

## 6.2 減災情報可視化システム（東京大学情報理工学系研究科）

### 6.2.1 研究の概要と目標

災害情報共有を進めるためには、本当に必要な情報を抽出し、それらの記述方法を統一した上で、様々なシステムで取り込めるようにすることが必要である。本テーマでは、救助計画立案の意思決定を支援するために、災害や事故が発生したときの錯綜かつ混乱した情報やマルチエージェントシミュレーションの結果を、多様な危機管理担当者にとって目的別に整理された最適な形で表示する技術と、表示に対する種々の入力インタフェースを開発する。ひと目で見てわかりやすい表示方式と、自然かつ容易な入力方式を目指す。すなわち、現行の紙地図やファックスに取って代わる、危機管理責任者のスムーズな意思決定支援に役立つ災害情報可視化システムを開発する。

上記の目的を達成するために、本年度は以下の2つの研究目標を立てた。

#### (1) 意思決定支援に使える情報提示プロトタイプの開発

多様なレベルでの災害対応意思決定を支援するために、災害対策本部、現場など各種の環境に応じた表示機器を想定した汎用的な情報提示方式の確立を目指す。異なる画面サイズ、あるいは異なる通信環境において、情報提示ポリシーの技術的一貫性を保つことがポイントである。その上で、ユーザごとに表示のカスタム化が行なえるメカニズムの構築を目指す。

不完全情報の表示方式はこれまでに出来た各種のアイデアを早めに一本化することを目指す。

#### (2) 表示制御や指示発令のための入力インタフェースの開発

上記の(1)で計画しているような多様な情報提示手段にはそれぞれ固有の入力手段が対応するが（例えば、キーボードがある／ない、マウス／ペンなど）、入力インタフェースの基本に一貫性をもたせた設計を行なうことを目指す。評価は災害時のような混乱状況でもすぐに使える習熟容易性と緊急即応性を重視して行なう。

なお、初年度の最終段階で創案した、多人数が複数のマウスを使用し、画面上での情報共有を円滑にする方法（マルチマウス）については、災害時の情報共有にとどまらない汎用性の高いアイデアであるので、早めの実装を目指す。

### 6.2.2 研究成果

災害対策本部や現場での情報提示を考えると、情報提示が入力インタフェースと密接に関係していることに容易に気づく。情報提示に対して、なんらかのアクションをとるうちのいくばくかはその情報提示画面に対応するなんらかの入力であるからである。したがって、「意思決定支援に使える情報提示プロトタイプの開発」と「表示制御や指示発令のための入力インタフェースの開発」の2つの研究項目は相互に密接に関連しており、切り分けて説明するのは適切ではない。そのため、情報提示プロトタイプの開発のほうでも入力インタフェースについて一部言及する。

### (1) 意思決定支援に使える情報提示プロトタイプの開発

プロトタイプを開発する前に、災害対応に適合した情報提示と入力についてのガイドラインの提案を定めた。さまざまな種類の機器上に情報提示や入力インタフェースを開発するときに、機器ごとにバラバラの原則が立っているようでは関係機関のあいだでスムーズな情報共有が行なえなくなるからである。現時点でのガイドライン案の概要について述べる。

災害時の情報提示には、[1] 災害が起こった、あるいは災害が起こると予想される現場・自治体から報告のあった被害情報の表示とともに、[2] すでに全国に張りめぐらされている情報インフラからもたらされる情報(例えば気象庁や防災科研からもたらされる雨量情報、雨雲情報、震度情報、潮位情報等)の表示、の2種類がある。後者は情報提示に関して、すでに一定の標準(デファクトスタンダード)があり、それに対して改善を加える提案は 特段のことがないかぎり行なわない。

災害時には地図画面をベースとする GUI が了解性が高く、有効である。このほかにも上部機関への報告や情報整理等のための帳票処理、被害集計表処理、速報的な文字テロップなどがあるが、ここでは地図画面上での情報表示(情報可視化)とそれに適合した情報入力の方式のガイドラインを提案する。情報は GUI 以外の形でも入力され得るが、GUI 入力されたものを含め、情報共有ハブ(DaRuMa ハブ)に一旦格納され、そこから専用 AP を通して種々のビューアやコンソール(およびそれに付随した入力機能をもつ AP)に可視化に適した情報として取り出される(図 6.2-1)。なお、GUI デザインをするにあたって、日本人間工学会・アーゴデザイン部会 スクリーンデザイン研究会編(代表編者 菊池安行、山岡俊樹) GUI デザイン・ガイドブック ―― 画面設計の実践的アプローチ 第2版、海文堂(2004年)を参考にした。

