセット
void	**setKASYO_MEI**(java.lang.String val) 箇所名のセット
void	**setSANPU_HABA**(java.lang.String val) 散布幅のセット
void	**setSANPU_HOUKOU**(java.lang.String val) 散布方向のセット
void	**setSANPU_JYUN**(java.lang.String val) 散布順のセット
void	**setSANPU_YAKUZAI**(java.lang.String val) 薬剤種類のセット
void	**setSYASEN_ID**(java.lang.String val) 車線 ID のセット
void	**setSYASEN_MEI**(java.lang.String val) 車線名のセット
void	**setSYURYO_KP**(java.lang.String val)

	終了 KP のセット

クラス common.MapMesh から継承したメソッド
getPOLYGON, getSANPU_RYO, getSYOZOKU_CD, setPOLYGON, setSANPU_RYO, setSYOZOKU_CD

フィールドの詳細

SYASEN_ID

private java.lang.String **SYASEN_ID**

車線 ID

KAISHI_KP

private java.lang.String **KAISHI_KP**

開始 KP

SYURYO_KP

private java.lang.String **SYURYO_KP**

終了 KP

SYASEN_MEI

private java.lang.String **SYASEN_MEI**

車線名

KASYO_MEI

private java.lang.String **KASYO_MEI**

箇所名

SANPU_HABA

private java.lang.String **SANPU_HABA**

散布幅

SANPU_HOUKOU

private java.lang.String **SANPU_HOUKOU**

散布方向

SANPU_YAKUZAI

private java.lang.String **SANPU_YAKUZAI**

薬剤種類

SANPU_JYUN

private java.lang.String **SANPU_JYUN**

散布順

コンストラクタの詳細

PatternMesh

public **PatternMesh**()

コンストラクタ

メソッドの詳細

setSYASEN_ID

public void **setSYASEN_ID**(java.lang.String val)

車線 ID のセット

パラメータ:

val - 車線名

---

setKAISHI_KP

public void **setKAISHI_KP**(java.lang.String val)

開始 KP のセット

パラメータ:

val - 開始 KP

---

setSYURYO_KP

public void **setSYURYO_KP**(java.lang.String val)

終了 KP のセット

パラメータ:

val - 終了 KP

---

setSYASEN_MEI

public void **setSYASEN_MEI**(java.lang.String val)

車線名のセット

パラメータ:

val - 車線名

---

setKASYO_MEI

public void **setKASYO_MEI**(java.lang.String val)

箇所名のセット

パラメータ:

val - 箇所名

---

setSANPU_HABA

public void **setSANPU_HABA**(java.lang.String val)

散布幅のセット

パラメータ:

val - 散布幅

---

setSANPU_HOUKOU

public void **setSANPU_HOUKOU**(java.lang.String val)

散布方向のセット

パラメータ:

val - 散布方向

---

setSANPU_YAKUZAI

public void **setSANPU_YAKUZAI**(java.lang.String val)

薬剤種類のセット

パラメータ:

val - 薬剤種類

---

setSANPU_JYUN

public void **setSANPU_JYUN**(java.lang.String val)

散布順のセット

パラメータ:

val - 散布順

---

getSYASEN_ID

public java.lang.String **getSYASEN_ID**()

車線 ID の取得

戻り値:

車線 ID

---

getKAISHI_KP

public java.lang.String **getKAISHI_KP**()

開始 KP の取得

戻り値:

開始 KP

---

getSYURYO_KP

public java.lang.String **getSYURYO_KP**()

終了 KP の取得

戻り値:

終了 KP

---

getSYASEN_MEI

public java.lang.String **getSYASEN_MEI**()

車線名の取得

戻り値:

車線名

---

getKASYO_MEI

public java.lang.String **getKASYO_MEI**()

箇所名の取得

戻り値:

箇所名

---

getSANPU_HABA

public java.lang.String **getSANPU_HABA**()

散布幅の取得

戻り値:

散布幅

---

getSANPU_HOUKOU

public java.lang.String **getSANPU_HOUKOU**()

散布方向の取得

戻り値:

散布方向

---

getSANPU_YAKUZAI

public java.lang.String **getSANPU_YAKUZAI**()

薬剤種類の取得

戻り値:

薬剤種類

---

getSANPU_JYUN

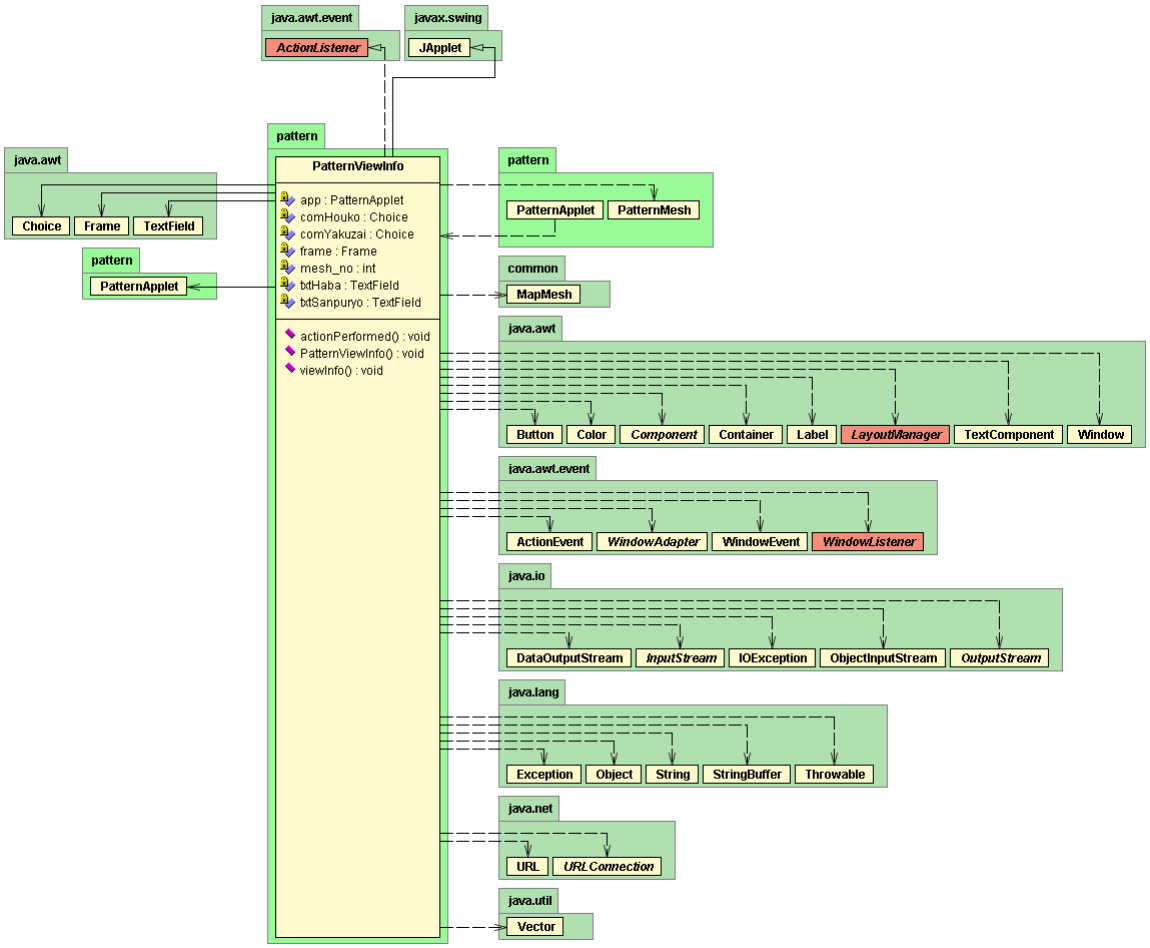
public java.lang.String **getSANPU_JYUN**()

散布順の取得

戻り値:

散布順

■ 3　クラス PatternViewInfo



java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Panel
                    |
                    +--java.applet.Applet
                          |
                          +--javax.swing.JApplet
                                |
                                +--**pattern.PatternViewInfo**

すべての実装インタフェース:

javax.accessibility.Accessible, java.awt.event.ActionListener, java.util.EventListener,

java.awt.image.ImageObserver, java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable

---

public class **PatternViewInfo**

extends javax.swing.JApplet

implements java.awt.event.ActionListener

標準パターン作成用メッシュダイアログ

フィールドの概要

private PatternApplet	**app** 散布計画作成アプレットクラス
private java.awt.Choice	**comHouko** 散布方向コンボボックス
private java.awt.Choice	**comYakuzai** 薬剤種類コンボボックス
private java.awt.Frame	**frame** 「散布計画」ダイアログ用フレーム
private int	**mesh_no** 現在のメッシュ番号（作業用）
private java.awt.TextField	**txtHaba** 散布幅入力ボックス
private java.awt.TextField	**txtSanpuryo** 散布量入力ボックス

コンストラクタの概要

PatternViewInfo(PatternApplet patapp, java.awt.Frame frm, int k, int j, int lx, int ly) コンストラクタ

メソッドの概要

void	**actionPerformed**(java.awt.event.ActionEvent e) 「OK」「キャンセル」ボタン押下時の処理
void	**viewInfo**(int k, int j, int lx, int ly) 「散布計画」ダイアログの表示

フィールドの詳細

txtSanpuryo

private java.awt.TextField **txtSanpuryo**

散布量入力ボックス

---

txtHaba

private java.awt.TextField **txtHaba**

散布幅入力ボックス

---

comHouko

private java.awt.Choice **comHouko**

散布方向コンボボックス

---

comYakuzai

private java.awt.Choice **comYakuzai**

薬剤種類コンボボックス

---

frame

private java.awt.Frame **frame**

「散布計画」ダイアログ用フレーム

---

app

private PatternApplet **app**

散布計画作成アプレットクラス

---

mesh_no

private int **mesh_no**

現在のメッシュ番号（作業用）

コンストラクタの詳細

PatternViewInfo

public **PatternViewInfo**(PatternApplet patapp,
java.awt.Frame frm,
int k,
int j,
int lx,
int ly)

コンストラクタ

パラメータ:

patapp - 散布計画作成アプレットクラス

k - カウンタ（重なっている場合の連番）

j - メッシュ番号

lx - X 方向ダイアログ表示位置

ly - Y 方向ダイアログ表示位置

メソッドの詳細

viewInfo

public void **viewInfo**(int k,
int j,
int lx,
int ly)

「散布計画」ダイアログの表示

パラメータ:

k - カウンタ（重なっている場合の連番）

j - メッシュ番号

lx - X 方向ダイアログ表示位置

ly - Y 方向ダイアログ表示位置

---

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent e)

「OK」「キャンセル」ボタン押下時の処理

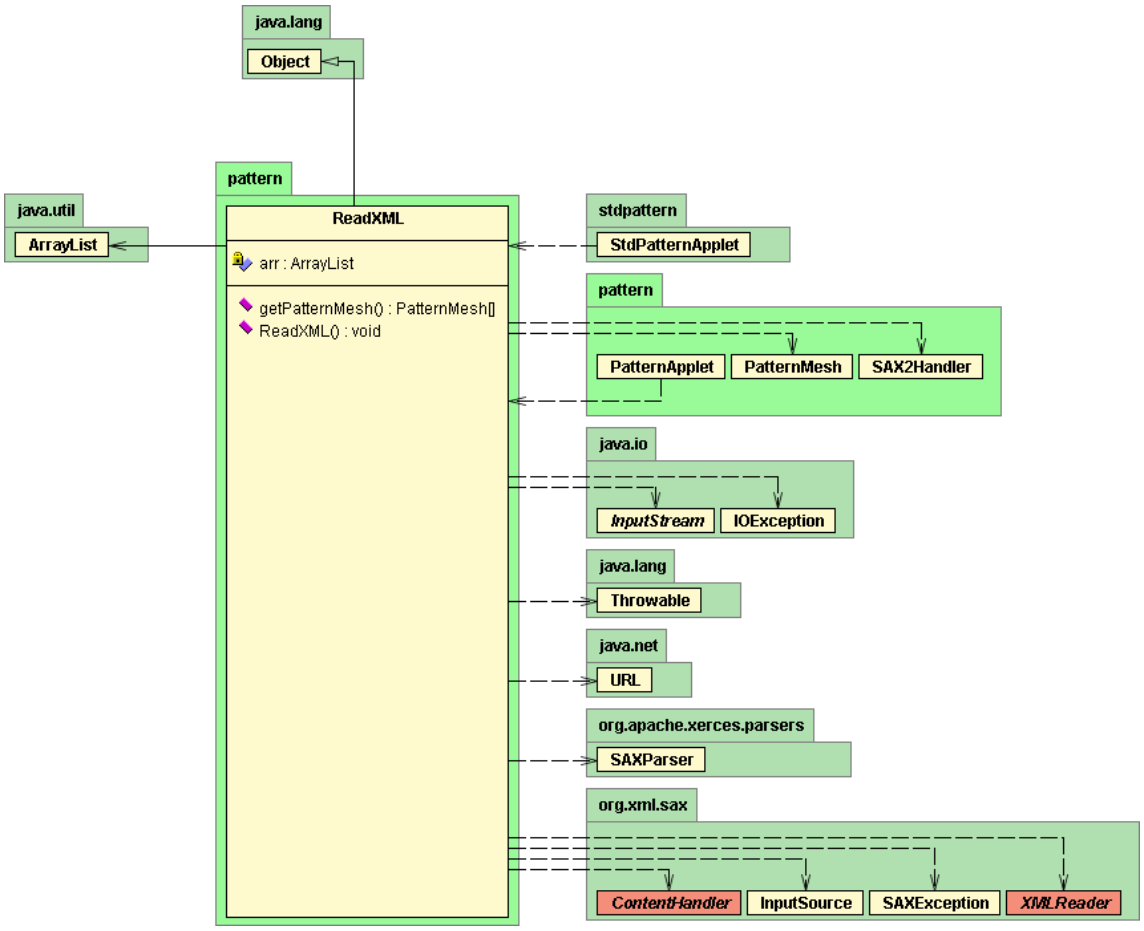
定義:

インタフェース java.awt.event.ActionListener 内の actionPerformed

パラメータ:

e -

■ 4　クラス ReadXML



java.lang.Object

　|

　+--**pattern.ReadXML**

---

public class **ReadXML**

extends java.lang.Object

設定ファイル（XML）を読み込むクラス

フィールドの概要

private　java.util.ArrayList	**arr** メッシュ格納リスト

コンストラクタの概要

ReadXML(java.net.URL xmlFileUrl) コンストラクタ

メソッドの概要

PatternMesh[]	**getPatternMesh**() メッシュクラスの取得

フィールドの詳細

arr

private java.util.ArrayList **arr**

メッシュ格納リスト

コンストラクタの詳細

ReadXML

public **ReadXML**(java.net.URL xmlFileUrl)

コンストラクタ

パラメータ:

xmlFileUrl - XML ファイル名

メソッドの詳細

getPatternMesh

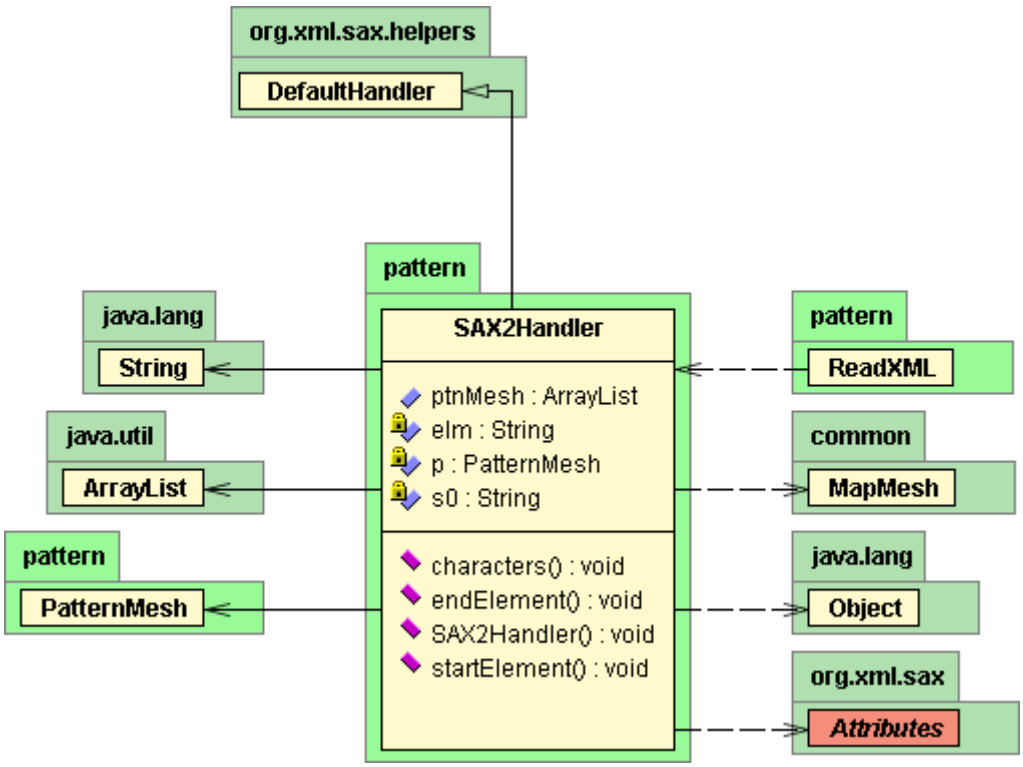
public PatternMesh[] **getPatternMesh**()

メッシュクラスの取得

戻り値:

メッシュクラス

■ ５　クラス SAX2Handler



java.lang.Object

  |

  +--org.xml.sax.helpers.DefaultHandler

        |

        +--**pattern.SAX2Handler**

すべての実装インタフェース:

org.xml.sax.ContentHandler, org.xml.sax.DTDHandler, org.xml.sax.EntityResolver, org.xml.sax.ErrorHandler

---

public class **SAX2Handler**

extends org.xml.sax.helpers.DefaultHandler

標準パターン表示用 XML 読込クラス

フィールドの概要

private java.lang.String	**elm**

	要素名格納作業用文字
private PatternMesh	**p** 散布パターンメッシュクラス
java.util.ArrayList	**ptnMesh** 散布パターンメッシュ格納リスト
private java.lang.String	**s0** SAX2 文字バッファー対応

コンストラクタの概要

SAX2Handler() コンストラクタ

メソッドの概要

void	**characters**(char[] chars, int start, int length) 要素の値の取り出し
void	**endElement**(java.lang.String uri, java.lang.String local, java.lang.String raw) 終了要素の取り出し
void	**startElement**(java.lang.String uri, java.lang.String local, java.lang.String raw, org.xml.sax.Attributes attrs) 開始要素の取り出し

フィールドの詳細

ptnMesh

public java.util.ArrayList **ptnMesh**

散布パターンメッシュ格納リスト

---

p

private PatternMesh **p**

散布パターンメッシュクラス

---

elm

private java.lang.String **elm**

要素名格納作業用文字

---

s0

private java.lang.String **s0**

SAX2 文字バッファー対応

コンストラクタの詳細

SAX2Handler

public **SAX2Handler**()

コンストラクタ

メソッドの詳細

startElement

public void **startElement**(java.lang.String uri,
java.lang.String local,
java.lang.String raw,
org.xml.sax.Attributes attrs)

開始要素の取り出し

オーバーライド:

クラス org.xml.sax.helpers.DefaultHandler 内の startElement

---

endElement

```
public void endElement(java.lang.String uri,
                       java.lang.String local,
                       java.lang.String raw)
```

終了要素の取り出し

オーバーライド:

クラス org.xml.sax.helpers.DefaultHandler 内の endElement

パラメータ:

uri -

local -

raw -

---

characters

```
public void characters(char[] chars,
                       int start,
                       int length)
```

要素の値の取り出し

オーバーライド:

クラス org.xml.sax.helpers.DefaultHandler 内の characters

パラメータ:

chars -

start -

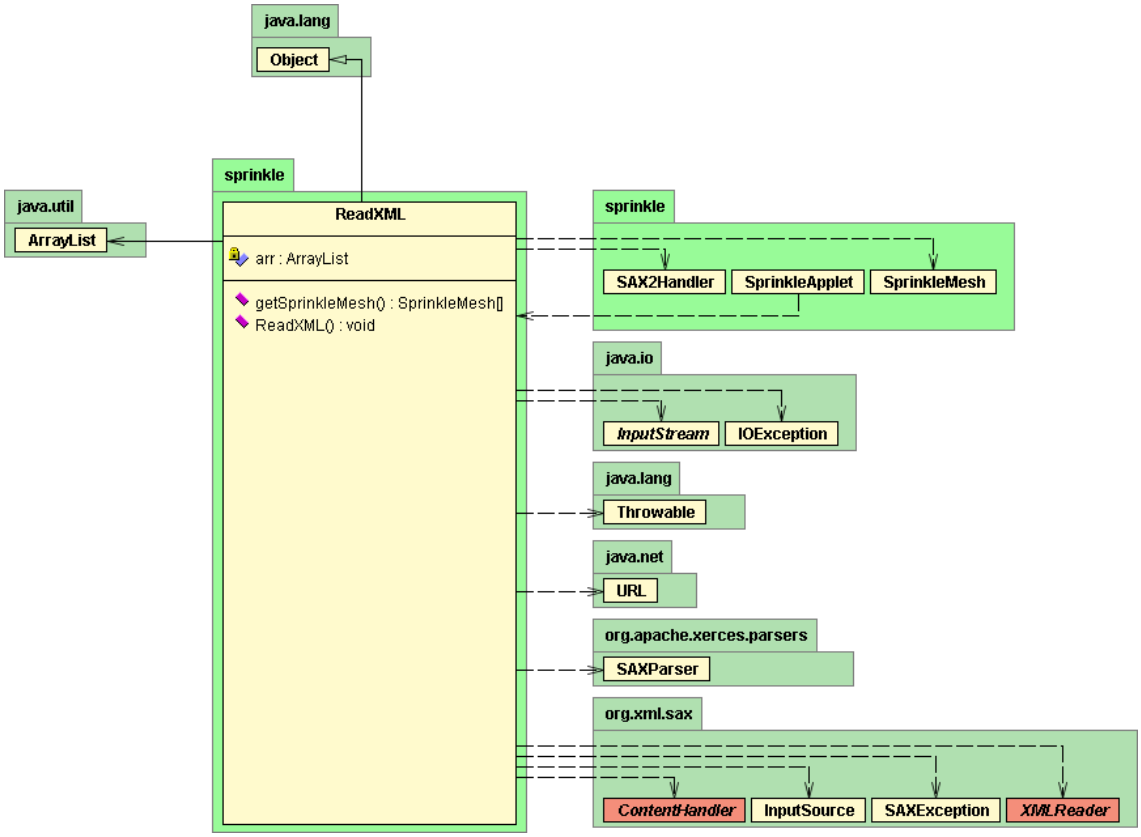
length -

---

k） パッケージ sprinkle

クラスの概要	
ReadXML	設定ファイル（XML）を読み込むクラス
SAX2Handler	散布履歴表示用 XML 読込クラス
SprinkleApplet	散布履歴表示画面
SprinkleMesh	標準パターンメッシュ情報
SprinkleViewInfo	散布履歴確認用メッシュダイアログ

■ 1　クラス ReadXML



java.lang.Object

　|

　+--**sprinkle.ReadXML**

---

public class **ReadXML**

extends java.lang.Object

設定ファイル（XML）を読み込むクラス

フィールドの概要

private　java.util.ArrayList	**arr** メッシュ格納リスト

コンストラクタの概要

ReadXML(java.net.URL xmlFileUrl) コンストラクタ

メソッドの概要

SprinkleMesh[]	**getSprinkleMesh**() メッシュクラスの取得

フィールドの詳細

arr

private java.util.ArrayList **arr**

メッシュ格納リスト

コンストラクタの詳細

ReadXML

public **ReadXML**(java.net.URL xmlFileUrl)

コンストラクタ

パラメータ:

xmlFileUrl - XML ファイル名

メソッドの詳細

getSprinkleMesh

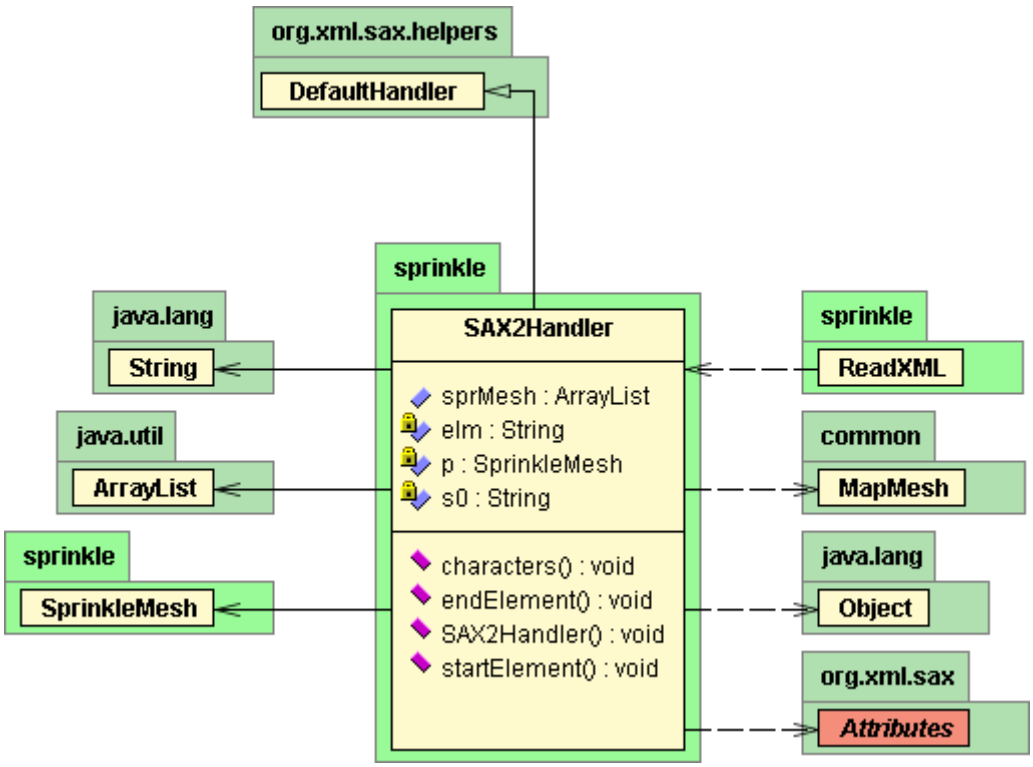
public SprinkleMesh[] **getSprinkleMesh**()

メッシュクラスの取得

戻り値:

メッシュクラス

■ 2　クラス SAX2Handler



java.lang.Object

　|

　+--org.xml.sax.helpers.DefaultHandler

　　|

　　+--**sprinkle.SAX2Handler**

すべての実装インタフェース:

　　org.xml.sax.ContentHandler,　org.xml.sax.DTDHandler,　org.xml.sax.EntityResolver,

　　org.xml.sax.ErrorHandler

---

public class **SAX2Handler**

extends org.xml.sax.helpers.DefaultHandler

散布履歴表示用 XML 読込クラス

フィールドの概要

private　java.lang.String	**elm**

	要素名格納作業用文字
private SprinkleMesh	**p** 散布履歴メッシュクラス
private java.lang.String	**s0** SAX2 文字バッファー対応
java.util.ArrayList	**sprMesh** 散布履歴メッシュ格納リスト

コンストラクタの概要

SAX2Handler() コンストラクタ

メソッドの概要

void	**characters**(char[] chars, int start, int length) 要素の値の取り出し
void	**endElement**(java.lang.String uri, java.lang.String local, java.lang.String raw) 終了要素の取り出し
void	**startElement**(java.lang.String uri, java.lang.String local, java.lang.String raw, org.xml.sax.Attributes attrs) 開始要素の取り出し

フィールドの詳細

sprMesh

public java.util.ArrayList **sprMesh**

散布履歴メッシュ格納リスト

---

p

private SprinkleMesh **p**

散布履歴メッシュクラス

---

elm

private java.lang.String **elm**

要素名格納作業用文字

---

s0

private java.lang.String **s0**

SAX2 文字バッファー対応

コンストラクタの詳細

SAX2Handler

public **SAX2Handler**()

コンストラクタ

メソッドの詳細

startElement

public void **startElement**(java.lang.String uri,
java.lang.String local,
java.lang.String raw,
org.xml.sax.Attributes attrs)

開始要素の取り出し

オーバーライド:

クラス org.xml.sax.helpers.DefaultHandler 内の startElement

---

endElement

```
public void endElement(java.lang.String uri,
                       java.lang.String local,
                       java.lang.String raw)
```

終了要素の取り出し

オーバーライド:

クラス org.xml.sax.helpers.DefaultHandler 内の endElement

パラメータ:

uri -

local -

raw -

---

characters

```
public void characters(char[] chars,
                       int start,
                       int length)
```

要素の値の取り出し

オーバーライド:

クラス org.xml.sax.helpers.DefaultHandler 内の characters

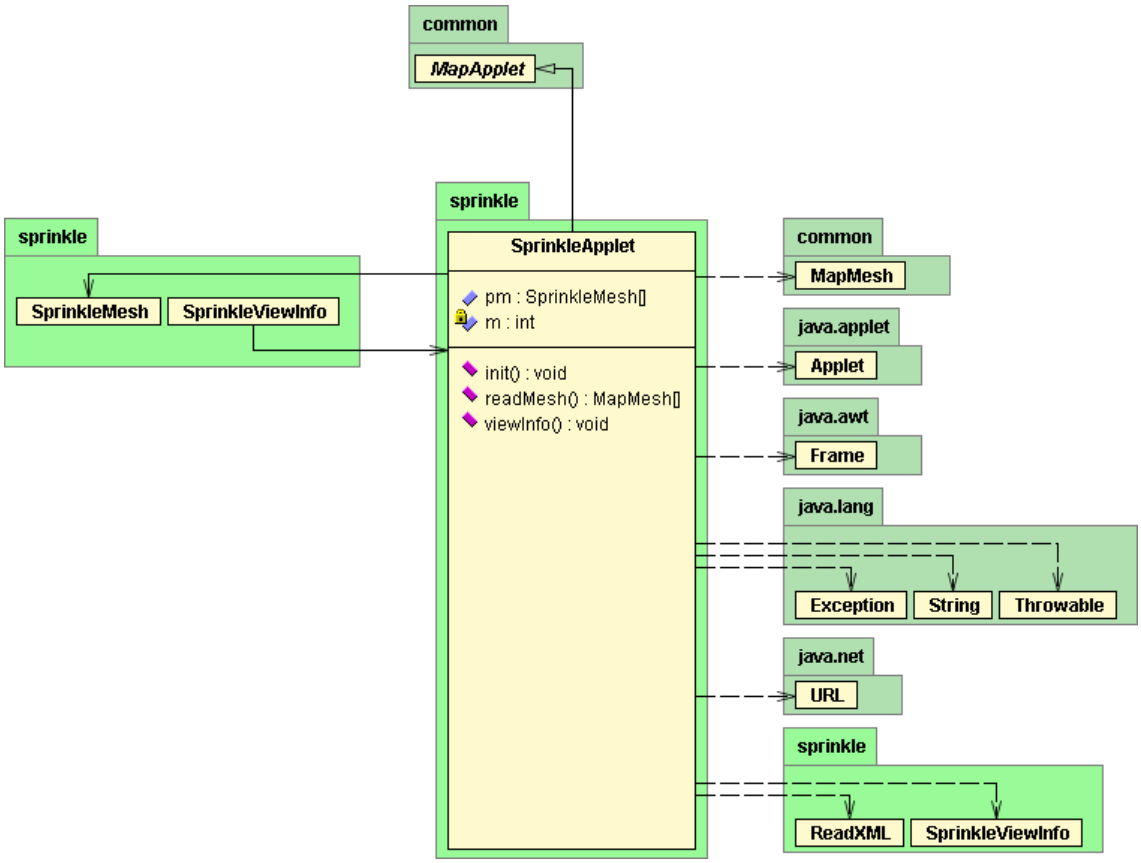
パラメータ:

chars -

start -

length -

■ 3　クラス SprinkleApplet



java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Panel
                    |
                    +--java.applet.Applet
                          |
                          +--javax.swing.JApplet
                                |
                                +--common.MapApplet
                                      |
                                      +--**sprinkle.SprinkleApplet**

すべての実装インタフェース:

javax.accessibility.Accessible, java.awt.image.ImageObserver,

java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable

---

public class **SprinkleApplet**

extends MapApplet

散布履歴表示画面

クラス common.MapApplet から継承した内部クラス
MapApplet.ButtonActionListener

フィールドの概要

private int	**m** メッシュ数
SprinkleMesh[]	**pm** メッシュクラス

クラス common.MapApplet から継承したフィールド
area, center, comMsize, comSsize, cont, down, drawmesh, east_ini, f, fMove, fRange, HEIGHT, iconMove, iconMove2, iconRange, iconRange2, id, image, left, leftdown, leftup, move, north_ini, p, range, right, rightdown, rightup, sessionID, south_ini, up, url, urladdr, user, west_ini, WIDTH, zoomin, zoomout

コンストラクタの概要

SprinkleApplet()

メソッドの概要

void	**init**() アプレットの初期化
MapMesh[]	**readMesh**() 散布計画 XML ファイル読込 MapApplet クラスで使用
void	**viewInfo**(java.awt.Frame frame, int k, int j, int lx, int ly) 「散布メッシュ情報」ダイアログの表示

クラス common.MapApplet から継承したメソッド
CalcMapDistance, close_roadgis, getConfig, getMap, getScale, IniMapSize, init_roadgis, setBtn, stop

フィールドの詳細

m

private int **m**

メッシュ数

---

pm

public SprinkleMesh[] **pm**

メッシュクラス

コンストラクタの詳細

SprinkleApplet

public **SprinkleApplet**()

メソッドの詳細

init

public void **init**()

アプレットの初期化

オーバーライド:

クラス MapApplet 内の init

---

readMesh

public MapMesh[] **readMesh**()

散布計画 XML ファイル読込 MapApplet クラスで使用

オーバーライド:

クラス MapApplet 内の readMesh

戻り値:

MapMesh[] MapMesh クラスの配列

---

viewInfo

public void **viewInfo**(java.awt.Frame frame,
int k,
int j,
int lx,
int ly)

「散布メッシュ情報」ダイアログの表示

オーバーライド:

クラス MapApplet 内の viewInfo

パラメータ:

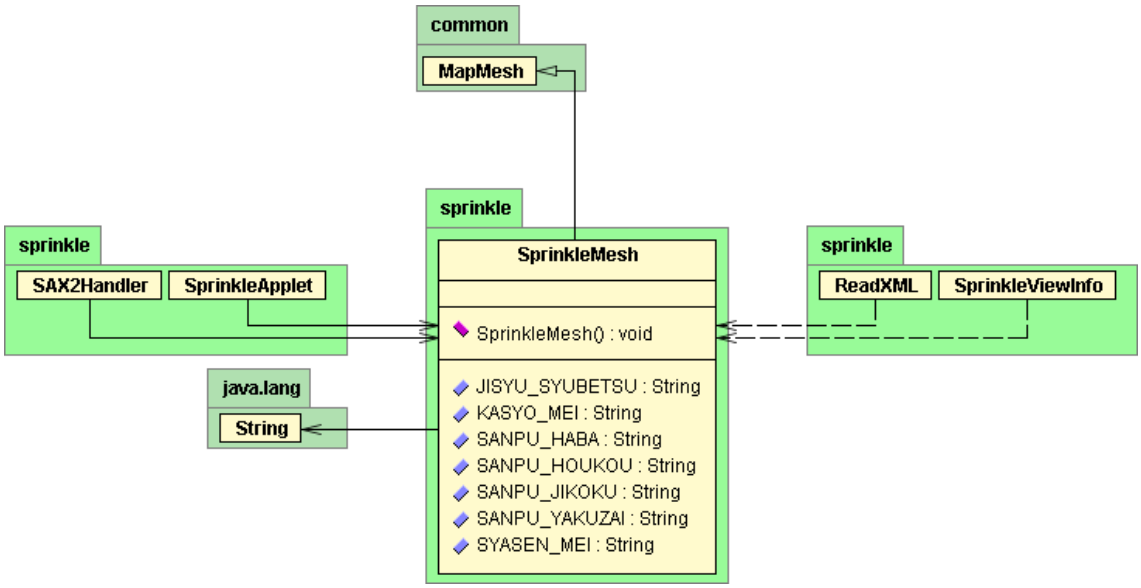
k - カウンタ（重なっている場合の連番）

j - メッシュ番号

lx - X 方向ダイアログ表示位置

ly - Y 方向ダイアログ表示位置

■ 4　クラス SprinkleMesh



java.lang.Object

　|

　+--common.MapMesh

　　　|

　　　+--**sprinkle.SprinkleMesh**

---

public class **SprinkleMesh**

extends MapMesh

標準パターンメッシュ情報

フィールドの概要	
private　java.lang.String	**JISYU_SYUBETSU** 自動・手動種別
private　java.lang.String	**KASYO_MEI** 箇所名
private　java.lang.String	**SANPU_HABA** 散布幅
private　java.lang.String	**SANPU_HOUKOU**

	散布方向
private java.lang.String	**SANPU_JIKOKU** 散布時刻
private java.lang.String	**SANPU_YAKUZAI** 薬剤種類
private java.lang.String	**SYASEN_MEI** 車線名

クラス common.MapMesh から継承したフィールド
POLYGON, psize, SANPU_RYO, SYOZOKU_CD

コンストラクタの概要

SprinkleMesh() コンストラクタ

メソッドの概要

java.lang.String	**getJISYU_SYUBETSU**() 自動・手動種別の取得
java.lang.String	**getKASYO_MEI**() 箇所名の取得
java.lang.String	**getSANPU_HABA**() 散布幅の取得
java.lang.String	**getSANPU_HOUKOU**() 散布方向の取得
java.lang.String	**getSANPU_JIKOKU**()

	散布時刻の取得
java.lang.String	**getSANPU_YAKUZAI**() 薬剤種類の取得
java.lang.String	**getSYASEN_MEI**() 車線名の取得
void	**setJISYU_SYUBETSU**(java.lang.String val) 自動・手動種別のセット
void	**setKASYO_MEI**(java.lang.String val) 箇所名のセット
void	**setSANPU_HABA**(java.lang.String val) 散布幅のセット
void	**setSANPU_HOUKOU**(java.lang.String val) 散布方向のセット
void	**setSANPU_JIKOKU**(java.lang.String val) 散布時刻のセット
void	**setSANPU_YAKUZAI**(java.lang.String val) 薬剤種類のセット
void	**setSYASEN_MEI**(java.lang.String val) 車線名のセット

クラス common.MapMesh から継承したメソッド
getPOLYGON, getSANPU_RYO, getSYOZOKU_CD, setPOLYGON, setSANPU_RYO, setSYOZOKU_CD

フィールドの詳細

SYASEN_MEI

private java.lang.String **SYASEN_MEI**

車線名

KASYO_MEI

private java.lang.String **KASYO_MEI**

箇所名

SANPU_JIKOKU

private java.lang.String **SANPU_JIKOKU**

散布時刻

SANPU_HABA

private java.lang.String **SANPU_HABA**

散布幅

SANPU_HOUKOU

private java.lang.String **SANPU_HOUKOU**

散布方向

SANPU_YAKUZAI

private java.lang.String **SANPU_YAKUZAI**

薬剤種類

JISYU_SYUBETSU

private java.lang.String **JISYU_SYUBETSU**

自動・手動種別

コンストラクタの詳細

SprinkleMesh

public **SprinkleMesh**()

コンストラクタ

メソッドの詳細

setSYASEN_MEI

public void **setSYASEN_MEI**(java.lang.String val)

車線名のセット

パラメータ:

val - 車線名

---

setKASYO_MEI

public void **setKASYO_MEI**(java.lang.String val)

箇所名のセット

パラメータ:

val - 箇所名

---

setSANPU_JIKOKU

public void **setSANPU_JIKOKU**(java.lang.String val)

散布時刻のセット

パラメータ:

val - 散布時刻

---

setSANPU_HABA

public void **setSANPU_HABA**(java.lang.String val)

散布幅のセット

パラメータ:

val - 散布幅

---

setSANPU_HOUKOU

public void **setSANPU_HOUKOU**(java.lang.String val)

散布方向のセット

パラメータ:

val - 散布方向

---

setSANPU_YAKUZAI

public void **setSANPU_YAKUZAI**(java.lang.String val)

薬剤種類のセット

パラメータ:

val - 薬剤種類

---

setJISYU_SYUBETSU

public void **setJISYU_SYUBETSU**(java.lang.String val)

自動・手動種別のセット

パラメータ:

val - 自動・手動種別

---

getSYASEN_MEI

public java.lang.String **getSYASEN_MEI**()

車線名の取得

戻り値:

車線名

---

getKASYO_MEI

public java.lang.String **getKASYO_MEI**()

箇所名の取得

戻り値:

箇所名

---

getSANPU_JIKOKU

public java.lang.String **getSANPU_JIKOKU**()

散布時刻の取得

戻り値:

散布時刻

---

getSANPU_HABA

public java.lang.String **getSANPU_HABA**()

散布幅の取得

戻り値:

散布幅

---

getSANPU_HOUKOU

public java.lang.String **getSANPU_HOUKOU**()

散布方向の取得

戻り値:

散布方向

---

getSANPU_YAKUZAI

public java.lang.String **getSANPU_YAKUZAI**()

薬剤種類の取得

戻り値:

薬剤種類

---

getJISYU_SYUBETSU

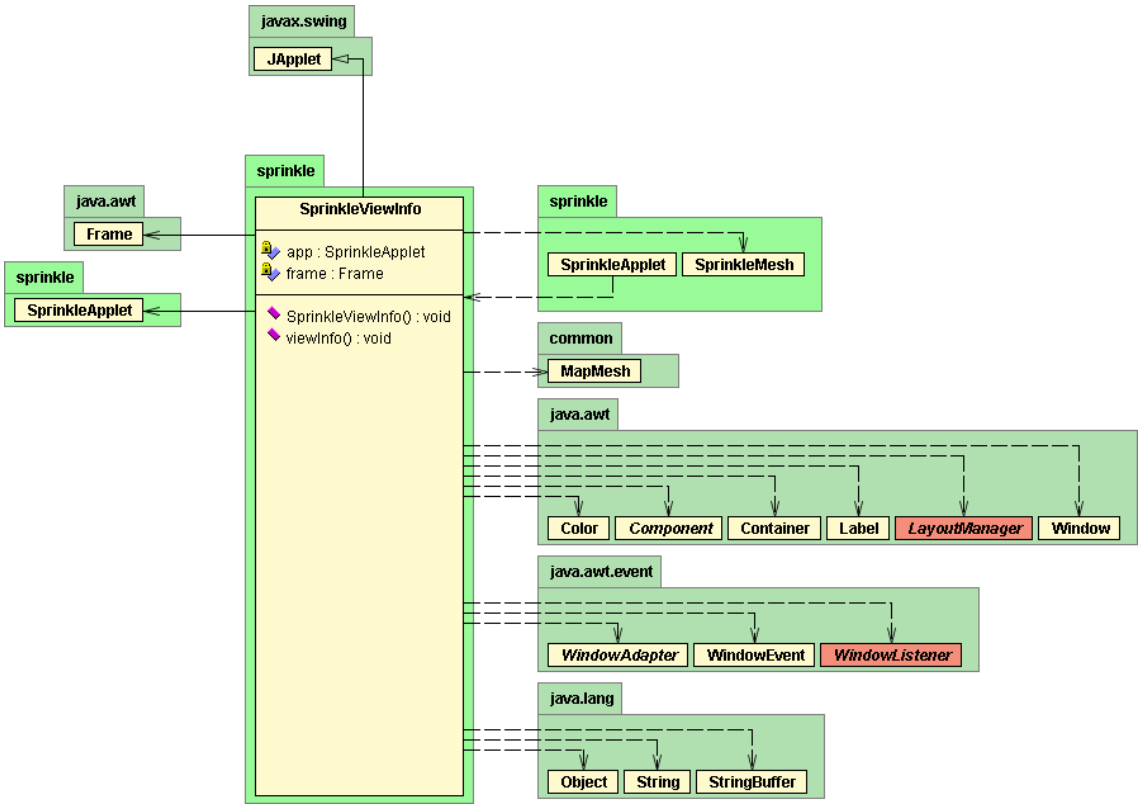
public java.lang.String **getJISYU_SYUBETSU**()

自動・手動種別の取得

戻り値:

自動・手動種別

■ ５　クラス SprinkleViewInfo




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Panel
                    |
                    +--java.applet.Applet
                          |
                          +--javax.swing.JApplet
                                |
                                +--sprinkle.SprinkleViewInfo
```

すべての実装インタフェース:

javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable

---

public class **SprinkleViewInfo**

extends javax.swing.JApplet

散布履歴確認用メッシュダイアログ

フィールドの概要

private SprinkleApplet	**app** 散布履歴確認アプレットクラス
private java.awt.Frame	**frame** 「散布メッシュ情報」ダイアログ用フレーム

コンストラクタの概要

SprinkleViewInfo(SprinkleApplet sprapp, java.awt.Frame frm, int k, int j, int lx, int ly) コンストラクタ

メソッドの概要

void	**viewInfo**(int k, int j, int lx, int ly) 「散布メッシュ情報」ダイアログの表示

フィールドの詳細

frame

private java.awt.Frame **frame**

「散布メッシュ情報」ダイアログ用フレーム

app

private SprinkleApplet **app**

散布履歴確認アプレットクラス

コンストラクタの詳細

SprinkleViewInfo

public **SprinkleViewInfo**(SprinkleApplet sprapp,
java.awt.Frame frm,
int k,
int j,
int lx,
int ly)

コンストラクタ

パラメータ:

sprapp - 散布履歴確認アプレットクラス

k - カウンタ（重なっている場合の連番）

j - メッシュ番号

lx - X 方向ダイアログ表示位置

ly - Y 方向ダイアログ表示位置

メソッドの詳細

viewInfo

public void **viewInfo**(int k,
int j,
int lx,
int ly)

「散布メッシュ情報」ダイアログの表示

パラメータ:

k - カウンタ（重なっている場合の連番）

j - メッシュ番号

lx - X 方向ダイアログ表示位置

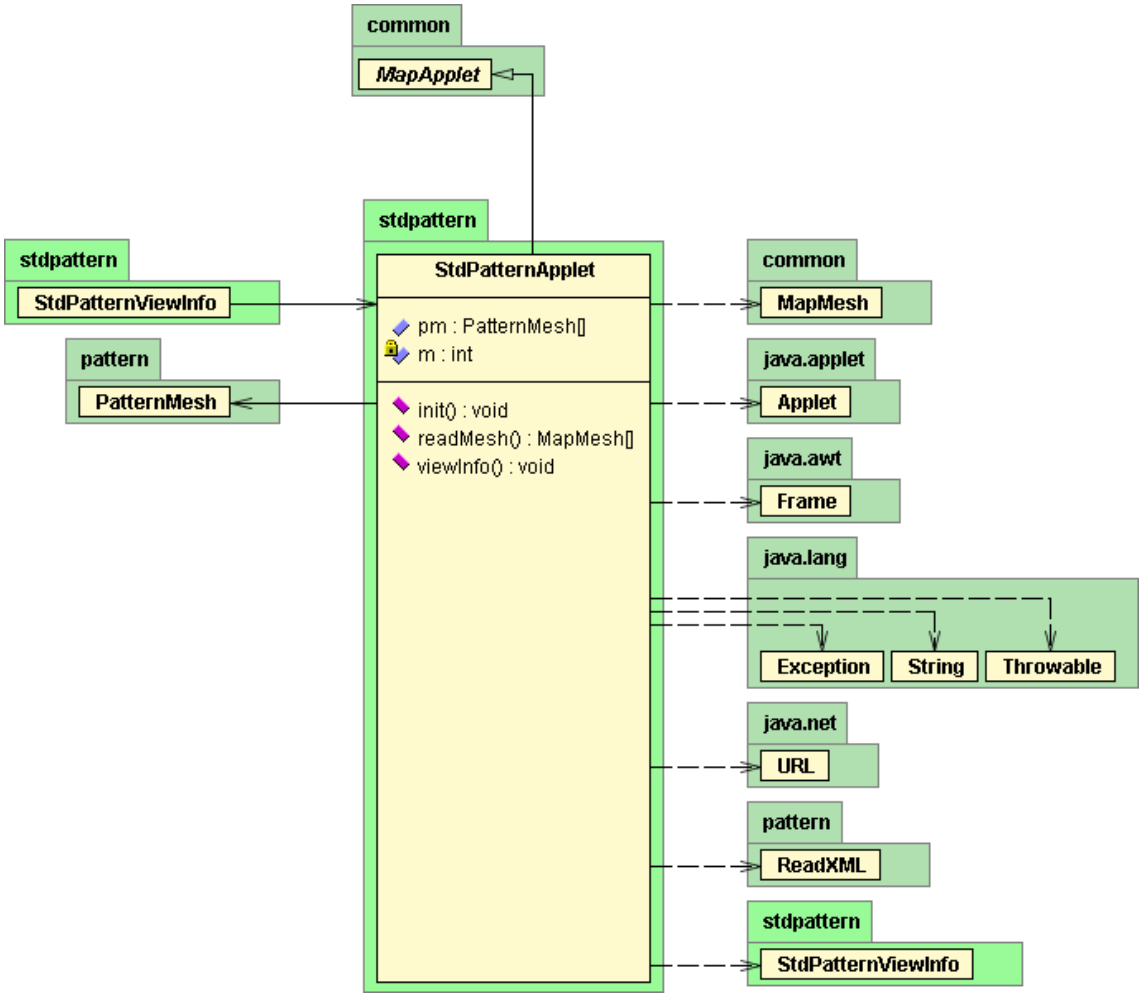
ly - Y 方向ダイアログ表示位置

---

l） パッケージ stdpattern

クラスの概要	
StdPatternApplet	標準散布パターン表示画面
StdPatternViewInfo	標準パターン確認用メッシュダイアログ

■ 1　クラス StdPatternApplet




```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Panel
                    |
                    +--java.applet.Applet
                          |
                          +--javax.swing.JApplet
                                |
                                +--common.MapApplet
                                      |
```

+--**stdpattern.StdPatternApplet**

すべての実装インタフェース:

javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable

---

public class **StdPatternApplet**
extends MapApplet

標準散布パターン表示画面

クラス **common.MapApplet** から継承した内部クラス
MapApplet.ButtonActionListener

フィールドの概要

private int	**m** メッシュ数
PatternMesh[]	**pm** メッシュクラス

クラス **common.MapApplet** から継承したフィールド
area, center, comMsize, comSsize, cont, down, drawmesh, east_ini, f, fMove, fRange, HEIGHT, iconMove, iconMove2, iconRange, iconRange2, id, image, left, leftdown, leftup, move, north_ini, p, range, right, rightdown, rightup, sessionID, south_ini, up, url, urladdr, user, west_ini, WIDTH, zoomin, zoomout

コンストラクタの概要

StdPatternApplet()

メソッドの概要

void	**init**() アプレットの初期化
MapMesh[]	**readMesh**() 散布計画 XML ファイル読込 MapApplet クラスで使用
void	**viewInfo**(java.awt.Frame frame, int k, int j, int lx, int ly) 「散布計画メッシュ情報」ダイアログの表示

クラス common.MapApplet から継承したメソッド
CalcMapDistance, close_roadgis, getConfig, getMap, getScale, IniMapSize, init_roadgis, setBtn, stop

フィールドの詳細

m

private int **m**

メッシュ数

---

pm

public PatternMesh[] **pm**

メッシュクラス

コンストラクタの詳細

StdPatternApplet

public **StdPatternApplet**()

メソッドの詳細

init

public void **init**()

アプレットの初期化

オーバーライド:

クラス MapApplet 内の init

---

readMesh

public MapMesh[] **readMesh**()

散布計画 XML ファイル読込 MapApplet クラスで使用

オーバーライド:

クラス MapApplet 内の readMesh

戻り値:

MapMesh[] MapMesh クラスの配列

---

viewInfo

public void **viewInfo**(java.awt.Frame frame,
int k,
int j,
int lx,
int ly)

「散布計画メッシュ情報」ダイアログの表示

オーバーライド:

クラス MapApplet 内の viewInfo

パラメータ:

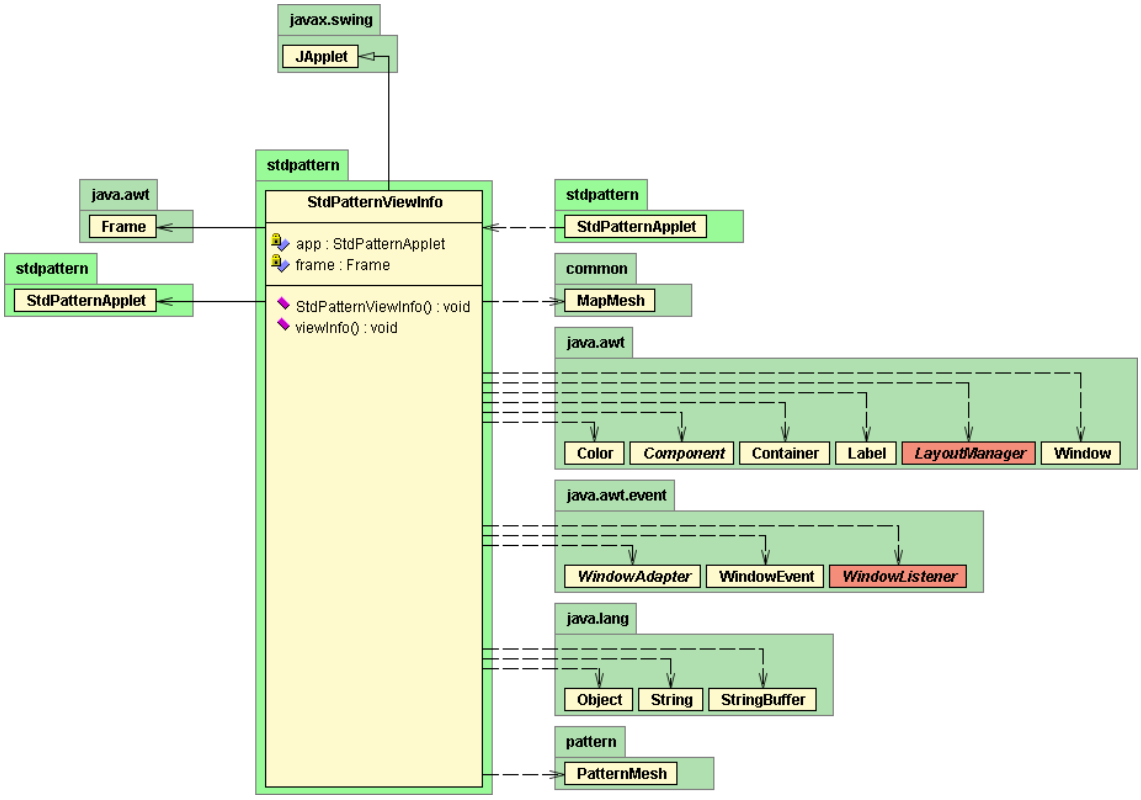
k - カウンタ（重なっている場合の連番）

j - メッシュ番号

lx - X 方向ダイアログ表示位置

ly - Y 方向ダイアログ表示位置

■ 2　クラス StdPatternViewInfo



java.lang.Object

  |

  +--java.awt.Component

        |

        +--java.awt.Container

              |

              +--java.awt.Panel

                    |

                    +--java.applet.Applet

                          |

                          +--javax.swing.JApplet

                                |

                                +--**stdpattern.StdPatternViewInfo**

すべての実装インタフェース:

javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable

---

public class **StdPatternViewInfo**

extends javax.swing.JApplet

標準パターン確認用メッシュダイアログ

フィールドの概要

private StdPatternApplet	**app** 標準散布パターン確認アプレットクラス
private java.awt.Frame	**frame** 「散布メッシュ情報」ダイアログ用フレーム

コンストラクタの概要

StdPatternViewInfo(StdPatternApplet stdapp, java.awt.Frame frm, int k, int j, int lx, int ly)
コンストラクタ

メソッドの概要

void	**viewInfo**(int k, int j, int lx, int ly) 「散布計画メッシュ情報」ダイアログの表示

フィールドの詳細

frame

private java.awt.Frame **frame**

「散布メッシュ情報」ダイアログ用フレーム

---

app

private StdPatternApplet **app**

標準散布パターン確認アプレットクラス

コンストラクタの詳細

StdPatternViewInfo

public **StdPatternViewInfo**(StdPatternApplet stdapp,
java.awt.Frame frm,
int k,
int j,
int lx,
int ly)

コンストラクタ

パラメータ:

stdapp - 標準散布パターン確認アプレットクラス

k - カウンタ（重なっている場合の連番）

j - メッシュ番号

lx - X 方向ダイアログ表示位置

ly - Y 方向ダイアログ表示位置

メソッドの詳細

viewInfo

public void **viewInfo**(int k,
int j,
int lx,
int ly)

「散布計画メッシュ情報」ダイアログの表示

パラメータ:

k - カウンタ（重なっている場合の連番）

j - メッシュ番号

lx - X 方向ダイアログ表示位置

ly - Y 方向ダイアログ表示位置

3.4. 計画支援システムの製作

製作した本システムのソースコードを別添資料に添付する。

3.5. 計画支援システムの試験（単体）

(1) 試験概要

本試験成績書は、計画支援システムについての単体試験結果を示すものである。単体試験方針としては、画面の遷移及び表示情報の整合性、データベースへの登録、更新、削除、参照についての動作確認をブラウザ（IE6.0）から実施した。なお、ホワイトボックステストでは全体的な処理の確認が不可能なため、機能部分のテストを重点においたブラックボックステストを行うこととした。

なお、試験は巡回支援システムが途中追加となったため２回に分けて試験を実施している。

(2) 試験結果

以下に、試験結果を示す。

表 3-188　単体試験結果

ID	画面	テスト項目	テスト概要	結果
Menu	トップメニュー	起動時	所属コードをセットする	○
A01	標準パターン一覧	起動時	登録された順に標準散布パターン名・説明を表示する	○
		標準パターン名リンク	選択したデータを基に標準パターン確認画面に遷移する	○
		戻るボタン	トップメニュー画面に遷移する	○
A02	標準パターン確認	起動時	選択した標準散布パターン名のXMLファイル名を取得する	○
			一時ファイル名(workfile+日時+.csv)を取得する	○
		出力ボタン	散布計画出力画面に遷移する	○
		戻るボタン	標準パターン一覧画面に遷移する	○
A03	散布計画出力	起動時	標準パターン一覧画面で選択された標準パターン名・説明が表示されている	○
		ＯＫボタン	標準パターン名＋.csvのファイルを任意のフォルダへ保存する	○
		戻るボタン	標準パターン確認画面に遷移する	○
		一覧へ戻るボタン	標準パターン一覧画面に遷移する	○
		メニューへボタン	トップメニュー画面に遷移する	○

表 3-189　単体試験結果

ID	画面	テスト項目	テスト概要	結果
B01	履歴選択	最新作業履歴表示リンク	最新データを基に散布履歴表示画面に遷移する	○
			最新作業履歴データがなければ「データが存在しません。」とエラーメッセージを表示する	○
		過去作業履歴選択リンク	散布履歴日時選択画面に遷移する	○
			過去作業履歴データがなければ「データが存在しません。」とエラーメッセージを表示する	○
		最新気温データ表示リンク	最新データを基に気温データ表示画面に遷移する	○
			最新気温データがなければ「データが存在しません。」とエラーメッセージを表示する	○
		過去気温データ選択リンク	気温データ表示履歴選択画面に遷移する	○
			過去気温データがなければ「データが存在しません。」とエラーメッセージを表示する	○
		戻るボタン	トップメニュー画面に遷移する	○
B11	散布履歴日時選択	起動時	作業日・作業時間で降順に作業日・作業時間・作業者名を表示する	○
		作業日リンク	選択したデータを基に散布履歴表示画面に遷移する	○
		戻るボタン	履歴選択画面に遷移する	○
B12	散布履歴表示	起動時	選択した散布履歴データのXMLファイル名を取得する	○
		戻るボタン	履歴選択画面から呼び出されたとき履歴選択画面に遷移する	○
			散布履歴日時選択画面から呼び出されたとき散布履歴日時選択画面に遷移する	○
		メニューへボタン	トップメニュー画面に遷移する	○
B21	気温データ表示履歴選択		―	―
B22	気温データ表示		―	―

表 3-190　単体試験結果

ID	画面	テスト項目	テスト概要	結果
C01	登録ファイル種別選択	散布作業記録の登録リンク	散布作業記録ファイル選択画面に遷移する	○
		気温データの登録リンク	気温データファイル選択画面に遷移する	○
		戻るボタン	トップメニュー画面に遷移する	○
C11	散布作業記録ファイル選択	参照ボタン	ファイル選択ダイアログを表示する	○
		送信ボタン	参照ファイル名が未入力のときの「ファイルが選択されていません。ファイルを選択してください。」とエラーメッセージを表示する	○
			存在しないファイル名を入力されたとき「ファイルが存在しないかもしくはデータがありません。」とエラーメッセージを表示する	○
			0バイトのファイル名を入力されたとき「ファイルが存在しないかもしくはデータがありません。」とエラーメッセージを表示する	○
			散布記録ファイルのデータ部が存在しないとき「データがありません。」とエラーメッセージを表示する	○
			散布記録ファイルフォーマット以外のとき「指定されたファイルを読込めません。」とエラーメッセージを表示する	○
		戻るボタン	登録ファイル種別選択画面に遷移する	○
C12	散布作業記録ファイル登録完了	起動時	登録した作業日・作業時間・作業者名・標準パターン名を表示する	○
		メニューへボタン	トップメニュー画面に遷移する	○
C21	気温データファイル選択		ー	ー
C22	気温データファイル登録完了		ー	ー

表 3-191　単体試験結果

ID	画面	テスト項目	テスト概要	結果
D01	標準散布パターン作成メニュー	管理メッシュ作成リンク	管理メッシュ一覧画面に遷移する	○
		散布基本データの作成リンク	散布基本データ一覧画面に遷移する	○
		標準散布パターン作成リンク	標準散布パターン一覧（登録・修正・削除）画面に遷移する	○
		標準散布パターン確認リンク	標準散布パターン一覧（照会）画面に遷移する	○
		戻るボタン	トップメニュー画面に遷移する	○
D11	管理メッシュ一覧	起動時	開始KP・終了KP・車線区別で降順にメッシュID・路線名・車線区別・開始KP・終了KPを表示する	○
		メッシュIDリンク	管理メッシュ修正画面に遷移する	○
		新規ボタン	管理メッシュ新規登録画面に遷移する	○
		戻るボタン	標準散布パターン作成メニュー画面に遷移する	○
		メニューへボタン	トップメニュー画面に遷移する	○

表 3-192　単体試験結果

ID	画面	テスト項目	テスト概要	結果
D12	管理メッシュ修正	起動時	車線マスタに登録されている車線名称を車線区別コンボボックスの表示する	○
			管理メッシュ一覧で選択されたメッシュIDを基にメッシュID・路線名・車線区別・開始KP・終了KPを表示する	○
		修正ボタン	路線名が未入力のときの「路線名が入力されていません。」とエラーメッセージを表示する	○
			開始KPが未入力のときの「開始KPが入力されていません。」とエラーメッセージを表示する	○
			終了KPが未入力のときの「終了KPが入力されていません。」とエラーメッセージを表示する	○
			路線名の入力文字を10文字までとする	○
			開始KPが数値以外を入力されたとき「開始KPは数値を入力してください。」とエラーメッセージを表示する	○
			終了KPが数値以外を入力されたとき「終了KPは数値を入力してください。」とエラーメッセージを表示する	○
			上り車線のとき開始KP<終了KPなら「開始KP>=終了KPの範囲で入力してください。」とエラーメッセージを表示する	○
			下り車線のとき開始KP>終了KPなら「開始KP<=終了KPの範囲で入力してください。」とエラーメッセージを表示する	○
			メッシュマスタに入力された開始KPが存在しないとき「この車線にはこの開始KPが存在しません。正しい開始KPを入力してください。」とエラーメッセージを表示する	○
			メッシュマスタに入力された終了KPが存在しないとき「この車線にはこの終了KPが存在しません。正しい終了KPを入力してください。」とエラーメッセージを表示する	○
		複製ボタン	表示されているメッシュIDを基に管理メッシュ新規登録画面に遷移する	○
		削除ボタン	表示されているメッシュIDを削除後、管理メッシュ一覧画面に遷移する	○
		戻るボタン	管理メッシュ一覧画面に遷移する	○
		ヘルプボタン	管理メッシュヘルプ画面を別ウィンドウで表示する	○

表 3-193　単体試験結果

ID	画面	テスト項目	テスト概要	結果
D13	管理メッシュ新規登録	起動時	車線マスタに登録されている車線名称を車線区別コンボボックスに表示する	○
			管理メッシュ修正画面からの複製ボタンで呼び出された場合、路線名・車線区別・開始KP・終了KPを表示する	○
		登録ボタン	メッシュIDが入力のとき「メッシュIDが入力されていません。」とエラーメッセージを表示する	○
			路線名が未入力のとき「路線名が入力されていません。」とエラーメッセージを表示する	○
			開始KPが未入力のとき「開始KPが入力されていません。」とエラーメッセージを表示する	○
			終了KPが未入力のとき「終了KPが入力されていません。」とエラーメッセージを表示する	○
			メッシュIDの入力文字を12文字までとする	○
			路線名の入力文字を10文字までとする	○
			開始KPが数値以外を入力されたとき「開始KPは数値を入力してください。」とエラーメッセージを表示する	○
			終了KPが数値以外を入力されたとき「終了KPは数値を入力してください。」とエラーメッセージを表示する	○
			上り車線のとき開始KP<終了KPなら「開始KP>=終了KPの範囲で入力してください。」とエラーメッセージを表示する	○
			下り車線のとき開始KP>終了KPなら「開始KP<=終了KPの範囲で入力してください。」とエラーメッセージを表示する	○
			入力されたメッシュIDがすでに登録されていたとき「このメッシュIDは既に登録済みです。違うメッシュIDを入力してください。」とエラーメッセージを表示する	○
			メッシュIDに「 ' ″ 」が含まれているとき「メッシュIDに『 ' ″ 』は入力できません。」とエラーメッセージを表示する	○
			メッシュマスタに入力された開始KPが存在しないとき「この車線にはこの開始KPが存在しません。正しい開始KPを入力してください。」とエラーメッセージを表示する	○
			メッシュマスタに入力された終了KPが存在しないとき「この車線にはこの終了KPが存在しません。正しい終了KPを入力してください。」とエラーメッセージを表示する	○
		戻るボタン	管理メッシュ一覧画面に遷移する	○
		ヘルプボタン	管理メッシュヘルプ画面を別ウィンドウで表示する	○

表 3-194　単体試験結果

ID	画面	テスト項目	テスト概要	結果
D21	散布基本データ一覧	起動時	散布量・散布幅で昇順にデータID・散布量・散布幅・散布方向・散布薬剤を表示	○
		データIDリンク	散布基本データ修正画面に遷移する	○
		新規ボタン	散布基本データ新規登録画面に遷移する	○
		戻るボタン	標準散布パターン作成メニュー画面に遷移する	○
		メニューへボタン	トップメニュー画面に遷移する	○
D22	散布基本データ修正	起動時	散布方向マスタに登録されている散布方向名称を散布方向コンボボックスに表示	○
			散布薬剤マスタに登録されている散布薬剤名称を散布薬剤コンボボックスに表示	○
			散布基本データ一覧で選択されたデータIDを基にデータID・散布量・散布幅・散布方向・散布薬剤を表示する	○
		修正ボタン	~~散布量が未入力のとき「散布量が入力されていません。」とエラーメッセージを表示する~~	○
			~~散布幅が未入力のとき「散布幅が入力されていません。」とエラーメッセージを表示する~~	○
			~~散布量が数値以外を入力されたとき「散布量は数値を入力してください。」とエラーメッセージを表示する~~	○
			~~散布幅が数値以外を入力されたとき「散布幅は数値を入力してください。」とエラーメッセージを表示する~~	○
			~~散布量が15から50の間以外を入力されたとき「散布量は15から50の間で入力してください。」とエラーメッセージを表示する~~	○
			~~散布幅が3.0から7.0の間以外を入力されたとき「散布幅は3.0から7.0の間で入力してください。」とエラーメッセージを表示する~~	○
		複製ボタン	表示されているデータIDを基に散布基本データ新規登録画面に遷移する	○
		削除ボタン	表示されているデータIDを削除後、散布基本データ一覧画面に遷移する	○
		戻るボタン	散布基本データ一覧画面に遷移する	○
		ヘルプボタン	散布基本データヘルプ画面を別ウィンドウで表示する	○

表 3-195　単体試験結果

ID	画面	テスト項目	テスト概要	結果
D23	散布基本データ新規登録	起動時	散布方向マスタに登録されている散布方向名称を散布方向コンボボックスに表示する	○
			散布薬剤マスタに登録されている散布薬剤名称を散布薬剤コンボボックスに表示する	○
			散布基本データ修正画面からの複製ボタンで呼び出された場合、散布量・散布幅・散布方向・散布薬剤を表示する	○
		登録ボタン	データIDが未入力のとき「データIDが入力されていません。」とエラーメッセージを表示する	○
			~~散布量が未入力のとき「散布量が入力されていません。」とエラーメッセージを表示する~~	○
			~~散布幅が未入力のとき「散布幅が入力されていません。」とエラーメッセージを表示する~~	○
			データIDの入力文字を６文字までとする	○
			~~散布量が数値以外を入力されたとき「散布量は数値を入力してください。」とエラーメッセージを表示する~~	○
			~~散布幅が数値以外を入力されたとき「散布幅は数値を入力してください。」とエラーメッセージを表示する~~	○
			入力されたデータIDがすでに登録されていたとき「このデータIDは既に登録済みです。違うデータIDを入力してください。」とエラーメッセージを表示する	○
			データIDに「’″」が含まれているとき「データIDに『’″』は入力できません。」とエラーメッセージを表示する	○
			~~散布量が15から50の間以外を入力されたとき「散布量は15から50の間で入力してください。」とエラーメッセージを表示する~~	○
			~~散布幅が3.0から7.0の間以外を入力されたとき「散布幅は3.0から7.0の間で入力してください。」とエラーメッセージを表示する~~	○
		戻るボタン	散布基本データー覧画面に遷移する	○
		ヘルプボタン	散布基本データヘルプ画面を別ウィンドウで表示する	○

表 3-196　単体試験結果

ID	画面	テスト項目	テスト概要	結果
D31	標準散布パターン一覧 （登録・修正・削除）	起動時	登録された順に標準散布パターン名・説明を表示する	○
		標準散布パターン名リンク	標準散布パターン修正画面に遷移する	○
		戻るボタン	標準散布パターン作成メニュー画面に遷移する	○
		メニューへボタン	トップメニュー画面に遷移する	○
D32	標準散布パターン修正	起動時	管理メッシュに登録されているメッシュIDを管理メッシュIDコンボボックスに表示する	○
			散布基本データに登録されているデータIDを散布基本データIDコンボボックスに表示する	○
			1から5を散布順コンボボックスに表示する	○
			標準散布パターン一覧（登録・修正・削除）で選択されたデータを基に標準散布パターン名・説明・管理メッシュID・散布基本データID・散布順を表示する	○
		修正ボタン	説明が未入力のとき「説明が入力されていません。」とエラーメッセージを表示する	○
			説明の入力文字を300文字までとする	○
			管理メッシュ・散布基本データが１つ以上存在しないとき「管理メッシュ・散布基本データを作成してください。」とエラーメッセージを表示する	○
		複製ボタン	表示されている標準散布パターン名を基に標準散布パターン新規登録画面に遷移する	○
		削除ボタン	表示されている標準散布パターン名を基にデータを削除して標準散布パターン一覧（登録・修正・削除）画面に遷移する	○
		修正・削除ボタン	管理メッシュ散布基本データの修正・削除画面に遷移する	○
		追加ボタン	選択された管理メッシュID・散布基本データID・散布順を追加する	○
		戻るボタン	標準散布パターン一覧（登録・修正・削除）画面に遷移する	○
		ヘルプボタン	標準散布パターンヘルプ画面を別ウィンドウで表示する	○

表 3-197　単体試験結果

ID	画面	テスト項目	テスト概要	結果
D33	標準散布パターン新規登録	起動時	管理メッシュに登録されているメッシュIDを管理メッシュIDコンボボックスに表示する	○
			散布基本データに登録されているデータIDを散布基本データIDコンボボックスに表示する	○
			1から5を散布順コンボボックスに表示する	○
			標準散布パターン修正画面からの複製ボタンで呼び出された場合、説明・管理メッシュID・散布基本データID・散布順を表示する	○
		登録ボタン	標準散布パターン名が未入力のとき「標準散布パターン名が入力されていません。」とエラーメッセージを表示する	○
			説明が未入力のとき「説明が入力されていません。」とエラーメッセージを表示する	○
			標準散布パターン名の入力文字を40文字までとする	○
			説明の入力文字を300文字までとする	○
			管理メッシュ・散布基本データが１つ以上存在しないとき「管理メッシュ・散布基本データを作成してください。」とエラーメッセージを表示する	○
			標準散布パターン名の先頭が「 . 」のとき「標準散布パターン名の先頭に『.』は入力できません。」とエラーメッセージを表示する	○
			標準散布パターン名に「 ￥ / ： ； * ? ″ < > | 」が含まれているとき「標準散布パターン名に『 ￥ / ： ； * ? ″ < > | 』は入力できません。」とエラーメッセージを表示する	○
		修正・削除ボタン	管理メッシュ散布基本データの修正・削除画面に遷移する	○
		追加ボタン	選択された管理メッシュID・散布基本データID・散布順を追加する	○
		戻るボタン	標準散布パターン一覧（登録・修正・削除）画面に遷移する	○
		ヘルプボタン	標準散布パターンヘルプ画面を別ウィンドウで表示する	○

表 3-198　単体試験結果

ID	画面	テスト項目	テスト概要	結果
D34	管理メッシュ散布基本データの 修正・削除	起動時	標準散布パターン名・説明・管理メッシュID・散布基本データID・散布順を表示する	○
		修正ボタン	管理メッシュID・散布基本データID・散布順を修正する	○
			標準散布パターン修正画面から呼び出されたとき標準散布パターン修正画面に遷移する	○
			標準散布パターン新規登録画面から呼び出されたとき標準散布パターン新規登録画面に遷移する	○
		削除ボタン	管理メッシュID・散布基本データID・散布順を削除する	○
			標準散布パターン修正画面から呼び出されたとき標準散布パターン修正画面に遷移する	○
			標準散布パターン新規登録画面から呼び出されたとき標準散布パターン新規登録画面に遷移する	○
		戻るボタン	標準散布パターン修正画面から呼び出されたとき標準散布パターン修正画面に遷移する	○
			標準散布パターン新規登録画面から呼び出されたとき標準散布パターン新規登録画面に遷移する	○
		ヘルプボタン	標準散布パターンヘルプ画面を別ウィンドウで表示する	○
D41	標準散布パターン一覧（照会）	起動時	登録された順に標準散布パターン名・説明を表示する	○
		標準散布パターン名リンク	標準散布パターン表示画面に遷移する	○
		戻るボタン	標準散布パターン作成メニュー画面に遷移する	○
D42	標準散布パターン表示	起動時	選択した標準散布パターン名のXMLファイル名を取得する	○
		戻るボタン	標準散布パターン一覧（照会）画面に遷移する	○
		メニューへボタン	トップメニュー画面に遷移する	○

表 3-199　単体試験結果

ID	画面	テスト項目	テスト概要	結果
B01	履歴選択	最新巡回記録表示リンク	最新の巡回記録が表示またはダウンロードされる	○
			最新巡回記録がなければ「データが存在しません。」とエラーメッセージを表示する	○
		過去巡回記録選択リンク	散布履歴日時選択画面に遷移する	○
			過去巡回記録がなければ「データが存在しません。」とエラーメッセージを表示する	○
B21	巡回記録日時選択	起動時	巡回日・巡回時間で降順に巡回日・巡回時間・運転者名を表示する	○
		作業日リンク	選択したデータを基に巡回記録が表示またはダウンロードされる	○
		戻るボタン	履歴選択画面に遷移する	○
B31	巡回記録修正版登録	起動時	巡回日・巡回時間で降順に巡回日・巡回時間・巡回記録ファイル名を表示する	○
		参照ボタン	ファイル選択ダイアログを表示する	○
		送信ボタン	参照ファイル名が未入力のときの「ファイルが選択されていません。ファイルを	○
			登録されている巡回記録ファイル名と名前が異なるファイルを選択した時、エ	
		戻るボタン	登録ファイル種別選択画面に遷移する	○
B32	散布作業記録ファイル登録完了	起動時	登録完了画面が表示される	○
		戻るボタン	履歴選択画面に遷移する	
		メニューへボタン	トップメニュー画面に遷移する	○

表 3-200　単体試験結果

ID	画面	テスト項目	テスト概要	結果
C21	巡回記録ファイル選択	参照ボタン	ファイル選択ダイアログを表示する	○
		送信ボタン	参照ファイル名が未入力のときの「ファイルが選択されていません。ファイルを選択してください。」とエラーメッセージを表示する	○
			存在しないファイル名を入力されたとき「ファイルが存在しないかもしくはデータがありません。」とエラーメッセージを表示する	○
			0バイトのファイル名を入力されたとき「ファイルが存在しないかもしくはデータがありません。」とエラーメッセージを表示する	○
			巡回記録ファイルのデータ部が存在しないとき「データがありません。」とエラーメッセージを表示する	○
			巡回記録ファイルフォーマット以外のとき「指定されたファイルを読込めません。」とエラーメッセージを表示する	○
		戻るボタン	登録ファイル種別選択画面に遷移する	○
C22	巡回記録ファイル登録完了	起動時	登録した巡回日・巡回時間・運転者名を表示する	○
		メニューへボタン	トップメニュー画面に遷移する	○
Menu	トップメニュー	作業者名リスト作成リンク	作業者名リスト一覧が表示される	○

表 3-201　単体試験結果

ID	画面	テスト項目	テスト概要	結果
E01	作業者名一覧	起動時	作業者名と修正・削除ボタンを表示し、一番最後に作業者名の入力欄と追加ボタンが表示される	○
		修正削除ボタン	作業者名修正画面に遷移する	○
		追加ボタン	作業者名が未入力のときエラーメッセージが表示される	○
			作業者名が入力されたとき一覧に作業者名が追加される	
		出力ボタン	ファイル保存ダイアログが表示されファイル保存ができる	○
		戻るボタン	トップメニュー画面に遷移する	○
E02	作業者名修正	起動時	選択した作業者名が表示される	○
		修正ボタン	作業者名が未入力のときエラーメッセージが表示される	○
			作業者名が入力されたとき一覧画面に遷移し、作業者名の修正が反映されている	○
		削除ボタン	削除確認ボタンが表示され、ＯＫボタンを押したとき一覧画面に遷移し、一覧から作業者名が削除されていること	○
		戻るボタン	作業者名一覧画面に遷移する	○

表 3-202　単体試験結果

ID	画面	テスト項目	テスト概要	結果
APPLET	散布計画出力	道路GIS画像	正常に表示され、拡大、縮小、スクロールができる	○
		散布メッシュ	道路GIS画像とアーカイブされている	○
		計画変更ダイアログ	散布メッシュの情報が正しく表示される	○
			散布メッシュの情報が訂正できる	○
			散布メッシュの訂正情報で計画ファイルがダウンロードできる	○
	散布履歴表示	道路GIS画像	正常に表示され、拡大、縮小、スクロールができる	○
		散布履歴メッシュ	道路GIS画像とアーカイブされている	○
		履歴詳細ダイアログ	散布履歴情報が正しく表示される	○
	散布履歴表示	道路GIS画像	正常に表示され、拡大、縮小、スクロールができる	○
		散布計画メッシュ	道路GIS画像とアーカイブされている	○
		計画詳細ダイアログ	散布計画情報が正しく表示される	○

4. 巡回支援システムの製作

4.1. 巡回支援システムの基本設計

(1) 適用範囲

本節は、巡回支援システムにおける基本設計についてのみ記載するものとする。なお、巡回記録表示機能は計画支援システムに含まれるためここでは記載しない。

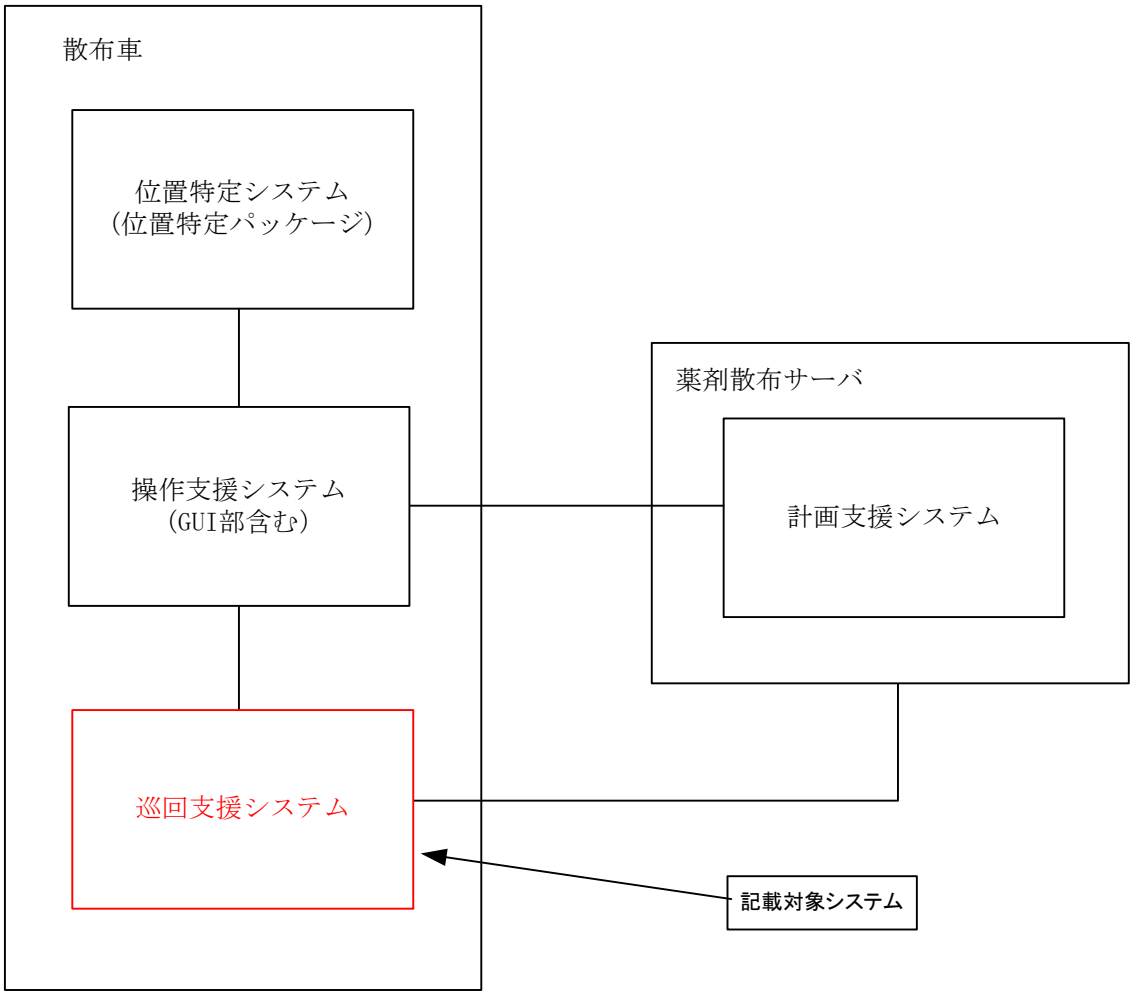


図 4-1　記載範囲

(2) 要件定義の確認

本システム全体における要件を以下のユースケース図に示す。なお、本文に記載する要件については、赤色で示されるユースケース部分である。

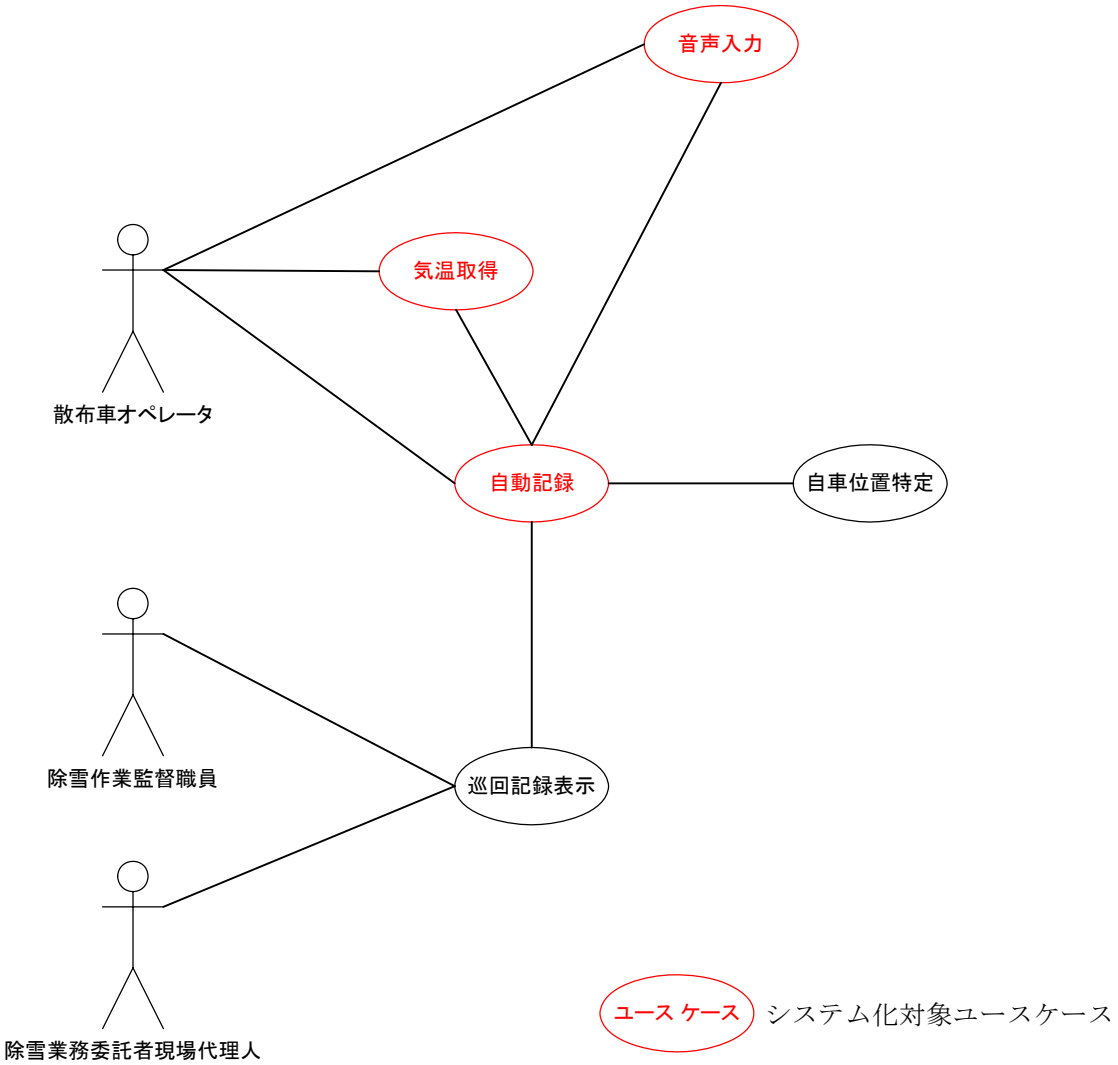


図 4-2　システム全体ユースケース図

(3) システム全体構成

システムの全体構成および処理の流れを以下に示す。

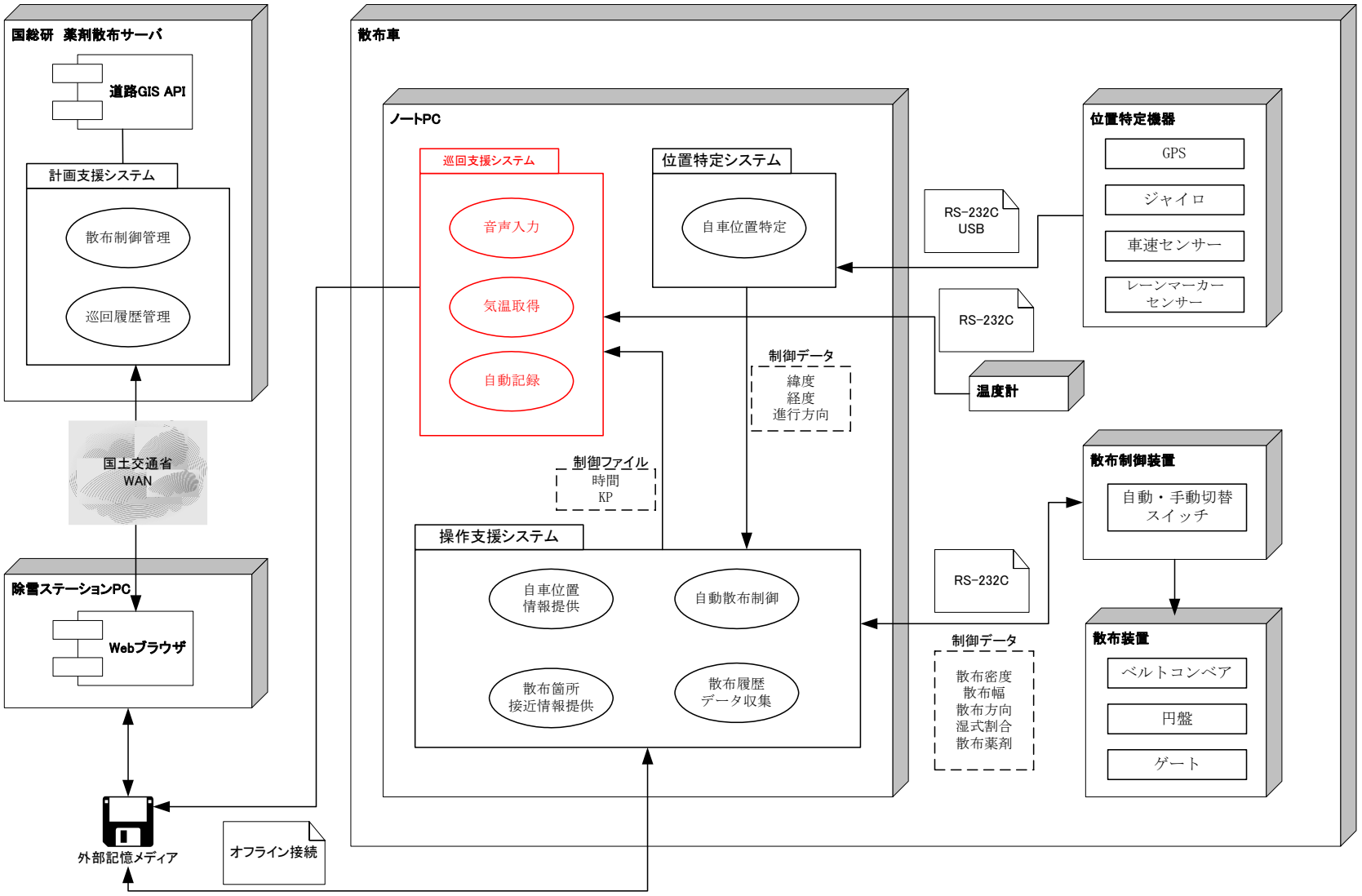


図 4-3　システム全体構成図

また、システム全体における処理の流れを以下のシーケンス図に示す。なお、図中の赤色の部分が本基本設計書に該当する箇所である。

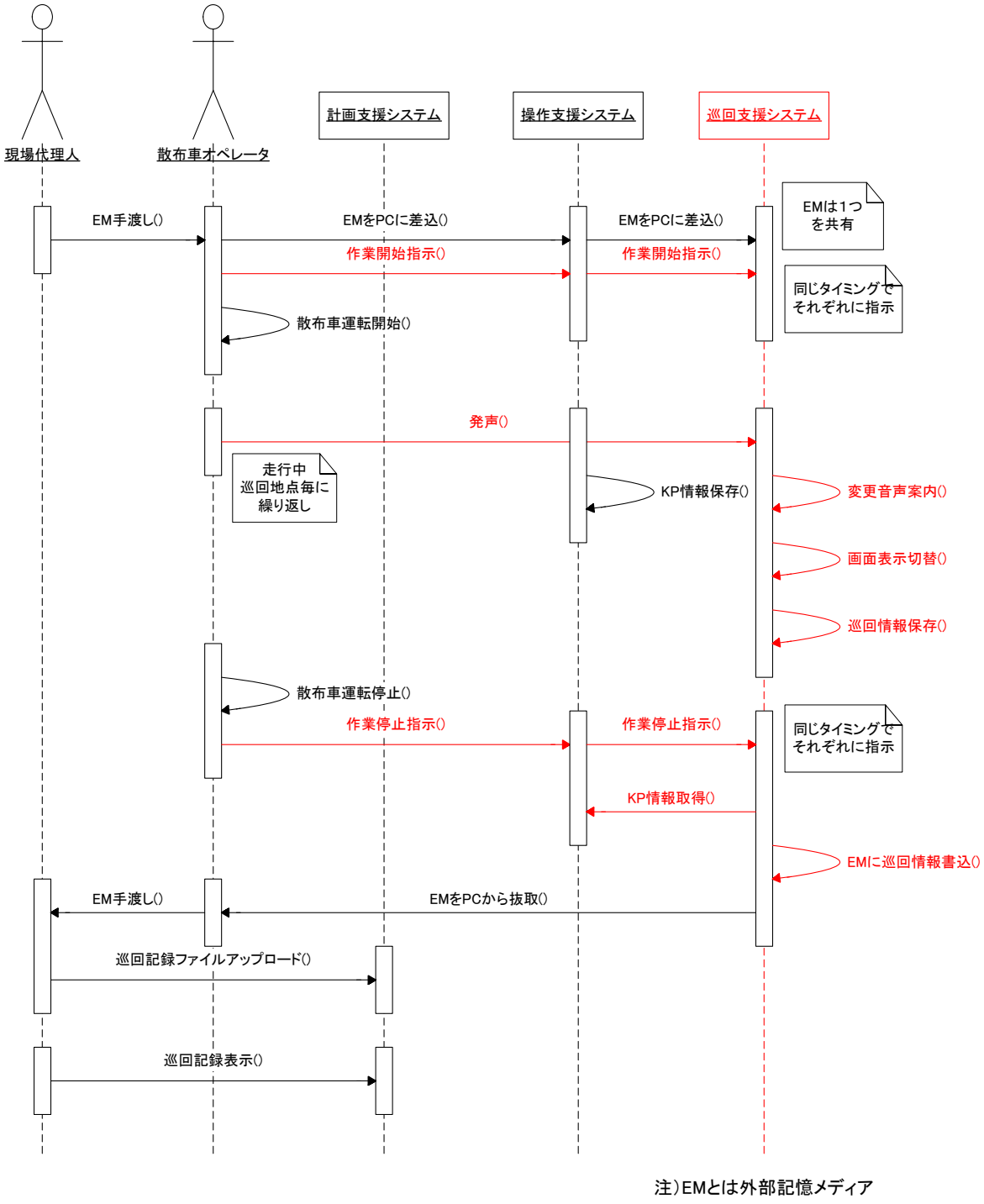


図 4-4　システム全体シーケンス図

(4) 機能設計

■ ユースケースの詳細化

システムとして必要な機能を詳細化するために、各ユースケースをさらに詳細化する。詳細化した機能を表 4-1 に示す。

表 4-1　システムに必要な機能

機能		詳細機能	機能概要
巡回支援	音声入力	音声入力機能	巡回地点の状況を音声によってシステム上に入力する機能。
		入力状況音声案内機能	巡回地点の状況が音声によって正しくシステム上に反映されたことを知らせるため、反映された状況を音声によって知らせる機能。
		設定状況表示機能	現在の設定巡回状況を画面上に表示する機能。
	気温取得	自動気温取得機能	外気温を温度計から自動的に取得し、巡回記録に反映させる機能。
		気温表示機能	現在の気温を画面上に表示する機能。
	自動記録	時間取得機能	巡回地点の通過時刻を自動的に取得する機能。
		巡回地点自動判定機能	巡回地点を通過した時の操作を行わずに、巡回地点の判定を行う機能。
		巡回地点情報収集機能	音声入力された巡回地点の情報をファイルに格納し、計画支援システム上で巡回記録を自動生成するためのデータを自動作成する機能。

■ **音声入力機能**

音声入力機能は、巡回記録に必要な情報をスイッチやパソコンの操作を行わず、音声によって入力を行うものである。機能の制約としては、以下とする。

- 音声は散布者オペレーター（運転手）が発声するものとし、運転中は音声以外の操作が不要なものとする
- 音声はシンプルなものとし、発声がし易くなおかつ、巡回記録の出力に忠実なものとする
- 発声した音声は、散布車の騒音状況によっては認識が悪くなる可能性があるので、認識された場合には、認識された音声をパソコンの音声で知らせることができること
- 現在の設定の状況が分からなくなった時のために、画面上に文字で現在の設定が表示されること

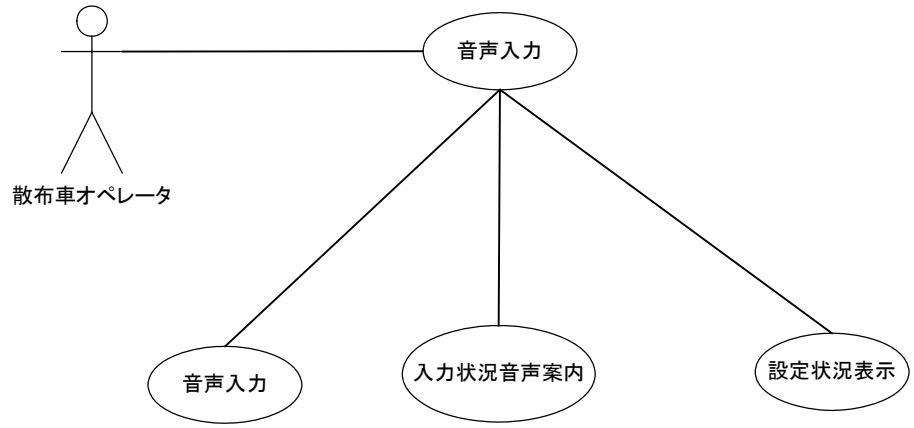


図 4-5　音声入力　詳細ユースケース図

表 4-2　音声入力　ユースケースシナリオ

ユースケース	音声入力	主アクタ	散布車オペレーター

事前条件

マイクがパソコンに正しく接続されており、音声認識ができること。

散布車の発進箇所は、除雪ステーションとする。

メインフロー

１．散布車オペレーターは、画面上の「開始」ボタン（操作支援システム）を押す。

２．散布車オペレーターは、散布車を発進する。

３．散布車オペレーターは、巡回地点に来たので、巡回地点の状況（天候、路面状況、交通状況、雪種別、積雪深、消雪パイプ稼動状況）を音声で発音する。

４．本システムは、散布車オペレーターの発声した音声を認識した場合、認識した状況を PC 上の音声で発生する。

５．散布車オペレーターの発声した音声を認識しない場合は、PC 上の音声は何も発生しない（散布者オペレーターは再度発声を行う）。

６．本システムは、現在の巡回情報を作業ファイルに保存する。

７．３から６の処理を繰り返す

８．散布車オペレーターは、散布車を停止する。

９．散布車オペレーターは、画面上の「停止」ボタン（操作支援システム）を押す。

１０．本システムは、操作支援システムの時間－KP 対応ファイルを読み込んで、巡回地点の情報をマッチングし、巡回記録ファイルを生成し、システムを終了する：例外処理１

事後条件

散布車の停止箇所は、除雪ステーションとする。

例外フロー

例外処理１：巡回支援システムをそのまま終了するか、引き続き、別の箇所の巡回を実施するかを選択して終了する。別の箇所の巡回を実施する場合は、巡回記録ファイルは作成せず、次回本システム起動時に前回の設定を読み込んで巡回情報をマージする。

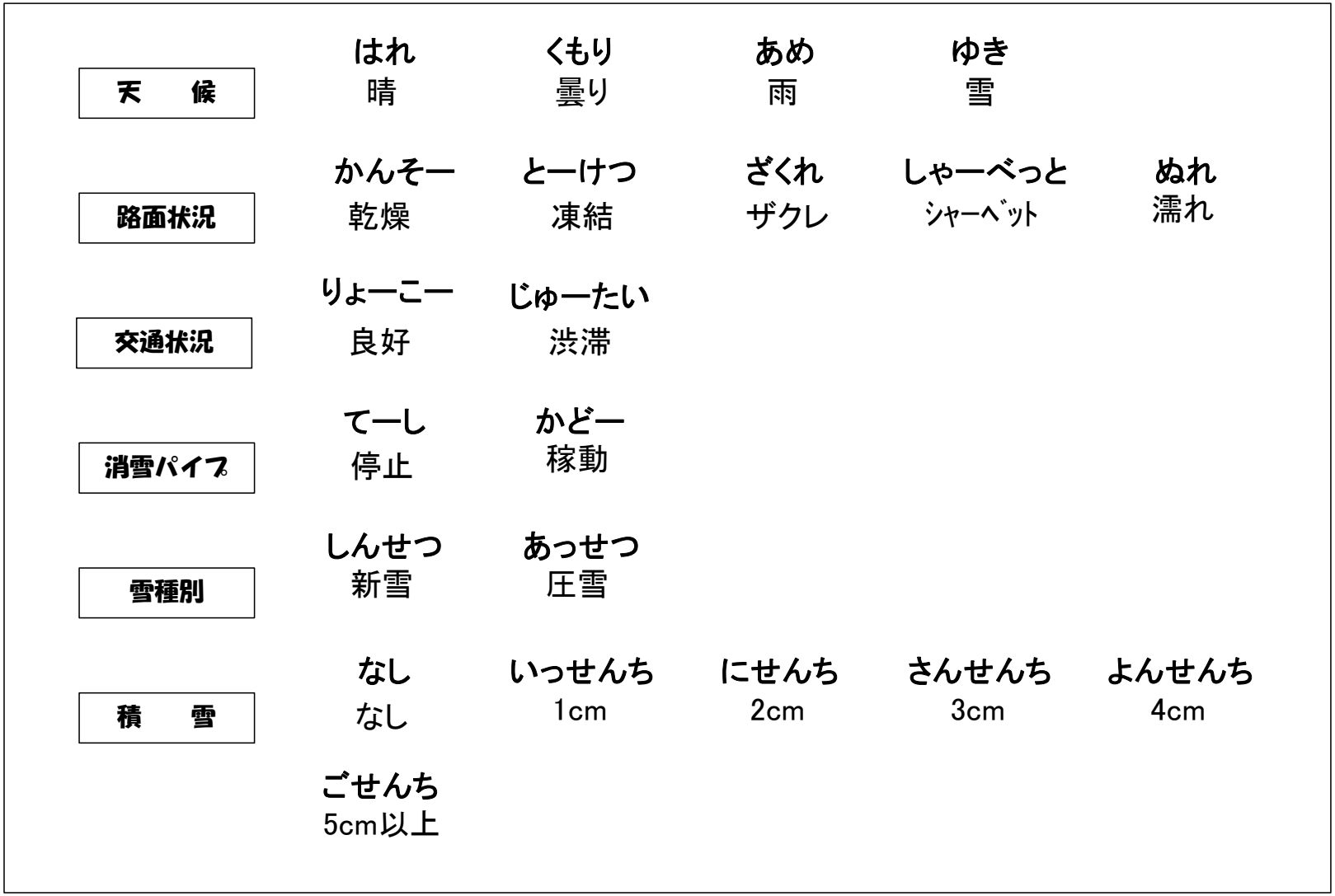

天　候
はれ
晴
くもり
曇り
あめ
雨
ゆき
雪
路面状況
かんそー
乾燥
とーけつ
凍結
ざくれ
ザクレ
しゃーべっと
ｼｬｰﾍﾞｯﾄ
ぬれ
濡れ
交通状況
りょーこー
良好
じゅーたい
渋滞
消雪パイプ
てーし
停止
かどー
稼動
雪種別
しんせつ
新雪
あっせつ
圧雪
積　雪
なし
なし
いっせんち
1cm
にせんち
2cm
さんせんち
3cm
よんせんち
4cm
ごせんち
5cm以上

図 4-6　音声一覧

■ **気温取得機能**

気温取得機能は、散布車両に気温計を取り付け、自動的に気温を取得し巡回記録として取り込むものである。機能の制約としては、以下とする。

- 気温計から気温を取得する
- 取得した気温を PC の画面に表示する
- 取得した気温を巡回記録に反映させる

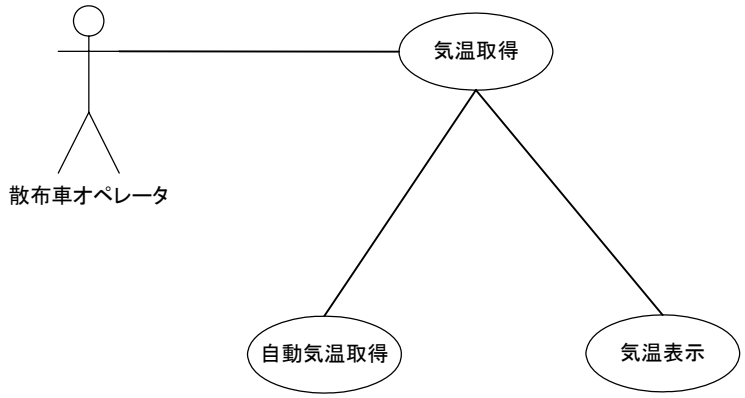


図 4-7　気温取得　詳細ユースケース図

表 4-3　気温取得　ユースケースシナリオ

ユースケース	気温取得	主アクタ	散布車オペレータ ー

事前条件

気温計が散布車に設置されており、PC との通信が可能な状態になっていること。

メインフロー

1．本システムは、システム起動時に、気温計との通信を確立する：例外処理１

2．本システムは、気温計から気温を取得する。

3．取得した気温を PC の画面上に数値（摂氏）で表示する。

4．気温を作業ファイルに記録する。

5．散布車オペレーターが本システムを終了するときに、気温計との通信を切断する。

事後条件

例外フロー

例外処理１：気温計の電源が入っていない場合は、電源の投入を促すメッセージを表示する。

■ **自動記録機能**

自動記録機能は、巡回地点の位置と散布車の走行位置とのマッチングを行い、巡回地点の通過時間や巡回地点の状況を自動的に生成するものである。機能の制約としては、以下とする。

- 巡回地点の位置と散布車の走行位置とのマッチングを行う
- 巡回地点の通過時刻を自動的に取得する
- 巡回地点の自動判定を行う
- 位置情報と音声入力された巡回地点の情報および気温をマッチングさせ、巡回記録生成用のデータファイルを作成する

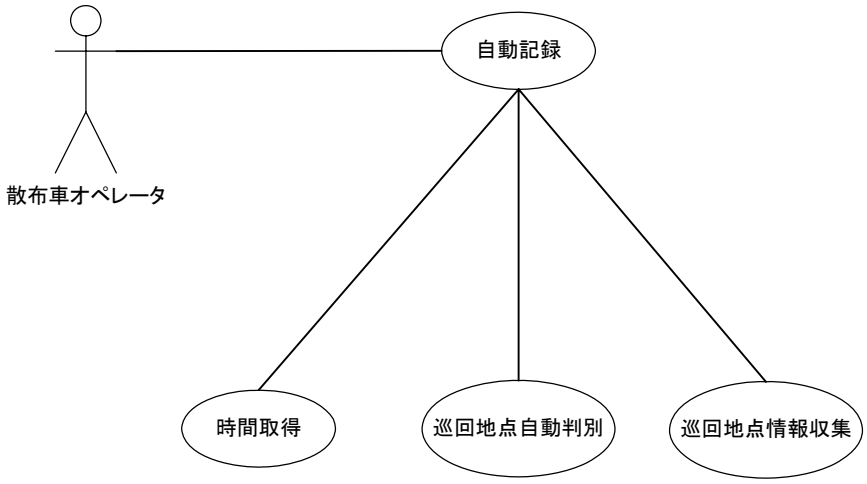


図 4-8　自動記録　詳細ユースケース図

表 4-4　自動記録機能　ユースケースシナリオ

ユースケース	自動記録機能	主アクタ	散布車オペレーター

事前条件

操作支援システム、音声入力、気温計が正常に動作していること。

操作支援システムで、時間－KP 対応ファイルが正しく生成されていること。

外部記憶メディアが PC に接続されていること。

メインフロー

1．本システムを開始する：例外処理１

2．時間および気温、巡回情報を設定インターバル毎に作業ファイルに出力する。

3．本システムを停止する。

4．本システムは、操作支援システムで生成された時間－KP 対応ファイルを読み込み、時間をキーとして KP と気温および巡回情報を対応させる。

5．４での処理をもとに巡回地点の KP と対応を行い、巡回地点の情報のみを抜き出し、外部記憶メディアに保存する。

6．５の処理後、本システムを終了する。

7．外部記憶装置に保存されたファイルを、計画支援システムに登録する。

8．除雪作業監督職員または除雪業務委託者現場代理人は、計画支援システムを利用して、巡回記録を閲覧する。

事後条件

例外フロー

例外処理１：前回の巡回記録作業ファイルが存在している場合には、そのファイルを読み込んで前回の巡回情報を保持してから、システムを開始する。

(5) システムモデリング

本システムを概念的にモデリングした概念クラス図および、シーケンス図を以下に示す。

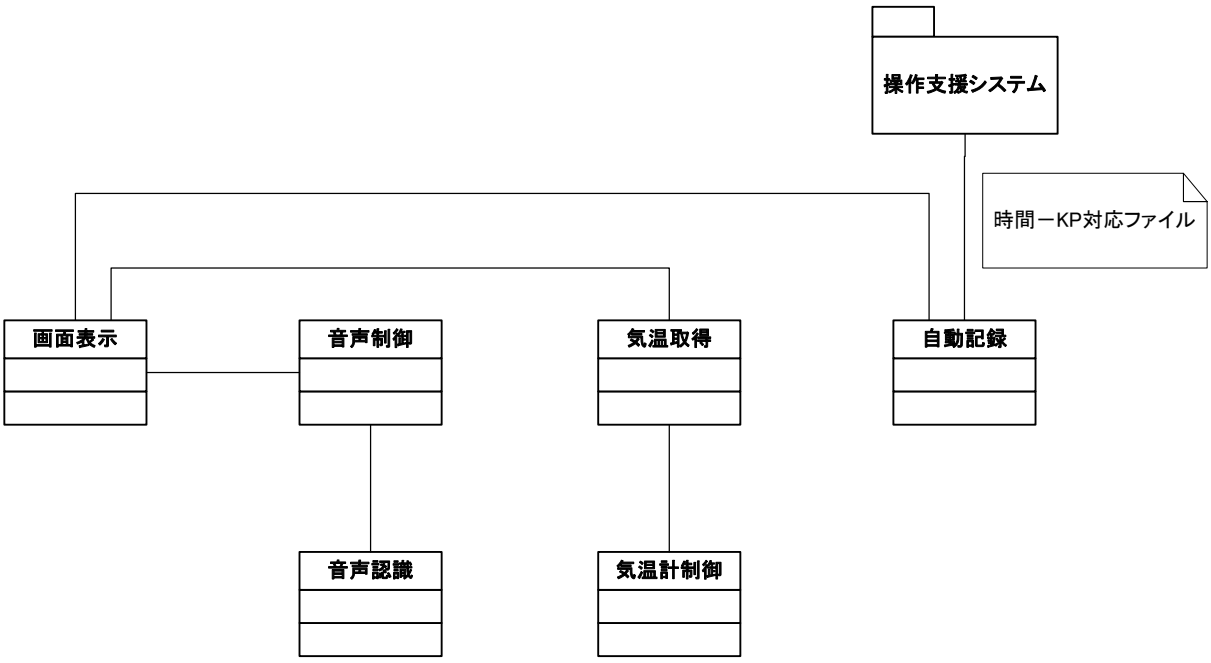


図 4-9　概念クラス図

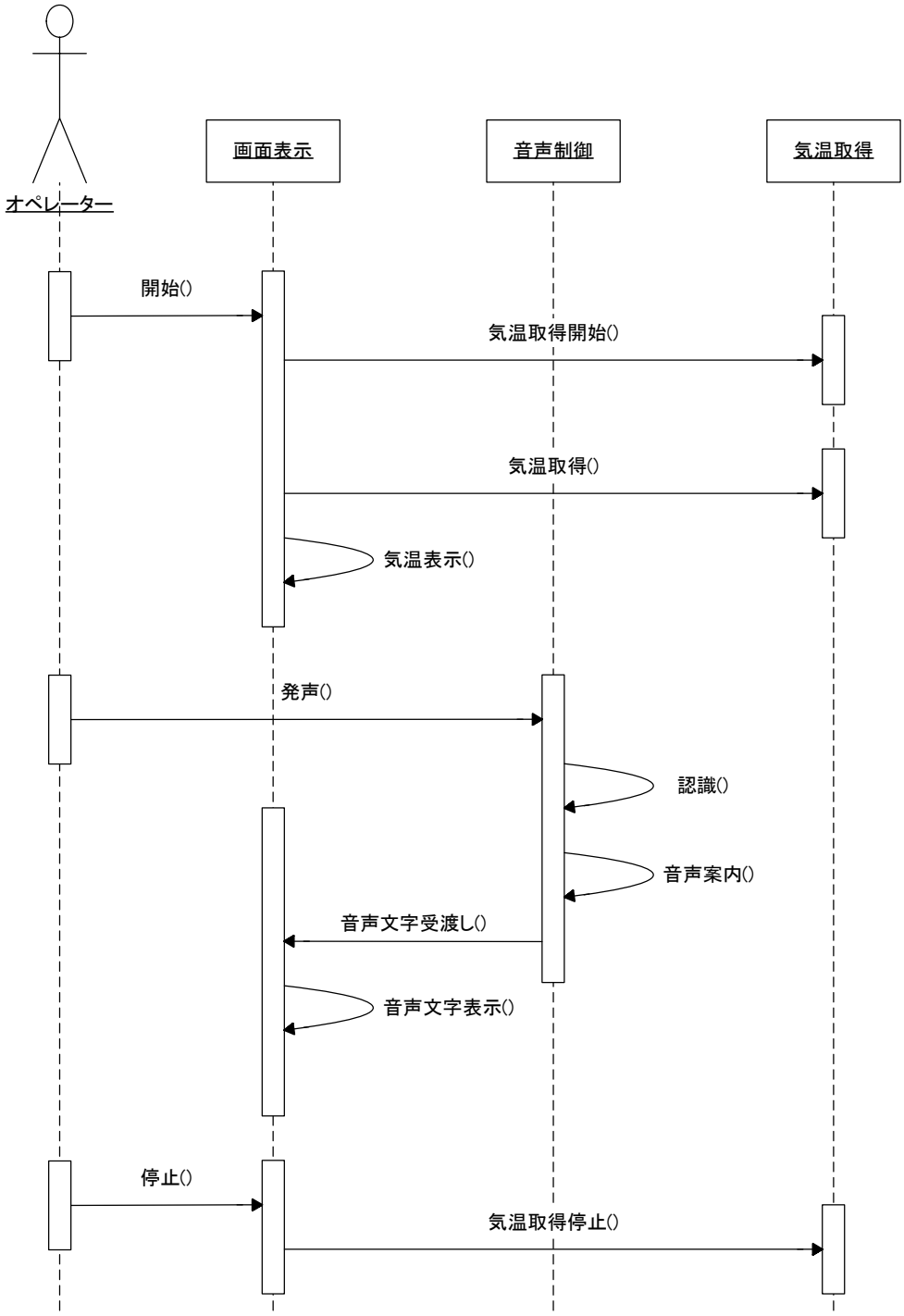


図 4-10　画面表示シーケンス図

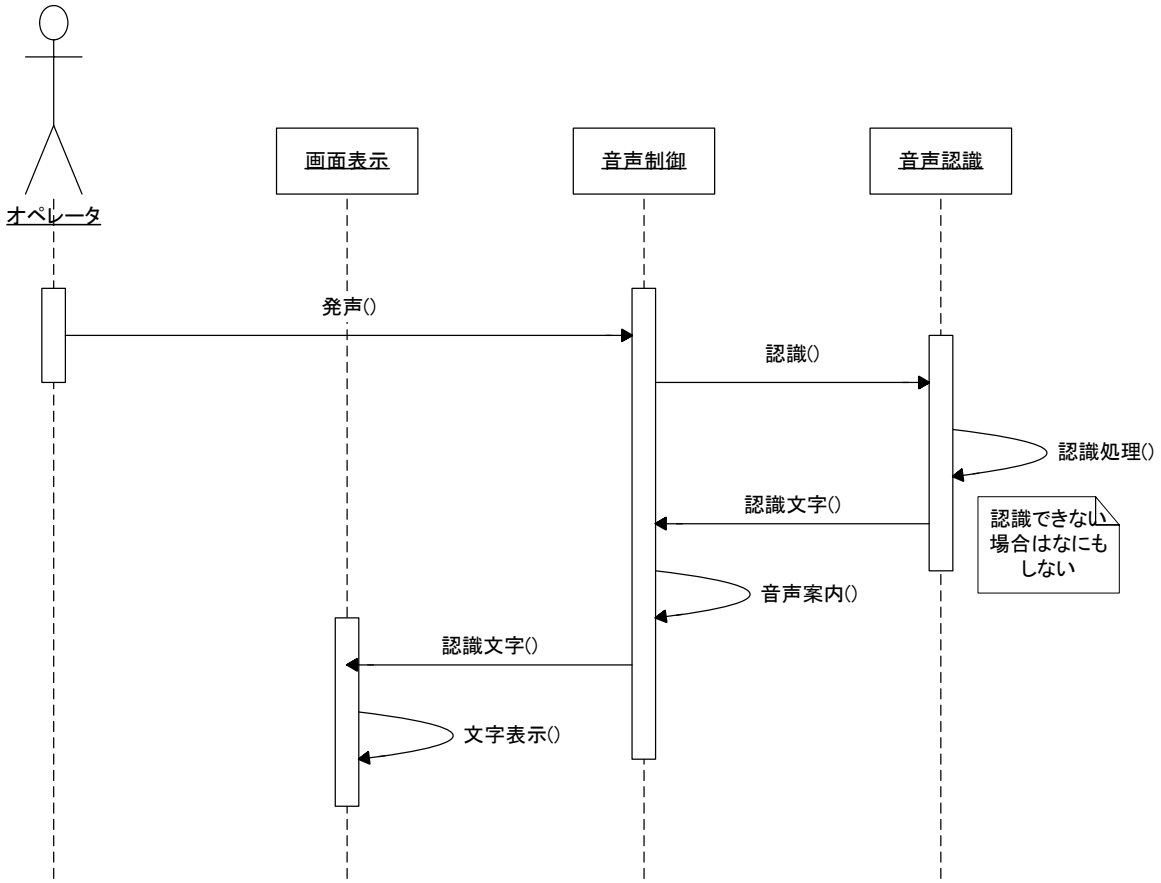


図 4-11　音声制御シーケンス図

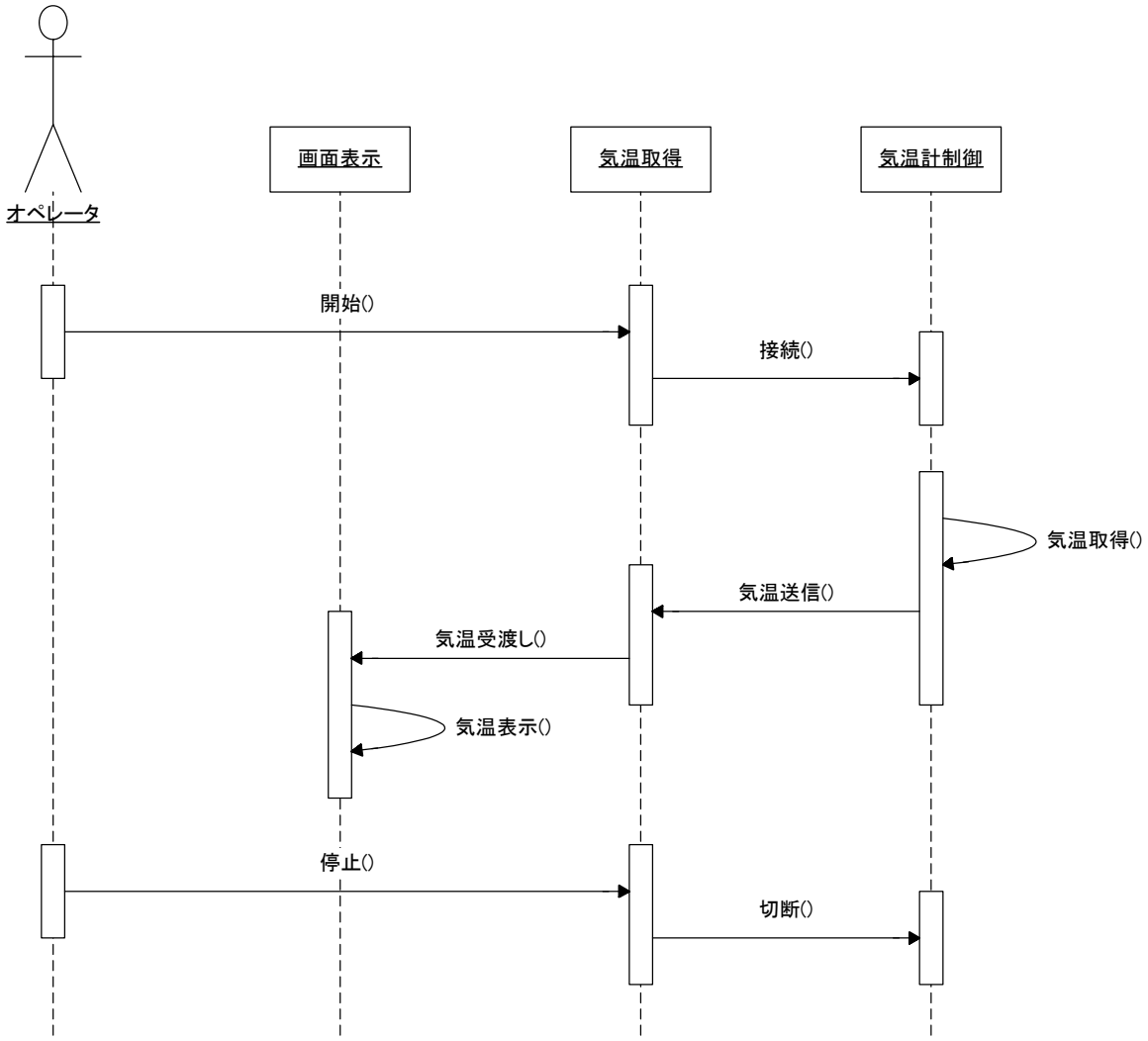


図 4-12　気温取得シーケンス図

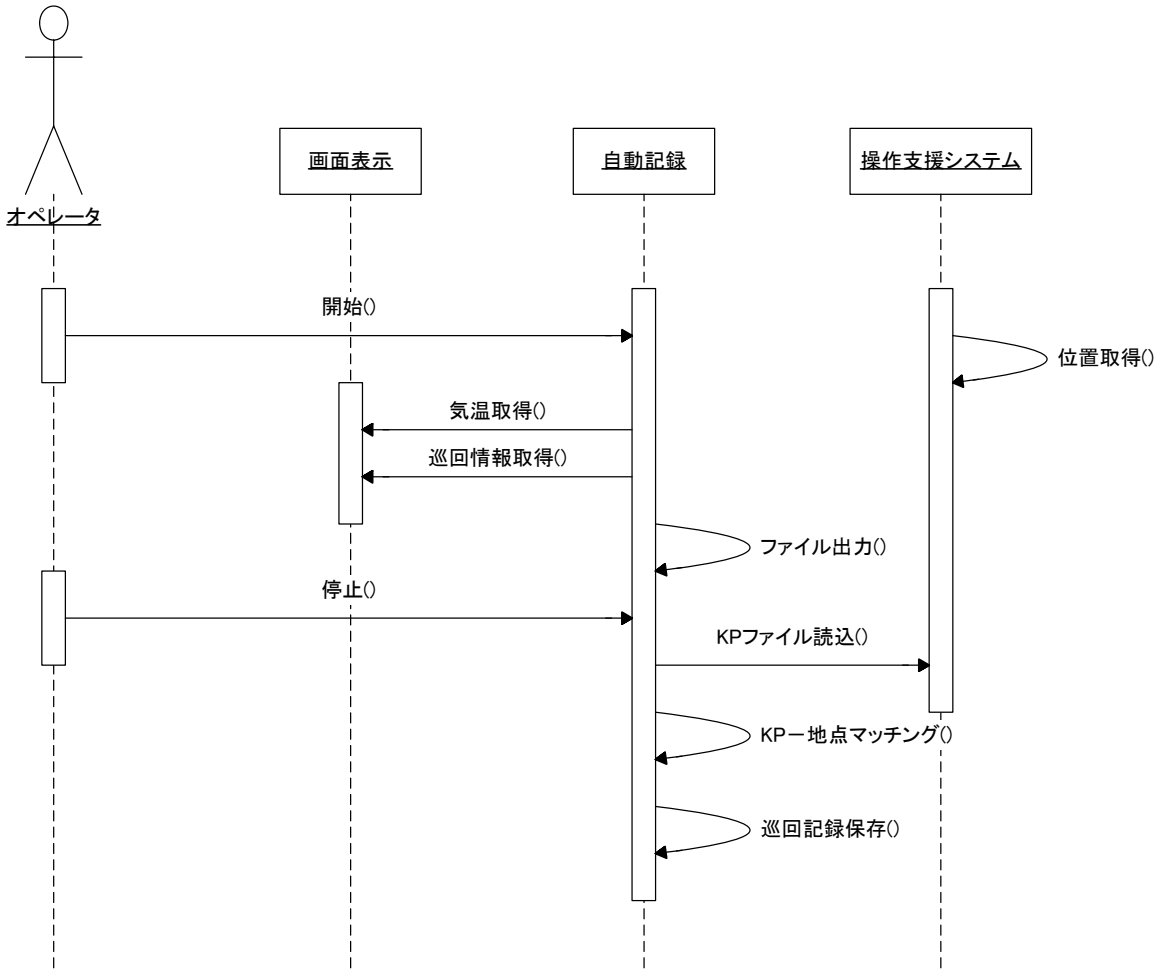


図 4-13　自動記録シーケンス図

(6) 画面設計

本システムにおける画面および画面内の表示の遷移図を以下に示す。

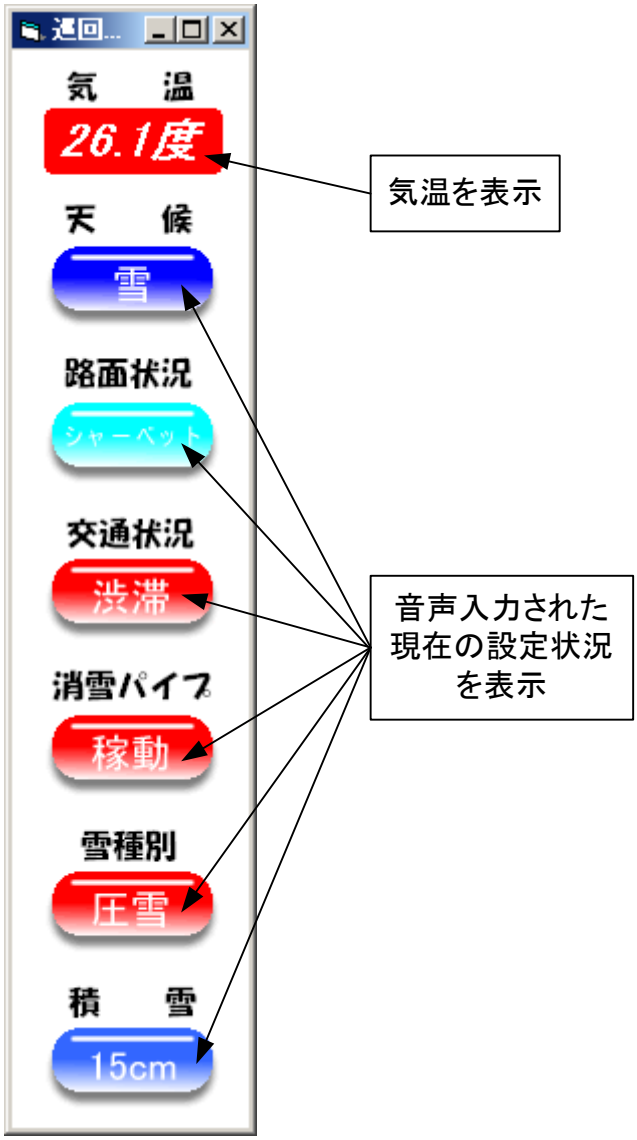


図 4-14　画面イメージ

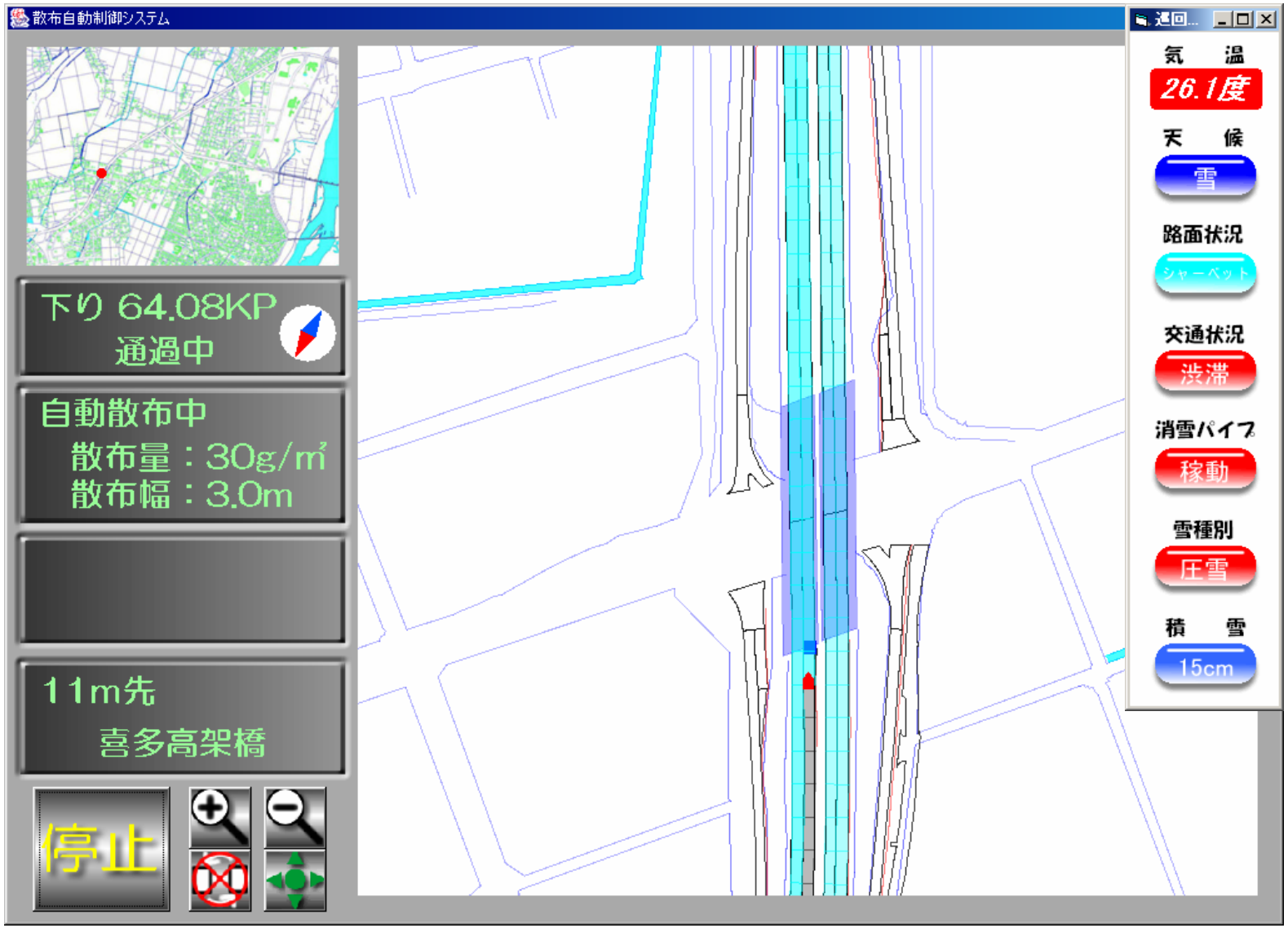


図 4-15　操作支援システム結合画面イメージ

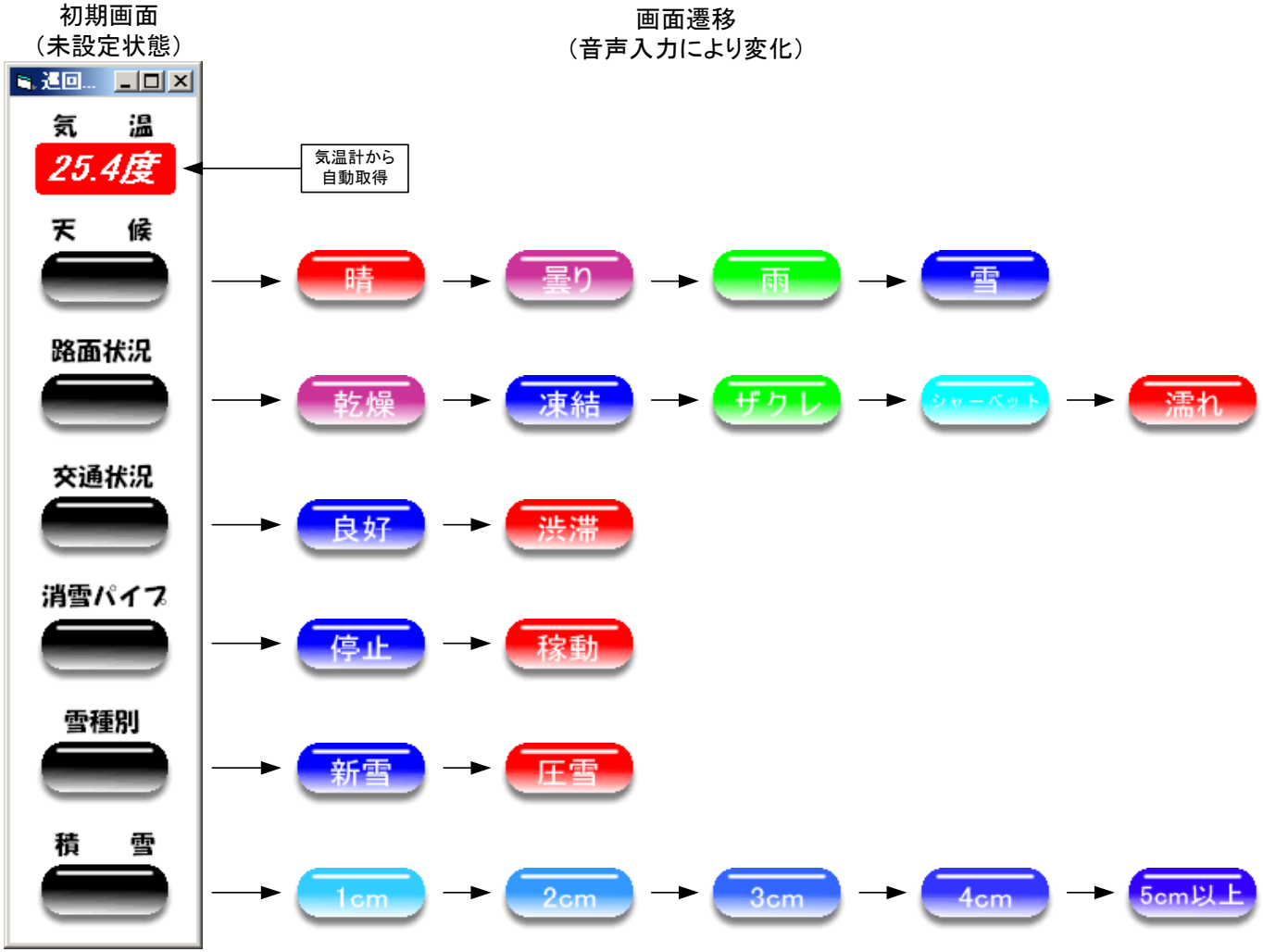

初期画面
（未設定状態）
画面遷移
（音声入力により変化）
気　温
25.4度
気温計から
自動取得
天　候
晴
曇り
雨
雪
路面状況
乾燥
凍結
ザクレ
シャーベット
濡れ
交通状況
良好
渋滞
消雪パイプ
停止
稼動
雪種別
新雪
圧雪
積　雪
1cm
2cm
3cm
4cm
5cm以上

図 4-16　画面遷移

4.2. 巡回支援システムの詳細設計

(1) 適用範囲

本節は、巡回支援システムにおける詳細設計についてのみ記載するものとする。

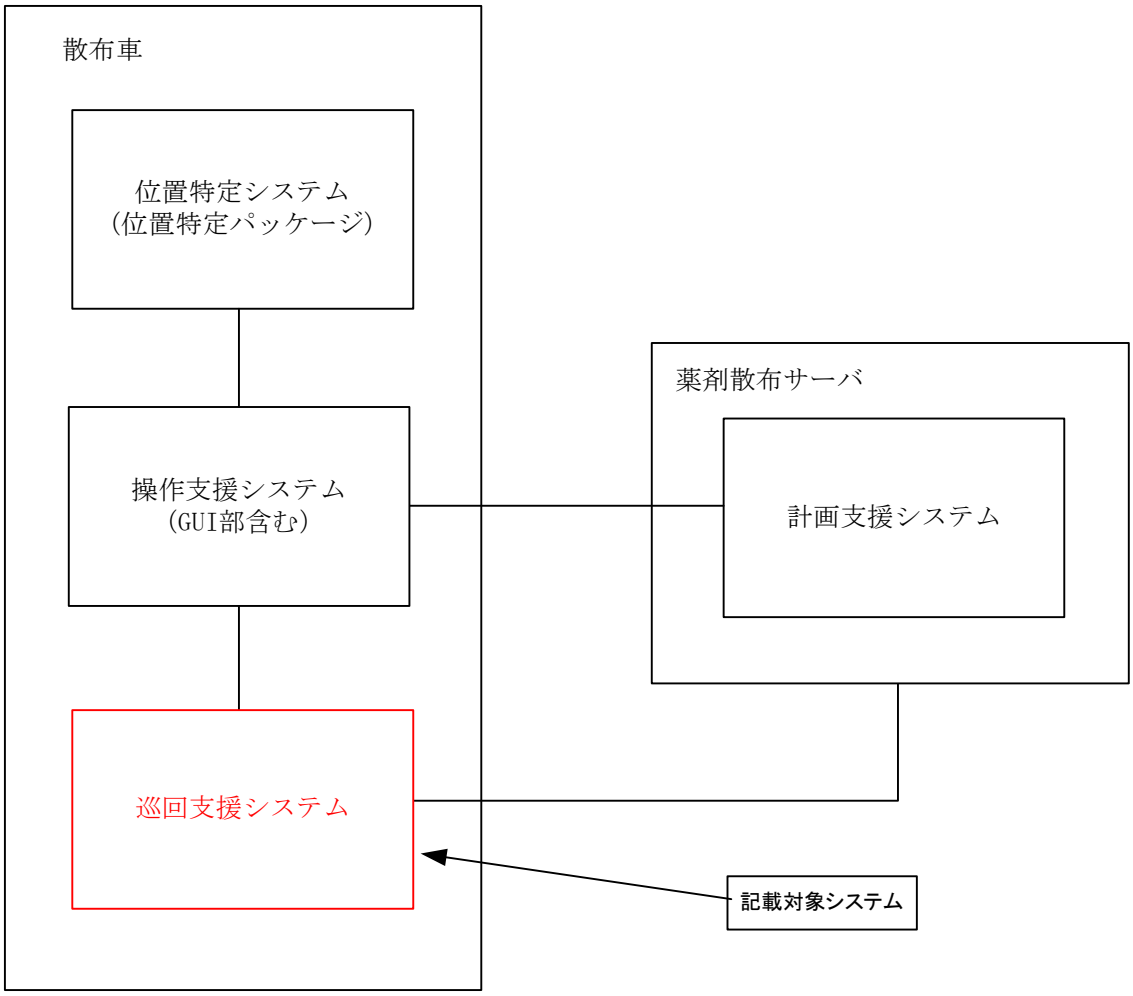


図 4-17　記載範囲

(2) ソフトウェア構成

本システムにおけるソフトウェアのクラス図、シーケンス図を以下に示す。

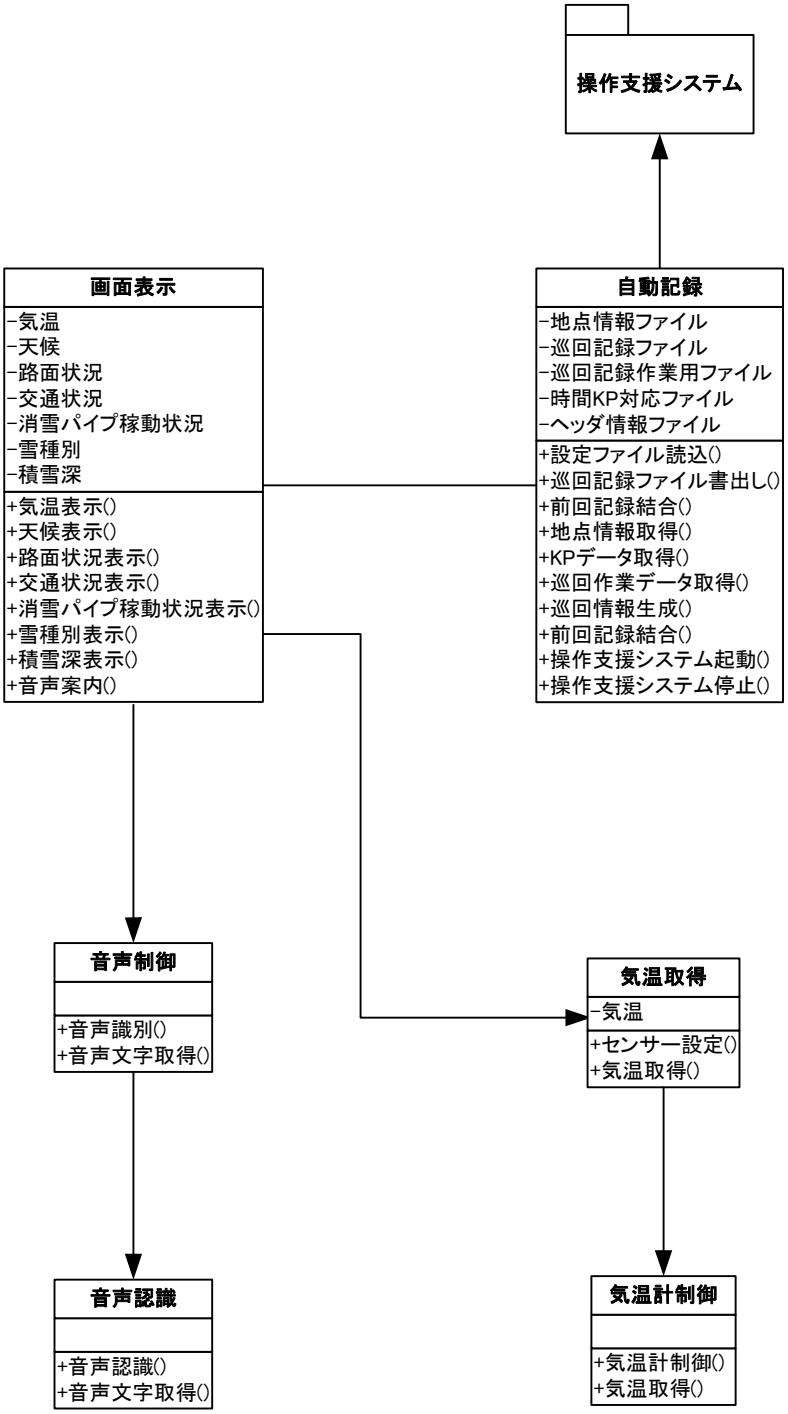


図 4-18　クラス図

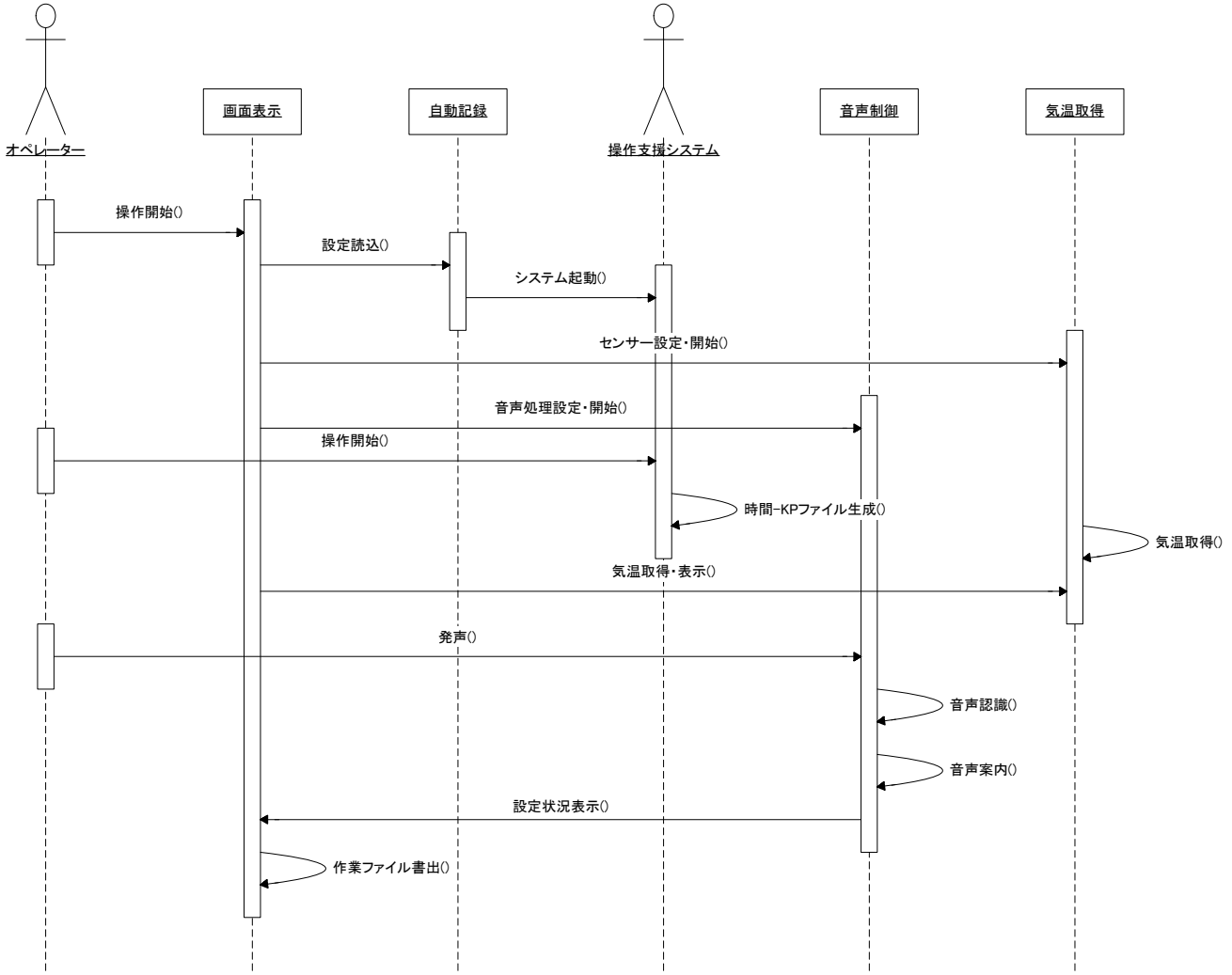


図 4-19　全体処理シーケンス図

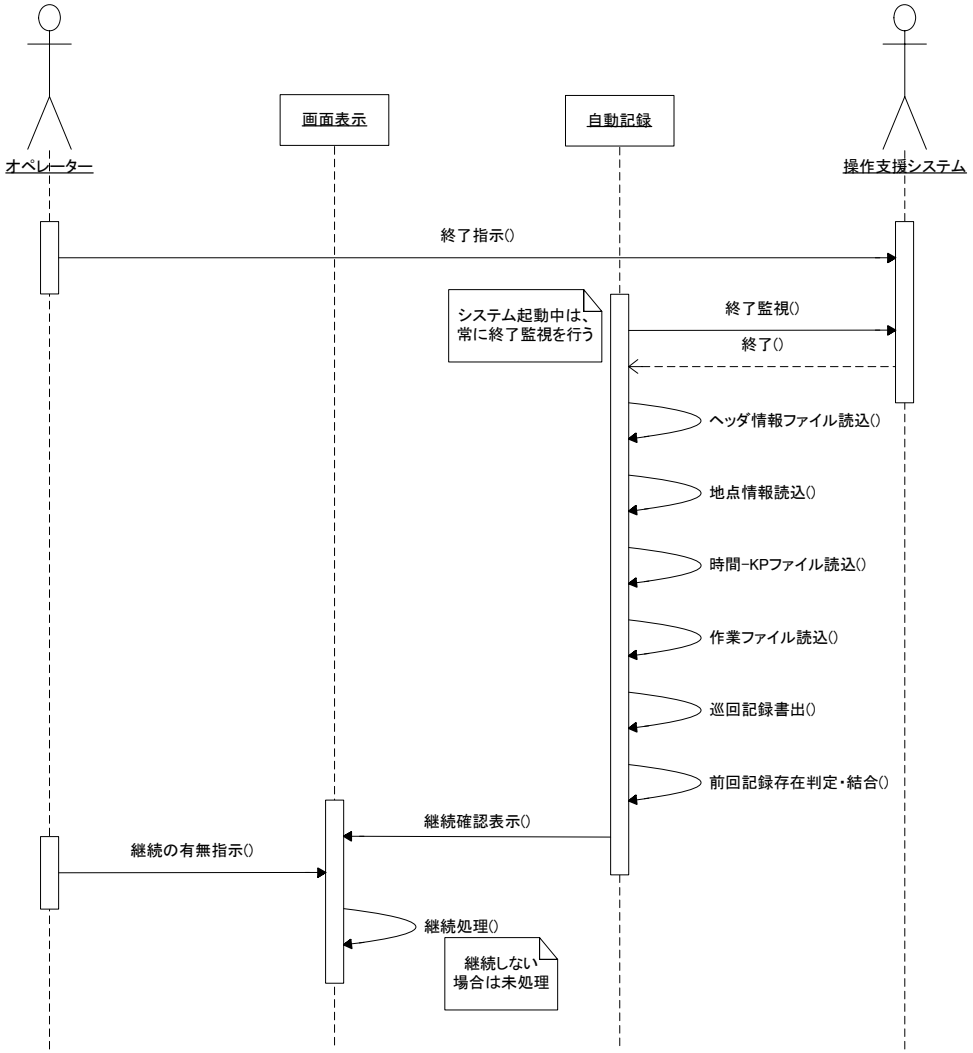


図 4-20　巡回記録生成シーケンス図

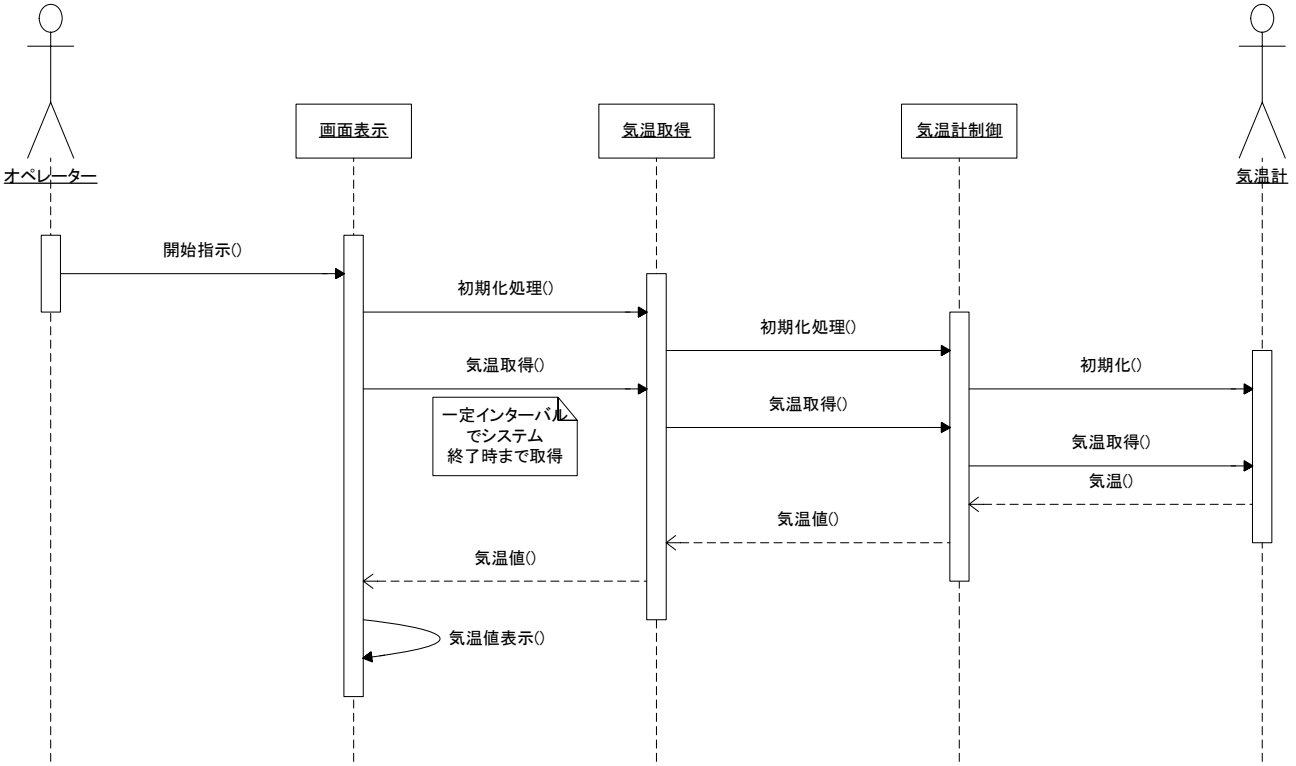


図 4-21　気温取得シーケンス図

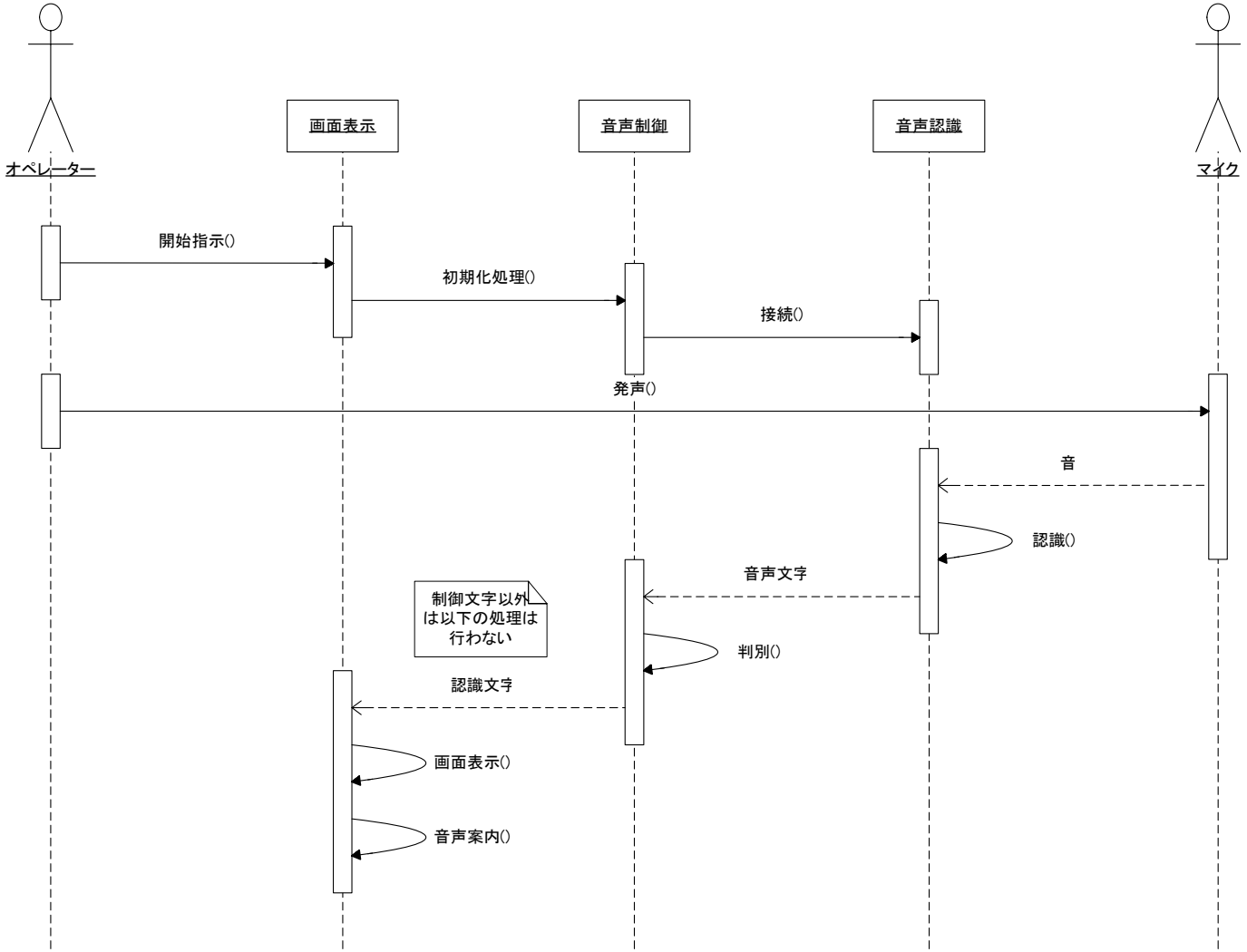


図 4-22　音声認識シーケンス図

表 4-5　画面表示クラスの機能

クラス名	画面表示

入力データ

設定ファイル

機能概要

本クラスは、全体コントロールと画面の表示を行うものであり、本システムのメインプログラムを含むものである。

(1) 設定読込指示

設定ファイルの読み込み指示をする。

(2) 気温計初期化指示

気温計の初期化および設定指示をする。

(3) 音声初期化指示

音声認識の初期化および設定をする。

(4) 操作支援システムの起動および終了監視

操作支援システムを起動を行い、終了したかどうかの監視を行う。

(5) 巡回記録作業ファイル作成

巡回記録作業用のデータファイルを作成する。

(6) 巡回記録作成指示

巡回記録の作成指示をする。

(7) 音声文字取得および音声案内

音声制御クラスより音声文字を受け取り、本システムで使用する文字の場合には、画面に表示し、さらに該当する音声ファイルを読み込んで再生する。

出力データ

巡回記録作業用ファイル

表 4-6　自動記録クラスの機能

クラス名	自動記録

入力データ

地点情報ファイル

時間－KP 対応ファイル

ヘッダ情報ファイル

巡回記録作業用ファイル

機能概要

本クラスは、巡回記録ファイルを生成するものである。

(1) 設定読込

設定ファイルの読込みを行う。

(2) 操作支援システムの起動、停止判定

操作支援システムの起動と停止したかどうかの判定を行う。

(3) 地点情報の取得

地点情報ファイルの読込みを行う。

(4) 時間－KP と巡回情報の対応

時間－KP ファイルと巡回記録作業ファイルを読込み、時間をキーとして KP と巡回情報を対応づける。

(5) 巡回記録作成

巡回地点と巡回情報を KP をキーに対応させ、巡回記録ファイルを作成する。

(6) 前回作業記録結合

前回の作業記録が残っている場合は、今回の巡回記録との結合を行う。

出力データ

巡回記録ファイル

表 4-7　気温取得クラスの機能

クラス名	気温取得

入力データ

なし

機能概要

本クラスは、気温取得のコントロールを行うものである。

(1) センサー設定

接続されている COM ポート番号を調べて、気温計の接続されている COM ポートのオープン処理を行い、気温データの取得が可能となるように気温計制御クラスに指示を出す。

(2) 気温取得

気温計制御クラスに指示出して、気温を取得する。

出力データ

気温値

表 4-8　気温計制御クラスの機能

クラス名	気温計制御

入力データ

なし

機能概要

本クラスは、気温計から直接気温データを取得するものである。

(1) 接続確認

気温計が接続されているか確認を行う。

(2) 該当する気温計の機器情報を返す

管理番号を指定して、接続されている気温計の COM ポート番号を取得し、気温計との接続を行う。

(3) 現在値取得

現在の気温値の取得を行う。

出力データ

気温値

表 4-9　音声制御クラスの機能

クラス名	音声制御

入力データ

音声認識文字

機能概要

本クラスは、音声認識クラスによって得られた音声文字から本システムで必要な文字を抽出し、抽出された場合にイベントを発生して画面表示クラスに認識を知らせる。

(1) 音声識別

音声認識クラスからの音声文字の識別を行う。

(2) 音声文字取得

音声が識別された場合に、識別されたイベントを発生し識別された文字を返す。

出力データ

音声識別文字

表 4-10　音声認識クラスの機能

クラス名	音声認識

入力データ

音声認識文字ファイル

音声（マイクより人間が話すことにより入力）

機能概要

本クラスは、マイクから入力された音声を認識し、文字に変換するものである。

(1) 音声認識

マイクからの音声を認識して、文字に変換する。

(2) 音声文字取得

音声が認識された場合に、認識された文字を返す。

出力データ

音声認識文字

(3) ファイル設計

本システムで使用するファイルは、以下のとおりである。

表 4-11　ファイル一覧

ファイル名	形式	I/O	説　　明
M_地点情報	CSV	I	帳票（道路巡回日報）に出力する地点及び消雪パイプ位置を記録したファイル。
巡回記録	CSV	O	巡回結果を記録したファイル。「散布支援システム」へのアップロード用として使用。
巡回記録作業用	CSV	I/O	巡回記録の作業用ファイル。
時間 KP 対応	CSV	I	時刻と通過 KP を対応させたファイル。散布制御システムにより生成される。
巡回情報	TXT	I/O	巡回情報として散布制御システムがシステム終了時に生成するファイル。
設定情報	INI	I	本システムの設定情報ファイル。

表 4-12　ファイルフォーマット

ファイル名	M_地点情報	形式	CSV
説明	帳票（道路巡回日報）に出力する地点及び消雪パイプ位置を記録したファイル。このファイルを基に、各地点の通過時刻、天候、気温、路面状況、交通状況、消雪パイプの稼動状況を出力する。		

フォーマット

1 行 1 地点とし、以下の項目をカンマで区切った CSV 形式とする。帳票に出力する地点数（上り、下りを別とする）分及び消雪パイプの箇所数（上り、下りを別とする）分必要。

- 地点または消雪パイプの別（地点、消雪）
- KP(km)
- 地点名称 または 消雪パイプ（箇所）名称

●区別が“地点”の場合

- 区間開始 KP
- 区間終了 KP

※空行または1カラム目が;の場合コメントとみなす。

表 4-13　ファイルフォーマット

ファイル名	巡回記録	形式	CSV
説明	巡回記録ファイル。巡回後に、「M_地点情報」ファイルに記録されている箇所の巡回結果を記録したファイル。「散布支援システム」へのアップロード用として使用。		

フォーマット

１行目：作業者名

２行目：標準散布パターン名

３行目：開始日時

４行目：終了日時

５行目以降：１行１地点とし、以下の項目をカンマで区切った CSV 形式とする。

●区別が「地点」の場合

- “地点“
- KP(km)
- “上り” または “下り”
- 地点名称
- 通過時間
- 天候（晴、曇り、雨、雪）
- 気温
- 路面状況（乾燥、濡れ、凍結、ザクレ、シャーベット）
- 交通状況（良好、渋滞）
- 区間開始 KP
- 区間終了 KP

●区別が「消雪」の場合

- “消雪”
- KP(km)
- “上り” または “下り”
- 消雪設備（箇所）名称
- 通過時間
- 稼動状況（停止、稼動）

表 4-14　ファイルフォーマット

ファイル名	巡回記録作業用	形式	CSV
説明	巡回記録の作業用ファイル。気温、天候、路面状況、交通状況、消雪パイプの稼動状況を、一定周期（1分間隔）で出力。		

フォーマット

1 行 1 周期とし、以下の項目をカンマで区切った CSV 形式とする。

- 時間
- 気温
- 天候（晴、曇り、雨、雪）
- 路面状況（乾燥、濡れ、凍結、ザクレ、シャーベット）
- 交通状況（良好、渋滞）
- 消雪パイプの稼動状況（停止、稼動）

表 4-15　ファイルフォーマット

ファイル名	時間 KP 対応	形式	CSV
説明	時刻と通過 KP を対応させたファイル。散布制御システムにより生成されるファイル。このファイルと、「巡回記録作業用」ファイルを対応させて「巡回記録」ファイルを生成する。		

フォーマット

1 行 1 周期とし、以下の項目をカンマで区切った CSV 形式とする。

- 時間
- 車線 ID
- KP(km)

表 4-16　ファイルフォーマット

ファイル名	巡回情報	形式	TXT
説明	巡回情報として散布制御システムがシステム終了時に生成するファイル。このファイルを「巡回記録」ファイルのヘッダ情報として出力する。		

フォーマット

1行目：作業者名

2行目：標準散布パターン名

3行目：開始日時

4行目：終了日時

表 4-17　ファイルフォーマット

ファイル名	設定情報	形式	INI
説明	巡回支援システムの設定情報ファイル。		

フォーマット

１行目：出力インターバル（秒）

２行目：地点通過後、何秒後に巡回記録を取得するかを指定

３行目：地点情報ファイル名（入力用）

４行目：巡回記録ファイル名（出力用、散布支援システムアップロード用）

５行目：巡回記録作業用ファイル名（出力用）

６行目：散布制御システム側、時間 KP 対応ファイル名

７行目：散布制御システム側、巡回情報ファイル名

８行目：散布制御システム実行ファイル名

９行目：巡回記録作業継続用ファイル名

１０行目：音声認識雑音消去率（0-10）0 で雑音消去しない

※空行または１カラム目が；の場合コメントとみなす。

4.3. 巡回支援システムの実装設計

(1) 適用範囲

本節は、巡回支援システムにおける実装設計についてのみ記載するものとする。

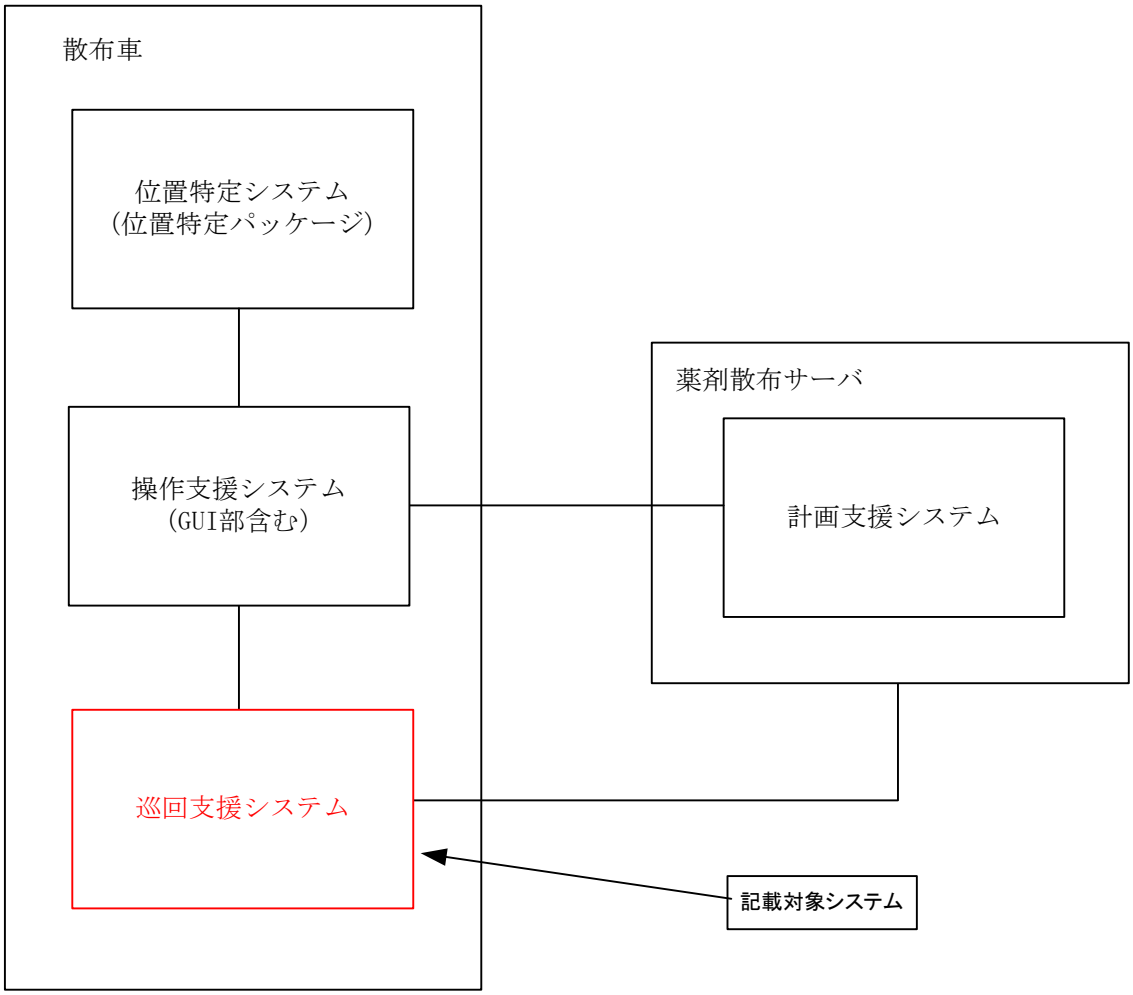


図 4-23　記載範囲

(2) システム構成

本システムのソフトウェア構成図およびハードウェア構成図を以下に示す。

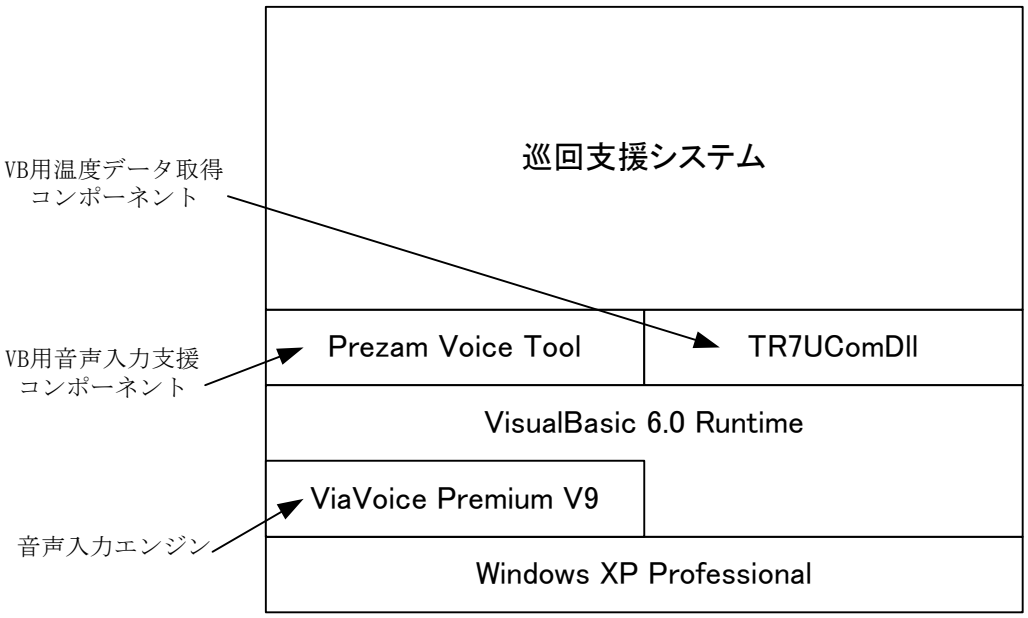


図 4-24　ソフトウェア構成

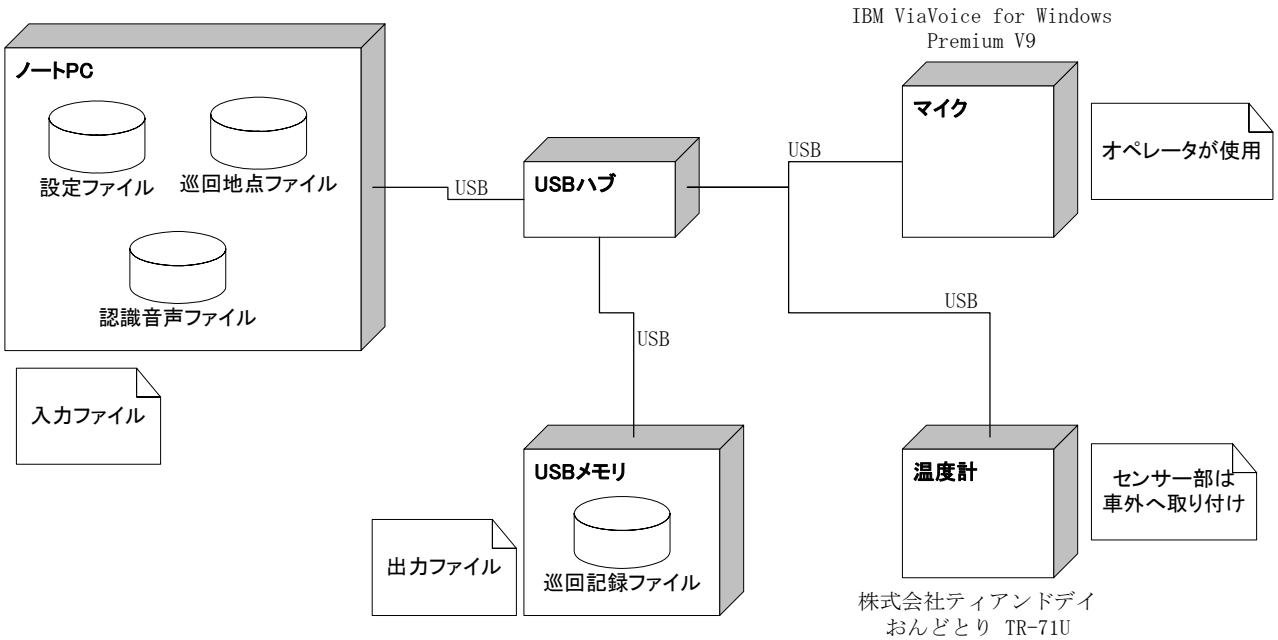


図 4-25　ハードウェアウェア構成

(3)　製品仕様

a） ViaVoice for

IBM ViaVoice for Windows Premium V9 の製品仕様を以下に示す。

表 4-18　音声入力製品仕様

	Pro USB 、PremiumCPU		
CPU	Pentium-II 333MHz 以上（互換プロセッサーを含む）*1		
OS （各日本語に対応）※1	Windows XP Professional Windows XP Home	Windows 2000 Professional (Service Pack2)	Windows Millennium iEdition Windows 98 Second Edition
必要メモリー※2	256MB 以上	96MB 以上	64MB 以上
ハードディスク	510MB 以上　*3、*7		
サウンド・カード	マイク入力端子付き Windows 互換 16 ビット・サウンド・カード*4 、*5 または USB ポート*6		
CD-ROM ドライブ	4 倍速以上（インストール時に必要）		
インターネット接続	インターネット閲覧ソフトと E メール・ソフトが必要な機能があります。 これらの設定は別途行ってください。		

*1 ViaVoice のホームページ「ボイスらんど」では各メーカー製パソコンでの ViaVoic e の稼働確認状況を公開しています。

*2 Windows XP では 128MB でも動作しますが、動作速度が遅くなる場合があります。なお、基本ソフトウェアにかかわらず、より高速の

プロセッサーを使用したり、メモリーを追加すると ViaVoic e の応答速度が向上します。

*3 インストール時の値です。この容量にボーナスパックに含まれているソフトやデータの容量は含まれていません。

*4 認識率はサウンド・カードの録音品質に大きく左右されます。

*5 サウンド・カードによっては、アダプターを接続する必要のあるものがあります。詳しくは同梱のマニュアルをご覧ください。

*6 US B ハブをお使いの場合は電源付きハブをお使いください。

*7 圧縮されたディスクにはインストールしないでください。

b） おんどとり TR-71U

株式会社ティアンドデイ おんどとり TR-71U の製品仕様を以下に示す。

図 4-26　温度センサー

表 4-19　温度計製品仕様

	TR-71U
測定要素	温度
チャンネル数	2 チャンネル(内蔵 1ch./外付けセンサ 2ch.から選択)
内蔵温度センサ	-10～60□
付属センサ	-40～110□
オプション 温度センサ	-60～155□□1
内蔵センサ 熱時定数	12 分
測定精度 （付属センサ）	平均±0.3□（-20～80□） 平均±0.5□（-40～-20/80～110□）
測定・表示分解能	0.1□
センサ	サーミスタ
記録間隔	1・2・5・10・15・20・30 秒・1・2・5・10・15・20・30・60 分 合計 15 通りから選択
記録容量	8000 データ×2 チャンネル
記録モード	エンドレスモード（記憶容量がいっぱいになると、先頭のデータに上書きして記録する） ワンタイムモード（記録容量がいっぱいになると、記録を停止する）
液晶表示	測定値・測定記録状態・電池寿命警告・測定範囲オーバー・記録データ量・単位
電源	単 3 アルカリ電池(LR6) 1 本
電池寿命 □2	約 1 年
データバックアップ	電池電圧低下時・スイッチ OFF 時
インターフェイス	USB 通信・シリアル通信（RS-232C）
通信速度	データ吸い上げ時 データフルで 1 台約 8 秒
本体寸法	H55×W78×D18mm
本体質量	約 62ｇ（単 3 アルカリ電池　1 本を含む）
本体動作環境	温度：-10～60□・湿度：90%RH 以下（結露しないこと）
付属センサ	TR-0106：TPE 樹脂被覆センサ　2 本
付属品	単 3 アルカリ電池（LR6）　1 本・USB 通信ケーブル 1 本（US-15C）　1 本 取扱説明書（保証書）一式・ソフトウェア一式

□1TR-71U 用の温度センサの温度測定範囲は 2 種類あります。詳しくはセンサの仕様をご覧ください。
□2 電池寿命は、測定環境、記録間隔、使用電池の性能により異なります。

(4) モジュール設計

a） モジュール一覧

モジュール	
Form1.frm	全体のコントロール及び画面の表示を行う
basWindow.bas	画面を最前面に出す設定及び日付ファイル名の生成
getExitProcess.bas	外部プロセスの実行状態を監視する
getTemperature.bas	温度計の制御を行う
outJunkaiFile.bas	巡回記録の生成を行う

なお、音声制御については市販コンポーネントである Prezam Voice Tool fro IBM ViaVoice を使用し、気温取得については、おんどとり付属のコンポーネント TdTr7Com DLL を使用する。

Prezam Voice Tool fro IBM ViaVoice および TdTr7Com DLL の仕様については、別紙資料に示すものとする。

表 4-20　共通変数仕様

モジュール名	Form1.frm
変数名	説　明
valWeather	天候設定値
valRoad	路面状況設定値
valTraffic	交通状況設定値
valSnow	雪種別設定値
valDepth	積雪設定値
valPipe	消雪パイプ設定値
flag	音声認識フラグ true:認識（音声未認識時音声案内用：未使用）
level	雑音判定フラグ、Via Voice の設定値(0-10)
MyPath	プログラムの実行パス格納変数

表 4-21　メソッド仕様

module name	Form1.frm	method name	Form_Load()

input data

なし

function

プログラム起動時の初期設定を行い、温度センサーや音声認識を開始する。

- 設定値の初期化を行う。
- デフォルトパスを取得する。
- 初期設定ファイルの読込を指示する。
- 気温センサー現在接続数の取得と接続を行う。
- 散布制御システムを起動する。
- ウィンドウを最前面に表示する。
- 巡回記録作業ファイルをオープンする。
- 音声制御モジュールの初期化を行い、音声認識を可能にする。
- 巡回記録作業ファイルに出力する時間間隔を設定する。
- 散布制御システムが終了したかどうかを確認する時間間隔を設定する。

output data

なし

表 4-22　メソッド仕様

module name	Form1.frm	method name	Timer1_Timer()

input data

なし

function

タイムインターバル毎に呼び出され、以下の処理を行う。

- 気温計から気温データを取得する。
- 気温値を画面表示変数にセットする。
- 巡回記録作業ファイルに現在の設定値を書き出す。

output data

なし

表 4-23　メソッド仕様

module name	Form1.frm	method name	Timer2_Timer()

input data

なし

function

設定インターバル毎に操作支援システムの終了監視を行い、終了した場合にシステム全体の終了処理を指示する。

- 操作支援システムの終了状態を受け取る。
- 操作支援システムが終了していた場合には、以下の処理を行う。
- タイマーを OFF にする。
- プロセスハンドルのクローズを指示する。
- 巡回記録作業ファイルのクローズを行う
- 巡回履歴の作成を指示する
- 作業ファイル名を変更する
- 音声認識エンジンの終了処理を指示する。
- サウンド再生の終了処理を行う。
- システム終了を行う。

output data

なし

表 4-24　メソッド仕様

module name	Form1.frm	method name	chWorkFileName()

input data

なし

function

システム終了時に巡回を続けるかどうかのダイアログを表示し、巡回を続ける場合には、巡回記録ファイルを前回巡回記録ファイル名に変更する。

- 時間－KP ファイルを日付付きファイル名に変更する。
- 巡回記録継続確認用のダイアログボックスを表示する。
- 巡回記録を継続する場合は、巡回記録ファイルを前回巡回記録ファイル名に変更する。

output data

なし

表 4-25　メソッド仕様

module name	Form1.frm	method name	SpeechCtrl1_OnPhraseRecognized(

input data

○ ByVal Phrase As String　：　音声認識文字

function

音声が認識イベントを取得し、音声認識文字毎に処理を振り分ける。

音声認識文字が、以下の時、画面表示文字の設定や音声案内の指示を行う。

- 「晴」、「曇り」、「雨」、「雪」
- 「乾燥」、「凍結」、「ザクレ」、「シャーベット」、「濡れ」
- 「良好」、「渋滞」
- 「停止」、「稼動」
- 「新雪」、「圧雪」
- 「なし」、「１ｃｍ」、「２ｃｍ」、「３ｃｍ」、「４ｃｍ」、「５ｃｍ」

output data

なし

表 4-26　メソッド仕様

module name	Form1.frm	method name	Sub11()

input data

なし

function

認識音声が「晴」の時の処理を行う。

- マイクを使用不可にする。
- 天候の設定値を「晴」にする。
- 画面の天候表示画像を「晴」にする。
- 音声案内ファイルに「晴」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-27　メソッド仕様

module name	Form1.frm	method name	Sub12()

input data

なし

function

認識音声が「曇り」の時の処理を行う。

- マイクを使用不可にする。
- 天候の設定値を「曇り」にする。
- 画面の天候表示画像を「曇り」にする。
- 音声案内ファイルに「曇り」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-28　メソッド仕様

module name	Form1.frm	method name	Sub13()

input data

なし

function

認識音声が「雨」の時の処理を行う。

- マイクを使用不可にする。
- 天候の設定値を「雨」にする。
- 画面の天候表示画像を「雨」にする。
- 音声案内ファイルに「雨」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-29　メソッド仕様

module name	Form1.frm	method name	Sub14()

input data

なし

function

認識音声が「雪」の時の処理を行う。

- マイクを使用不可にする。
- 天候の設定値を「雪」にする。
- 画面の天候表示画像を「雪」にする。
- 音声案内ファイルに「雪」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-30　メソッド仕様

module name	Form1.frm	method name	Sub21()

input data

なし

function

認識音声が「乾燥」の時の処理を行う。

- マイクを使用不可にする。
- 路面状況の設定値を「乾燥」にする。
- 画面の路面状況表示画像を「乾燥」にする。
- 音声案内ファイルに「乾燥」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-31　メソッド仕様

module name	Form1.frm	method name	Sub22()

input data

なし

function

認識音声が「凍結」の時の処理を行う。

- マイクを使用不可にする。
- 路面状況の設定値を「凍結」にする。
- 画面の路面状況表示画像を「凍結」にする。
- 音声案内ファイルに「凍結」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-32　メソッド仕様

module name	Form1.frm	method name	Sub23()

input data

なし

function

認識音声が「ザクレ」の時の処理を行う。

- マイクを使用不可にする。
- 路面状況の設定値を「ザクレ」にする。
- 画面の路面状況表示画像を「ザクレ」にする。
- 音声案内ファイルに「ザクレ」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-33　メソッド仕様

module name	Form1.frm	method name	Sub24()

input data

なし

function

認識音声が「シャーベット」の時の処理を行う。

- マイクを使用不可にする。
- 路面状況の設定値を「シャーベット」にする。
- 画面の路面状況表示画像を「シャーベット」にする。
- 音声案内ファイルに「シャーベット」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-34　メソッド仕様

module name	Form1.frm	method name	Sub25()

input data

なし

function

認識音声が「濡れ」の時の処理を行う。

- マイクを使用不可にする。
- 路面状況の設定値を「濡れ」にする。
- 画面の路面状況表示画像を「濡れ」にする。
- 音声案内ファイルに「濡れ」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-35　メソッド仕様

module name	Form1.frm	method name	Sub31()

input data

なし

function

認識音声が「良好」の時の処理を行う。

- マイクを使用不可にする。
- 交通状況の設定値を「良好」にする。
- 画面の交通状況表示画像を「良好」にする。
- 音声案内ファイルに「良好」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-36　メソッド仕様

module name	Form1.frm	method name	Sub32()

input data

なし

function

認識音声が「渋滞」の時の処理を行う。

- マイクを使用不可にする。
- 交通状況の設定値を「渋滞」にする。
- 画面の交通状況表示画像を「渋滞」にする。
- 音声案内ファイルに「渋滞」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-37　メソッド仕様

module name	Form1.frm	method name	Sub41()

input data

なし

function

認識音声が「停止」の時の処理を行う。

- マイクを使用不可にする。
- 消雪パイプ稼動状況の設定値を「停止」にする。
- 画面の消雪パイプ稼動状況表示画像を「停止」にする。
- 音声案内ファイルに「停止」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-38　メソッド仕様

module name	Form1.frm	method name	Sub42()

input data

なし

function

認識音声が「稼動」の時の処理を行う。

- マイクを使用不可にする。
- 消雪パイプ稼動状況の設定値を「稼動」にする。
- 画面の消雪パイプ稼動状況表示画像を「稼動」にする。
- 音声案内ファイルに「稼動」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-39　メソッド仕様

module name	Form1.frm	method name	Sub51()

input data

なし

function

認識音声が「新雪」の時の処理を行う。

- マイクを使用不可にする。
- 雪種別の設定値を「新雪」にする。
- 画面の雪種別表示画像を「新雪」にする。
- 音声案内ファイルに「新雪」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-40　メソッド仕様

module name	Form1.frm	method name	Sub52()

input data

なし

function

認識音声が「圧雪」の時の処理を行う。

- マイクを使用不可にする。
- 雪種別の設定値を「圧雪」にする。
- 画面の雪種別表示画像を「圧雪」にする。
- 音声案内ファイルに「圧雪」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-41　メソッド仕様

module name	Form1.frm	method name	Sub61()

input data

なし

function

認識音声が「なし」の時の処理を行う。

- マイクを使用不可にする。
- 積雪深の設定値を「なし」にする。
- 画面の積雪深画像を「なし」にする。
- 音声案内ファイルに「なし」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-42　メソッド仕様

module name	Form1.frm	method name	Sub62()

input data

なし

function

認識音声が「１センチ」の時の処理を行う。

- マイクを使用不可にする。
- 積雪深の設定値を「1」にする。
- 画面の積雪深画像を「１ｃｍ」にする。
- 音声案内ファイルに「１ｃｍ」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-43　メソッド仕様

module name	Form1.frm	method name	Sub63()

input data

なし

function

認識音声が「２センチ」の時の処理を行う。

- マイクを使用不可にする。
- 積雪深の設定値を「2」にする。
- 画面の積雪深画像を「２ｃｍ」にする。
- 音声案内ファイルに「２ｃｍ」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-44　メソッド仕様

module name	Form1.frm	method name	Sub64()

input data

なし

function

認識音声が「３センチ」の時の処理を行う。

- マイクを使用不可にする。
- 積雪深の設定値を「3」にする。
- 画面の積雪深画像を「３ｃｍ」にする。
- 音声案内ファイルに「３ｃｍ」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-45　メソッド仕様

module name	Form1.frm	method name	Sub65()

input data

なし

function

認識音声が「4 センチ」の時の処理を行う。

- マイクを使用不可にする。
- 積雪深の設定値を「4」にする。
- 画面の積雪深画像を「４ｃｍ」にする。
- 音声案内ファイルに「４ｃｍ」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-46　メソッド仕様

module name	Form1.frm	method name	Sub66()

input data

なし

function

認識音声が「5 センチ」の時の処理を行う。

- マイクを使用不可にする。
- 積雪深の設定値を「5」にする。
- 画面の積雪深画像を「５ｃｍ以上」にする。
- 音声案内ファイルに「５ｃｍ以上」を指定して、音声案内を指示する。
- マイクを使用可にする。

output data

なし

表 4-47　メソッド仕様

module name	Form1.frm	method name	Sound0n()

input data

○filekind As String　：　音声ファイル名

function

指定された音声ファイルを読み込んで、音声案内を行う。

- 音声ファイルに実行パスを指定して、絶対ファイル名を生成する。
- 音声ファイル読み込んで再生する。

output data

なし

表 4-48　共通変数仕様

モジュール名	BasWindow.bas
変数名	説　明
SetWindowPos	"user32.dll"の呼び出し宣言
HWND_TOPMOST	定数、値：&HFFFF、最前面に表示する設定用の定数
SWP_NOSIZE	定数、値：&H1、サイズを変更しない設定用の定数
SWP_NOMOVE	定数、値：&H2、位置を変更しない設定用の定数

表 4-49　メソッド仕様

module name	BasWindow.bas	method name	SetTopMost ()

input data

○hWnd As Long　:　WindowID

function

ウインドウを最前面に表示する指示を行う。

・ “user32.dll”の SetWindowPos()を呼び出す。

output data

なし

表 4-50　メソッド仕様

module name	BasWindow.bas	method name	NowFileName()

input data

○baseFile As String　：　日付を付加する前のファイル名

function

日付を付加したファイル名を返す。

- 日付付加前のファイルから、最後の”.”でファイル名を分解する。
- 現在日時を″yyyymmddhhmmss″形式に変換する。
- 分解された”.”より前のファイル名と現在日時を結合する。
- 現在日時と結合したファイル名と”.”を含む分解された”.”以降のファイル名と結合する。
- 日付と結合したファイル名を返す。

output data

○NowFileName As String　：　日付を付加したファイル名

表 4-51　共通変数仕様

モジュール名	getExitProcess.bas
変数名	説　明
OpenProcess	"kernel32.dll"の呼び出し宣言
GetExitCodeProcess	"kernel32.dll"の呼び出し宣言
CloseHandle	"kernel32.dll"の呼び出し宣言
PROCESS_QUERY_INFORMATION	定数、値：&H400&、OpenProcess()の設定値
STILL_ACTIVE	定数、値：&H103&、OpenProcess()の設定値
lngProcHandle	指定アプリケーションのタスク ID

表 4-52　メソッド仕様

module name	getExitProcess.bas	method name	runProcess()

input data

○runFileName As String　：　実行プログラム名

function

外部プログラム（操作支援システム）の実行を開始する。

- 外部プログラムを実行する。
- 起動した指定アプリケーションのプロセスハンドルを取得する。

output data

なし

表 4-53　メソッド仕様

module name	getExitProcess.bas	method name	thisProcess()

input data

なし

function

外部プログラム（操作支援システム）が終了したかどうかを判定する。

- プロセスの終了状態を取得する。
- プロセスが終了している場合には False を、実行中の場合には True を返す。

output data

thisProcess() As Boolean　:　True->実行中、False->終了

表 4-54　メソッド仕様

module name	getExitProcess.bas	method name	closeProcess()

input data

なし

function

実行プロセスのハンドルをクローズする。

・ CloseHandle()を呼び出して、実行プロセスのハンドルをクローズする。

output data

なし

表 4-55　共通変数仕様

モジュール名	getTemperature.bas
変数名	説　明
roop	温度センサー接続数
SerialNo	接続シリアルポート番号
macCode	機種コード
macName	機種名
useSencer	温度センサー使用状況
TdTr7UCom_IsConnect	"TdTr7UCom.dll"の呼び出し宣言
TdTr7UCom_GetConctMachInfo	"TdTr7UCom.dll"の呼び出し宣言
TdTr7UCom_GetCurrent	"TdTr7UCom.dll"の呼び出し宣言

表 4-56　メソッド仕様

module name	getTemperature.bas	method name	NowConnect()

input data

なし

function

温度センサーの接続確認と接続数を設定する。

- 温度センサーの接続確認を行う。
- 温度センサーが接続されていない場合は、確認ダイアログを表示する。
- 確認ダイアログで「はい」の時は、再度温度センサーの接続確認を行う。
- 確認ダイアログで「いいえ」の時は、温度センサーを使用しないので、温度センサー使用状況変数に未使用（False）をセットする。
- 確認ダイアログで「キャンセル」の時は、システムを終了する。
- 温度センサーが接続されている時は、温度センサー使用状況変数に使用（True）をセットする。
- 温度センサー接続確認時に返された、温度センサー接続数を温度センサー接続数変数にセットする。

output data

なし

表 4-57　メソッド仕様

module name	getTemperature.bas	method name	MacInfo()

input data

なし

function

温度センサーの接続情報を取得する。

- 温度センサー未使用時は、リターンする。
- 温度センサー接続数分ループし、温度センサーの接続シリアル番号、機種コード、機種名を格納変数にセットする。
- 接続シリアル番号は８桁の整数値とする。

output data

なし

表 4-58　メソッド仕様

module name	getTemperature.bas	method name	getTemp()

input data

なし

function

温度センサーから現在の温度値を取得する。

- 温度センサー未使用時は、温度値に”999.9”をセットする。
- 格納変数にセットされている温度センサー接続シリアル番号を用いて、温度センサーから温度値を取得する。
- チャンネル１の温度計から温度値が取得できた場合は、チャンネル１の温度を変数にセットする。
- チャンネル２の温度計から温度値が取得できた場合は、チャンネル２の温度を変数にセットする。
- チャンネル１とチャンネル２の温度値が取得できた場合は２つの平均値を温度値としてセットする。
- チャンネル１の温度値が取得でき、チャンネル２の温度値が取得できない場合は、チャンネル１の温度値を温度値としてセットする。
- チャンネル２の温度値が取得でき、チャンネル１の温度値が取得できない場合は、チャンネル２の温度値を温度値としてセットする。
- 温度値の書式は整数部３桁、小数部１桁とする（###.0)。

output data

getTemp() As String　：　現在の温度値

表 4-59　共通変数仕様

モジュール名	outJunkaiFile.bas
変数名	説　明
interval	出力インターバル（秒）
nextInterval	地点通過から巡回記録取得までのタイムラグ
chitenInfoFile	地点情報ファイル名
junkaiKirokuFile	巡回記録ファイル名
junkaiWorkFile	巡回記録作業用ファイル名
junkaiKpFile	散布制御システム側、時間 KP 対応ファイル名