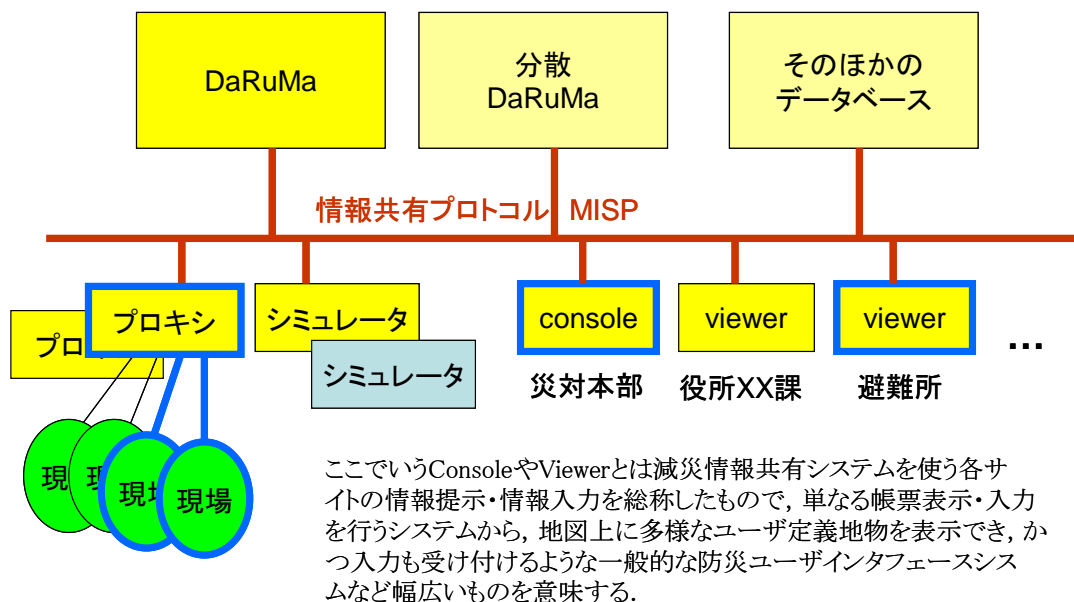


図 6.2-1 DaRuMa をベースとした情報共有プラットフォーム

### a) 情報提示・入力インタフェースのガイドライン

以下の説明では入力インタフェースをマウスに限定して記述する。

・画面は最大限地図表示のために使う。特に携帯機器のような小画面ではこれは必須である。すなわち、メニューバーや凡例などを常時は表示しない。地図には表示すべき文字列が多数含まれるが、画面サイズの制約があるので、防災に必要なもの（町丁目、ランドマーク、避難所等）を選択して限定的に常時表示する。

- ・地図上には元来ある地物のほかに、アイコン、折れ線、領域などの「ユーザ定義地物」を描く必要がある。しかし、これらの意味の説明的な文字列は常時表示しない。しかし、マウスオーバー（マウスが画面上の地物表現の上に来ること）により、地物に関する短い情報を吹き出しで出す。
- ・メニューはマウスクリックによりポップアップする。背後の地図をなるべく隠さないように、メニューは半透明とする。
- ・アイコンはすべてオブジェクトアイコン（その場所になにかがあることを示す）または状態表現アイコン（なんらかの状態であることを表現する）とする。操作アイコンは原則として使わない。現場に対する指令等は間違いを起こさないために、アイコンではなく、指令者と受信者に共通で理解性の高い言語表現を媒介に行なう。ただし、現場からの報告等ではわかりやすいアイコンを用いる。
- ・ユーザ定義地物のデザインには、一貫性のある一種の文法をもたせる。例えば、オレンジ、赤、紫の順に危険度・緊急度を高くするなど。これは使用者の認知負荷を下げ、いきなり使い始めてもすぐ習熟できるようにするためである（図 6.2-2）。なお、国土地理院の定めた地図記号を可能なかぎり遵守する。


















色	住民に関するもの	
	対応行動	
	被害程度	 <  <  < 
	安全・対応完了	
明るさ	何らかの程度	 <  <  < 
太さ	重要度	 <  < 
線の形	過去・現在・未来	 過去  現在  未来

図 6.2-2 すぐになじめる GUI

- GUI を経て入力する際には、原則として、機能先行方式ではなく、オブジェクト先行方式とする。つまり、まず対象となるオブジェクト（地物、ユーザ定義地物）を選び、それに対してどのような操作をするか（機能）を選ぶ。機能先行にするためには、すべての機能アイコンを画面上に表示しておく必要があるからである。
- 地図上のオブジェクト先行方式におけるオブジェクト選択は一般的に多義である。例えば、家を指したのか、その家を含む街区を指したのか、その家を含む町丁目を指したのかなど1回のオブジェクト選択アクションでは決められない。多義性の排除はユーザ指定の優先度を用いて、最もよく使う選択対象地物を優先し、ポップアップメニューの最初に「その他の選択」という共通メニュー項目を入れて、優先度を低くしたものを簡単に選択できるようにする。
- オブジェクト先行方式においてどのようなメニューがポップアップするかは、XML で簡潔に記述できるようにする。ユーザがこの XML を修正すれば、それがただちにメニューの形やボタンなどの並び方に反映される。ただし、メニューボタン等の選択に対するアクションはプログラムとして用意しておく必要がある。XML はメニューと処理プログラムの対応関係を明確にする。
- 入力はユーザの認知負荷の少ないモードレスを基本とする。モードが必要な場合は、注視点にあるマウスカーソルなどの形・色を変えてユーザにどのモードであるかを常に見せるようにしておく。そのモードで行なえない動作をした場合はエラーではなく、原則として無視することとするが、カーソルの形を一瞬（1~2 秒程度）「？」の形にして注意を喚起するのが望ましい。
- 地図には広い領域の一覧性があるが、一覧性を要求すると、部分領域の詳細情報が不足する。またその逆も起こりうる。1枚の画面で一覧性と詳細性を同時に両立させることはせず、縮尺の切り替えを行なうか（主に1人が見る場合）、2枚の画面を使って異なる縮尺を同時に表示するようにする（CSCWのように主に複数人が大画面を通して協調作業をするような場合）。
- 得られた多数の情報から地図を色分けして表示することが有効な場合がある。例えば、震度分布、浸水水位分布、町丁目単位で被害程度の分布など。これらについては、既存の方法を援用する。
- マルチマウスを用いた複数人による画面共有入力に対応する。消防、土木など、異なる災害対応機関が使うマウスごとに異なる機能や権限を付与することができる。これらも XML で記述する。これにより、同じ地物を選択したときに職掌範囲ごとに異なるメニューをポップアップさせることができる。