junkaiInfoFile	散布制御システム側、巡回情報ファイル名
sanpuAppFile	操作支援システム、アプリケーション名
junkaiBeforeFile	巡回記録作業継続用ファイル名
voiceValue	音声認識雑音消去率
np	地点情報数
nk	KP 情報数
nj	巡回情報数
code()	地点情報：種別（地点 or 消雪）
KP()	地点情報：地点合わせ用 KP
houko()	地点情報：方向（上り or 下り）
chitenMei()	地点情報：地点名
tsukaJikan()	地点情報：通過時間
kion()	地点情報：気温
tenko()	地点情報：天候
romen()	地点情報：路面状態

表 4-60　共通変数仕様

モジュール名	outJunkaiFile.bas
変数名	説　明
kotsu()	地点情報：交通状況
yuki()	地点情報：雪種別
shin()	地点情報：積雪深
syopi()	地点情報：消雪パイプ稼動状態
viewKP()	地点情報：表示用 KP
kaishiKP()	地点情報：区間開始 KP
syuryoKP()	地点情報：区間終了 KP
sflg()	地点情報：車線フラグ
tim()	時間-KP 情報：時間
sid()	時間-KP 情報：車線 ID
kp0()	時間-KP 情報：KP
wtim()	巡回情報：時間
temp()	巡回情報：気温
weat()	巡回情報：天候
road()	巡回情報：路面状況
traf()	巡回情報：交通状況
snow()	巡回情報：雪種別
dept()	巡回情報：積雪深
pipe()	巡回情報：消雪パイプ稼動状態

表 4-61　メソッド仕様

module name	outJunkaiFile.bas	method name	GetIniFile()

input data

○fileName As String　：　設定ファイル名

function

設定ファイルの内容を読み込む。

・設定ファイルのフォーマットに従って設定情報を読み込む。

output data

なし

表 4-62　メソッド仕様

module name	outJunkaiFile.bas	method name	OutputJunkaiFile()

input data

なし

function

巡回記録ファイルを編集して出力する。

- 巡回記録ファイルをオープンする。
- 巡回情報ファイルを読み込んで、巡回記録ファイルに出力する。
- 地点情報ファイルの読込み指示を行う。
- 時間―KP 情報ファイルの読込み指示を行う。
- 巡回作業ファイルの読込み指示を行う。
- 巡回情報作成の指示を行う。
- 巡回記録を時間順に並び替える。
- 巡回記録を「地点」と「消雪」に分けて出力を行う。温度計を使用しない場合は、温度に"NONE"とセットして出力する。
- 巡回地点を通過しなかった場合（通過時間が設定されていない場合）は、巡回情報に"NONE"をセットして出力する。
- 巡回記録ファイルをクローズする。

output data

なし

表 4-63　メソッド仕様

module name	outJunkaiFile.bas	method name	AddBeforeFile()

input data

なし

function

前回巡回記録がある場合にファイルを結合する。

- 前回巡回記録ファイルがあるか判定を行う。
- 前回巡回記録ファイルがある場合は、前回巡回記録ファイルと今回巡回記録ファイルの結合指示を行う。
- 結合完了後、前回巡回記録ファイルを削除する。

output data

なし

表 4-64　メソッド仕様

module name	outJunkaiFile.bas	method name	GetInfo()

input data

なし

function

地点情報ファイルから地点情報を読み込み、変数配列に格納する。

- 変数配列の初期サイズを設定する。
- 地点情報ファイルをオープンする。
- 地点情報ファイルのフォーマットに従って、地点情報を読み込む。
- 読込みデータ数が変数配列の初期サイズを超えた場合は、配列サイズを拡大する。
- 地点情報ファイルをクローズする。
- 地点情報数を変数にセットする。

output data

なし

表 4-65　メソッド仕様

module name	outJunkaiFile.bas	method name	GetKP()

input data

なし

function

時間―KP 対応ファイルから時間―KP 対応情報を読み込み、変数配列に格納する。

- 変数配列の初期サイズを設定する。
- 時間―KP 対応ファイルをオープンする。
- 時間―KP 対応ファイルのフォーマットに従って、時間―KP 対応情報を読み込む。
- 読込みデータ数が変数配列の初期サイズを超えた場合は、配列サイズを拡大する。
- 時間―KP 対応ファイルをクローズする。
- 時間―KP 対応情報数を変数にセットする。

output data

なし

表 4-66　メソッド仕様

module name	outJunkaiFile.bas	method name	GetWork()

input data

なし

function

巡回記録作業ファイルから巡回記録情報を読み込み、変数配列に格納する。

- 変数配列の初期サイズを設定する。
- 巡回記録作業ファイルをオープンする。
- 巡回記録作業ファイルのフォーマットに従って、巡回記録情報を読み込む。
- 読込みデータ数が変数配列の初期サイズを超えた場合は、配列サイズを拡大する。
- 巡回記録作業ファイルをクローズする。
- 巡回記録情報数を変数にセットする。

output data

なし

表 4-67　メソッド仕様

module name	outJunkaiFile.bas	method name	SetJunkai()

input data

なし

function

時間―KP 情報と時間巡回記録情報を時間をキーとして対応させ、KP－巡回情報を変数配列に格納する。

- 巡回地点の KP から巡回地点の通過時間を変数配列にセットする。
- 通過時間における巡回情報を巡回地点における巡回情報として、変数配列にセットする。

output data

なし

表 4-68　メソッド仕様

module name	outJunkaiFile.bas	method name	SortJunkai()

input data

なし

function

巡回記録を帳票出力するため、時間順に並び替える。

・全ての巡回情報変数を巡回時間をキーとして、巡回時間の古い順に並び替えを行う。

output data

なし

表 4-69　メソッド仕様

module name	outJunkaiFile.bas	method name	JoinBeforeFile()

input data

○　fileName1 As String　：　前回巡回記録ファイル名

○　fileName2 As String　：　今回巡回記録ファイル名

function

前回巡回記録ファイルと今回巡回記録ファイルを結合する。

・　前回巡回記録ファイルをオープンする。
・　前回巡回記録ファイルを巡回記録ファイルフォーマットに従って、変数配列に読み込む。
・　前回巡回記録ファイルをクローズする。
・　今回巡回記録ファイルをオープンする。
・　今回巡回記録ファイルを巡回記録ファイルフォーマットに従って、変数配列に読み込む。
・　今回巡回記録ファイルをクローズする。
・　前回巡回記録と今回巡回記録を結合する。巡回情報は今回巡回記録情報を使用する。
・　前回巡回記録に値が入っており、今回巡回記録に値が入っていない場合は、前回巡回記録値を変数配列にセットする。
・　今回巡回記録に値が入っており、前回巡回記録に値が入っていない場合は、今回巡回記録値を変数配列にセットする。
・　前回巡回記録、今回巡回記録ともに値が入っている場合は、今回巡回記録値を変数配列にセットする。
・　前回巡回記録、今回巡回記録ともに値が入っていない場合は、”NONE”を変数配列にセ

output data

なし

4.4. 試験巡回支援システムの製作

製作した本システムのソースコードを別添資料に添付する。

4.5. 巡回支援システムの試験（単体）

(1) 試験概要

本試験成績書は、巡回支援システムについての単体試験結果を示すものである。単体試験方針としては、音声認識、画面の表示変化、生成される巡回記録データの妥当性の確認を行った。なお、ホワイトボックステストでは全体的な処理の確認が不可能なため、機能部分のテストを重点においたブラックボックステストを行うこととした。

(2) 試験結果

以下に、試験結果を示す。

表 4-70　単体試験結果

No	テスト項目	テスト概要	結果
1	気温表示	温度計の接続が確認できること	○
2		温度計のデータが正しく取得できること	○
3		温度計のデータが正しく画面に表示されること	○
4		温度計未接続時に画面に未使用と表示されること	○

表 4-71　単体試験結果

No	テスト項目	テスト概要	結果
5	音声入力	「はれ」と発音したときに「はれ」の音声案内が流れ、天候の画面表示が「晴」に変わること	○
6		「くもり」と発音したときに「くもり」の音声案内が流れ、天候の画面表示が「曇り」に変わること	○
7		「あめ」と発音したときに「あめ」の音声案内が流れ、天候の画面表示が「雨」に変わること	○
8		「ゆき」と発音したときに「ゆき」の音声案内が流れ、天候の画面表示が「雪」に変わること	○
9		「かんそー」と発音したときに「かんそー」の音声案内が流れ、路面状況の画面表示が「乾燥」に変わること	○
10		「とーけつ」と発音したときに「とーけつ」の音声案内が流れ、路面状況の画面表示が「凍結」に変わること	○
11		「ざくれ」と発音したときに「ざくれ」の音声案内が流れ、路面状況の画面表示が「ザクレ」に変わること	○
12		「しゃーべっと」と発音したときに「しゃーべっと」の音声案内が流れ、路面状況の画面表示が「シャーベット」に変わること	○
13		「ぬれ」と発音したときに「ぬれ」の音声案内が流れ、路面状況の画面表示が「濡れ」に変わること	○
14		「りょーこー」と発音したときに「りょーこー」の音声案内が流れ、交通状況の画面表示が「良好」に変わること	○
15		「じゅーたい」と発音したときに「じゅーたい」の音声案内が流れ、交通状況の画面表示が「渋滞」に変わること	○
16		「てーし」と発音したときに「てーし」の音声案内が流れ、消雪パイプの画面表示が「停止」に変わること	○
17		「かどー」と発音したときに「かどー」の音声案内が流れ、消雪パイプの画面表示が「稼動」に変わること	○
18		「」と発音したときに「」の音声案内が流れ、雪種別の画面表示が「」に変わること	○
19		「しんせつ」と発音したときに「しんせつ」の音声案内が流れ、雪種別の画面表示が「新雪」に変わること	○
20		「あっせつ」と発音したときに「あっせつ」の音声案内が流れ、雪種別の画面表示が「圧雪」に変わること	○
21		「なし」と発音したときに「なし」の音声案内が流れ、積雪の画面表示が「なし」に変わること	○
22		「いっせんち」と発音したときに「いっせんち」の音声案内が流れ、積雪の画面表示が「1cm」に変わること	○
23		「にせんち」と発音したときに「にせんち」の音声案内が流れ、積雪の画面表示が「2cm」に変わること	○
24		「さんせんち」と発音したときに「さんせんち」の音声案内が流れ、積雪の画面表示が「3cm」に変わること	○
25		「よんせんち」と発音したときに「よんせんち」の音声案内が流れ、積雪の画面表示が「4cm」に変わること	○
26		「ごせんち」と発音したときに「ごせんちいじょう」の音声案内が流れ、積雪の画面表示が「5cm以上」に変わること	○

表 4-72　単体試験結果

No	テスト項目	テスト概要	結果
27	操作支援システム との連動	操作支援システムが起動されること	○
28		巡回支援システム画面が最前面に表示されること	○
29		操作支援システム終了時に巡回支援システムが終了すること	○

表 4-73　単体試験結果

No	テスト項目	テスト概要	結果
30	巡回記録	指定巡回地点における時刻が正しく巡回記録登録用ファイルに記録されていること	○
31		指定巡回地点における気温が正しく巡回記録登録用ファイルに記録されていること	○
32		指定巡回地点における天候が正しく巡回記録登録用ファイルに記録されていること	○
33		指定巡回地点における路面状況が正しく巡回記録登録用ファイルに記録されていること	○
34		指定巡回地点における交通状況が正しく巡回記録登録用ファイルに記録されていること	○
35		指定巡回地点における消雪パイプ稼動状況が正しく巡回記録登録用ファイルに記録されていること	○
36		指定巡回地点における雪種別が正しく巡回記録登録用ファイルに記録されていること	○
37		指定巡回地点における積雪深が正しく巡回記録登録用ファイルに記録されていること	○

別添資料３　現地実証試験

1. 結合・総合動作試験

表 1-1　総合試験実施状況

No	実施日	実施箇所	確認概要	結合システム					備　考	
				位置	操作	散布車	計画	巡回		
1	10月17日	国総研試走路	散布車制御装置との結合	○	○	○			位置特定パッケージ調整中	表5-2
2	11月27日	国総研試走路	計画支援システムとの連携	○	○	○	○		位置特定パッケージ調整中	表5-3
3	12月5日	国総研試走路	車線変更、手動散布制御確認	○	○	○	○		位置特定パッケージ調整中	表5-4
4	12月10日	国総研試走路	位置特定パッケージ修正による車線変更、手動散布制御確認	○	○	○	○			表5-5
5	12月12日	国総研試走路	国総研最終総合試験	○	○	○	○			表5-6
6	12月17日	長岡宮本工区	動作確認、データチェック、パフォーマンス	○	○	○	○		12月18日まで実施	表5-7,8
7	2月6日	長岡宮本工区	巡回支援システム結合、パフォーマンス	○	○	○	○	○		表5-9

※結合システムは以下のとおり
位置：位置特定パッケージ
操作：操作支援システム
散布車：散布車制御装置
計画：計画支援システム
巡回：巡回支援システム

別 3-1-1

別 3-1-2

表 1-2　総合試験実施結果（10 月 17 日）

システム名称	確認項目	合否	備考
位置特定パッケージ	走行位置が正しく表示される	△	位置特定パッケージ調整中のため、位置がずれることあり
操作支援システム	散布箇所が画面に正しく表示される	○	
	散布量、幅が正しく表示される	○	
	散布変更箇所までの距離が表示される	○	
	散布済み箇所の色が変更される	○	
	散布変更時にビープ音が鳴る	○	
	散布履歴ファイルが正しく生成される	×	散布履歴ファイルが生成されない
散布車制御装置	散布制御装置に指示が行われている	○	散布車制御装置の表示と操作支援システムの表示が同じことを確認
	散布車の散布ゲートの円盤の回転が切換る	○	
	散布車の散布ゲートが上下する	○	

（特記事項）
位置特定パッケージ調整中のため、散布制御の変更箇所の確認については未確認
散布履歴ファイルが生成されるようにプログラムの修正必要

表 1-3　総合試験実施結果（11 月 27 日）

システム名称	確認項目	合否	備考
計画支援システム	散布計画データが正しく作成できる	○	
	作成した散布計画データが正しく表示される	○	
	作成した散布計画データを正常にダウンロードできる	○	
操作支援システム	散布計画ファイルを選択できる	○	
	作成した散布計画どおりに散布箇所が画面に表示される	○	
	作成した散布計画にしたがって散布制御が行われる	○	
	散布履歴ファイルが正しく生成される	○	
計画支援システム	散布履歴ファイルを登録することができる	○	
	散布履歴を正しく表示することができる	△	散布箇所表示に一部抜けあり
	散布履歴詳細情報が正しく表示できる	○	

（特記事項）
位置特定パッケージ調整中
散布履歴の表示が正しくなるようにプログラムの修正必要

表 1-4　総合試験実施結果（12 月 5 日）

システム名称	確認項目	合否	備考
操作支援システム	車線変更時に画面上の走行位置が正しく表示される	○	
	車線変更時に散布の制御が正しく変更される	×	車線変更時の散布制御ロジックが不十分のため、変更される場合とされない場合あり
散布車制御装置	手動散布に切替えたとき正しく制御が行える	○	
操作支援システム	手動散布時に手動の散布制御情報が画面に表示される	○	
	手動から自動に散布を切替えた時に、自動散布制御に切り替わる	○	
	自動から手動に散布を切替えた時に、自動散布の制御が行われない	○	
	自動散布済みの箇所の色が変わる	○	手動散布済み箇所については、色が変わらない仕様
	散布履歴ファイルが正しく生成される	○	
計画支援システム	散布履歴を正しく表示することができる	○	

（特記事項）
位置特定パッケージ調整中
車線変更に対応できるようにプログラムの修正必要（当初仕様では車線変更しても散布制御を変更しないため）
手動散布済み箇所についても色が変わるよう機能追加する

表 1-5　総合試験実施結果（12 月 10 日）

システム名称	確認項目	合否	備考
位置特定パッケージ	走行位置が正しく表示される	○	位置特定パッケージの改良版との結合
	車線変更時に画面上の走行位置が正しく表示される	○	
操作支援システム	散布開始地点で散布が開始される	○	
	散布変更地点で散布が変更される	○	
	散布終了地点で散布が停止される	○	
	手動散布済みの箇所の色が変わる	○	
	車線変更時に散布の制御が正しく変更される	○	
	散布予定車線以外を走行したときに散布制御が行われない	○	
	散布予定車線以外から散布予定箇所に入ったときに正しく散布制御が行われる	×	走行の仕方によって未散布箇所が散布済みとなる
計画支援システム	散布履歴を正しく表示することができる	○	

（特記事項）
散布予定箇所を走行した場合にも正しく散布制御が行えるようにプログラムの修正必要

表 1-6　総合試験実施結果（12 月 12 日）

システム名称	確認項目	合否	備考
操作支援システム	散布予定車線以外から散布予定箇所に入ったときに正しく散布制御が行われる	○	
	散布予定車線以外や散布予定車線内を複雑な動きをしても散布制御に影響しない	○	
	同じ箇所を２回走行した場合に２回目は散布されない	○	
	分岐の時の制御が１回目、２回目において正しく行われる	○	
	散布順を間違って走行しても散布される	○	
計画支援システム	散布履歴を正しく表示することができる	○	
（特記事項）			

表 1-7　総合試験実施結果（12 月 17 日）

システム名称	確認項目	合否	備考
操作支援システム	背景地図が正しく表示される	○	
	走行位置が正しく表示される	○	
	地物検索が正しく表示される	○	
	散布制御が正しく行える	△	散布車停止時において、未散布箇所が散布済みとなる
	散布履歴ファイルが正しく生成される	○	
位置特定パッケージ	データが100%取得できる	△	PCカード暴走する時あり
散布車制御装置	散布制御装置に指示が行われている	△	PCカード暴走時に通信エラーが発生し、その後制御不可となる

（特記事項）
下り車線においてメッシュデータの仕様違いによりマップマッチングしない（現地で修正して対応）
ランプ部分岐点において、副道走行中であっても、マーカーを検知するまでは、主道走行と判定されるため、散布順エラーとなる
散布車の位置表示が遅れる
地図データが多く表示に負荷がかかり、PCの処理が間に合わない可能性あり。表示地物を減らして地図データを小さくする
通信エラー時の散布車制御装置への散布制御を１秒以上あけるようにプログラムを修正

表 1-8　総合試験実施結果（12 月 17 日）

システム名称	確認項目	合否	備考
計画支援システム	事務所から接続可能である	○	
	出張所から接続可能である	○	
	ステーションから接続可能である	○	
	散布計画データが正しく作成できる	○	
	作成した散布計画データが正しく表示される	○	
	作成した散布計画データを正常にダウンロードできる	○	
	散布履歴ファイルを登録することができる	○	
	散布履歴を正しく表示することができる	○	
	散布履歴詳細情報が正しく表示できる	○	

（特記事項）
背景地図の表示時間は、初回約15秒、２回目以降約9秒

表 1-9　総合試験実施結果（2 月 6 日）

システム名称	確認項目	合否	備考
巡回支援システム	操作支援システムが起動される	○	
	気温が表示される	○	
	音声認識が正常にできる	○	騒音状態により何度か言い直しは必要であるが実用上は問題なし
	認識された音声にしたがって画面の表示が切り替わる	○	
	音声が認識された時に認識された音声にしたがって音声案内が行われる	○	
	巡回地点における巡回情報が正しく巡回記録ファイルに出力される	○	
	操作支援システム終了後、巡回支援システムが終了される	○	
	前回巡回記録がある場合、正常に結合される	○	
位置特定パッケージ	データが100%取得できる	○	CPU使用率40%程度
計画支援システム	巡回記録データが登録できる	○	
	巡回記録ファイルがダウンロードできる	○	Excelが未インストールのためダウンロードの確認
	巡回記録修正ファイルがアップロードできる	○	

（特記事項）
地図データの建物を公共施設のみとし家は削除、また施設名も削除（データ量は前回の約10分の1）

2. 操作マニュアルの作成

計画支援システムおよび操作支援システムの操作マニュアルを作成した。操作マニュアルの目次を次に示す。また、操作マニュアルを別添資料に示す。

＜計画支援システムのマニュアル目次＞

１．システムでできる機能
２．URL
３．ログイン
４．通常メニュー
　４－１．散布計画表示
　４－２．散布履歴表示
　４－３．巡回記録表示
　４－４．散布履歴登録
　４－５．巡回記録登録
５．管理者メニュー
　５－１．散布計画作成
　５－２．作業者名リスト作成
６．地図表示の使い方

＜操作支援システムのマニュアル目次＞

１．パソコンの説明
２．接続機器
３．位置特定機器
４．パソコンと周辺機器の接続
５．操作手順
　５－１．開始操作手順
　５－２．システム起動準備
　５－３．システムの起動
　５－４．操作開始時
　５－５．音声入力
　５－６．音声一覧

５－７．自動散布と手動散布の切替え
５－８．操作停止時
５－９．終了操作手順
６．拡張機能
７．故障かなと思ったら

別添資料４　長岡国道の実験データ分析

（散布計画・履歴、巡回記録データ）

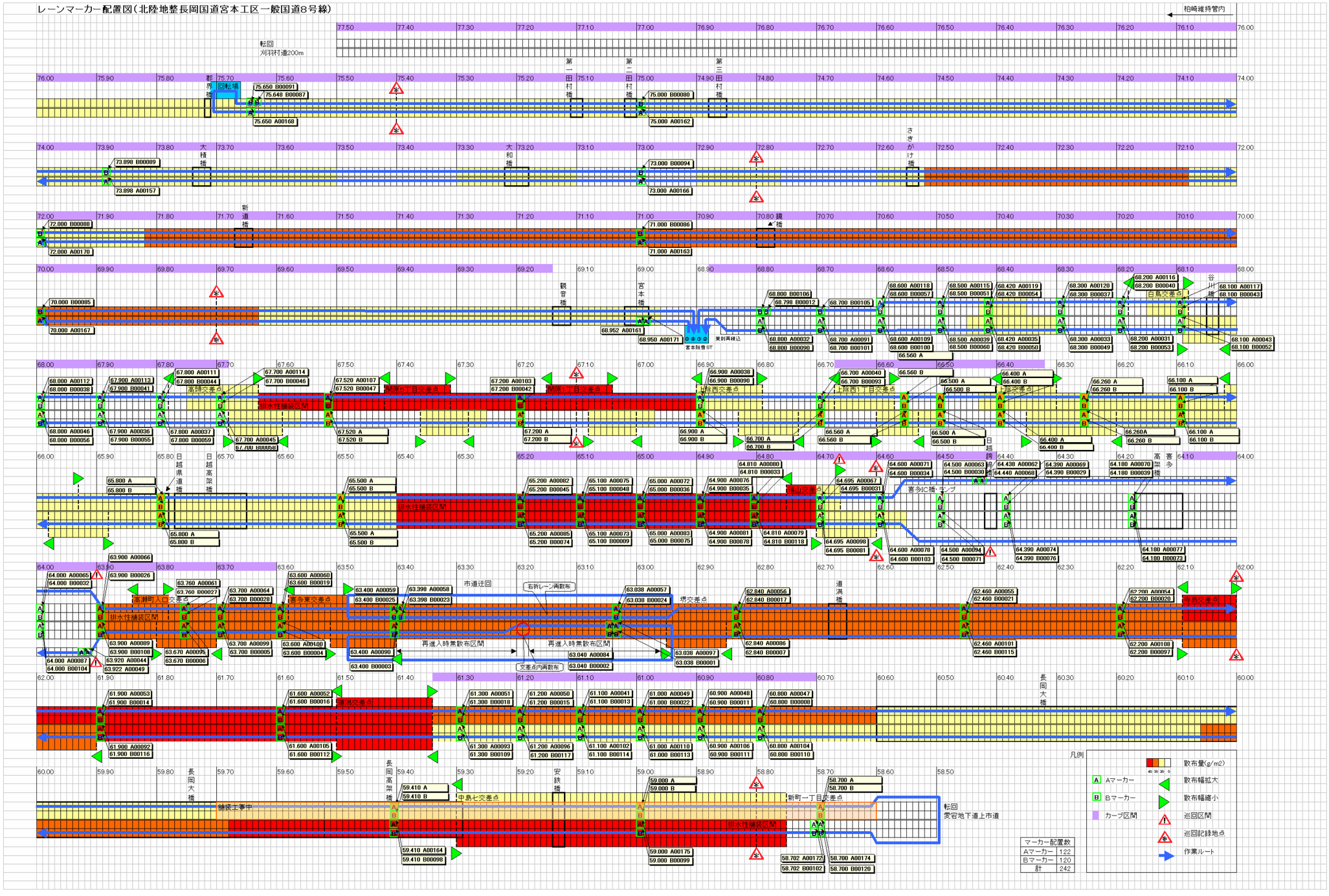


図 4-1　長岡宮本工区のレーンマーカ配置図

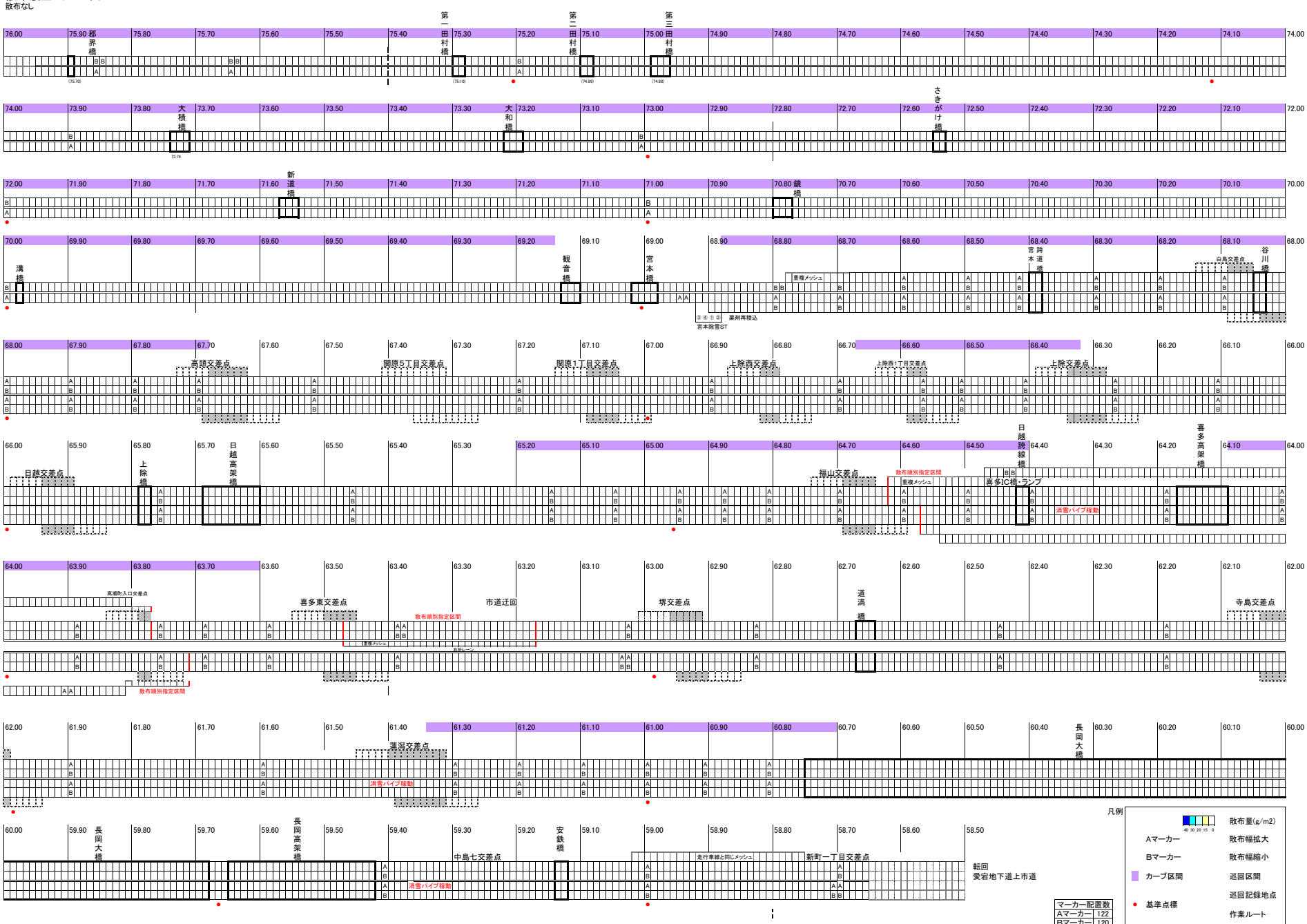
1. 散布計画　散布履歴メッシュ図

2. 自動散布　散布履歴メッシュ図

3. 手動散布　散布履歴メッシュ図

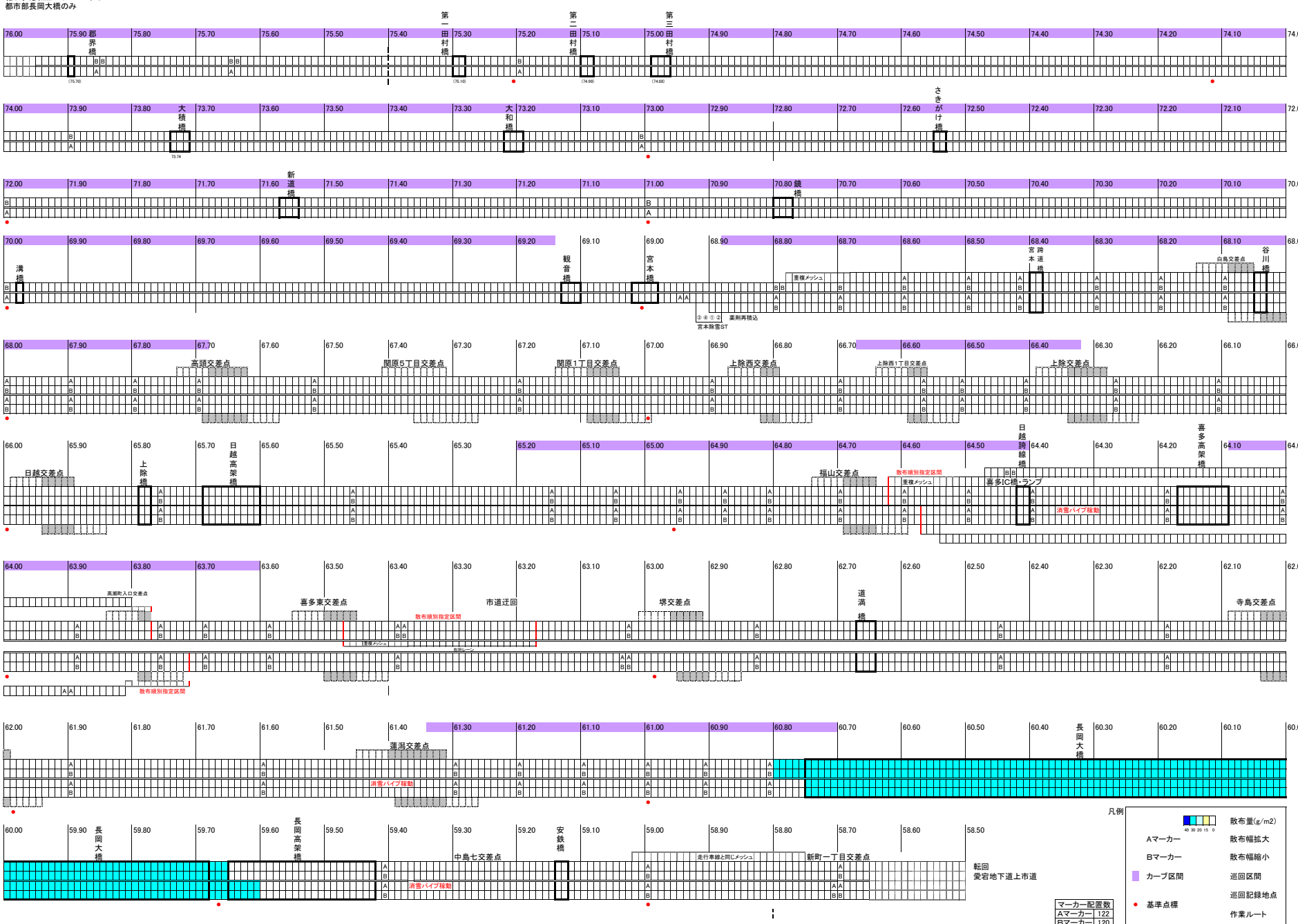
散布計画 **散布履歴メッシュ図**

散布なし



散布計画 **散布履歴メッシュ図**

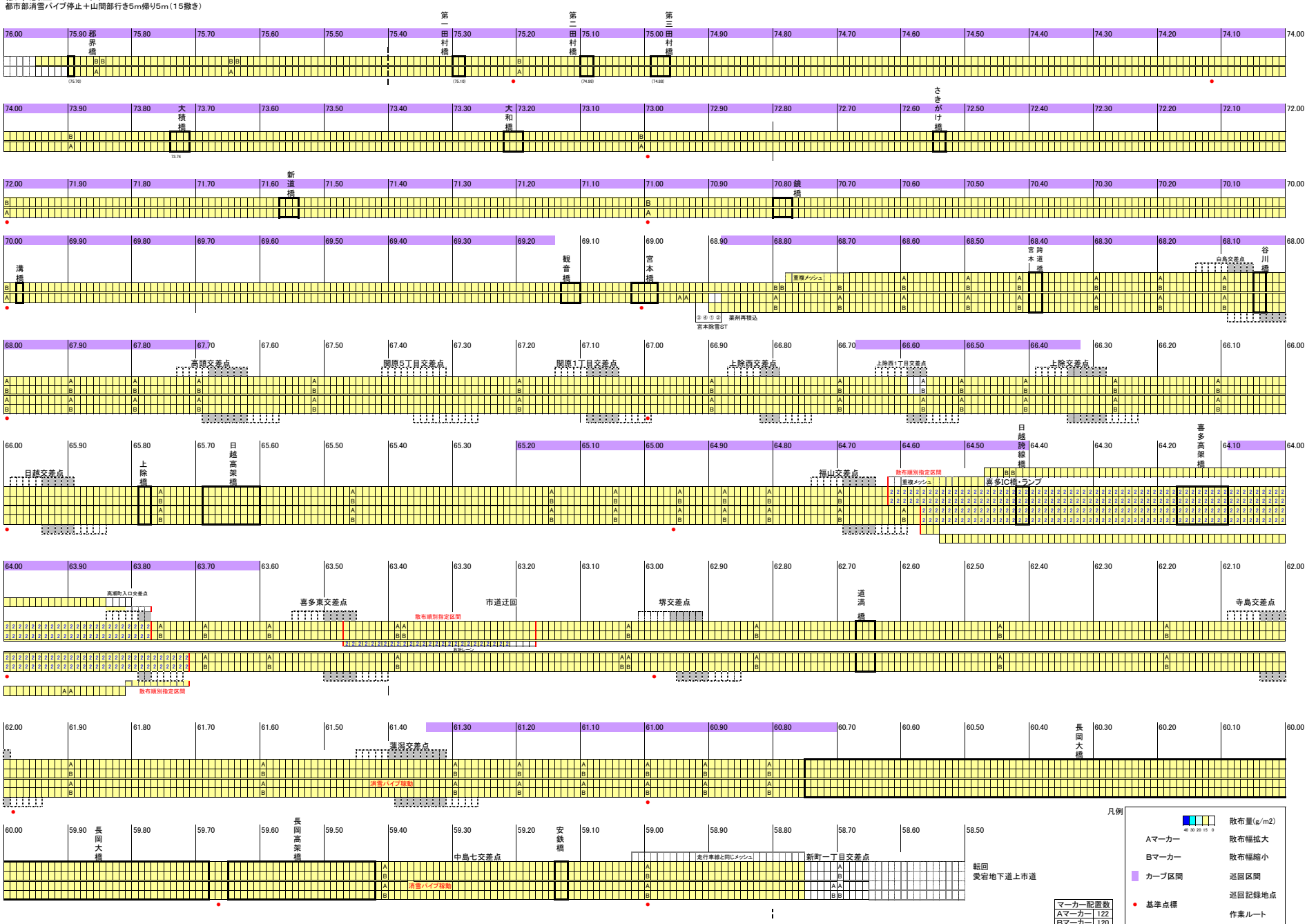
都市部長岡大橋のみ



マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

散布計画 **散布履歴メッシュ図**

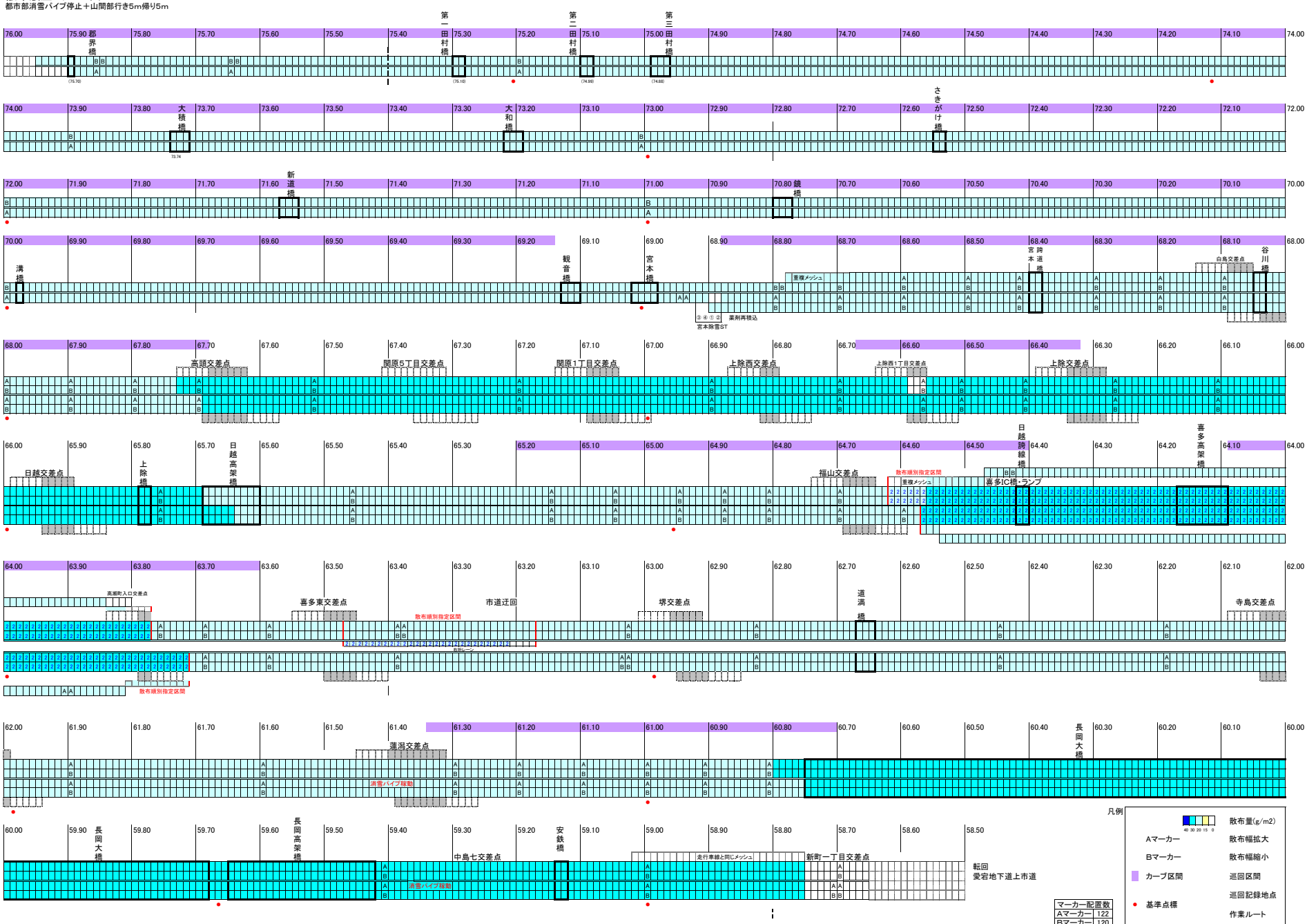
都市部消雪パイプ停止＋山間部行き5m帰り5m（15撒き）



マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

散布計画 **散布履歴メッシュ図**

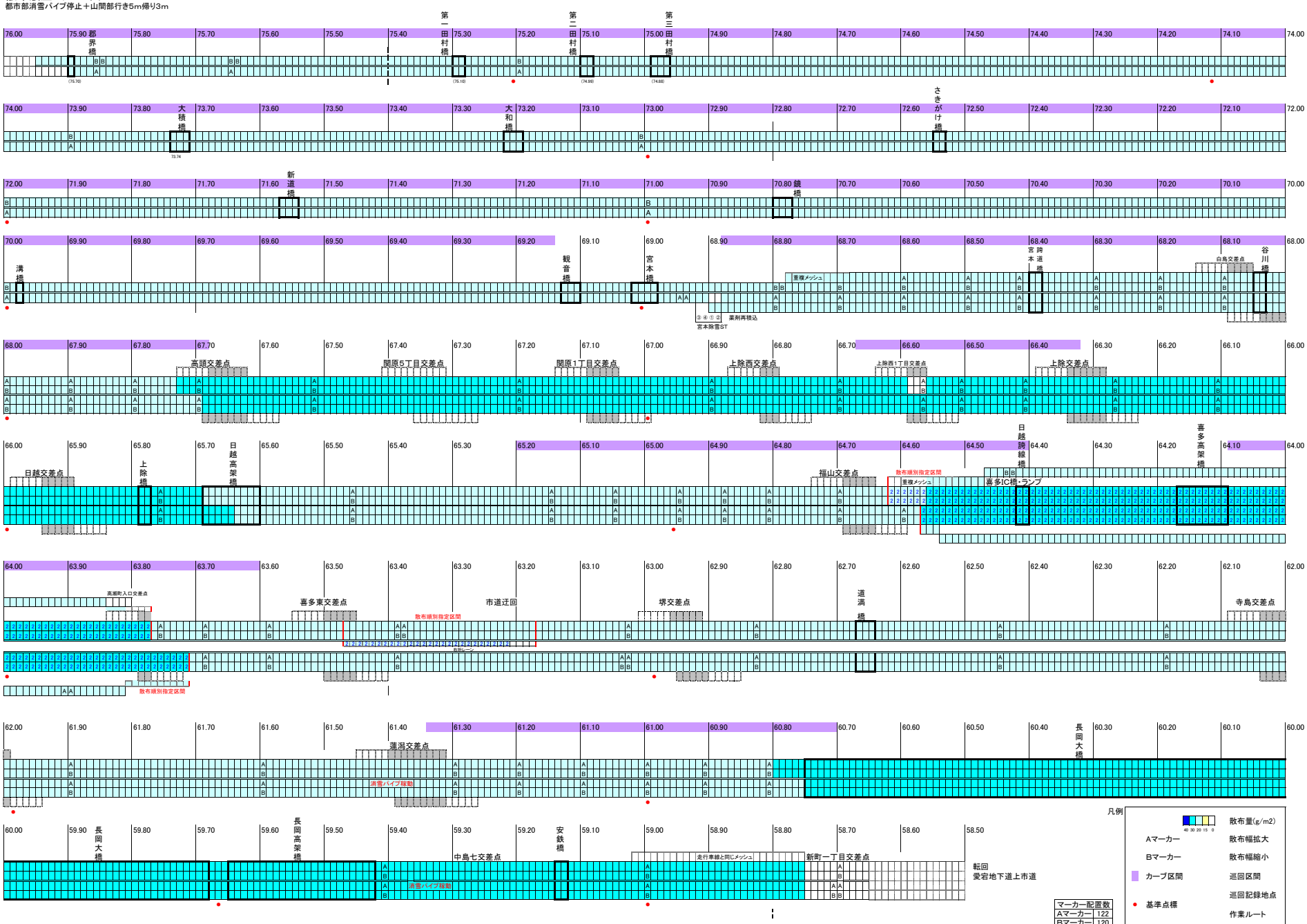
都市部消雪パイプ停止＋山間部行き5m帰り5m



マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

散布計画 **散布履歴メッシュ図**

都市部消雪パイプ停止＋山間部行き5m帰り3m



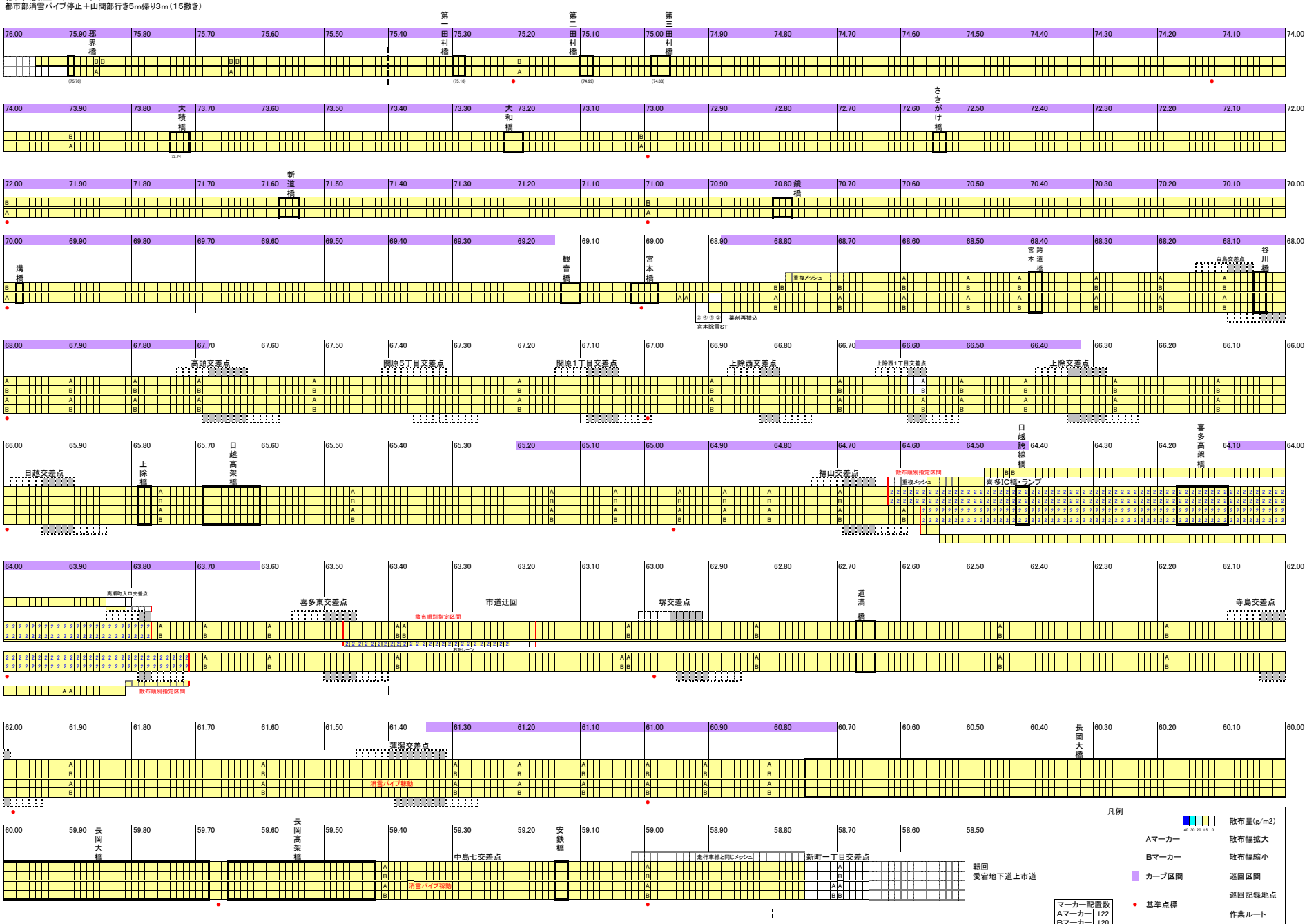
凡例

Aマーカー	散布量(g/m2)
Bマーカー	散布幅拡大
カーブ区間	散布幅縮小
基準点標	巡回区間
	巡回記録地点
	作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

散布計画 **散布履歴メッシュ図**

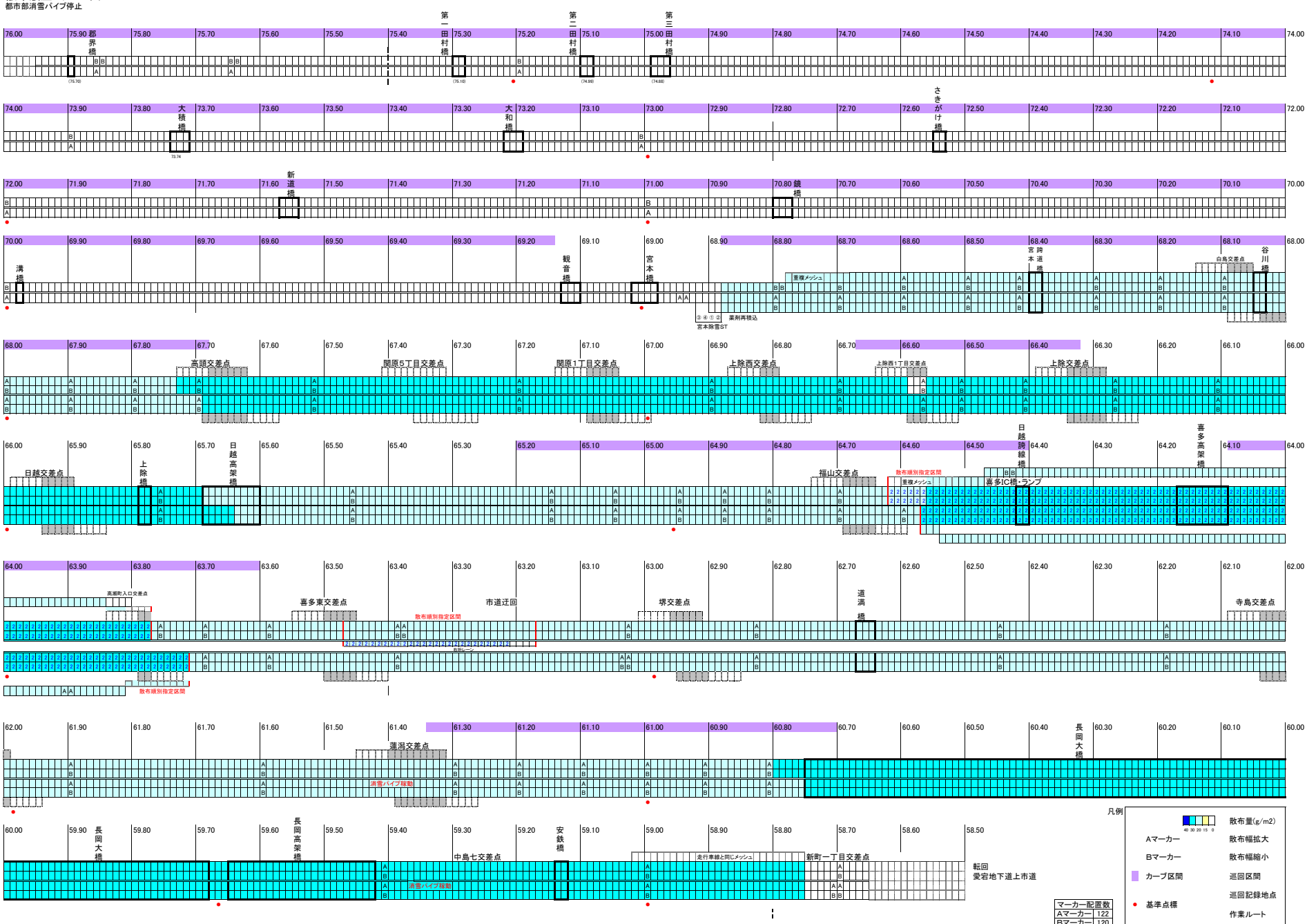
都市部消雪パイプ停止＋山間部行き5m帰り3m（15撒き）



マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

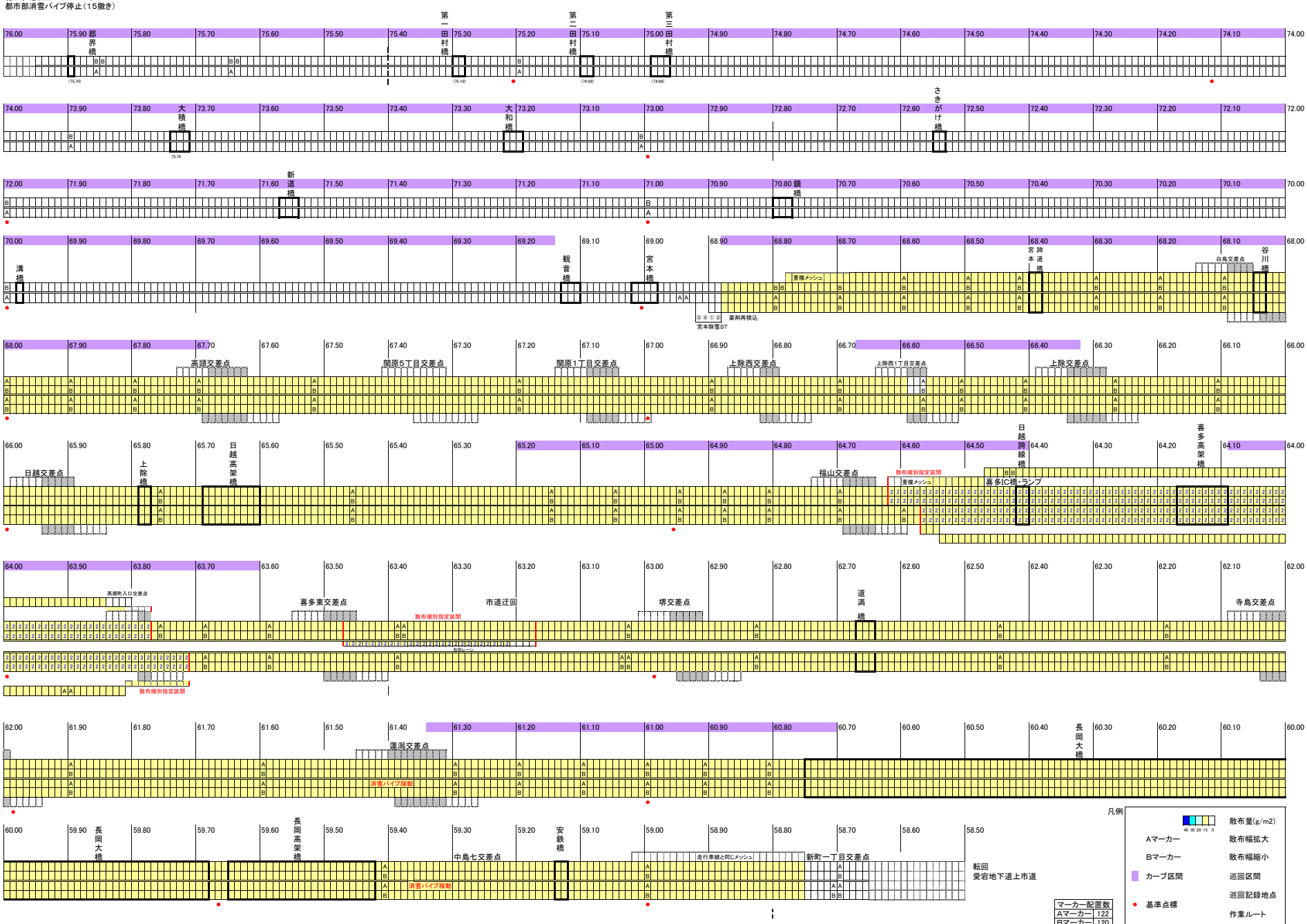
散布計画 **散布履歴メッシュ図**

都市部消雪パイプ停止



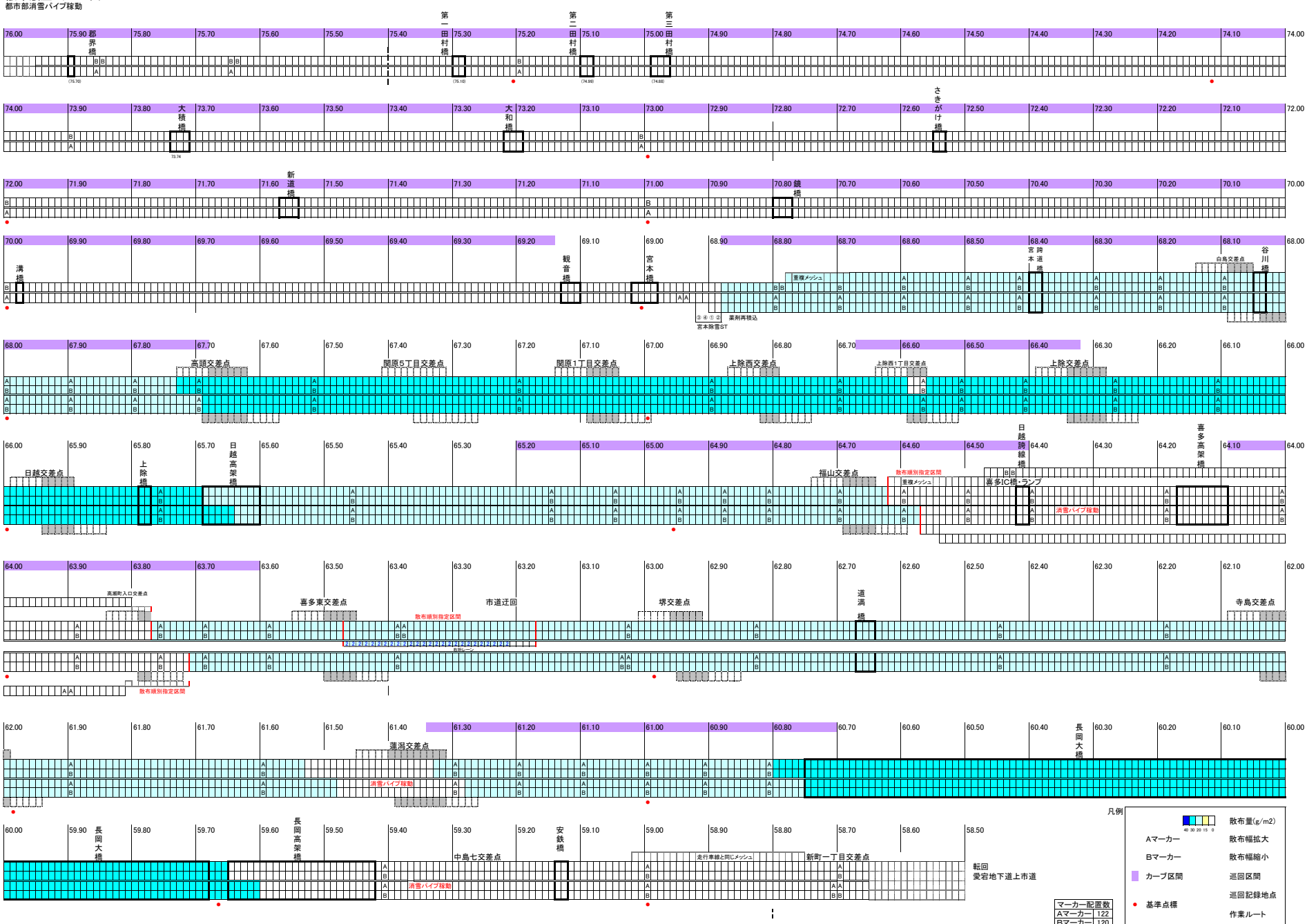
散布計画 **散布履歴メッシュ図**

都市部消雪パイプ停止（15撒き）



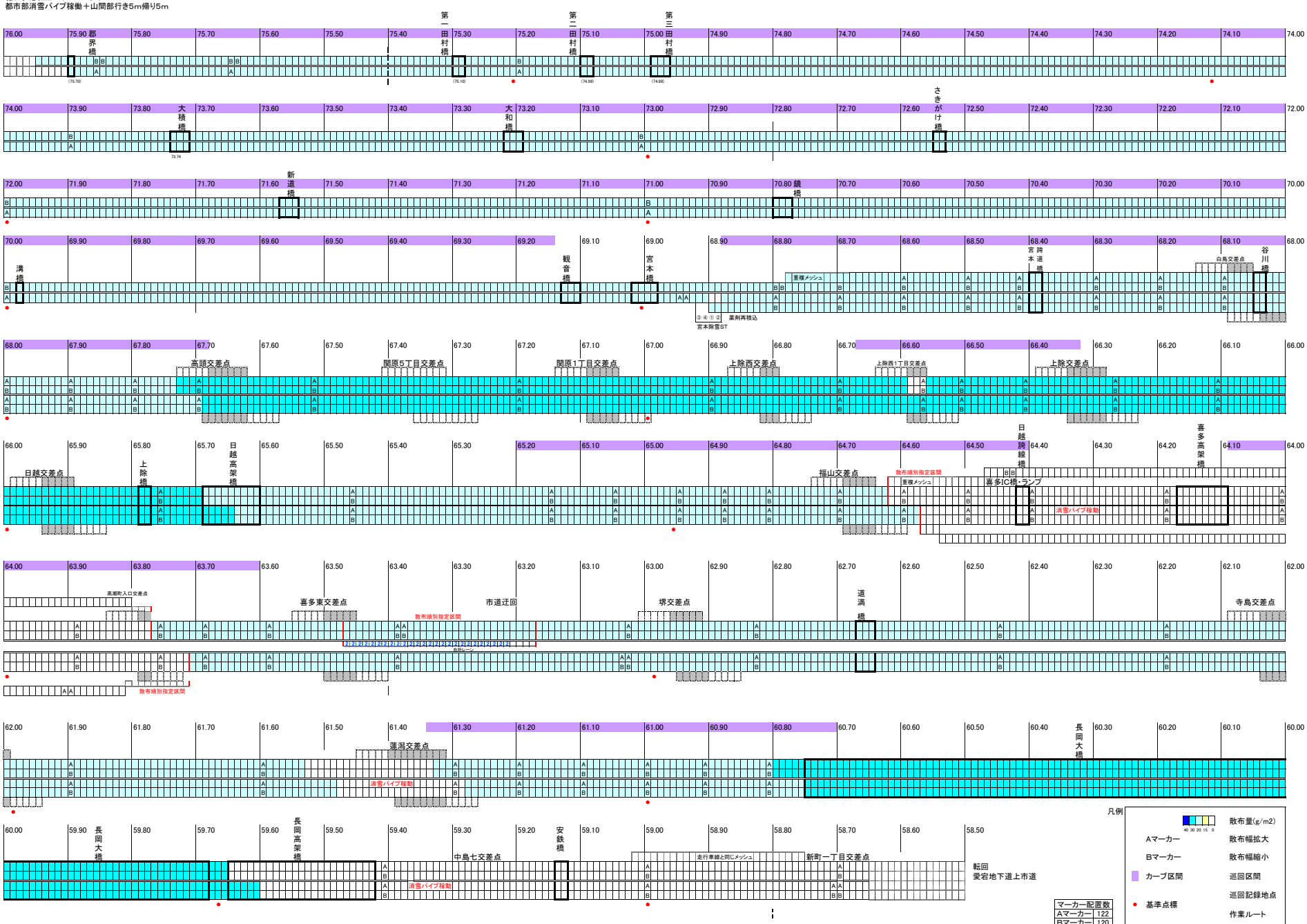
散布計画 **散布履歴メッシュ図**

都市部消雪パイプ稼動



散布計画 **散布履歴メッシュ図**

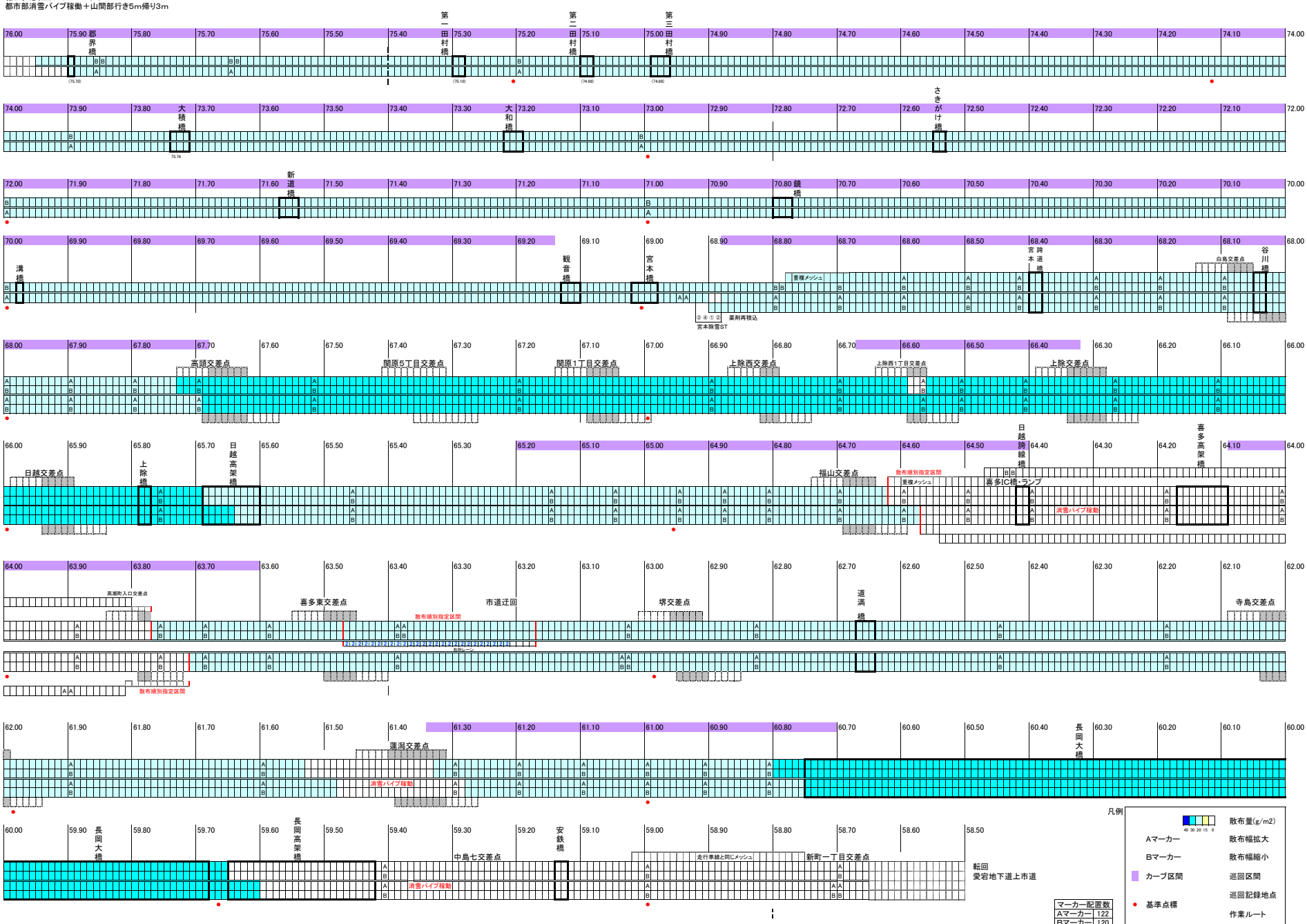
都市部消雪パイプ稼働＋山間部行き5m帰り5m



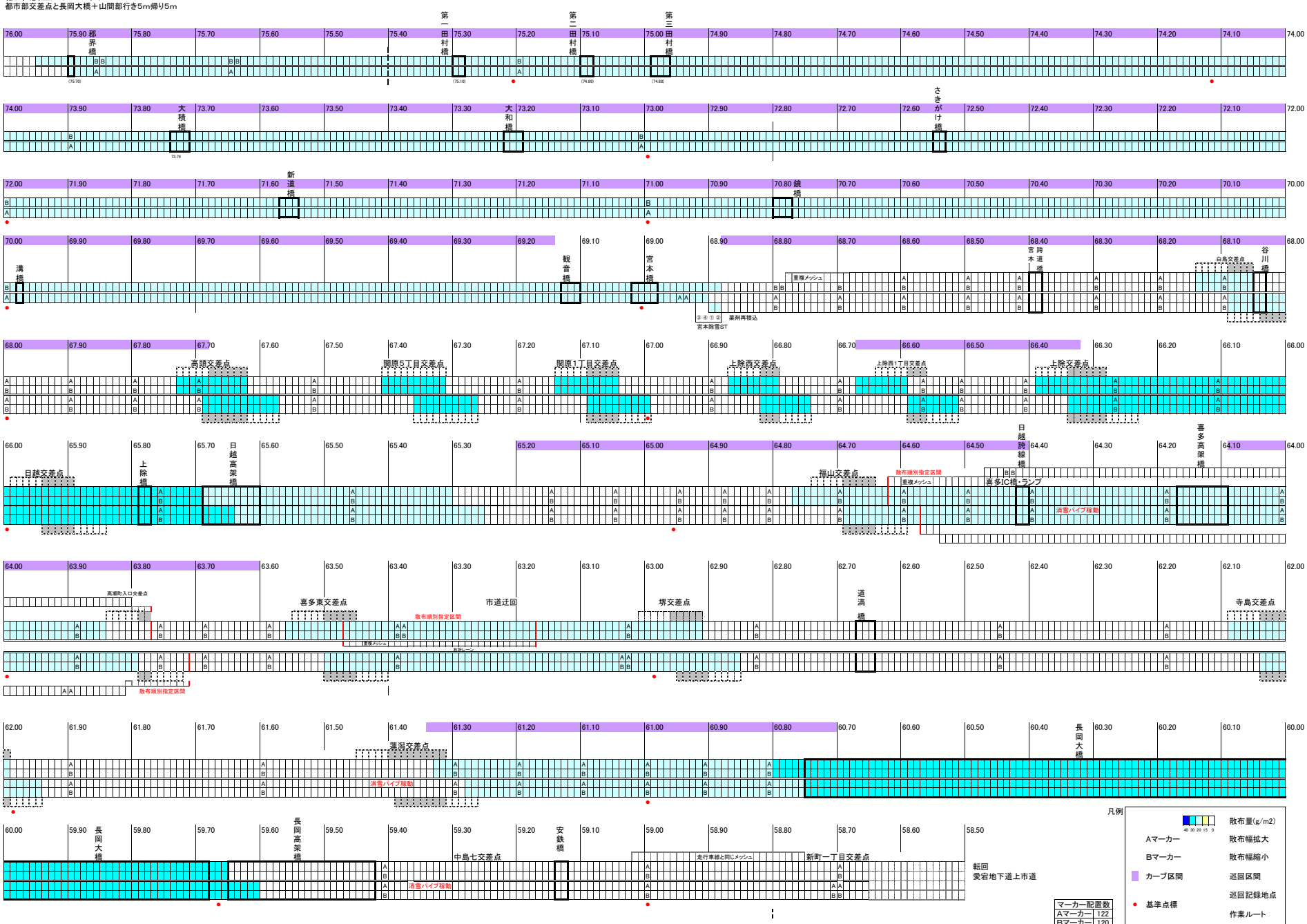
マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

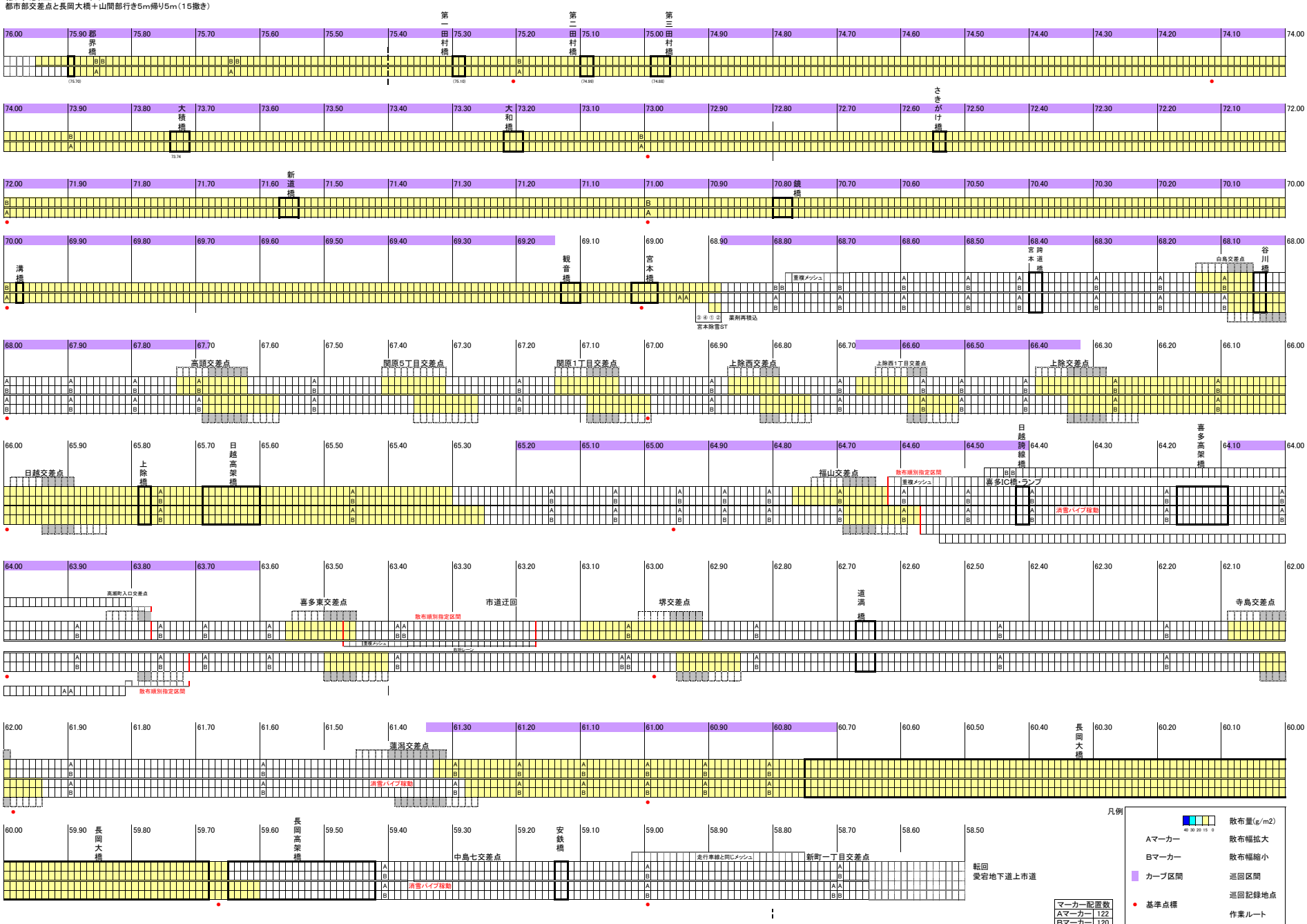
散布計画 **散布履歴メッシュ図**

都市部消雪パイプ稼働＋山間部行き5m帰り3m

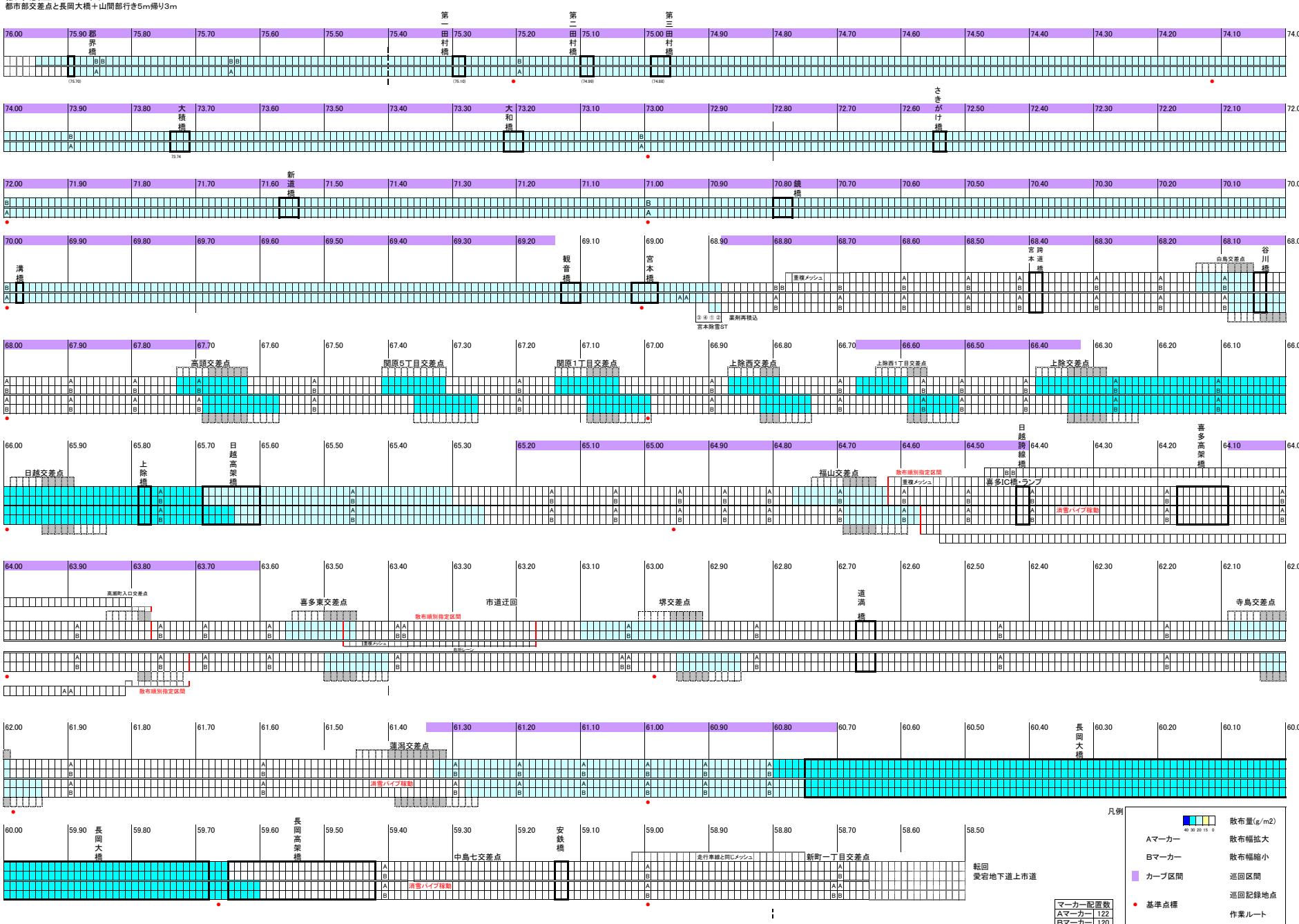


散布計画 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り5m

散布計画 **散布履歴メッシュ図**

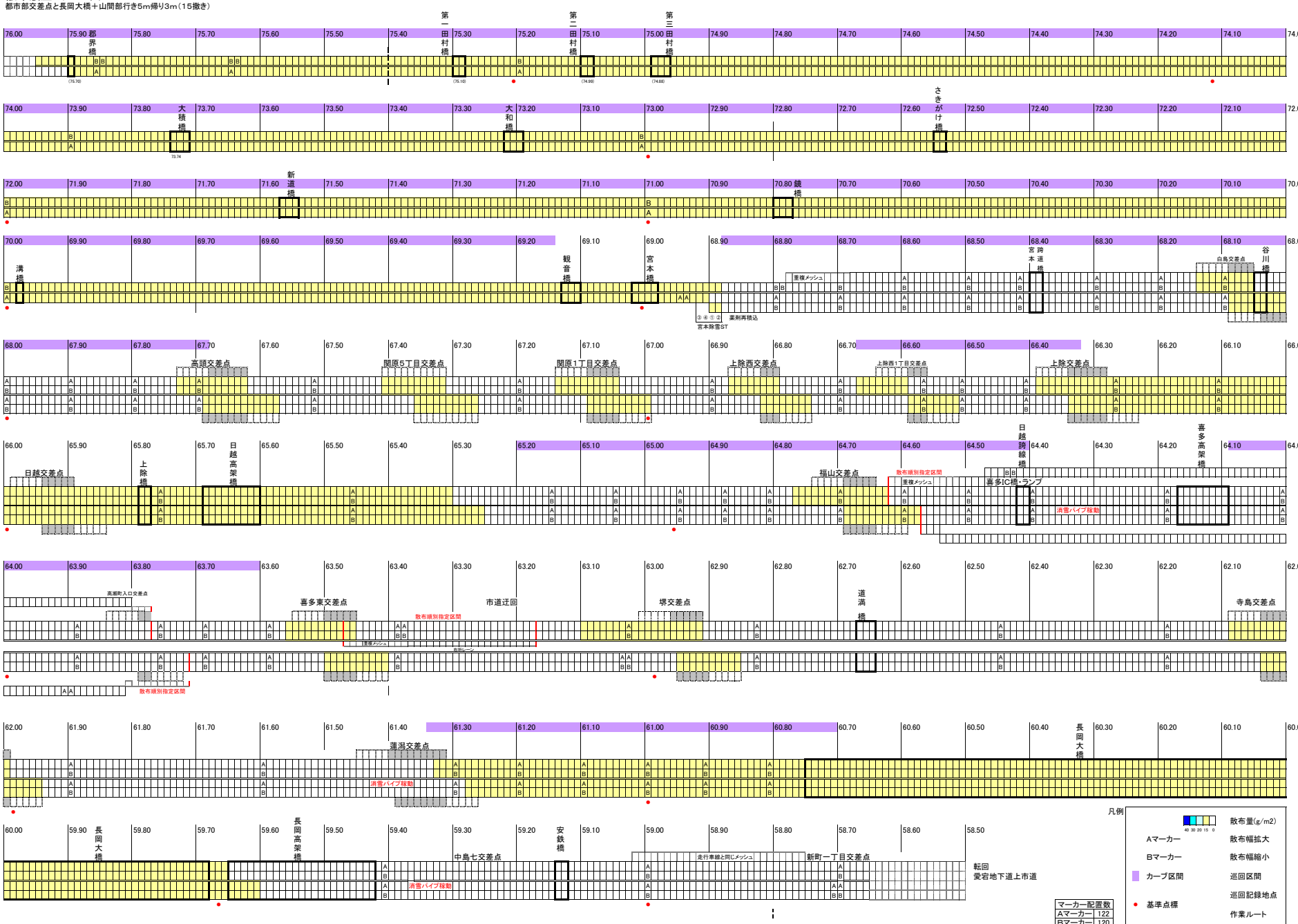
都市部交差点と長岡大橋＋山間部行き5m帰り5m（15撒き）

散布計画 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り3m



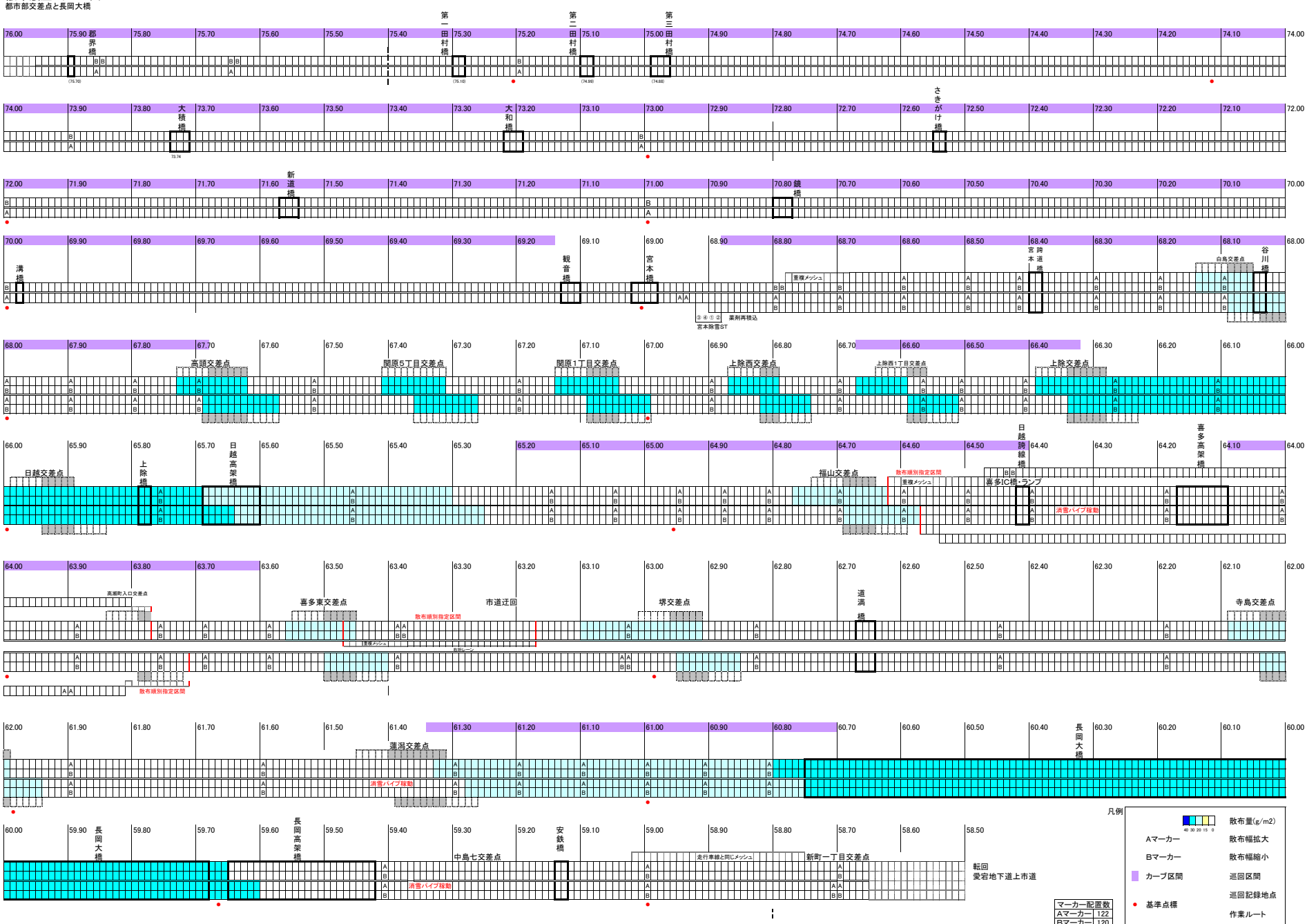
散布計画 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り3m（15撒き）



散布計画 **散布履歴メッシュ図**

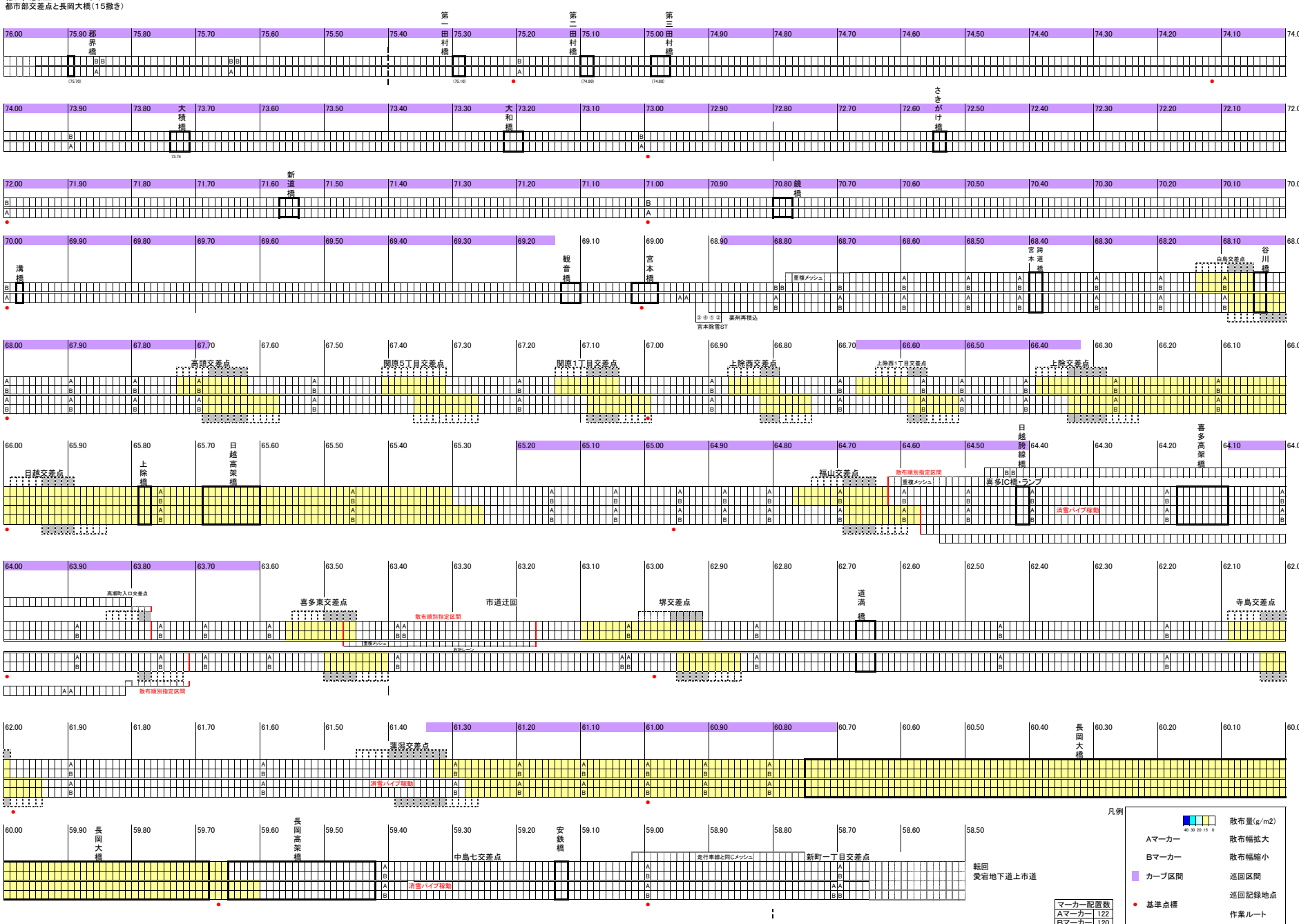
都市部交差点と長岡大橋



マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

散布計画 **散布履歴メッシュ図**

都市部交差点と長岡大橋（15撒き）



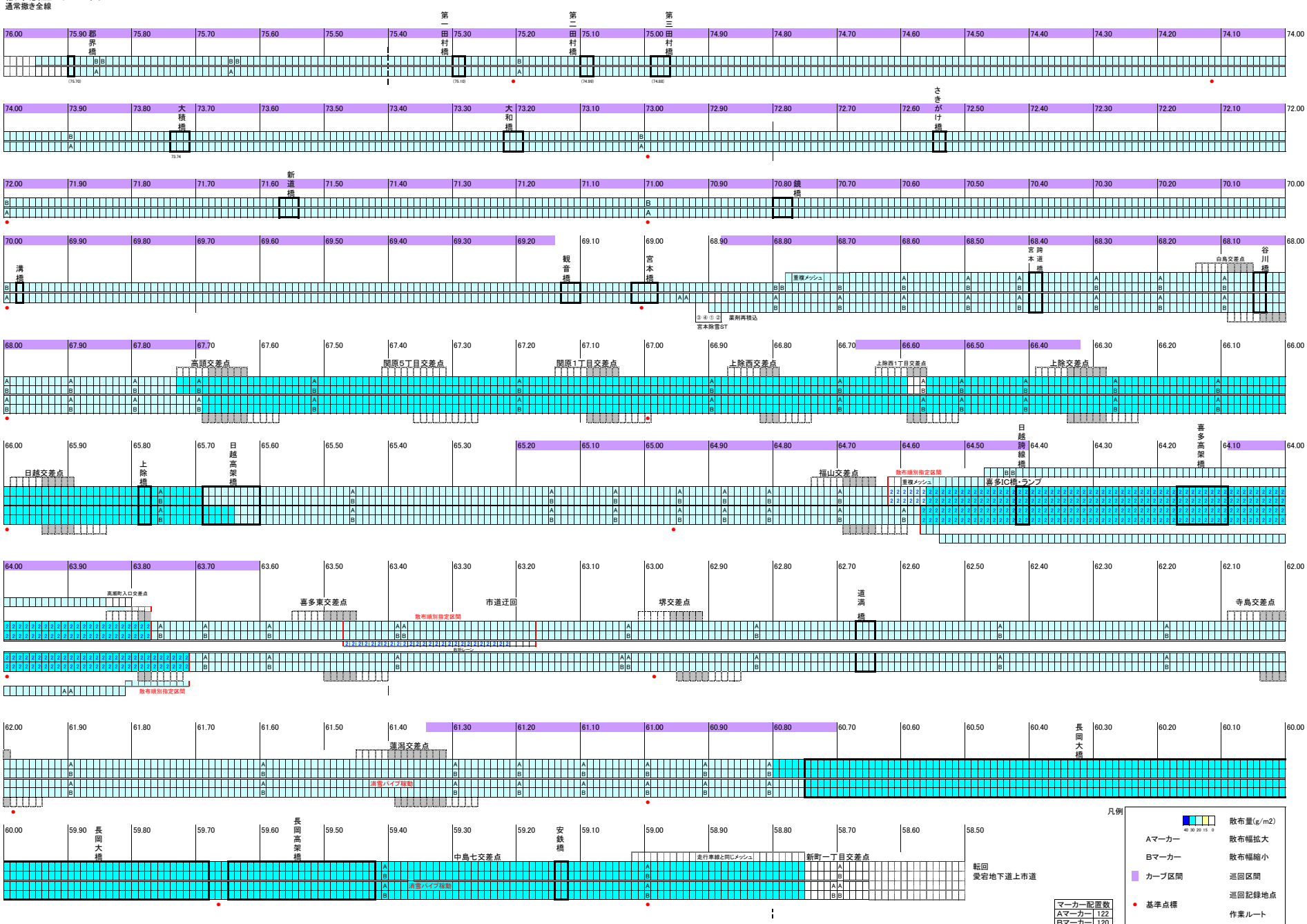
凡例

散布量(g/m2)	40 30 20 15 0
Aマーカー	散布幅拡大
Bマーカー	散布幅縮小
カーブ区間	巡回区間
基準点標	巡回記録地点
	作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

散布計画 **散布履歴メッシュ図**

通常撒き全線



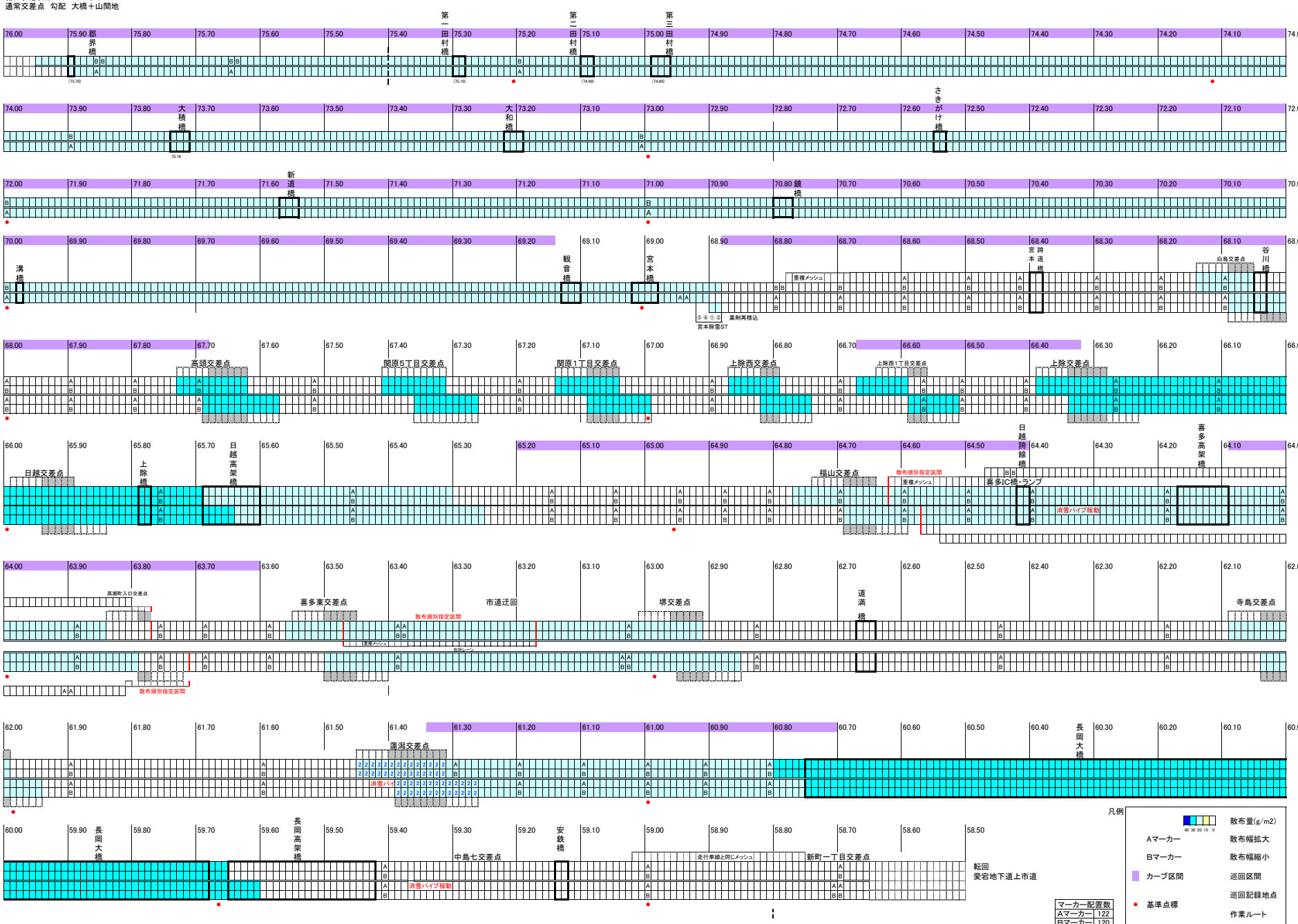
凡例

Aマーカー	散布量(g/m2)
Bマーカー	散布幅拡大
カーブ区間	散布幅縮小
基準点標	巡回区間
	巡回記録地点
	作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

散布計画 **散布履歴メッシュ図**

通常交差点　勾配　大橋＋山間地



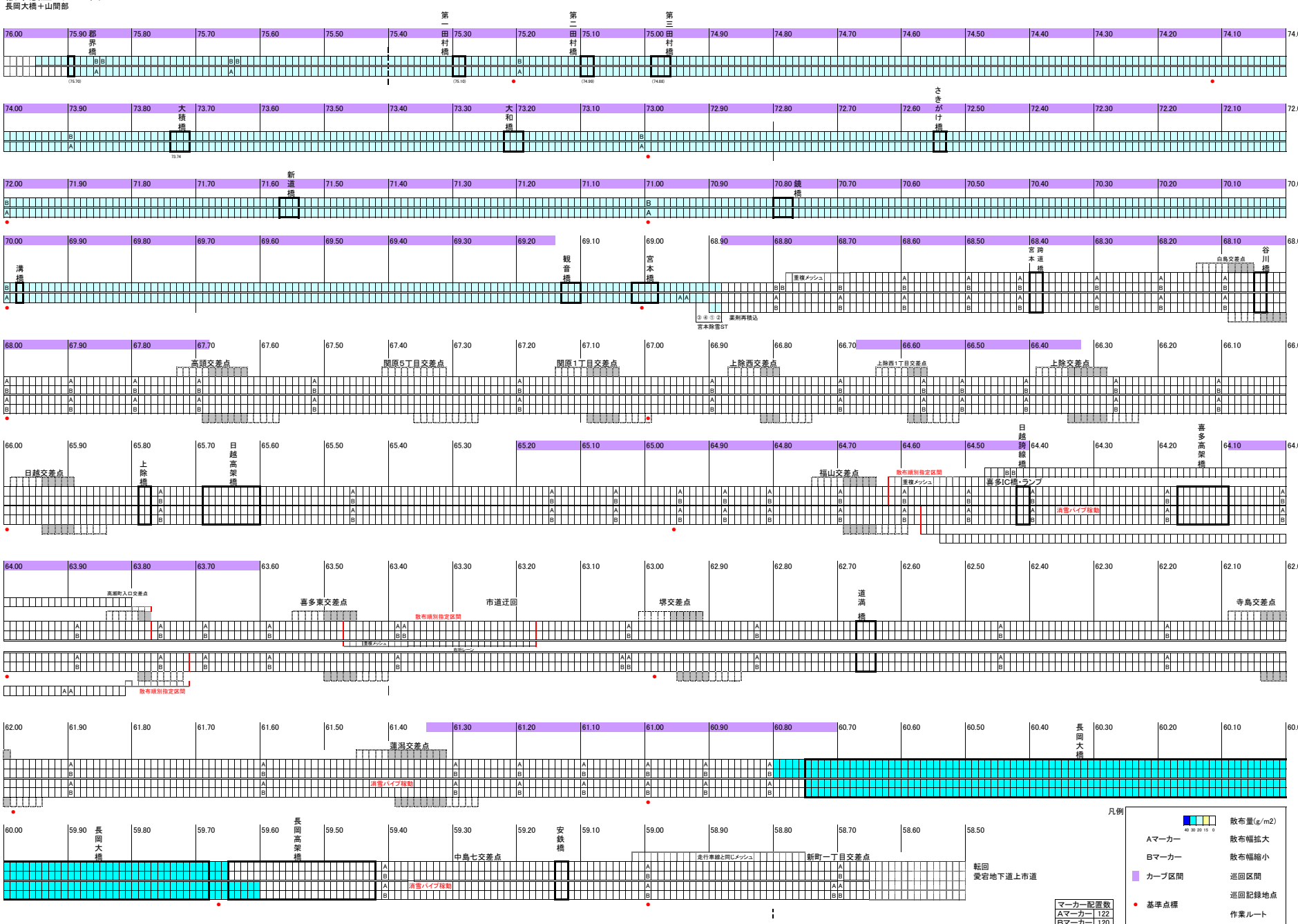
凡例

Aマーカー	散布量(g/m2)
Bマーカー	散布幅拡大
カーブ区間	散布幅縮小
基準点標	巡回区間
	巡回記録地点
	作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

散布計画 **散布履歴メッシュ図**

長岡大橋＋山間部



凡例

散布量(g/m2)　40 30 20 15 0

Aマーカー　散布幅拡大

Bマーカー　散布幅縮小

カーブ区間　巡回区間

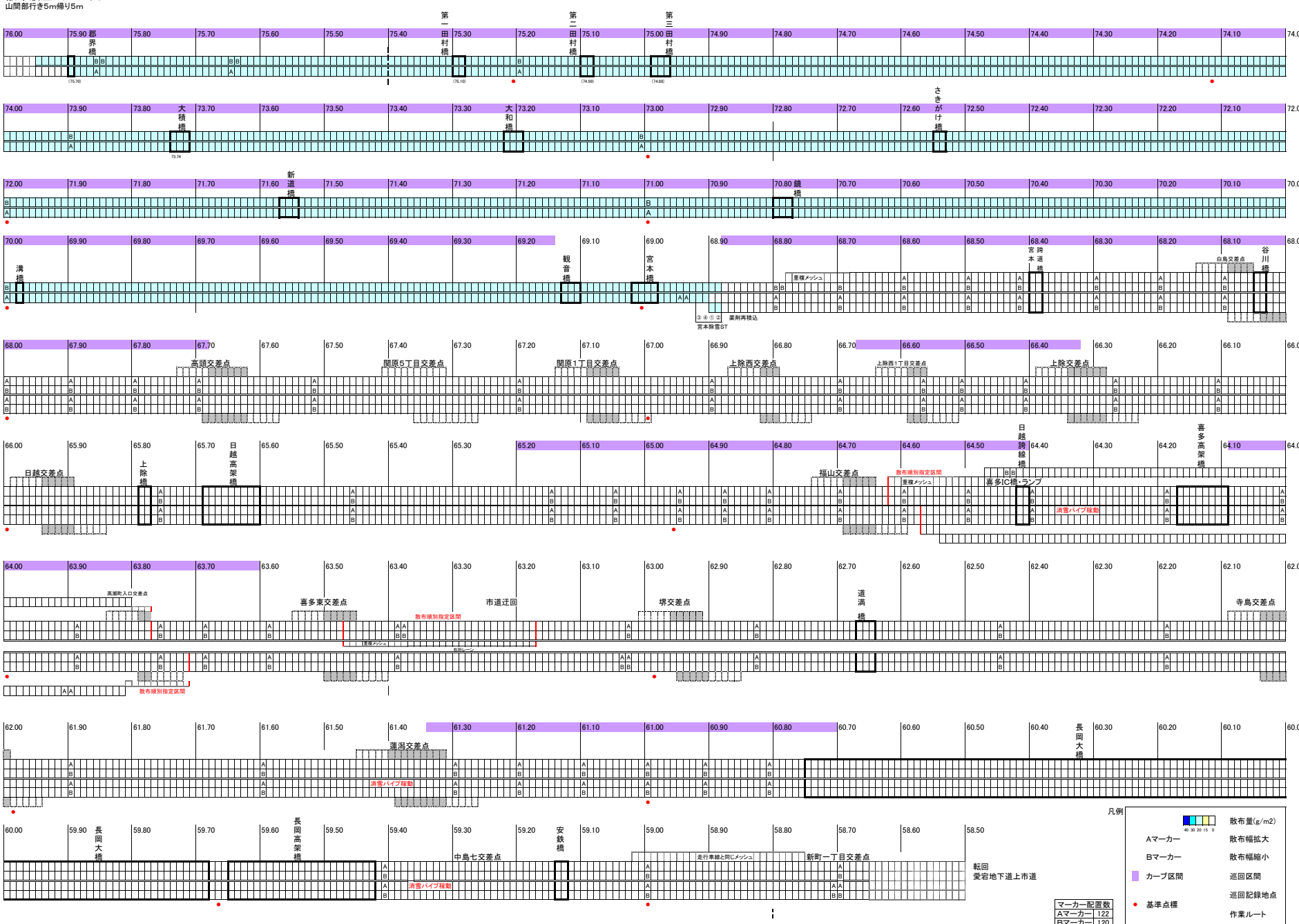
基準点標　巡回記録地点

作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

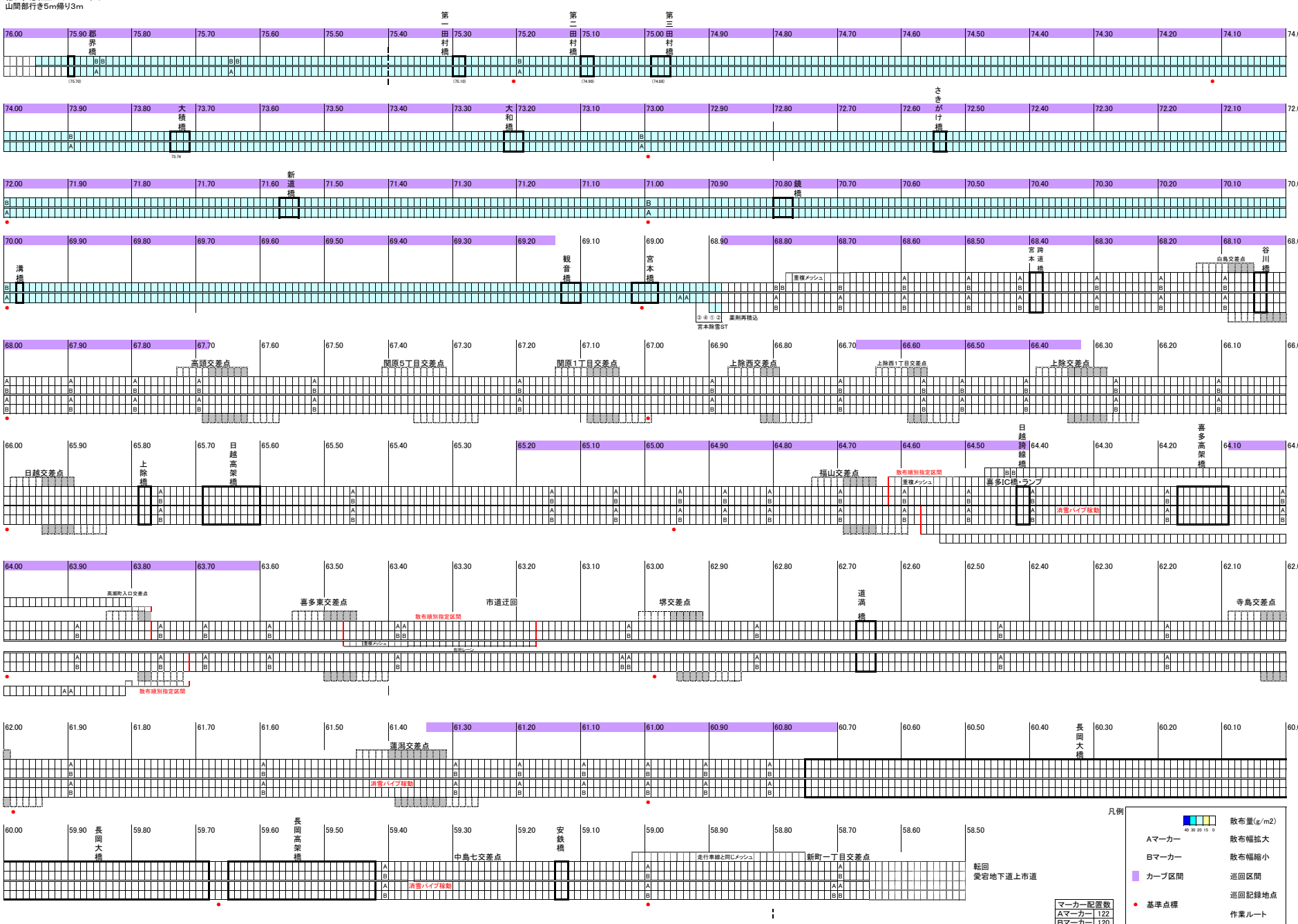
散布計画 **散布履歴メッシュ図**

山間部行き5m帰り5m



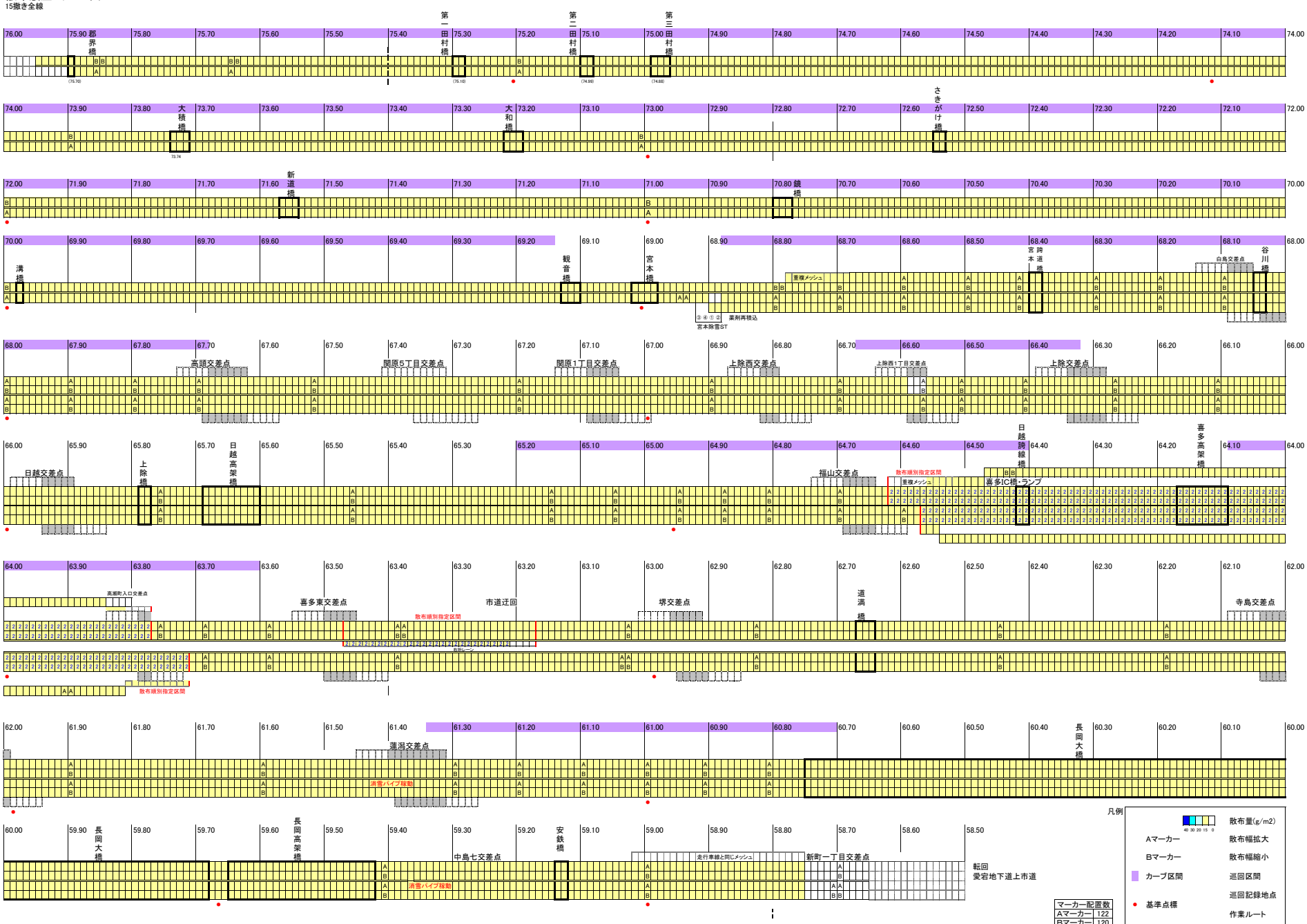
散布計画 **散布履歴メッシュ図**

山間部行き5m帰り3m

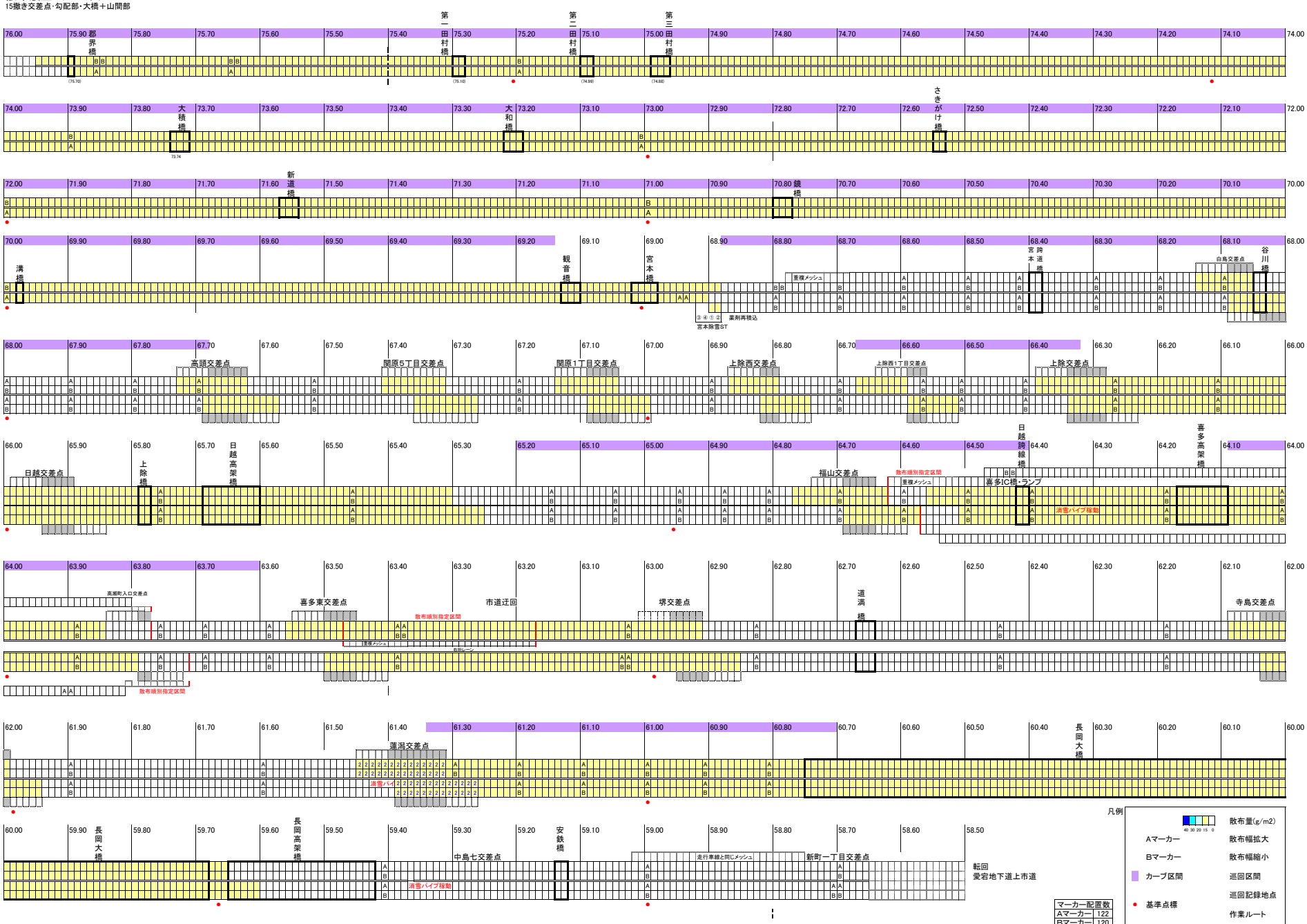


散布計画 **散布履歴メッシュ図**

15撒き全線



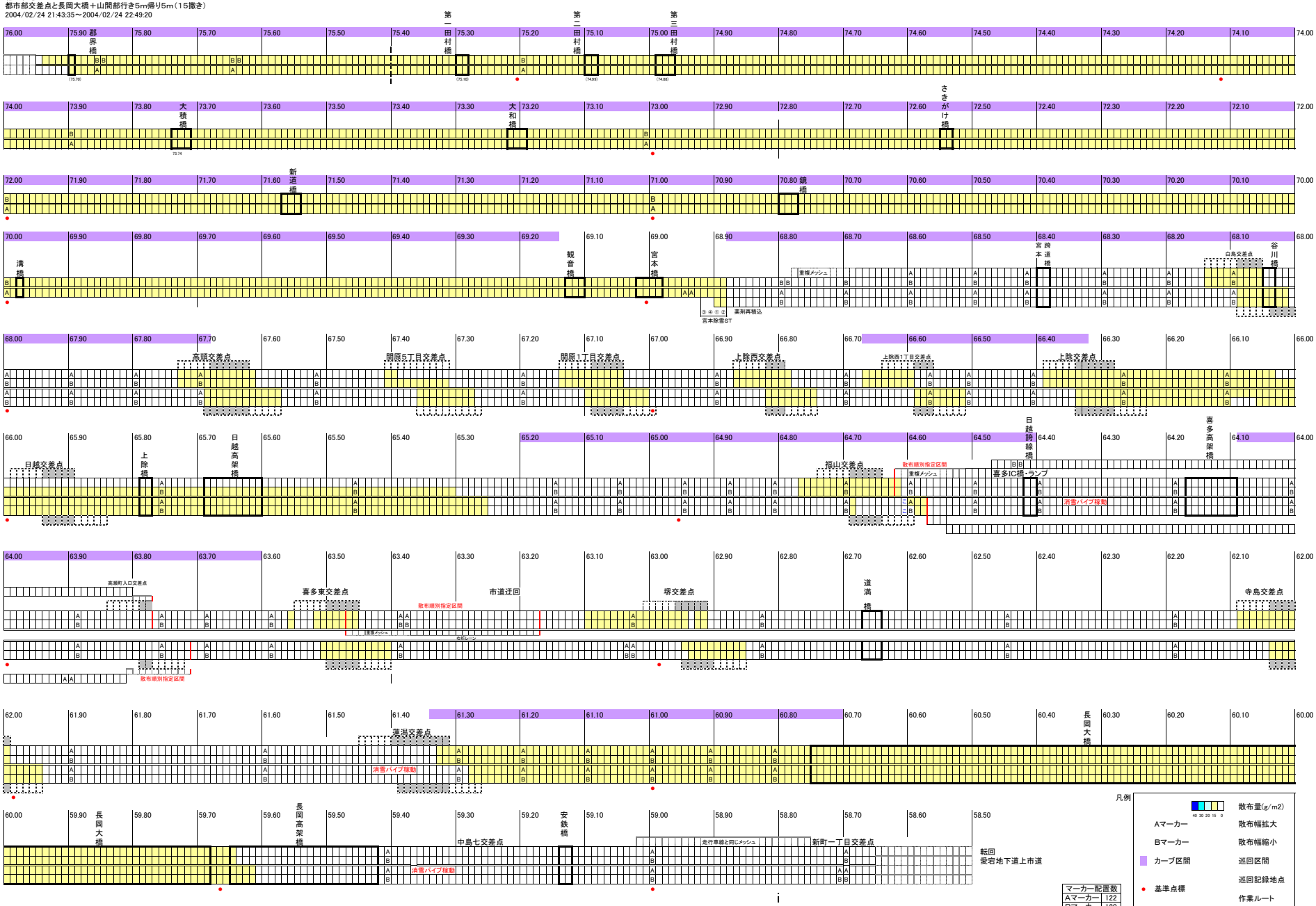
散布計画 **散布履歴メッシュ図**

15撒き交差点・勾配部・大橋＋山間部

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

自動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り5m（15撒き）

2004/02/24 21:43:35～2004/02/24 22:49:20

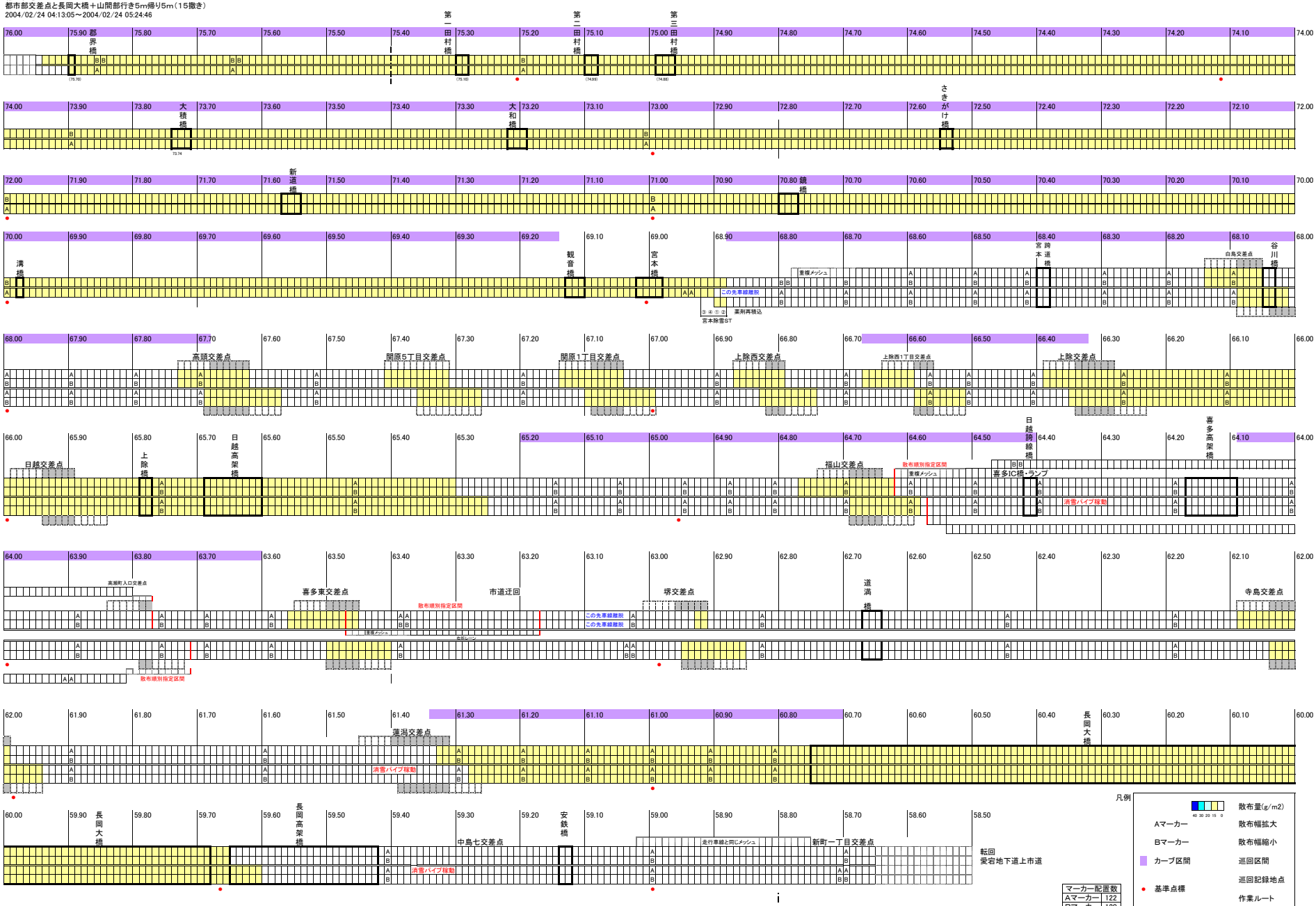


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

自動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り5m（15撒き）

2004/02/24 04:13:05～2004/02/24 05:24:46

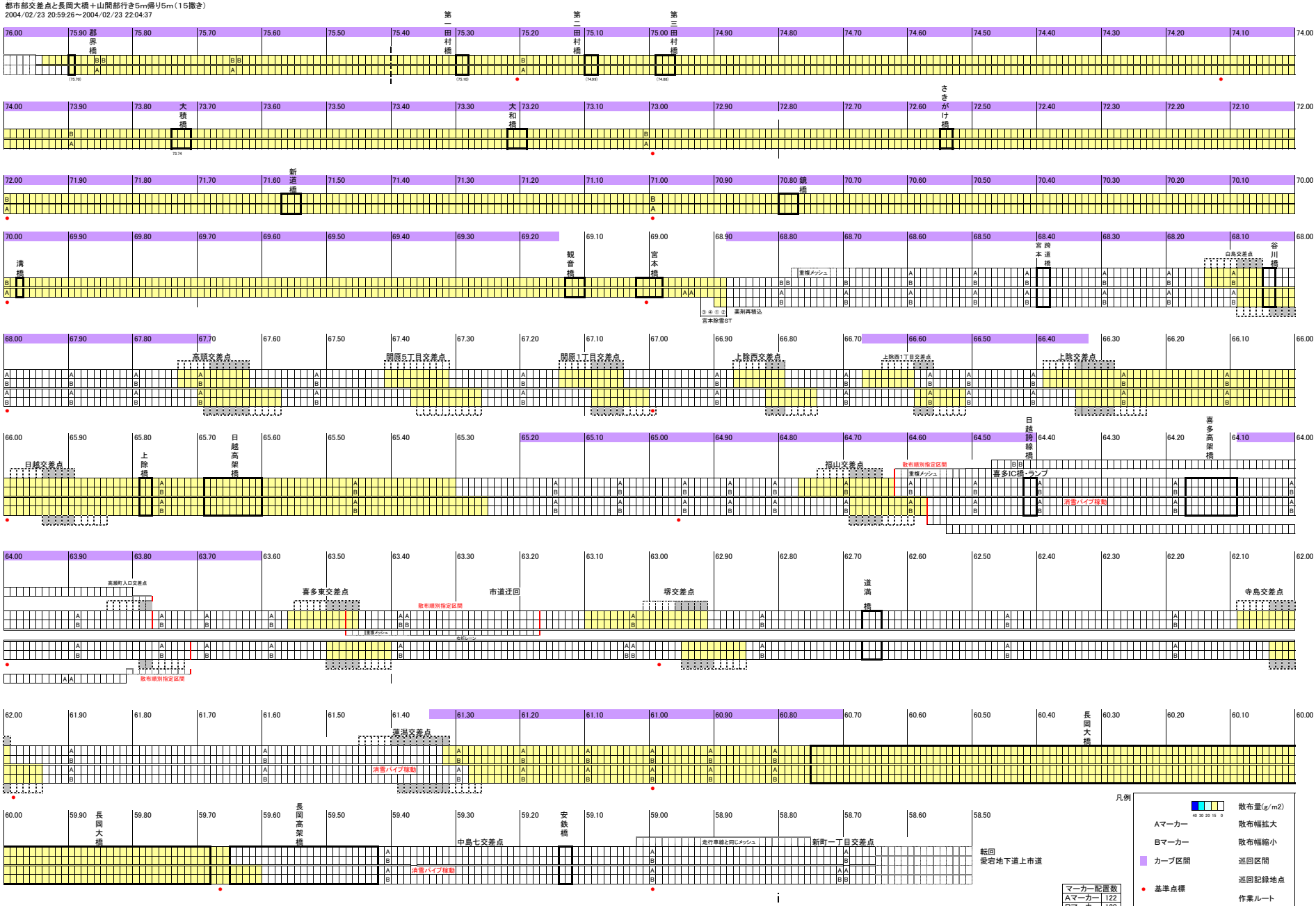


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

自動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り5m（15撒き）

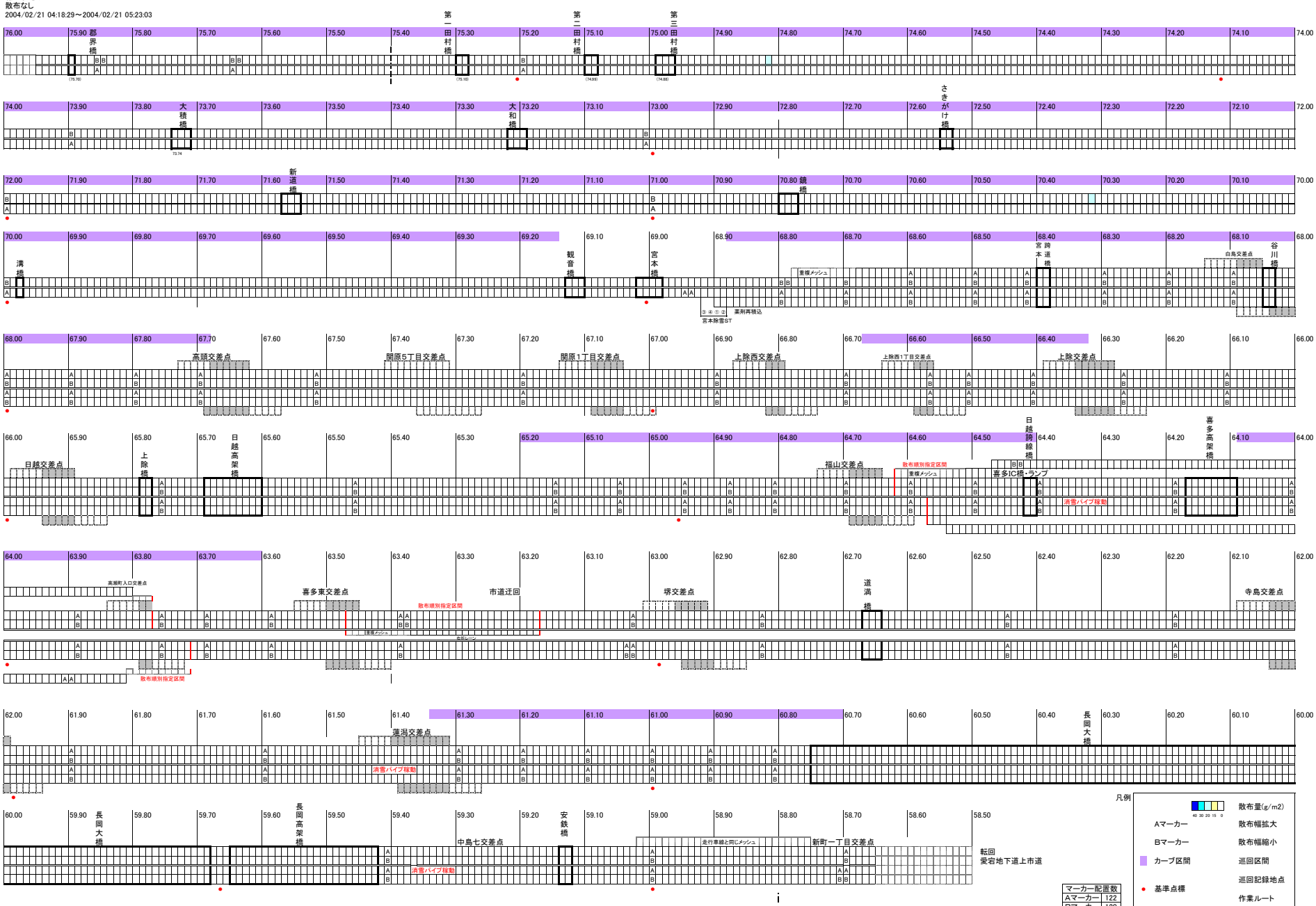
2004/02/23 20:59:26～2004/02/23 22:04:37



自動散布 **散布履歴メッシュ図**

散布なし

2004/02/21 04:18:29～2004/02/21 05:23:03

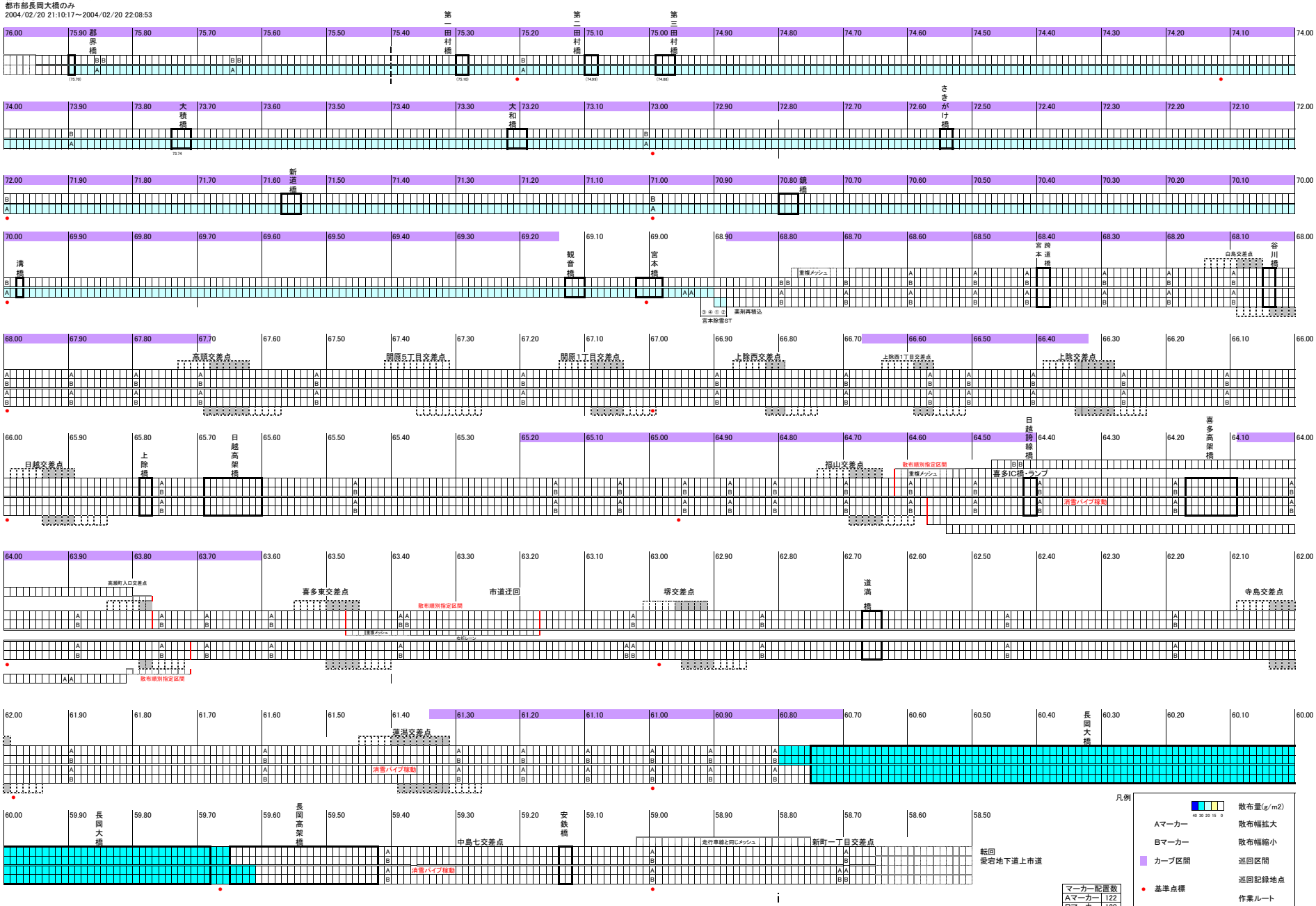


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

自動散布 **散布履歴メッシュ図**

都市部長岡大橋のみ

2004/02/20 21:10:17～2004/02/20 22:08:53

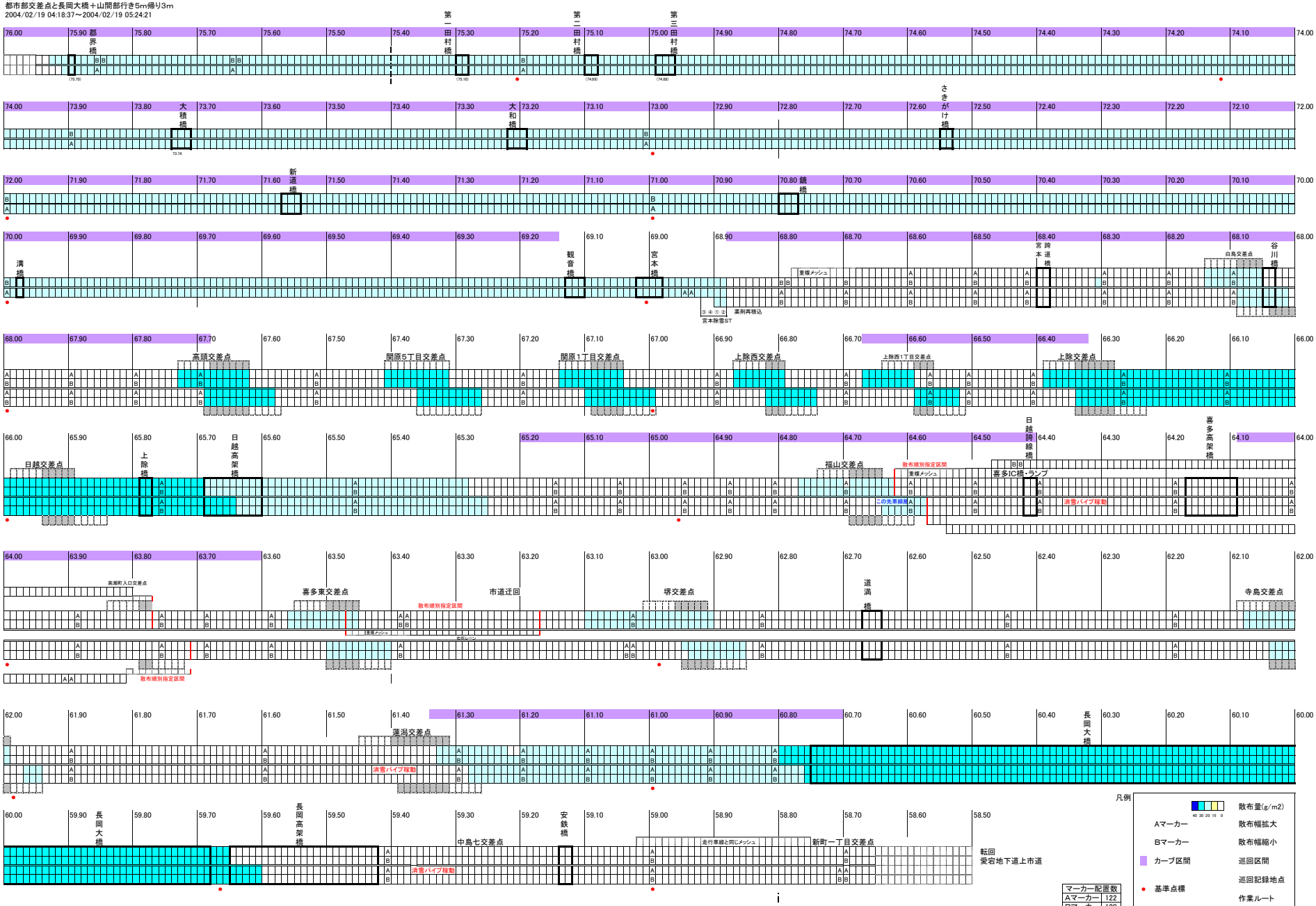


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

自動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り3m

2004/02/19 04:18:37～2004/02/19 05:24:21

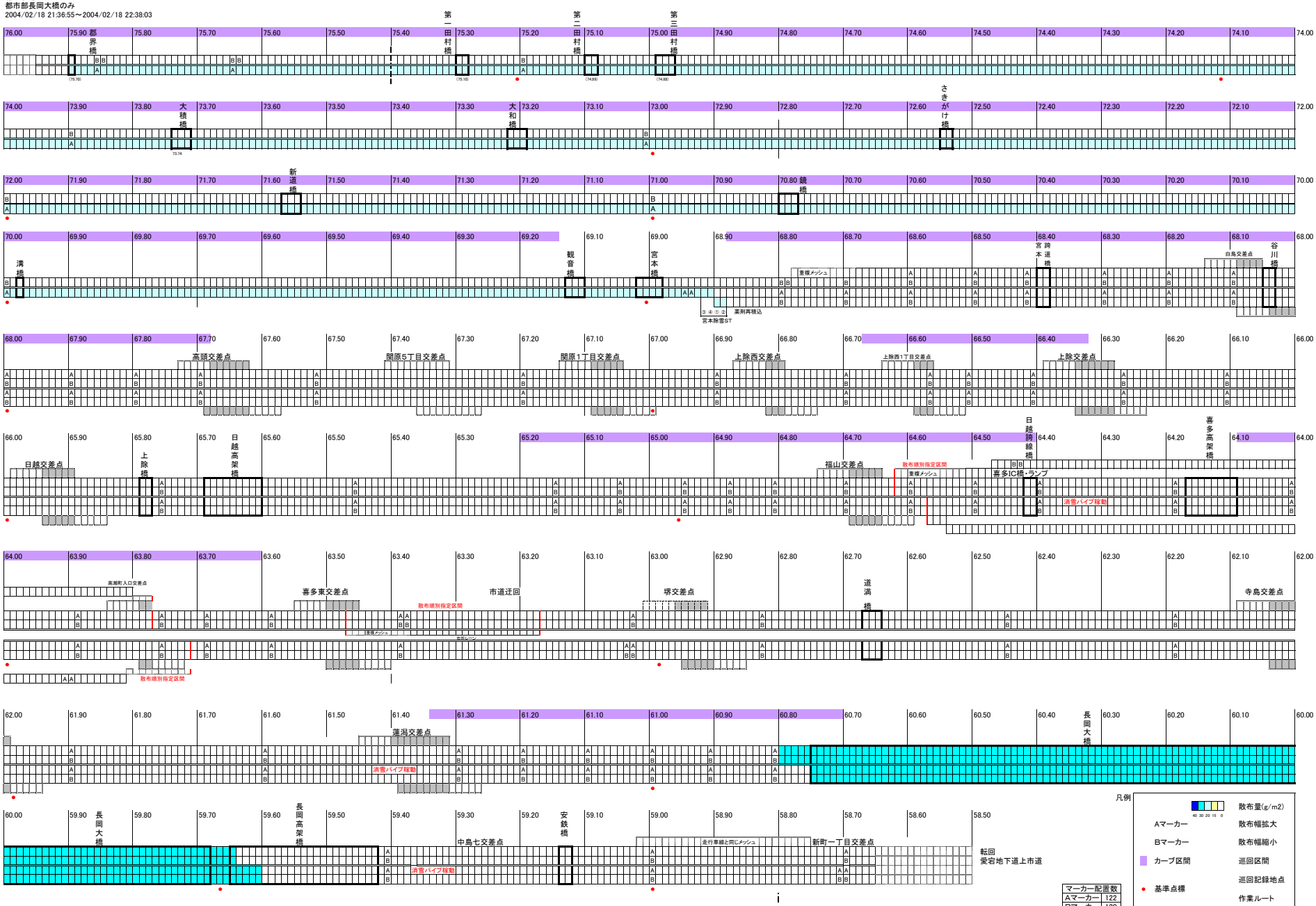


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

自動散布 **散布履歴メッシュ図**

都市部長岡大橋のみ

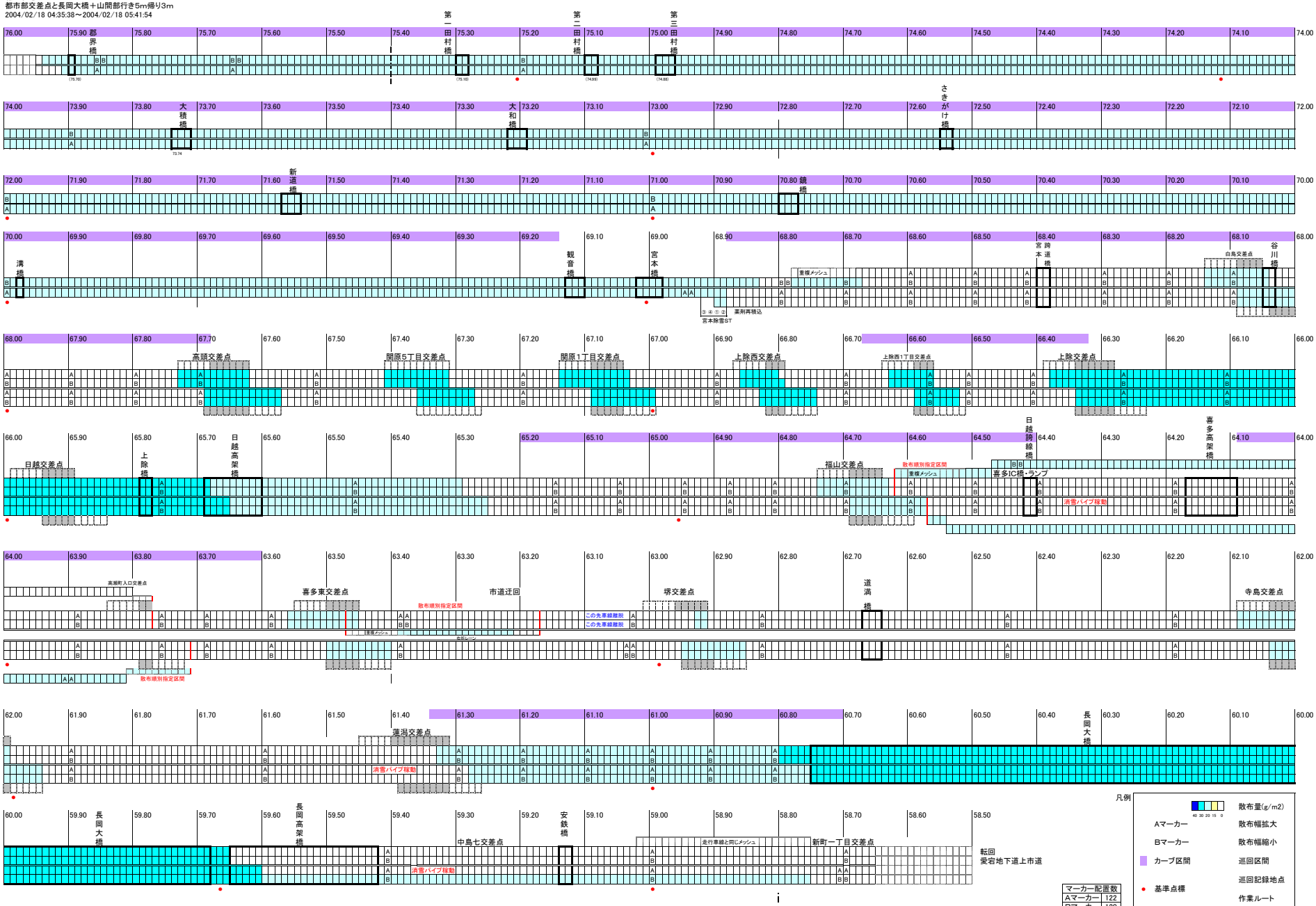
2004/02/18 21:36:55～2004/02/18 22:38:03



自動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り3m

2004/02/18 04:35:38～2004/02/18 05:41:54

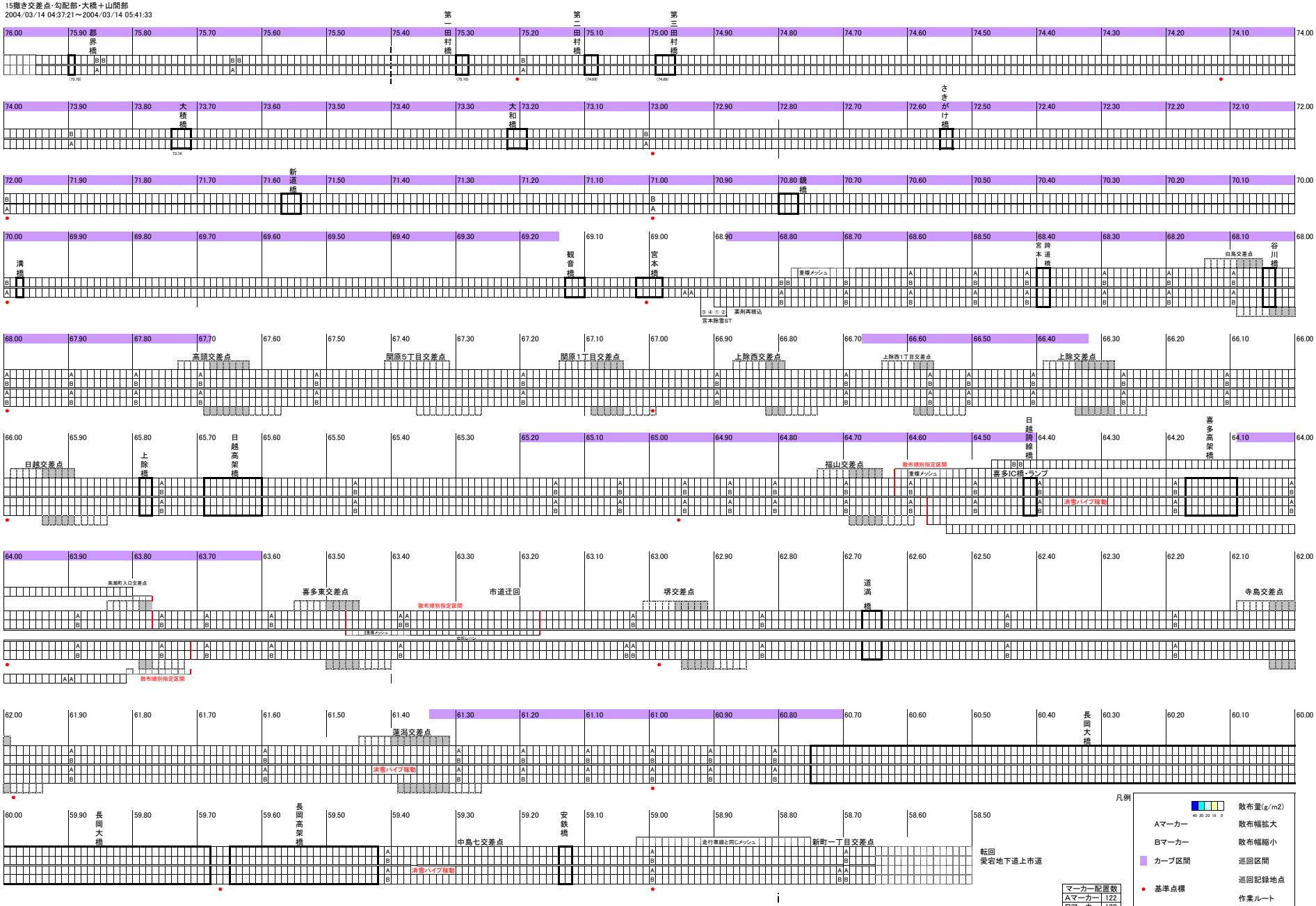


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

15撒き交差点・勾配部・大橋＋山間部

2004/03/14 04:37:21～2004/03/14 05:41:33

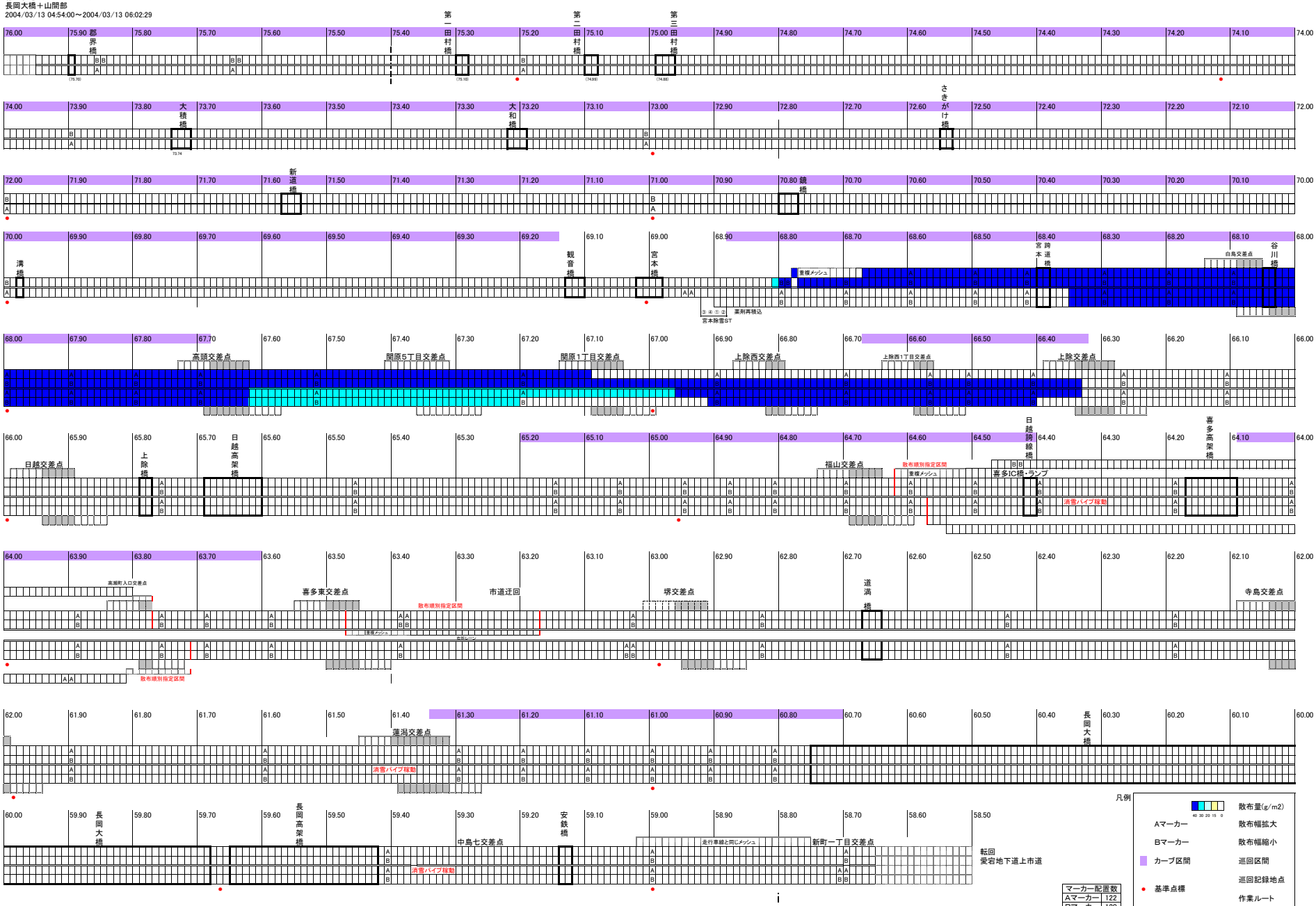


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

長岡大橋＋山間部

2004/03/13 04:54:00～2004/03/13 06:02:29

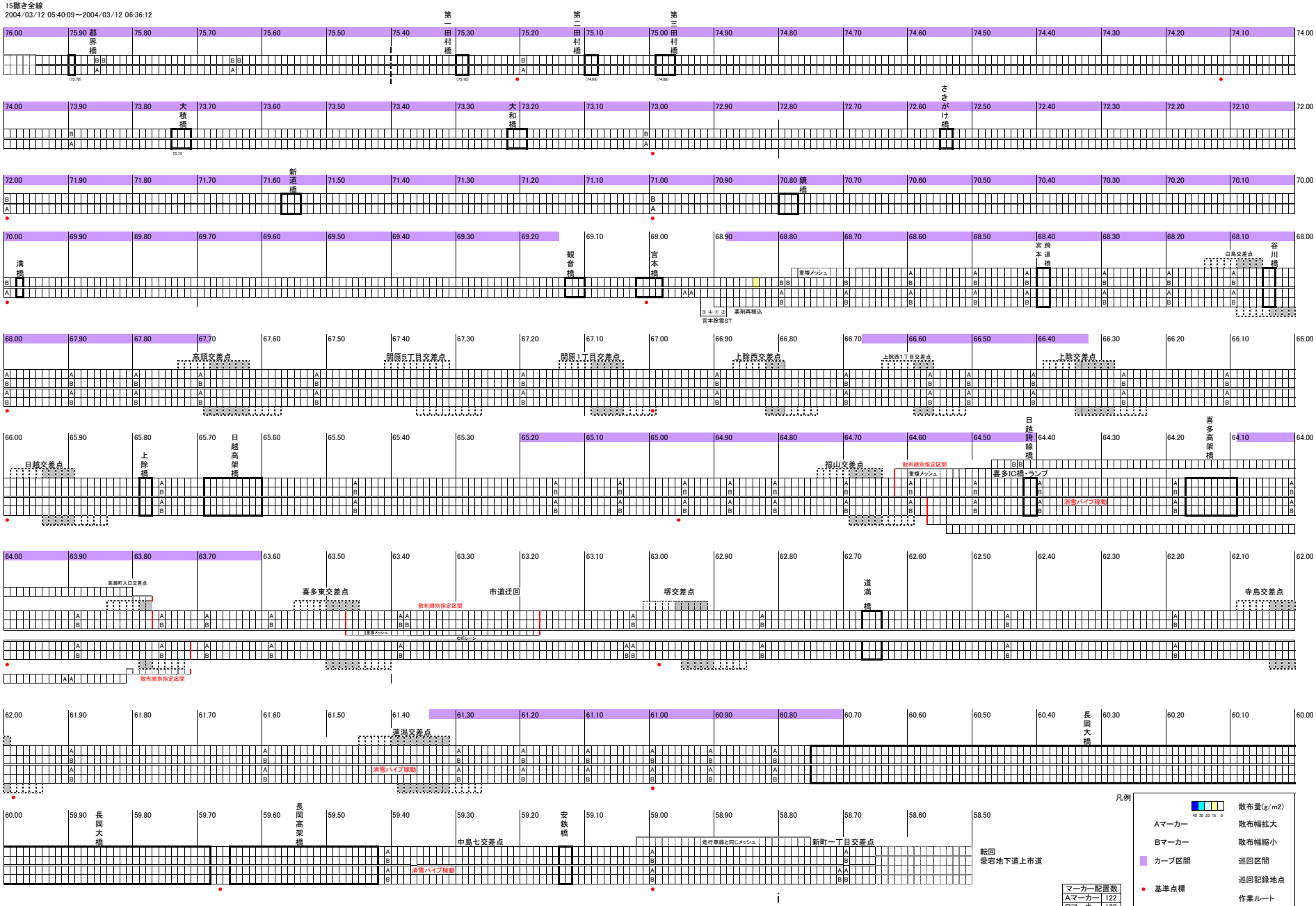


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

15撒き全線

2004/03/12 05:40:09～2004/03/12 06:36:12

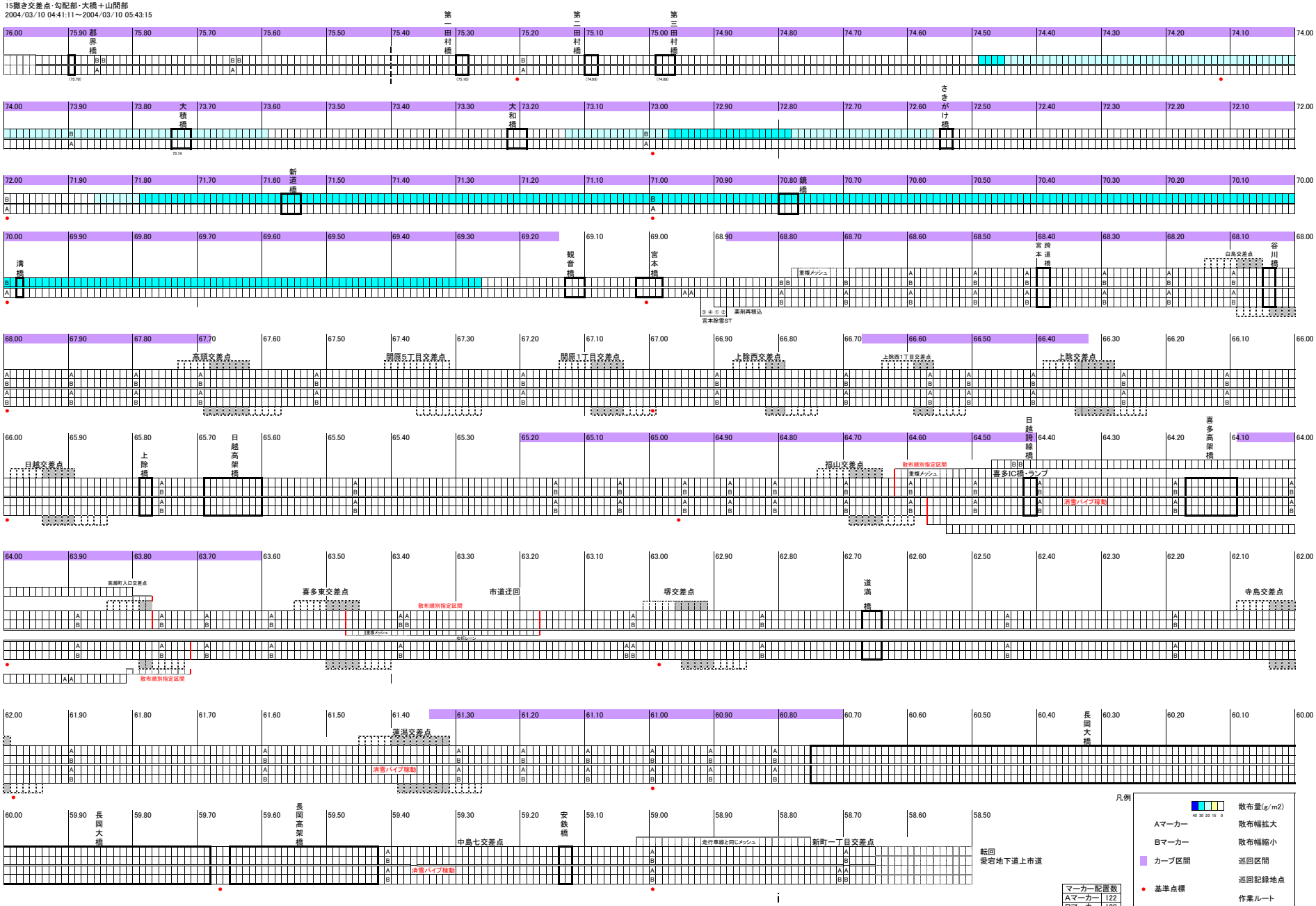


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

15撒き交差点・勾配部・大橋＋山間部

2004/03/10 04:41:11～2004/03/10 05:43:15

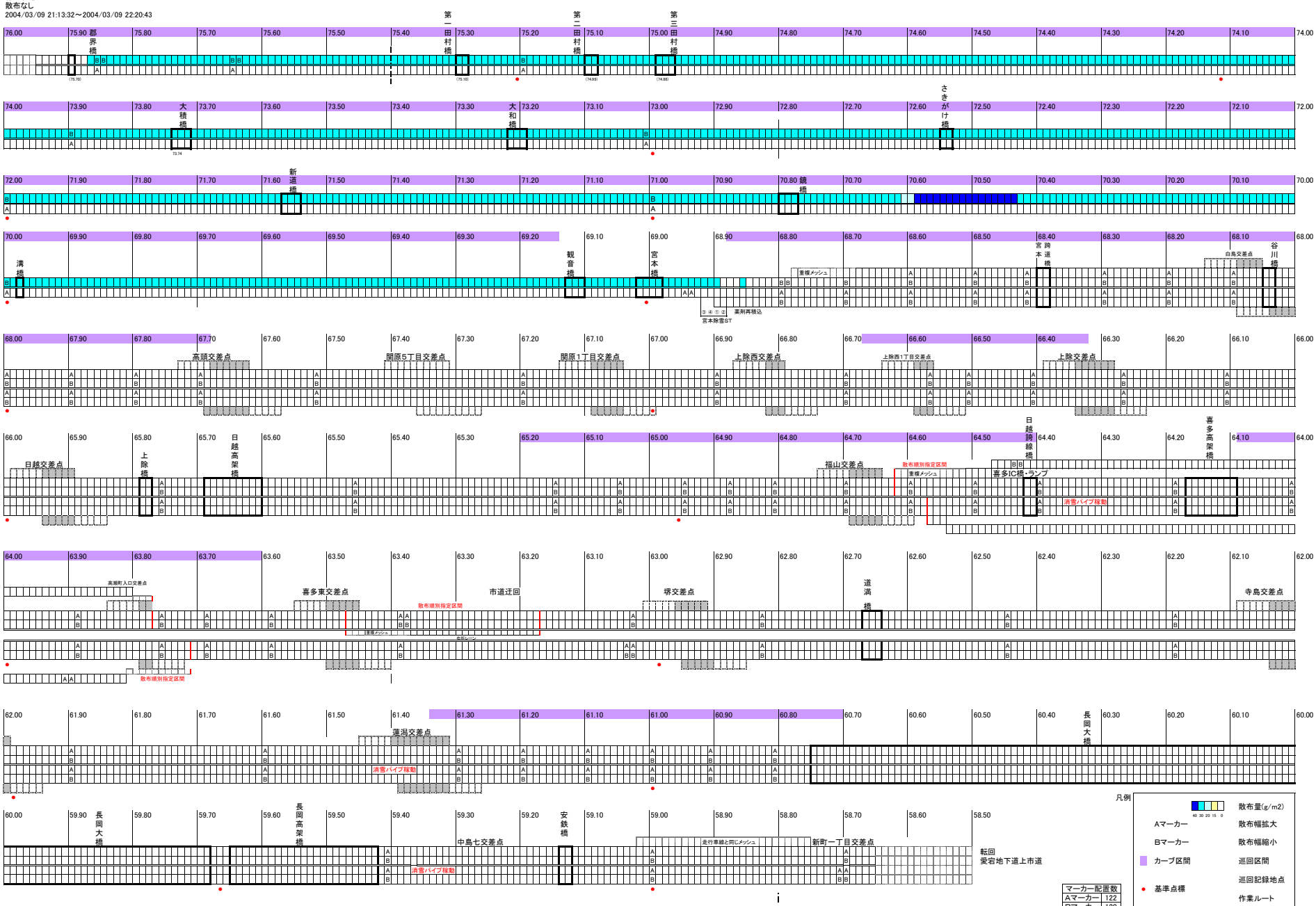


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

散布なし

2004/03/09 21:13:32～2004/03/09 22:20:43

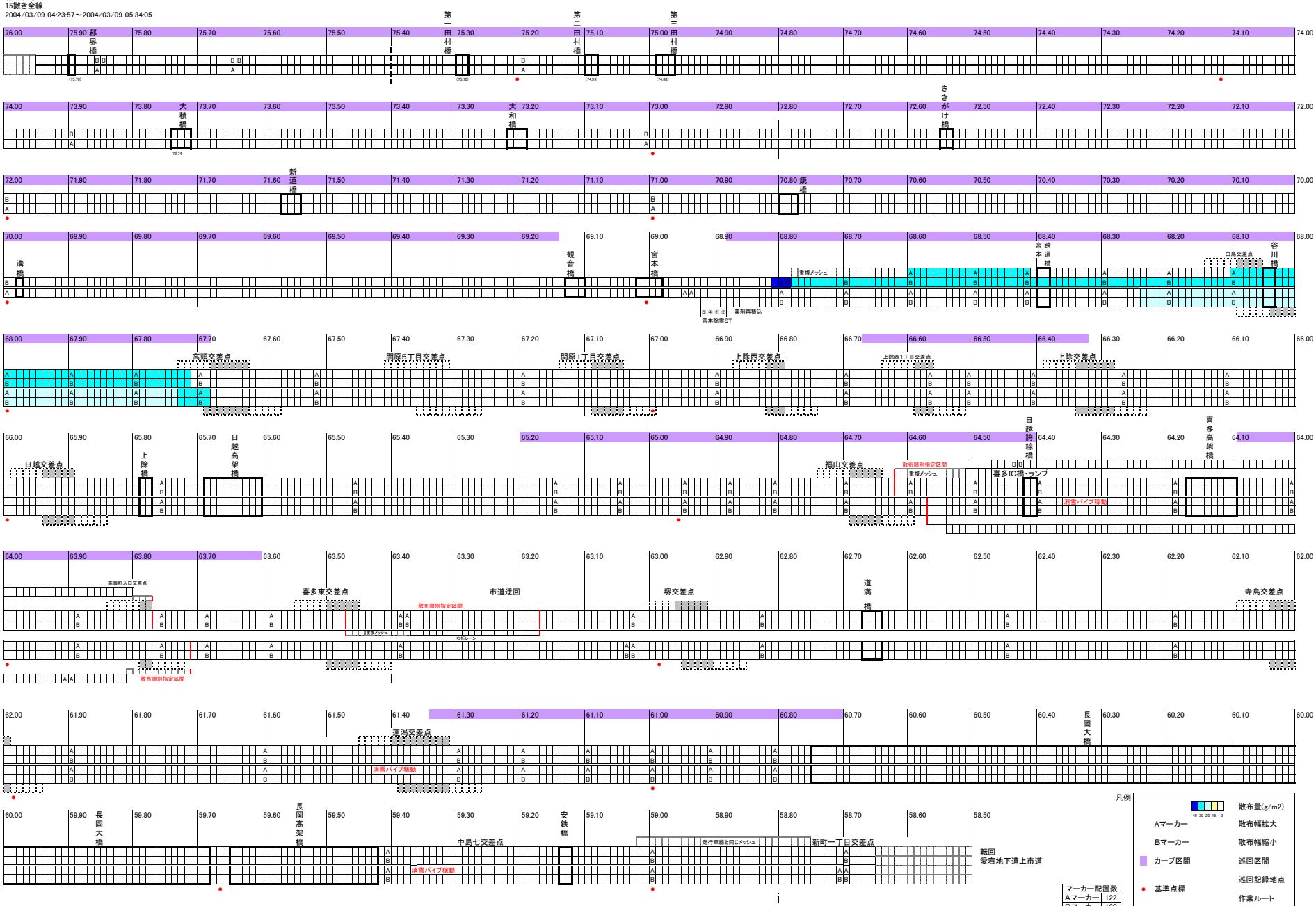


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

15撒き全線

2004/03/09 04:23:57～2004/03/09 05:34:05



凡例

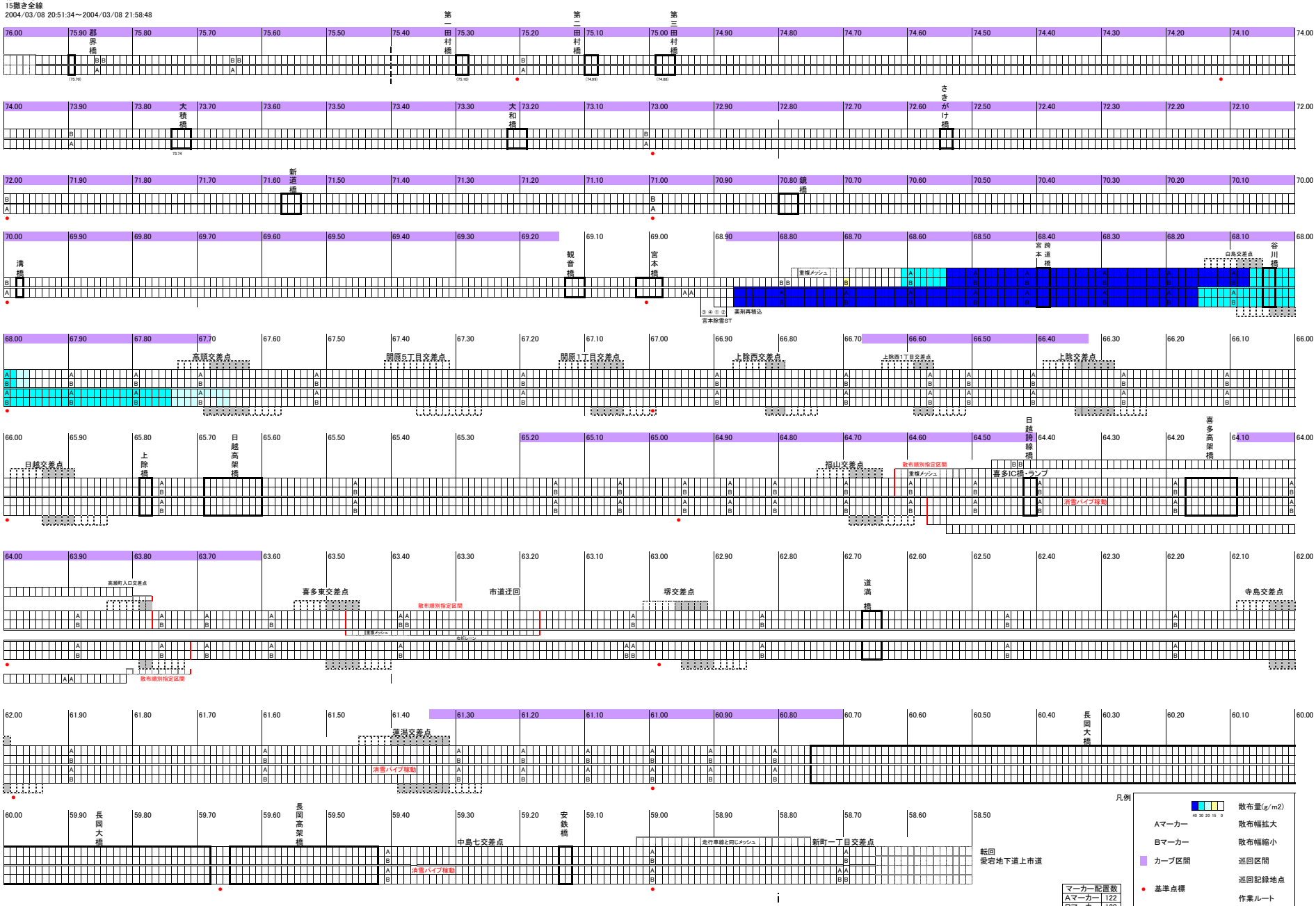
- 散布量(g/m2) 40 30 20 15 0
- Aマーカー
- Bマーカー
- カーブ区間
- 基準点標
- 散布幅拡大
- 散布幅縮小
- 巡回区間
- 巡回記録地点
- 作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

15撒き全線

2004/03/08 20:51:34〜2004/03/08 21:58:48



凡例

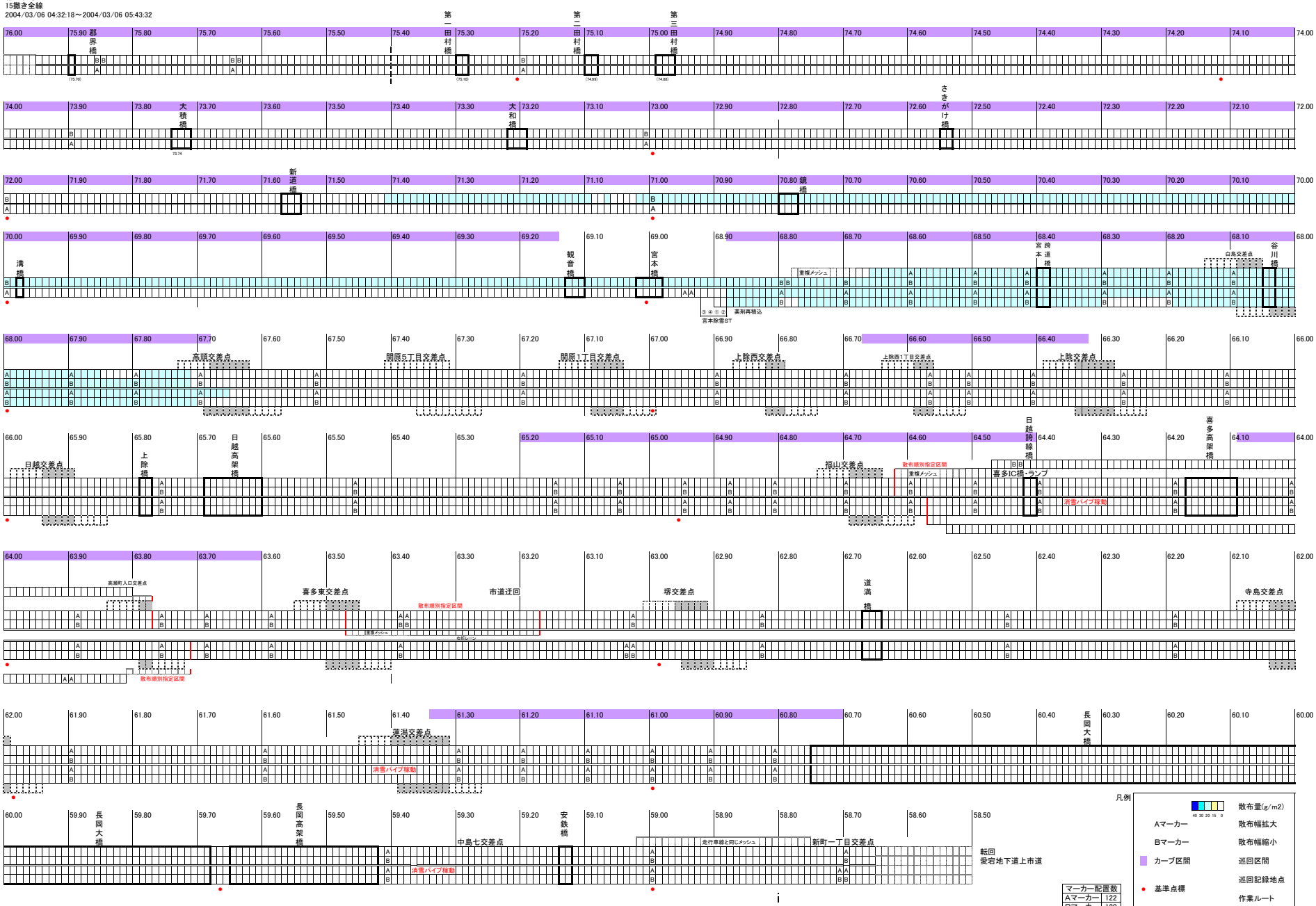
散布量(g/m2)	40 30 20 15 0
Aマーカー	散布幅拡大
Bマーカー	散布幅縮小
カーブ区間	巡回区間
	巡回記録地点
基準点標	作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

15撒き全線

2004/03/06 04:32:18～2004/03/06 05:43:32

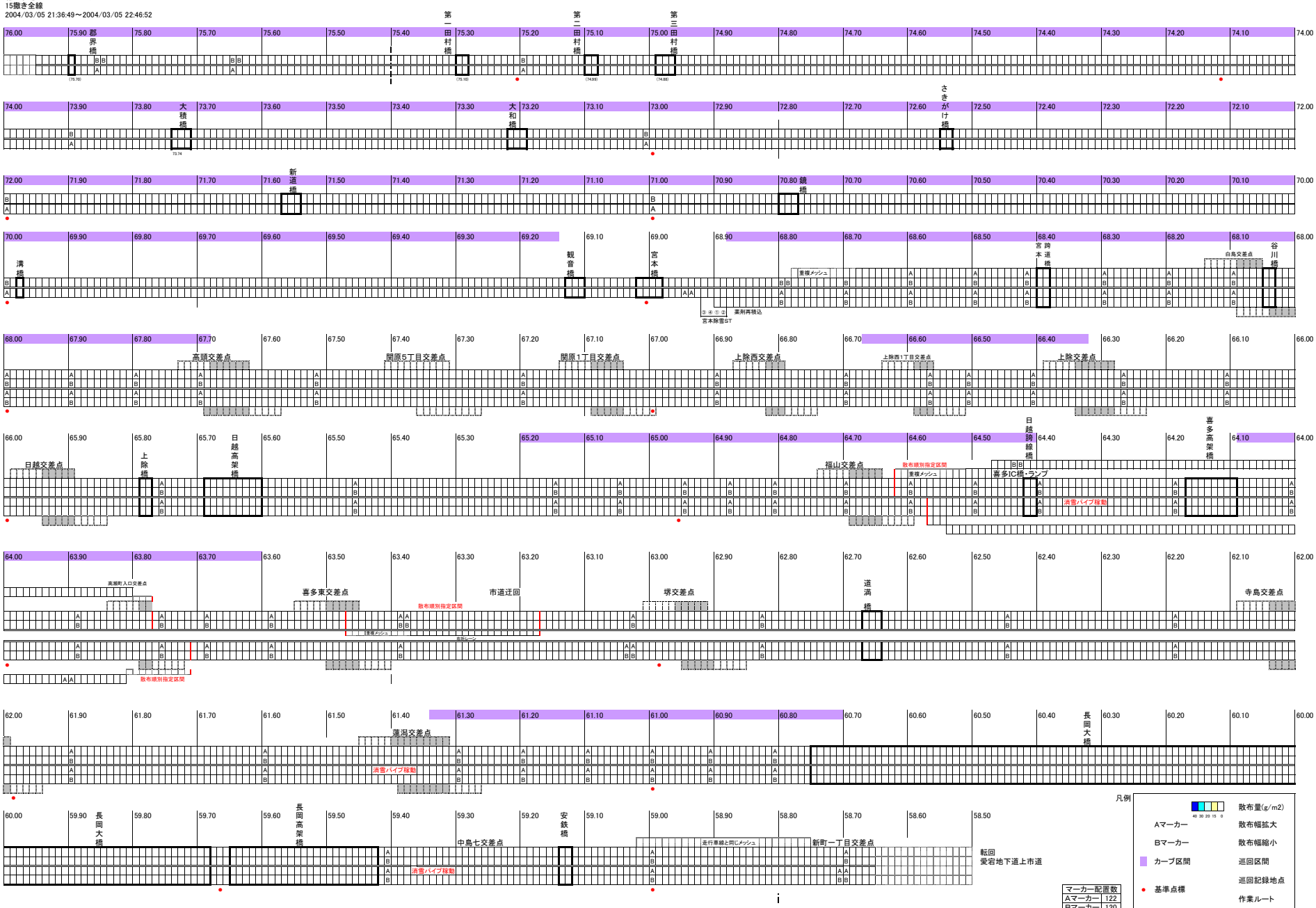


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

15撒き全線

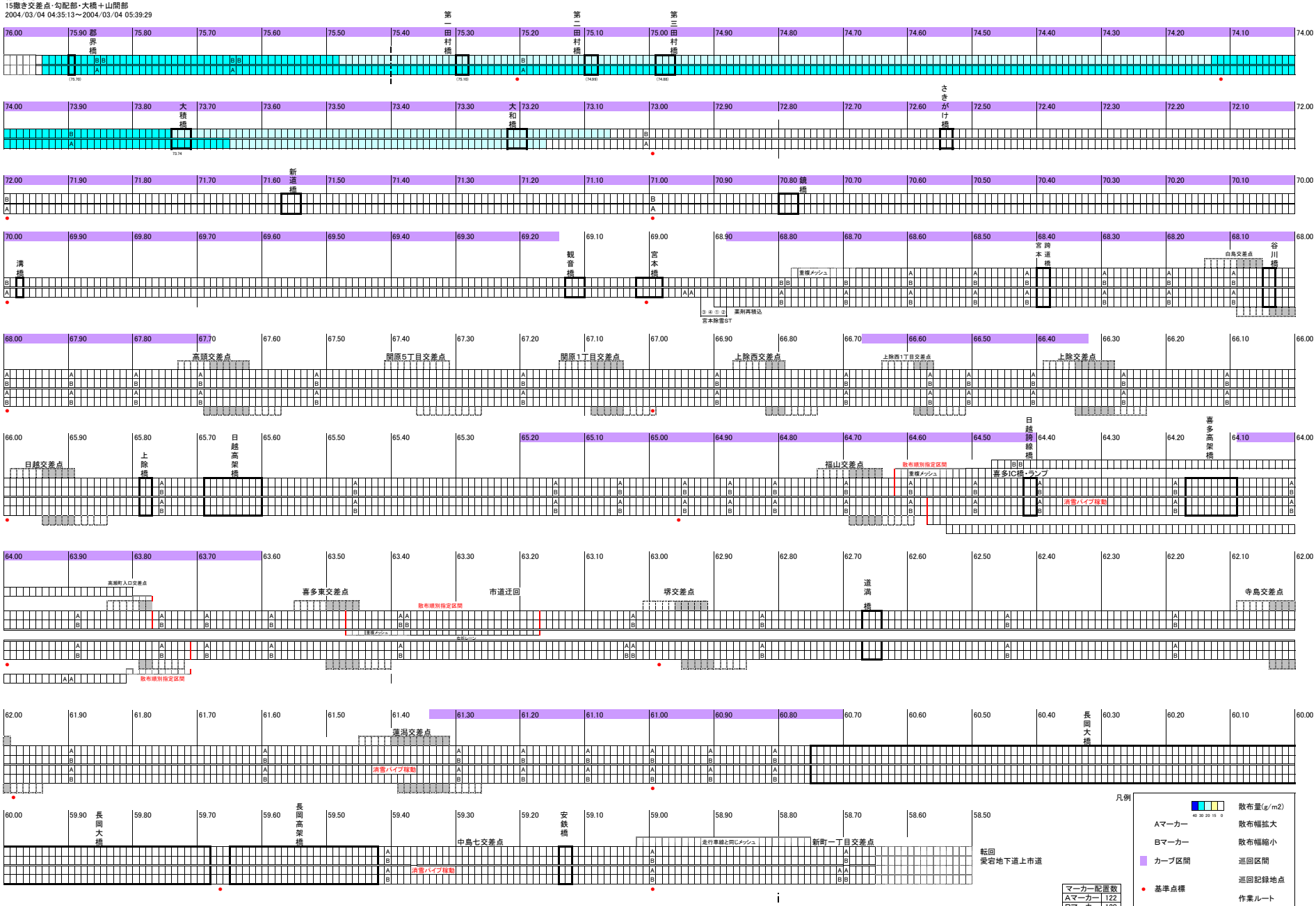
2004/03/05 21:36:49～2004/03/05 22:46:52



手動散布 **散布履歴メッシュ図**

15撒き交差点・勾配部・大橋＋山間部

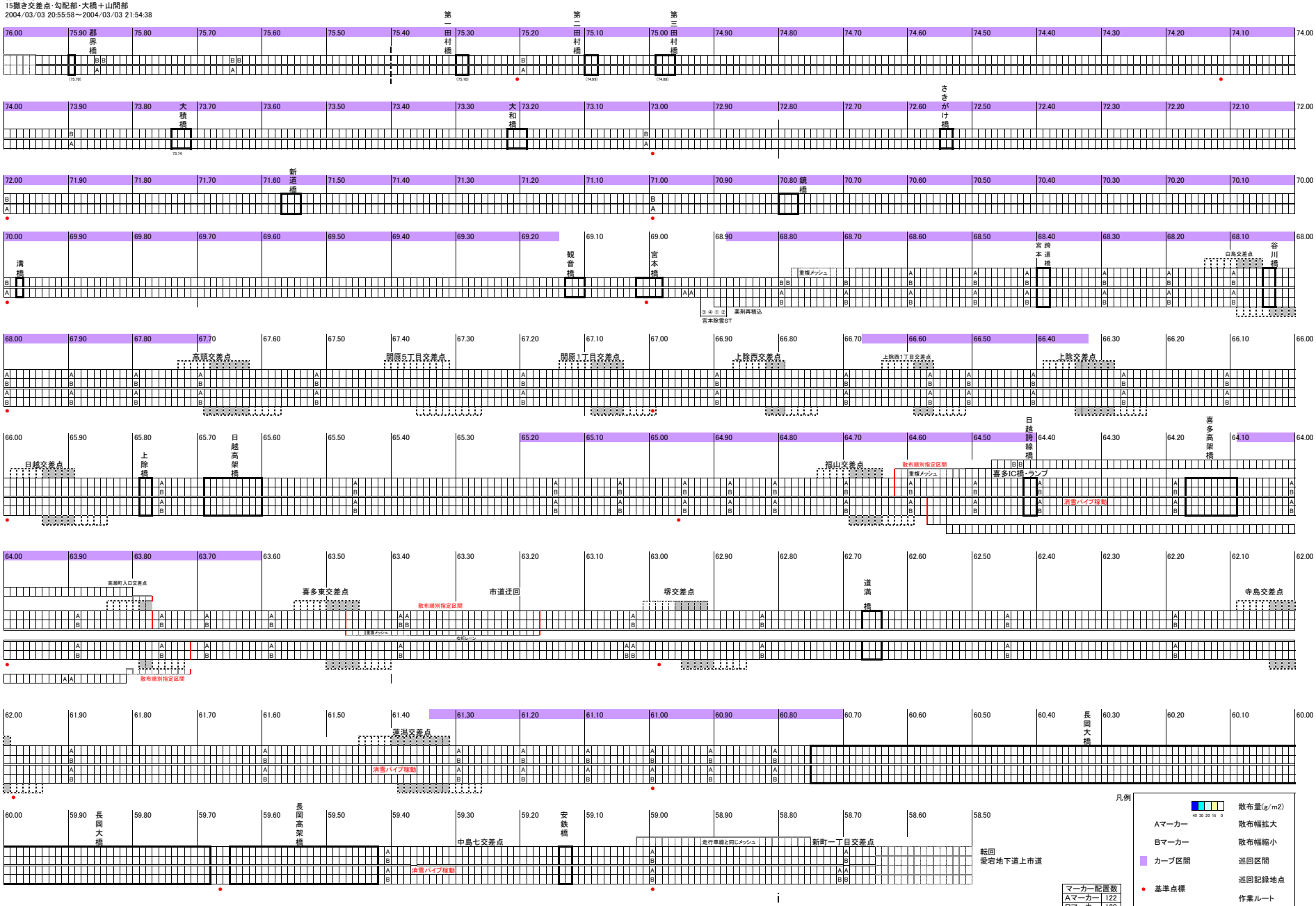
2004/03/04 04:35:13～2004/03/04 05:39:29



手動散布 **散布履歴メッシュ図**

15撒き交差点・勾配部・大橋＋山間部

2004/03/03 20:55:58～2004/03/03 21:54:38



凡例

Aマーカー	散布幅拡大
Bマーカー	散布幅縮小
カーブ区間	巡回区間
基準点標	巡回記録地点
	作業ルート

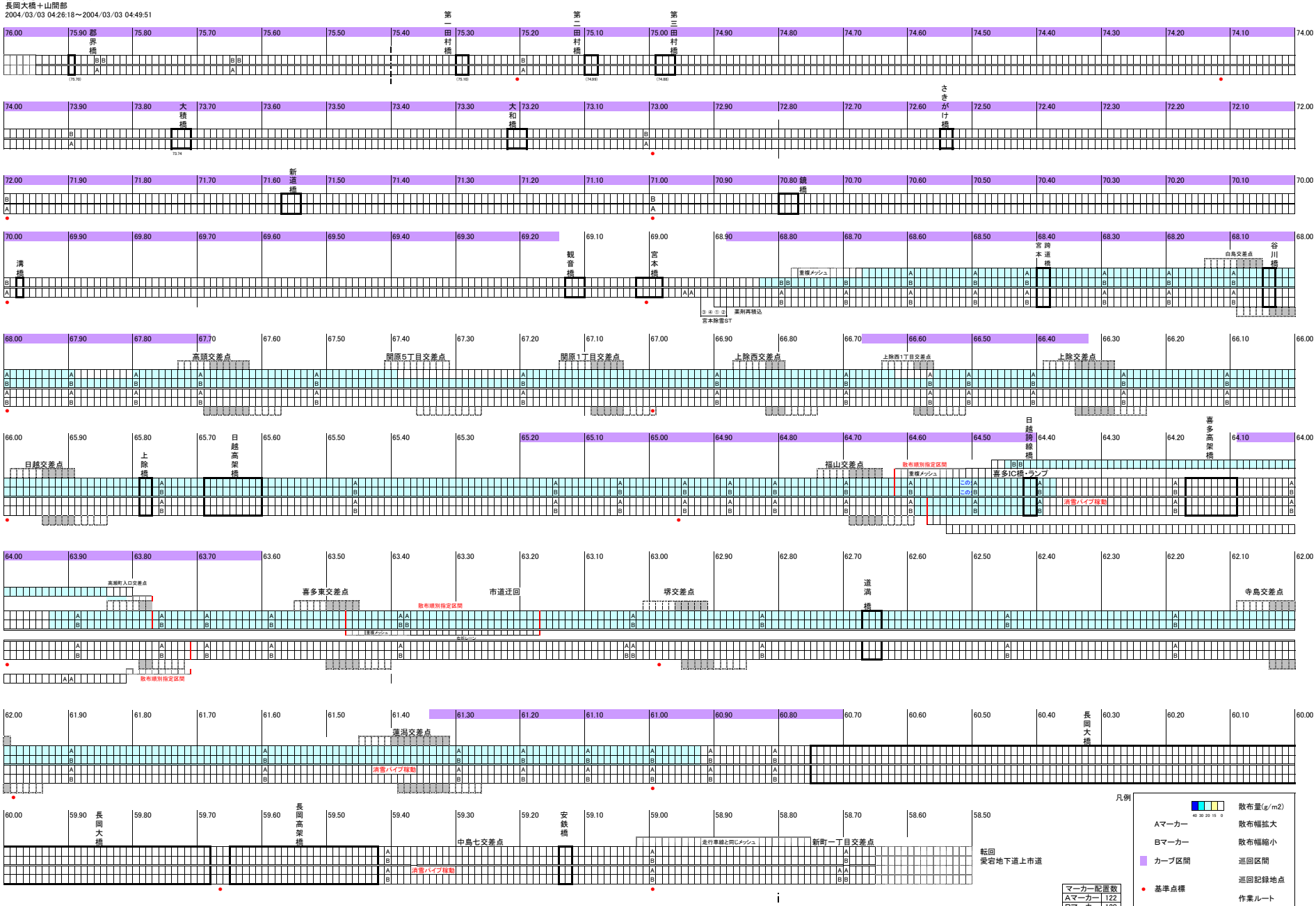
散布量(g/m2) 40 30 20 15 0

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

長岡大橋＋山間部

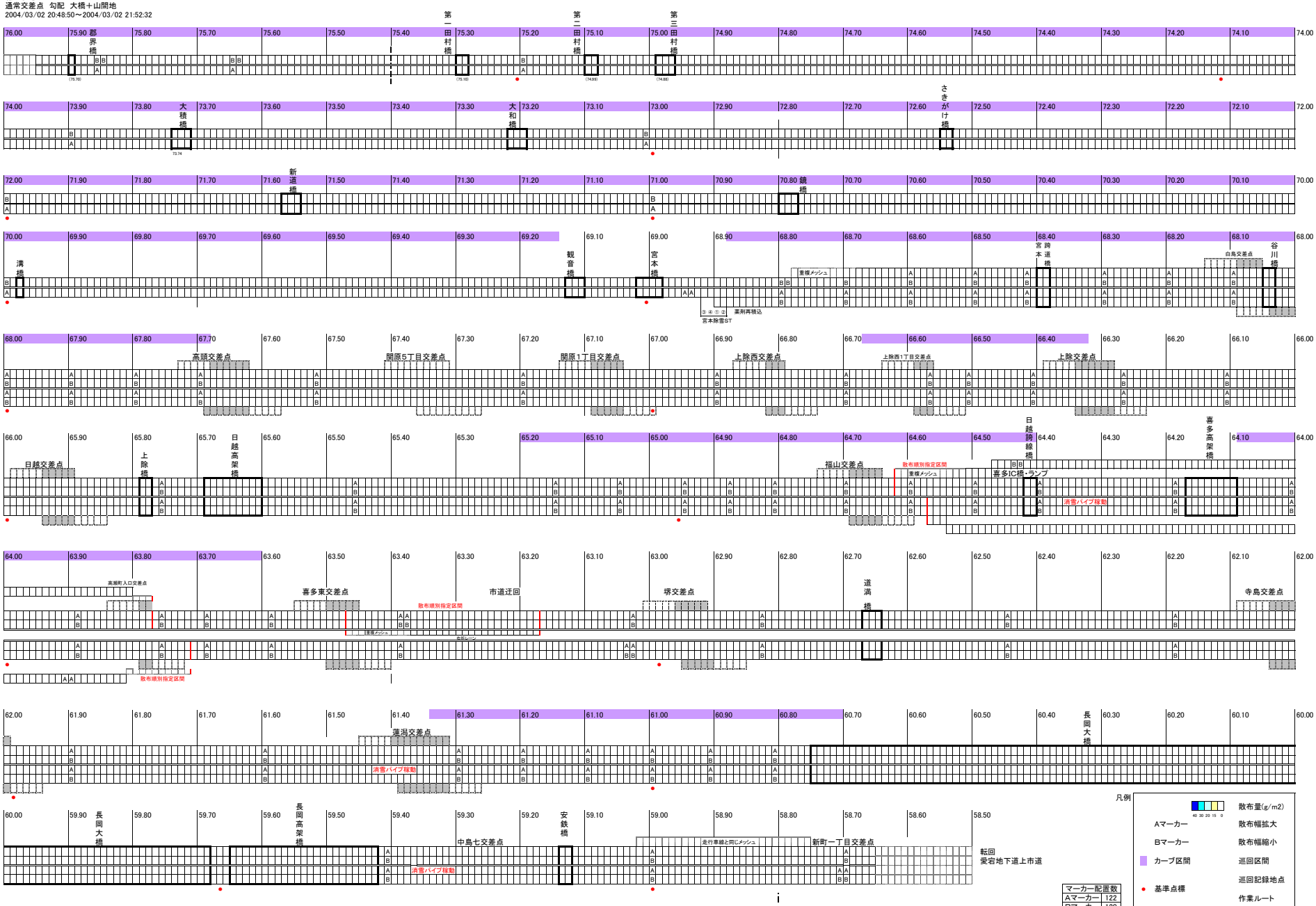
2004/03/03 04:26:18～2004/03/03 04:49:51



手動散布 **散布履歴メッシュ図**

通常交差点　勾配　大橋＋山間地

2004/03/02 20:48:50～2004/03/02 21:52:32



凡例

Aマーカー	散布幅拡大
Bマーカー	散布幅縮小
カーブ区間	巡回区間
基準点標	巡回記録地点
	作業ルート

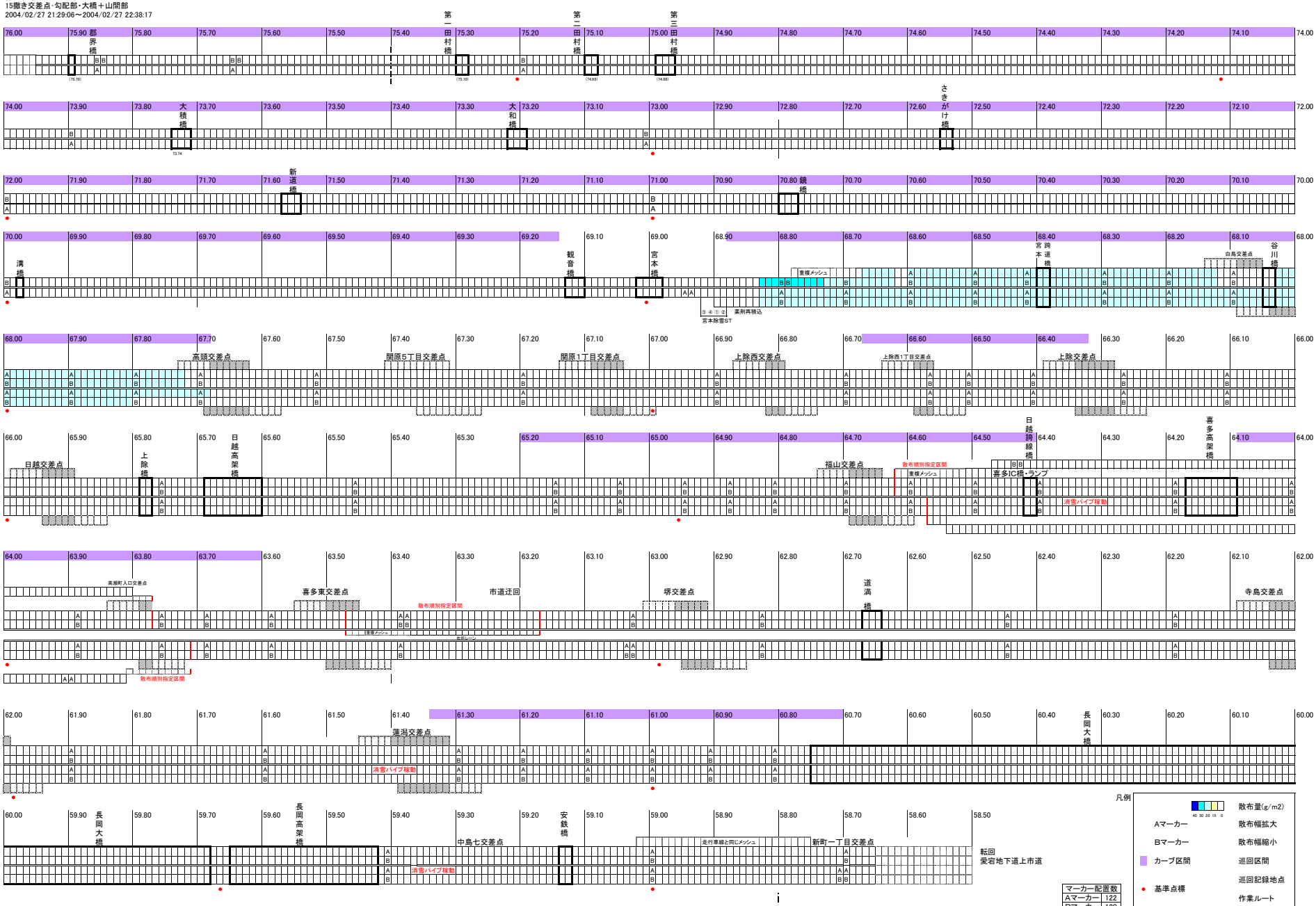
散布量(g/m2)　40 30 20 15 0

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

15撒き交差点・勾配部・大橋＋山間部

2004/02/27 21:29:06～2004/02/27 22:38:17

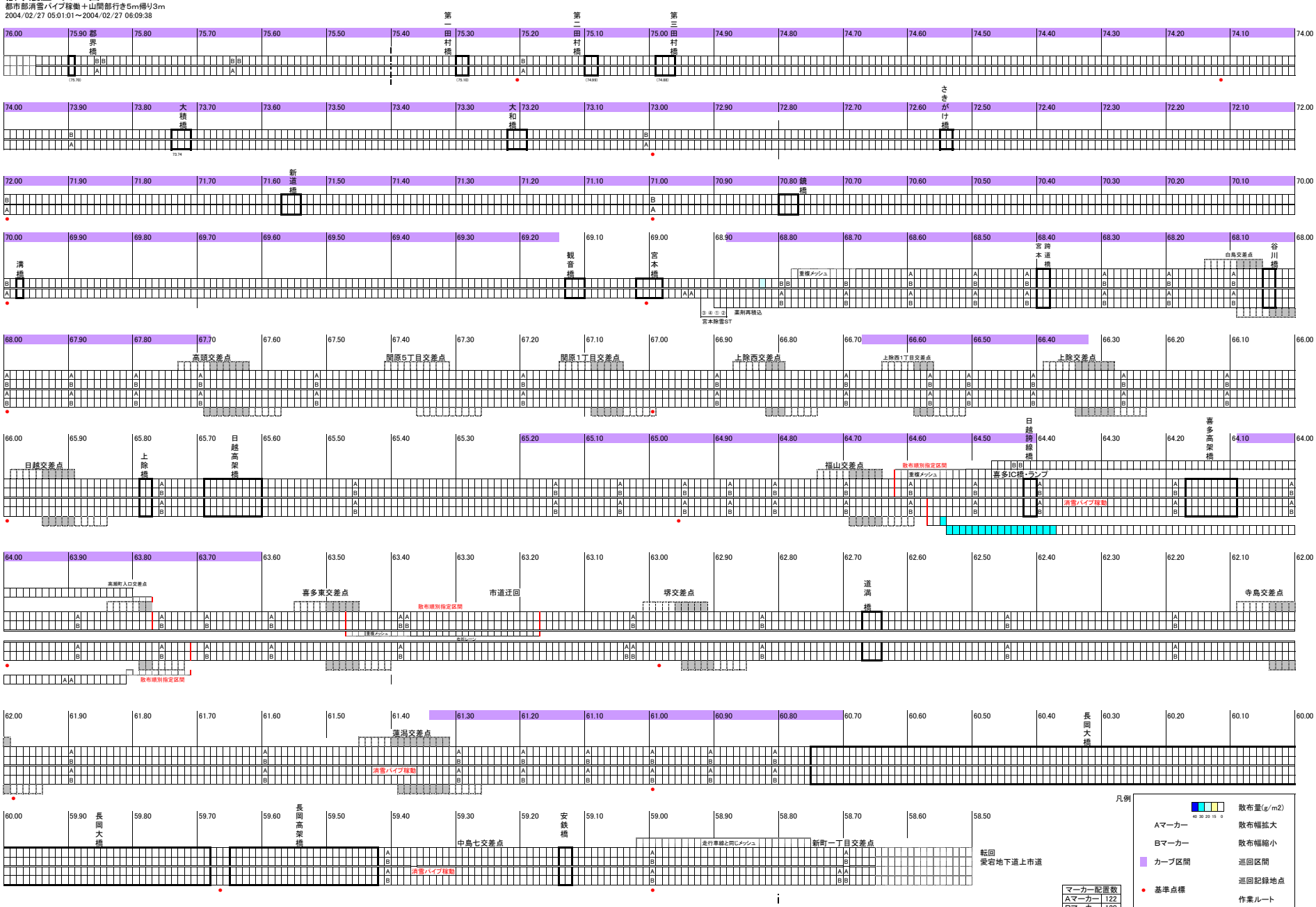


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

都市部消雪パイプ稼働＋山間部行き5m帰り3m

2004/02/27 05:01:01～2004/02/27 06:09:38

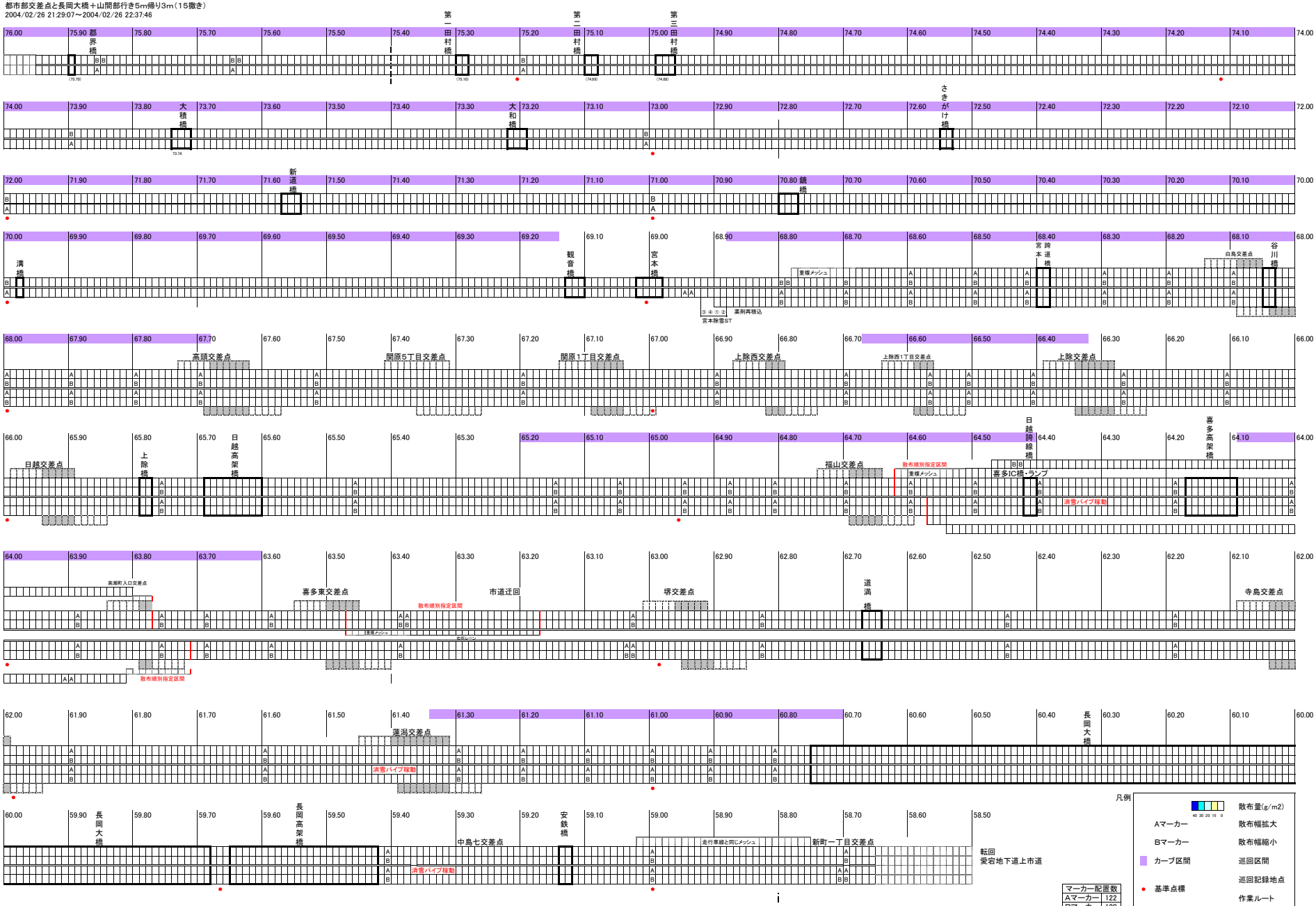


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り3m（15撒き）

2004/02/26 21:29:07～2004/02/26 22:37:46



凡例

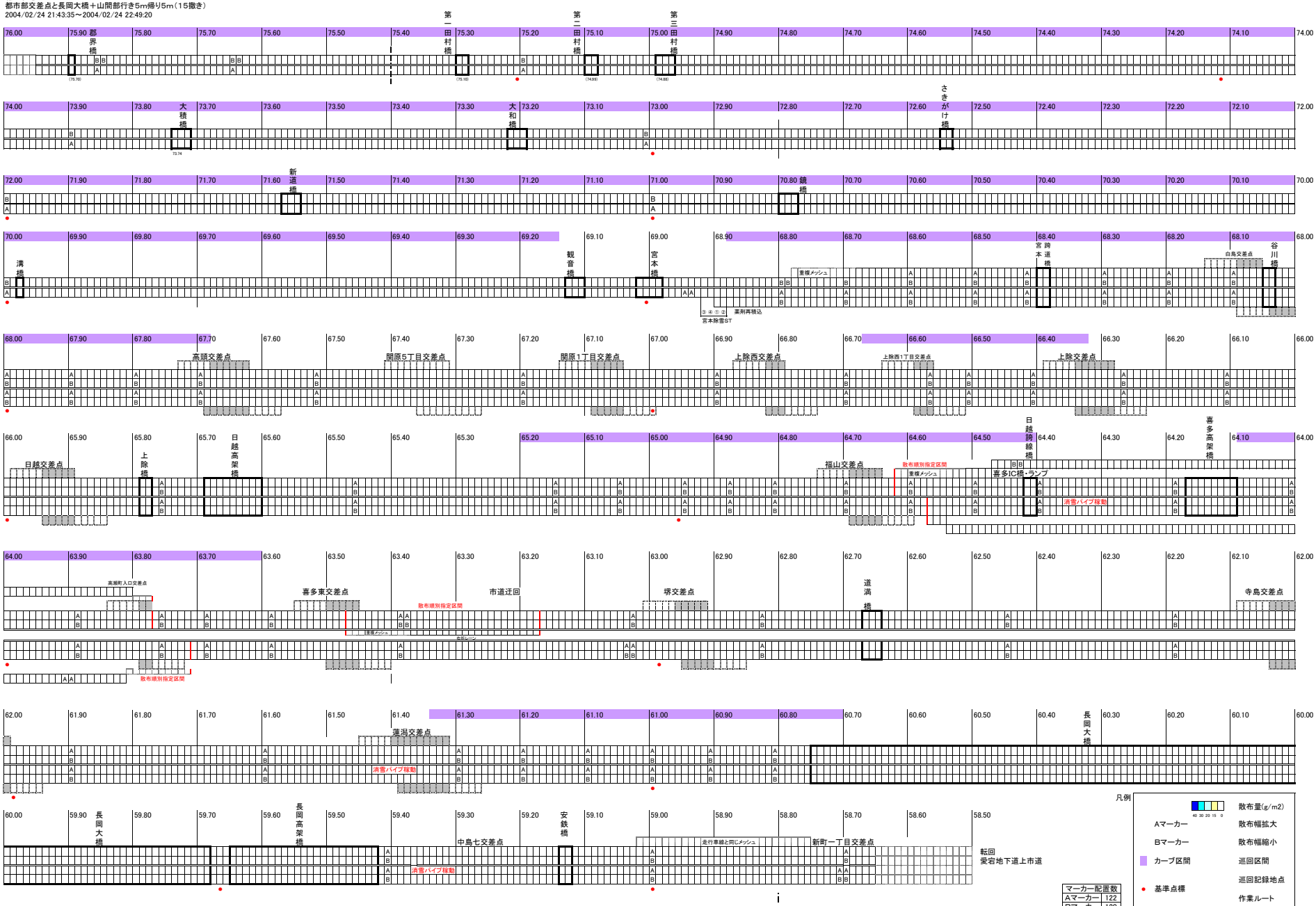
散布量(g/m2)	40 30 20 15 0
Aマーカー	散布幅拡大
Bマーカー	散布幅縮小
カーブ区間	巡回区間
	巡回記録地点
基準点標	作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り5m（15撒き）

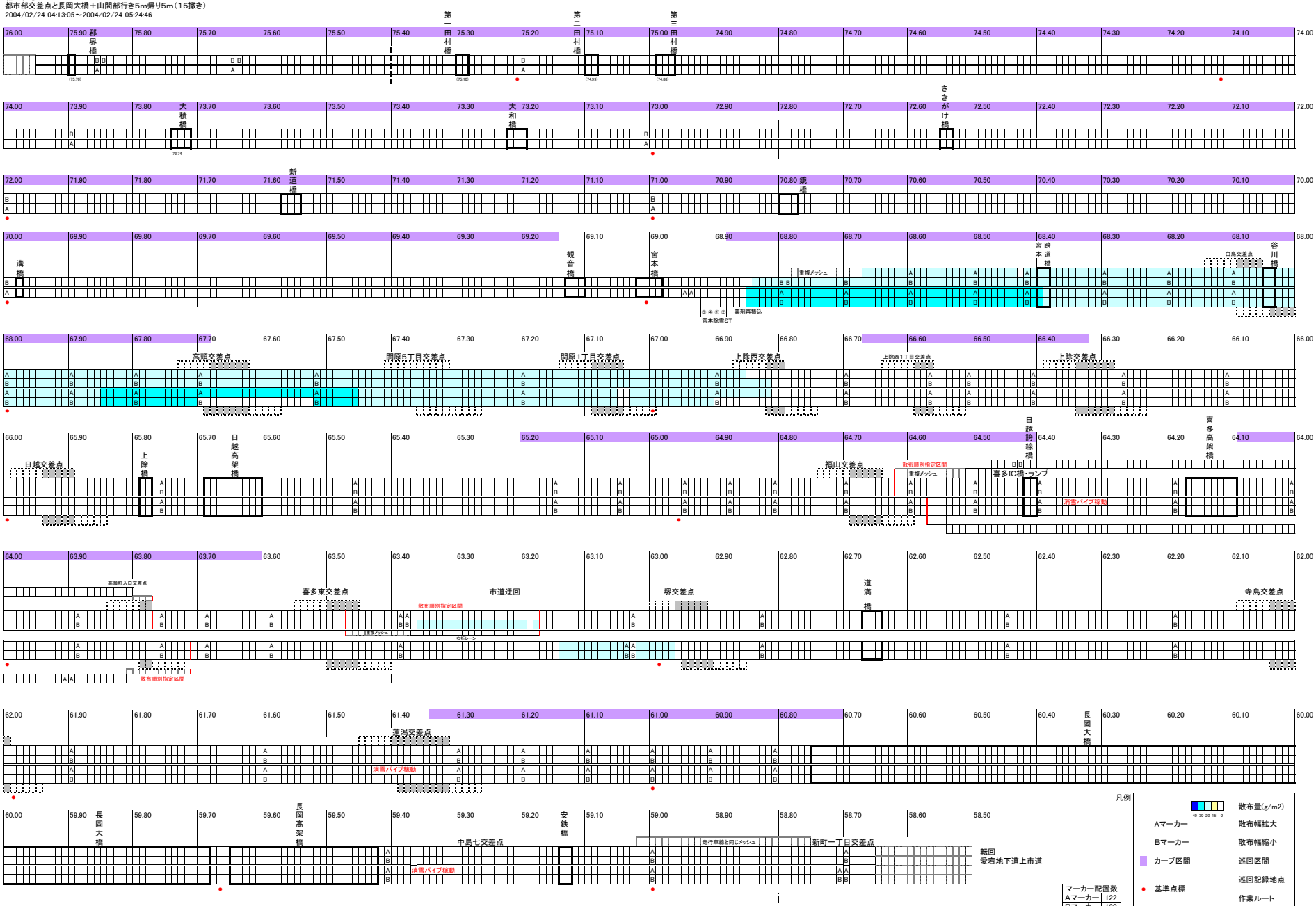
2004/02/24 21:43:35～2004/02/24 22:49:20



手動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り5m（15撒き）

2004/02/24 04:13:05～2004/02/24 05:24:46

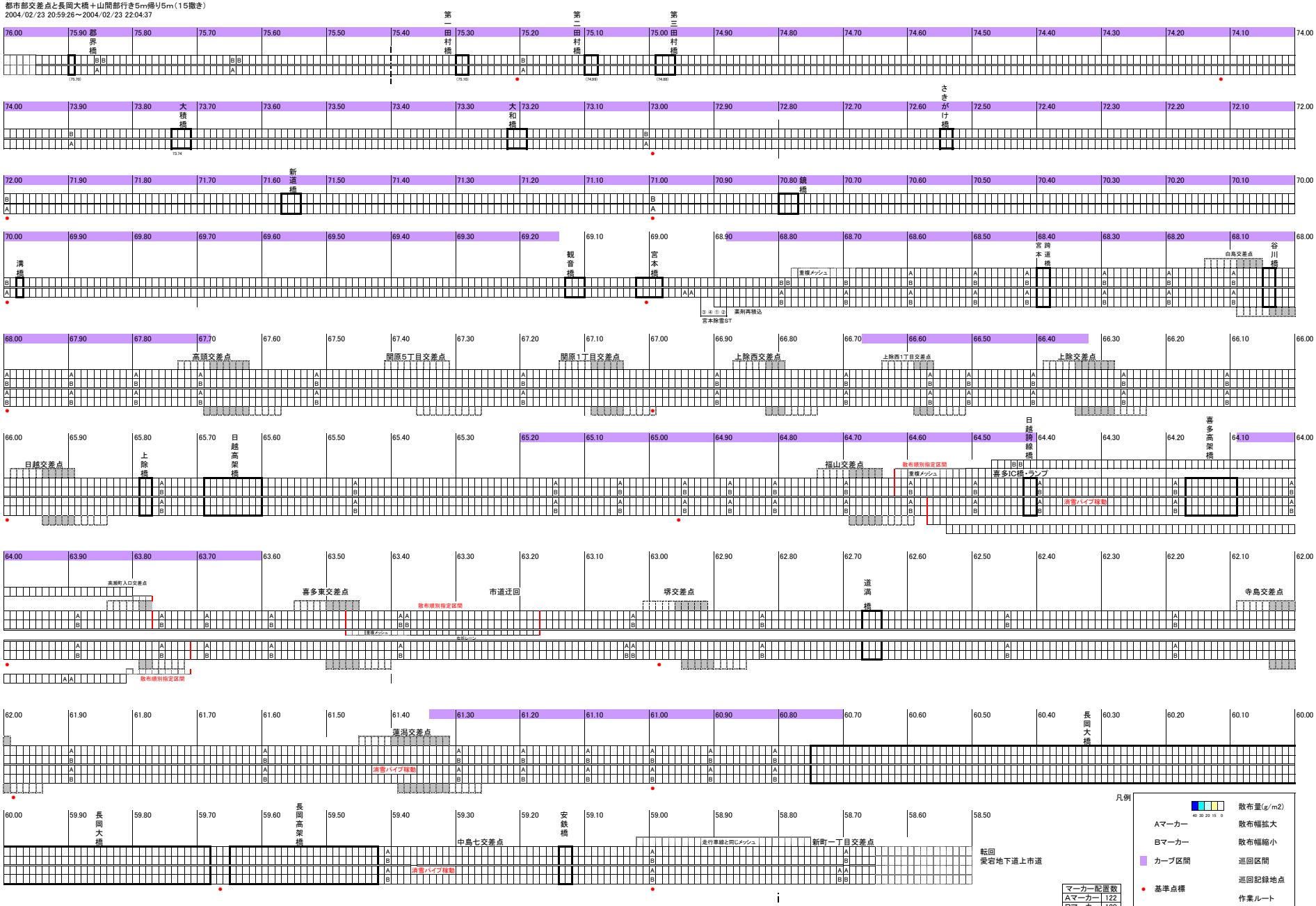


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り5m（15撒き）

2004/02/23 20:59:26～2004/02/23 22:04:37



凡例

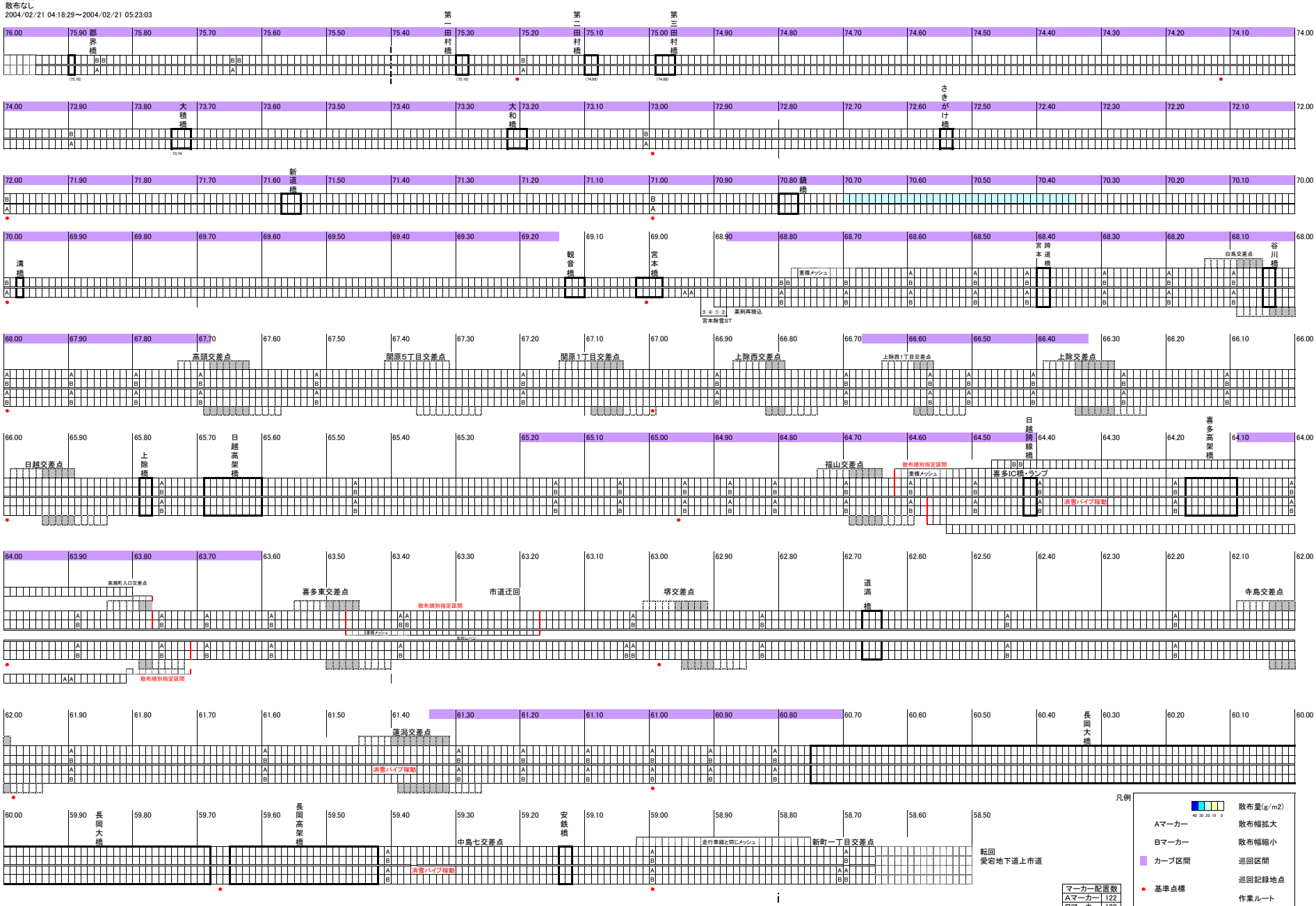
散布量(g/m2)	40 30 20 15 0
Aマーカー	散布幅拡大
Bマーカー	散布幅縮小
カーブ区間	巡回区間
	巡回記録地点
基準点標	作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

散布なし

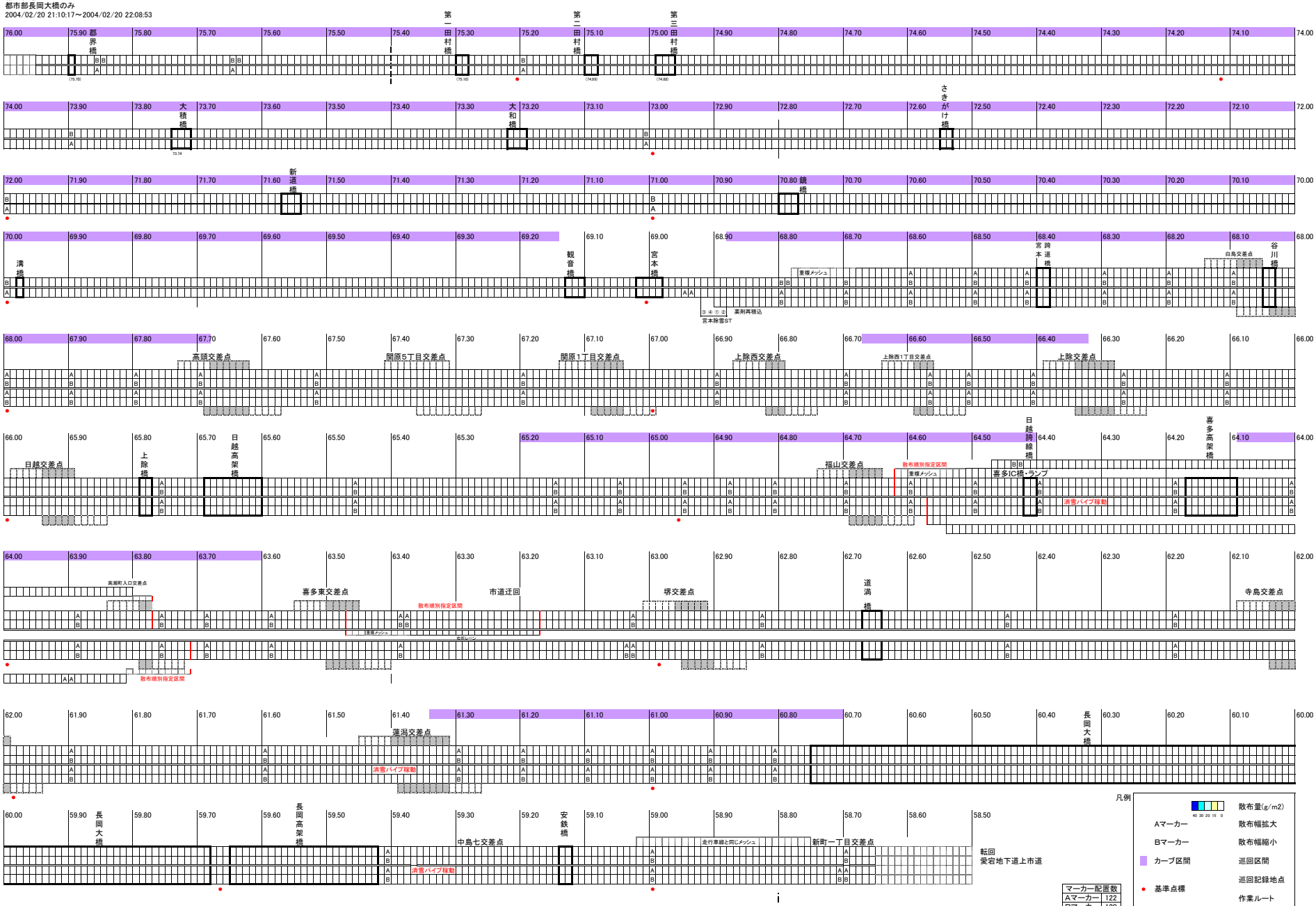
2004/02/21 04:18:29～2004/02/21 05:23:03



手動散布 **散布履歴メッシュ図**

都市部長岡大橋のみ

2004/02/20 21:10:17～2004/02/20 22:08:53

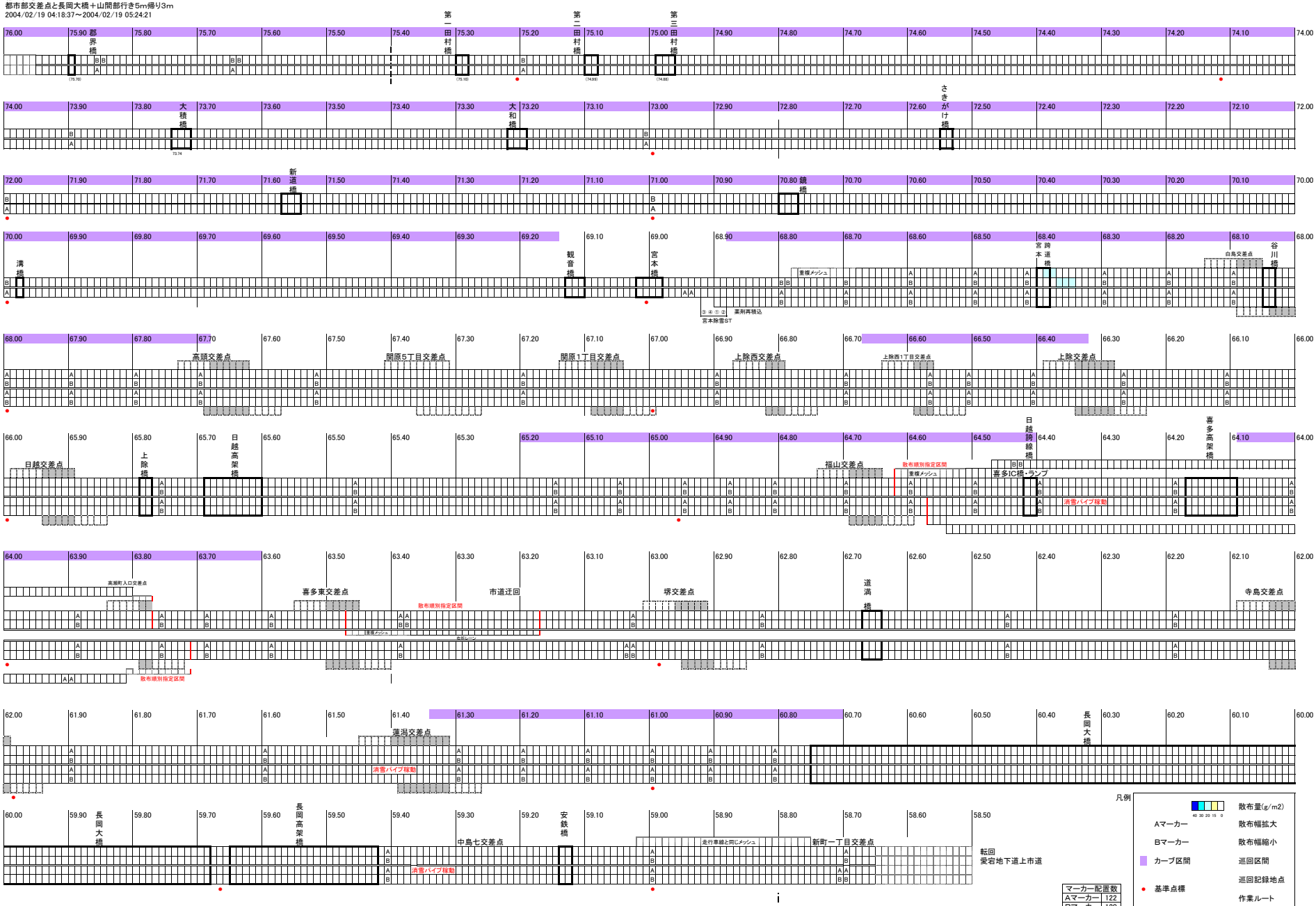


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り3m

2004/02/19 04:18:37～2004/02/19 05:24:21

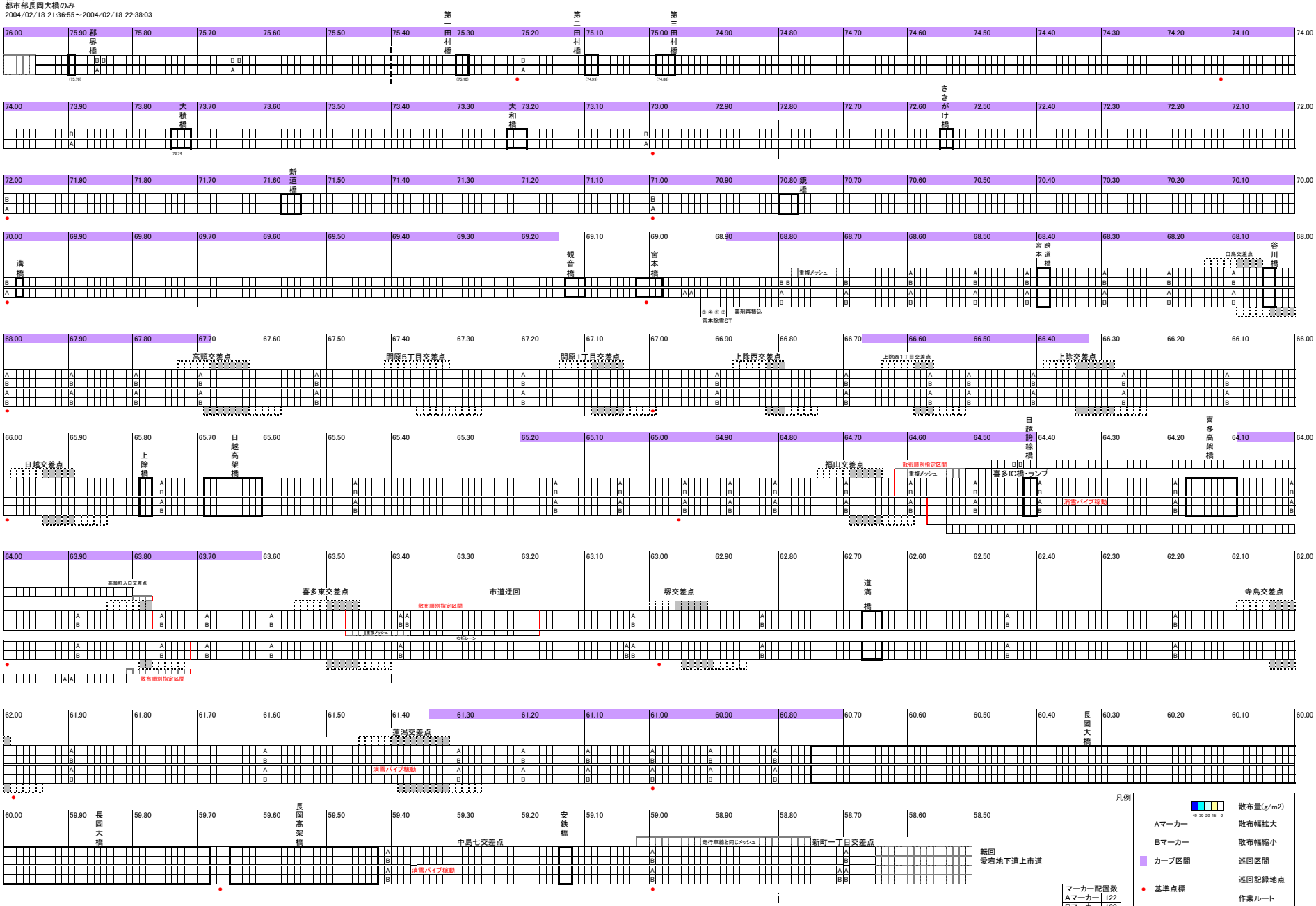


マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

都市部長岡大橋のみ

2004/02/18 21:36:55～2004/02/18 22:38:03



凡例

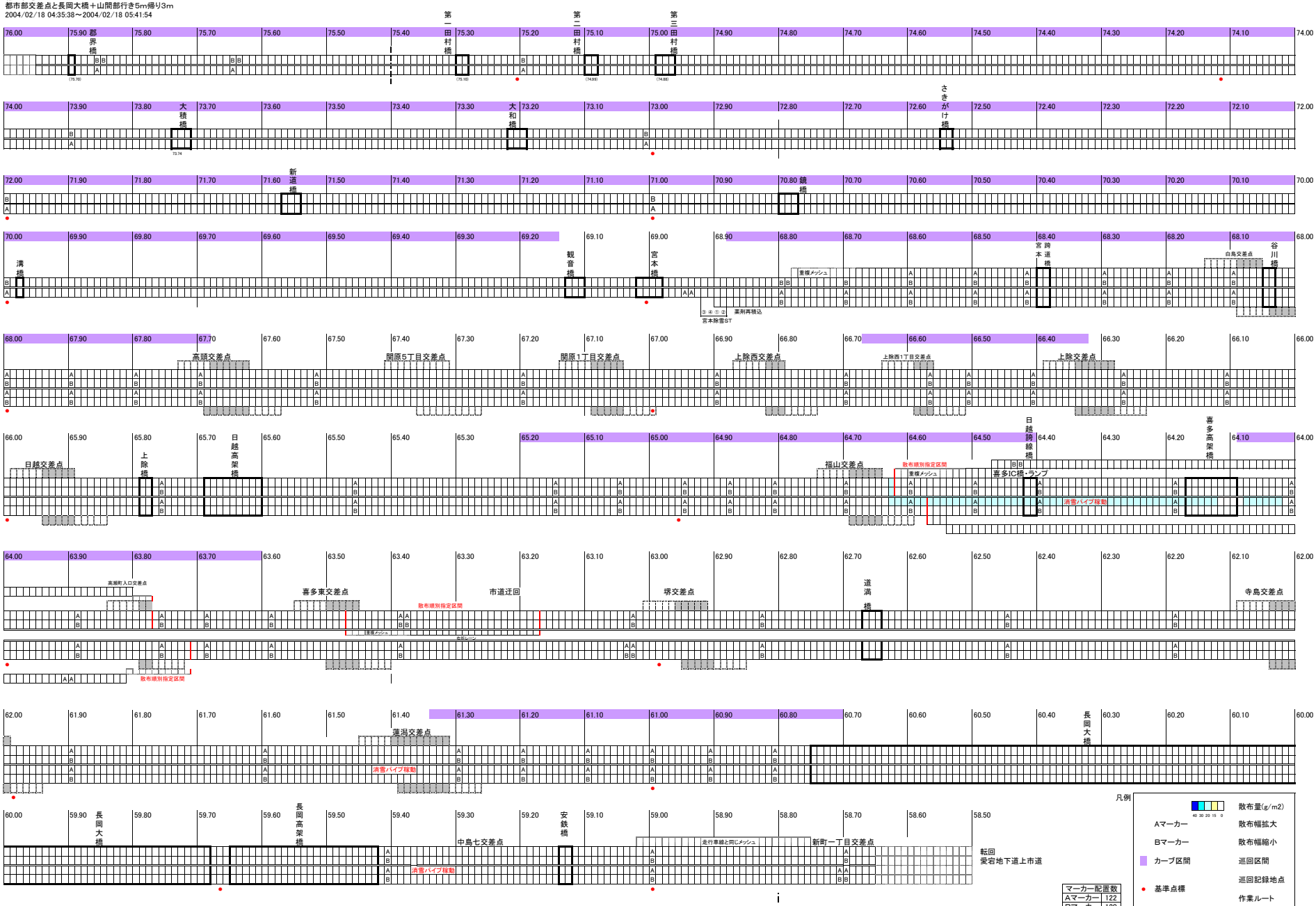
散布量(g/m2)	40 30 20 15 0
Aマーカー	散布幅拡大
Bマーカー	散布幅縮小
カーブ区間	巡回区間
	巡回記録地点
基準点標	作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

手動散布 **散布履歴メッシュ図**

都市部交差点と長岡大橋＋山間部行き5m帰り3m

2004/02/18 04:35:38～2004/02/18 05:41:54



凡例

散布量(g/m2) 40 30 20 15 0	
Aマーカー	散布幅拡大
Bマーカー	散布幅縮小
カーブ区間	巡回区間
	巡回記録地点
基準点標	作業ルート

マーカー配置数	
Aマーカー	122
Bマーカー	120
計	242

別添資料５　道路巡回日報

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月14日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時37分　帰着時間 5時41分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時40分	5時18分	4時43分	4時51分	4時57分	5時11分	5時02分	5時05分	5時22分	5時40分	5時26分	5時36分	5時29分	5時33分
天　候	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り
気　温	1.8度	1.8度	2.4度	2.2度	2.8度	2.2度	2.2度	2.4度	0.8度	0.5度	0.5度	0.4度	0.4度	0.4度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC	喜多IC	蓮潟	蓮潟	水道町	西神田2			大積1	大積1		
							中島7	中島7						
							西神田2	水道町						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線上下	発見及び発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 佐藤 吉一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月13日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 21時36分　帰着時間 21時36分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	21時38分	22時10分	21時43分	22時06分	21時48分	22時02分	21時53分	21時56分	22時13分	22時30分	22時17分	22時27分	22時20分	22時24分
天　候	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り
気　温	2.0度	2.1度	2.1度	2.1度	2.3度	2.4度	2.6度	2.4度	1.3度	0.7度	0.4度	0.4度	0.2度	0.2度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 佐藤 吉一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月13日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時54分　帰着時間 6時02分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時57分	5時32分	5時01分	5時27分	5時08分	5時22分	5時13分	5時16分	5時36分	5時54分	5時40分	5時51分	5時44分	5時47分
天　候	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴
気　温	2.4度	1.6度	2.5度	2.1度	2.7度	2.4度	3.6度	3.3度	0.1度	0.0度	−0.4度	−0.4度	−0.6度	−0.5度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 千野 征夫 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月12日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 21時16分　帰着時間 21時17分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	21時41分		21時37分		21時19分	21時33分	21時24分	21時27分	21時44分	22時04分	21時48分	22時00分	21時53分	21時56分
天　候	晴		晴		晴	晴	晴	晴	晴	晴	晴	晴	晴	晴
気　温	4.6度		5.0度		5.4度	4.6度	7.0度	6.1度	3.7度	3.0度	3.2度	2.6度	3.0度	2.7度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	新雪　cm 圧雪　cm	乾燥 新雪　0 cm 圧雪　0 cm	新雪　cm 圧雪　cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	その他	良好 その他	その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ： 塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 千野 征夫 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月12日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 5時40分　帰着時間 6時36分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	5時42分	6時12分	5時47分	6時08分	5時51分	6時04分	5時55分	5時59分	6時16分	6時34分	6時20分	6時30分	6時24分	6時27分
天　候	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り
気　温	4.7度	4.9度	4.4度	3.9度	4.8度	4.4度	5.2度	5.1度	4.8度	5.3度	3.7度	3.0度	2.8度	2.8度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ： 塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 保坂 浩史 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月10日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時41分　帰着時間 5時43分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時43分	5時20分	4時47分	4時48分	5時01分	5時13分	5時06分	5時08分	5時23分	5時42分	5時27分	5時38分	5時31分	5時34分
天　候	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴
気　温	3.0度	3.3度	3.9度	3.3度	3.4度	1.4度	1.9度	1.3度	-0.4度	-1.4度	-2.0度	-2.6度	-2.6度	-2.6度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備					喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×
									中島7 ×	中島7 ×				
									西神田2 ×	水道町 ×				
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線上下	発見及び発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 村山　清一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月9日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 21時13分　帰着時間 22時20分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	21時16分	21時49分	21時21分	21時44分	21時25分	21時39分	21時30分	21時33分	21時58分	22時19分	22時03分	22時15分	22時07分	22時10分
天　候	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴
気　温	4.1度	5.0度	4.8度	4.7度	4.7度	4.1度	4.6度	4.8度	2.4度	1.1度	1.2度	0.7度	0.7度	0.5度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 村山　清一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月9日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時23分　帰着時間 5時34分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時26分	5時08分	4時29分	4時33分	4時42分	4時54分	4時47分	4時50分	5時12分	5時29分	5時15分	5時25分	5時19分	5時22分
天　候	晴	晴	晴	晴	晴	晴	晴	晴	晴	曇り	晴	曇り	晴	晴
気　温	1.5度	0.9度	1.7度	1.3度	1.8度	1.1度	2.5度	2.3度	0.5度	0.4度	0.0度	0.1度	0.0度	0.3度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 西川　修三 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月8日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 20時51分　帰着時間 21時58分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	20時54分	21時30分	20時59分	21時27分	21時08分	21時22分	21時13分	21時16分	21時35分	21時54分	21時39分	21時49分	21時42分	21時46分
天　候	晴	曇り	晴	晴	晴	晴	晴	晴	雪	曇り	雪	曇り	曇り	曇り
気　温	2.4度	2.2度	2.4度	2.5度	2.7度	2.6度	4.0度	3.9度	2.3度	1.4度	1.4度	1.3度	1.0度	1.1度
路面状況	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 西川 修三 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月6日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時32分　帰着時間 5時43分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時34分	5時18分	4時39分	5時05分	4時48分	5時00分	4時53分	4時55分	5時21分	5時38分	5時25分	5時35分	5時28分	5時31分
天　候	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り
気　温	−1.8度	−0.9度	−2.1度	−1.2度	−2.2度	−2.2度	−1.4度	−1.3度	−1.3度	−0.8度	−1.2度	−0.9度	−1.0度	−0.9度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 保坂 浩史 記入者	
	出　km	
	着　km	
	走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月5日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 21時36分　帰着時間 22時46分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	21時39分	22時24分	21時43分	22時09分	21時51分	22時05分	21時56分	22時00分	22時27分	22時45分	22時31分	22時41分	22時34分	22時37分
天　候	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り
気　温	−0.9度	−1.2度	−1.1度	−1.1度	−1.1度	−1.2度	−0.6度	−0.4度	−1.1度	−1.0度	−1.1度	−1.0度	−0.9度	−0.9度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 保坂 浩史 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月4日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時35分　帰着時間 5時39分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時37分	5時18分	4時42分	4時43分	4時52分	5時04分	4時57分	5時00分	5時21分	5時38分	5時25分	5時34分	5時28分	5時31分
天　候	晴	晴	晴	晴	晴	晴	晴	晴	晴	曇り	晴	晴	晴	晴
気　温	−1.8度	−1.7度	−1.9度	−2.2度	−1.5度	−2.2度	−1.0度	−1.6度	−1.9度	−1.9度	−2.4度	−2.3度	−2.5度	−2.2度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 渡貫 博 記入者 出　km 着　km 走行　km	

(注)情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月3日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 20時55分　帰着時間 21時54分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	20時59分	21時39分	21時04分	21時06分	21時16分	21時30分	21時22分	21時24分	21時43分	22時03分	21時48分	21時59分	21時52分	21時55分
天　候	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り
気　温	−0.8度	0.0度	−0.4度	−0.1度	0.0度	0.5度	0.1度	0.4度	−0.5度	−0.7度	−0.6度	−1.0度	−0.8度	−0.5度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 村山　清一 記入者	
	出　km	
	着　km	
	走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月3日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時26分　帰着時間 4時49分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時29分	5時11分	4時34分	4時36分	4時47分	5時02分	4時53分	4時56分	5時21分	5時41分	5時25分	5時37分	5時29分	5時32分
天　候	晴	雪	曇り	曇り	曇り	曇り	曇り	曇り	雪	雪	雪	雪	雪	雪
気　温	−0.6度	0.0度	−0.6度	0.5度	−0.1度	0.8度	1.3度	0.4度	0.1度	−0.1度	−0.7度	−1.0度	−1.0度	−0.5度
路面状況	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm	ザクレ 新雪　2 cm 圧雪　2 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ○	喜多IC ○	蓮潟 ○	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ○	大積1 ○		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ： 塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 千野　征夫 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年3月2日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 20時48分　帰着時間 21時52分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	20時51分	21時28分	20時56分	21時23分	21時04分	21時18分	21時09分	21時13分	21時32分	21時51分	21時36分	21時47分	21時40分	21時43分
天　候	雪	雪	雪	雪	雪	雪	雪	雪	雪	雪	雪	雪	雪	雪
気　温	−0.9度	0.4度	−0.4度	−0.4度	0.4度	−0.5度	1.2度	0.9度	−0.6度	−0.8度	−1.1度	−1.1度	−1.2度	−0.9度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm	ザクレ 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ○	蓮潟 ×	蓮潟 ○	水道町 ○ 中島7 ○ 西神田2 ○	西神田2 ○ 中島7 ○ 水道町 ○			大積1 ○	大積1 ○		
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ： 塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 千野 征夫 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月29日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時31分　帰着時間 4時31分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時34分	5時07分	4時37分	4時58分	4時41分	4時55分	4時45分	4時49分	5時10分	5時27分	5時14分	5時24分	5時17分	5時20分
天　候	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り
気　温	5.1度	5.3度	5.9度	5.4度	4.2度	4.3度	5.4度	4.6度	3.1度	2.4度	1.9度	1.6度	1.3度	1.3度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 保坂 浩史 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月28日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 21時34分　帰着時間 21時34分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	21時37分	22時13分	21時41分	22時02分	21時46分	21時58分	21時50分	21時53分	22時17分	22時36分	22時21分	22時32分	22時25分	22時28分
天　候	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴
気　温	2.4度	4.4度	3.7度	3.1度	3.4度	3.0度	3.4度	3.4度	1.3度	0.1度	−0.1度	−0.6度	−0.9度	−0.9度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線上下	発見及び発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 保坂 浩史 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月27日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 21時29分　帰着時間 22時38分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	21時32分	22時10分	21時36分	22時06分	21時49分	22時02分	21時55分	21時57分	22時13分	22時33分	22時17分	22時29分	22時21分	22時25分
天　候	晴	雪	雨	雪	雨	雪	雨	雪	雪	雪	雪	雪	雪	雪
気　温	1.1度	0.8度	1.1度	1.6度	1.6度	0.9度	2.1度	1.9度	0.8度	0.6度	0.5度	0.5度	0.4度	0.8度
路面状況	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	濡れ 新雪　cm 圧雪　cm	ザクレ 新雪　cm 圧雪　cm	ザクレ 新雪　cm 圧雪　cm	ザクレ 新雪　cm 圧雪　cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ○	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ○	西神田2 ○			大積1 ○	大積1 ○		
							中島7 ○	中島7 ○						
							西神田2 ○	水道町 ○						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ： 塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 西川 修三 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月27日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 5時01分　帰着時間 6時09分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	5時03分	5時46分	5時07分	5時32分	5時17分	5時29分	5時21分	5時24分	5時50分	6時08分	5時54分	6時05分	5時58分	6時01分
天　候	雪	雪	雪	雪	雪	雪	雪	雪	雪	雪	雪	雪	雪	雪
気　温	0.2度	−0.6度	−0.3度	−0.4度	0.0度	−0.4度	0.1度	0.3度	−0.1度	−0.1度	0.1度	−0.2度	−0.5度	−0.5度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	シャーベット 新雪　0 cm 圧雪　0 cm	シャーベット 新雪　0 cm 圧雪　0 cm	シャーベット 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ○	喜多IC ○	蓮潟 ○	蓮潟 ○	水道町 ○	西神田2 ○			大積1 ○	大積1 ○		
							中島7 ○	中島7 ○						
							西神田2 ○	水道町 ○						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ： 塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 渡貫 博 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月26日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 21時29分　帰着時間 22時37分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	21時31分	22時13分	21時37分	21時40分	21時51分	22時05分	21時57分	21時59分	22時17分	22時36分	22時21分	22時32分	22時25分	22時28分
天　候	雪	曇り	雪	雪	曇り	曇り	曇り	曇り	雪	曇り	曇り	曇り	曇り	曇り
気　温	1.9度	3.3度	2.1度	2.1度	2.3度	4.0度	3.0度	2.7度	2.4度	1.9度	2.1度	1.9度	1.4度	2.0度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 村山　清一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月25日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 5時07分　帰着時間 5時23分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	5時07分	5時46分	5時13分	5時42分	5時24分	5時38分	5時30分	5時32分	5時49分	6時08分	5時53分	6時04分	5時57分	6時00分
天　候	晴	曇り	晴	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り
気　温		4.5度		3.4度	3.6度	3.4度	4.5度	5.8度	4.4度	5.1度	8.6度	8.7度	7.5度	7.4度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 佐藤 吉一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月24日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 21時43分　帰着時間 22時49分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	21時46分	22時26分	21時51分	21時59分	22時06分	22時17分	22時10分	22時13分	22時30分	22時48分	22時33分	22時44分	22時37分	22時40分
天　候	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴
気　温	2.3度	2.6度	3.1度	3.2度	3.6度	2.7度	4.1度	3.8度	0.7度	−0.2度	−0.3度	−0.8度	−0.9度	−1.0度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 佐藤 吉一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月24日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時13分　帰着時間 5時24分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時15分	4時55分	4時20分	4時50分	4時32分	4時45分	4時37分	4時40分	4時58分	5時17分	5時02分	5時13分	5時06分	5時09分
天　候	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴
気　温	2.1度	0.9度	1.8度	1.3度	2.1度	1.4度	2.8度	2.9度	1.3度	0.9度	0.6度	0.5度	0.5度	0.4度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 佐野 義一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月23日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 20時59分　帰着時間 22時04分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	21時03分	21時40分	21時08分	21時36分	21時17分	21時31分	21時23分	21時26分	21時44分	22時03分	21時48分	21時59分	21時52分	21時55分
天　候	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	雪	曇り	曇り	曇り
気　温	2.9度	3.6度	2.8度	4.6度	3.1度	3.9度	3.2度	3.4度	4.0度	3.4度	3.7度	3.0度	2.9度	3.4度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線上下	発見及び発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 佐野 義一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月21日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時18分　帰着時間 5時23分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時21分	4時59分	4時26分	4時54分	4時32分	4時48分	4時38分	4時42分	5時02分	5時23分	5時07分	5時19分	5時11分	5時15分
天　候	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴
気　温	1.3度	3.7度	3.4度	2.8度	1.0度	0.9度	1.8度	2.0度	-0.7度	-1.7度	-1.9度	-2.5度	-2.2度	-2.5度
路面状況	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ○		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線上下	発見及び発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 村山 清一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月20日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 21時10分　帰着時間 22時08分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	21時13分	21時56分	21時18分	21時20分	21時30分	21時47分	21時36分	21時40分	22時00分	22時20分	22時04分	22時16分	22時08分	22時12分
天　候	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り
気　温	2.7度	2.3度	3.4度	3.3度	3.3度	3.5度	3.9度	3.8度	0.8度	0.2度	−0.3度	−0.5度	−0.6度	−0.6度
路面状況	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm	乾燥 新雪　cm 圧雪　cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 村山　清一 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月19日　1回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時18分　帰着時間 5時24分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時22分	5時01分	4時27分	4時35分	4時41分	4時55分	4時46分	4時49分	5時04分	5時23分	5時08分	5時19分	5時12分	5時15分
天　候	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴	晴
気　温	−0.6度	0.2度	0.1度	0.0度	0.7度	0.9度	0.9度	0.8度	−0.7度	−1.0度	−1.1度	−1.3度	−1.3度	−1.1度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　kg　SM　kg　散布延長　km　散布箇所 ：													
	塩カル　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線上下	発見及び発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
	運転員 保坂 浩史 記入者 出　km 着　km 走行　km	

（注）情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月18日　3回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 4時35分　帰着時間 5時41分　現場代理人

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	4時38分	5時19分	4時45分	4時46分	4時55分	5時09分	5時00分	5時03分	5時23分	5時40分	5時26分	5時37分	5時30分	5時33分
天　候	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り	曇り
気　温	3.9度	3.6度	4.0度	3.7度	3.0度	5.2度	3.4度	3.3度	4.4度	3.9度	4.3度	4.0度	3.9度	4.1度
路面状況	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm	乾燥 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩 3000 kg　SM　　kg　散布延長　　km　散布箇所 ：　長岡大橋　交差点部 塩カル　　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
試験運用散布実施	運転員	
路面状況異常無し	舛岡　喜一	
	記入者	
	出　km	
	着　km	
	走行　km	

(注)情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

様式－4（第9条）

道　路　巡　回　日　報

3（宮　本）工区除雪作業　　平成16年2月17日　2回目　　出動区分（ 指示 ・ 警察 ・ ﾓﾆﾀｰ ・ 一般 ・ その他 ）　出発時間 22時03分　帰着時間 22時03分　現場代理人　新保 信行

地点名	関原交差点 67.1Km		福山交差点 64.6Km		寺島交差点 62Km		昭和通交差点 58.8Km		宮本東方交差点 69.7Km		大積2丁目交差点 72.8Km		大積田代交差点 75.4Km	
区　間	68.9km ～ 67.1km		67.1km ～ 64.6km		64.6km ～ 62km		62km ～ 58.8km		67.1km ～ 69.7km		69.2km ～ 72.8km		72.8km ～ 76.7km	
往・復路	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り	行　き	帰　り
観測時間	22時04分	22時41分	22時09分	22時13分	22時18分	22時32分	22時24分	22時27分	22時44分	23時04分	22時49分	23時00分	22時53分	22時56分
天　候	雨	雨	雨	雨	雨	雨	雨	雨	曇り	曇り	曇り	曇り	雨	雨
気　温	3.5度	3.5度	2.7度	2.9度	2.8度	4.7度	3.3度	3.3度	4.1度	3.9度	4.0度	4.1度	3.9度	4.5度
路面状況	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm	濡れ 新雪　0 cm 圧雪　0 cm
交通状況	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他	良好 その他
消雪設備			喜多IC ×	喜多IC ×	蓮潟 ×	蓮潟 ×	水道町 ×	西神田2 ×			大積1 ×	大積1 ×		
							中島7 ×	中島7 ×						
							西神田2 ×	水道町 ×						
薬剤散布状況	原塩　－ kg　SM　　kg　散布延長　　km　　散布箇所 ：													
	塩カル　　kg													

消雪設備については、別紙消雪設備一覧表の設置番号と、稼動は○、不稼動は×を記入し、操作したときは特記事項の欄に操作内容を記入する。

障　害　箇　所

場　所	距離票	車線 上下	発見及び 発生時間	障　害　内　容	摘　要
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		
	km	上　下	時　分		

特　記　事　項（渋滞原因等も記入）	運転員及び記入者 走　行　距　離	備　考
試験運用散布(巡回)実施	運転員	
路面状況異常無し	西川　修三	
	記入者	
	出　km	
	着　km	
	走行　km	

(注)情報板・標識の雪の状況や、排水不良箇所、登坂不能車、路上チェーン装着等、障害が起きそうな箇所及び車両について記入する。

道路 GIS

Software Development Kit

データベース 設計書

平成 16 年 3 月

【道路 GIS SDK データベース設計書　目次】

1. テーブル定義 3

1.1. テーブル一覧 3
1.2. テーブル設計詳細 5
1.2.1. RgstHist テーブル 5
1.2.2. UserManagement テーブル 6
1.2.3. WK_Extent テーブル 7
1.2.4. MemoCategory テーブル 8
1.2.5. MemoList テーブル 9
1.2.6. LandMarkNameCatalog テーブル 10
1.2.7. LandMarkList テーブル 11
1.2.8. PlaceNameCatalog テーブル 12
1.2.9. PlaceNameList テーブル 13
1.2.10. SpotMarkNameCatalog テーブル 14
1.2.11. SpotMarkList テーブル 15
1.2.12. WK_Portraiture テーブル 16
1.2.13. WK_ReferenceRaster テーブル 17
1.2.14. Reg_GM_Primitive テーブル 18
1.2.15. Reg_GM_Point テーブル 19
1.2.16. Reg_DirectPosition テーブル 20
1.2.17. Reg_Coordinate テーブル 21
1.2.18. Reg_GM_OrientablePrimitive テーブル 22
1.2.19. Reg_GM_CompositeCurve テーブル 23
1.2.20. Reg_GM_Curve テーブル 24
1.2.21. Reg_GM_LineString テーブル 25
1.2.22. Reg_GM_Surface テーブル 26
1.2.23. Reg_Exterior テーブル 27
1.2.24. Reg_Interior テーブル 28
1.2.25. Reg_TM_ReferenceSystem テーブル 29
1.2.26. Reg_TM_Period テーブル 30
1.2.27. Reg_TM_Instant テーブル 31
1.2.28. Reg_TM_CalDate テーブル 32
1.2.29. Reg_tbl_caldate テーブル 33
1.2.30. Reg_tbl_begunby テーブル 34
1.2.31. Reg_tbl_endedby テーブル 35

1.2.32. WK_Point テーブル 36
1.2.33. WK_Point_SL テーブル 37
1.2.34. WK_Curve テーブル 38
1.2.35. WK_Surface テーブル 39
1.2.36. WK_Period テーブル 40
1.2.37. WK_Instant テーブル 41
1.2.38. WK_CalDate テーブル 42
1.2.39. FeatureInstance テーブル 43
1.2.40. CodeList テーブル 44
1.2.41. AppSchemaManagement テーブル 45

1. テーブル定義

1.1. テーブル一覧

テーブル一覧を以下に示す。以下に示すテーブルはいずれもアプリケーション運用のためのテーブルである。

表 1　テーブル一覧

分類	テーブル名	名称	備考
システム管理	RegistHist	道路基盤データ登録履歴	
システム管理	UserManagement	ユーザ管理	ユーザ認証用
システム管理	WK_Extent	地域定義	初期表示用描画範囲の定義
メモ機能	MemoCategory	メモ分類	
メモ機能	MemoList	メモ情報	
目標物検索情報	LandMarkNameCatalog	目標物分類	
目標物検索情報	LandMarkList	目標物情報	
住所検索情報	PlaceNameCatalog	地名分類	
住所検索情報	PlaceNameList	地名情報	
距離標検索情報	SpotMarkNameCatalog	距離標分類	
距離標検索情報	SpotMarkList	距離標情報	
描画情報	WK_Portraiture	描画法	
描画情報	WK_ReferenceRaster	ラスタデータ管理	
空間属性インスタンス管理	Reg_GM_Primitive	GM_Primitive 管理	
空間属性インスタンス管理	Reg_GM_Point	GM_Point 管理	
空間属性インスタンス管理	Reg_DirectPosition	DirectPosition 管理	
空間属性インスタンス管理	Reg_Coordinate	座標値管理	
空間属性インスタンス管理	Reg_GM_OrientablePrimitive	GM_ OrientablePrimitive 管理	
空間属性インスタンス管理	Reg_GM_CompositeCurve	Reg_GM_CompositeCurve 管理	
空間属性インスタンス管理	Reg_GM_Curve	GM_Curve 管理	
空間属性インスタンス管理	Reg_GM_LineString	GM_LineString 管理	
空間属性インスタンス管理	Reg_GM_Surface	GM_Surface 管理	

空間属性インスタンス管理	Reg_Exterior	Exterior 管理	
空間属性インスタンス管理	Reg_Interior	Interior 管理	
時間属性インスタンス管理	Reg_TM_ReferenceSystem	TM_ReferenceSystem 管理	
時間属性インスタンス管理	Reg_TM_Period	TM_Period 管理	
時間属性インスタンス管理	Reg_TM_Instant	TM_Instant 管理	
時間属性インスタンス管理	Reg_TM_CalDate	TM_CalDate 管理	
時間属性インスタンス管理	Reg_tbl_caldate	tbl_caldate 管理	
時間属性インスタンス管理	Reg_tbl_begunby	tbl_begunby 管理	
時間属性インスタンス管理	Reg_tbl_endedby	tbl_endedby 管理	
描画用空間属性	WK_Point	WK_Point 管理	
描画用空間属性	WK_Point_SL	WK_Point_SL 管理	
描画用空間属性	WK_Curve	WK_Curve 管理	
描画用空間属性	WK_Surface	WK_Surface 管理	
表示用時間属性	WK_Period	WK_Period 管理	
表示用時間属性	WK_Instant	WK_Instant 管理	
表示用時間属性	WK_CalDate	WK_CalDate 管理	
地物ｲﾝｽﾀﾝｽ管理	FeatureInstance	地物インスタンス管理	
コードリスト管理	CodeList	コードリスト管理	
応用スキーマ管理	AppSchemaManagement	応用スキーマ管理	

1.2. テーブル設計詳細

1.2.1. RgstHist テーブル

用途: 道路基盤データのデータベース登録履歴管理に使用

ﾃｰﾌﾞﾙ名称	道路基盤データ登録履歴		ﾃｰﾌﾞﾙ名	RgstHist				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	id	character(14)			否	p1	
2	データ種別	kind	character varying(32)			否		"道路基盤データ"固定
3	XMLﾌｧｲﾙ名称	filename	character varying(256)			否		元データファイル名
4	メモ	memo	character varying(1024)			否		

制約事項: ID は 登録日時（年月日時分秒）の１４桁

コード　:

フラグ　:

備考　:

1.2.2. UserManagement テーブル

用途: システムログオン時のユーザ認証で使用

ﾃｰﾌﾞﾙ名称	ユーザ管理		ﾃｰﾌﾞﾙ名	UserManagement				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ユーザ名	username	character varying(100)			否	p1	
2	パスワード	password	character varying(100)					
3	ｾｯｼｮﾝID	sessionid	character varying(100)					現在未使用
4	ﾚﾍﾞﾙ	lev	character varying(100)					現在未使用
5	ﾀｲﾑｽﾀﾝﾌﾟ	WKtimestamp	timestamp			否		登録日時

制約事項:

コード　:

フラグ　:

備考　: ｾｯｼｮﾝ ID、ﾚﾍﾞﾙは現在未使用

1.2.3. WK_Extent テーブル

用途: ログオン時に指定地域の範囲を取得するために使用

ﾃｰﾌﾞﾙ名称	地域定義		ﾃｰﾌﾞﾙ名	WK_Extent				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	最大緯度座標	North	double precision			否		
2	最小経度座標	West	double precision			否		
3	最小緯度座標	South	double precision			否		
4	最大経度座標	East	double precision			否		
5	地域名	LOCATION	character varying(10)			否	p1	
6	ﾀｲﾑｽﾀﾝﾌﾟ	WKtimestamp	timestamp			否		登録日時

制約事項:

コード　:

フラグ　:

備考　: 初期表示範囲の座標を最大最小緯度経度座標で設定する。

地域名(LOCATION)は、地方整備局や工事事務所等の地域を表す名称　(ex. “NAGAOKA”)

すでに登録済みの Location は、以下の通り。

地域	Location
北海道開発局	**HOKKAIDO**
東北地方整備局	**TOHOKU**
北陸地方整備局	**HOKURIKU**
関東地方整備局	**KANTO**
中部地方整備局	**CHUBU**
近畿地方整備局	**KINKI**
中国地方整備局	**CHUGOKU**
四国地方整備局	**SHIKOKU**
九州地方整備局	**KYUSHU**
沖縄総合事務事務局	**OKINAWA**
長岡国道事務所	**NAGAOKA**
岡山国道事務所	**OKAYAMA**
福岡国道事務所	**FUKUOKA**
山形河川国道事務所	**YAMAGATA**

1.2.4. MemoCategory テーブル

用途: メモの種類の情報を管理

ﾃｰﾌﾞﾙ名称	メモ分類		ﾃｰﾌﾞﾙ名	MemoCategory				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	種別ID	CategoryID	integer			否	p1	
2	種別名称	name	character varying(100)					
3	文字表示色	ForeGroundColor	character varying(100)					
4	背景色	BackGroundColor	character varying(100)					
5	ﾌｫﾝﾄｻｲｽﾞ	FontSize	NUMERIC(5,2)					
6	太字表示ﾌﾗｸﾞ	FlgBold	integer					
7	斜体表示ﾌﾗｸﾞ	FlgItalic	integer					

制約事項: 文字表示色，背景色は ”#rrggbb ” の１６進表記

コード　:

フラグ　: 太字表示ﾌﾗｸﾞ，斜体表示ﾌﾗｸﾞは、表示=1,非表示=0

備考　:

1.2.5. MemoList テーブル

用途: メモ情報を格納

ﾃｰﾌﾞﾙ名称	メモ情報		ﾃｰﾌﾞﾙ名	MemoList				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	MemoID	integer			否	p1	
2	ｶﾃｺﾞﾘID	CategoryID	integer					
3	メモ共有ﾌﾗｸﾞ	flgPub	integer					
4	有効期限開始日付	BeginDate	integer					
5	有効期限終了日付	EndDate	integer					
6	メモ内容	Description	character varying(100)					
7	作成者	Auther	character varying(100)					
8	経度	Logitude	double precision					
9	緯度	Latitude	double precision					

制約事項: ｶﾃｺﾞﾘ ID は MemoCategory テーブルの種別 ID に結合

有効期限開始日付，有効期限終了日付は”年月日(YYYYMMDD)”の８桁の数であること

コード　：

フラグ　：メモ共有ﾌﾗｸﾞは、共有=1,非共有=0

備考　：

1.2.6. LandMarkNameCatalog テーブル

用途：目標物検索に使用する検索キーの分類を定義

ﾃｰﾌﾞﾙ名称	目標物分類		ﾃｰﾌﾞﾙ名	LandMarkNameCatalog				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	検索ﾚﾍﾞﾙ	lev	integer			否	a2	
2	検索分類列名	name	character varying(100)					
3	検索ﾌｨｰﾙﾄﾞ名	Annotation	character varying(100)					
4	検索可能ﾌﾗｸﾞ	FlgEnable	Integer					
5	表示ﾌﾗｸﾞ	FlgVisible	Integer					
6	ふりがな分類ﾌﾗｸﾞ	FlgOrder	Integer					
7	地域名	LOCATION	character varying(10)			否	a1	

制約事項：

コード　：

フラグ　：検索可能ﾌﾗｸﾞは,検索実行=1,非検索実行=0
表示ﾌﾗｸﾞは検索ダイアログに表示=1,非表示=0
ふりがな分類ﾌﾗｸﾞは，ふりがなで検索=1，ふりがなナ検索なし=0

備考　：検索ﾚﾍﾞﾙはカテゴリの階層番号(1～)
検索分類列名は LandMarkList テーブルの目標物種別(Category)か目標物名称(LandMark)の列名
検索ﾌｨｰﾙﾄﾞ名はカテゴリ名称
地域名(LOCATION)は、地方整備局や工事事務所等の地域を表す名称　(ex.“NAGAOKA”)

1.2.7. LandMarkList テーブル

用途: 目標物検索の検索キーとなる目標物情報を管理

ﾃｰﾌﾞﾙ名称	目標物情報		ﾃｰﾌﾞﾙ名	LandMarkList				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	integer			否	p1	
2	ﾚﾍﾞﾙ	LEV	integer					
3	大分類	COUNTRY	character varying(100)					現在未使用
4	中分類	DOMAIN	character varying(100)					現在未使用
5	小分類	CITY	character varying(100)					現在未使用
6	目標物種別	CATEGORY	character varying(100)					
7	目標物名称	LANDMARK	character varying(100)					
8	ふりがな	ANNOTATION	character varying(100)					
9	最大緯度	NORTHEND	double precision					
10	最小緯度	SOUTHEND	double precision					
11	最小経度	WESTEND	double precision					
12	最大経度	EASTEND	double precision					
13	地域名	location	character varying(10)					

制約事項: ﾚﾍﾞﾙは LandMarkNameCatalog テーブルの検索ﾚﾍﾞﾙと結合
ふりがな検索を使用する場合は”ふりがな”列に値を設定

コード　:

フラグ　:

備考　: 最大最小緯度経度座標で検索実行後の表示範囲を設定
地域名(LOCATION)は、地方整備局や工事事務所等の地域を表す名称　(ex.“NAGAOKA”)

1.2.8. PlaceNameCatalog テーブル

用途: 住所検索に使用する検索キーの分類を定義

ﾃｰﾌﾞﾙ名称	地名分類		ﾃｰﾌﾞﾙ名	PlaceNameCatalog				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	検索ﾚﾍﾞﾙ	lev	integer			否	a2	
2	検索分類列名	name	character varying(100)					"City"固定
3	検索ﾌｨｰﾙﾄﾞ名	Annotation	character varying(100)					
4	検索可能ﾌﾗｸﾞ	FlgEnable	integer					
5	表示ﾌﾗｸﾞ	FlgVisible	integer					
6	ふりがな分類ﾌﾗｸﾞ	FlgOrder	integer					
7	地域名	location	character varying(10)			否	a1	

制約事項:

コード　:

フラグ　: 検索可能ﾌﾗｸﾞは,検索実行=1,非検索実行=0
表示ﾌﾗｸﾞは検索ダイアログに表示=1,非表示=0
ふりがな分類ﾌﾗｸﾞは，ふりがなで検索=1，フリガナ検索なし=0

備考　: 検索分類列名は PlaceNameList テーブルの大中小分類の列名
検索ﾚﾍﾞﾙはカテゴリの階層番号(1～)
検索ﾌｨｰﾙﾄﾞ名はカテゴリ名称
地域名(LOCATION)は、地方整備局や工事事務所等の地域を表す名称　(ex. “ NAGAOKA ”)

1.2.9. PlaceNameList テーブル

用途: 住所検索の検索キーとなる住所情報を管理

ﾃｰﾌﾞﾙ名称	地名情報		ﾃｰﾌﾞﾙ名	PlaceNameList				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	integer			否	p1	
2	検索ﾚﾍﾞﾙ	LEV	integer					
3	整備局名称	COUNTRY	character varying(100)					
4	工事事務所名称	DOMAIN	character varying(100)					
5	市町村名称	CITY	character varying(100)					
6	大字名称	MAJOR	character varying(100)					現在未使用
7	小字、丁目名称	MINOR	character varying(100)					現在未使用
8	街区番号	PARCEL	character varying(100)					現在未使用
9	ふりがな	ANNOTATION	character varying(100)					
10	最大緯度	NORTHEND	double precision					
11	最小緯度	SOUTHEND	double precision					
12	最小経度	WESTEND	double precision					
13	最大経度	EASTEND	double precision					
14	地域名	location	character varying(10)					

制約事項: ﾚﾍﾞﾙは PlaceNameCatalog テーブルの検索ﾚﾍﾞﾙと結合

フリガナ検索を使用する場合は ”ふりがな”列に値を設定

コード　:

フラグ　:

備考　: 最大最小緯度経度座標で検索実行後の表示範囲を設定

地域名(LOCATION)は、地方整備局や工事事務所等の地域を表す名称　(ex. “NAGAOKA”)

1.2.10. SpotMarkNameCatalog テーブル

用途: キロポスト検索に使用する検索キーの分類を定義

ﾃｰﾌﾞﾙ名称	距離標分類		ﾃｰﾌﾞﾙ名	SpotMarkNameCatalog				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ﾚﾍﾞﾙ	lev	integer			否	a2	
2	検索分類列名	name	character varying(100)					
3	検索ﾌｨｰﾙﾄﾞ名	Annotation	character varying(100)					
4	検索可能ﾌﾗｸﾞ	FlgEnable	Integer					
5	表示ﾌﾗｸﾞ	FlgVisible	Integer					
6	ふりがな分類ﾌﾗｸﾞ	FlgOrder	Integer					0固定
7	地域名	LOCATION	character varying(10)			否	a1	

制約事項:

コード　:

フラグ　: 検索可能ﾌﾗｸﾞは,検索実行=1,非検索実行=0
表示ﾌﾗｸﾞは検索ダイアログに表示=1,非表示=0
ふりがな分類ﾌﾗｸﾞは, ふりがな検索なし=0 固定

備考　: 検索分類列名は PlaceNameList テーブルの分類列名 (ex.”Domain”)
検索ﾚﾍﾞﾙはカテゴリの階層番号(1～)
検索ﾌｨｰﾙﾄﾞ名はカテゴリ名称
地域名(LOCATION)は、地方整備局や工事事務所等の地域を表す名称 (ex.“NAGAOKA”)

1.2.11. SpotMarkList テーブル

用途: 道路基盤データのデータベース登録履歴管理に使用

ﾃｰﾌﾞﾙ名称	距離標情報		ﾃｰﾌﾞﾙ名	SpotMarkList				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	integer			否	pl	
2	検索ﾚﾍﾞﾙ	LEV	integer					
3	分類１	COUNTRY	character varying(100)					整備局名称
4	分類２	DOMAIN	character varying(100)					工事事務所名称
5	分類３	ROADTYPE	character varying(100)					道路種別
6	分類４	ROUTE	character varying(100)					路線名称
7	分類５	SECTION	character varying(100)					区間名称
8	分類６	SPOTMARK	character varying(100)					キロポスト名称
9	ふりがな	ANNOTATION	character varying(100)					現在未使用
10	最大緯度	NORTHEND	double precision					
11	最小緯度	SOUTHEND	double precision					
12	最小経度	WESTEND	double precision					
13	最大経度	EASTEND	double precision					
14	開始日付	b_period	character varying(8)			否		
15	終了日付	e_period	character varying(8)			否		
16	ﾀｲﾑｽﾀﾝﾌﾟ	WKtimestamp	timestamp			否		
17	路線番号	routeno	character varying(4)					
18	地域名	location	character varying(10)					

制約事項: ﾚﾍﾞﾙは PlaceNameCatalog テーブルの検索ﾚﾍﾞﾙと結合

最大経度、最小経度は同値(表示範囲はシステムが決定)

最大緯度、最小緯度は同値(表示範囲はシステムが決定)

コード　:

フラグ　:

備考　: 検索ﾚﾍﾞﾙはカテゴリの階層番号(1～)

地域名(LOCATION)は、地方整備局や工事事務所等の地域を表す名称　(ex. “ NAGAOKA ”)

1.2.12. WK_Portraiture テーブル

用途: 描画法管理に使用

ﾃｰﾌﾞﾙ名称	描画法		ﾃｰﾌﾞﾙ名	WK_Portraiture				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ユーザ名	Username	character varying(100)			否		ユーザ別設定のキー
2	セッションID	SessionID	character varying(100)					使用していない(予備)
3	スキーマ名	schema_name	character varying(100)					地物別設定のキー(スキーマ名称)
4	クラス名	ClassName	character varying(100)			否		地物別設定のキー(パッケージ名称)
5	パッケージ名	packagename	character varying(100)			否		地物別設定のキー(クラス名称)
6	地物名	featurename	character varying(100)			否		地物名称
7	地物日本語表記名	annotation	character varying(100)			否		地物日本語名称
8	表示フラグ	state	integer			否		表示／非表示設定(0=非表示,1=表示)
9	表示レイヤ	DrawLevNo	integer			否		表示レイヤー番号(昇順に描画)
10	ラスタフラグ	FlgRaster	integer			否		ラスタデータフラグ(0=非ラスタ,1=ラスタ)
11	線色	P_LineColor	character varying(6)					線色
12	線幅	P_LineWide	character varying(6)					線幅
13	線種	P_LinePattern	character varying(20)					線パターン
14	ブラシ前景色	P_BrushFGColor	character varying(6)					ブラシ前景色
15	ブラシ背景色	P_BrushBGColor	character varying(6)					ブラシ背景色
16	ブラシ種	P_BrushPattern	character varying(20)					ブラシパターン
17	シンボル名	P_SymbolName	character varying(100)					シンボル表示アイコンファイル名
18	表示最小スケール	P_MinScale	integer			否		最小表示スケール
19	表示最大スケール	P_MaxScale	integer			否		最大表示スケール
20	タイムスタンプ	WKtimestamp	timestamp			否		タイムスタンプ
21	透過率	alpha	double precision					透過設定(0.0(=非表示)～1.0(=透過なし))
22	シンボル/テキスト種別	sl_kind	integer					シンボル／テキスト判別フラグ(0=シンボル,1=テキスト)
23	表示角度指定有効フラグ	sl_flgangle	integer					表示角度指定有効フラグ(0=無効,1=有効)
24	基準スケール	sl_baseScale	integer					基準サイズで表示するスケール
25	基準サイズ	sl_basesize	integer					基準サイズ(シンボル:縦横ドット数,テキスト:ポイント数)
26	倍率の重み係数	sl_rate	double precision					可変表示サイズを計算する際の倍率に掛ける重み係数(>0.0)
27	テキスト表示色	sl_textcolor	character(6)					RRGGBB('ffffff')
28	表示サイズ可変有効フラグ	sl_flgrate	integer					可変サイズ設定フラグ(0=サイズ固定,1=サイズ可変)
29	シンボルイメージフラグ	flgsymbolinstance	integer					1=インスタンス設定を有効,2=クラス設定を有効
30	クラス識別	classification	character varying(300)					スキーマ名_パッケージ名_クラス名

制約事項: 線色は rrggbb の１６進表記で登録

ﾌﾞﾗｼ前景色は rrggbb の１６進表記で登録

ﾌﾞﾗｼ背景色は rrggbb の１６進表記で登録

コード : 線種は"SOLID", "DASH", "DOT", "DASHDOT", "DASHDOT2"

ﾌﾞﾗｼ種は"EMPTY", "FDIAGONAL"(／方向の斜線), "BDIAGONAL"(＼方向の斜線), "HORIZONTAL"（ー方向の線）, "VERTICAL"（｜方向の線）,"CROSS"（＋）, "DIAGONALCROSS"（×）

フラグ : 表示ﾌﾗｸﾞ : 0=非表示、1=表示

ﾗｽﾀ･ﾌﾗｸﾞ : 0=非ラスタ,1=ラスタ

備考 : ｾｯｼｮﾝ ID は未使用

表示ﾚｲﾔ : 0(下層ﾚｲﾔ)～

ラスタデータを登録する場合は、クラス名、パーケージ名、地物名に WK_ReferenceRaster テーブルのﾗｽﾀ種別名を登録する。ただし、索引図（IndexMap）の情報は登録しない。

1.2.13. WK_ReferenceRaster テーブル

用途: ラスタデータの管理に使用

ﾃｰﾌﾞﾙ名称	ラスタデータ管理		ﾃｰﾌﾞﾙ名	WK_ReferenceRaster				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ﾗｽﾀ種別名	RasterName	character varying(100)			否	a1	
2	ﾗｽﾀﾌｧｲﾙ名	FileName	character varying(100)			否		
3	最小経度座標	minX	double precision			否	a4	
4	最小緯度座標	minY	double precision			否	a6	
5	最大経度座標	maxX	double precision			否	a3	
6	最大緯度座標	maxY	double precision			否	a5	
7	ﾀｲﾑｽﾀﾝﾌﾟ	WKtimestamp	timestamp			否		登録日時
8	地域名	LOCATION	character varying(10)				a2	索引図登録時のみ使用

制約事項: ﾗｽﾀ種別名は WK_Portraiture テーブルのｸﾗｽ名と結合する

地域名列は、索引図を登録する場合のみ使用する。

コード : ﾗｽﾀ種別名は、以下のとおり。

“AeroPhoto”: 航空写真

“DigitalMap200000”: 数値地図 200000

“DigitalMap25000”: 数値地図 25000

“SatellitePhoto” : 衛星写真

“IndexMap”: 索引図

フラグ :

備考 : 最大最小緯度経度はラスタデータの範囲

1.2.14. Reg_GM_Primitive テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	GM_PRIMITIVE管理		ﾃｰﾌﾞﾙ名	Reg_GM_PRIMITIVE				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	GM_PrimitiveのID
2	座標参照系参照ID	CRS	character varying(100)			否		
3	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.15. Reg_GM_Point テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	GM_Point管理		ﾃｰﾌﾞﾙ名	Reg_GM_Point				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	GM_PointのID
2	Directposition参照ID	Position	character varying(100)			否		
3	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.16. Reg_DirectPosition テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	DirectPosition管理		ﾃｰﾌﾞﾙ名	Reg_DirectPosion				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	DirectPositionｵﾌﾞｼﾞｪｸﾄのID
2	最大次元数	DIMENSION	integer			否		座標の最大次元数
3	座標参照系参照ID	CRS	character varying(100)			否		
4	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.17. Reg_Coordinate テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	座標値管理		ﾃｰﾌﾞﾙ名	Reg_Coordinate				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	DirectPositionのID
2	次元	SEQ	integer			否	p2	座標の次元
3	座標値	VALUE	double precision			否		
4	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.18. Reg_GM_OrientablePrimitive テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	GM_OrientablePrimitive管理		ﾃｰﾌﾞﾙ名	Reg_GM_OrientablePrimitive				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	GM_OrientablePrimitiveのID
2	方向	ORIENTATION	integer			否		Primitiveの方向(0:-,1:+)
3	GM_Primitive参照ID	PRIMITIVE	character varying(100)			否		
4	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.19. Reg_GM_CompositeCurve テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	GM_CompositeCurve管理		ﾃｰﾌﾞﾙ名	Reg_GM_CompositeCurve				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1,a1	GM_CompositeCurveのID
2	方向	orientation	integer			否		
3	ｼｰｹﾝｽ	SEQ	integer			否	a2	GM_OrientableCurveのｼｰｹﾝｽ
4	GM_OrientableCurve参照ID	Ocurve	character varying(100)			否	p1	
5	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.20. Reg_GM_Curve テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	GM_Curve管理		ﾃｰﾌﾞﾙ名	Reg_GM_CURVE				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	GM_CurveのID
2	ｼｰｹﾝｽ	SEQ	integer			否	p2	GM_LineStringのｼｰｹﾝｽ
3	GM_LineString参照ID	LineString	character varying(100)			否		
4	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.21. Reg_GM_LineString テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	GM_LineString管理		ﾃｰﾌﾞﾙ名	Reg_GM_LineString				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	GM_LineStringのID
2	ｼｰｹﾝｽ	SEQ	integer			否	p2	DirectPositionのｼｰｹﾝｽ
3	DirectPosition参照ID	DIRECTPOSITION	character varying(100)			否		
4	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.22. Reg_GM_Surface テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	GM_Surface管理		ﾃｰﾌﾞﾙ名	Reg_GM_Surface				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	GM_SurfaceのID
2	方向	orientation	integer					
3	GM_PolygonのID	GM_Polygon	character varying(100)			否	p2	
4	GM_SurfaceBoundaryのID	GM_SurfaceBoundary	character varying(100)			否	p3	
5	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.23. Reg_Exterior テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	Exterior管理		ﾃｰﾌﾞﾙ名	Reg_Exterior				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1,a1	GM_SurfaceBoundaryのID
2	GM_Ring参照ID	RING	character varying(100)			否		基盤データ0.5版はNULL値
3	方向	orientation	integer					
4	ｼｰｹﾝｽ	SEQ	integer			否	a2	GM_OrientableCurveのｼｰｹﾝｽ
5	GM_OrientableCurve参照ID	Ocurve	character varying(100)			否	p2	
6	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.24. Reg_Interior テーブル

用途: 空間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	Interior管理		ﾃｰﾌﾞﾙ名	Reg_Interior				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1,a1	GM_SurfaceBoundaryのID
2	Interior番号	INTERIORNO	integer			否	a2	Interior識別子
3	GM_Ring参照ID	Ring	character varying(100)			否		基盤データ0.5版はNULL値
4	方向	orientation	integer					
5	ｼｰｹﾝｽ	SEQ	integer			否	a3	GM_OrientableCurveのｼｰｹﾝｽ
6	GM_OrientableCurve参照ID	Ocurve	character varying(100)			否	p2	
7	ﾀｲﾑｽﾀﾝﾌﾟ	wktimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.25. Reg_TM_ReferenceSystem テーブル

用途: 時間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	TM_ReferenceSystem管理		ﾃｰﾌﾞﾙ名	Reg_TM_ReferenceSystem				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	
2	時間参照系名	name	character varying(100)			否		
3	有効範囲	dov	character varying(100)			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.26. Reg_TM_Period テーブル

用途: 時間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	TM_Period管理		ﾃｰﾌﾞﾙ名	Reg_TM_Period				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	
2	期間開始	period1	character varying(100)			否		TM_InstantのID
3	期間終了	period2	character varying(100)			否		TM_InstantのID

制約事項:

コード　:

フラグ　:

備考　:

1.2.27. Reg_TM_Instant テーブル

用途: 時間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	TM_Instant管理		ﾃｰﾌﾞﾙ名	Reg_TM_Instant				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	
2	TM_CalDate	caldate	character varying(100)			否		TM_CalDateのID

制約事項:

コード　:

フラグ　:

備考　:

1.2.28. Reg_TM_CalDate テーブル

用途: 時間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	TM_CalDate管理		ﾃｰﾌﾞﾙ名	Reg_TM_CalDate				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	
2	年号	EraName	character varying(50)			否		
3	時間参照系ID	rs	character varying(100)			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.29. Reg_tbl_caldate テーブル

用途: 時間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	tbl_caldate管理		ﾃｰﾌﾞﾙ名	Reg_tbl_caldate				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否	p1	
2	ｼｰｹﾝｽ	seq	integer			否	p2	
3	時間要素	caldate	character varying(100)			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.30. Reg_tbl_begunby テーブル

用途: 時間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	tbl_begunby管理		ﾃｰﾌﾞﾙ名	Reg_tbl_begunby				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否		TM_InstantのID
2	TM_Period	begunby	character varying(100)			否		TM_PeriodのID

制約事項:

コード　:

フラグ　:

備考　:

1.2.31. Reg_tbl_endedby テーブル

用途: 時間属性の構造維持管理に用いる。

ﾃｰﾌﾞﾙ名称	tbl_endedby管理		ﾃｰﾌﾞﾙ名	Reg_tbl_endedby				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	ID	character varying(100)			否		TM_InstantのID
2	endedby	endedby	character varying(100)			否		TM_PeriodのID

制約事項:

コード　:

フラグ　:

備考　:

1.2.32. WK_Point テーブル

用途: 描画用空間属性を管理する。

ﾃｰﾌﾞﾙ名称	WK_Point管理		ﾃｰﾌﾞﾙ名	WK_Point				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	id	character varying(100)			否		
2	データ種別	flggf	integer			否		0=レガシーデータ,1=地理情報標準データ
3	スキーマ名	schemaname	character varying(100)			否	p1,a5,b1	
4	パッケージ名	packagename	character varying(100)			否	p2,a6,b2	
5	クラス名	classname	character varying(100)			否	p3,a7,b3	
6	primitive_id	primitive_id	character varying(100)			否	p4	
7	表示角度	deg	integer					
8	シンボルファイル名	symbolimage	character varying(100)					
9	X座標	x	double precision			否	a1,b4,c2	
10	Y座標	y	double precision			否	a2,b5,c3	
11	TM_Period ID	tm_id	character varying(100)					
12	設置期間開始日付	b_period	character(8)			否	p5,a3,b6,c4	
13	設置期間終了日付	e_period	character(8)			否	p6,a4,b7,c5	
14	削除フラグ	flgdel	integer			否	p7	
15	更新ユーザ	username	character varying(100)					
16	タイムスタンプ	wktimestamp	timestamp(0) without time zone			否	p8	
17	クラス識別	classification	character varying(300)				c1	

制約事項:

コード　:

フラグ　:

備考　:

1.2.33. WK_Point_SL テーブル

用途:描画用空間属性を管理する。

ﾃｰﾌﾞﾙ名称	WK_Point_SL管理		ﾃｰﾌﾞﾙ名	WK_Point_SL				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	id	character varying(100)			否		
2	データ種別	flgfd	character varying(100)			否		0=レガシーデータ,1=地理情報標準データ
3	スキーマ名	schemaname	character varying(100)			否		
4	クラス名	classname	character varying(100)			否		
5	地物名	featurename	character varying(100)			否		
6	primitive_id	primitive_id	character varying(100)			否		
7	シンボル／テキスト種別	kind	integer					
8	表示角度	deg	integer					
9	テキスト	text	text					
10	TM_Period ID	tm_id	character varying(100)			否		
11	設置期間開始日付	b_period	character(8)			否		
12	設置期間終了日付	e_period	character(8)			否		
13	ＸＹ座標	coordinate	character varying(100)			否		
14	Ｘ座標	x	double precision					
15	Ｙ座標	y	double precision					
16	タイムスタンプ	wktimestamp	timestamp without time zone					
17	クラス識別	classification	character varying(300)					

制約事項:

コード　:

フラグ　:

備考　:

1.2.34. WK_Curve テーブル

用途: 描画用空間属性を管理する。

ﾃｰﾌﾞﾙ名称	WK_Curve管理		ﾃｰﾌﾞﾙ名	WK_Curve				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	id	character varying(100)			否		
2	データ種別	flggf	integer			否		0=レガシーデータ,1=地理情報標準データ
3	ﾚｺｰﾄﾞ継続番号	ridx	integer			否	p5,a5,b12,c12,d10	
4	スキーマ名	schemaname	character varying(100)			否	p1,a1,b7,c1	
5	パッケージ名	packagename	character varying(100)			否	p2,a2,b8,c2	
6	クラス名	ClassName	character varying(100)			否	p3,a3,b9,c3	
7	GM_Primitive ID	Primitive_id	character varying(100)			否	p4,a4,b11,c11,d9	
8	座標列(CSV形式)	Coordinate	character varying(4000)			否		
9	最小X座標	minX	double precision			否	b1,c4,d2	
10	最小Y座標	minY	double precision			否	b2,c5,d3	
11	最大X座標	maxX	double precision			否	b3,c6,d4	
12	最大Y座標	maxY	double precision			否	b4,c7,d5	
13	TM_Period ID	TM_ID	character varying(100)					
14	設置期間開始日付	B_Period	character(8)			否	p6,a6,b5,c8,d6	
15	設置期間終了日付	E_Period	character(8)			否	p7,a7,b6,c9,d7	
16	削除フラグ	flgdel	integer			否	p8,a8,b10,c10,d8	
17	更新ユーザ	username	character varying(100)					
18	ﾀｲﾑｽﾀﾝﾌﾟ	WKtimestamp	timestamp(0) without time zone			否	p9,a9	
19	クラス識別	classification	character varying(300)				d1	

制約事項:

コード　:

フラグ　:

備考　:

1.2.35. WK_Surface テーブル

用途: 描画用空間属性を管理する。

ﾃｰﾌﾞﾙ名称	WK_Surface管理		ﾃｰﾌﾞﾙ名	WK_Surface				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	id	character varying(100)			否		
2	データ種別	flggf	integer			否		0=レガシーデータ,1=地理情報標準データ
3	ﾚｺｰﾄﾞ継続番号	ridx	integer			否	p5,a5,b11,c11,d9	
4	スキーマ名	schemaname	character varying(100)			否	p1,a1,b7,c1	
5	パッケージ名	packagename	character varying(100)			否	p2,a2,b8,c2	
6	クラス名	classname	character varying(100)			否	p3,a3,b9,c3	
7	GM_Primitive ID	primitive_id	character varying(100)			否	p4,a4,b10,c10,d8	
8	外周座標列(CSV形式)	exterior	character varying(4000)					
9	内周座標列(CSV形式)	interior	character varying(4000)					
10	最小X座標	minx	double precision			否	b1,c4,d2	
11	最小Y座標	miny	double precision			否	b2,c5,d3	
12	最大X座標	maxx	double precision			否	b3,c6,d4	
13	最大Y座標	maxy	double precision			否	b4,c7,d5	
14	TM_Period ID	tm_id	character varying(100)					
15	設置期間開始日付	b_period	character(8)			否	p6,a6,b5,c8,d6	
16	設置期間終了日付	e_period	character(8)			否	p7,a7,b6,c9,d7	
17	削除フラグ	flgdel	integer			否	p8,a8	
18	更新ユーザ	username	character varying(100)					
19	タイムスタンプ	wktimestamp	timestamp(0) without time zone			否	p9,a9	
20	クラス識別	classification	character varying(300)				d1	

制約事項:

コード　:

フラグ　:

備考　:

1.2.36. WK_Period テーブル

用途: 表示用時間属性を管理する。

ﾃｰﾌﾞﾙ名称	WK_Period		ﾃｰﾌﾞﾙ名	WK_Period				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	id	character varying(100)			否	p1	
2	開始日付	B_Period	character(8)			否		
3	終了日付	E_Period	character(8)			否		
4	ﾀｲﾑｽﾀﾝﾌﾟ	WKtimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.37. WK_Instant テーブル

用途: 表示用時間属性を管理する。

ﾃｰﾌﾞﾙ名称	WK_Instant		ﾃｰﾌﾞﾙ名	WK_Instant				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	id	character varying(100)			否	p1	
2	CalDate	CalDate	character(8)			否		
3	ﾀｲﾑｽﾀﾝﾌﾟ	WKtimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.38. WK_CalDate テーブル

用途: 表示用時間属性を管理する。

ﾃｰﾌﾞﾙ名称	WK_CalDate		ﾃｰﾌﾞﾙ名	WK_CalDate				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	id	character varying(100)			否	p1	
2	CalDate	CalDate	character(8)			否		
3	ﾀｲﾑｽﾀﾝﾌﾟ	WKtimestamp	timestamp(0) without time zone			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.39. FeatureInstance テーブル

用途：地物インスタンスの管理に用いる。

ﾃｰﾌﾞﾙ名称	地物属性		ﾃｰﾌﾞﾙ名	FeatureInstance				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	ID	id	character varying(100)			否	p1	
2	データ種別	flggf	integer			否		地理情報標準に準拠したデータ 0=非準拠,1=準拠
3	レコード継続番号	ridx	integer			否	p1	
4	スキーマ名	schema_name	character varying(100)			否	p1	
5	パッケージ名	package_name	character varying(100)			否	p1	
6	クラス名	class_name	character varying(100)			否	p1	
7	属性値(XML形式)	xmlvalue	character varying(2048)			否		
8	削除フラグ	flgdel	integer			否	p1	
9	有効期間開始	b_period	character(8)			否	p1	
10	有効期間終了	e_period	character(8)			否		
11	タイムスタンプ	wktimestamp	timestamp(0) without time zone			否		

制約事項：

コード　：

フラグ　：

備考　：

1.2.40. CodeList テーブル

用途: コードリストの管理に用いる

ﾃｰﾌﾞﾙ名称	コードテーブル		ﾃｰﾌﾞﾙ名	codelist				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	スキーマ名	schema_name	character varying(100)			否		
2	パッケージ名	package_name	character varying(100)			否		
3	クラス名	class_name	character varying(100)			否		
4	コード	code1	character varying(100)			否		
5	値	value	character varying(100)			否		

制約事項:

コード　:

フラグ　:

備考　:

1.2.41. AppSchemaManagement テーブル

用途：応用スキーマの管理に用いる。

ﾃｰﾌﾞﾙ名称	応用スキーマ管理		ﾃｰﾌﾞﾙ名	AppSchemaManagement				
項番	名称	列名	データ型	属性	初期値	NULL値可否	ｲﾝﾃﾞｯｸｽ	記事
1	スキーマ名	schema_name	character varying(30)			否		スキーマ名称
2	パッケージ名	package_name	character varying(100)			否		パッケージ名称
3	クラス名	class_name	character varying(100)			否		クラス名称
4	データ種別	flggf	integer			否		地理情報標準に準拠したデータか識別するためのフラグ 0=非準拠,1=準拠
5	クラス日本語名	class_jpname	character varying(100)			否		クラス日本語名称
6	地物構成分類	gffeaturetype	character varying(100)			否		flggf=0 の場合は先頭が"GF_"で始まり,flggf=1 の場合は先頭が"LF_"で始まる G(L)F_FeatureType G(L)F_ThematicAttributeType G(L)F_AttributeType G(L)F_SpatialAttributeType GF_TemporalAttributeType GF_AssociationType GF_AssociationRole
7	属性名称	attribute_name	character varying(100)					gffeaturetype が"G(L)F_ThematicAttributeType", "G(L)F_AttributeType","G(L)F_SpatialAttributeType", "GF_TemporalAttributeType"のレコードで有効. (上記以外のレコードでは NULL 値を設定する)
8	属性日本語名称	attribute_jpname	character varying(100)					gffeaturetype が"G(L)F_ThematicAttributeType", "G(L)F_AttributeType","G(L)F_SpatialAttributeType", "GF_TemporalAttributeType"のレコードで有効 (上記以外のレコードでは NULL 値を設定する) 属性表示の際の項目名
9	データ型	datatype	character varying(100)					gffeaturetype="GF_InheritanceRelation"以外のレコードで有効. (gffeaturetype="GF_InheritanceRelation"のレコードでは NULL 値を設定)
10	図形形状フラグ	wkspatialtype	character varying(100)					gffeaturetype="G(L)F_FeatureType"のレコードで有効. (上記以外のレコードでは NULL 値を設定する) 形状番号[:形状番号]の形式で指定する.

								形状番号は 1=点,2=線,3=面 形状を持たない地物クラスの場合は 0=形状なしとする.
11	属性参照フラグ	flgattributeref	integer					gffeaturetype が "G(L)F_ThematicAttributeType", "G(L)F_AttributeType","G(L)F_SpatialAttributeType", "GF_TemporalAttributeType"のレコードで有効. (上記以外のレコードでは NULL 値を設定する) 0=直接属性を持つ 1=クラス型または関連を示すため, ID で参照する
12	属性への Xpath	refxpath	character varying(1024)					gffeaturetype="GF_InheritanceRelation"以外のレコードで有効. (gffeaturetype="GF_InheritanceRelation"のレコードでは NULL 値を設定) ルートから当該属性までの Xpath ルートから Class_Name までのパスが当該クラスへの Xpath となる
13	空間属性 ID 参照 Xpath	refsp	character varying(100)					gffeaturetype="G(L)F_SpatialAttributeType"のレコードで有効. (上記以外のレコードでは NULL 値を設定する) 当該クラスのタグを先頭とした空間属性までの Xpath. (Wkspatialtype の値が"形状番号:形状番号[:形状番号]の場合", "～/pkgspatial.GM_Primitive" となり,実際の属性は pkgspatial.GM_Primitive を継承したインスタンスが出現する形式となる)
14	時間属性 ID 参照 Xpath	reftm	character varying(100)					gffeaturetype="GF_TemporalAttributeType"のレコードで有効. (上記以外のレコードでは NULL 値を設定する) 当該クラスのタグを先頭とした時間属性までの Xpath
15	有効期間フラグ	flgperiod	integer					gffeaturetype="GF_TemporalAttributeType"のレコードで有効 (上記以外のレコードでは NULL 値を設定する)
16	関連種別	flgassociation	integer			否?		gffeaturetype="GF_AssociationType","GF_AssociationRole"以外の時は 0 0=なし,1=アグレゲーション,2=コンポジション,3=関連

17	多重度管理フラグ	flgcardinality	integer					gffeaturetype が "GF_FeatureType","GF_InheritanceRelation"以外のレコードで有効 上記以外のレコードでは NULL 値を設定する 0=なし,1=あり
18	順列管理フラグ	flgsequence	integer					flgcardinality と対で使用する gffeaturetype が "GF_FeatureType","GF_InheritanceRelation"以外のレコードで有効 上記以外のレコードでは NULL 値を設定する 0=なし,1=あり(順列を管理する属性)
19	ロール名1(自身)	role1	character varying(100)					gffeaturetype が "GF_AssociationType","GF_AssociationRole"のレコードで有効 上記以外のレコードでは NULL 値を設定する 上記のレコードでも NULL 値の場合もある
20	ロール名2(相手)	role2	character varying(100)					gffeaturetype が "GF_AssociationType","GF_AssociationRole"のレコードで有効 上記以外のレコードでは NULL 値を設定する 上記のレコードでも NULL 値の場合もある
21	関連名称	association_name	character varying(100)					gffeaturetype が "GF_AssociationType","GF_AssociationRole"のレコードで有効 上記以外のレコードでは NULL 値を設定す
22	関連日本語名称	association_jpname	character varying(100)					association_name と同じ
23	継承元地物名称	inheritancerelation	character varying(100)					gffeaturetype="G(L)F_FeatureType"のレコードで有効 (上記以外のレコードでは NULL 値を設定する) 継承元の地物クラス名
24	テーブル参照フラグ	flgreftbl	integer					0=属性をテーブルに格納する,1=属性をテーブルに格納しない
25	相手のスキーマ名称	refschema_name	character varying(100)					関連相手のスキーマ名称 gffeaturetype="G(L)F_FeatureType"以外のレコードで有効 (上記以外のレコードでは NULL 値を設定する)
26	相手のパッケージ名称	refpackage_name	character varying(100)					関連相手のパッケージ名称 gffeaturetype="G(L)F_FeatureType"以外のレコードで有効 (上記以外のレコードでは NULL 値を設定する)
27	相手のクラス名称	refclass	character varying(100)					関連相手のクラス名称 gffeaturetype="G(L)F_FeatureType"以外のレコードで有効 (上記以外のレコードでは NULL 値を設定する)
28	地物インスタンステーブル名称	itbl_name	character varying(100)					現在は使用していない(将来使用する)
29	地物インスタン	itbl_jpname	character					現在は使用していない(将来使用

	ステーブル日本語名称		varying(100)					する)
30	属性格納レコードインデックス	ridx	integer					地物インスタンステーブルの属性格納レコード番号
31	属性格納カラム名称	itbl_column_name	character varying(100)					地物インスタンステーブルの属性格納列名称
32	属性格納カラム日本語名称	itbl_column_jpname	character varying(100)					地物インスタンステーブルの属性格納列日本語名称
33	インスタンステーブル参照フラグ	flginstance	integer					flggf=0(汎用データ)の場合に有効 属性データの格納先を設定する 0=格納なし,1=XML のみ(FeatureInstance テーブル),2=インデックステーブル, 3=1 と 2 の両方
34	検索用インデックテーブル格納フラグ	flgsidx	integer					0=格納しない,1=格納する
35	検索用インデックテーブル名称	sidxtbl_name	character varying(100)					flgsidx=1 の時有効
36	検索用インデックテーブル格納列名称	sidxtbl_column_name	character varying(100)					flgsidx=1 の時有効
37	検索用インデックテーブル格納列日本語名称	sidxtbl_column_jpname	character varying(100)					flgsidx=1 の時有効
38	コードテーブル参照フラグ	flgcode	integer					0=参照しない,1=参照する
39	コードテーブル参照キー1	code_schema_name	character varying(100)					flgcode=1 の時有効
40	コードテーブル参照キー2	code_package_name	character varying(100)					flgcode=1 の時有効
41	コードテーブル参照キー3	code_class_name	character varying(100)					flgcode=1 の時有効
42	コードテーブル名称	codetbl_name	character varying(100)					flgcode=1 の時有効
43	コードテーブルのコード格納列名称	codetbl_column_name	character varying(100)					flgcode=1 の時有効
44	コードテーブルの値格納列名称	code_value	character varying(100)					flgcode=1 の時有効
45	タグ出現順序	attributeseq	integer					0=出現しない,1～=出現順序
46	属性表示フラグ	flgview	integer					0=非表示,1～=表示順序(将来対応)
47	編集可フラグ	flgedit	integer					flggf=0(汎用データ)の場合に有効 gffeaturetype="LF_FeatureType"のレコードで有効 0=編集不可,1=編集可
48	タイムスタンプ	wktimestamp	timestamp without time zone					
49	スキーマ名+パッケージ名+クラス名	classification	character varying(300)					schema_name+'_'+package_name+'_'+class_name

制約事項:

コード　:

フラグ　:

備考　:

道路 GIS

Software Development Kit

開発者マニュアル

平成 16 年 3 月

【道路 GIS SDK 開発者マニュアル目次】

1. システムの概要 3

2. 道路 GIS アプリケーションの構成 4

3. 道路 GIS WEB API の利用サンプル 6

3.1. サーバ／クライアント間の通信 6

3.1.1. XML データによる通信 6

3.1.2. 画像データの取得 8

3.2. ユーザ認証 9

3.3. サンプルプログラム 10

4. API 仕様 11

4.1. セッション処理 API 12

4.1.1. セッションの初期化(*InitSesstion*) 12

4.1.2. セッションの破棄(*TerminateSession*) 13

4.2. イメージ処理 API 14

4.2.1. 地図画像の取得(*GetMapImage*) 14

4.2.2. 索引図画像の取得(*GetIndexMapImage*) 15

4.2.3. 印刷イメージの取得(*PrintMap*) 16

4.3. 地物の表示切替 API 17

4.3.1. 地物クラスの表示状態の取得(*GetFeatureClassState*) 17

4.3.2. 地物クラスの表示状態の設定(*SetFeatureClassState*) 18

4.4. データ範囲の取得 API 19

4.4.1. データ範囲の取得(*GetExtent*) 19

4.5. 描画法情報の設定・取得処理 API 20

4.5.1. 地物の描画法情報の取得(*GetFeatureClassPortraiture*) 20

4.5.2. 地物の描画法情報の設定(*SetFeatureClassPortraiture*) 22

4.6. 時点の設定・取得処理 API 23

4.6.1. 時点の取得(*GetTimePoint*) 23

4.6.2. 時点の設定(*SetTimePoint*) 24

4.7. 位置検索処理 API 25

4.7.1. 住所検索情報の取得(*GetPlaceNameCatalog*) 25

4.7.2. 住所の位置情報の取得(*GetPlaceNameSearchInfo*) 27

4.7.3. 目標物検索情報の取得(*GetLandMarkCatalog*) 30

4.7.4. 目標物の位置情報の取得(*GetLandMarkSearchInfo*) 32

4.7.5. キロポスト検索情報の取得(*GetSpotMarkCatalog*) 34

4.7.6. キロポストの位置情報の取得(*GetSpotMarkSearchInfo*) 36
4.8. 地物定義の取得 API 38
4.8.1. 地物一覧の取得（GetFeatureClassList*）* 38
4.8.2. 地物の詳細情報の取得（GetFeatureClassInfo*）* 39
4.8.3. コードリストの取得（GetFeatureClassList) 41
4.9. 属性情報の取得 API 42
4.9.1. 一点を指定して検索（*ScanPoint）* 42
4.9.2. 一点を指定して検索（*GetFeatureByScanPoint）* 44
4.9.3. ID を指定して検索（*GetFeatureByID）* 47
4.10. メモ機能の API 49
4.10.1. メモの作成(*CreateMemo）* 49
4.10.2. メモの取得（*FindMemo）* 50
4.10.3. メモの更新（*UpdateMemo）* 52
4.10.4. メモの削除(*RemoveMemo）* 53
4.10.5. メモ分類情報の取得（*GetMemoCategory）* 54
4.10.6. メモ分類情報の作成(*CreateMemoCategory）* 55
4.10.7. メモ分類情報の更新(*UpdateMemoCategory）* 56
4.10.8. メモ分類情報の削除（*RemoveMemoCategory）* 57
4.11. 基盤データの取得と更新 API 58
4.11.1. 基盤データの取得（*GetRoadData）* 58
4.11.2. 基盤データの更新（*UpdateRoadData*) 59

1. システムの概要

道路 GIS SDK の基本構成を下図の通りです。

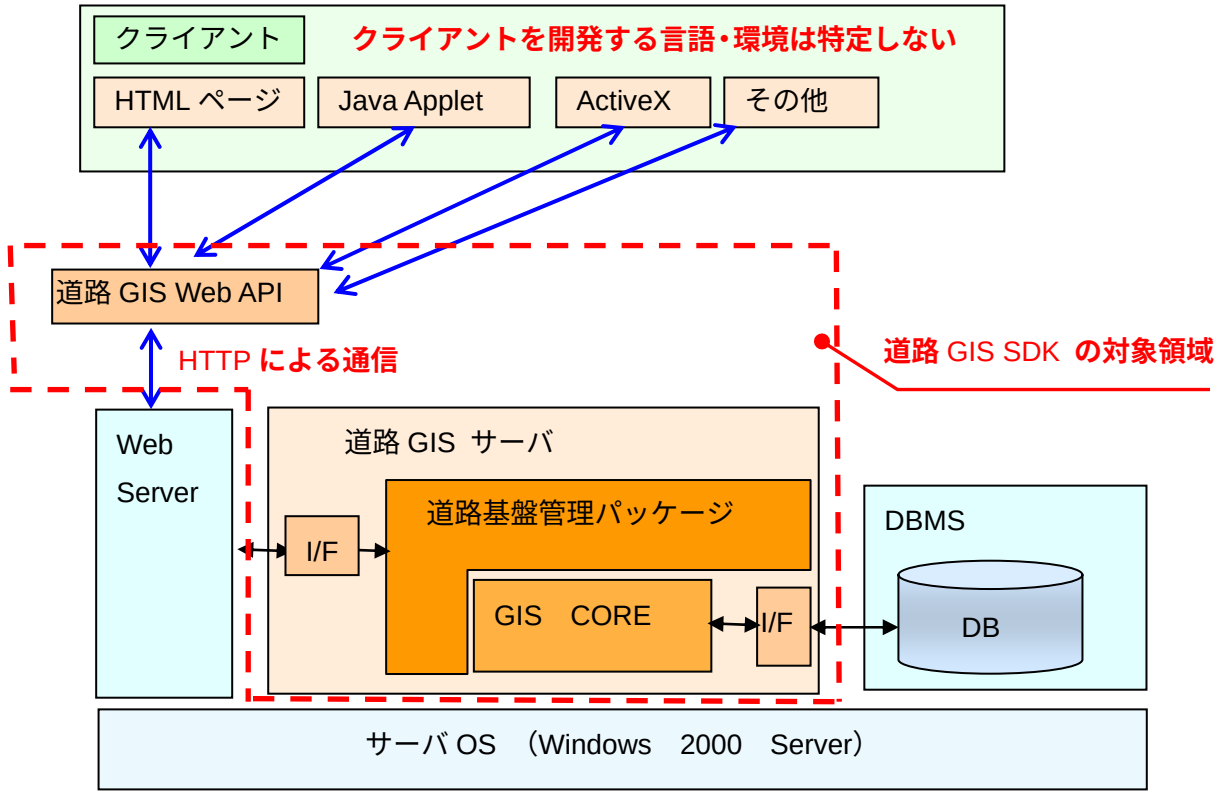


図 1 道路 GIS SDK の構成

道路 GIS SDK のサーバ部分は、道路基盤データを操作・処理するための **GIS CORE** モジュールと、GIS CORE モジュールの機能を内包し、その他背景データなど、道路基盤データ以外で必要となるデータとあわせて実務に必要な機能を提供する**道路基盤管理パッケージ**から構成されます。これらサーバ部分は、Java サーブレットで構成され、Web サーバーを介してその機能を公開します。・

道路 GIS アプリケーションのクライアント部分は、道路 GIS SDK の提供する**道路 GIS Web API** を用いて、ユーザが画面上で行った操作に対応した問い合わせを行い、その処理結果を取得し、その結果を解析してアプリケーションの画面に反映します。

本マニュアルは、道路 GIS SDK を用いて、道路基盤データを利用したアプリケーション開発のための方法を説明するものです

なお、本文中でのサンプルプログラムの説明は Java2 SDK 1.4.1 を用いています。

2. 道路 GIS アプリケーションの構成

道路 GIS アプリケーションの基本構成を**エラー! 参照元が見つかりません。**に示します。

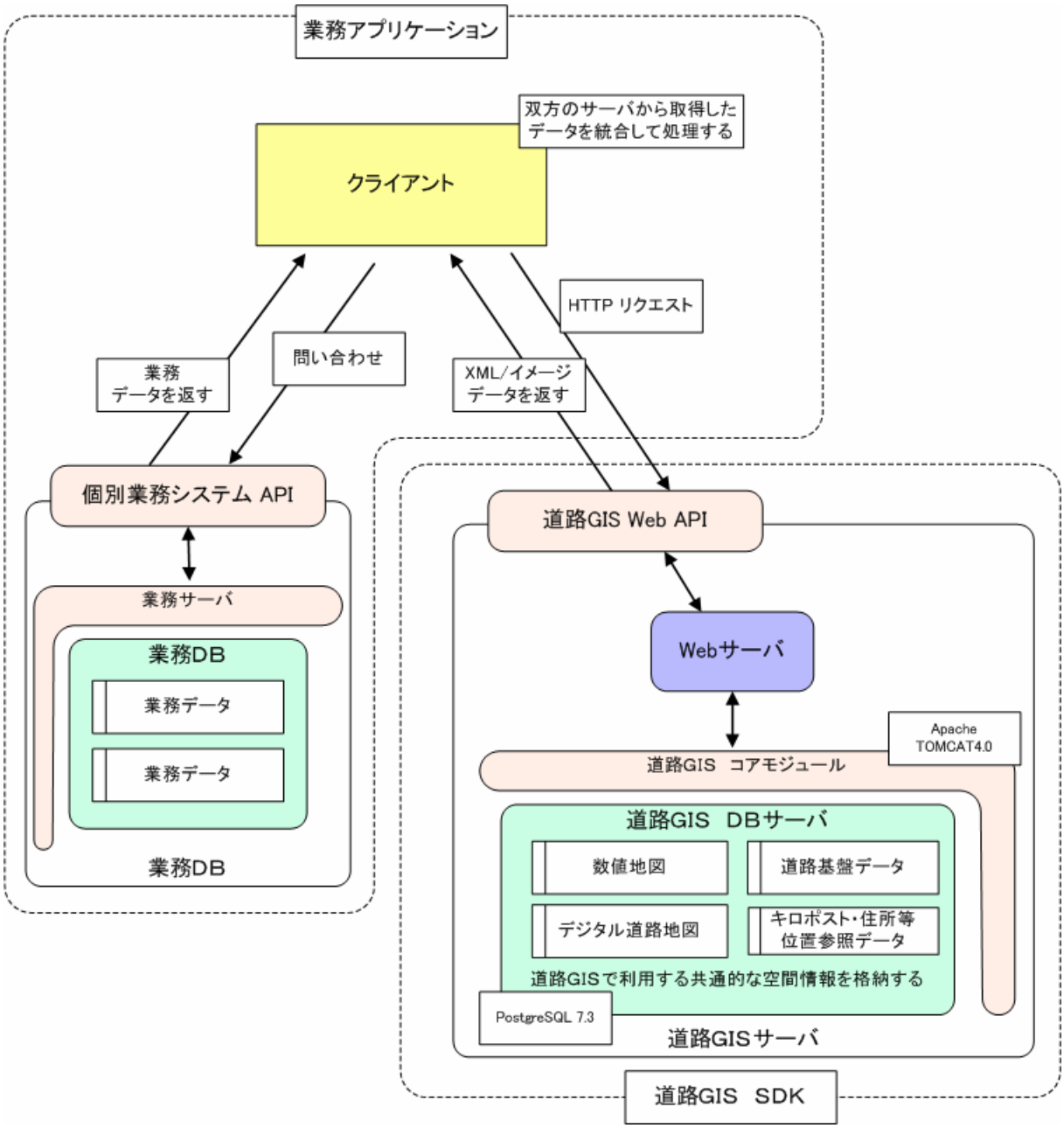


図 2 道路 GIS アプリケーションの構成

道路 GIS SDK を用いてアプリケーションを開発する場合には、道路基盤データやデジタル道路地図など道路ＧＩＳの基本的な機能を提供する**道路 GIS サーバ**と、その他業務機能を実装した**業務サーバ**の二つのサーバを利用します。**クライアント**は、道路 GIS サーバから取得される道路関係の基本情報と、業務サーバから取得される個別業務の情報を併合し、望む処理を実行します。

道路 GIS SDK を用いて、アプリケーション開発を行う場合には、具体的な手順は以下の通りです。

1) IS サーバに、道路 GIS WebAPI を使って、指定範囲の地図や、道路地物の属性情報などを問い合わせる
2) 道路 GIS サーバから、問い合わせ結果を取得する
   2-1) 属性情報などは、XML データとして取得されるので、API の仕様に従い解読し、必要な処理を行う
   2-2) 図形データを要求した場合には、地図画像として取得されるので、画面上に表示する等の処理を行う
3) 個別アプリケーションを構成するための情報を、業務サーバに問い合わせる
4) 業務サーバから取得された情報を取得する
5) 必要に応じて、業務サーバから取得した情報を地図上に重ね合わせる等の処理を行う

道路 GIS サーバから指定範囲を取得して地図画像を取得し、業務サーバから取得した情報を重ね合わせた例をに示します。

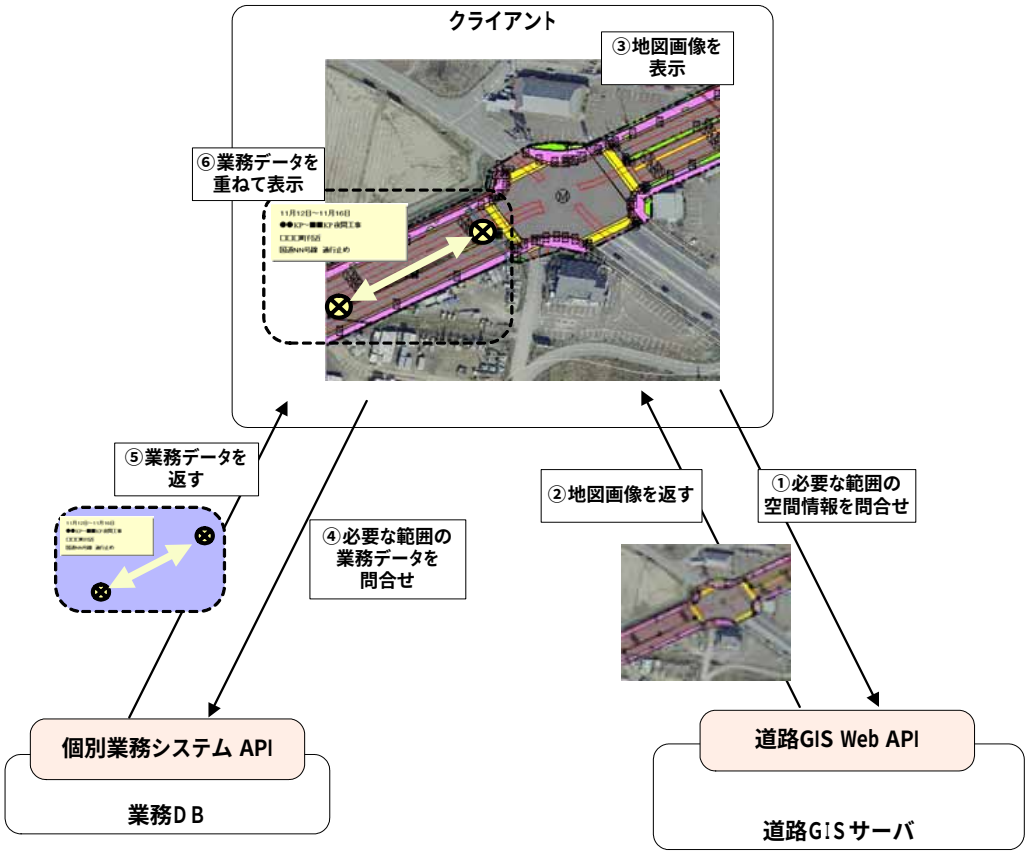


図 3　情報の重ね合わせの例

このようにして、道路基盤データの情報と業務情報を組み合わせて道路 GIS アプリケーションを開発します。

3. 道路 GIS Web API の利用サンプル

ここでは、道路 GIS Web API を用いて、アプリケーションを構築する際の基本的な手順を、実際のプログラムを通して説明します。

3.1. サーバ／クライアント間の通信

道路 GIS SDK では、サーバ／クライアント間の通信には、HTTP プロトコルを用いており、情報の受け渡しには、通常のデータ連携には XML を、地図画像の取得には、image データによる情報の取得を行います。

ここでは、基本的な情報の受け渡しの方法を説明します。

3.1.1. XML データによる通信

まず、指定の API により、道路 GIS サーバにリクエストを要求する方法を示します。

サーバへのリクエストには、HTTP による POST メソッドを用います。