## b) 地物表現のガイドライン

災害対応では画面地図上にいろいろなものが書き込まれるので、地図は基本的に白地図とするが、淡泊な色がついていたほうがよい。例えば、河川や用水は薄い水色とする。また、文字、行政界、道路、家の外形は、上に別の文字を書き込んだときに邪魔にならないように黒でなくてやや濃い灰色にする。

地名や河川名の文字列表示は地図がどこを表しているのかがわかる程度の必要最小限にとどめる。災害対応に必要なランドマークには地物名を入れる（例えば公園，駅，病院等）。

個々の建物が表示されるような縮尺の場合，家は薄いペーパージュ等で着色してもよい。ただし，防災対策上重要な地物はやや濃いめの着色をし，地図記号と名称をつける（図 6.2-3）。個々の建物が表示されないような縮尺の場合，通常の写真記号だけを描く。地物名は最小限とする。

地図記号はアイコンとして実装する。アイコンを選択することによって，それに関連した詳細な情報が得られる。例えば避難所であれば，その避難所の現在の収容状況等の情報などが表示される。このほか，町丁目単位に情報アイコンを置き，それを選択すると，その町丁目に関する情報が得られるという形も考えられる。

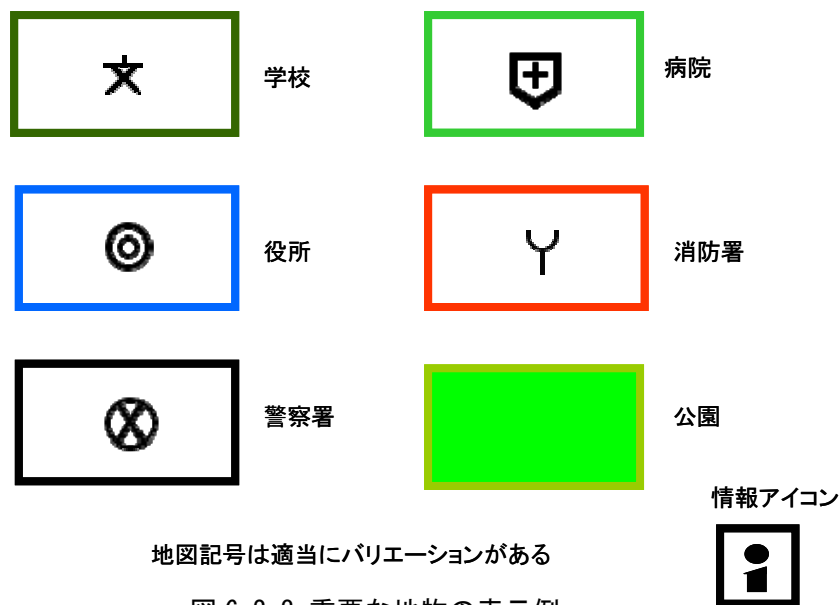


図 6.2-3 重要な地物の表示例

### c) ユーザ定義地物

地図に元来表現されている地物（システム定義地物）以外の地物をユーザ定義地物と呼ぶ。ユーザ定義地物は，アイコン，幾何図形（単独点，折れ線，または領域）の4種類である。幾何図形のうち折れ線，領域には原則として意味を示すアイコンが付随する。単独点は位置を指定するための一時的なマークである。

アイコンは画面に多数のアイコンが入ることを考慮し，原則として16x16，あるいは24x24ドットの大きさとする。しかし，視認性を高めるために，それより大きいサイズのアイコンを使うこともある。アイコンは了解性の高い絵柄とする。地図記号のように抽象度の高いアイコンは小型でも十分である。