```
private Node RequestXML(String api, String arg)
{
	Document	responseNode;
	URL url;
	URLConnection ucon;
	String urlstr = serverURL + api;	//リクエストにはサーバのURL+API名を用いる
	try{
		url = new URL(urlstr);
		ucon = url.openConnection();	//コネクションを開く
	} catch (Exception e) {
		e.printStackTrace();
		return null;
	}
	try {
		// HTTP接続
		String req = "";
		if( arg.length() > 0 ){
			//リクエストに用いる文字列を生成する
			String tmpstr = "Request=<Request>" + arg + "</Request>";
			req = new String(tmpstr.getBytes(), "MS932");
		}else{
			req = "";
		}
		// HTTP/POSTによりリクエストの送信
		ucon.setDoOutput(true);
		ucon.setDoInput(true);
		ucon.setUseCaches(false);
		Enumeration keys = paramtable.keys();

		while(keys.hasMoreElements()){
			String key = (String)keys.nextElement();
			ucon.setRequestProperty(key, (String)paramtable.get(key));
		}
		// 出力ストリームを取得し、リクエストを送る
		PrintWriter out = new PrintWriter(ucon.getOutputStream());
		out.print(req);
		out.close();
	} catch (Exception e) {
		e.printStackTrace();
		return null;
	}
```

リクエストの結果を受け取り、データを取得する際には、DOMを用います。ここではサーバからのレスポンスを取得し、UTF-8の文字列にエンコードして受け取っています。