アイコンは多種類にわたるが，なるべく基本概念を表す構成要素（基本語彙）の組合せで複合概念を表現するようにして，意味を判別しやすくする（図 6.2-4，図 6.2-5）。図 6.2-4 で黒いバーと黒いハットが使われているが，例えばバーの上下で意味を区別していない理由は，平面上にディスプレイしたときにどちらが上下かわからなくなるのを防ぐためである。

アイコンの意味は、マウスオーバーの吹き出しの中で短い日本語で表示される（アイコンの注目）。さらに詳細な情報を知りたければそこでクリックすればよい（アイコンの選択）。マウスの注目を使えば、凡例の中で同じ図形を探す手間が省かれる。しかし、必要であれば、位置非依存メニューで凡例を画面に出すことも可能である。

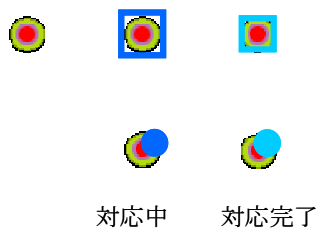
事件あるいは現象を示すアイコンが青い太枠の四角で囲まれている、あるいは青い丸が重ねて置かれているのは、対応活動がとられていることを表す。それがライトブルーに変わったら、対応行動が完了したことを表す。なお、ライトブルーの枠はアイコンの大きさに縮める。対応が完了したものは重要度が下がるからである。四角の枠にするか、丸で書くかは、見やすさを主体にして適宜選択する。対応中が丸で、対応完了が四角の枠といった混合も可能である。図 6.2-6 は、左から順に、閉じ込められた重症者が発見・報告された、救急隊が向かっている・救出作業をしている、救出完了という意味となる。

	単数	複数
けが人(軽傷)		
けが人(重傷)		
死亡		
要救出者(無傷・埋没)		
要救出者(軽傷・埋没)		
要救出者(重傷・埋没)		
死亡(埋没)		
要救出者(無傷・閉じ込め)		
要救出者(軽傷・閉じ込め)		
要救出者(重傷・閉じ込め)		
死亡(閉じ込め)		

図 6.2-4 被害者に関するアイコン

床下浸水		床上浸水	
冠水(畑)		堤防危険	
堤防漏水		堤防溢れ	
堤防決壊		溢水	
河川増水		避難勧告・指示	
汚泥・ごみ		土石流	
工事・作業中		事件発生	
通行止め		車両通行止め	

図 6.2-5 水害に関するアイコン



四角の枠と、丸の使い分けは適宜でもよい。また2種類を混合してもよい。たとえば、折れ線アイコンはもともと別の色の四角で囲まれているので、その色を変えるか丸を重ねるかは見やすいほうを選ぶ。対応中が四角で、対応完了を丸にしてもよい。

図 6.2-6 報告, 対応, 完了

折れ線は、折れ線のリンク（折れ曲がり点）の指定、自由曲線、あるいはその混合で入力してよい。自由曲線で入力した部分は、データ量を圧縮するために自動的に折れ線へ単純化される。領域は閉じた折れ線の内側として入力される。

折れ線は単独点（リンク）の列なので、すでに描かれた単独点をドラッグすることで補正は可能である。折れ線の途中に単独点を増減する操作は当面の間オプションとする。

折れ線と領域はどちらも描かれた折れ線の線の太さ、形状種類、色と、その途中に描かれたアイコンで意味が表現される。アイコンを注目すると簡単な説明が吹き出し、選択すると詳細情報メニューがポップアップする。

幾何図形は色、線の太さ、線の種類（直線、破線等）、さらに領域の場合は領域境界内側の（部分）ハッチの模様で大体の意味を表す（図 6.2-7）。ユーザの習熟を容易にするために、一貫性のある文法をもたせる。これはアイコンの補助として視認性を高めるためのものであり、機器の描画能力に依存して視認性がやや制限される場合がある。

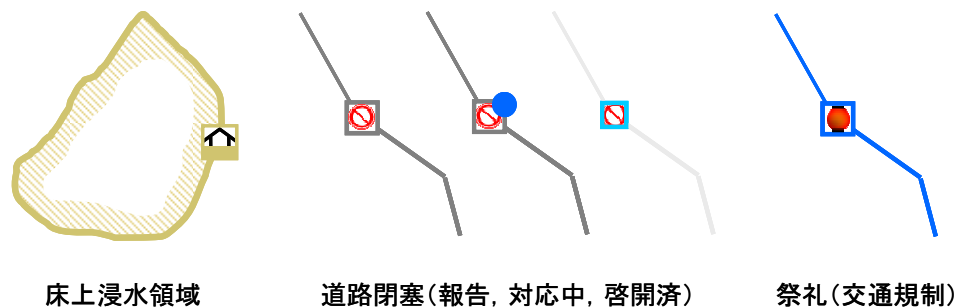


図 6.2-7 幾何図形の表示例

幾何図形の基本的な約束（文法）は以下の通りとする（図 6.2-2 を参照）。

- ・ 住民に関するものは緑系統とする。
- ・ 被害に関するものは黄、橙、赤、紫（後ろのほうはより危険度・緊急度が高い）。
- ・ 明度が低いほど程度が大きいことを示す。
- ・ 無被害が確認されているものは薄い水色とする。
- ・ 災害対応に関するものは原則として青系統とする。ただし、消防、警察、重機、自衛隊は誤解のない範囲でそれぞれ赤、黒、黄、モスグリーンを補助的に使う。

- ・ 災害対応が完了した事件や現象は、事件や現象を表すアイコンの枠をライトブルーに変えるか、ライトブルーの丸をそのアイコンに部分的に重ねる。そして幾何図形の線の明度を上げ、かつ彩度を下げてやや無彩色化し、すでに重要度の高い情報ではないことを表す。例えば、線で表された道路閉塞は、道路閉塞の線を薄灰色に近付けると同時に、道路閉塞アイコンをライトブルーの四角で囲む。対応が終わったものは、対応前の情報を薄く残しておくが、対応済み情報を非表示にする。例えば、停電が復旧した場合は、その停電に関する情報は比較的早く消してもよい。
- ・ 線は実線が現時点の状況、密度の濃い点線は未来、密度の薄い点線は過去を表す。そのほかの形（破線、一点鎖線、波線、二重線など）は個別の意味に対応する

#### d) 最新情報や指令・報告の即時的表示

情報は基本的にすべて MISP（減災情報共有プロトコル）を通して DaRuMa に一旦蓄積される。それを基盤データベースとして必要な画面に表示を行なう AP は DaRuMa を定期的にポーリングしなければならない。その AP から、必要なビューアへプッシュ型で必要な情報が送られる。この AP とビューアのあいだの情報は必ずしも MISP である必要は無い（例えば、AP とビューアが一体化している場合など）。

地図上には新旧取り混ぜた情報が表示される。一般に新しい情報には素早く対応しなければならないものが含まれている。そのため、最新情報が表示された場合、画面を見ている関係者にそれとわかるように表示することが必要である。また、現場で PDA を持参しているような職員には本部から指令が来たことがしばらく画面から目を離していてもわかるようになっていなければならない。

基本的には最新情報をしばらく（重要度により時間幅を変える）ブリンクさせる。最低でも 1 分、標準的には 5 分程度ブリンクするべきであろう。重要度に応じてどのようにブリンクさせるかはユーザがカスタマイズできるようにする。ブリンクはその情報アイコンを選択することで止めることができる。このほかに文字で表された情報がスクロールするような共有(大)画面を別のディスプレイ上に用意し、そこで緊急度・重要度に応じた表示を行なう。非常に重要な情報の提示には音の利用も考慮する。現場職員の PDA のようなものでは、画面中央に「新着情報」のような目立つアニメーションを行なうほか、音を十分に活用する。

#### e) メニューの制御とカスタマイズ

マウスクリックによってどのようなメニューやダイアログボックスが出てくるかは XML で記述する。XML で記述されるのはメニューやダイアログボックスの論理的な構造であり、ボタンの形、色などの見掛けの多くは、それとは独立に規定される。

メニューは、PC のデスクトップ画面におけるコンテキストメニューのような階層構造をなしている。このメニューの階層構造はあらかじめ XML Schema によって記述されたスキーマに則った XML 形式で記述される。このスキーマの定義する範囲内で、ユーザはメニューを定義している XML ファイルを作成・変更することができ、メニューの論理的な形、機能などをある程度自由に変更することができる。



適当なタイミングでこの XML ファイルをロードすると、メニューのレイアウトが自動的に決定され、作成したメニューを利用することができるようになる。

1つのメニューは、1つ以上のメニュー項目とメニュー付加情報からなる。メニュー付加情報にはメニューの種類に関する情報が含まれる。

メニュー矩形の左上部分にある矩形アイコンをドラッグすることにより、メニューを移動させることができる。この矩形アイコンはメニューの概要がわかるようなものにする。できるだけ画面の可視部分を増やすためにメニューの移動中はメニューが最小化される。また、ドラッグが終了すると元の大きさに戻る。

メニュー内でサブメニューを開くボタンをクリックすると、そのボタンに関連付けられている子メニューがポップアップする（メニュー階層を下げる）。また、右クリックによって現在のメニューが消え、親メニューへ戻る（メニュー階層を上げる）。

このメニュー階層を上下する操作中、ユーザはメニューの階層を一度に一階層しか見ることができない。例えばサブメニューが出現するときには、呼び出し元のメニューは一旦消え、呼び出し元のメニューと同じ場所にサブメニューが出現する。これにより、可能な限り地図の可視部分を増やすことができる。また、現在いるメニュー階層が把握できるようにメニューの右上の部分に「←」を表示する。

メニューの一部または全体を、メニューの使用者の権限によってアクセス制限することができる。各メニュー項目に対するアクセス制限は、ACL (Access Control List) を模した機構によって制御する。各メニュー項目には Read (可視) と Execute (実行可能) 属性が設定可能である。Read 属性が true になっていれば、ユーザに対してそのメニュー項目が表示される。Execute 属性が true になっていれば、そのメニュー項目特有の機能を実行できる。メニュー項目特有の機能とは、例えばボタンであれば押すことができるかどうか、情報を表示するテキストならその項目を編集できるかどうかなどを指す。