```
		try{
			// 取得したXMLメッセージをDOMへ
			DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
			DocumentBuilder db = null;
			db = dbf.newDocumentBuilder();

			if(ucon.getContentType().toLowerCase().indexOf("jis") == -1 ){
				// Unicodeできた場合はそのままParserへ
				responseNode = db.parse(ucon.getInputStream());
			}else{
				// Shift-JISできた場合はEncodingする。(機種依存文字対応)
				//    byteArray -> MS932形式で読み込み ->
				// Unicode -> ParserへはUTF-8形式で渡す -> Unicode
				DataInputStream input = new DataInputStream(ucon.getInputStream());
				ByteArrayOutputStream rbuf = new ByteArrayOutputStream();

				int retval = 1;
				while(true){
					byte result[];
					result = new byte[1024];
					retval = input.read(result);
					if(retval < 1) break;
					rbuf.write(result, 0, retval);
				}

				//MS932形式で読み込み
				String xmlstr = new String(rbuf.toByteArray(), "MS932");
				int idx = xmlstr.indexOf("?>");
				//UTF-8に変換
				String tmpstr = xmlstr.substring(0,idx)
					.toLowerCase().replaceAll("shift_jis","UTF-8");
				xmlstr = tmpstr + xmlstr.substring(idx, xmlstr.length());
				//UTF-8としてパーサに渡す
				responseNode = db.parse(new ByteArrayInputStream(
					xmlstr.getBytes("UTF-8")));
				}

			} catch (Exception e) {
				e.printStackTrace();
				return null;
			}
		//DOMノードを返す。
		return responseNode;
	}
```

3.1.2. 画像データの取得

画像データの場合も、同様に、HTTP/POST によるリクエストを行います。

リクエスト結果は、画像データとして取得します。

```
public Image RequestImage(String api, String arg)
{
	try {
		// HTTP接続
		String urlstr = serverURL + api;
		URL url = new URL(urlstr);
		String req = "";
		//リクエストを生成する。
		if( arg.length() > 0 ){
			String tmpstr = "Request=<Request>" + arg + "</Request>";
			req = new String(tmpstr.getBytes(), "MS932");
		}else{
			req = "";
		}

		// POSTメソッドでリクエストを行う
		URLConnection ucon = url.openConnection();
		ucon.setDoOutput(true);
		ucon.setDoInput(true);
		ucon.setUseCaches(false);
		Enumeration keys = paramtable.keys();
		while(keys.hasMoreElements()){
			String key = (String)keys.nextElement();
			ucon.setRequestProperty(key, (String)paramtable.get(key));
		}
		PrintWriter out = new PrintWriter(ucon.getOutputStream());
		out.print(req);
		out.close();

		//結果をJPEG画像として取得する
		JPEGImageDecoder decoder = JPEGCodec.createJPEGDecoder(ucon.getInputStream());
		BufferedImage bufImage = decoder.decodeAsBufferedImage();
		return bufImage;
	} catch (Exception e) {
		e.printStackTrace();
		return null;
	}
}
```

3.2. ユーザ認証

道路 GIS SDK では、ログイン時にユーザ認証を行うことにより、システムの利用権限やリソースの管理を行っています。

Login メソッドに成功すると、Server との間のセッション ID が取得されます。このセッション ID をリクエストのパラメータに付与することにより、同一ユーザからのログインであることを保障します。

```
StringnNameList[] = {"SessionID","Permmision","ReturnFlag"};

boolean retVal = false;
String Arg;
Node responseNode;
Node returnValue;
Node categoryNode;
NodeList childNodes;
Node component;

// LOGINを行い、セッションを取得する
Arg = "<UserName>" + user + "</UserName>" +
            "<PassWord>" + pass + "</PassWord>";
mCommunicator.events(RGISEventHandler.EVENT_LOGIN, Arg);
responseNode = mCommunicator.getResponseNode();

if( responseNode == null )      return false;
mUserName = user;

// 戻り値取得
childNodes = responseNode.getChildNodes();
if (childNodes != null) {
      int numberChildNodes = childNodes.getLength();
      for (int loopIndex = 0; loopIndex < numberChildNodes; loopIndex++ ){
            component = childNodes.item(loopIndex);
            if( component.getNodeType() == Node.ELEMENT_NODE ){
                  int nidx;
                  for( nidx = 0; nidx < nNameList.length; nidx++){
                        if( nNameList[nidx].compareToIgnoreCase(component.
                                    etNodeName()) == 0 )      break;
                        }
                        Node fchild = component.getFirstChild();
                        String tval;
```

```
                        switch(nidx){
                                case 0:  //セッションIDを取得
                                        mID = fchild.getNodeValue();
                                        break;
                                case 1:  //権限レベルを取得
                                        Long cod = Long.valueOf(fchild.getNodeValue());
                                        mPermmision = cod.longValue();
                                        mPermmision = 2;
                                        break;
                                case 2:  //ログインの可否を設定
                                        tval = fchild.getNodeValue().toLowerCase();
                                        retVal =
                                                tval.compareTo("true") == 0 ? true : false;
                                        break;
                                }
                        }
                }
        }


        if( retVal == false )    return false;
        if(mServiceType == RGISEventHandler.SERVICE_SERVLET){
                //セッションIDをクッキーに保持
                mCommunicator.setParameter("Cookie", "JSESSIONID="+mID);
                mCommunicator.setParameter("USERNAME", user);
                mCommunicator.setParameter("Login", "true");
        }
}
```

セッションを確保した後には、その他の API を用いることで、クライアントから道路 GIS サーバを呼び出し、各種機能を利用することが可能になります。

3.3. サンプルプログラム

サーバとの間でユーザ認証を行い、その後サーバ情報を取得して対象範囲の全域図を取得するまでのサンプルを巻末に添付します。

4. API 仕様

道路 GIS スタータキットが外部に対して公開する API は、HTTP プロトコルによって伝送されます。下に、API のリストとその概要、およびレスポンスの種類を示します。

分類	API名称	概要	レスポンスの種類
セッション処理	*InitSession*	セッションを開始する	text/xml
	TerminateSession	セッションを終了する	text/xml
イメージ処理	*GetMapImage*	地図画像を取得する	image/png, image/jpeg
	GetIndexMapImage	索引図を取得する	image/png, image/jpeg
	PrintMap	印刷イメージを取得する	image/png,
地物の表示切替	*GetFeatureClassState*	特定の地物クラスの表示状態を取得する	text/xml
	SetFeatureClassState	特定の地物クラスの表示状態を設定する	text/xml
データ範囲の取得	*GetExtent*	システムで扱うことのできるデータ範囲を取得する	text/xml
描画法	*GetFeatureClassPortraiture*	特定の地物の描画法を取得する	text/xml
	SetFeatureClassPortraiture	特定の地物の描画法を設定する	text/xml
時点処理	*GetTimePoint*	問い合わせを行う時点を取得する	text/xml
	SetTimePoint	問い合わせを行う時点の設定する	text/xml
位置検索	*GetPlaceNameCatelog*	住所検索のための情報を取得する	text/xml
	GetPlaceNameSearchInfo	住所の位置情報を取得する	text/xml
	GetLandMarkCatalog	目標物検索のための情報を取得する	text/xml
	GetLandMarkSearchInfo	目標物の位置情報を取得する	text/xml
	GetSpotMarkCatalog	キロポスト検索のための情報を取得する	text/xml
	GetSpotMarkSearchInfo	キロポストの位置情報を取得する	text/xml
地物定義の取得	*GetFeatureClassList*	地物一覧を取得する	text/xml
	GetFeatureClassInfo	地物の詳細情報を取得する	text/xml
	GetCodeList	コードリストを取得する	text/xml
属性情報の取得	*ScanPoint*	ある地点を含む地物の属性情報を取得する	text/xml
	GetFeatureByScanPoint	ある地点を含む地物の属性情報を取得する	text/xml
	GetFeatureByID	地物IDを指定して地物情報を取得する。	text/xml
メモ機能	*CreateMemo*	メモの生成を行う	text/xml
	FindMemo	特定範囲のメモを取得する	text/xml
	UpdateMemo	メモの内容を更新する	text/xml
	RemoveMemo	メモの削除を行う	text/xml
	GetMemoCategory	登録されているメモカテゴリのリストを取得する	text/xml
	CreateMemoCategory	新しくメモカテゴリを登録する	text/xml
	UpdateMemoCategory	メモカテゴリを更新する	text/xml
	RemoveMemoCategory	メモカテゴリを削除する	text/xml
基盤データの取得と更新	*GetRoadData*	特定条件に合致ずる道路基本データを取得する	text/xml
	UpdateRoadData	道路基本データの更新を行う	text/xml

1) サーブレットの呼び出しにあたっては、POST メソッドを利用するものとします
2) リスエスト引数はすべて XML 形式とします
3) レスポンスの形式については、一部の画像データを返すリスエストのほかは、XML 形式のデータを返します
4) リクエスト引数、レスポンスの形式などについては、個々の API の仕様に示します。

4.1. セッション処理 API

4.1.1. セッションの初期化(*InitSesstion*)

API	*InitSesstion*
機　能	セッションを初期化します
引数の形式	<Request> <UserName> *（ユーザー名）* </UserName > <PassWord> *（パスワード）* </PassWord> </Request>
引数の内容	ユーザ名、及びパスワード
レスポンスの種類	text/xml セッションIDと成功/失敗を返す
レスポンスの内容	<Response> <SessionID>*(セッションIDの番号、エラー時は0)*</SessionID> <Permmision>*(ユーザの権限:1:一般ユーザ、2:管理者、エラーは0)*</Permmision> <ReturnFlag>*(True/False)*</ReturnFlag> </Response>
解　説	セッションを生成します。 クライアントにはセッションIDと、成功の失敗を示す返り値フラグを返します。 セッション生成時にエラーがあった場合には、帰り値フラグにFalseを返します。

4.1.2. セッションの破棄(*TerminateSession*)

API	*TerminateSession*
機　能	セッションを破棄します
引数の形式	なし
引数の内容	なし
レスポンスの種類	text/xml セッションIDと成功/失敗を返す
レスポンスの内容	<Response> 　<SessionID>*(セッションIDの番号)*</SessionID> 　<ReturnFlag>*(True/False)*</ReturnFlag> </Response>
解　説	セッションを破棄します。 クライアントにはセッションIDと、成功の失敗を示す返り値フラグを返します。 セッション破棄時にエラーがあった場合には、帰り値フラグにFalseを返します。 ＊現行仕様 レスポンスのセッションIDの値は、“０”固定です。

4.2. イメージ処理 API

4.2.1. 地図画像の取得(*GetMapImage*)

API	*GetMapImage*
機　能	地図画像を取得します
引数の形式	text/xml
引数の内容	<Request> <NorthWest> <DirectPosition> <Coordinate>*(設定する表示範囲の最北の緯度)*</Coordinate> <Coordinate>*(設定する表示範囲の最西の経度)*</Coordinate> </DirectPosition> </NorthWest> <SouthEast> <DirectPosition> <Coordinate>*(設定する表示範囲の最南の緯度)*</Coordinate> <Coordinate>*(設定する表示範囲の最東の経度)*</Coordinate> </DirectPosition> </SouthEast> <XSize >*(要求する地図画像のXサイズ（ピクセル単位の整数値）*</XSize > <YSize >*(要求する地図画像のYサイズ（ピクセル単位の整数値）*</YSize > <ImageType> *(要求する画像フォーマット(image/jpegまたはimage/png))* </ImageType> <Scale>*（取得する画像の縮尺）*</Scale> </Request>
レスポンスの種類	image/jpeg　または　image/png
レスポンスの内容	指定フォーマットの画像データ
解　説	現在位置の画像を取得します。 クライアントには指定範囲の、指定形式の画像データを返します。 現在の位置などの情報はセッションが保持する値を使います。 画像が作成できない場合は、空データを返します。

4.2.2. 索引図画像の取得(*GetIndexMapImage*)

API	*GetIndexMapImage*
機　能	索引図の画像情報を取得します
引数の形式	text/xml
引数の内容	<Request> 　<Location>（*地域名*）</Location> 　<NorthWest> 　　< DirectPosition> 　　<Coordinate>*(索引図取得範囲で最北の緯度)*</Coordinate> 　　<Coordinate>*(索引図取得範囲で最西の経度)*</Coordinate> 　　</DirectPosition> 　</NorthWest> 　<SouthEast> 　　<DirectPosition> 　　<Coordinate>*(索引図取得範囲で最南の緯度)*</Coordinate> 　　<Coordinate>*(索引図取得範囲で最東の経度)*</Coordinate> 　　</DirectPosition> 　</SouthEast> 　<XSize >*(要求する地図画像のX サイズ（ピクセル単位の整数値）*</XSize> 　<YSize >*(要求する地図画像のY サイズ（ピクセル単位の整数値）*</YSize> 　<ImageType> 　　*(要求する画像フォーマット(image/jpeg またはimage/png))* 　</ImageType> </Request>
レスポンスの種類	image/jpeg　または　image/png
レスポンスの内容	指定フォーマットの画像データ
解　説	指定した地域名の索引図を取得します。 引数には、索引図取得範囲を示す、対角の緯度経度、取得する画像サイズ、画像フォーマットを指定する。 クライアントには指定形式の画像データを返します。 索引図が取得できない場合には、空データを返します。 ＊現行仕様 　レスポンスの画像データ形式はjpegのみ対応。

4.2.3. 印刷イメージの取得(*PrintMap*)

API	*PrintMap*
機　能	現在の表示位置を中心に、縮尺指定して、印刷イメージを取得します。
引数の形式	text/xml
引数の内容	<Request> 　<NorthWest> 　　<DirectPosition> 　　<Coordinate>*(設定する表示範囲の最北の緯度)*</Coordinate> 　　<Coordinate>*(設定する表示範囲の最西の経度)*</Coordinate> 　　</DirectPosition> 　</NorthWest> 　<SouthEast> 　　<DirectPosition> 　　<Coordinate>*(設定する表示範囲の最南の緯度)*</Coordinate> 　　<Coordinate>*(設定する表示範囲の最東の経度)*</Coordinate> 　　</DirectPosition> 　</SouthEast> 　<XSize >*(要求する地図画像のXサイズ（ピクセル単位の整数値）*</XSize > 　<YSize >*(要求する地図画像のYサイズ（ピクセル単位の整数値）*</YSize > 　<ImageType> 　　*(要求する画像フォーマット(image/jpegまたはimage/png))* 　</ImageType> 　<Scale>*（取得する画像の縮尺）*</Scale> </Request>
レスポンスの種類	image/jpeg　または　image/png　または　application/pdf
レスポンスの内容	指定フォーマットの画像データ
解　説	印刷イメージを取得します。 クライアントには指定形式の画像データを返します Imageのサイズが、image/jpegまたはimage/pngの場合は、表示画面を中心に指定範囲の図形をイメージに変換して返します。 application/pdfの場合は、指定のサーバ上で生成したPDF画像を返します（実装は未定） ＊現行仕様 　レスポンスの画像データ形式はjpegのみ対応。

4.3. 地物の表示切替 API

4.3.1. 地物クラスの表示状態の取得(*GetFeatureClassState*)

API	*GetFeatureClassState*
機　能	指定の地物クラスの表示状態を取得します
引数の形式	text/xml
引数の内容	<Request> 　<Path>*(状態の取得を要求する地物クラス（グループ）へのXpath)*</Path> </Request>
レスポンスの種類	text/xml
レスポンスの内容	#Xpath=/GroupOfRoadFundamentalFeature/GroupOfRoadBasicFeature/ GroupOfBasicSurface/GroupOfCrossSection/GroupOfRoadway #とした時のレスポンス例 <Response> <GroupOfRoadway> <Annotation>車道</Annotation> <Lane> <Annotation>車線</Annotation> <State>1</State> </Lane> <TransitionArea> <Annotation>すりつけ区間</Annotation> <State>1</State> </TransitionArea> <Crossing> <Annotation>車道交差部</Annotation> <State>1</State> </Crossing> <Roadway> <Annotation>その他の車道部</Annotation> <State>1</State> </Roadway> </GroupOfRoadway> </Response>
解　説	各地物クラスの表示状態を取得します。 引数には、取得したい地物クラスを示すXpathを記述します。 たとえば。すべての地物クラスのリストがほしい場合には、ルートノードを指定します。 クライアントには、Response要素を返し、その内容は指定の地物カタログの要素となります。

4.3.2. 地物クラスの表示状態の設定(*SetFeatureClassState*)

API	*SetFeatureClassState*
機　能	指定の地物クラスの表示状態を設定します
引数の形式	text/xml
引数の内容	#Xpath=/GroupOfRoadFundamentalFeature/GroupOfRoadBasicFeature/ GroupOfBasicSurface/GroupOfCrossSection/GroupOfRoadway #とした時の要求例 <Request> <GroupOfRoadway> <Lane> <State>1</State> </Lane> <TransitionArea> <State>1</State> </TransitionArea> <Crossing> <State>1</State> </Crossing> <Roadway> <State>1</State> </Roadway> </GroupOfRoadway> <Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> <ReturnFlag>*(成功・失敗を返すフラグ：True/False)*</ReturnFlag> </Response>
解　説	事前にGetFeatureClassStateによって地物カタログの内容を取得します。 地物クラスの表示状態を変更したい場合には、取得した地物カタログのSTATE要素を変更します。 クライアントには、処理の成否を返り値フラグとして返します。 変更した地物カタログをSetFeatureClassStateリクエストで返送することで、 各地物クラスの表示状態が更新されます。

4.4. データ範囲の取得 API

4.4.1. データ範囲の取得(*GetExtent*)

API	*GetExtent*
機　能	システムで扱うことのできるデータ範囲を取得します
引数の形式	text/xml
引数の内容	<Request> 　<Location>（*地域名*）</Location> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<NorthWest> 　　< DirectPosition> 　　<Coordinate>*(データ中で最北の緯度)*</Coordinate> 　　<Coordinate>*(データ中で最西の経度)*</Coordinate> 　　</DirectPosition> 　</NorthWest> 　<SouthEast> 　　<DirectPosition> 　　<Coordinate>*(データ中で最南の緯度)*</Coordinate> 　　<Coordinate>*(データ中で最東の経度)*</Coordinate> 　　</DirectPosition> 　</SouthEast> </Response>
解　説	システムで扱うことのできるデータ範囲を取得します クライアントには北西と南東の端を示す座標を返します。 具体的には、全域に拡大する際や、索引図の表示範囲を取得したい際に利用します。

4.5. 描画法情報の設定・取得処理 API

4.5.1. 地物の描画法情報の取得(*GetFeatureClassPortraiture*)

API	*GetFeatureClassPortraiture*
機　能	特定地物の描画法情報を取得する
引数の形式	text/xml
引数の内容	<Request> 　<Path>*(描画法情報の取得を要求する地物クラス（グループ）へのXpath)*</Path> </Request>
レスポンスの種類	text/xml
レスポンスの内容	#Xpath=/GroupOfRoadFundamentalFeature/GroupOfRoadBasicFeature/ GroupOfBasicSurface/GroupOfCrossSection/GroupOfRoadwayLane #とした時のレスポンス例 <Response> <Lane> <Annotation>車線</Annotation> <Portaiture> <LineColor>#c40000 </LineColor> <LineWidth>1</LineWidth> <LinePattern>SOLID</LinePattern> <BrushFGColor>#82ff7f </BrushFGColor> <BrushBGColor>#ffffff</BrushBGColor> <BrushPattern>SOLID </BrushPattern> <SymbolName>Star</SymbolName> <MaxScale>2500</MaxScale> <MinScale>0</MinScale> </Portaiture> </Lane> </Response>
解　説	各地物クラスの描画法の情報を取得します。 引数には、取得したい地物クラスを示すXpathを記述します。 たとえば。すべての地物クラスの描画法のリストがほしい場合には、ルートノードを指定します。 クライアントには、指定地物クラス毎の描画法を示す要素を返します。

Portaitureタグ内の内容

LineColor: 線の色。
“#” +r(0～255の16進表記) +g(同左) +b(同左) として表す

LineWidth: 線の太さ。ピクセル単位。0の時は太さなしの線とする。

LinePattern: 線の種類。次の種類から選択する。

SOLID	実線
DASH	破線
DOT	点線
DASHDOT	破線
DASHDOT2	破線

BrushFGColor: 塗りつぶしの表示色

BrushBGColor: 塗りつぶしの背景色

BrushPattern: 塗りつぶしのパターン

SOLID	表示色で塗りつぶし
EMPTY	塗りつぶしなし
FDIAGONAL	“／”方向の斜線
BDIAGONAL	“＼”方向の斜線
HORIZONTAL	横線
VERTICAL	縦線
CROSS	“＋”のハッチ
DIAGONALCROSS	“×”のハッチ

SymbolName: 点図形に配置するシンボル名
シンボル用のpng形式ファイルを整備する必要有

MaxScale: 最大表示スケール

MinScale: 最小表示スケール

4.5.2. 地物の描画法情報の設定(*SetFeatureClassPortraiture*)

API	*SetFeatureClassPortraiture*
機　能	特定地物の描画法情報を設定します
引数の形式	text/xml
引数の内容	#Xpath=/GroupOfRoadFundamentalFeature/GroupOfRoadBasicFeature/ GroupOfBasicSurface/GroupOfCrossSection/GroupOfRoadwayLane #とした時の要求例 <Request> <Lane> <Portaiture> <LineColor>#c40000 </LineColor> <LineWidth>1</LineWidth> <LinePattern>SOLID</LinePattern> <BrushFGColor>#82ff7f </BrushFGColor> <BrushBGColor>#ffffff</BrushBGColor> <BrushPattern>SOLID </BrushPattern> <SymbolName>Star</SymbolName> <MaxScale>2500</MaxScale> <MinScale>0</MinScale> </Portaiture> </Lane> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> <ReturnFlag>*(成功・失敗を返すフラグ：True/False)*</ReturnFlag> </Response>
解　説	事前にGetFeatureClassPortraitureによって各地物の描画法の情報を取得します。 地物クラスの表示状態を変更したい場合には、取得した地物カタログのPortraiture要素を変更します。 クライアントには、処理の成否を示すフラグを返します。 変更した地物カタログをSetFeaturePortraitureリクエストで返送することで、 各地物クラスの描画法情報が更新されます。

4.6. 時点の設定・取得処理 API

4.6.1. 時点の取得(*GetTimePoint*)

API	*GetTimePoint*
機　能	表示時点の情報を取得する
引数の形式	なし
引数の内容	なし
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<Century>20</Century> 　<Year>02</Year> 　<Month>01</Month> 　<Day>01</Day> </Response>
解　説	表示時点の情報を取得します。 クライアントには、現在の表示時点を示す時点データを返します。 複数回この問合せを行う場合､キャ２ッシュを無効化するために必要に応じて引数にダミーパラメータを使用する。

4.6.2. 時点の設定(*SetTimePoint*)

API	*SetTimePoint*
機　能	表示時点の情報を設定する
引数の形式	text/xml
引数の内容	<Request> <Century>20</Century> <Year>02</Year> <Month>01</Month> <Day>01</Day> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> <ReturnFlag>*(成功・失敗を返すフラグ：True/False)*</ReturnFlag> </Response>
解　説	表示時点の設定をサーバに要求します。 クライアントには、処理の成否を示すフラグを返します

4.7. 位置検索処理 API

4.7.1. 住所検索情報の取得(*GetPlaceNameCatalog*)

API	*GetPlaceNameCatalog*
機　能	住所検索に使うパスの情報名を取得する
引数の形式	text/xml
引数の内容	<Request> 　<Location>*（地域名）*</Location> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<PlaceNameCatalog> 　　<Name>Country</Name>　　←この階層のクラス名 　　<Annotation>国名</Annotation>←その階層の名前 　　<Enable>False</Enable>　　←「確定」で位置指定できるか？ 　　<Visible>False</Visible>　　←UIに表示するか？ 　　<Order>False</Order>　　←フリガナによる分類を行うか？ 　</PlaceNameCatalog> 　<PlaceNameCatalog> 　　<Name>Domain</Name> 　　<Annotation>県名</Annotation> 　　<Enable>True</Enable> 　　<Visible>True</Visible> 　　<Order>False</Order> 　</PlaceNameCatalog> 　<PlaceNameCatalog> 　　<Name>City</Name> 　　<Annotation>市名</Annotation> 　　<Enable>True</Enable> 　　<Visible>True</Visible> 　　<Order>False</Order> 　</PlaceNameCatalog> 　<PlaceNameCatalog> 　　<Name>Major</Name> 　　<Annotation>大字・町名</Annotation> 　　<Enable>True</Enable> 　　<Visible>True</Visible> 　　<Order>True</Order> 　</PlaceNameCatalog> </Response>
解　説	住所検索に用いる情報を取得します。例は次のような情報を示しています。

1) 住所検索の第１レベルのタグ名は<Country>,表記は「国名」、位置の指定はできない、フリガナによる分類は行わない。確定事項なので画面表示はしない。
2) 住所検索の第２レベルのタグ名は<Domain>,表記は「県名」、位置の指定が可能、フリガナによる分類は行わない。画面表示を行う。
3) 住所検索の第３レベルのタグ名は<City>,表記は「市名」、位置の指定が可能、フリガナによる分類は行わない。画面表示を行う。
4) 住所検索の第４レベルのタグ名は<Major>,表記は「大字・町名」、表示位置の検索が可能、フリガナによる分類を行う。画面表示を行う。

これらの情報に従い、サーバに住所検索のための情報の取得を要求します。
要素の内容の要求・取得は次に示す、GetPlaceNameSearchInfoを用います。

4.7.2. 住所の位置情報の取得(*GetPlaceNameSearchInfo*)

API	*GetPlaceNameSearchInfo*
機　能	住所情報を取得します
引数の形式	text/xml
引数の内容	＃例１ <Request> <SearchTag> <Path>/Country</Path> </SearchTag> <Location>*（地域名）*</Location> </Request> ＃例２ <Request> <SearchTag> <Path>/Country/Domain/City/Major</Path> </SearchTag> <Restrict> <Path>/Country</Path> <NAME>日本</NAME>　　/Country/Nameは日本 </Restrict> <Restrict> <Path>/Country/Domain</Path> <NAME>○○県</NAME>　/Country/Domain/Nameは○○県 </Restrict> <Restrict> <Path>/Country/Domain/City</Path> <NAME>○○市</NAME>　/Countyr/Domain/City/Nameは○○市 </Restrict> <Location>*（地域名）*</Location> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response>　　＃例１のレスポンス <Country> <NAME>日本</NAME> <Annotation>にほん</Annotation> <SouthEast> <DirectPosition> <Coordinate>35.4565029313694</Coordinate> <Coordinate>137.866512436155</Coordinate> </DirectPosition> </SouthEast> <NorthWest>

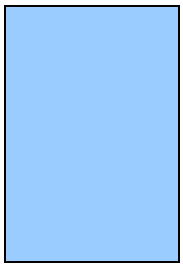
```
            <DirectPosition>
            <Coordinate>38.9526914904446</Coordinate>
              <Coordinate>135.861045221242</Coordinate>
            </DirectPosition>
          </NorthWest>
    </Country>
</Response>

<Response>   ＃例２のレスポンス
    <Major>
          <NAME>○○</NAME>
          <Annotation>○○○○</Annotation>
          <SouthEast>
            <DirectPosition>
              <Coordinate>37.4565029313694</Coordinate>
              <Coordinate>138.866512436155</Coordinate>
            </DirectPosition>
          </SouthEast>
          <NorthWest>
                <DirectPosition>
              <Coordinate>37.4526914904446</Coordinate>
              <Coordinate>138.861045221242</Coordinate>
            </DirectPosition>
          </NorthWest>
        </Major>
        ///省略
        <Major>
          <NAME>△△△</NAME>
          <Annotation>△△△△△△</Annotation>
          <SouthEast>
                <DirectPosition>
              <Coordinate>37.4895441639212</Coordinate>
              <Coordinate>138.846067813875</Coordinate>
            </DirectPosition>
          </SouthEast>
          <NorthWest>
            <DirectPosition>
              <Coordinate>37.4763168210009</Coordinate>
              <Coordinate>138.826750257868</Coordinate>
            </DirectPosition>
          </NorthWest>
        </Major>
</Response>
```

解　説

住所検索の情報を取得します。リクエストの内容（取得したいクラス名、その際の絞込み条件）に対応するクラスの、NAME要素、Annotation要素、SouthEast要素、NorthEast要素を返します。
レスポンス例は次のような情報を示しています。

1) まず、Country要素を、制約条件なしし取得します。（リクエスト例1）
2) クライアント側では、ここで取得された値をユーザインタフェースに登録し、ユーザに選択させます（ここでは１要素しかないので、選択の余地はない）
3) Country要素がユーザにより確定されたら、その要素の下にあるDomain要素を要



求します。
4) 同様にユーザインタフェースにより、/Country/Domainの要素が確定します
5) 同様にユーザインタフェースにより、/Country/Domain/Cityの要素を確定します。
6) City要素が確定したら、Major要素を要求します（リクエスト例２）
7) Major要素を取得します（レスポンス例2）

このような問い合わせを行うことで､各レベル毎の登録要素と登録要素毎の表示範囲を取得し、その表示範囲で地図画像を要求することで、位置検索を実現します

4.7.3. 目標物検索情報の取得(*GetLandMarkCatalog*)

API	*GetLandMarkCatalog*
機　能	目標物検索に使うパスの情報名を取得する
引数の形式	text/xml
引数の内容	<Request> 　<Location>*（地域名）*</Location> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<LandMarkCatalog> 　　<Name>Country</Name>　←この階層のクラス名 　　<Annotation>国名</Annotation>　←その階層の名前 　　<Enable>False</Enable>　←「確定」で位置指定できるか？ 　　<Visible>False</Visible>　←画面に表示するか？ 　　<Order>False</Order>　←フリガナによる分類を行うｶ 　</LandMarkCatalog> 　<LandMarkCatalog> 　　<Name>Domain</Name> 　　<Annotation>県名</Annotation> 　　<Enable>False</Enable> 　　<Visible>True</Visible> 　　<Order>False</Order> 　</LandMarkCatalog> 　<LandMarkCatalog> 　　<Name>City</Name> 　　<Annotation>市名</Annotation> 　　<Enable>False</Enable> 　　<Visible>True</Visible> 　　<Order>False</Order> 　</LandMarkCatalog> 　<LandMarkCatalog> 　　<Name>Category</Name> 　　<Annotation>分類</Annotation> 　　<Enable>False</Enable> 　　<Visible>True</Visible> 　　<Order>True</Order> 　</LandMarkCatalog> 　<LandMarkCatalog> 　　<Name>LandMark</Name> 　　<Annotation>名称</Annotation> 　　<Enable>True</Enable> 　　<Visible>True</Visible>

<Order>False</Order>
</LandMarkCatalog>
</Response>

解　説

目標物検索に用いる情報を取得します。レスポンス例は次のような情報を示しています。

5) 目標物検索の第１レベルのタグ名は<Country>,表記は「国名」、位置の指定はできない、フリガナによる分類は行わない。確定事項なので、画面表示は行わない。
6) 目標物検索の第２レベルのタグ名は<Domain>,表記は「県名」、位置の指定はできない、フリガナによる分類は行わない。画面表示を行う。
7) 目標物検索の第３レベルのタグ名は<City>,表記は「市名」、位置の指定はできない、フリガナによる分類は行わない。画面表示を行う。
8) 目標物検索の第４レベルのタグ名は<Category>,表記は「分類」、表示位置の検索はできない、フリガナによる分類を行う。画面表示を行う。
9) 目標物検索の第５レベルのタグ名は<LandMark>,表記は「名称」、表示位置の検索が可能、フリガナによる分類は行わない。画面表示を行う。

これらの情報に従い、サーバに目標物検索のための情報の取得を要求します。
要素の内容の要求・取得は次に示す、GetLandMarkCatalogを用います。

4.7.4. 目標物の位置情報の取得(*GetLandMarkSearchInfo*)

API	*GetLandMarkSearchInfo*
機　能	目標物の位置情報を取得する
引数の形式	text/xml
引数の内容	＃例 <Request> <SearchTag> <Path>/Country/Domain/City/Category/LandMark<Path> </SearchTag> <Restrict> <Path>/Country</Path> <NAME>日本</NAME> /Country/Nameは日本 </Restrict> <Restrict> <Path>/Country/Domain</Path> <NAME>○○県</NAME> /Country/Domain/Nameは○○県 </Restrict> <Restrict> <Path>/Country/Domain/City</Path> <NAME>○○市</NAME> /Countyr/Domain/City/Nameは○○市 </Restrict> <Restrict> <Path>/Country/Domain/City/Category</Path> <NAME>大学</NAME> /Countyr/Domain/City/Category/Nameは大学 </Restrict> <Location>*（地域名）*</Location> </Request>
レスポンスの種類	text/xml
レスポンスの内容	＃例１のレスポンス <Response> <LandMark> <Name>○○大学</Name> <SouthEast> <DirectPosition> <Coordinate>37.4224058094633</Coordinate> <Coordinate>138.780900554458</Coordinate> </DirectPosition> </SouthEast> <NorthWest> <DirectPosition> <Coordinate>37.4224058094633</Coordinate> <Coordinate>138.780900554458</Coordinate> </DirectPosition>

```
        </NorthWest>
    </LandMark>
    <LandMark>
        <Name>△△大学</Name>
        <SouthEast>
            <DirectPosition>
            <Coordinate>37.4635597445925</Coordinate>
            <Coordinate>138.834293986719</Coordinate>
            </DirectPosition>
        </SouthEast>
        <NorthWest>
            <DirectPosition>
            <Coordinate>37.4635597445925</Coordinate>
              <Coordinate>138.834293986719</Coordinate>
            </DirectPosition>
        </NorthWest>
    </LandMark>
</Response>
```

解　説

目標物情報を取得します。リクエストの内容（取得したいクラス名、その際の絞込み条件）に対応するクラスの、NAME要素、Annotation要素、SouthEast要素、NorthEast要素を返します。
レスポンス例は次のような情報を示しています。

8) まず、Country要素を、制約条件なしし取得します。
9) クライアント側では、ここで取得された値をユーザインタフェースに登録し、ユーザに選択させます（ここでは１要素しかないので、選択の余地はない）
10) Country要素がユーザにより確定されたら、その要素の下にあるDomain要素を要求します。
11) 同様にユーザインタフェースにより、/Country/Domain、/Country/Domain/City、/Country/Domain/City/Categoryの要素が確定します
12) Category要素が確定したら、LandMark要素を要求します(リクエスト例)
13) Major要素を取得します（レスポンス例2）

このような問い合わせを行うことで､各レベル毎の登録要素と登録要素毎の表示範囲を取得し、その表示範囲で地図画像を要求することで、位置検索を実現します

4.7.5. キロポスト検索情報の取得(*GetSpotMarkCatalog*)

API	*GetSpotMarkCatalog*
機　能	キロポスト検索に使うパスの情報名を取得する
引数の形式	text/xml
引数の内容	<Request> 　<Location>*（地域名）*</Location> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> <SpotMarkCatalog> <Name>Country</Name>　←この階層のクラス名 <Annotation>国名</Annotation>　←その階層の表記名 <Enable>False</Enable>　←「確定」で位置指定できるか？ <Visible>False</Visible>　←画面に表示するか？ <Order>False</Order>　←フリガナによる分類を行うｶ </SpotMarkCatalog> <SpotMarkCatalog> <Name>Domain</Name> <Annotation>県名</Annotation> <Enable>False</Enable> <Visible>True</Visible> <Order>False</Order> </SpotMarkCatalog> <SpotMarkCatalog> <Name>RoadType</Name> <Annotation>道路種別</Annotation> <Enable>False</Enable> <Visible>True</Visible> <Order>False</Order> </SpotMarkCatalog> <SpotMarkCatalog> <Name>Route</Name> <Annotation>路線名称</Annotation> <Enable>False</Enable> <Visible>True</Visible> <Order>False</Order> </SpotMarkCatalog> <SpotMarkCatalog> <Name>Section</Name> <Annotation>区間</Annotation> <Enable>True</Enable> <Visible>True</Visible>

```
            <Order>False</Order>
        </SpotMarkCatalog>
        <SpotMarkCatalog>
            <Name>SpotMark</Name>
            <Annotation>距離標</Annotation>
            <Enable>True</Enable>
            <Visible>True</Visible>
            <Order>False</Order>
        </SpotMarkCatalog>
          <Location>地域名</Location>
</Response>
```

解　説

キロポスト検索に用いる情報を取得します。レスポンス例は次のような情報を示しています。

10) キロポスト検索の第１レベルのタグ名は<Country>,表記は「国名」、位置の指定はできない、フリガナによる分類は行わない。確定事項なので、画面表示は行わない。
11) キロポスト検索の第２レベルのタグ名は<Domain>,表記は「県名」、位置の指定はできない、フリガナによる分類は行わない。画面表示を行う。
12) キロポスト検索の第３レベルのタグ名は<RoadType>,表記は「道路種別」、位置の指定はできない、フリガナによる分類は行わない。画面表示を行う。
13) キロポスト検索の第４レベルのタグ名は<Route>,表記は「路線名」、表示位置の検索はできない、フリガナによる分類を行う。画面表示を行う。
14) キロポスト検索の第５レベルのタグ名は<Section>,表記は「区間」、表示位置の検索が可能、フリガナによる分類を行う。画面表示を行う。
15) キロポスト検索の第６レベルのタグ名は<SpotMark>,表記は「距離標」、表示位置の検索が可能、フリガナによる分類は行わない。画面表示を行う。

これらの情報に従い、サーバに目標物検索のための情報の取得を要求します。
要素の内容の要求・取得は次に示す、GetSpotMarkSearchInfoを用います。

4.7.6. キロポストの位置情報の取得(*GetSpotMarkSearchInfo*)

API	*GetSpotMarkSearchInfo*
機　能	キロポストの位置情報を取得する
引数の形式	text/xml
引数の内容	<Request> <SearchTag> <Path>/Country/Domain/RoadType/Route/Section/SpotMark <Path> </SearchTag> <Restrict> <Path>/Country</Path> <NAME>日本</NAME>　Country要素の名前は日本 </Restrict> <Restrict> <Path>/Country/Domain</Path> <NAME>○○県</NAME>　Domain要素の名前は○○県 </Restrict> <Restrict> <Path>/Country/Domain/RoadType</Path> <NAME>一般国道</NAME>　RoadType要素の名前は長岡市 </Restrict> <Restrict> <Path>/Country/Domain/RoadType/Route</Path> <NAME>国道○号線</NAME>　Route要素の名前は国道○号線 </Restrict> <Restrict> <Path>/Country/Domain/RoadType/Route/Section</Path> <NAME>21.0K～22.0K </NAME>　Section要素の名前は21.0K～22.0K </Restrict> <Location>地域名</Location> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> <SpotMark> <NAME>21.0K</NAME> <Annotation> </Annotation> <SouthEast> <DirectPosition> <Coordinate>37.4565029313694</Coordinate> <Coordinate>138.866512436155</Coordinate> </DirectPosition> </SouthEast> <NorthWest> <DirectPosition>

```
            <Coordinate>37.4526914904446</Coordinate>
            <Coordinate>138.861045221242</Coordinate>
          </DirectPosition>
        </NorthWest>
      </SpotMark>
      ///省略
      <SpotMark>
        <NAME>21.9K </NAME>
        <Annotation></Annotation>
        <SouthEast>
          <DirectPosition>
          <Coordinate>37.4895441639212</Coordinate>
          <Coordinate>138.846067813875</Coordinate>
          </DirectPosition>
        </SouthEast>
        <NorthWest>
          <DirectPosition>
            <Coordinate>37.4763168210009</Coordinate>
            <Coordinate>138.826750257868</Coordinate>
          </DirectPosition>
        </NorthWest>
      </SpotMark>
</Response>
```

解　説

キロポスト情報を取得します。リクエストの内容（取得したいクラス名、その際の絞込み条件）に対応するクラスの、NAME要素、Annotation要素、SouthEast要素、NorthEast要素を返します。
レスポンス例は次のような情報を示しています。

1) まず、Country要素を、制約条件なしし取得します。
2) クライアント側では、ここで取得された値をユーザインタフェースに登録し、ユーザに選択させます（ここでは１要素しかないので、選択の余地はない）
3) Country要素がユーザにより確定されたら、その要素の下にあるDomain要素を要求します。
4) 同様にユーザインタフェースにより、/Country/Domainの要素が確定します
5) 同様にRoadTypeの要素、Route要素、Section要素を要求します
6) SpotMark要素を要求し（リクエスト例）、範囲のデータ取得します（レスポンス例）

このような問い合わせを行うことで､各レベル毎の登録要素と登録要素毎の表示範囲を取得し、その表示範囲で地図画像を要求することで、位置検索を実現します

4.8. 地物定義の取得 API

4.8.1. 地物一覧の取得 (GetFeatureClassList)

API	GetFeatureClassList
機　能	地物一覧を取得する
引数の形式	Text/xml
引数の内容	なし
レスポンスの種類	Text/xml
レスポンスの内容	<Response> <FeatureClass> <Schema_Name>（スキーマ名）</Schema_Name> <Package_Name>（パッケージ名）</Package_Name> <Class_Name>（クラス名）</Class_Name> <Class_JPName>（クラス名）</Class_JPName> </FeatureClass> <FeatureClass> <Schema_Name>（スキーマ名）</Schema_Name> <Package_Name>（パッケージ名）</Package_Name> <Class_Name>（クラス名）</Class_Name> <Class_JPName>（クラス名）</Class_JPName> </FeatureClass> <ReturnFlag>処理結果(注1)</ReturnFlag> <Status>ステータス<Status> </Response>
解　説	注1 処理結果は以下のとおり 正常終了:True 異常終了:False <Status>タグ部は将来の拡張用

4.8.2. 地物の詳細情報の取得 (GetFeatureClassInfo)

API

GetFeatureClassInfo

機　能

クラス名を指定して地物の属性詳細情報を取得する。

引数の形式

Text/xml

引数の内容

```
<Request>
    <FeatureClass>
        <Schema_Name>（スキーマ名）</Schema_Name>
        <Package_Name>（パッケージ名）</Package_Name>
        <Class_Name>（クラス名）</Class_Name>
    </FeatureClass>

                        :
                        :       数分繰り返す
                        :
    <FeatureClass>
        <Schema_Name>（スキーマ名）</Schema_Name>
        <Package_Name>（パッケージ名）</Package_Name>
        <Class_Name>（クラス名）</Class_Name>
    </FeatureClass>

</Request>
```

レスポンスの種類

Text/xml

レスポンスの内容

```
<Response>
      <FeatureClassInfo>
         <Schema_Name>（スキーマ名）</Schema_Name>
        <Package_Name>（パッケージ名）</Package_Name>
        <Class_Name>（クラス名）</Class_Name>
            <Class_JPName>（クラス名（日本語名称））</Class_JPName>
            <Shape>（点線面種別(注2)）</Shape>
        <SymbolImage>（None Or シンボルファイル名）</SymbolImage>
            <Attribute>
                  <Attribute_Name>（属性名）</Attribute_Name>
                  <Attribute_JPName>（属性名称（日本語名））</Attribute_JPName>
              <Attribute_DataType>（属性値の型(注3)）</Attribute_ DataType>
                   :
                   :
              <Attribute_Name>（属性名）</Attribute_Name>
              <Attribute_JPName>（属性名（日本語））</Attribute_JPName>
              <Attribute_DataType>Code(注3)</Attribute_DataType>
              <CodeClass>
                    <Code_Schema_Name>（スキーマ名）</Code_Schema_Name>
                    <Code_Package_Name>（パッケージ名）</Code_Package_Name >
```

```
                    <Code_Class_Name>（クラス名）</Code_Class_Name>
                <CodeClass>
                              :
                              :      属性数分繰り返す
                              :

                     <Attribute_Name>（属性名）</Attribute_Name>
                     <Attribute_JPName>（属性名称（日本語名））</Attribute_JPName>
                   <Attribute_DataType>（属性値の型(注3)）</Attribute_ DataType>
               </Attribute>
          </FeatureClassInfo>
      <ReturnFlag>（処理結果(注2) ）</ReturnFlag>
      <Status>ステータス<Status>
</Response>
```

解　説

注1
　点線面種別は0：なし　1：点　2：線　3：面

注2
　処理結果は以下のとおり
　正常終了:True
　異常終了:False

注3
　データ型の定義
　　String
　　Integer
　　Double
　　URL(HTTPを想定。同一サーバ内のURLを指定する場合、BaseURLを利用し、
　　そこからの相対パスでURLを指定する。他のサイトのURLを指定する場合、
　　http://から記述する。
　Date(日付。フォーマットは固定長でYYYYMMDDの8桁。ゼロパディング)
　Time(時間。フォーマットは固定長でHHMISSの6桁。ゼロパディング。24h表記)
　DayTime（日付と時間で、YYYYMMDDHHMISSの固定長14桁。ゼロパディング）

　<Status>タグ部は将来の拡張用

4.8.3. コードリストの取得 (GetFeatureClassList)

API	GetCodeList
機　能	クラス名を指定してコードリストを取得する。
引数の形式	text/xml
引数の内容	<Request> <CodeClass> <Code_Schema_Name>（スキーマ名）</Code_Schema_Name> <Code_Package_Name>（パッケージ名）</Code_Package_Name> <Code_Class_Name>（クラス名）</Code_Class_Name> <CodeClass> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> <Code_Schema_Name>（スキーマ名）</Code_Schema_Name> <Code_Package_Name>（パッケージ名）</Code_Package_Name> <Code_Class_Name>（クラス名）</Code_Class_Name> <CodeList> <Code>（コード ）</Code> <Code_Value>（コードに対応する値）</Code_Value> </CodeList> <CodeList> <Code>（コード）</Code>　例　01 <Code_Value>（コードに対応する値）</Code_Value>　例　東京 </CodeList> ・ ・ ・ <CodeList> <Code>（コード）</Code> <Code_Value>（コードに対応する値）</Code_Value> </CodeList> <ReturnFlag>（処理結果(注1)</ReturnFlag> <Status>（ステータス(注2)</Status> </Response>
解　説	注1 処理結果は以下のとおり 正常終了:True 異常終了:False 注2 <Status>タグ部は将来の拡張用

4.9. 属性情報の取得 API

4.9.1. 一点を指定して検索 (*ScanPoint)*

API	*ScanPoint*
機　能	特定地点を含む地物を列挙し、その属性情報を取得する
引数の形式	text/xml
引数の内容	<Request> <NorthWest> < DirectPosition> <Coordinate>(現在の表示範囲で最北の緯度)</Coordinate> <Coordinate>(現在の表示範囲で最西の経度)</Coordinate> </DirectPosition> </NorthWest> <SouthEast> <DirectPosition> <Coordinate>(現在の表示範囲で最南の緯度)</Coordinate> <Coordinate>(現在の表示範囲で最東の経度)</Coordinate> </DirectPosition> </SouthEast> <XSize >(現在表示している地図画像のXサイズ（ピクセル単位の整数値）</XSize> <YSize >(現在表示している地図画像のYサイズ（ピクセル単位の整数値）</YSize> <ScanPoint> <DirectPosition> <Coordinate>指定地点の緯度</Coordinate> <Coordinate>指定地点の経度</Coordinate> </DirectPosition> </ScanPoint> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> <FeatureInfo> <Feature>*(地物名を示す英数字)* </Feature> <FeatureName>*(地物名を示す日本語名称)*</FeatureName> <FeatureID>地物ID</FeatureID> <Attribute> <Annotation>取得された属性の表記名1</Annotation> <Value>内容1</Value> <Annotation>取得された属性の表記名2</Annotation> <Value>内容2</Value> ･･･取得された属性数分繰り返す </Attribute > </FeatureInfo>

・・・取得された地物分繰り返す
</Response>

解　説

特定地点を含む属性情報を取得します。
レスポンスの内容は、地物毎の属性リストになります。
地物クラスのカタログは別紙を参照してください。

4.9.2. 一点を指定して検索 (*GetFeatureByScanPoint)*

API	GetFeatureByScanPoint（現在未実装）
機　能	指定した地点の地物検索処理
引数の形式	Text/xml
引数の内容	<Request> <NorthWest> < DirectPosition> <Coordinate>(現在の表示範囲で最北の緯度)</Coordinate> <Coordinate>(現在の表示範囲で最西の経度)</Coordinate> </DirectPosition> </NorthWest> <SouthEast> <DirectPosition> <Coordinate>(現在の表示範囲で最南の緯度)</Coordinate> <Coordinate>(現在の表示範囲で最東の経度)</Coordinate> </DirectPosition> </SouthEast> <XSize >(現在表示している地図画像のXサイズ（ピクセル単位の整数値）</XSize> <YSize >(現在表示している地図画像のYサイズ（ピクセル単位の整数値）</YSize> <ScanPoint> <DirectPosition> <Coordinate>指定地点の緯度</Coordinate> <Coordinate>指定地点の経度</Coordinate> </DirectPosition> </ScanPoint> <FeatureClass> <Schema_Name>（スキーマ名）</Schema_Name> <Package_Name>（パッケージ名）</Package_Name> <Class_Name>（クラス名）</Class_Name> </FeatureClass> : <FeatureClass> <Schema_Name>（スキーマ名）</Schema_Name> <Package_Name>（パッケージ名）</Package_Name> <Class_Name>（クラス名）</Class_Name> </FeatureClass> </Reques
レスポンスの種類	text/xml
レスポンスの内容	<Response> <Feature> <Schema_Name>（スキーマ名）</Schema_Name> <Package_Name>（パッケージ名）</Package_Name> <Class_Name>（クラス名）</Class_Name>

```
            <Class_JPName>（クラス名（日本語名称）</Class_JPName>
            <FeatureID>（ID）<FeatureID>
            <Shape>（点線面種別(注2) ）</Shape>
        <SymbolImage>（None Or シンボルファイル名）</SymbolImage>
        <Angle>（角度）</Angle>

            <Coordinate>（座標データ(注3) ）</Coordinate>
            <Attribute>
                  <Attribute_Name>（属性名）</Attribute_Name>
                  <Attribute_JPName>（属性名称（日本語名）</Attribute_JPName>
            <Attribute_DataType>（属性値の型）</Attribute_ DataType>
                  <Attribute_Value>（属性値）</Attribute_Value>
                        ・
                        ・
                        ・
            <Attribute_Name>（属性名）</Attribute_Name>
            <Attribute_JPName>（属性名（日本語））</Attribute_JPName>
            <Attribute_DataType>Code</Attribute_DataType>
            <Attribute_Value>（コード）</Attribute_Value>  例　01
            <Attribute_Code>（コードの値）</Attribute_ Code>  例　東京
            <CodeClass>
              <Code_Schema_Name>（スキーマ名）</ Code_Schema_Name >
              <Code_Package_Name>（パッケージ名）</Code_Package_Name >
              <Code_Class_Name>（クラス名）</Code_Class_Name>
            <CodeClass>
                        :
                        :      属性数分繰り返す
                        :

                  <Attribute_Name>（属性名）</Attribute_Name>
                  <Attribute_JPName>（属性名称（日本語名））</Attribute_JPName>
            <Attribute_DataType>（属性値の型）</Attribute_ DataType>
                  <Attribute_Value>（属性値）</Attribute_Value>
             </Attribute>
      </Feature>
          ・
          ・  検索結果のインスタンス分だけ繰り返す.
          ・
      <Feature>
              .
              .
              .
      </Feature>

      <ReturnFlag>（処理結果(注1) ）</ReturnFlag>
    <Status>ステータス</Status>

</Response>
```

解　説

リクエストで指定する<FeatureClass>部は任意の指定で、GetFeatureClassList　API

で取得したスキーマ、パッケージ、クラス名をそれぞれ対応する子ノードに設定し、使用する。<FeatureClass>タグが存在しない場合、指定地点を含む画面上に表示されているすべて地物を結果として返す。<FeatureClass>タグが存在する場合、<FeatureClass>で指定した地物のみ検索対象とする。ただし、画面上に表示されていない地物は検索対象外となる。

注1
処理結果は以下のとおり
正常終了:True
異常終了:False
注2
点線面種別は0：なし　1：点　2：線　3：面
注3
座標データの形式は以下のとおり
点:“経度,緯度”
線:“経度,緯度,経度,緯度…”の並びのCSV
面:“<Exterior>外周多角形座標列</Exterior>
<Interior>内周多角形座標列</Interior>..”
座標列は線の形式と同一のCSV形式で、始点と同一座標の終点で終わる。
内周を持たない場合は、<Interior>内周多角形座標列</Interior>を省略する。
内周を複数持つ場合は<Interior>内周多角形座標列</Interior>を繰り返す。

<Status>タグ部は将来の拡張用

4.9.3. ID を指定して検索 (*GetFeatureByID*)

API	GetFeatureByID
機　能	指定したIDの地物の属性を取得する。
引数の形式	text/xml
引数の内容	<Request> <FeatureInfo> <Schema_Name>（スキーマ名）</Schema_Name> <Package_Name>（パッケージ名）</Package_Name> <Class_Name>（クラス名）</Class_Name> <FeatureID>（地物ID）</FeatureID> </FeatureInfo> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> <Feature> <Schema_Name>（スキーマ名）</Schema_Name> <Package_Name>（パッケージ名）</Package_Name> <Class_Name>（クラス名）</Class_Name> <Class_JPName>（クラス名（日本語名称）</Class_JPName> <FeatureID>（ID）<FeatureID> <Shape>（点線面種別(注2)）</Shape> <SymbolImage>（None Or シンボルファイル名）</SymbolImage> <Angle>（角度）</Angle> <Coordinate>（座標データ(注3)）</Coordinate> <Attribute> <Attribute_Name>（属性名）</Attribute_Name> <Attribute_JPName>（属性名称（日本語名））</Attribute_JPName> <Attribute_DataType>（属性値の型）</Attribute_ DataType> <Attribute_Value>（属性値）</Attribute_Value> ・ ・ <Attribute_Name>（属性名）</Attribute_Name> <Attribute_JPName>（属性名（日本語））</Attribute_JPName> <Attribute_DataType>Code</Attribute_DataType> <Attribute_Value>（コード）</Attribute_Value>　例　01 <Attribute_Code>（コードの値）</Attribute_Code>　例　東京 <CodeClass> <Code_Schema_Name>（スキーマ名）</Code_Schema_Name> <Code_Package_Name>（パッケージ名）</Code_Package_Name> <Code_Class_Name>（クラス名）</ Code_Class_Name> </CodeClass>

```
                                :
                                :       属性数分繰り返す
                                :

                <Attribute_Name>（属性名）</Attribute_Name>
                <Attribute_JPName>（属性名称（日本語名））</Attribute_JPName>
               <Attribute_DataType>（属性値の型）</Attribute_ DataType>
                <Attribute_Value>（属性値）</Attribute_Value>
                </Attribute>
        </Feature>