各メニュー項目について、別に定義されたグループ単位で各属性を設定する。設定されない場合は、どの属性も設定されていない false 状態になる。この状態は、そのメニューが使用不可能であることと同じである。

メニュー項目が定義される XML ファイル中に、各メニュー項目がどのグループに対してどの属性を true にするかを設定する。各グループ内の構成員はメニュー構造を記述している XML か、もしくは別の XML ファイルに記述する。

## (2) 表示制御や指示発令のための入力インタフェースの開発

これに関しては、前項で述べた GUI の入力インタフェースの部分と、画面共有入力の新方式であるマルチマウスの2つの成果がある。

### a) マウスによる災害情報へのアクセスと指令入力

以下、マウスの操作を以下のように略記する。

C	センターボタンのクリック
W	センターホイールの回転. 向こうに回すのを W+, 手前に回すのを W-と書く
L	左ボタンのシングルクリック
LL	左ボタンのダブルクリック
LD	左ボタンによるドラッグ
R	右ボタンのクリック
RR	右ボタンのダブルクリック
RD	右ボタンによるドラッグ

マウス操作はモードレスを基本とするが、3つの自明に区別されるモードがある。3つ目のモードは大画面を水平に置き、周囲から多数の人が画面を共有する場合以外は使用しない (Windows Vista が発売されてからサポートする予定である)。マウス操作の意味は、

result(モード, マウス位置, マウス操作)

という3引数の手続き result として理解するのが最もわかりやすい。

マウスの3つのモードは、通常モード (NoM)、幾何図形モード (GOM)、向き指定モード (OSM) である。モードはカーソルの形で区別される。マルチマウスで使っているとき、それぞれのマウスはカーソルの色で区別される。以下の説明では、NoM、GOM、OSM の記号を使う。

NoM: 通常モード (Normal Mode) 地図上をふつうにマウスを動かし、アイコンを注目したり (簡単な説明の吹き出しが出る)、選択したり (メニューがポップアップする)、メニューに入力するモード。通常、地図上から情報を読み取るだけであれば、このモード以外のモードになることはない。指の形をしたカーソルとする。なお、ドラッグのときはドラッグのあいだだけ、カーソルの形がつかみの手の形になる。LL に関しては、メニューは透過である。つまり、半透明のメニューの裏の地点を指定できる。

GOM: 幾何図形モード (Geographical Object Mode) 幾何図形を描画するモード。操作者が通常モードからこのモードに移す場合と、メニューの選択によって自動的にこのモードに移る場合がある。幾何図形の入力が終わると自動的に通常モードに戻る。矢印の形をしたカーソル。なお、折れ線を引いているあいだの線は黒の細い実線が引かれる。なお、GOM ではマウスオーバーの吹き出しは起こらない。GOM においては、メニューは透過である。つまり、半透明のメニューの裏の地点を指定できる。

OSM: 向き指定モード (Orientation Specifying mode) 水平に置かれた画面を多人数が周りから見ると、自分のいる向きに文字が出るようにするために、自分が見ている方向を指定するモード。円にトゲの生えた、羅針盤のような形をしたカーソルとする。

NoMにおけるマウス操作の基本は以下の通りである。位置非依存とは、マウス位置に依存しない結果が生じるものである。なお、権限がないとマウス操作をしても意図した結果が起きない。その場合はカーソルが一瞬「？」の形に変わってユーザーに注意を喚起する。

メニューが出ていないような画面上でLをすると、なんらかの地物が選択され、その地物に依存したメニューがポップアップしてくる。もし、選択したつもりのない地物が選択されたらメニューの最上段にある「他の選択」ボタンを押して、所望の地物を選択すればよい。

地図上でLLをすると、NoMからGOMへモードが変わり、そこを起点として幾何図形の描画が行なえる。

NoMのLDではアイコンやメニューボックスの移動などが行なえる。GOMのLDでは、折れ線を自由曲線で書くことができる。この場合、ドラッグが終わったところでも（つまり、マウスを離れたところでも）GOMが終了しないことに注意する。GOMの終了は必ずLLとする。よって、次にLボタンを押したところまで直線が自動的に引かれる。このようにして、自由曲線（を近似した折れ線）とリンクを指定した折れ線のハイブリッドな入力が可能になる。

Rはメニュー内では、メニュー階層を一段上げる。また、地物上では主にマウス操作の様態を制御するメニューが位置非依存でポップアップする。後者の場合、メニューを終了するまで、メニューの外にはカーソルが出られなくなる、あるいはメニューの外にカーソルが出たときにポップアップメニューがキャンセルになる。

RRは、メニューが出ていれば、メニューのキャンセルである。

なお、マルチマウスで使っているときは、他人の開いたメニューへの入力は不可能である。ただし、メニューボックス等の内部へマウスを入れることは可能である（例えば、幾何図形描画のときや、メニューボックスの下の地物を選択したいとき）。