        <ReturnFlag>（処理結果(注1) ）</ReturnFlag>
     <Status>（ステータス(注2) ）</Status>
</Response>
```

解　説

リクエストで指定する< FeatureInfo >部は、GetFeatureClassList APIで取得したスキーマ、パッケージ、クラス名をそれぞれ対応する子ノードに設定する。

注1
処理結果は以下のとおり
正常終了:True
異常終了:False

<Status>タグ部は将来の拡張用

4.10. メモ機能の API

4.10.1. メモの作成(*CreateMemo)*

API	*CreateMemo*
機　能	指定の位置にメモを作成する
引数の形式	text/xml
引数の内容	<Request> <Memo> <CategoryID>1</CategoryID> <Begin> <Century>20</Century> <Year>02</Year > <Month>03</Month> <Day>01</Day> </Begin> <End> <Century>20</Century> <Year>02</Year > <Month>04</Month> <Day>01</Day> </End> <Description>2002/4/1まで工事中</Description> <Auther>道路一郎</Auther> < DirectPosition> <Coordinate>*(要求する地点の緯度)*</Coordinate> <Coordinate>*(設定する地点の経度)*</Coordinate> </DirectPosition> </Memo> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> <MemoID>*(生成されたメモID)*</MemoID> <ReturnFlag>*(True/False)*</ReturnFlag> </Response>
解　説	指定の地点にメモを生成します。 引数にはメモの分類を示すIDと、有効期限の始まりと終わりを示す要素、表示するメモの内容、作成者名と、メモの配置位置を示す座標を指定します。 メモの配置点の座標は、メモの中央とします クライアントには、処理の成否を示すフラグと、生成されたメモのIDを返します。

4.10.2. メモの取得 (*FindMemo)*

API	*FindMemo*
機　能	特定範囲内に含まれるメモの情報を取得する
引数の形式	text/xml
引数の内容	<Request> 　<NorthWest> 　　<DirectPosition> 　　<Coordinate>*(設定する表示範囲の最北の緯度)*</Coordinate> 　　<Coordinate>*(設定する表示範囲の最西の経度)*</Coordinate> 　　</DirectPosition> 　</NorthWest> 　<SouthEast> 　　<DirectPosition> 　　<Coordinate>*(設定する表示範囲の最南の緯度)*</Coordinate> 　　<Coordinate>*(設定する表示範囲の最東の経度)*</Coordinate> 　　</DirectPosition> 　</SouthEast> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<Memo> 　　<MemoID>200203010001</MemoID> 　　<CategoryID>1</CategoryID> 　　<Begin> 　　　<Century>20</Century> 　　　<Year>02</Year > 　　　<Month>03</Month> 　　　<Day>01</Day> 　　</Begin> 　　<End> 　　　<Century>20</Century> 　　　<Year>02</Year > 　　　<Month>04</Month> 　　　<Day>01</Day> 　　</End> 　　<Description>2002/4/1まで工事中</Description> 　　<Auther>道路一郎</Auther> 　　< DirectPosition> 　　　<Coordinate>*(メモの配置点の緯度)*</Coordinate> 　　　<Coordinate>*(メモの配置点の経度)*</Coordinate> 　　</DirectPosition> 　</Memo> 　･･･取得されたメモの数だけ繰り返す･･･

</Response>

解　説

特定範囲に含まれるメモの内容を取得します。
引数には、検索範囲を示す座標値を設定します。
クライアントには、所定範囲にあったメモの内容を返します。

4.10.3. メモの更新 (*UpdateMemo)*

API	*UpdateMemo*
機　能	指定の位置にメモを作成する
引数の形式	text/xml
引数の内容	<Request> 　<Memo> 　　<MemoID>200203010001</MemoID> 　　<CategoryID>1</CategoryID> 　　<Begin> 　　　<Century>20</Century> 　　　<Year>02</Year > 　　　<Month>03</Month> 　　　<Day>01</Day> 　　</Begin> 　　<End> 　　　<Century>20</Century> 　　　<Year>02</Year > 　　　<Month>04</Month> 　　　<Day>20</Day> 　　</End> 　　<Description>2002/4/20まで工事中</Description> 　　<Auther>道路一郎</Auther> 　　< DirectPosition> 　　　<Coordinate>*(要求する地点の緯度)*</Coordinate> 　　　<Coordinate>*(設定する地点の経度)*</Coordinate> 　　</DirectPosition> 　</Memo> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<MemoID>*(更新されたメモID)*</MemoID> 　<ReturnFlag>*(True/False)*</ReturnFlag> </Response>
解　説	指定のメモの更新を行います。 メモの内容はもとより、位置を移動した際にもこのメソッドを利用します。 引数には修正したメモのIDと、その内容（メモの分類を示すIDと、有効期限の始まりと終わりを示す要素、表示するメモの内容、作成者名と、メモの配置位置を示す座標）を指定します。メモの配置点の座標は、メモの中央とします クライアントには、処理の成否を示すフラグと、更新されたメモのIDを返します。

4.10.4. メモの削除(*RemoveMemo)*

API	*RemoveMemo*
機　能	指定のIDのメモを削除します
引数の形式	text/xml
引数の内容	<Request> 　<Memo> 　　<MemoID>200203010001</MemoID> 　</Memo> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<MemoID>*(削除されたメモID)*</MemoID> 　<ReturnFlag>*(True/False)*</ReturnFlag> </Response>
解　説	指定IDのメモを削除します。 引数には修正したメモのIDを指定します。 クライアントには、処理の成否を示すフラグと、削除されたメモのIDを返します。

4.10.5. メモ分類情報の取得 (*GetMemoCategory)*

API	*GetMemoCategory*
機　能	メモ分類情報の取得
引数の形式	なし
引数の内容	なし
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<Category> 　　<CategoryID>1</CategoryID> 　　<Name>工事</Name> 　　<ForeGroundColor>#FF0000</ForeGroundColor> 　　<BackGroundColor>#FFFFFF</BackGroundColor> 　　<FontSize>12</FontSize> 　　<Bold>True</Bold> 　　<Italic>False</Italic> 　</Category> 　‥‥カテゴリ数分だけ繰り返す‥ </Response>
解　説	登録されているメモ分類情報を取得します クライアントに返される情報は次の通りです。 CategoryID：カテゴリ毎にユニークなID Name：カテゴリの名称 ForeGroundColor:前面の表示色（文字と枠線の色）、# (R) (G) (B)の16進表記 BackGroundColor:背景の色、# (R) (G) (B)の16進表記 FontSize:フォントの大きさ、ポイント指定 Bold:太字にするか Italic:斜体にするか ここで取得された情報を基に、クライアントサイドでのメモの描画を行います。

4.10.6. メモ分類情報の作成(*CreateMemoCategory)*

API	*CreateMemoCategory*
機　能	メモ分類情報の作成
引数の形式	text/xml
引数の内容	<Request> 　<Category> 　　<Name>工事</Name> 　　<ForeGroundColor>#FFFF00</ForeGroundColor> 　　<BackGroundColor>#FFFFFF</BackGroundColor> 　　<FontSize>12</FontSize> 　　<Bold>True</Bold> 　　<Italic>False</Italic> 　</Category> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<CategoryID>*(作成されたカテゴリID)*</CategoryID> 　<ReturnFlag>*(True/False)*</ReturnFlag> </Response>
解　説	メモ分類情報の作成を行います。 引数に指定する情報は次の通りです。 Name：カテゴリの名称 ForeGroundColor:前面の表示色（文字と枠線の色）、# (R) (G) (B)の16進表記 BackGroundColor:背景の色、# (R) (G) (B)の16進表記 FontSize:フォントの大きさ、ポイント指定 Bold:太字にするか Italic:斜体にするか クライアントには、処理の成否を示すフラグと、新規作成したカテゴリIDを返します。

4.10.7. メモ分類情報の更新(*UpdateMemoCategory)*

API	*UpdateMemoCategory*
機　能	メモ分類情報の更新
引数の形式	text/xml
引数の内容	<Request> 　<Category> 　　<CategoryID>1</CategoryID> 　　<Name>工事</Name> 　　<ForeGroundColor>#FFFF00</ForeGroundColor> 　　<BackGroundColor>#FFFFFF</BackGroundColor> 　　<FontSize>12</FontSize> 　　<Bold>True</Bold> 　　<Italic>False</Italic> 　</Category> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<CategoryID>*(更新されたカテゴリID)*</CategoryID> 　<ReturnFlag>*(True/False)*</ReturnFlag> </Response>
解　説	メモ分類情報の更新を行います。 引数に指定する情報は次の通りです。 CategoryID：カテゴリ毎にユニークなID Name：カテゴリの名称 ForeGroundColor:前面の表示色（文字と枠線の色）、# (R) (G) (B)の16進表記 BackGroundColor:背景の色、# (R) (G) (B)の16進表記 FontSize:フォントの大きさ、ポイント指定 Bold:太字にするか Italic:斜体にするか クライアントには、処理の成否を示すフラグと、更新対象となったカテゴリIDを返します。

4.10.8. メモ分類情報の削除 (*RemoveMemoCategory)*

API	*RemoveMemoCategory*
機　能	メモ分類情報の削除
引数の形式	text/xml
引数の内容	<Request> 　<Category> 　　<CategoryID>1</CategoryID> 　</Category> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<CategoryID>*(削除されたカテゴリID)*</CategoryID> 　<ReturnFlag>*(True/False)*</ReturnFlag> </Response>
解　説	メモ分類情報の削除を行います。 引数にはカテゴリのIDを指定します クライアントには、処理の成否を示すフラグと、削除対象となったカテゴリIDを返します。

4.11. 基盤データの取得と更新 API

4.11.1. 基盤データの取得 (*GetRoadData*)

＊バージョンアップ中につき利用できません＊

API	*GetRoadData*
機　能	特定の道路基盤データを取得する
引数の形式	text/xml
引数の内容	<Request> 　<NorthWest> 　　<DirectPosition> 　　<Coordinate>*(取得範囲の最北の緯度)*</Coordinate> 　　<Coordinate>*(取得範囲の最西の経度)*</Coordinate> 　　</DirectPosition> 　</NorthWest> 　<SouthEast> 　　<DirectPosition> 　　<Coordinate>*(取得範囲の最南の緯度)*</Coordinate> 　　<Coordinate>*(取得範囲の最東の経度)*</Coordinate> 　　</DirectPosition> 　</SouthEast> </Request>
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<ReturnFlag>*(True/False)*</ReturnFlag> 　<URL>(URL)<URL> </Response>
解　説	指定した範囲の地物をXML形式で取得します。 正常に処理された場合ReturnFlagにはTrueが返ります。 実際にデータを取得するには、戻り値のURLからダウンロードします。

4.11.2. 基盤データの更新 (*UpdateRoadData*)

＊バージョンアップ中につき利用できません＊

API	*UpdateRoadData*
機　能	特定の道路基盤データを更新する
引数の形式	
引数の内容	道路基盤データの更新データファイルを指定
レスポンスの種類	text/xml
レスポンスの内容	<Response> 　<ReturnFlag>*(True/False)*</ReturnFlag> </Response>
解　説	特定の道路基盤データを更新します

電子納品管理システム

外部仕様書

平成 15 年 2 月

【電子納品管理システム　外部仕様書　目次】

1.　目的 ..... エラー! ブックマークが定義されていません。

2.　システム構成 **4**

3.　動作環境 **5**

3.1.　道路 GIS サーバの動作環境 5

3.1.1.　ハードウェアの必要性能 5

3.1.2.　オペレーションシステム 5

3.1.3.　DBMS 5

3.1.4.　Web サーバ 5

3.1.5.　道路 GIS コアモジュールの動作に必要な環境 5

3.2.　電子納品管理サーバの動作環境 5

3.2.1.　ハードウェアの必要性能 5

3.2.2.　オペレーションシステム 5

3.2.3.　電子納品管理サーバの動作に必要な環境 5

3.3.　クライアントの動作環境 6

3.3.1.　ハードウェアの必要性能 6

3.3.2.　オペレーションシステム 6

3.3.3.　Web ブラウザ 6

3.3.4.　ビューワ・プラグインの動作に必要な環境 6

3.4.　開発環境 6

3.4.1.　開発言語 6

4.　機能定義 **7**

4.1.　画面構成 7

4.2.　地図画面の操作 13

4.2.1.　表示位置の移動 13

4.2.2.　電子納品成果物の登録 16

4.3.　表示画面の操作 19

4.3.1.　電子納品成果物のダウンロード 19

4.3.2.　電子納品成果物の更新 ..... **エラー! ブックマークが定義されていません。**

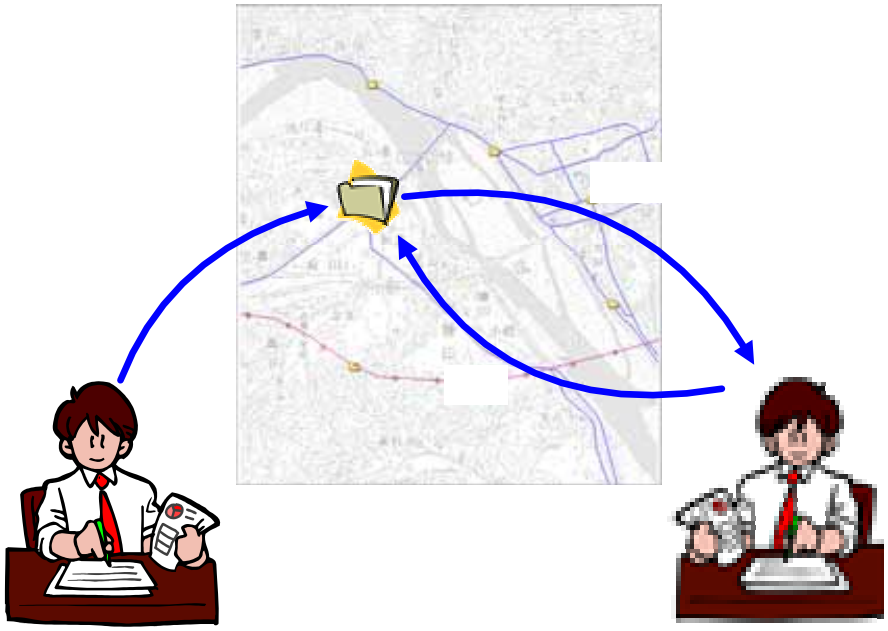
1. はじめに

1.1. 目的

電子納品は、調査、設計、工事などの各業務段階の最終成果を電子データで納品することにより、公共事業の各事業段階で利用している基礎資料を電子化し、共有・再利用することを目的としている。

電子納品により納められる電子データは、その情報を整理・格納するための索引情報を持っており、この情報を利用することにより、事業情報の効率的な管理が可能となる。

本システムは、電子納品の索引情報に格納された空間情報（位置・範囲の情報）を利用し、地図上で電子納品の登録・表示・検索を実現するものである。システムは、国土交通省内のネットワーク上で動作するものとし、ネットワークを介して地図上に電子納品の登録・削除および閲覧の機能を提供する。



システムの概念図

システムが提供する主な機能は次のようなものである。

- 電子納品成果の登録・修正・削除
- 地図上での電子納品成果の検索
- 索引情報による電子納品成果の検索
- 電子納品成果のダウンロード

1.2. 与条件

構築に当たっては、次の要領・基準を元に作成するものとする。

- 電子納品運用ガイドライン(案)
- 土木設計業務等の電子納品要領(案)
- 工事完成図書の電子納品要領(案)
- 測量成果電子納品要領（案）

また、電子納品成果を位置情報に関連づける際には、国総研内に設置されている道路 GIS サーバと連携して動作するものとする。道路 GIS サーバが公開する機能については、次の資料を参考にするものとする。

- 道路 GIS SDK 開発者マニュアル

1.3. 本文書の位置付け

本文書は、電子納品管理システムの外部的な仕様を定義するものである。
外部仕様としては、次の項目を定義する。

(1) アプリケーション仕様

- システム構成
- ハードウェアの動作環境
- ソフトウェアの動作環境
- 開発言語

(2) 機能要件

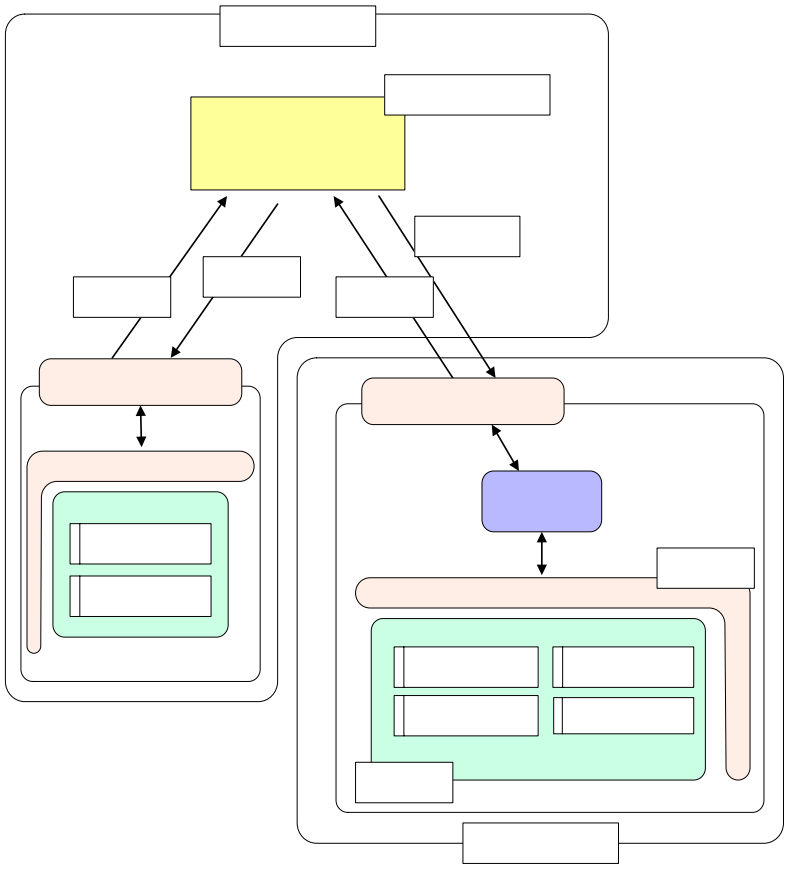
- 道路台帳管理データビューワへの要求機能

本アプリケーション開発では、、本仕様書に定義した内容に沿って開発を行うものとし、要求仕様に変更がある場合には、本文書の更新を含めて対応を行うものとする。

2. システム構成

電子納品管理システムの基本構成は、道路 GIS SDK の基本的な枠組みに沿って構築するものとする。

電子納品管理システムの基本構成を下図に示す。



電子納品管理システムのシステム構成は、次の通りとする。

- 道路に関する基本的な情報とその表示機能を提供する**道路ＧＩＳサーバ**と。電子納品データを一括して管理する**電子納品管理サーバ**、およびユーザが実際に操作を行うクライアントＰＣの Web ブラウザ上で動作する**クライアント**から構成される。
- **電子納品管理サーバ**は、電子納品成果を格納する**ディレクトリサービス**と、電子納品に関する管理情報を格納する**管理 DB** から構成され、その機能は**電子納品管理 API** として公開される。
- 道路に関する基本的な情報とその表示機能を提供する道路ＧＩＳサーバの仕様については、**道路 GIS SDK** の仕様に準拠するものとする。
- 電子納品管理システムのクライアントは、電子納品管理サーバ、および道路 GIS サーバの双方に接続し、情報のやり取りを行う。

本仕様書は、上記のような仕組みで動作するクライアントアプリケーションの機能の定義を行うものである。

業務
データを

電子納

3. 動作環境

次の環境で動作するものとする。

3.1. 道路 GIS サーバの動作環境

3.1.1. ハードウェアの必要性能

CPU: Intel Pentium 4 プロセッサ 1GHz 以上

メモリ: 1GB 以上

HDD: 80GB 以上

ネットワーク: 10Base-T、またはそれ以上の速度に対応したネットワーク環境とする

3.1.2. オペレーションシステム

Windows 2000 Server がインストールされていること。

3.1.3. DBMS

PostgreSQL ver 7.3.x がインストールされていること。

3.1.4. Web サーバ

Http サーバとして、IIS 5.x がインストールされていること。

3.1.5. 道路 GIS コアモジュールの動作に必要な環境

道路 GIS コアモジュールは、Java Servlet として、Web サーバと連携して動作する。

Java Servlet エンジンとして、Apache Tomcat 4.x がインストールされていること。

また、Java の動作環境としては、Java2 SDK 1.4.2_01 以降がインストールされていること。

3.2. 電子納品管理サーバの動作環境

3.2.1. ハードウェアの必要性能

CPU: Intel Pentium 4 プロセッサ 1GHz 以上

メモリ: 1GB 以上

HDD: 80GB 以上

ネットワーク: 10Base-T、またはそれ以上の速度に対応したネットワーク環境とする

3.2.2. オペレーションシステム

Windows 2000 Server がインストールされていること。

3.2.3. 電子納品管理サーバの動作に必要な環境

Http サーバとして、IIS 5.x がインストールされていること。

WEB を介したファイル送受信の手段として、FTP プロトコルが利用可能であること。

道路 GIS コアモジュールは、Java Servlet として、Web サーバと連携して動作する。

Java Servlet エンジンとして、Apache Tomcat 4.x がインストールされていること。
また、Java の動作環境としては、Java2 SDK 1.4.2_01 以降がインストールされていること。

3.3. クライアントの動作環境

3.3.1. ハードウェアの必要性能

CPU: Intel Pentium 2 プロセッサ 500Hz 以上
メモリ: 128MB 以上
HDD: 空き容量 200MB 以上
ネットワーク: 10Base-T、またはそれ以上の速度に対応したネットワーク環境とする

3.3.2. オペレーションシステム

Windows 98、Me、NT4.0(SP6a 以降)、Windows 2000(SP2 以降)、または WindowsXP インストールされていること。

3.3.3. Web ブラウザ

MicroSoft Internet Explorer 5.5 以降がインストールされていること。

3.3.4. ビューワ・プラグインの動作に必要な環境

地図画像を表示するための機能は Java Applet によって構築するものとする。
プラグイン機能の動作は、Java Runtime Environment 1.4.2_01 がインストールされていることを条件とする。

3.4. 開発環境

3.4.1. 開発言語

道路 GIS SDK の開発には、Java2 SDK 1.4.2_01 以降を利用。
電子納品管理システムのサーバ部分の開発には Java Servlet を利用するものとする。

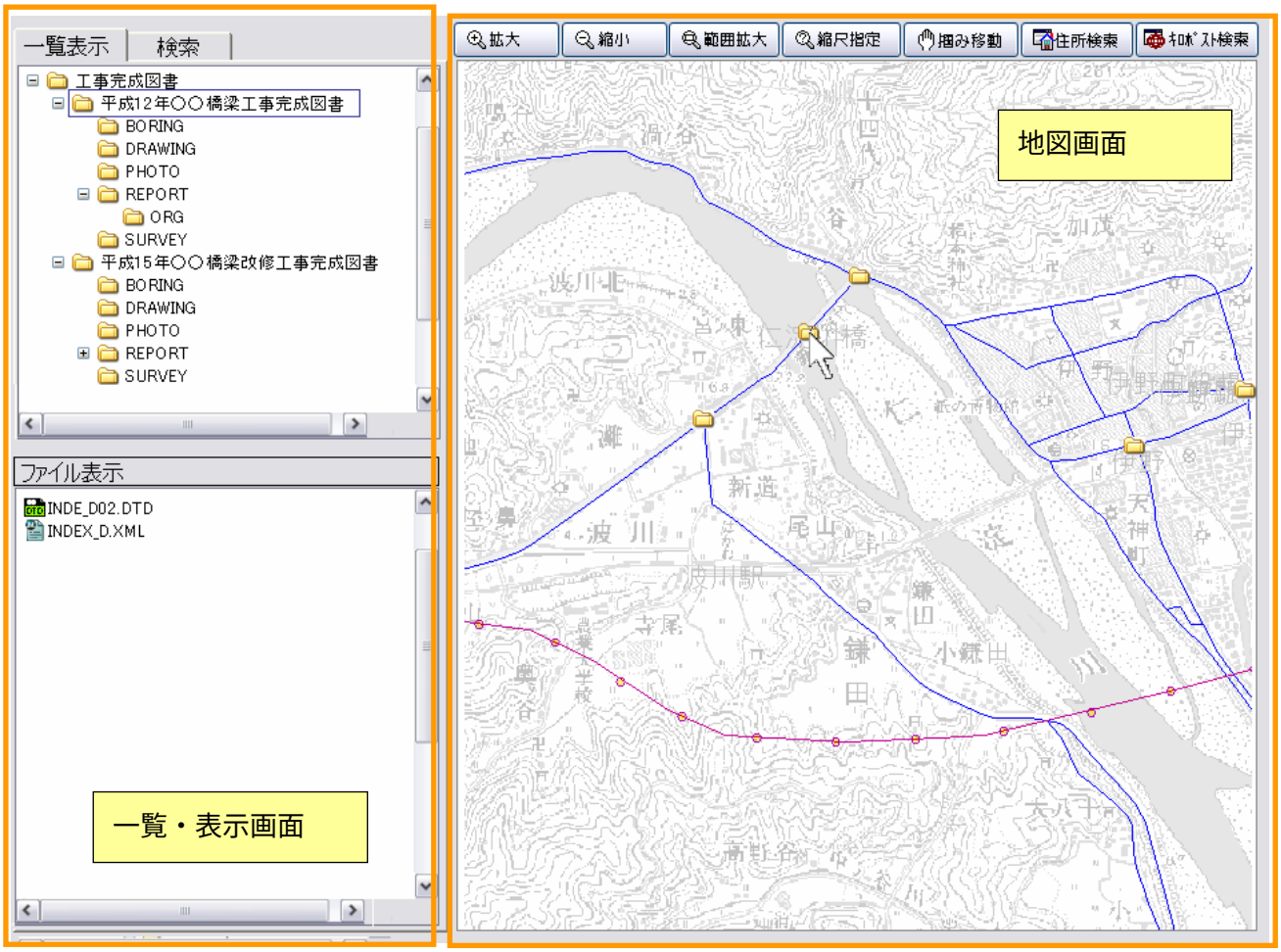
4. 機能定義

電子納品管理システムに必要な機能を定義する。

4.1. 画面構成

電子納品管理システムを構成する画面は、電子納品箇所を地図画面上で図示するための**地図画面**と、電子納品成果のファイルを表示する**表示画面**、また、属性データなどで電子納品の内容を検索する**検索画面**の３画面から構成される。

地図画面は画面全体の右側に配置され、表示画面と検索画面は画面左側に配置され、タブで切り替える構成とする。

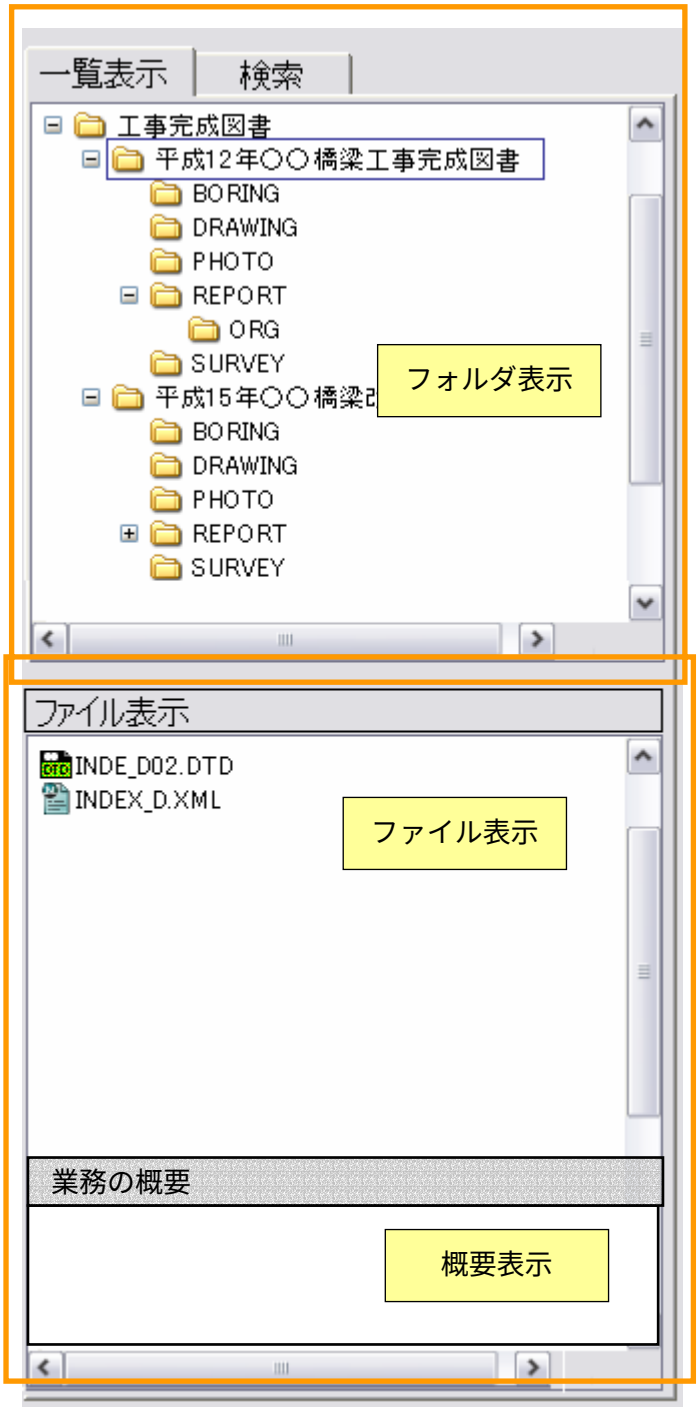


・地図画面では、道路基盤データ、DRM データ、数値地図 25000 を背景に表示し、その上に電子納品成果が登録されていることを示す、フォルダマークを配置できるものとする。

・地図上のフォルダマークをクリックすると、その位置に格納された電子納品成果が表示画面に表示される。

・地図上に電子納品成果を格納したフォルダをドラッグすると、その位置に電子納品成果を格納するための「登録ダイアログ」表示され、指示に従い操作することで、サーバーに電子納品成果を登録する。

【表示画面】



・表示画面には、現在選択中の電子納品成果の内容を表示する。

・画面の上半分は、ファイル構成を示すフォルダ表示を行い、下半分はフォルダに格納されているファイル名を表示する。

・フォルダ表示画面上のフォルダをクリックすると、そのフォルダに含まれるファイルが、ファイル表示画面に一覧表示される。

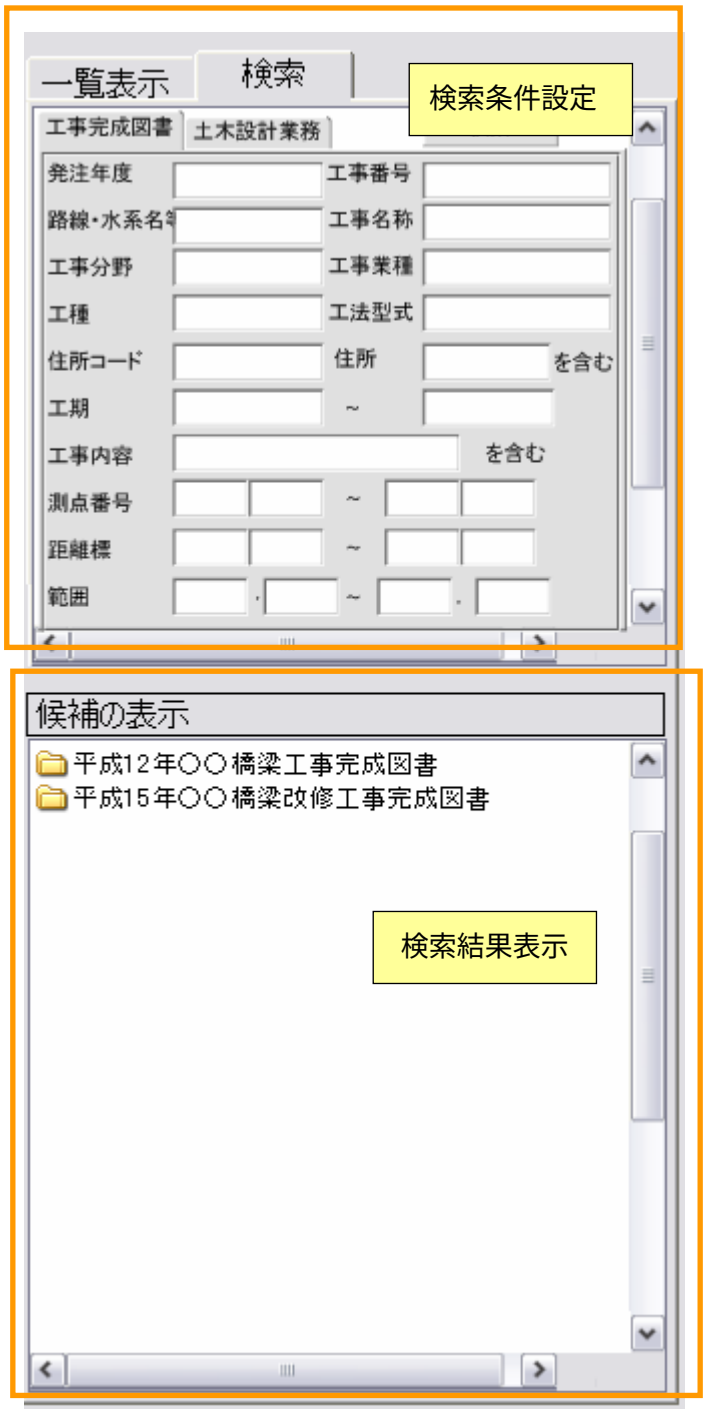
・フォルダ表示画面上に、電子納品を格納したフォルダをドラッグすると、登録ダイアログが表示され、電子納品成果物の登録を行う。
ただし、既存名称の電子納品成果物を登録しようとした場合には、置き換えするするか選択する。

・ファイル表示画面にファイルをドラッグすると、現在選択中のフォルダにファイルを追加する。ただし、同名のファイルが存在した場合は、更新するかどうかを確認する画面が表示される。

・フォルダ画面、またはファイル画面上のファイル、またはフォルダをダブルクリックすると、電子納品成果のダウンロード、または表示を選択するダイアログを表示する。

- 概要画面には、索引ファイルの次の項目を表示する。
  - 工事完成図書
    INDEX_C.XML の<工事内容>エレメント
  - 設計成果
    INDEX_D.XML の<業務概要>エレメント

【検索画面】



・検索画面では、電子納品成果をメタデータから検索するための操作を行う。

・画面の上半分では、検索条件の設定を行い、下半分に検索結果を表示する。

・検索条件設定画面では、電子納品の索引情報（INDEX_?.XML に記述された当該業務に関する情報）を検索条件として、その条件に当てはまる電子納品成果物を検索することができる。

・検索条件設定画面に設定できる検索条件は、分野別の電子納品要領により異なるため、それぞれの電子納品要領により、タブで切り替えを行う。

・「検索」ボタンをクリックすると、検索条件に合致した電子納品成果物の一覧が、検索結果表示画面に表示される。

・検索結果表示画面上のどれかをクリックすると、表示画面に切り替わり、その電子納品成果物に対応するファイルの一覧が表示される。

【検索画面の詳細】

①工事完成図書の場合

INDEX_C.XML の下記項目を検索対象とする。

<工事件名等>より

- 発注年度　・工事番号　・路線名　・工事名称(*1)
- 工事分野　・工事業種　・工種　・工事型式
- 住所コード　・住所(*1)　・工事開始日、工事終了日(*)
- 工事内容(*1)

<場所情報>より

- 起点側測点、終点側測点(*2)　・起点側距離標、終点側距離標(*2)
- 西側境界座標、東側境界座標、北側境界座標、南側境界座標(*2)

<施設情報>より

- 施設名称(*1)

＜発注者情報＞より

- 発注者-大分類(*1)　・発注者-中分類(*1)　・発注者-小分類(*1)
- 発注者コード

<請負者情報>より

・ 請負者名(*1)　　者コード

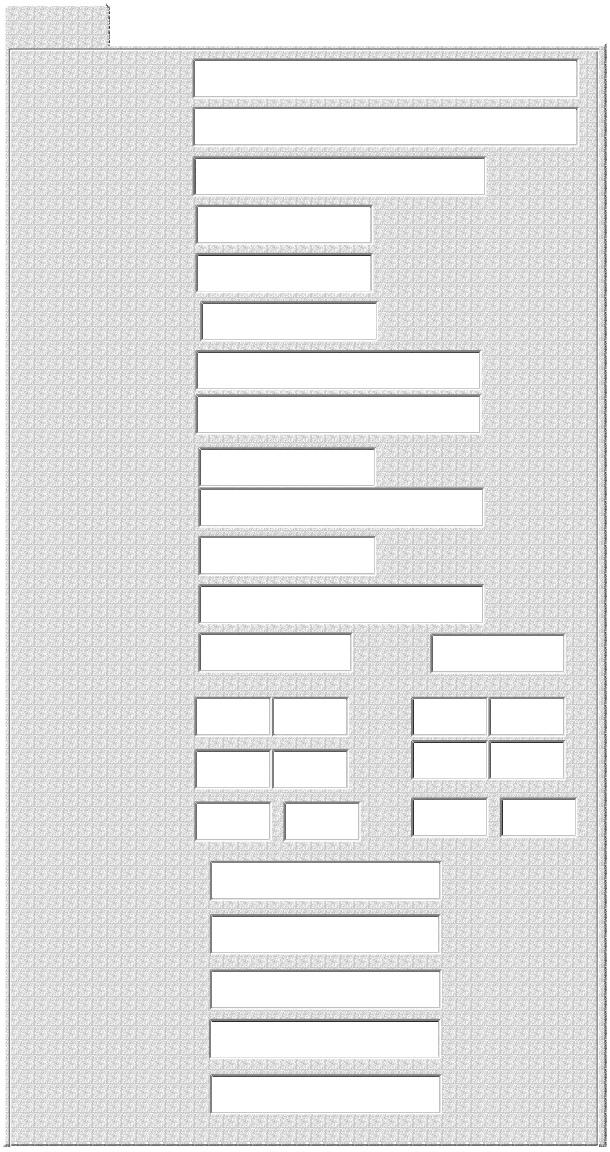
（*1） 入力したキーワードを含むものとする

(*2) 入力した範囲に、内包されるものとする。

（例） 検索条件の工事開始日＜データの工事開始日＜データの工事終了日＜検索条件の工事終了日

また、指定した検索のすべてに AND 条件で合致するもののみを抽出する。

②設計成果の場合



設計成果

設計書コード

-<業務件名等>から

・TECRIS 登録番号　・設計書コード　・業務名称(*1)

・住所コード　・住所(*1)　・業務対象水系路線等コード

・業務対象水系路線名(*1)　・履行期間-着手、履行期間-完了(*2)

<場所情報>から

・起点側測点、終点側測点(*2)　・起点側距離標、終点側距離標(*2)-

・西側境界座標経度、東側境界座標経度、北側境界座標緯度、北側境界座標緯度(*2)

<施設情報>から

・施設名称(*1)

<発注者情報>から

・発注者機関コード　・発注者機関事務所名(*1)

<受注者情報>から

・受注者名(*1)　・受注者コード

<業務情報>から

・主な業務の内容　・業務分野コード　・業務キーワード(*1)　・業務概要(*1)

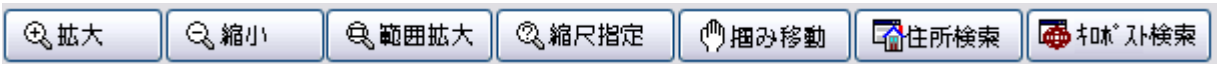
(*1).(*2)は工事完成図書と同様

4.2. 地図画面の操作

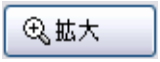
4.2.1. 表示位置の移動

表示位置の移動は、地図画面上部の操作ボタン群を用いて操作する。

【操作ボタン群】



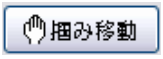
各ボタンの機能は以下の通りである。



現在表示中の範囲を中心に２倍に拡大する。

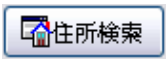
現在表示中の範囲を中心に 1/2 倍に縮小する。

矩形の範囲を指定し、その範囲が最大表示されるように拡大する。



マウスカーソルを手のひらモードを変化させ、画面上でクリックした点を中心に、マウスをドラッグした方向に画面を移動させる。

縮尺指定ダイアログにより表示縮尺を指定させ、現在の表示範囲を中心に指定縮尺で表示を行う。

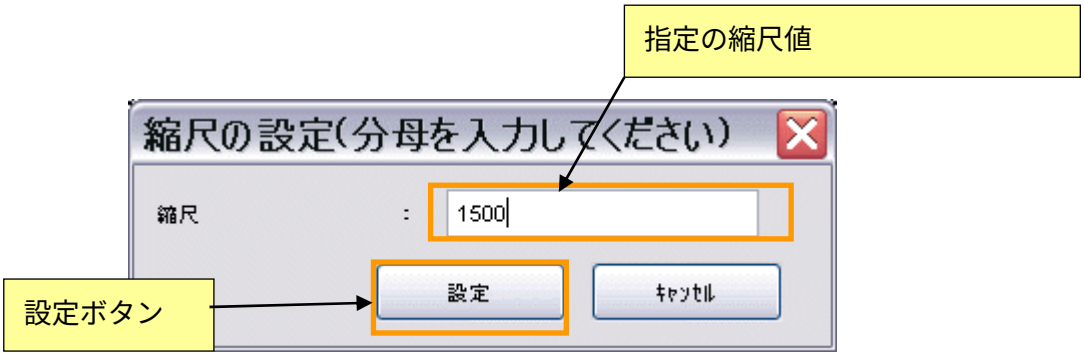


住所検索ダイアログを表示させ、指定した住所を適切な縮尺で表示する。

キロポスト指定ダイアログを表示させ、指定したキロポスト周辺を適切な縮尺で表示する。

4.2.1.1. 縮尺の指定

縮尺指定ダイアログの画面構成は以下の通りである。

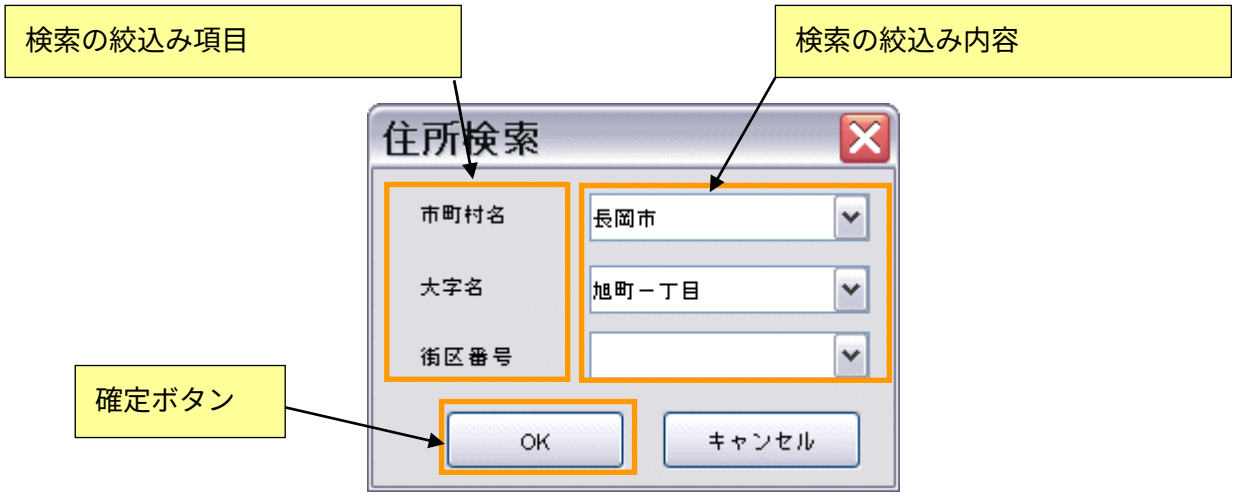


縮尺を設定し、「設定」ボタンをクリックすると、現在の表示範囲を中心に、指定縮尺で表示を行う。

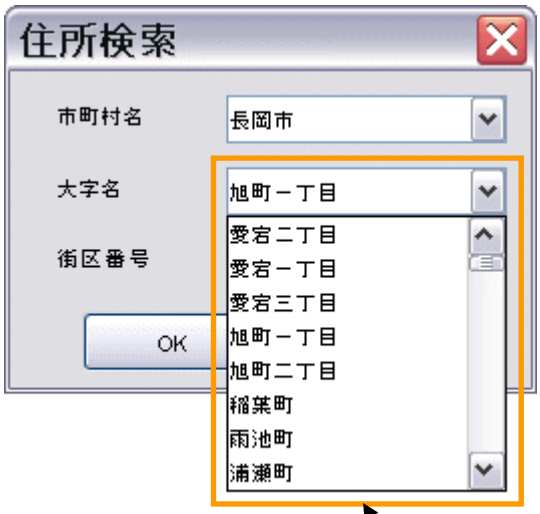
「キャンセル」ボタンをクリックすると、縮尺設定画面を閉じる。

4.2.1.2. 住所検索

住所検索ダイアログの画面構成は以下の通りである。



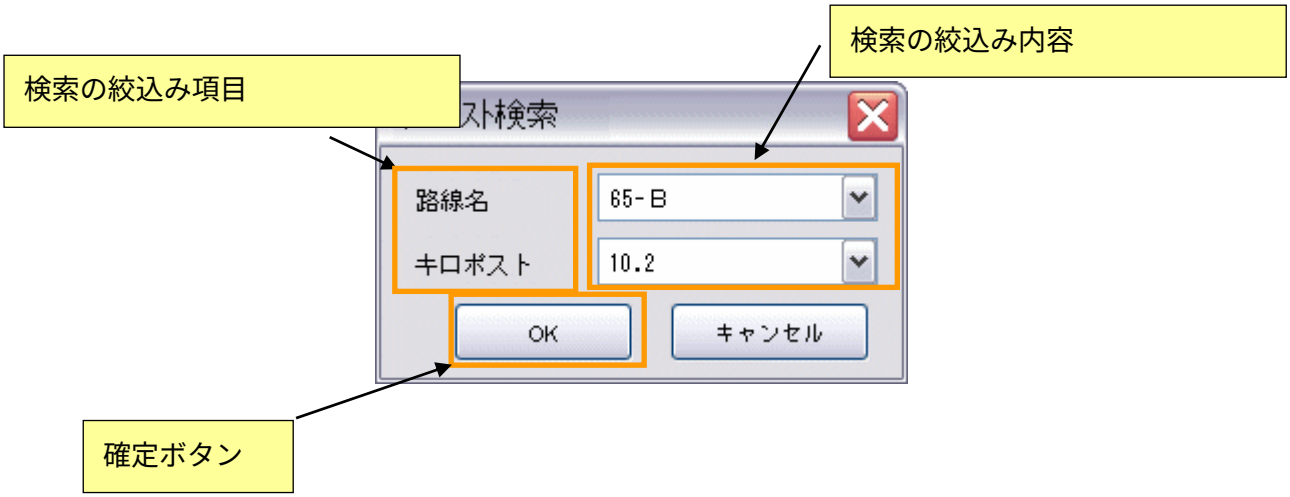
住所検索を行う際には、左の絞込み項目に対応した絞込み内容を右側のドロップダウンリストから選択し、徐々に検索対象を絞り込んでいく形式を採る。



ドロップダウンで絞り込み内容を選択

4.2.1.3. キロポスト検索

検索ダイアログの画面構成は以下の通りである。



「路線名」および「キロポスト」に検索したい路線とそのキロポストを指定する。

「ＯＫ」ボタンをクリックすると、その路線が表示される。

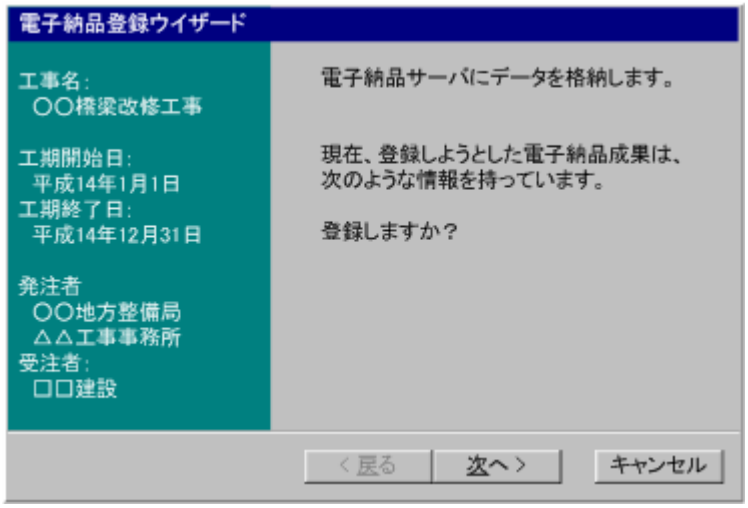
「キャンセル」ボタンをクリックすると、キロポスト検索画面を閉じる。

4.2.2. 電子納品成果物の登録

【登録開始画面】

地図画面上に、電子納品成果物をドラッグすると、次のような画面が表示される。

画面左側に、登録しようとする電子納品の概要が表示される。

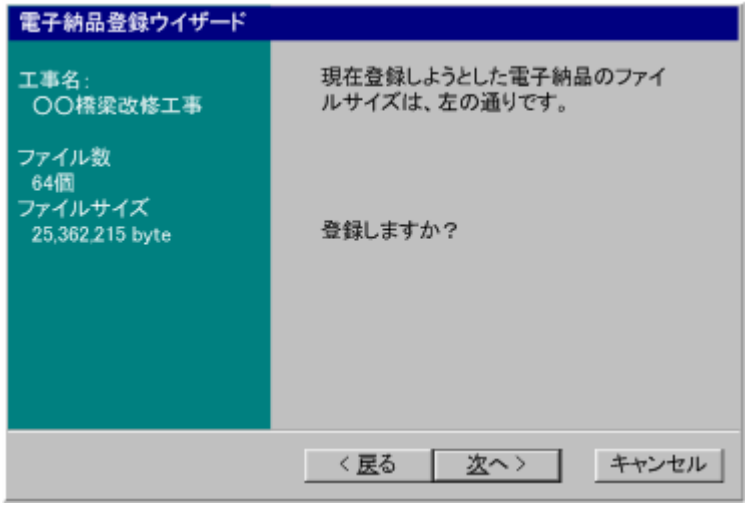


「次へ>」ボタンをクリックすると、登録確認画面に進む。

「キャンセル」ボタンをクリックすると、電子納品登録画面を閉じる。

【登録確認画面】

実際に登録するファイル群の情報が表示し、登録の確認画面が表示される。



「戻る」ボタンをクリックすると、登録開始画面に戻る

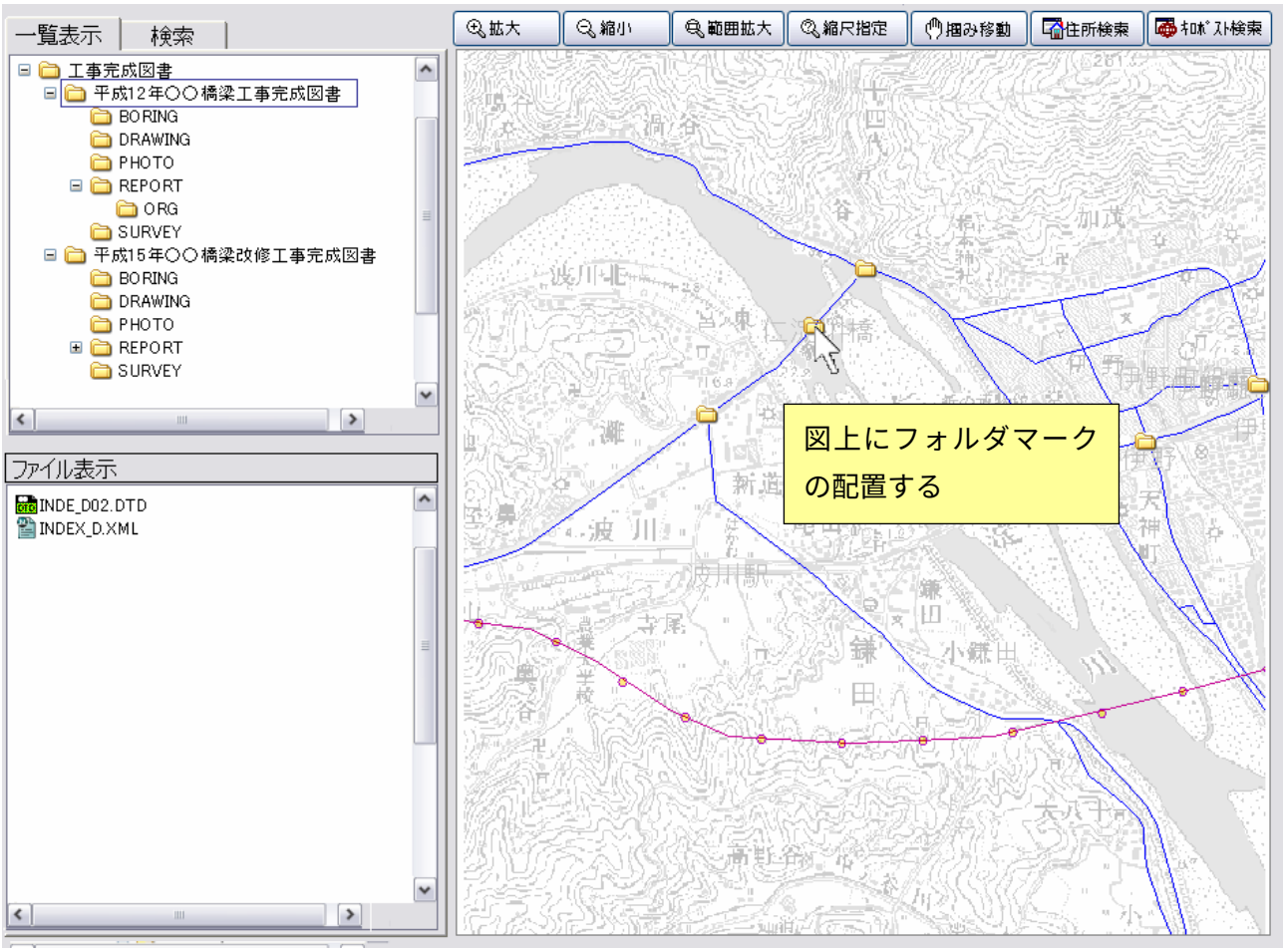
「次へ>」ボタンをクリックすると、つぎへ進む

「キャンセル」ボタンをクリックすると、電子納品登録画面を閉じる。

【位置表示】

登録情報に位置情報座標が登録されていたら、その中心点にフォルダマークを配置し、その場所にビューを移動し、1/1500 の縮尺で表示する

位置情報が登録されていない場合には、図形追加モードとなり、地図上のクリックした点にフォルダマークを追加する。

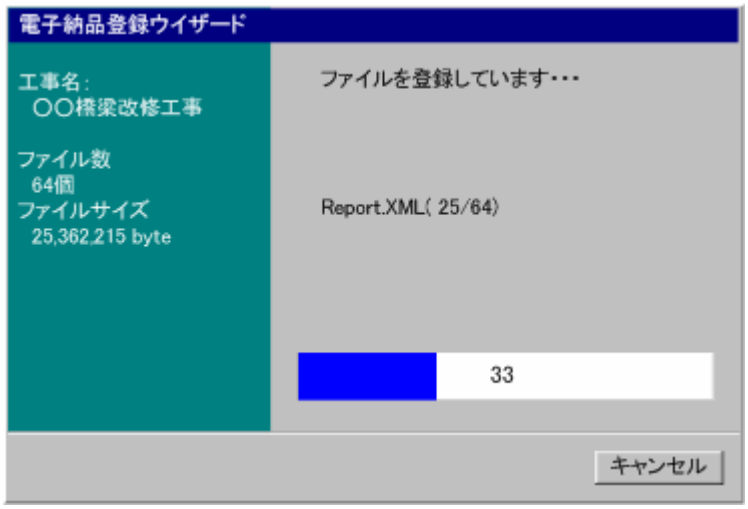


いずれの場合も、図上にフォルダマーク配置後に、「この場所に電子納品を登録しますか？」というメッセージダイアログを表示し、Yes の場合のみ、次に進む。No の場合には、登録画面から抜ける

再配置を選んだ場合には、フォルダマークを登録する位置を再度選択する。

【登録進行画面】

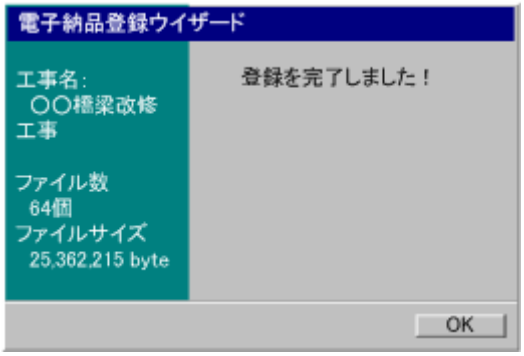
登録を開始すると、登録の進行状況を表示する画面が表示される。



「キャンセル」ボタンをクリックすると、登録を中断して終了する。

【登録完了画面】

登録が完了すると、登録が完了を知らせる画面が表示される。



「ＯＫ」ボタンをクリックすると、登録完了画面が閉じられる。

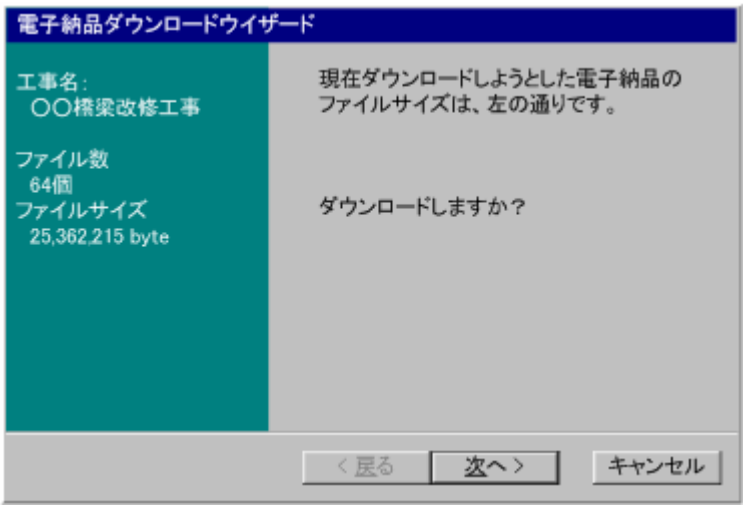
4.3. 表示画面の操作

4.3.1. 電子納品成果物のダウンロード

表示画面上で、フォルダ、またはファイルをダブルクリックすると、そのフォルダ、またはファイルをダウンロードのためのウイザード画面が表示される。

【ダウンロード確認画面】

画面左には、ダウンロードしようとするファイル／フォルダのサイズ情報が表示される。



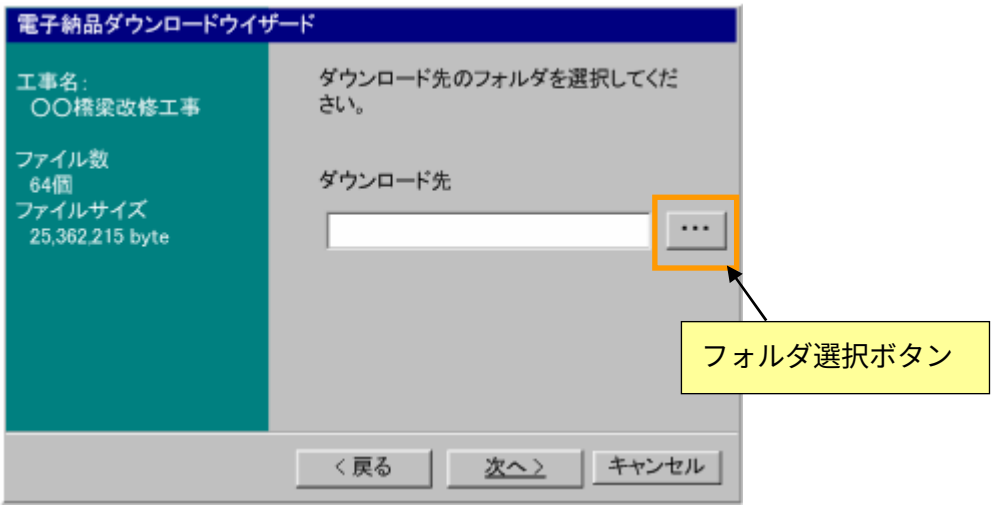
「次へ>」ボタンをクリックすると、ダウンロード先を指定する画面に移動する。

「キャンセル」ボタンをクリックすると、ダウンロード画面を閉じる。

【ダウンロード先指定画面】

ダウンロード先を指定する画面では、電子納品の成果を格納するフォルダを指定する。

フォルダ選択ボタンをクリックすると、ダウンロード先のフォルダを選択するダイアログが表示される。

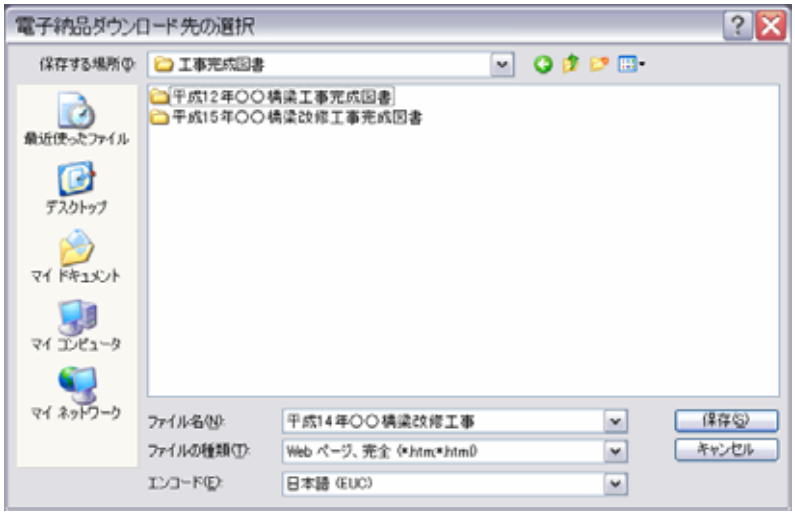


「ダウンロード先」のフォルダを指定した後、「次へ＞」ボタンをクリックすると、ダウンロード画面に進み、ダウンロードが実行される。

「＜戻る」ボタンをクリックすると、ダウンロード確認画面に戻る。

「キャンセル」ボタンをクリックすると、ダウンロードを中断する。

【ダウンロード先のフォルダの指定】



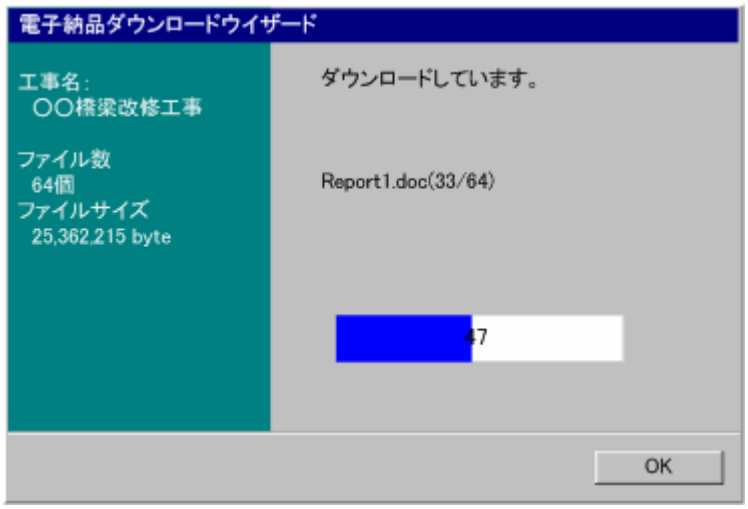
フォルダ選択ダイアログでは、ダウンロードを行うフォルダを指定する。

「保存」ボタンをクリックすると、選択したフォルダ名をダウンロード先指定画面の「ダウンロード先」のテキストボックスに代入する。

「キャンセル」ボタンをクリックすると、フォルダ指定画面を閉じる。

【ダウンロード画面】

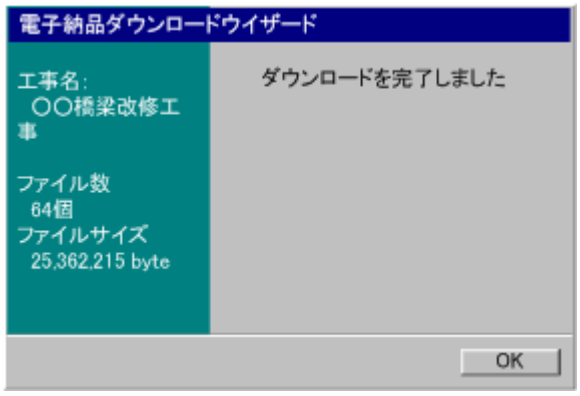
ダウンロードを開始し、ダウンロードが終了するまでの間、ダウンロード画面が表示される。



「キャンセル」ボタンをクリックすると、ダウンロードを中断する。

【ダウンロード完了画面】

ダウンロードが完了すると、ダウンロード完了画面が表示される。



「OK」ボタンをクリックすると、ダウンロード完了画面が表示される。

道路 GIS

サーバインストールマニュアル

平成 16 年　月

【道路 GIS　サーバインストールマニュアル 目次】

1. システムの概要 2

2. サーバ稼働環境 3

3. JAVAのインストール 3

4. WEBサーバのインストール 4

5. データベースのインストール 5

5.1. 本体のインストール 5
5.2. データベース環境設定 5
5.3. JDBCドライバのインストール 9
5.4. 開発用データのインポート 9

6. サーブレットコンテナのインストール 10

6.1. インストール 10
6.2. IISとの連携 11

7. 道路GISのインストール 19

8. 道路GISの動作確認 19

9. その他のデータのインポートについて 20

9.1. DRMデータのインポート 20
9.2. ラスタデータの登録 23
9.2.1. インデックスマップの登録 23
9.2.2. 数値地図の登録 24
9.2.3. その他のラスタデータの登録 25

10. TOMCATのライセンスについて 26

1. システムの概要

道路 GIS SDK の基本構成を下図の通りです。

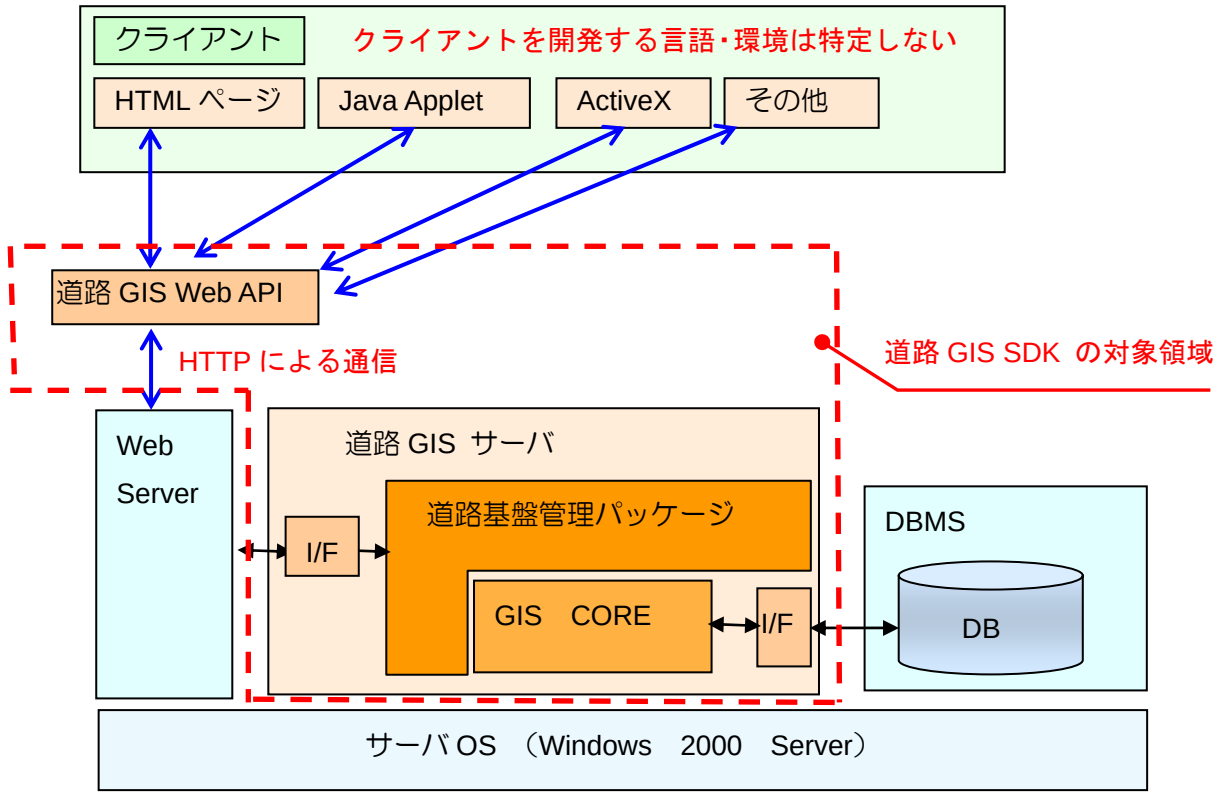


図 1 道路 GIS SDK の構成

道路 GIS のサーバ部分は、道路基盤データを操作・処理するための **GIS CORE** モジュールと、GIS CORE モジュールの機能を内包し、その他背景データなど、道路基盤データ以外で必要となるデータとあわせて実務に必要な機能を提供する**道路基盤管理パッケージ**から構成されます。これらサーバ部分は、Java サーブレットで構成され、Web サーバーを介してその機能を公開します。・

道路 GIS アプリケーションのクライアント部分は、道路 GIS SDK の提供する**道路 GIS Web API** を用いて、ユーザが画面上で行った操作に対応した問い合わせを行い、その処理結果を取得し、その結果を解析してアプリケーションの画面に反映します。

本マニュアルは、道路 GIS のクライアントアプリケーション開発用サーバのインストールについて解説したものです。

2. サーバ稼働環境

道路 GIS のサーバ稼働環境は、以下のとおりです。

（1） 稼動推奨環境

OS:Windows 2000 （SP2 以上）Server、開発には WindowsXP も利用可。

メモリ：1 GB 以上

HDD:30GB 程度（データ量により異なる）

（2） 動作ソフト

Web サーバ：IIS 5.0

Java 実行環境：Java2 Runtime Environment, Standard Edition 1.4.2_02（推奨）

Servlet コンテナ：Tomcat 3.2.4

データベース:PowerGres(Windows 版 PostgreSQL) 1.0

WebGIS：道路 GIS

（3） フォルダ構成概要

```
C:¥
 ├──¥jakarta-tomcat-3.2.4   -----サーブレットコンテナ
 ├──¥skroot       ----------道路 GIS HP データ  （仮想ディレクトリ skpb）
 ├──¥sttkdat      ----------道路 GIS データフォルダ
 ├──¥pdata        ----------データベースデータフォルダ
 ├──¥j2sdk1.4.2_02    ---------- Java2 SDK  1.4.2_02
 └──¥jdbc             ---------- JDBC ドライバー
```

3. Java のインストール

Java 開発環境をインストールします。インターネットから J2SE v 1.4.2_02 の SDK を以下の URL からダウンロードします。

URL http://java.sun.com/j2se/1.4.2/ja/download.html

ダウンロードしたファイルをインストールします。画面の指示に従い以下のパスにインストールしてください。

インストールパス： C:¥j2sdk1.4.2_02

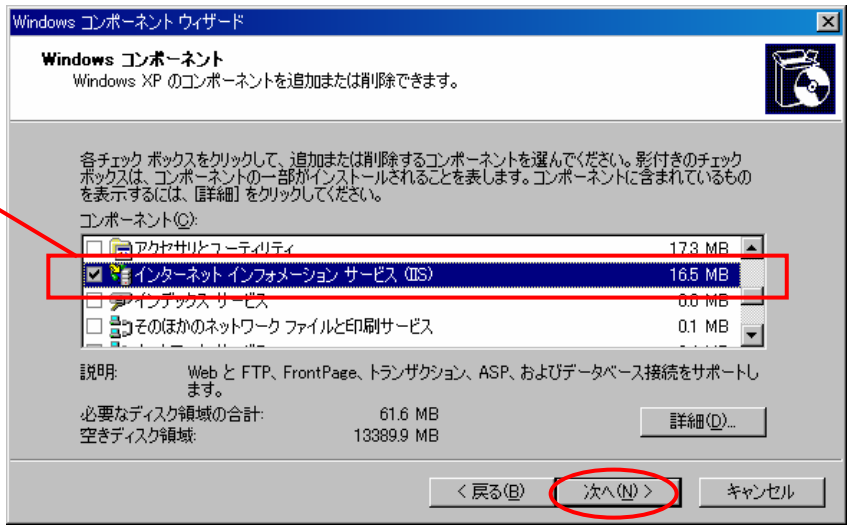
他の任意のパスにインストールもできますが、このパスにインストールすると、設置済みである Tomcat の環境設定ファイルの編集が不要になります。

4. Web サーバのインストール

開発のために利用する Web サーバを IIS として解説します。IIS がすでにインストール済みである場合、再度インストール作業をする必要はありません。

コントロールパネルのアプリケーションの追加と削除（WindowsXp の場合、プログラムの追加と削除）を選択します。 Windows コンポーネントの追加と削除を選択します。 Windos コンポーネントウィザードの画面のコンポーネントの欄から、“インターネット インフォメーション サービス(IIS)”のチェックを ON にします。

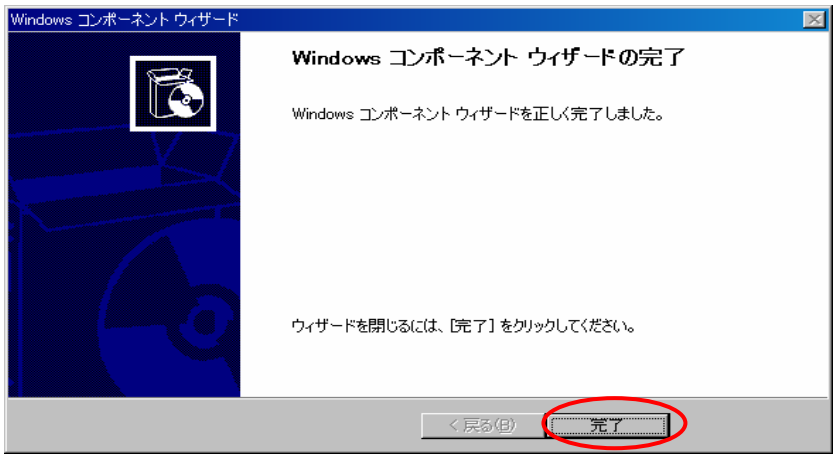
チェックを ON にする。



＜次へ＞ボタンをクリックして、インストールが終了するまで待ちます。

途中で CD-ROM を求められた場合は、当該 OS の CD を挿入し、画面の指示に従ってください。

以下の画面が表示されましたら＜完了＞ボタンをクリックしてください。IIS のインストールは終了です。



仮想ディレクトリの設定、サーブレットコンテナとの連携のための設定については、後述します。

5. データベースのインストール

道路 GIS でデータを管理するために用いるデータベースのインストールを行います。使用するデータベースは、本業務の開発では PowerGres（パワーグレス）を使用します。PowerGres は、オープンソースで実績のある PostgreSQL をベースに Windows 向けに最適化したデータベースです。サーバ１台についき PowerGres のライセンスが１つ必要ですが、接続ユーザ数に制限がないため、Web アプリケーションのデータベースとして経済的に優れているといえます。

5.1. 本体のインストール

PowerGres を CD-ROM より、インストールします。インストール手順については、販売元の株式会社 SRA 社の製品付属の PowerGres 操作マニュアルの９ページのインストールの手順を参照してインストールしてください。次に、PowerGres を起動する前に、以下に示す製品マニュアルの１２ページから１７ページの設定を行います。

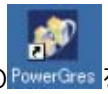
- データベースディレクトリの設定（X:¥pdata とします。X:は任意のドライブです。）
- データベースディレクトリの初期化
- データベースの基本の設定
- ネットワーク接続認証の設定
- Windows サービスに登録

5.2. データベース環境設定

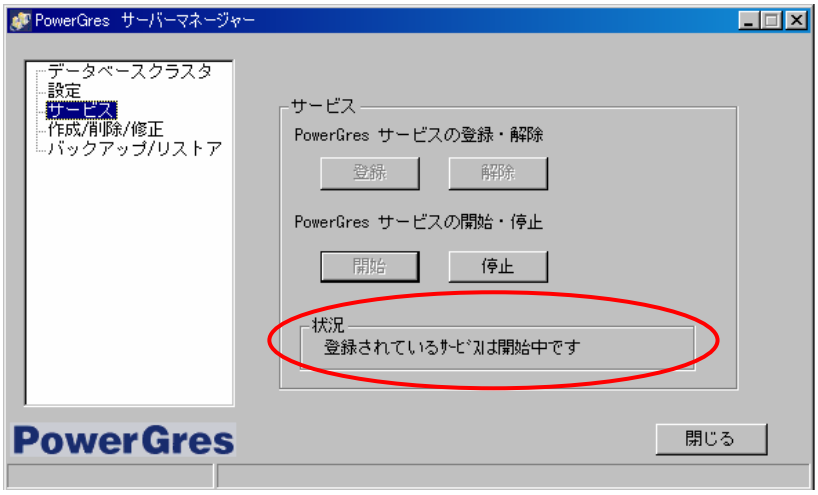
道路 GIS で使用するデータベースの環境を設定します。この作業により作成済みのテーブルがデータベースに登録されます。データベースに登録されているデータは、いかなる理由があろうとも道路 GIS のアプリケーションの開発目的以外に使用することはできません。本データを第三者に販売、譲渡その他処分を無断で行うことを禁止します。

PowerGres のデータベースディレクトリを X:¥pdata とします。(x:は任意のドライブです)　データをインストールするには、PowerGres が動作している場合、停止させてから作業を行います。PowerGres の動作状況の確認をします。

①PowerGres の動作状況の確認方法

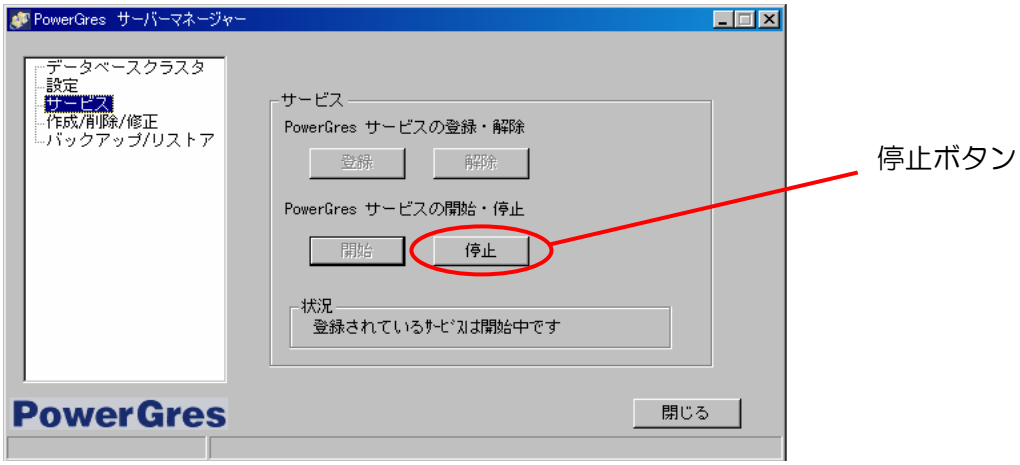
デスクトップの 

をクリックし、PowerGres サーバマネージャーを起動します。
サーバマネージャーのコンソール画面の左側のツリーより、[サービス]をマウスで選択します。
以下の画面の例のように状況“登録されているサービスは開始中です”と表示されている場合、データベースは起動中です。したがって、次の手順でデータベースを停止します。



②PowerGres の停止方法

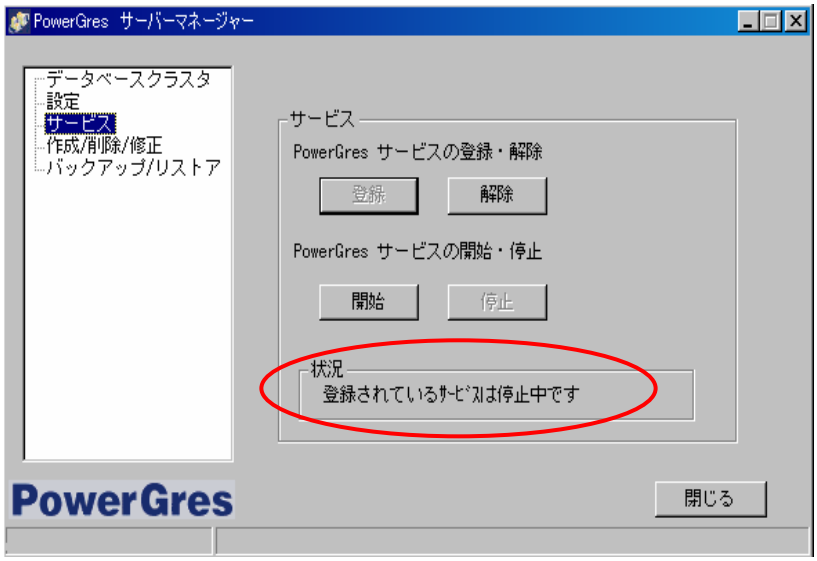
PowerGres のサーバマネージャーを起動します。サービス欄の PowerGres サービスの開始・停止で、開始ボタンにマスクがかかっており、停止ボタンにマスクがかかっていない場合、PowerGres が起動している状態です。PowerGres が起動している状態である場合、＜停止＞ボタンをクリックし、PowerGres を停止します。



状況の欄を確認します。以下に示すように、「登録されているサービスは停止中です」と表示されていれば、データベースは停止している状態です。

③開発用データベースの環境構築

以下の画面の状態で、開発用データベースの環境を保存したデータをインストールします。



PowerGres のサーバマネージャーを閉じるボタンで閉じます。データベース停止の状態で、X:¥pdata フォルダ配下に、以下のコールドバックアップ済みのデータをコピーします。まず、サーバの X:¥pdata フォルダ配下の以下のディレクトリを削除します。

￥base
￥global
￥pg_clog
￥pg_xlog

インストール CD より、￥開発用 DB 設定の以下の 4 つのフォルダをサーバの X:¥pdata にコピーします。

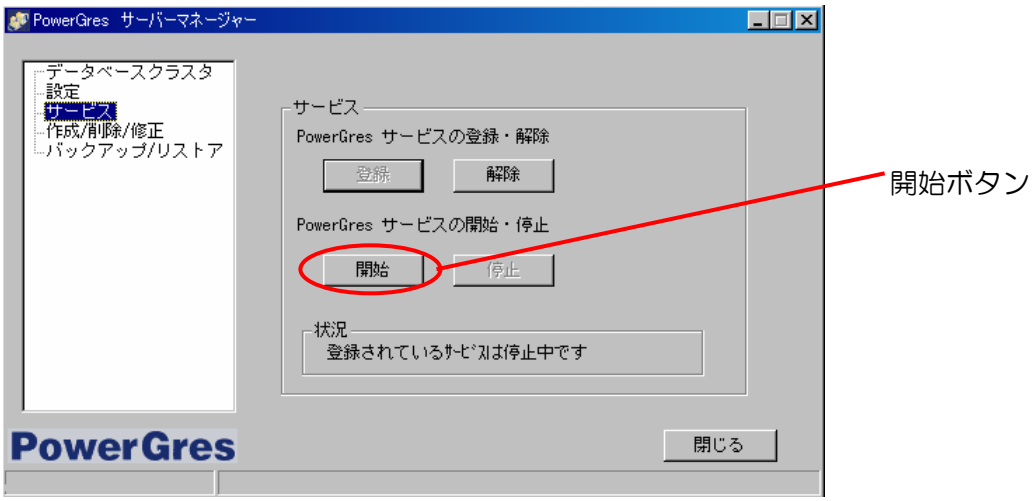
￥開発用 DB 設定￥DB￥base
￥global
￥pg_clog
￥pg_xlog

コピー後のサーバのフォルダ構成は、以下の通りです。

x:¥pdata￥base
￥global
￥pg_clog
￥pg_xlog

③PowerGres の起動方法

PowerGres のサーバマネージャーを起動します。以下の画面の開始ボタンをクリックしてデータベースを起動します。



状況欄のメッセージが「登録されているサービスは開始中です」と表示されれば、サービスが起動しています。　正常にインストールできたか確認するために、PowerGres のサーバマネージャーを起動します。作成/削除/修正を選択し、コマンドラインツールを起動します。プロンプトで、“psql -U postgres –l”と入力します。以下のようなメッセージが表示され、database name が rfd0 ,Owner が rfdadmin の確認できれば正常に動作しています。

```
x:¥PowerGresDB¥bin>psql -U postgres -l
             List of databases
  Name       | Owner    | Encoding
   ----------- + ---------- + ----------
  rfd0         | rfdadmin | EUC_JP
  template0 | postgres | EUC_JP
  template1 | postgres | EUC_JP
(3 rows)
x:¥PowerGresDB¥bin>
```

道路 GIS のデータベースへの接続設定は以下の通りです。

・ユーザ名//パスワード：rfdadmin/rfdpwd

・データベース名：rfd0

5.3. JDBC ドライバのインストール

サーブレットがデータベースアクセスに利用する JDBC ドライバーをコピーします。PowerGres の製品版 CD－ROM の¥Others¥JDBC フォルダの pg73jdbc3.jar ファイルを C:¥jdbc¥lib フォルダにコピーします。

5.4. 開発用データのインポート

次にアプリケーションで利用するためのデータをツールを使用し、データベースへインポートします。各ツールで登録可能なデータは以下の通りです。

〇道路基盤データ 0.5 版

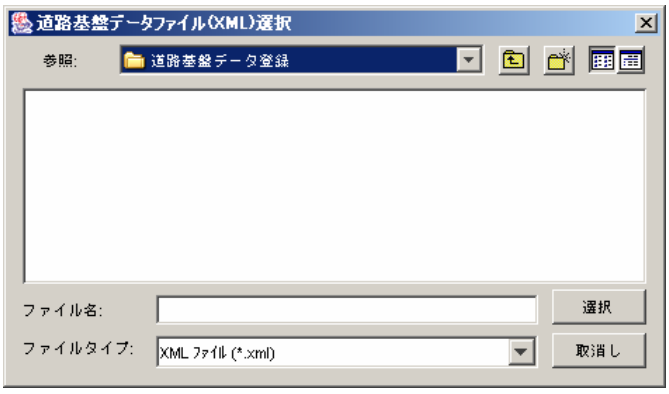
本アプリケーションは、道路基盤データ 0.5 版のデータを道路 GIS のデータベースに登録するためのツールです。道路基盤データのエラーをチェックするためのアプリケーションではありません。したがって、対象の道路基盤データは、0.5 版の論理検査が済んだエラーのないファイルを想定しています。DTD とのチェックは行わないため、登録対象の XML ファイルの文頭に DOCTYPE 宣言のタグがある場合、エディタで削除しておきます。

①環境設定

インストール CD の¥データ登録ツール¥道路基盤データ 0.5 版 DB 登録フォルダを任意のパスにコピーして使用します。道路基盤データ０．５版ＤＢ登録.bat の 1 行目、2 行目は、コピーしたパスに編集します。また 3 行目の JAVA 実行環境の設定を使用の環境に合わせて編集します。メモリの設定は、登録するファイルのサイズに２００MB 程度加えた値を設定することを推奨します。

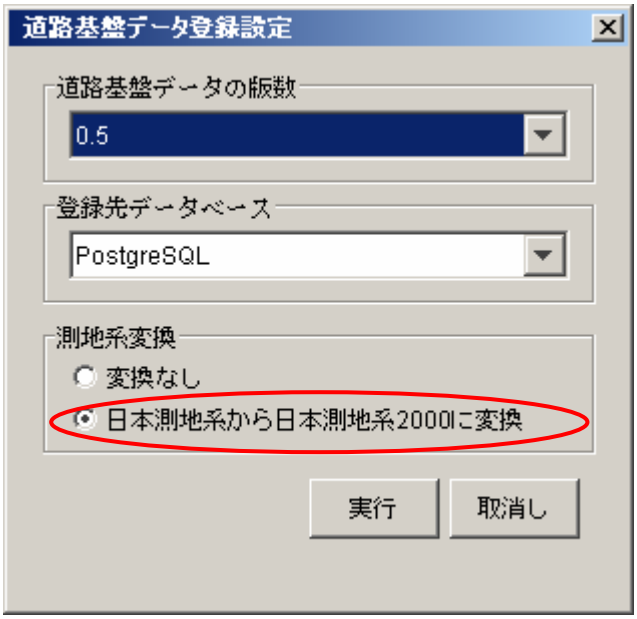
②使用方法

道路基盤データＤＢ登録.bat をダブルクリックしてアプリケーションを起動します。以下の画面で登録する道路基盤データの XML ファイルを選択します。



次に以下の設定画面が表示されます。座標系が、日本測地系のデータを登録する場合、日本測地系 2000 に変換する必要があります。以下のようにオプションの「日本測地系から日本測地系 2000 に変換」

を選択して実行ボタンをクリックしてください。



実行ボタンをクリックすると道路基盤データがデータベースに登録されます。
注）ファイルサイズにより、かなり時間を要する場合があります。

6. サーブレットコンテナのインストール

6.1. インストール

開発用サーバで利用するサーブレットコンテナは、Tomcat です。インストール CD の Tomcat フォルダにある jakarta-tomcat-3.2.4.zip を解凍します。解凍すると￥jakarta-tomcat-3.2.4 というフォルダができあがりますので、これを C:¥に移動します。
（C:¥以外のパスで利用する場合、後述する設定ファイルの編集が必要です）
道路 GIS のインストール CD から、設定ファイルとサーブレット本体をコピーします。
インストール CD の¥TOMCAT 設定ファイル￥tomcat3.2.4¥bin¥のファイルをすべて、先ほどの C:¥jakarta-tomcat-3.2.4¥bin に上書きします。Tomcat のバージョンやインストールパスが異なる場合、以下のファイルを編集する必要があります。

x:¥jakarta-tomcat-3.2.4¥bin¥startup.bat

x:¥jakarta-tomcat-3.2.4¥bin¥startup.bat の最初に以下の行がありますので、インストールパスに編集します。

c:
cd C:¥jakarta-tomcat-3.2.4¥bin

さらに、次の行に JAVA_HOME の設定行が以下のようにあるので、こちらも Java をインストールした

パスに変更します。

set JAVA_HOME=C:¥j2sdk1.4.2_02　←パスやバージョンが異なる場合、編集する。

また、C:¥jakarta-tomcat-3.2.4¥bin の Tomcat.bat の以下の設定パスを確認し、パスが異なる場合、編集してください。以下の例は、C:¥jdbc¥lib フォルダに pg73jdbc3.jar がある場合です。

set CP=%CP%;C:¥jdbc¥lib¥pg73jdbc3.jar

同様にインストール CD から¥TOMCAT 設定ファイル¥tomcat3.2.4¥conf 以下のファイルを C:¥jakarta-tomcat-3.2.4¥conf に上書きします。
Tomcat のバージョンやインストールパスが異なる場合、以下のファイルを編集する必要があります。

- workers.properties
  workers.tomcat_home=c:¥jakarta-tomcat-3.2.4　←TOMCAT_HOMEのパスを記述
  workers.java_home=C:¥j2sdk1.4.2_02　←インストールしたJavaのディレクトリを記述
- workers.properties
  wrapper.tomcat_home= c:¥jakarta-tomcat-3.2.4　←TOMCAT_HOMEのパスを記述
  wrapper.java_home= C:¥j2sdk1.4.2_02　←インストールしたJavaのディレクトリを記述

同様にインストール CD から¥TOMCAT 設定ファイル¥tomcat3.2.4¥lib を C:¥jakarta-tomcat-3.2.4¥lib に上書きします。

6.2. IIS との連携

IIS と TOMCAT を連携させるには、ISAPI の設定をします。C:¥jakarta-tomcat-3.2.4¥bin に前項の作業でコピー済みの isapi_redirect.dll があることを確認します。レジストリエディタ（Regedit）をコマンドプロントから実行し、起動します。以下のレジストリキーを登録します。

HKEY_LOCAL_MACHINE¥SOFTWARE¥Apache Software Foundation¥Jakarta Isapi Redirector¥1.0

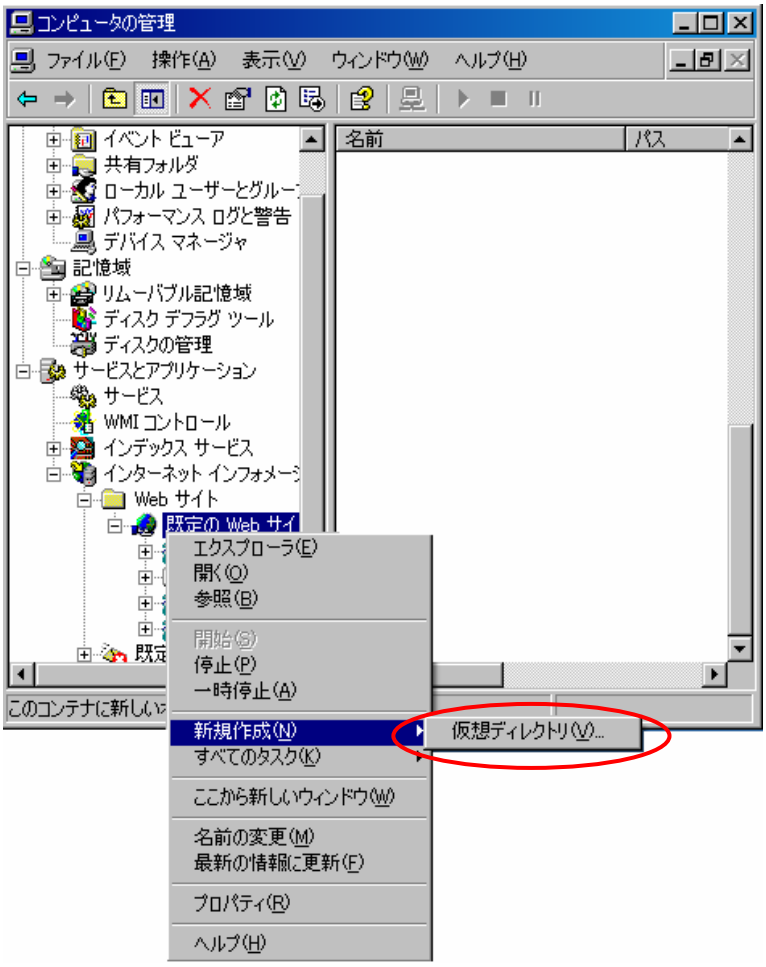
さらに、以下の文字列属性を入力します。

```
extension_uri="/jakarta/isapi_redirect.dll"
log_file=" x:¥jakarta-tomcat-3.2.4¥log¥iis_redirect.log"
log_level="error"
worker_file=" x:¥jakarta-tomcat-3.2.4 ¥conf¥workers.properties"
worker_mount_file=" x:¥jakarta-tomcat-3.2.4 ¥conf¥uriworkermap.properties"
```

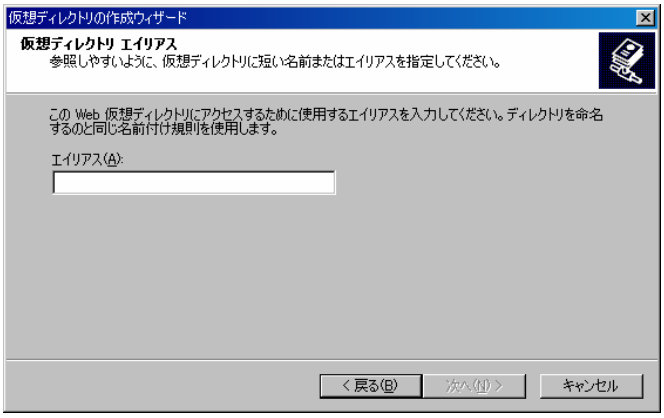
注：x:は、インストールした任意のドライブ名です。

インターネットインフォメーションサービス（IIS）で ISAPI フィルタの設定をします。
管理ツールより「インターネットインフォメーションサービス」を選択します。
以下の仮想ディレクトリを設定します。
仮想ディレクトリ：/jakarta
物理ディレクトリ: X:¥jakarta-tomcat-3.2.4¥bin¥
アプリケーション名：jakarta
実行アクセス権：スクリプトおよび実行可能ファイル
ディレクトリセキュリティ：匿名アクセス有効

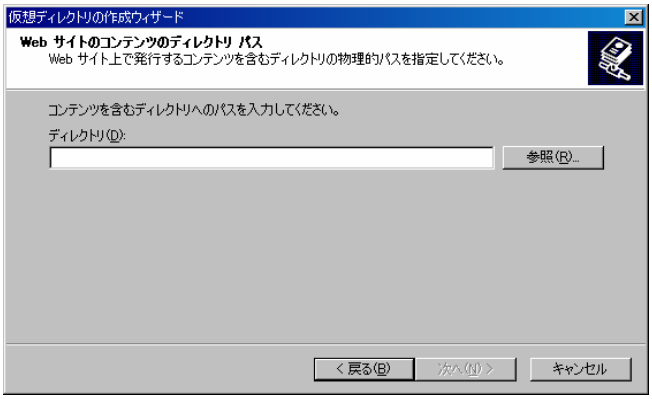
〇仮想ディレクトリの設定方法
マイコンピュータを右クリックし、ショートカットメニューの[管理]を選択します。以下のコンピュータの管理画面が表示されますので、左側のツリーより[サービスとアプリケーション]ー[インタネットインフォメーションサーバ]- [Web サイト]-[既定の Web サイト]を右クリックし、ショートカットメニューを表示します。ショートカットメニューより、[新規作成]-[仮想ディレクトリ]を指定します。



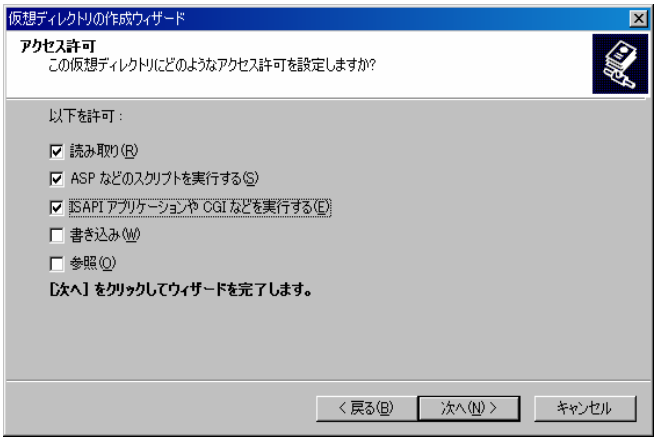
仮想ディレクトリ作成ウィザードが起動しますので、＜次へ＞ボタンをクリックします。
以下の画面が表示されます。エイリアスに仮想ディレクトリの名前を入力してください。1 文字目に “/” は、必要ありません。



次に仮想ディレクトリに設定する物理フォルダを指定します。＜参照＞ボタンをクリックし、仮想ディレクトリに設定するフォルダを設定します。＜次へ＞ボタンをクリックしてください。

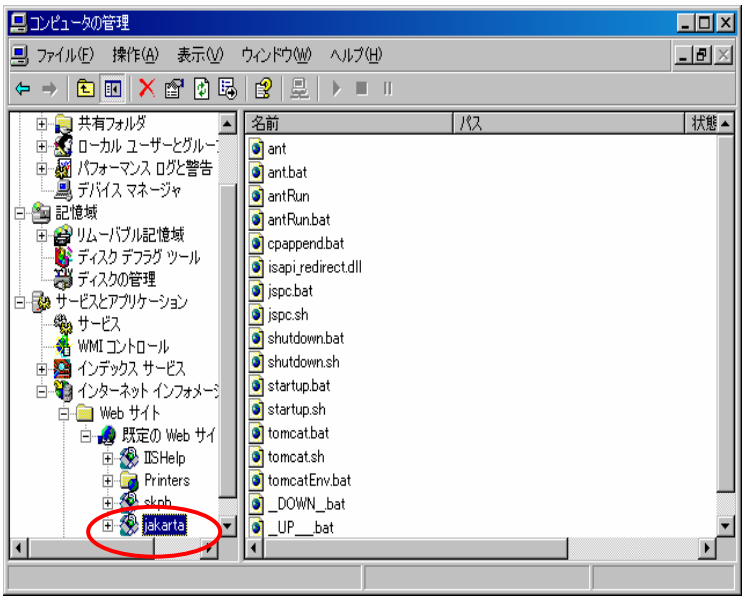


つぎにアクセス権を以下の画面で設定します。セキュリティを考慮し、適切なアクセス権を設定します。Jakarta のアクセス権は、ISAPI アプリケーションや CGI を実行する“にチェックが必要です。



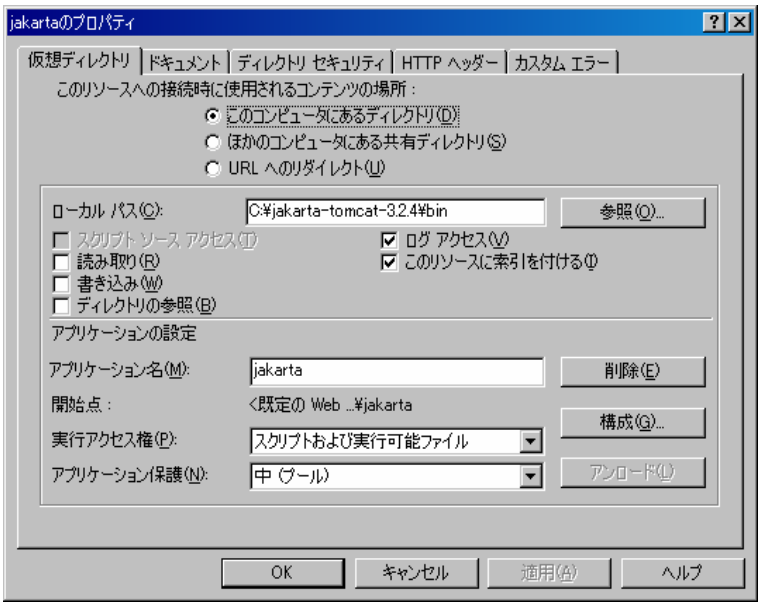
以上で仮想ディレクトリのアクセス権の設定が完了しました。

正しく設定できているか次の手順で確認します。コンピュータの管理から、左側のツリーのインターネットインフォメーションサービスの仮想ディレクトリ　Jakarta を右クリックし、ショートカットメニューより[プロパティ]を選択します。

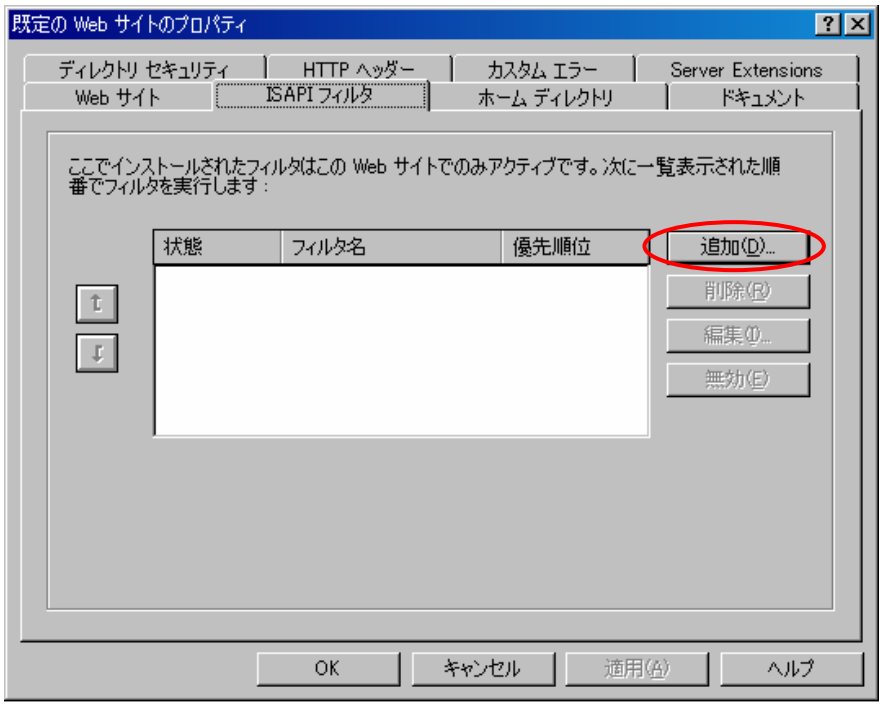


仮想ディレクトリ Jakarta のプロパティを表示します。以下のように仮想ディレクトリが設定されてるか確認します。

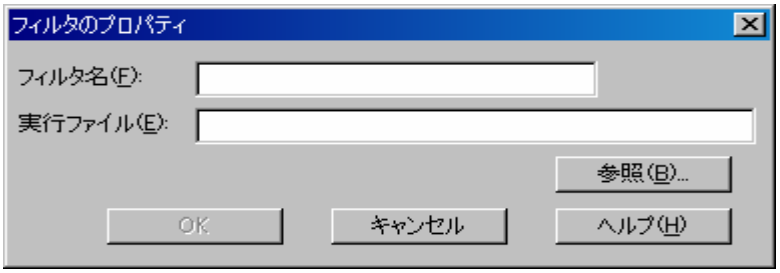
ローカルパス（物理ディレクトリ）：X:¥jakarta-tomcat-3.2.4¥bin¥
アプリケーション名：jakarta
実行アクセス権：スクリプトおよび実行可能ファイル
ディレクトリセキュリティ：匿名アクセス有効



次に ISAPI フィルタの設定をします。既定の Web サイトのプロパティを表示します。プロパティウインドウの ISAPI フィルタタブを選択します。



この画面の＜追加＞ボタンをクリックして追加します。以下のフィルタのプロパティウインドウが表示されます。



フィルタ名と実行ファイルは、それぞれ以下のように入力します。

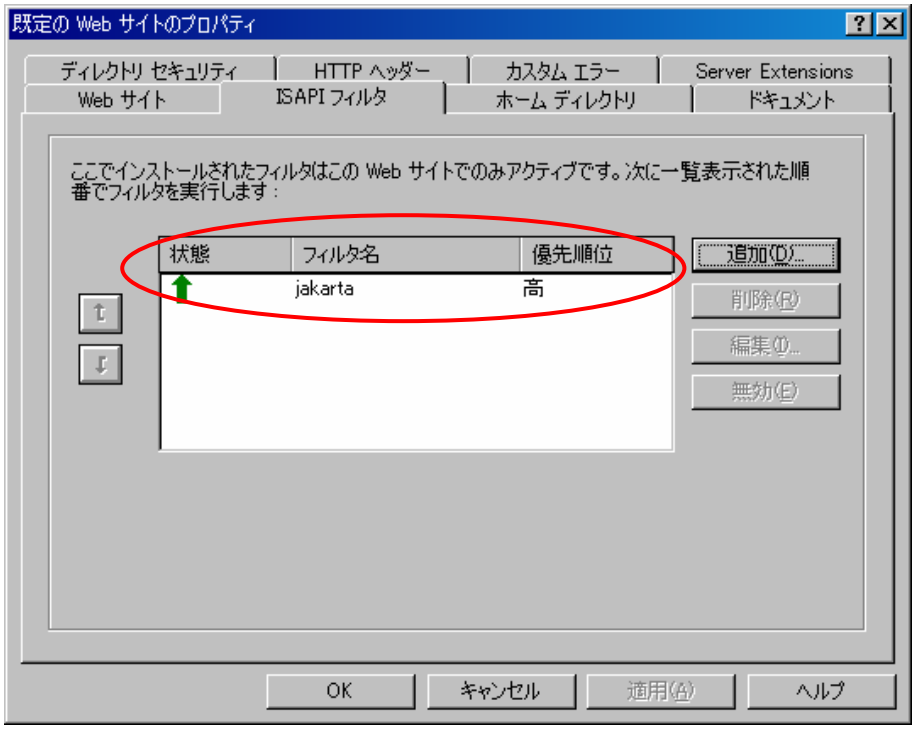
isapi_redirect.dll のバージョン番号はリネームにより削除し、isapi_redirect.dll というファイル名で登録します。

フィルタ名：jakarta

実行ファイル：X:¥jakarta-tomcat-3.2.4¥bin¥isapi_redirect.dll

＜OK＞ボタンをクリックし、このダイアログを閉じます。

既定の Web サイトのプロパティの＜適用＞ボタンを指定後、＜OK＞ボタンをクリックし、このダイアログを閉じます。再度、既定の Web サイトのプロパティを表示します。以下の画面のように ISAPI フィルタの状態を確認します。状態が上向き矢印（↑）であれば、正常に登録されています。状態に下向きの矢印が表示されている場合、正常に登録できていません。登録した内容を修正する必要があります。



【NTFS のディスクで利用される場合の注意事項】

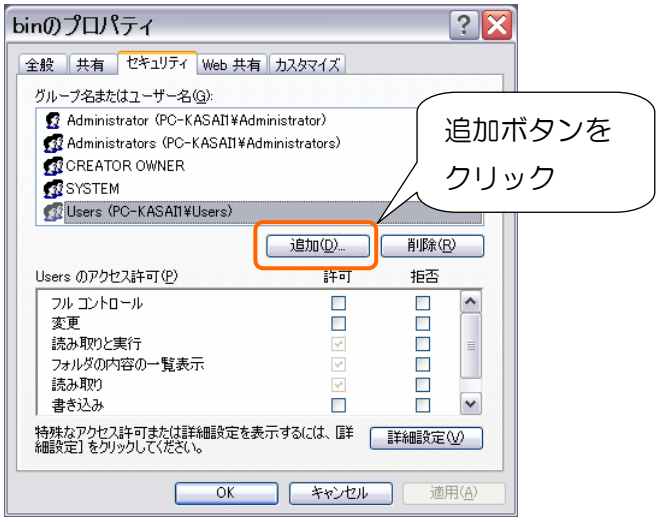
NTFS のディスクで利用される場合には、IIS と Tomcat の連携に必要なディレクトリの実行権限に留意する必要があります。

【設定方法】

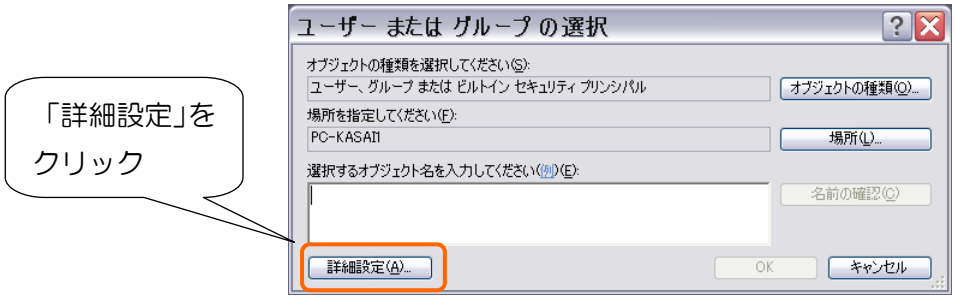
1) エクスプローラ上で、Jakarta tomcat の実行形式を格納した物理ディレクトリのプロパティを確認します。（実行形式のディレクトリ: X:¥jakarta-tomcat-3.2.4¥bin¥）

2) IIS からの実行権限を設定します。

1) 「追加」ボタンをクリックします

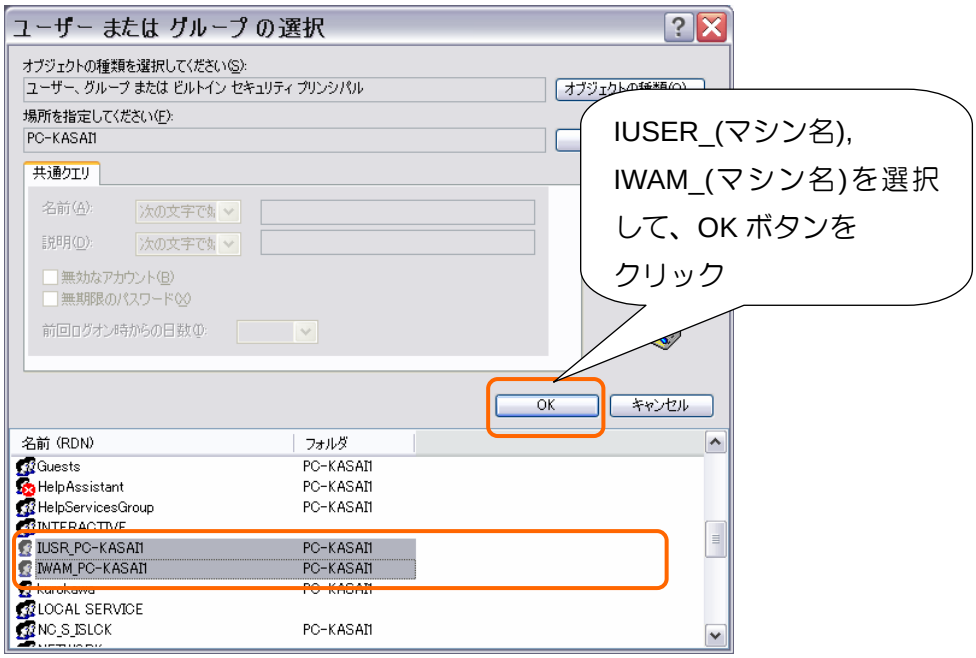


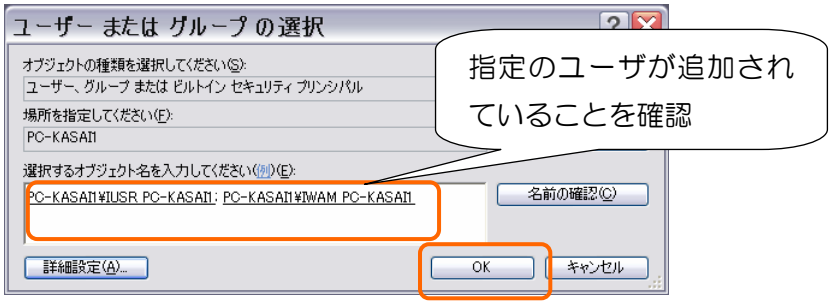
2) 「詳細設定」ボタンをクリックします



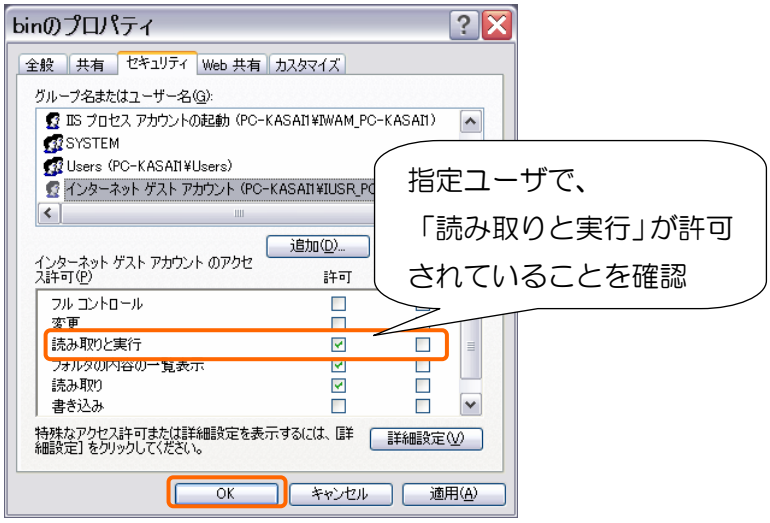
3) 「今すぐ検索」ボタンをクリックします

4) IUSR_(サーバのマシン名)、IWAM_(サーバのマシン名)に追加します。





5) IIS プロセスアカウントの起動（IWAM_(サーバのマシン名)）、およびインターネットゲストアカウント（IUSR_(サーバのマシン名)）に実行権限を付与します。



7. 道路 GIS のインストール

道路 GIS のサーバプログラムおよびデータのインストールを行います。インストール CD の以下のフォルダをファイルごとコピーします。なお、これらのファイルは、いかなる理由があっても道路 GIS のアプリケーションの開発目的以外に使用することはできません。本データを第三者に販売、譲渡その他処分を無断で行うことを禁止します。

￥道路 GIS￥servlet¥rgis----------サーブレット
￥道路 GIS￥WebPages ¥skroot-----HTML
￥道路 GIS¥data¥sttkdat----------画像データ他

￥道路 GIS￥servlet の¥rgis を X:¥jakarta-tomcat-3.2.4¥webapps 配下にコピーします。
コピー後、X:¥jakarta-tomcat-3.2.4¥webapps￥rgis となります。
￥道路 GIS￥WebPages の ¥skroot を任意のパス（c:¥など）にコピーします。
コピー先の¥skroot を仮想ディレクトリ、/skpb に設定します。

￥道路 GIS¥data¥の¥sttkdat を任意のパス（c:¥など）にコピーします。
コピー先が C:¥でない場合、C:¥jakarta-tomcat-3.2.4¥webapps¥rgis¥WEB-INF¥web.xml の以下に示す context-param の値を編集します。

```
<context-param>
  <param-name>PATH_FEATUREXML</param-name>
      <param-value>C:/sttkdat/FeatureList.xml</param-value>
  </context-param>
<context-param>
  <param-name>PATH_STTK_IMAGE</param-name>
      <param-value>C:/sttkdat/images/</param-value>
</context-param>
```

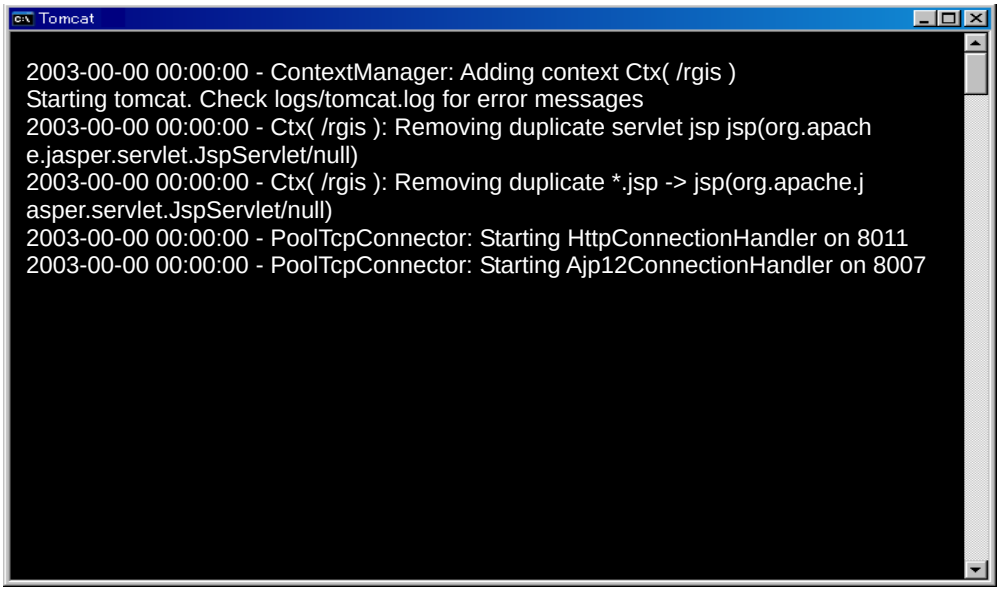
コピーしたパスに編集します。

8. 道路 GIS の動作確認

Web サーバ、データベースが起動していることを確認後、コマンド　プロンプトを起動し、以下のコマンドを入力し、TOMCAT を起動します。

```
>X:
>CD　X:¥jakarta-tomcat-3.2.4¥bin
>startup.bat
```

以下の画面が表示されます。



この画面が表示されれば、TOMCAT が起動しています。設定に誤りがなければ、クライアントの Web ブラウザで,http://<domain>/skpb/index.html から、コンテンツ（「道路 GIS 利用画面」）をたどりアクセス可能です。　停止するには、shutdown.bat を実行します。

9. その他のデータのインポートについて

9.1. DRM データのインポート

データベースに DRM データ（全国デジタル道路地図データベース）をツールによって登録する方法を説明します。このツールにより基本道路、全道路、水系、行政界、鉄道の空間属性が登録されます。

注：全国の DRM をデータベースに登録する場合、数時間かかります。

インストール CD の¥データ登録ツール から ¥DRM データ DB 登録 フォルダをフォルダごと道路 GIS サーバにコピーして使用します。そのときフォルダのプロパティの読み取り専用の属性をはずしておきます。ツールは、データ変換ツールとデータ登録ツールの２構成になっており以下の動作環境で実行する必要があります。ランタイムがない場合、入手してインストールしてください。

ツール名	動作環境
DRM データ変換	Visual Basic6SP5 のランタイム
DRM データ登録	Java ランタイム（推奨バージョン 1.4.2_02）

・コピー後のフォルダ構成

x:¥DRM データ DB 登録¥DRM データ変換

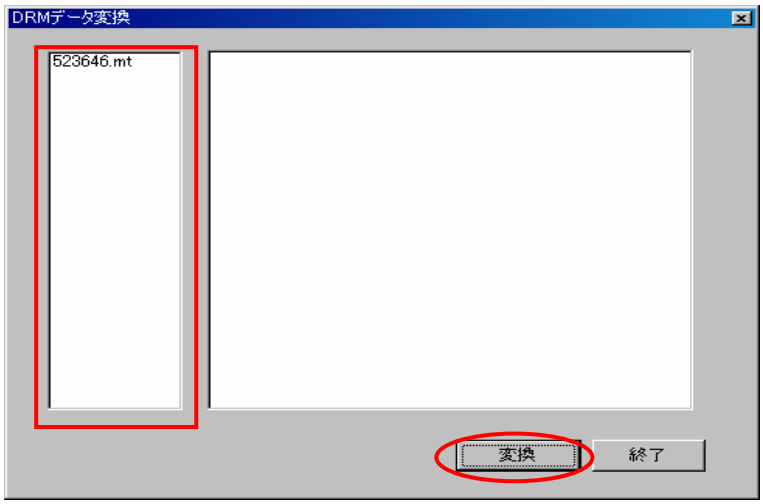
x:¥DRM データ DB 登録¥DRM データ変換¥data

x:¥DRM データ DB 登録¥DRM データ登録

ツールの使用の前に、登録する拡張子（.mt）形式の DRM データを x:¥DRM データ DB 登録¥DRM データ変換¥data にコピーしておきます。

①データ変換ツール

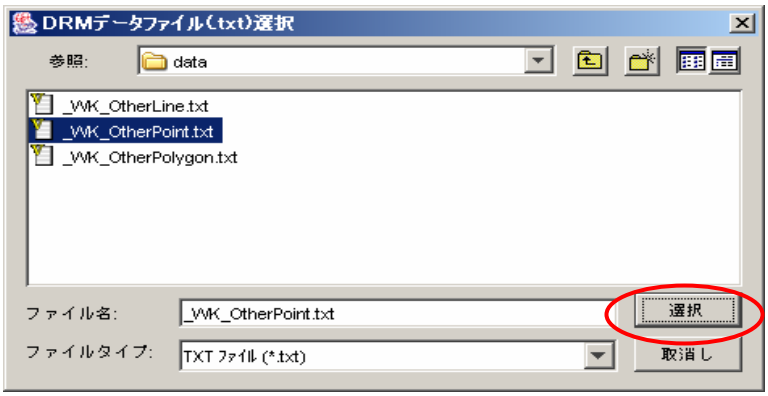
データ変換ツールを起動します。コピー先の x:¥DRM データ DB 登録¥DRM データ変換¥DRM データ変換.exe を起動します。DRM データを左の一覧から変換対照のファイルを任意に選択します。複数選択可能です。選択後、＜変換＞ボタンをクリックすると登録用の中間ファイルを¥DRM 変換¥data フォルダに出力します。



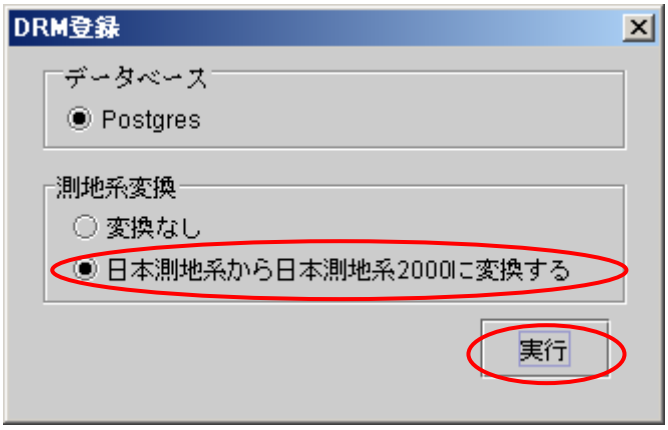
②データ登録ツール

データ登録ツールを起動します。x:¥DRM データ DB 登録¥DRM データ登録¥DRM データ登録.bat に実行パス、Java ランタイムのパスを記述していますので、環境に合わせて編集します。

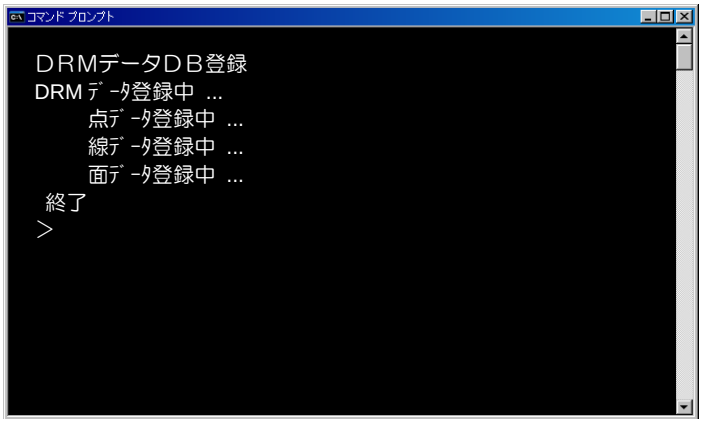
編集が終わりましたら、x:¥DRM データ DB 登録¥DRM データ登録¥DRM データ登録.bat をクリックし起動します。以下の画面が表示されるので、①で作成した中間ファイルを x:¥DRM 変換¥data フォルダから選択します。中間ファイルは 3 種類表示されますが、どれか 1 つを選択するとすべてのファイルが処理されます。



次に以下の画面が表示されるので、「日本測地系から日本測地系 2000 に変換する」を選択した状態で、＜実行＞ボタンをクリックします。



コマンドプロンプトに以下の画面のように登録処理状況が表示されます。終了が表示されると登録処理が完了です。



③描画法

描画法はデータベースの WK_Portraiture テーブルに登録されている情報が使用されます。x:¥DRM データ DB 登録¥DRM_Portraiture.csv を WK__Poratraiture テーブルに登録してください。WK__Poratraiture テーブルの詳細は道路 GIS SDK データベース設計書を参照願います。

③Feature.XML の更新

7 章の道路 GIS サーバインストールで使用した X:/sttkdat/FeatureList.xml に DRM について追記が必要になります。FeatureList.xml の<FeatureList>の子ノードに以下の XML を追加します。

FeatureList.xml に DRM に関して追記済みのデータファイルを x:¥DRM データ DB 登録

¥FeatureList.xml に用意してあります。