## b) スタイラスによる災害情報へのアクセスと情報入力

現場で使われる機器にはマウスではなく、スタイラスが用いられることが多い。そこでは小画面で使われることが仮定されるので、1つのメニューの中で選択肢は多くて9（3x3）個までを原則とする。本年度の予備実験で用いたPDAであるVaio-Uのように比較的大きな画素数をもっている画面では、外での見やすさを考慮して、選択肢アイコンのサイズは、64x64ピクセルとした。

メニューに表示されるのはアイコンのみとすることを原則とするが、それが難しい場合によっては言葉も活用する。

地図上のオブジェクトの注目と選択は、スタイラスのタップ（スタイラスの先で画面を叩くこと — マウスのクリックに相当）で行なう。その時点でメニューがポップアップし、情報の閲覧や入力が可能になる。メニューにおいて、選択オブジェクトを変更する「他の選択」があるのはマウスと同じである。

マウスと同様、メニューは選択オブジェクトを反映したものがポップアップする。1つのオブジェクトに対して、複数の報告カテゴリがあるような場合、小さな画面でそれらをすべて表示することは難しいので、報告カテゴリ（建物に対する火災、倒壊、要救出者など）を選択するよう

なステップを入れる。すなわち、メニューは「選択されたオブジェクトの種類」、「報告カテゴリ」、「カテゴリ内の選択肢」の3つの部分からなる。

メニューのキャンセルはメニューの外をタップすることでできる（マウスの右ダブルクリックに相当する）。

入力の修正は基本的にメニューを通して行なう。ソフトウェアキーボード類は、メニューの選択を行なって必要になった場合にのみポップアップする。

NoMの画面でダブルタップすると、幾何図形モード（GOM）に入る。スタイラスが画面から離れたところで、幾何図形の描画が終了し、NoMに戻る。マウスと異なり、リンクを次々指定して、折れ線を描くのではなく、スタイラスで自由曲線を描くことによって幾何図形を指定する。しかし、システムは適宜折れ線化し、情報量を減らす。領域は始点と終点を近づければ描くことができる。

位置非依存のスタイラス操作はないので、地図のスクロールや縮尺変更などは、以下のようにスタイラスのドラッグ動作で行なう。

- ・つまみ型の地図のスクロール

画面の端のほうにスタイラスをつけ、そのままスタイラスを中心方向に動かすと、最初スタイラスを置いた地図上の点が、スタイラスの動きに合わせて移動する。つまり、地図の1点をつまんで移動させるというスクロールである。

- ・連続飛ばし型の地図のスクロール

画面の任意の場所でプレスアンドホールド（タップした地点でしばらくスタイラスをとどめておくこと）し、それから少し離れた場所にスタイラスを画面から離さないで滑べらせる（ドラッグさせる）と、最初の地点とドラッグしたあとの地点の距離に応じた速度で画面が（ドラッグした方向に）スクロールする。スタイラスを離すまでスクロールが続く。

- ・地図の縮尺

画面の中心を中心にして、左回りで円をドラッグすると地図が中心を変えずに拡大、右回りに円をドラッグすると地図が中心を変えずに縮小する。回転半径が大きいほど拡大や縮小の比率が大きくなる。なお、この左と右は、すでに Sony の PDA 等で標準になっている向きである。

### c) マルチマウスミドルウェア

災害対策本部や避難所では、多数の人々が大きな情報提示画面を見ながら、情報共有だけではなく、情報の入力を共同作業として行ないたい場合がある。実際、紙地図を使っている災害対策本部では地図を取り囲んで、さまざまな部署の関係者が情報を交換し、有効な対策を練る。近年盛んになってきている災害図上訓練（DIG, Disaster Imagination Game）でも同様である。

しかし、紙の地図上での災害対応作業は煩雑であり混乱も多い。また、地図に記入された情報を広く共有したり、二次利用したりすることにも手間が伴う。この地図を電子化し、災害情報の共有と再利用を促進しなければならない。

従来のほとんどの CSCW（Computer Supported Cooperative Work）では、実空間の共有というより、分散システムを前提とした論理空間における情報の共有を主眼としてきたが、近年、子供の協調作業の教育などに1台のディスプレイを共有して、情報共有することの有効性が見直され

てきている。このような CSCW を SDG (Single Display Groupware) と呼ぶ。災害対応に SDG の考え方を採用すると、1 枚の地図を取り囲むという実空間の共有とあいまって、離れたところで同士で地図情報を共有するよりは緊密な協調作業が可能になる。また、1 枚の画面にたくさんの人が同時に入力できるとすると、少ない計算機資源、すなわち低いコストで効率のよい入力が可能になる。例えば、避難所に来ている多数の人に安否情報を入力してもらうときには、1 台のノート PC に同時並行して入力できるとスムーズである。

災害対応システムという大規模なものが多いが、大規模システムは運用や保守にコストがかかる。SDG を採用することでコストを削減し、誰にでも取扱いやすく、また開発しやすいシステムを実現できる。さらに、どこにでもある装置を用いればよいという特徴は、緊急を要する災害対応に適している。