```
<GroupOfDigitalRoadMap>
        <Annotation>デジタル道路地図</Annotation>
        <BasicRoadDRM>
                <Annotation>基本道路</Annotation>
        </BasicRoadDRM>
        <AllRoadDRM>
                <Annotation>全道路</Annotation>
        </AllRoadDRM>
        <GroupOfBackGroundDRM>
                <Annotation>その他データ</Annotation>
                <WaterSystemPolygonDRM>
                        <Annotation>水系(面)</Annotation>
                </WaterSystemPolygonDRM>
                <WaterSystemLineDRM>
                        <Annotation>水系(線)</Annotation>
                </WaterSystemLineDRM>
                <DomainDRM>
                        <Annotation>行政界</Annotation>
                </DomainDRM>
                <RailWayDRM>
                        <Annotation>鉄道</Annotation>
                </RailWayDRM>
        </GroupOfBackGroundDRM>
</GroupOfDigitalRoadMap>
```

9.2. ラスタデータの登録

9.2.1. インデックスマップの登録

インデックスマップは、ラスタファイルを使用しています。インデックスマップの登録をするには、IndexMap を登録するには、512*512 ピクセル以下の PNG 形式、または JPEG 形式の画像ファイルを作成します。任意のファイル名で作成後、以下のインストール済みフォルダにコピーします。

x:¥sttkdat¥image¥indexmap

コピーしたファイルの情報を以下のサンプルのように WK_Referenceraster テーブルに登録します。

表　WK_Referenceraster テーブル登録サンプル

項番	名称	列名	値
1	ﾗｽﾀ種別名	RasterName	IndexMap
2	ﾗｽﾀﾌｧｲﾙ名	FileName	indexmap/indexmapnagaoka.jpg
3	最小経度座標	minX	138.3487508
4	最小緯度座標	minY	36.63823001
5	最大経度座標	maxX	139.4012492
6	最大緯度座標	maxY	37.77843666
7	ﾀｲﾑｽﾀﾝﾌﾟ	WKtimestamp	2003-10-01 00:00:00
8	地域名	Location	NAGAOKA

ラスタ種別名：索引図であることを示すため、固定で“IndexMap”とします。

ラスタファイル名：x:¥sttkdat¥image　フォルダ以下のパスを記述します。

サンプルは, x:¥sttkdat¥image¥Indexmap¥indexmapnagaoka.jpg ファイルの登録例です

タイムスタンプ：wktimestamp 列にはデータ登録日時を記述します。

地域名：　WK_Referenceraster テーブルのの地域名は索引図の場合のみ情報を登録します。地域名を登録する場合、WK_Eextent に登録されている名前である必要があります。

WK_Extent に登録されいない場合、こちらのテーブルにも情報を登録します。

詳細は、道路 GIS　SDK　データベース設計書の WK_Extent テーブルの項を参照願います。

9.2.2.　数値地図の登録

背景に数値地図 25000 のラスタデータを登録するには、数値地図 25000 を加工する必要があります。

○数値地図 25000 の事前処理

地図の外側の枠を除去します。除去は任意のソフトで行ってください。

ファイル：2 次メッシュ番号.PNG という名前で、８BIT（256 色）の PNG ファイルにします。

ファイルサイズ：1 ファイルサイズは、大きくても 2MB 程度とし、推奨 1 MB 以下とします。

ファイルサイズを考慮し画像のリサイズを行います。最大　5250*3500　とし、用途やパフォーマンスを考慮してサイズを決めます。

処理済みのファイルを以下のフォルダに*.PNG ファイルをコピーします。

X:¥sttkdat¥image¥DigitalMap25000

数値地図 25000 は、日本測地系で作成されていますが、道路 GIS では日本測地系 2000 としてマッピングされます。WK_Referranceraster テーブルに数値地図 25000 については既定の 2 次メッシュファイルに関する情報が使用されます。 描画法はデータベースの WK_Portraiture テーブルに登録されている情報が使用されます。x:¥データ登録ツール¥数値地図 25000 登録¥DigitalMap25000_Portraiture.csv を WK__Poratraiture テーブルに登録してください。

また、FeatureList.xml に以下の XML を追記する必要があります。また、7 章の道路 GIS サーバインストールで使用した X:/sttkdat/FeatureList.xml に数値地図 25000 について追記が必要になります。FeatureList.xml の<FeatureList>の子ノードに以下の XML を追加します。

FeatureList.xmlに数値地図2500関して追記する分のみ書き出したデータファイルをx:¥データ登録ツール¥数値地図 25000 登録¥ DigitalMap25000_FeatureList.xml に用意してります。

```
<GroupOfNationalDigitalMap>
        <Annotation>数値地図</Annotation>
        <DigitalMap25000>
                <Annotation>数値地図 25000（地図画像）</Annotation>
        </DigitalMap25000>
</GroupOfNationalDigitalMap>
```

（解説）

<FeatureList>タグの子ノードには、GroupOf という接頭語をつけたパッケージ名でタグを作成します。子ノードの<Anontation>タグは、パッケージの日本語名称を記述します。
次に地物タグ（クラス名）を記述します。子ノードの<Annotation>タグは、この地物の日本語名称を記述します。

9.2.3. その他のラスタデータの登録

その他のラスタデータを登録するには、数値地図 25000 の例を参考に事前処理を行い登録します。

〇その他のラスタファイルの事前処理

座標系：日本測地系 2000 とします。平面直角座標系で作成されているラスタファイルは日本測地系 2000 に変換する必要があります。

ファイル：“任意の管理名”+.PNG という名前で、８BIT（256 色）の PNG ファイルにします。2 次メッシュの単位のファイルにする必要はありません。ファイルサイズは、1 ファイル大きくても 2MB 程度とし、推奨１MB 以下とします。ファイルサイズを考慮し画像のリサイズを行い

ます。最大　5250*3500　とし、用途やパフォーマンスを考慮してサイズを決めます。処理が完了したら以下のフォルダに*.PNG ファイルをコピーします。

C:¥sttkdat¥image¥（管理フォルダ名）

管理フォルダ名は、他のラスタデータと混在しないように任意に決めます。規定のラスタ種別であれば、管理フォルダ名はラスタ種別名にすることを推奨します。
既定のラスタ種別名は、以下のとおりです。

“AeroPhoto” ： 航空写真
“DigitalMap200000”: 数値地図 200000
“DigitalMap25000”: 数値地図 25000
“SatellitePhoto” ： 衛星写真

数値地図 25000 の例に従いデータベースの WK_Referenceraster テーブル、WK_Portraiture にラスタデータの情報を登録します。テーブルの詳細については、道路 GIS　SDK　データベース設計書を参照願います。また、FeatureList.xml の内容を編集します。
インストール CD の￥データ登録ツール¥その他ラスタ登録　フォルダに以下のサンプルデータがあります。

・FeatureList.xml サンプルファイル（FeatureList.xml）
・WK_Referenceraster テーブル登録データサンプル（WK_Referenceraste.csv）
・WK_Portraiture テーブル登録データサンプル（WK_Portraiture.csv）

10.　TOMCAT のライセンスについて

道路 GIS は、サーブレットコンテナに TOMCAT を使用しています。インストール CD の￥tomcat フォルダに TOMCAT のバイナリーパッケージを同梱しています。ライセンスについては、
以下のとおりです。

この製品/商品には、Apache Software Foundation (http://www.apache.org/)において開発されたソフトウェアが含まれます。Tomcat の全てのソースコード・ドキュメントは、Apache Software Foundation が著作権を有しており、以下のライセンス規約により利用可能です。

Apache Software License

==========

The Apache Software License, Version 1.1

Copyright (c) 2001 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:
"This product includes software developed by the Apache Software Foundation (http://www.apache.org/)."
Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" and "Apache Turbine" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", "Apache Turbine", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==========

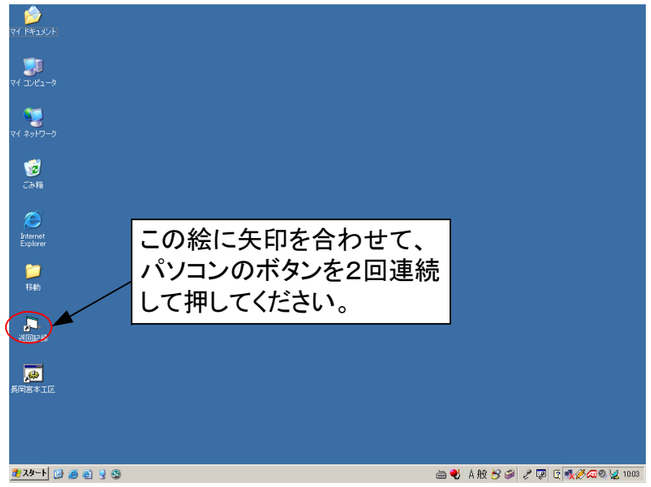
This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see http://www.apache.org.

音声認識発音一覧

天　候	はれ 晴	くもり 曇り	あめ 雨	ゆき 雪	
路面状況	かんそー 乾燥	とーけつ 凍結	ざくれ ザクレ	しゃーべっと シャーベット	ぬれ 濡れ
交通状況	りょーこー 良好	じゅーたい 渋滞			
消雪パイプ	てーし 停止	かどー 稼動			
雪種別	しんせつ 新雪	あっせつ 圧雪			
積　雪	なし なし	いっせんち 1cm	にせんち 2cm	さんせんち 3cm	よんせんち 4cm
	ごせんち 5cm以上				

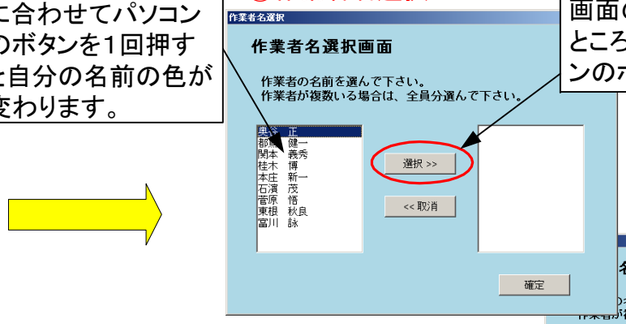
巡回記録、自動散布システム簡易説明書

①システム起動

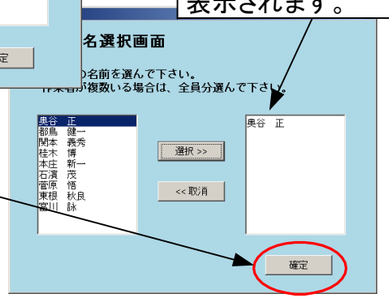


②作業者名選択

矢印を自分の名前に合わせてパソコンのボタンを1回押すと自分の名前の色が変わります。



自分の名前の色が変わったら、画面の「選択＞＞」という文字のところに矢印を移動して、パソコンのボタンを1回押してください。

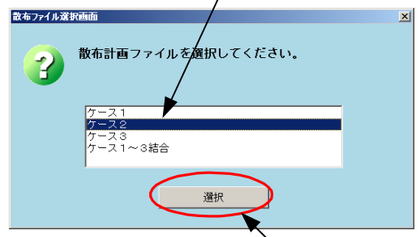


正しく自分の名前が選ばれた場合には、画面の右側に自分が表示されます。

自分の名前が表示されたら画面右下の「確定」という文字に矢印を合わせて、パソコンのボタンを1回押してください。成功すると次の画面が現れます。

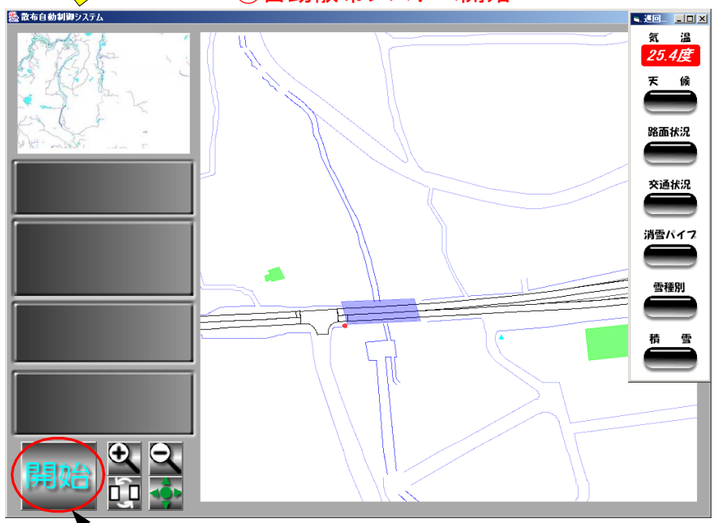
③散布計画選択

現場代理人の方から聞いている文字の部分（散布のパターン名）に矢印を合わせて、パソコンのボタンを1回押してください。文字の色が変わります。

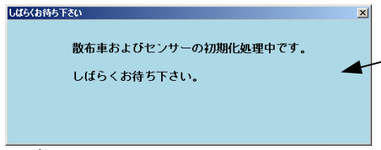


選らんだ文字に間違いがなければ、画面の「選択」に矢印を合わせてパソコンのボタンを1回押してください。成功すると次の画面が現れます。

④自動散布システム開始



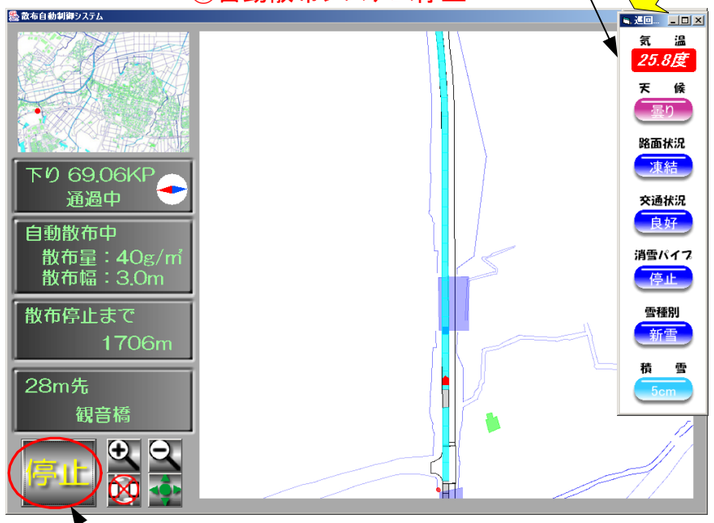
画面の「開始」という文字に矢印を合わせてパソコンのボタンを1回押してください。しばらく、画面の真ん中に次の画面が表示されます。



この画面が表示されている間は散布車を走らせないでください。この画面が消えたら散布車の出発準備完了です。

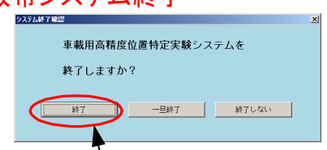
⑤自動散布システム停止

巡回地点に来たら、マイクで巡回地点の状況をしゃべってください。正しく状況が認識された場合は、音声で案内されます。



散布が終わりステーションに到着したら、画面の「停止」の文字に矢印を合わせて、パソコンのボタンを1回押してください。次の画面が表示されます。

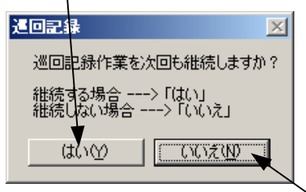
⑥自動散布システム終了



画面の「終了」に矢印を合わせて、パソコンのボタンを押してください。しばらく（約5秒）してから次の画面が表示されます。

⑦システム終了

まだ巡回地点が残っていて巡回する場合は「はい」に矢印を合わせて、パソコンのボタンを押してください。



全ての巡回地点の巡回が終わった場合は「いいえ」に矢印を合わせて、パソコンのボタンを押してください。

計画支援システム　取扱説明書

1．できる機能
2．URL
3．ログイン
4．通常メニュー
　4－1．散布計画表示
　4－2．散布履歴表示
　4－3．巡回記録表示
　4－4．散布履歴登録
　4－5．巡回記録登録
5．管理者メニュー
　5－1．散布計画作成
　5－2．作業者名リスト作成
6．地図表示の使い方

できる機能

計画支援システムでは、Webブラウザを使用して以下のことが行えます。

通常メニュー

- 散布計画の確認と散布計画ファイルのダウンロード
- 散布履歴の地図表示および散布結果の詳細表示
- 巡回記録Excelファイルの表示（ダウンロード）
- 散布結果のサーバーへの登録
- 巡回記録のサーバーへの登録

管理者メニュー

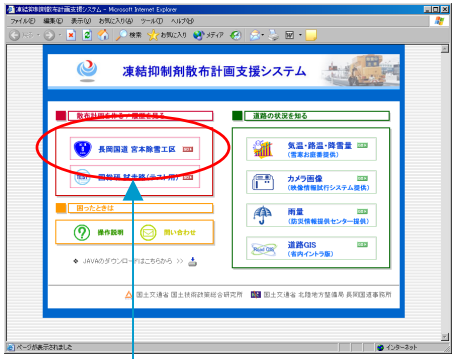
- 散布計画データの作成、修正、確認
- 作業員リストファイルの作成、修正

散布結果と巡回記録のサーバーへの登録は、散布及び巡回を行った場合、必ず実施してください。

URL

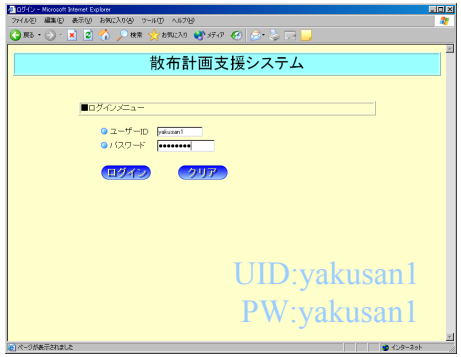
散布支援システムを利用するには、Webブラウザ（Internet Explorer）を起動して、以下のURLへアクセスしてください。

http://10.98.2.66/

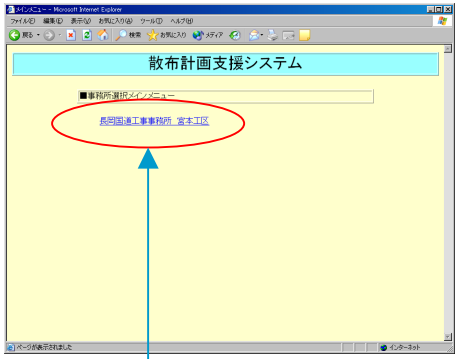


ここをクリックして、ログイン画面へ

ログイン

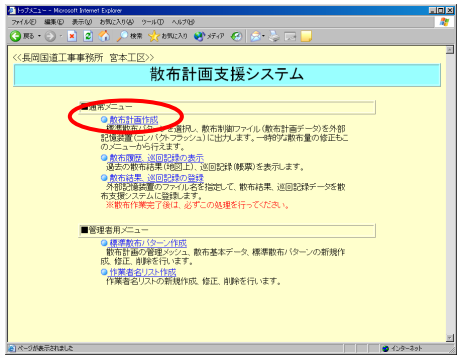


ログインに成功すると、次の画面が現れます

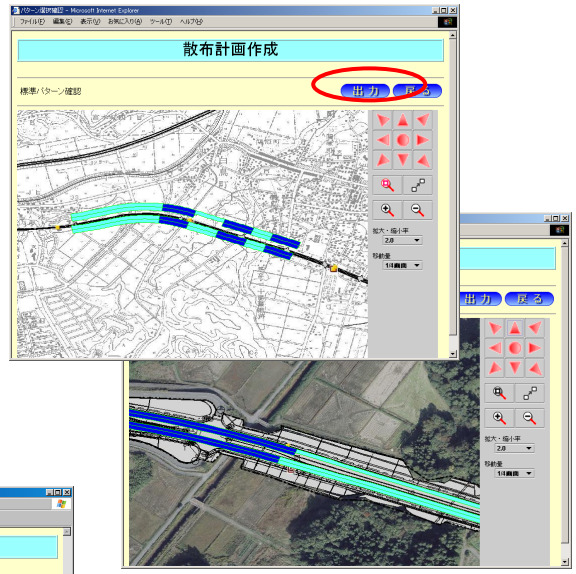


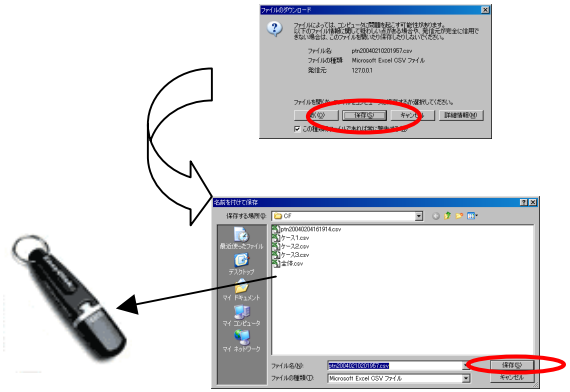
ここをクリックして、次画面へ進んでください

散布計画の確認と散布計画ファイルのダウンロード



ダウンロードするパターンを選択

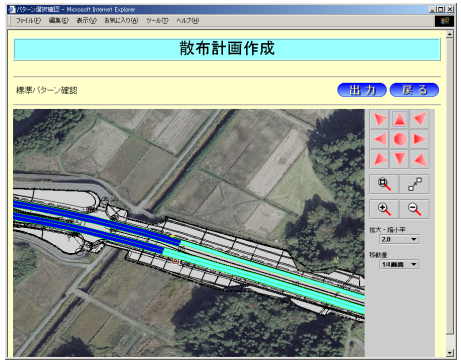




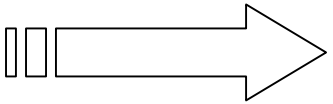
USBメモリに、ファイル名を変更せずに保存してください。ファイル名は「ptn20040216133726.csv」のようにptn日時.csvで作成されます。

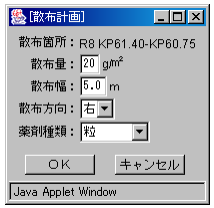
散布計画の一時変更

標準散布パターンの散布量や散布幅を一時的に変更して、散布計画ファイルとしてダウンロードすることが可能です。ここでの変更は、サーバー側の標準パターンデータを変更するものではありません。あくまでダウンロードする散布計画ファイルを修正するためのものです。



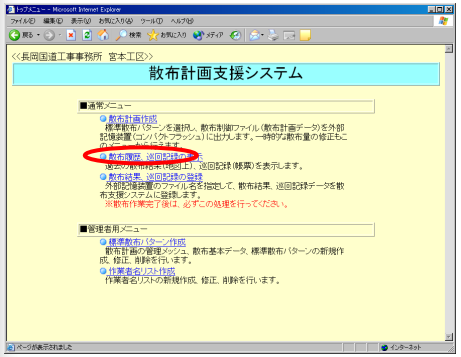
変更するメッシュをクリックして、このダイアログを表示してください。

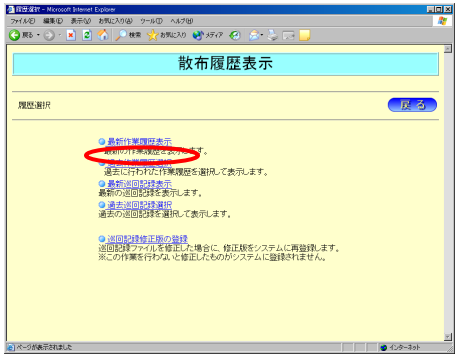




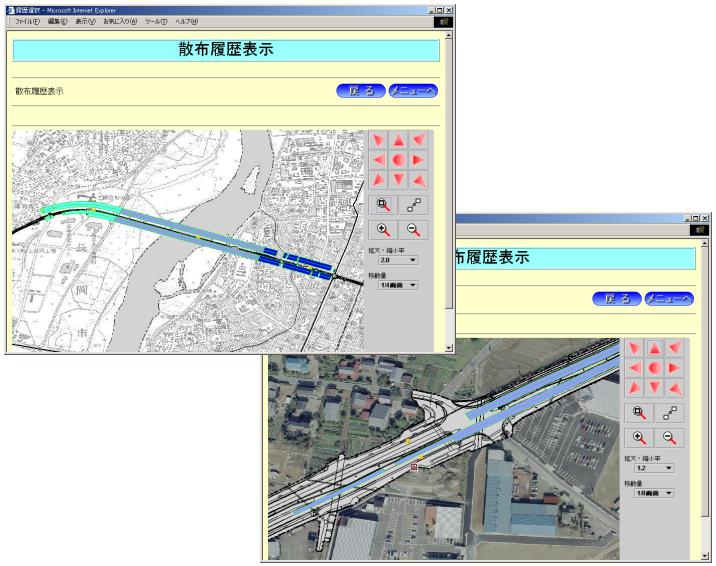
値を変更して「OK」ボタンを押すと、出力される散布計画ファイルの指定した散布メッシュの散布条件が変更されます。

散布履歴の地図表示





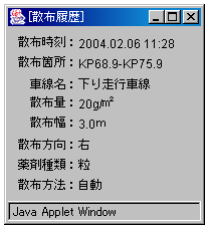
表示する履歴を選択



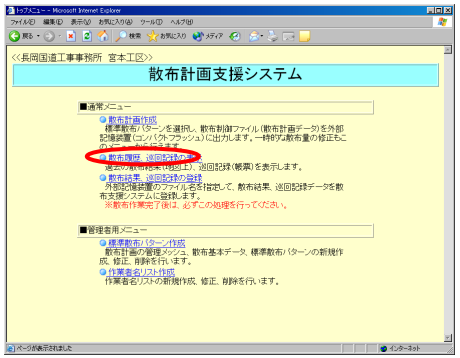
散布結果の詳細表示

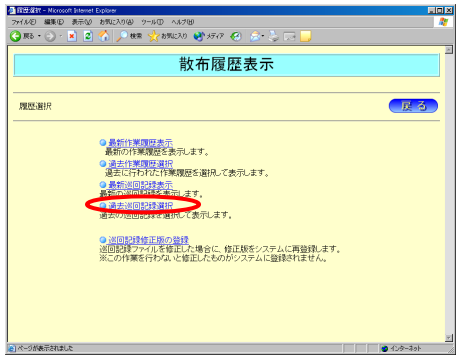
散布時刻や散布量、散布幅を確認したい場合は、地図上の散布メッシュをクリックすることによって確認できます。

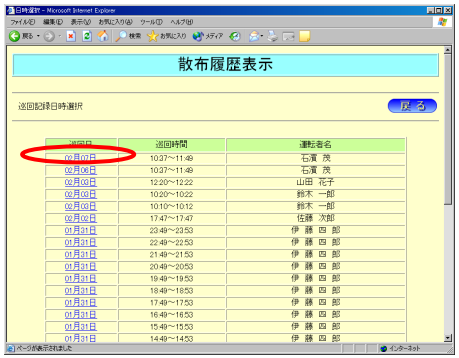
散布結果を見たいメッシュをクリックして、このダイアログを表示してください。



巡回記録Excelファイルの表示



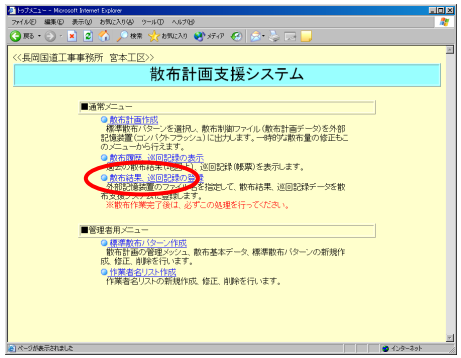


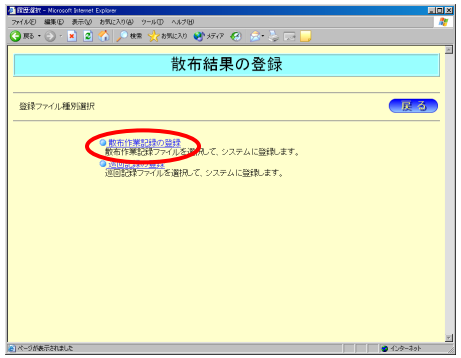


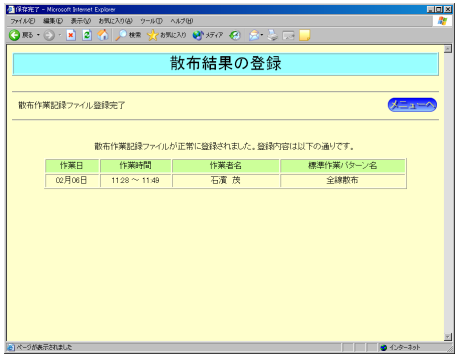
巡回記録はExcelファイルとして表示され印刷も可能です。自分のパソコンに保存して、編集することもできます。
編集したExcelファイルをサーバーに登録しておくことも可能です。

パソコンにExcelがインストールされていない場合は、ファイルを保存するかどうかの確認画面が表示されます。

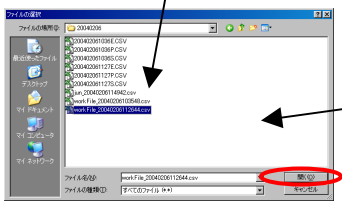
散布結果のサーバーへの登録

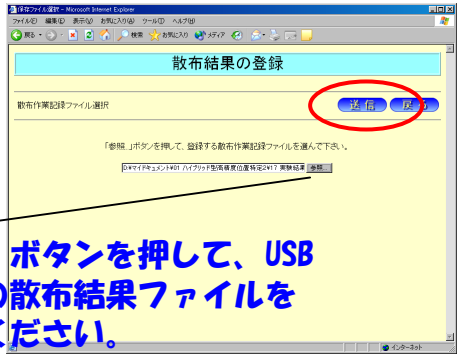






USBメモリには、既に登録済みの散布結果ファイルが残っています。登録が完了したファイルは不要ですので、削除してかまいません。

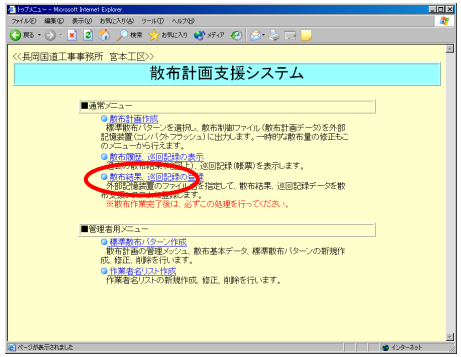


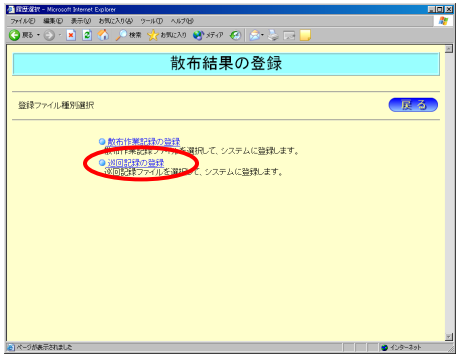


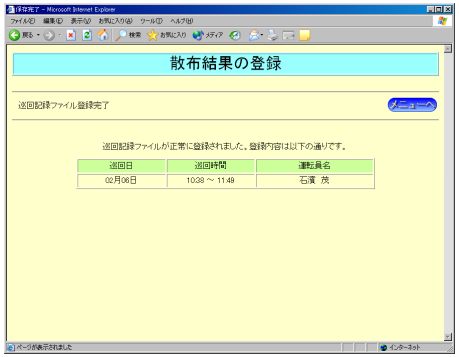
「参照…」ボタンを押して、USBメモリ内の散布結果ファイルを選択してください。

操作支援システムにより作成された「散布結果ファイル」をサーバーへ登録する必要があります。散布が終わったら必ずこの処理を行ってください。散布結果ファイル名は「sanpu20040216133726.txt」のようにsanpu日時.txtでUSBメモリ内に作成されてます。

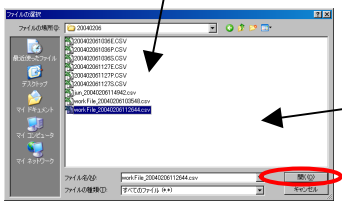
巡回記録のサーバーへの登録

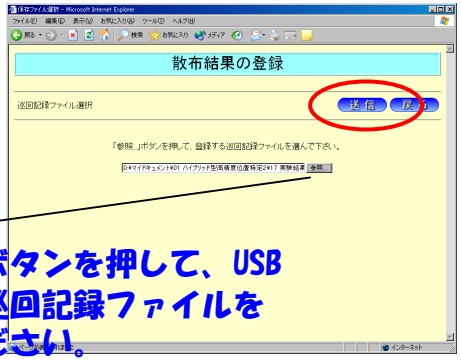






USBメモリには、既に登録済みの巡回記録ファイルが残っています。登録が完了したファイルは不要ですので、削除してかまいません。



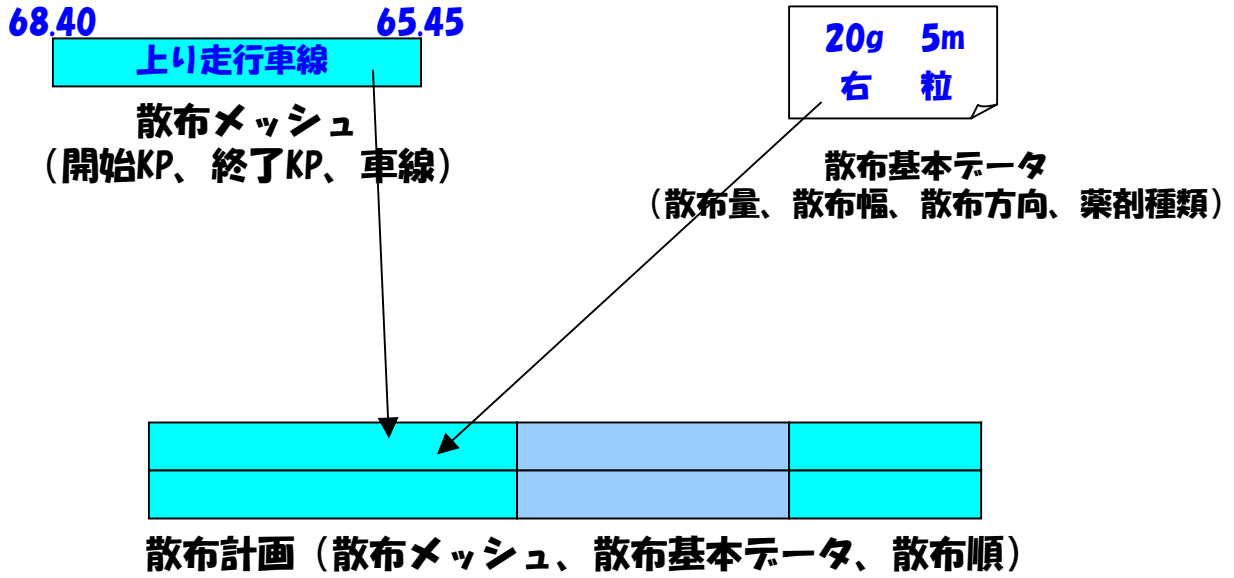


「参照...」ボタンを押して、USBメモリ内の巡回記録ファイルを選択してください。

操作支援システムにより作成された「巡回記録ファイル」をサーバーへ登録する必要があります。散布が終わったら必ずこの処理を行ってください。巡回記録ファイル名は「junkai20040216133726.txt」のようにjunkai日時.txtでUSBメモリ内に作成されてます。

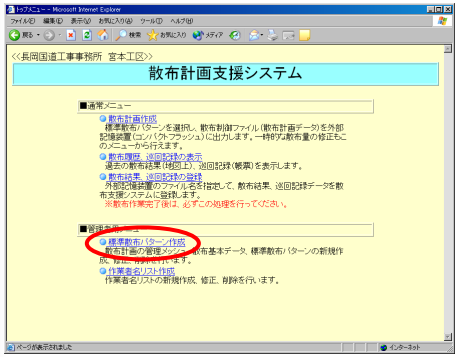
散布計画の作成方法

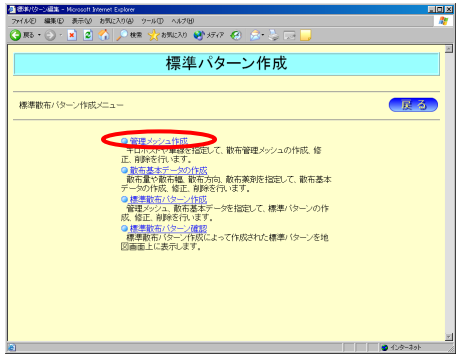
散布計画パターンは、散布メッシュと散布基本データを複数組み合せることにより作成します。

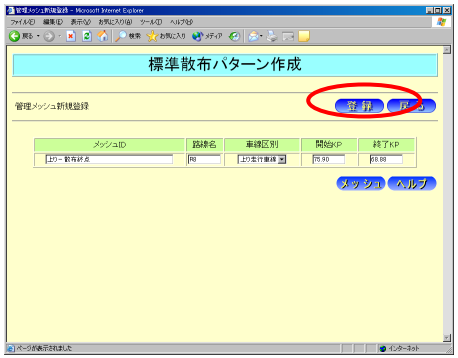


散布量や散布幅を変える必要のKPや車線毎に散布メッシュを作成し、その散布メッシュと散布基本データの組み合わせをさらに複数組み合せることによって、散布計画パターンを作成します。

管理メッシュの作成



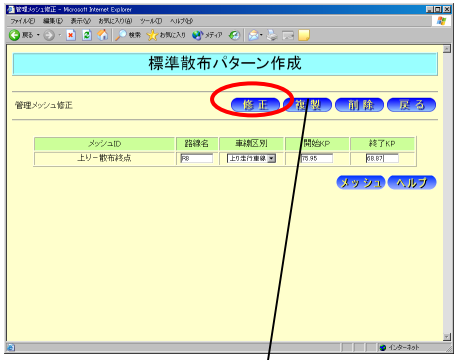




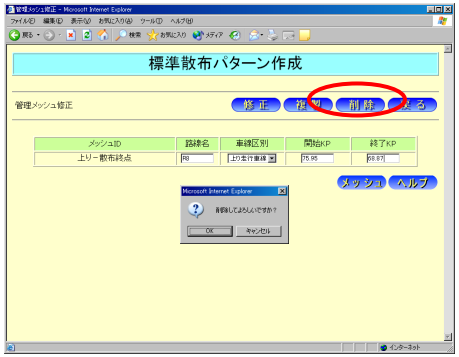
全項目を入力して「登録」ボタンを押してください。入力項目の詳細は「ヘルプ」ボタンを押すと確認できます。また、登録できるKPは「メッシュ」ボタンでExcelファイルが表示されますので、それで確認ください。

散布の開始KP、終了KPは、実際に散布される箇所ではなく、手動散布時に散布制御のスイッチを切替える位置です。オペレーターが散布制御を切替えるタイミングに合わせて作成してください。

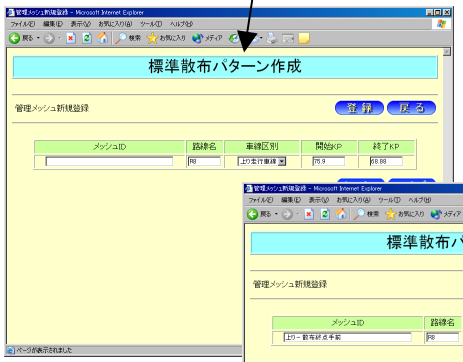
散布メッシュの修正・削除・複製

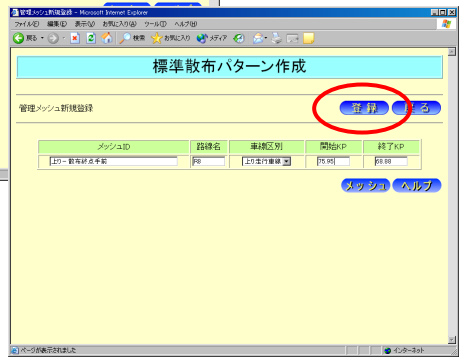


データを修正する場合は、修正項目を直接修正して、「修正」ボタンを押してください。



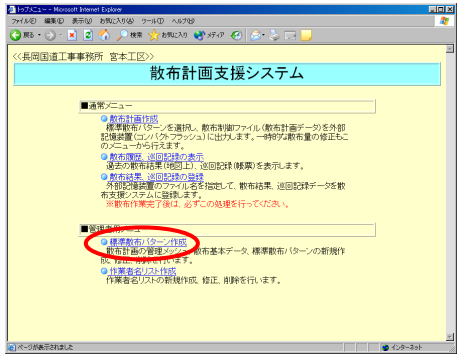
データを削除する場合は、「削除」ボタンを押してください。

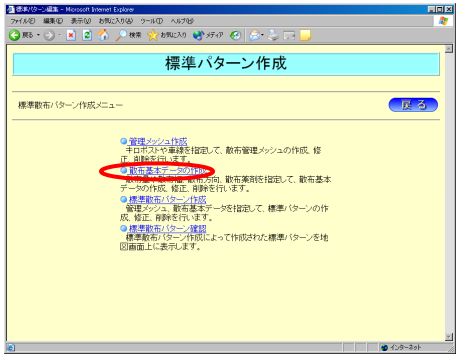


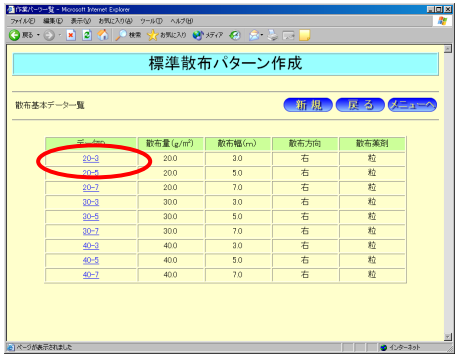


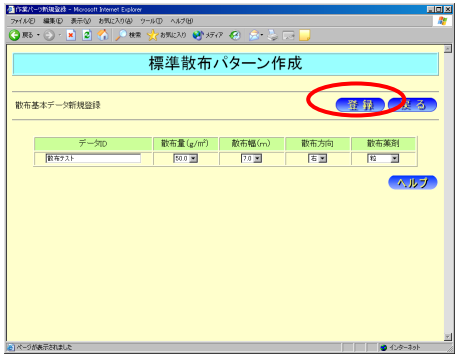
現在表示されているデータの複製を作成して、新たなデータを作成することができます。「複製」ボタンを押してから必要項目を入力して、「登録」ボタンを押してください。

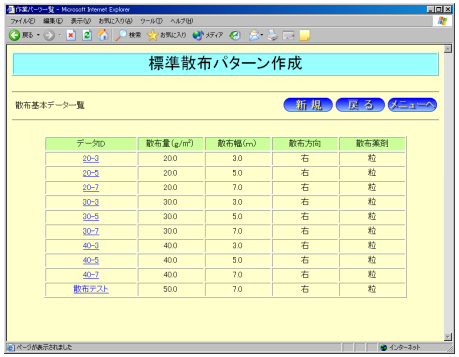
散布基本データの作成





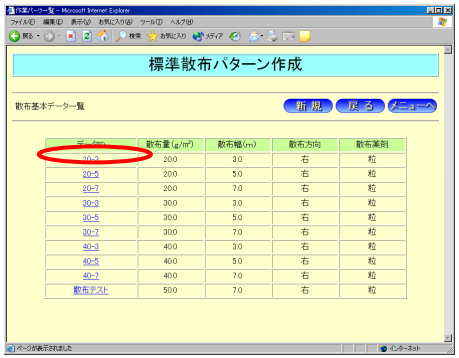


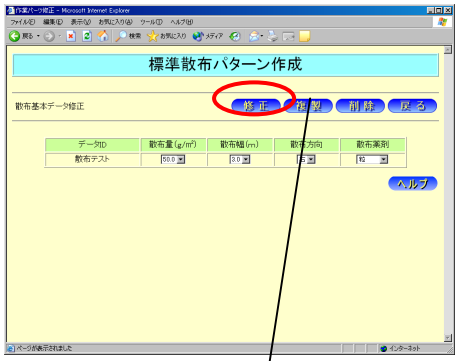




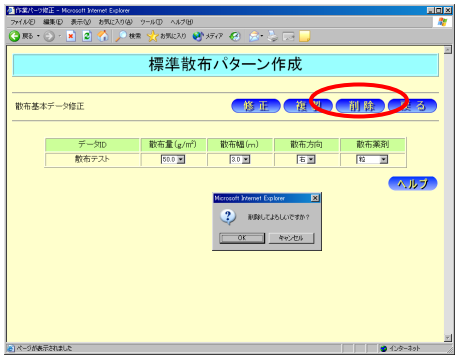
全項目を入力して「登録」ボタンを押してください。入力項目の詳細は「ヘルプ」ボタンを押すと確認できます。

散布基本データの修正・削除・複製

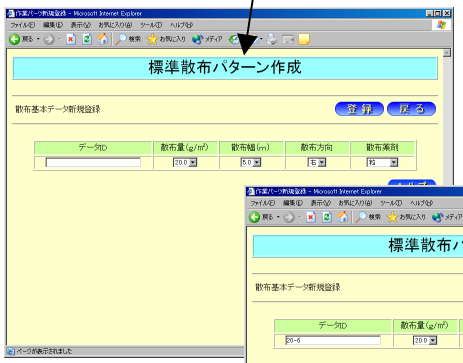




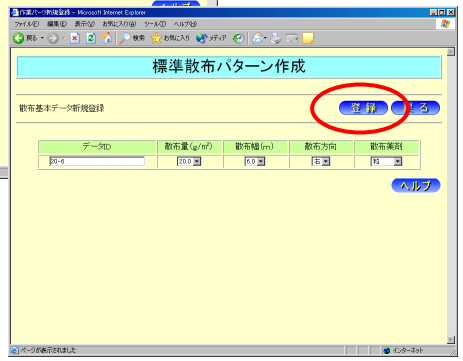
データを修正する場合は、修正項目を直接修正して、「修正」ボタンを押してください。



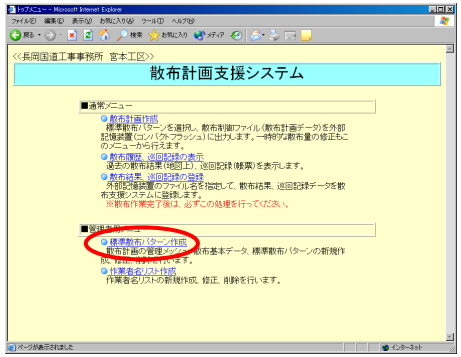
データを削除する場合は、「削除」ボタンを押してください。

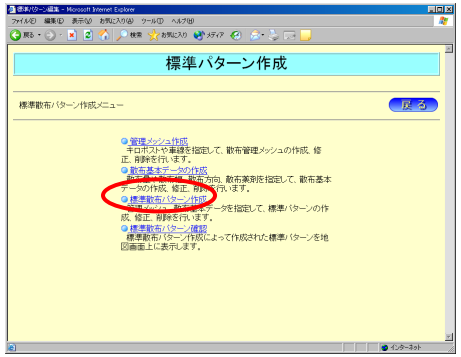


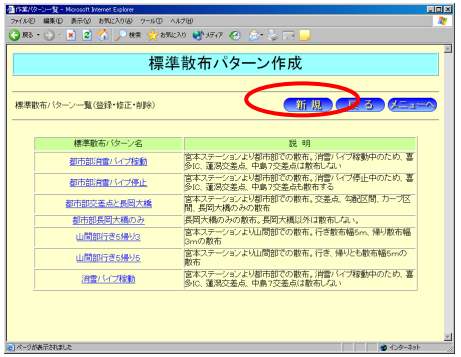
現在表示されているデータの複製を作成して、新たなデータを作成することができます。「複製」ボタンを押してから必要項目を入力して、「登録」ボタンを押してください。

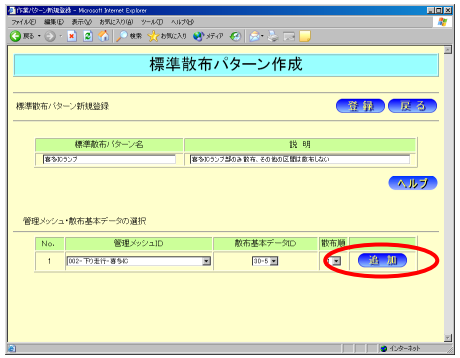


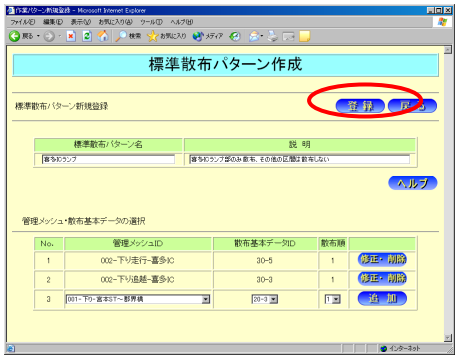
散布計画の作成

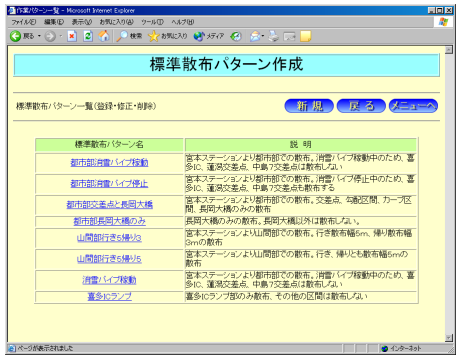








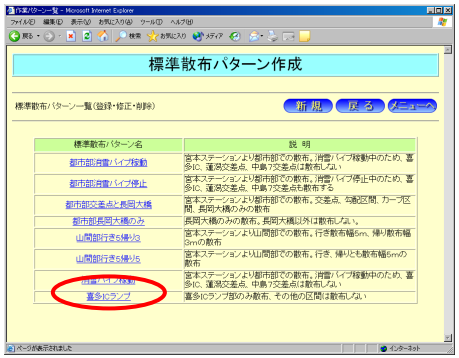


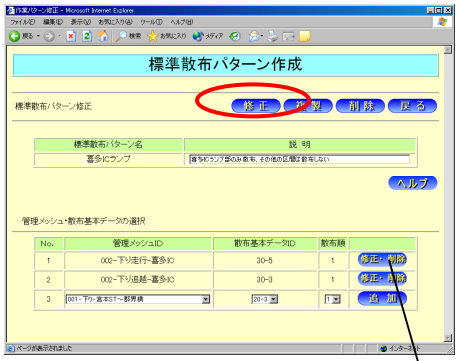


管理メッシュと散布基本データを組み合わせて「追加」ボタンを押すと、１つの散布メッシュが登録されます。散布パターンに必要な分の散布メッシュを追加してから、「登録」ボタンを押してください。

散布順は通常は１でOKですが、ランプ部のように本線と副道があるような場合には、散布の指示を分岐する手前から行う必要があるため指定するもので、どちらかが１でもう一方が２になります。

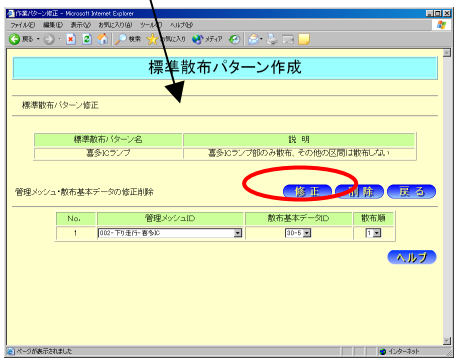
散布計画の修正



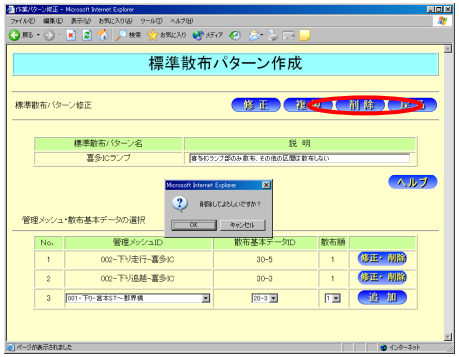


下の画面での修正だけでは、修正の変更がデータに反映されません。最後に上の画面の「修正」ボタンを押して修正を確定してください。

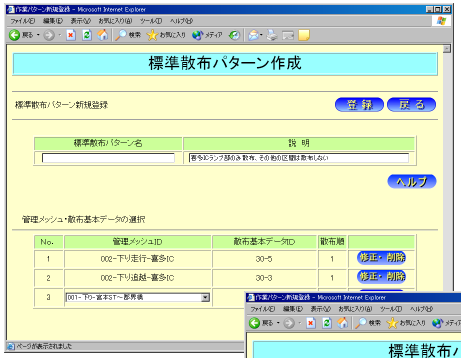
データを修正する場合は、「修正・削除」ボタンを押して、各散布メッシュの情報を表示てから。散布メッシュの情報を修正して「修正」ボタンを押すか「削除」ボタンを押して変更を行ってください。



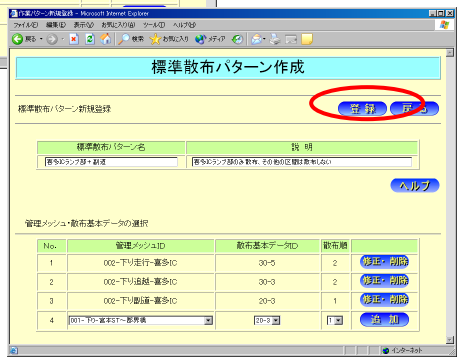
散布計画の削除・複製



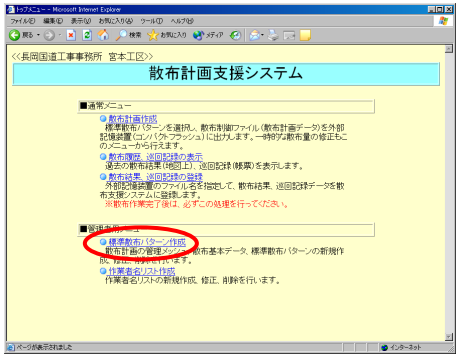
データを削除する場合は、「削除」ボタンを押してください。

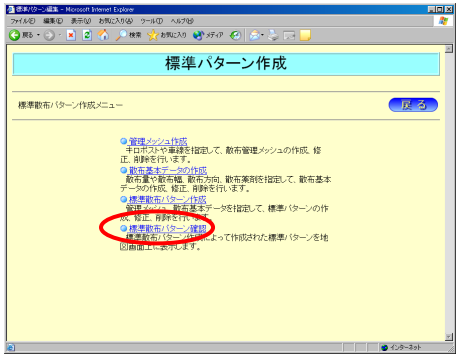


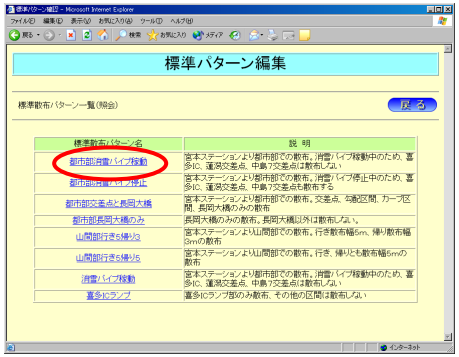
現在表示されているデータの複製を作成して、新たなデータを作成することができます。「複製」ボタンを押してから必要項目を入力して、「登録」ボタンを押してください。



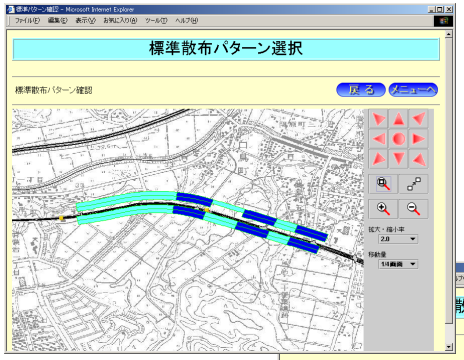
散布計画の確認

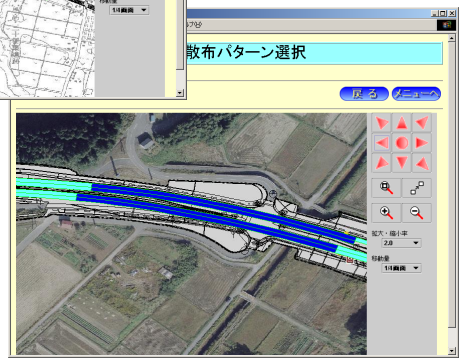






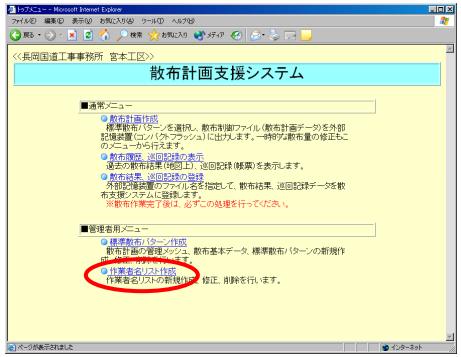
表示するパターンを選択

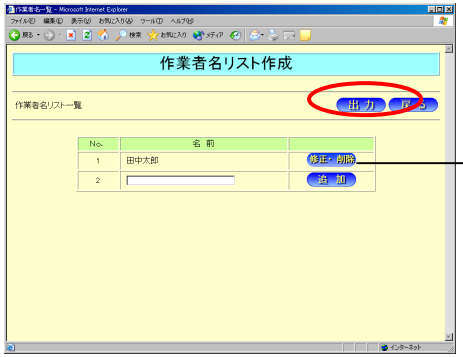


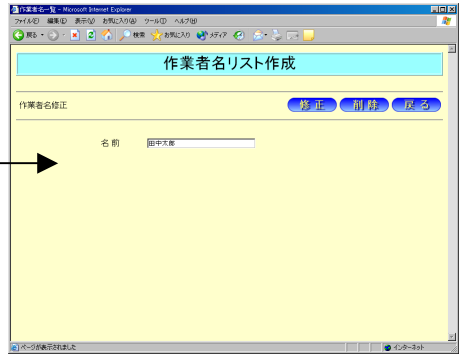


作業者名リストの作成

操作支援システムで表示される作業者名リストを作成するためのものです。ここで作成された作業者名リストファイルをUSBメモリに保存することによって、操作支援システムの作業者名リストが自動的に更新されます。

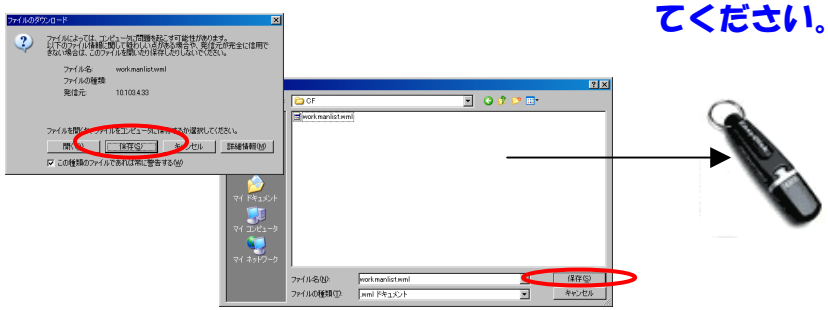






データを修正する場合は、「修正・削除」ボタンを押して修正項目を直接修正してください。

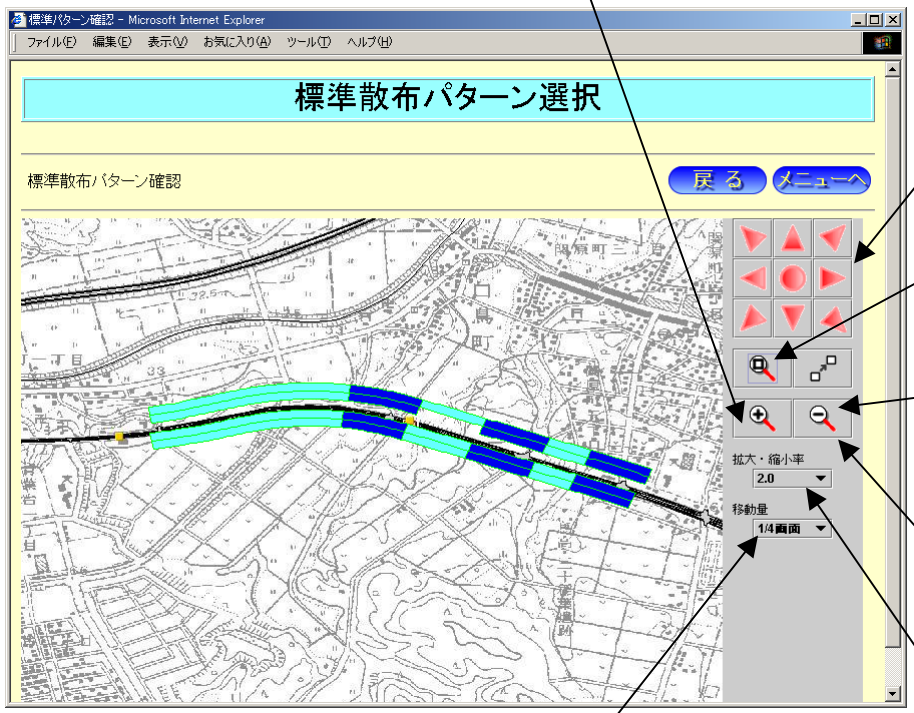
操作支援システムに反映させるには、「出力」ボタンを押して作業者名ファイルをUSBメモリに保存してください。



ファイル名は変更せずに保存してください。

地図表示の使い方

地図を拡大するためのボタンです。



地図をスクロールするためのボタンです。中央の●ボタンを押すと初期表示に戻ります。

矩形を指定して地図を拡大するためのボタンです。ボタンの色が赤色の時に拡大が可能です。

地図を移動するためのボタンです。ボタンの色が赤色の時に、マウスボタンを押しながら地図を動かすと地図が移動して表示されます。

地図を縮小するためのボタンです。

スクロールボタンを押した時のスクロール量を指定します。

拡大、縮小ボタンを押した時の拡大・縮小率を指定します。

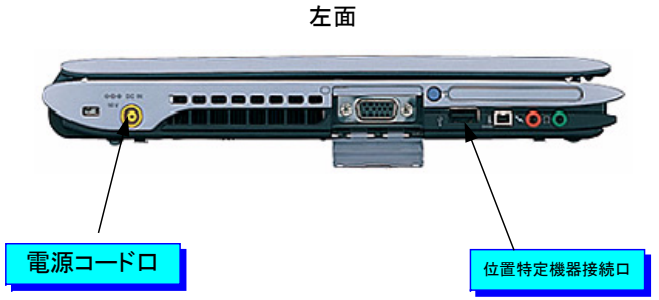
巡回記録および操作支援システム

取扱説明書

1．パソコンの説明
2．接続機器
3．位置特定機器
4．パソコンと周辺機器の接続
5．操作手順
　5－1．開始操作手順
　5－2．システム起動準備
　5－3．システムの起動
　5－4．操作開始時
　5－5．音声入力
　5－6．音声一覧
　5－7．自動散布と手動散布の切替え
　5－8．操作停止時
　5－9．終了操作手順
6．拡張機能
7．故障かなと思ったら

パソコンの説明

左面
右面
移動
（タッチパッド）
ボタン
（マウス左ボタン）
電源スイッチ


左面
電源コードロ
位置特定機器接続ロ

右面
マイク、温度計接続ロ

接続機器

USBメモリ

ヘッドホン付きマイク

車外に取付済

温度センサー

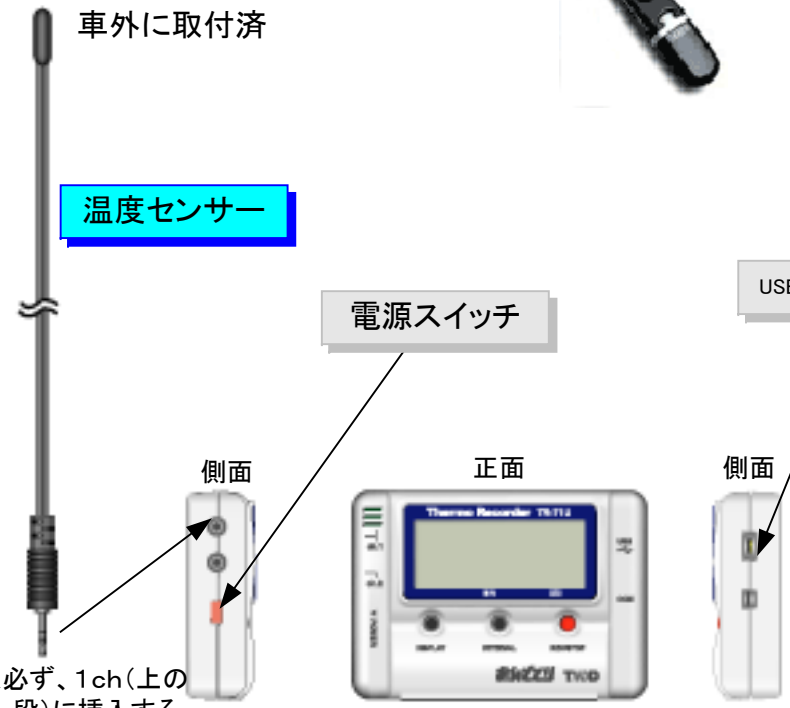
電源スイッチ

USBケーブル接続口

側面

正面

側面

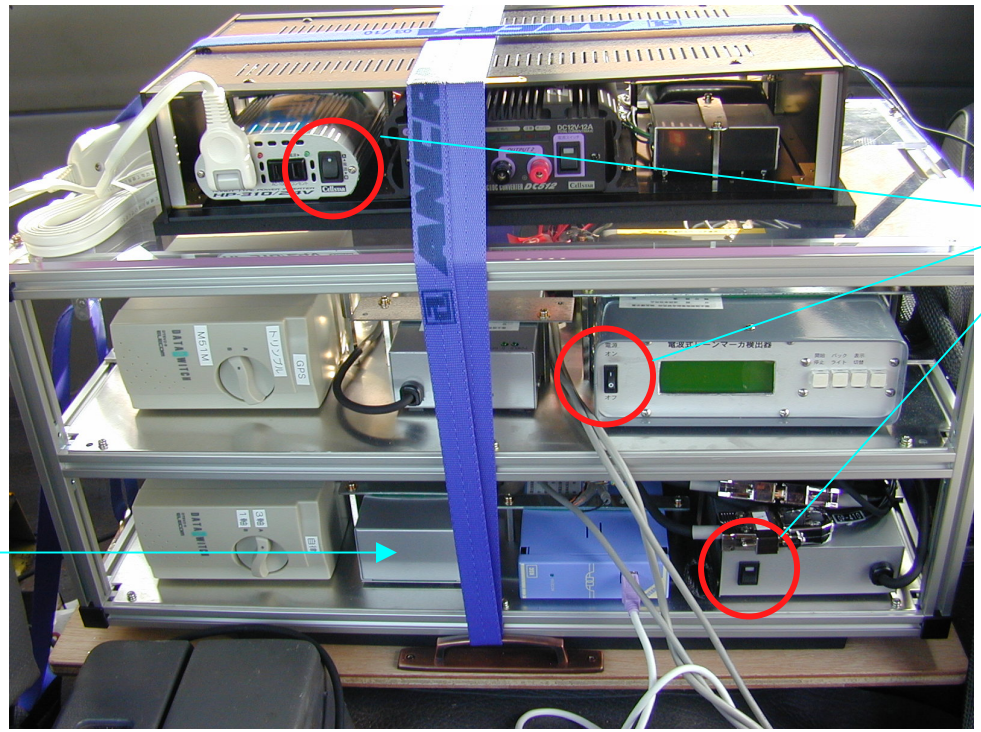


必ず、1ch（上の段）に挿入する

温度センサーロガー

位置特定機器接続ケーブル

位置特定機器



位置特定機器の電源スイッチ

上側がへこんだ状態で電源ONです

この背面に高精度自律センサのLEDランプがあります

車内の中央後ろにあります

パソコンと周辺機器の接続

右側

パソコンと周辺機器の接続

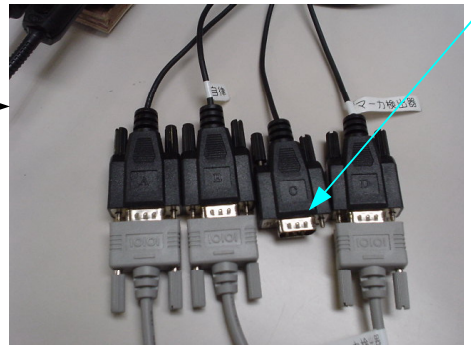
左側

左側

車速センサーケーブル

位置特定機器接続ケーブル

ここに、散布制御装置からの
ケーブルを差し込む



各ケーブルにセンサーの名前が付
いていますので、名前を合わせて、
しっかりとはめこんでください。

開始操作手順

本システムの開始操作は以下の手順で行って下さい。

①USBメモリをステーションから持ってくる
②ケーブル類の接続を確認する
③散布車のエンジンをかける
④シガーをシガーソケットに差し込む
⑤位置特定機器の電源を入れる
⑥ パソコンの電源を入れる
⑦ ログインする
⑧温度計の電源を入れる
⑨パソコンと周辺機器を接続する
⑩散布制御装置のスイッチを入れ、レバーを「PC入り」にする
⑪高精度自律センサのLEDランプが点滅から点灯に変わるのを待つ
⑫ システムを起動する
⑬ 「開始」ボタンを押す
⑭「しばらくお待ちください」の画面が消えるまで待つ

散布車を発進させてください。巡回記録および自動散布が開始されます。

システム起動時における注意

●車両は、必ず屋根などの障害物がない場所で車両が停車した状態でシステムの起動を行ってください。障害物があると現在位置を取得することができません。

●エンジン起動後、パソコンおよび接続機器の電源を入れ、高精度自律センサのLEDが点滅から点灯になるまでしばらく（約３分）お待ちください。（次ページの図参照）

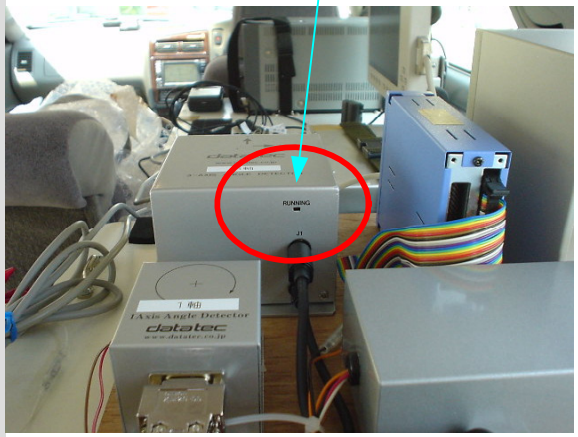
●GPSは電源投入後すぐは位置を確定できません。特に前回の場所から距離が離れた場合には位置の特定に時間がかかります。システムを開始して、「しばらくお待ちください」の画面が表示されている間はこの処理を行っていますので、この画面が表示中は車両を発進させないでください。

●パソコンの電源投入時に、ごくまれに、矢印の移動（タッチパッド）が使用できない場合があります。その時は、位置特定機器接続ケーブルを１度抜いて再度差し込んでください。（次ページの図参照）

システム起動時における注意

つづき

このLEDランプが「点滅」から「点灯」に変わったら準備OKです。

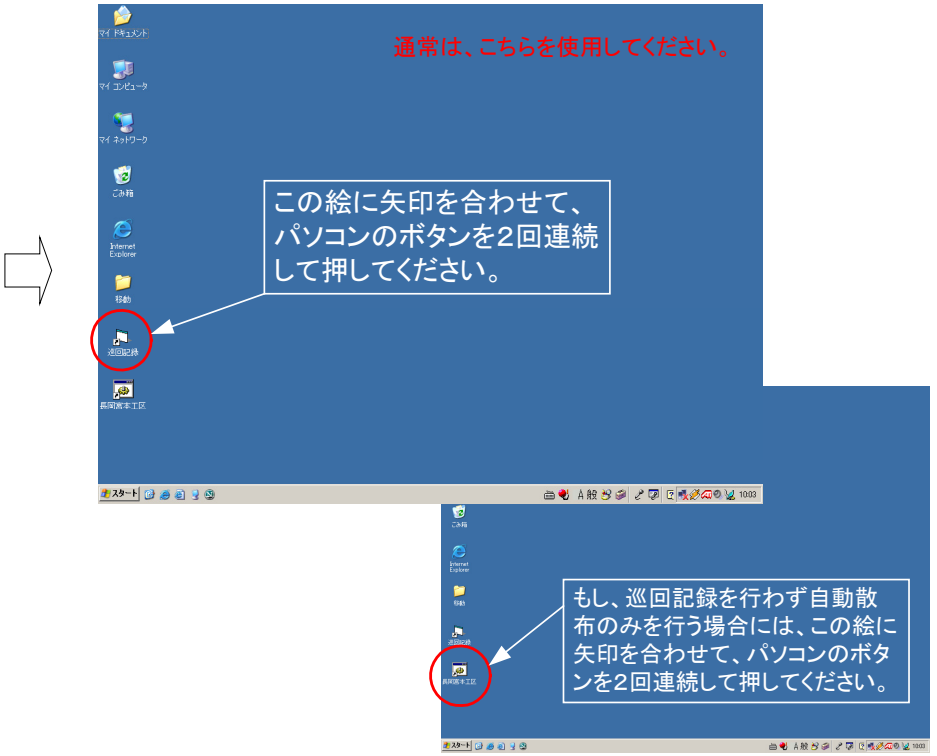


散布車では、運転席左後ろにある位置特定機器電源スイッチ面の裏側にあります。ラックの関係で非常に見えにくい場所にありますので、手鏡等で確認してください。

パソコンの電源投入時に、矢印の移動（タッチパッド）ができない場合は、ここを１度抜いてから再度差し込んでください。

システムの起動

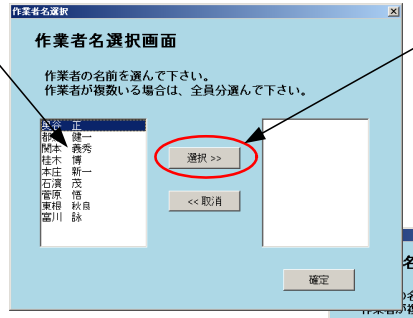
パソコン前面の電源スイッチを右に押すとパソコンが起動します。



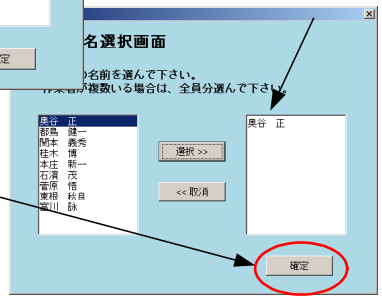
操作開始時

作業者名の選択

矢印を自分の名前に合わせてパソコンのボタンを1回押すと自分の名前の色が変わります。

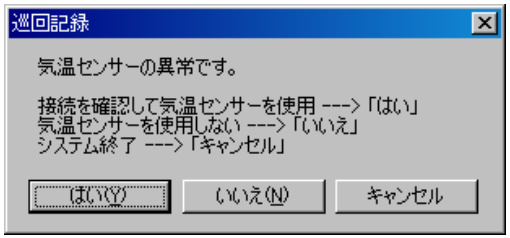


自分の名前の色が変わったら、画面の「選択＞＞」という文字のところに矢印を移動して、パソコンのボタンを1回押してください。



自分の名前が表示されたら画面右下の「確定」という文字に矢印を合わせて、パソコンのボタンを1回押してください。成功すると次の画面が現れます。

間違って名前を選択した場合は、右側の名前に矢印を合わせてパソコンのボタンを押し名前の色を変えてから、画面の「＜＜取消」という文字のところに矢印を移動してパソコンのボタンを1回押すと、選択が解除されます。

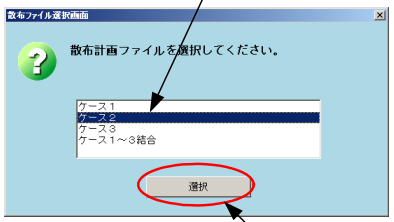


システム起動時にこの画面が表示された場合は、温度センサーの電源が入っていないか、接続されていない状態です。電源および接続を確認してください。
電源および接続を確認後「はい」を押すと正常に動作します。気温センサーを使用しない場合は「いいえ」を押してください。

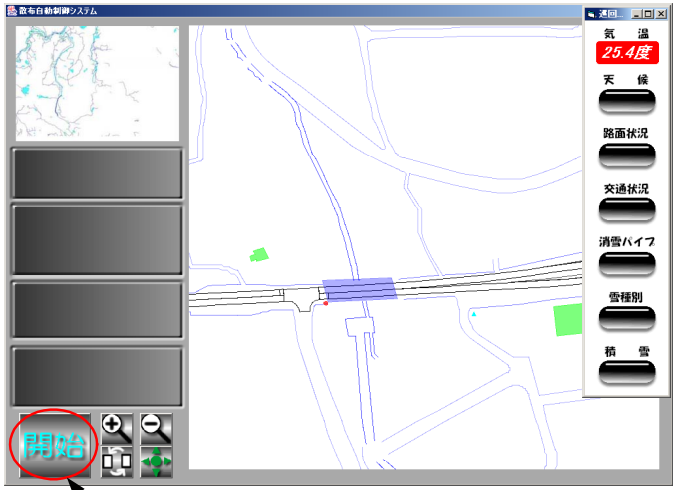
操作開始時

散布計画ファイルの選択および散布開始

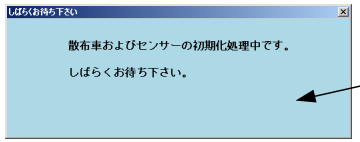
現場代理人の方から聞いている文字の部分（散布のパターン名）に矢印を合わせて、パソコンのボタンを1回押してください。文字の色が変わります。



選らんだ文字に間違いがなければ、画面の「選択」に矢印を合わせてパソコンのボタンを1回押してください。成功すると次の画面が現れます。



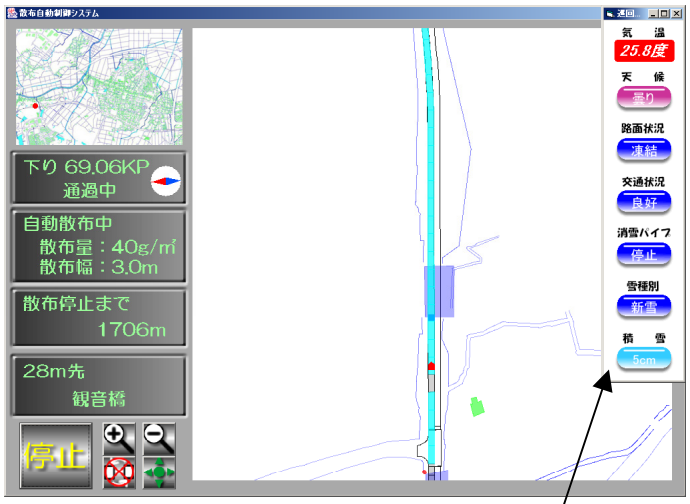
画面の「開始」という文字に矢印を合わせてパソコンのボタンを1回押してください。しばらく、画面の真ん中に次の画面が表示されます。



この画面が表示されている間は散布車を走らせないでください。この画面が消えたら散布車の出発準備完了です。

音声入力

巡回地点に来たら、巡回地点の状況をマイクに向かって話してください。



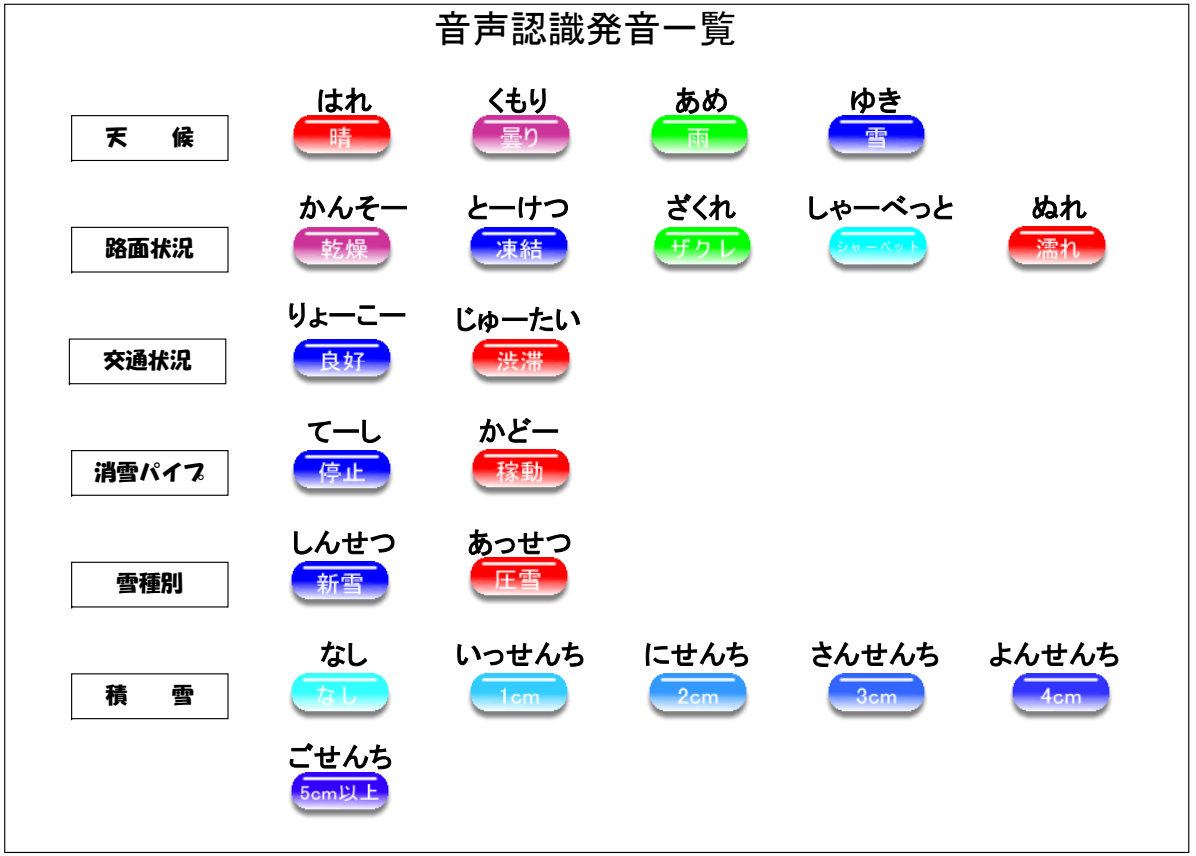
音声認識されると表示が変わる

正常にシステムに登録されると、音声で設定内容をお知らせし、画面の表示が変わります。もし、音声が聞こえなかった場合は、システムに正常に登録されていませんので、再度話してください。

人や騒音状況によっては、音声の認識が悪い場合があります。その時は何度か話してみて下さい。

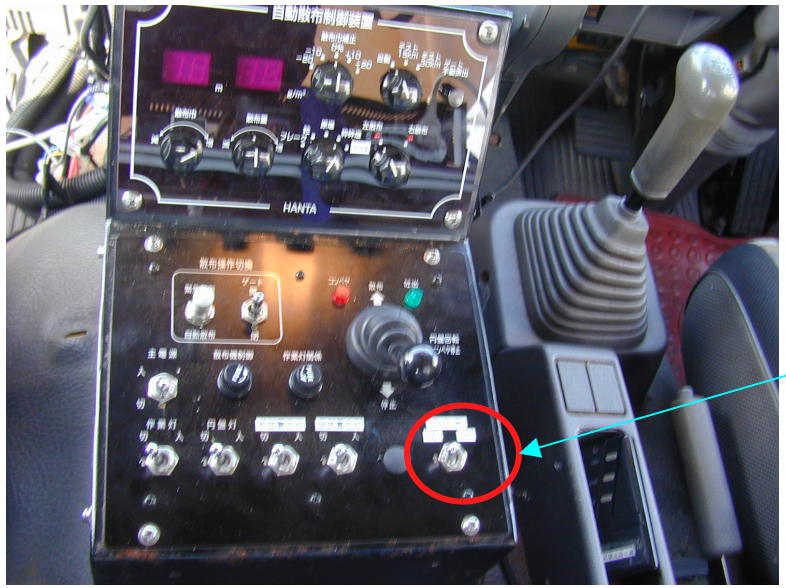
音声一覧

登録できる単語は、下記の通りです。



自動散布と手動散布の切替え

自動散布と手動散布の切替えは、散布制御装置の「PC入り・PC切り」レバーを切替えるだけです。「PC入り」の場合は自動散布、「PC切り」の場合は手動散布となります。手動散布の場合は従来どおり、ダイアルやレバーによって散布量や散布幅を切替えてください。



PC入り：自動散布
PC切り：手動散布

「PC入り・PC切り」レバー

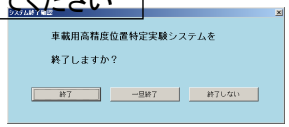
操作停止時

まだ巡回地点が残っていて巡回する場合は「はい」に矢印を合わせて、パソコンのボタンを押してください。

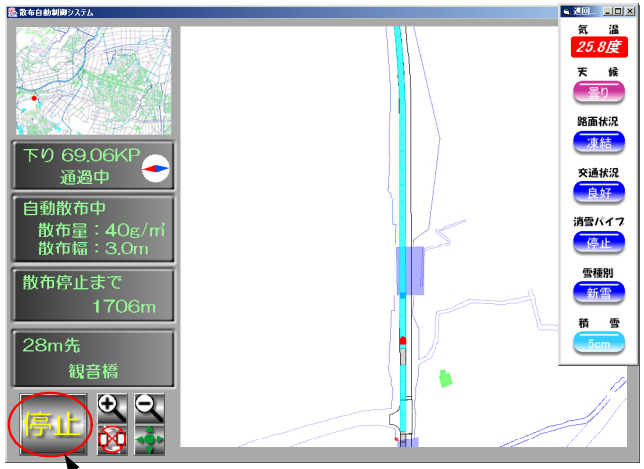
全ての巡回地点の巡回が終わった場合は「いいえ」に矢印を合わせて、パソコンのボタンを押してください。

「終了」または「一旦終了」を押した場合、約5秒後に上の画面が表示されます。

選択する項目に矢印を合わせて、パソコンのボタンを押してください



「終了」---> システムを完全に終了します。
「一旦終了」---> 現在の散布制御情報を残したままシステムを終了します。現在の散布制御を引き続き行う必要があるのに、システムの電源を切る（散布車のエンジンを停止）必要がある場合に使用します。
「終了しない」---> 巡回記録および自動散布システムを一時的に中断します。システムはそのまま表示されていますので、「開始」ボタンを押すと再度システムが動作します。



散布が終わりステーションに到着したら、画面の「停止」の文字に矢印を合わせて、パソコンのボタンを1回押してください。次の画面が表示されます。

終了操作手順

巡回および散布が終了しステーションに戻ったら以下の操作を行って下さい。

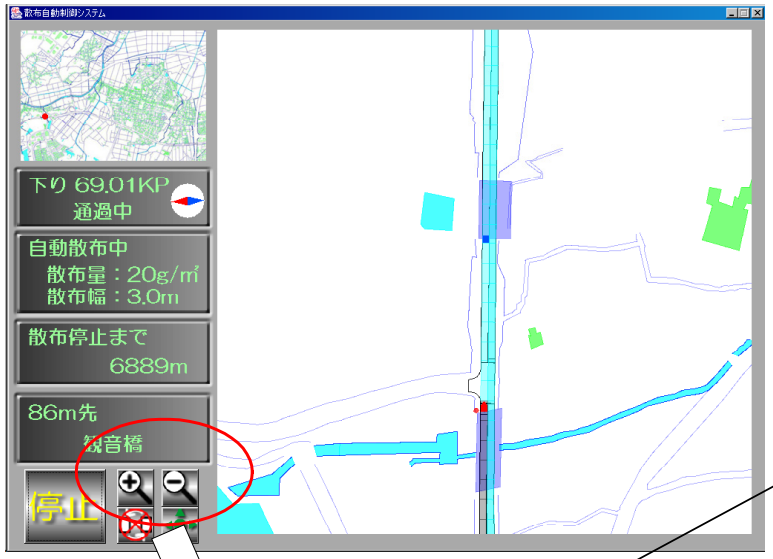
薬剤積込みのために戻った場合についてはこの操作は不要です。散布車のエンジンを切らないのであれば特に何もする必要はありません。ただし、散布車のエンジンを切った場合には、もう一度開始操作から始め直してください。

①「停止」ボタンを押して、システムを終了する
②位置特定機器の電源を切る
③温度計の電源を切る
④パソコンの電源を切る
⑤シガーソケットからシガーを抜き取る
⑥パソコンから周辺機器を抜き取る
⑦散布車のエンジンを停止する
⑧USBメモリを抜き取る
⑨USBメモリをステーションに持っていく

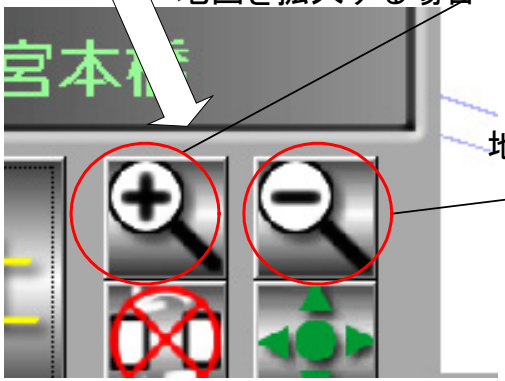
電源等の場所は開始操作時の説明を参照してください。

拡張機能

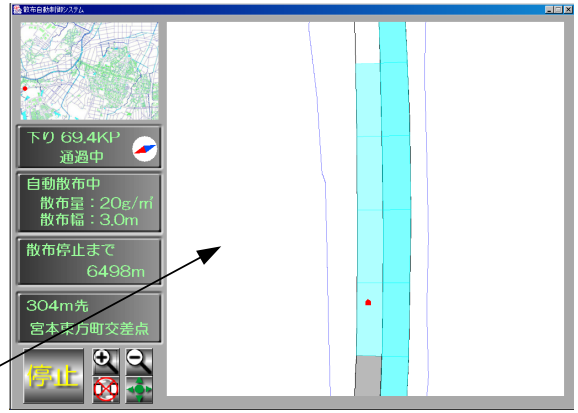
地図画面の拡大・縮小

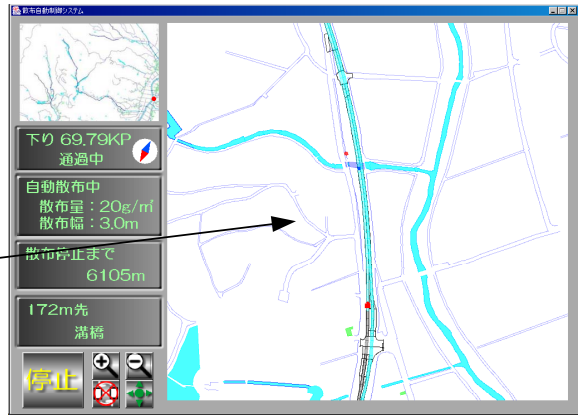


地図を拡大する場合



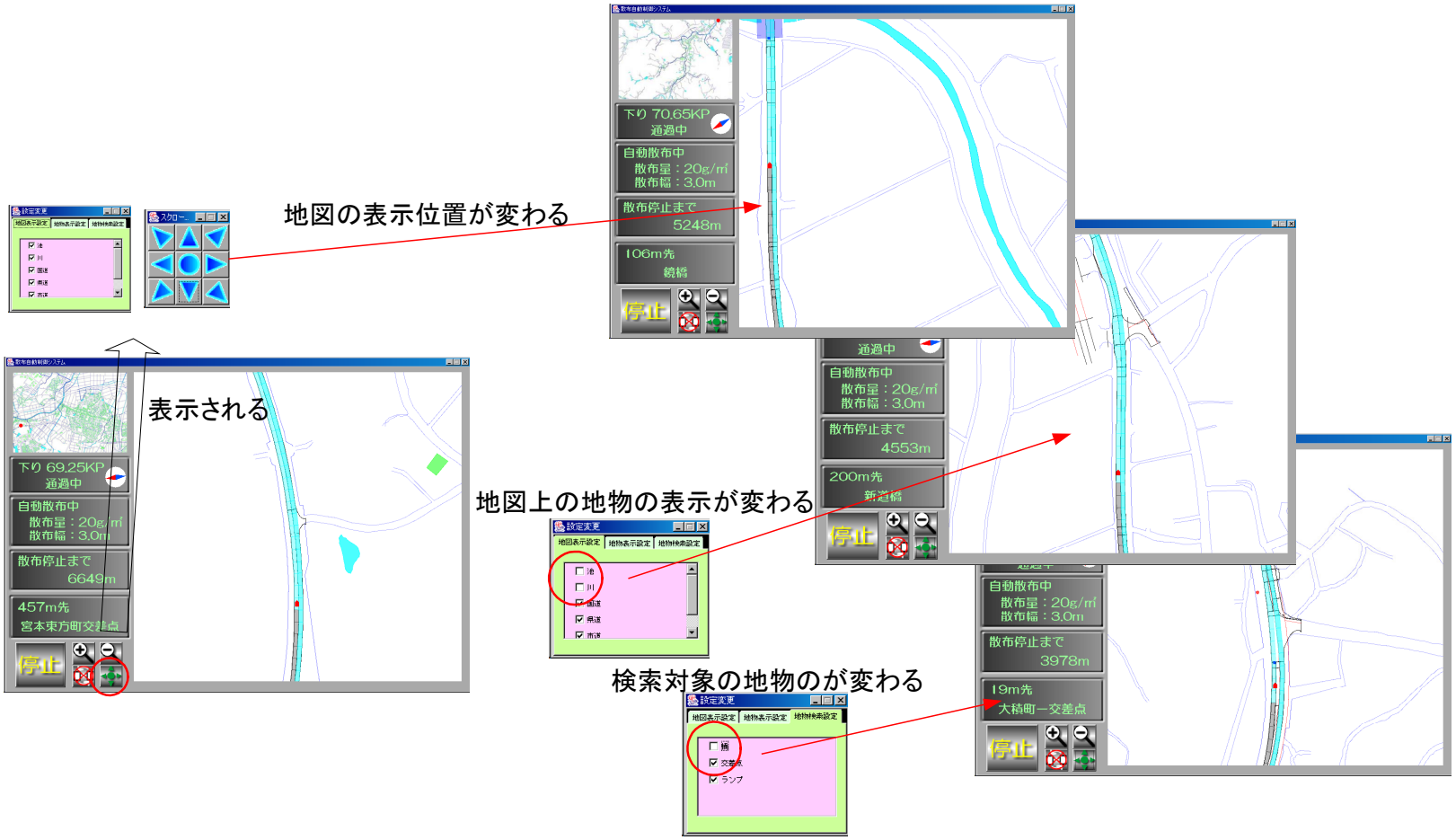
地図を縮小する場合





拡張機能

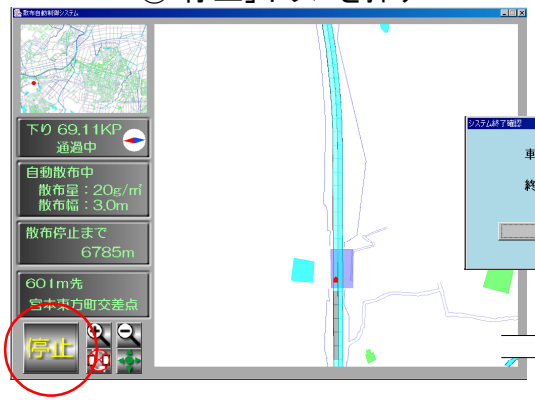
地図画面の表示位置の切替（スクロール）および表示地物、検索対象地物の変更



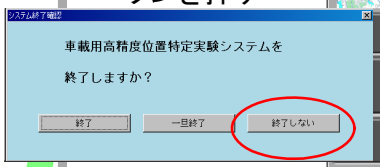
拡張機能

散布計画ファイルの切替

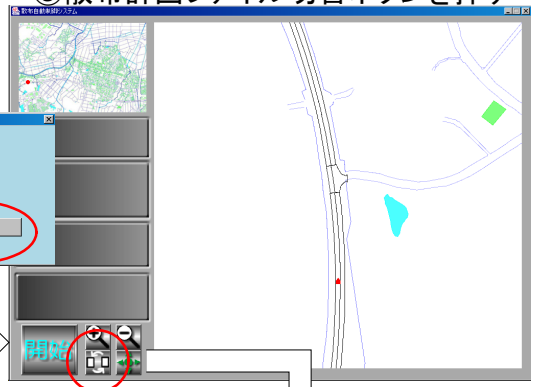
①「停止」ボタンを押す



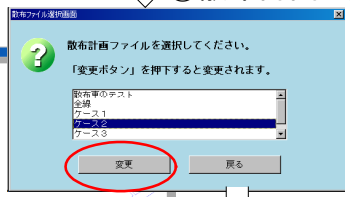
②「終了しない」ボタンを押す



③散布計画ファイル切替ボタンを押す



④散布計画ファイルを選択する



⑤「開始」ボタンを押す

⑥散布計画ファイルが切換り散布が開始される

※複数の散布順が指定されている散布計画ファイルの場合には、切替え前の状態が正常に散布履歴に反映されない恐れがありますので注意ください。

故障かなと思ったら

現在位置がずれて地図上に表示される　　**機器ケーブルの接続を確認してください**

位置特定機器が準備できていないのに発進してしまった可能性があります。
車を停止させ、自動散布システムの「停止」を押して１度システムを終了し、
位置特定機器の準備状態を確認してから、再度システムを起動しなおしてください。

自動散布されない　　**散布制御装置とのケーブルの接続を確認してください**

散布制御装置の電源が入っているか確認してください。電源が入っている場合は、自動・手動切替えスイッチが「PC入り」側になっていることを確認してください。

ビープ音が鳴り続ける

システムが高負荷により、処理が間に合わなくなったことが考えられます。
自動散布システムの「停止」を押して１度システムを終了し、
位置特定機器の準備状態を確認してから、再度システムを起動しなおしてください。

故障かなと思ったら

音声が認識されない　　マイクの接続とUSBハブの電源を確認してください

システム起動後にマイクを差し込むと音声が認識されません。マイクを接続後、システムを一度停止し再度起動してください。
また、音声は、人や周りの騒音状態によって認識率が変化します。認識されるまで何度か話してみてください（それでも認識が悪い場合は、Via Voiceのマニュアルを参照して音声の登録を行ってください）。

気温が取得されない　　気温計の接続とUSBハブの電源を確認してください

気温計のスイッチが入っていることを確認してください。スイッチが入らない場合は、気温計の電池切れが考えられますので、新しい電池（単三１本）に入れ替えてください。

画面が動かない

何らかの原因でパソコンのプログラムが異常状態になっています。パソコンの再起動を行ってから、再度システムを起動しなおしてください。（パソコンの再起動ができない場合は、電源スイッチを５秒以上右へ押した状態にしておいてください。パソコンが再起動されます）