本研究で開発したマルチマウスミドルウェアは、SDG における複数人による同時入力のうち、特にマウス入力をサポートするものである。1 台の PC に最大 120 台以上の USB マウスをつなげてそれぞれが単独のマウスとして動作する環境をサポートできる。

以下、ミドルウェアについて簡単に説明する。

本ミドルウェアの動作するシステムは Windows XP 以降である。WIN32 API を用い C++ で開発したが、DLL としてコンパイルし、アクセス用のクラスを用意することで、C# でも利用可能である。現在のところ、マウスやトラックボール等の相対座標入力を行なう装置に対応している。ペンタブレット等の絶対座標入力を行なう装置やキーボードへの対応は今後の課題である。

SDG を開発するには、1 台の PC、すなわち 1 本の Windows にマウスを複数つなぐなければならない。1 台の PC にマウスを 2 台以上接続すると、Windows はそれらをまとめて 1 つのマウスカーソルを駆動するためのデバイスとして認識する。このままでは SDG を実現できない。

複数の入力デバイスそれぞれからの情報を取得するための機構として、Windows XP 以降の Windows では、WIN32 API に Raw Input という API 群が用意されている。Raw Input 環境を用い、マウスの動きをアプリケーションプログラムが解析し、その結果に応じて動作すればよい。すなわち、SDG を実現するには処理を一段はさめばよい。

Raw Input 環境で必要になるマウス動作の解析処理は単純なものばかりであるが、実装すると煩雑なものとなる。特にダブルクリックなどはクリックのたびにその時刻まで管理しなければならない。そこで Raw Input の解析処理をモジュール化してまとめ、ミドルウェアとして提供することとした。

今回開発したミドルウェアは、通常の Windows アプリケーション開発に慣れた開発者が容易に SDG を開発できるような機能を盛り込んだ。

- Raw Input データの解析

Raw Input API からはデバイスの生入力データしか取得できない。例えばマウスの場合は単位時間あたりの移動距離とボタンやホイールの状態変化のみである。アプリケーションプログラムからマウスを利用する場合、利用しやすいのはマウスの現在位置やクリック動作といった、ある

程度具体的な動作であるので、Raw Input API からのデータをデバイス毎に累算処理し蓄積するようにした。蓄積するデータは、マウスの座標、ボタンの状態、そしてホイールの位置である。

- ・マウスカーソルの描画

マルチマウスでは、マウスカーソルをマウスの台数分表示しなければならない。しかしながら Windows が描画するマウスカーソルは1つのみである。そこで Windows のカーソルは非表示にし、偽のカーソルをマウスの台数分描画するようにした。マウスカーソルの描画には Layered Window という機構を用いた。Windows 標準のカーソルに比べ遜色のない描画表示が可能である。

- ・マウスカーソルの管理

SDG では多数のユーザが同時に操作を行なった場合の競合や干渉が問題となる。この問題の解決の補助として、ミドルウェアレベルでマウスカーソルを管理するようにした。現時点では、マウス毎に、有効/無効の切替え、移動範囲の制限、スピードの指定、XY 軸の回転角度の指定が可能である。特に移動範囲の制限は、多様な作業レイアウトに対応できるように、任意の多角形で指定できるようにした。

- ・マウス毎のイベント管理

Windows と同様、アプリケーションプログラムはミドルウェアへコールバック関数を登録し、ミドルウェアからのイベントの受渡しはこのコールバック関数を用いるようにした。

- ・子ウインドウの管理

ボタンやメニュー、スクロールバー等、従来のシングル入力用コントロールはそのままではマルチ入力環境で利用できない。本ミドルウェアには、従来のコントロールに向けてウインドウメッセージを合成する機構を実装し、あたかもシングル入力環境で動いているように、コントロールを「騙す」ようにした。

- ・ログ記録、ログ再生

デバッグおよび統計処理のため、マウスの動作のログをとる機能を実装した。記録される情報はイベントすべてとそれらが発生した時刻（ミリ秒単位）である。ただし、マウス移動イベントはすべてを記録すると膨大な量になるため、50 ミリ秒間隔の記録とした。ログの再生は、記録されたイベントを逐次アプリケーションプログラムに届けることで行なう。リアルタイム再生と、時間軸を無視した論理的再生とを可能にした。

本ミドルウェアを使用した場合のプログラムの流れを簡潔に説明する。あるウインドウについてアプリケーションプログラムが本ミドルウェアを使用するように登録（register）すると、そのウインドウに Raw Input メッセージが届くようになる。また従来のウインドウメッセージは届かないようにすることができる。

次に、アプリケーションプログラムは、そのウインドウに届いた Raw Input メッセージを逐一ミドルウェアへ通知する。ミドルウェアはこれを解析し、カーソルを描画し、イベントが発生した場合はコールバックによりアプリケーションプログラムへ通知する。

シングル入力の場合は、ウインドウの WndProc() が入力メッセージを受け付け、適切な処理を行なう箇所であるが、本ミドルウェアを利用する場合、WndProc() は Raw Input メッセージをミ

ドルウェアヘフオワードする関数として用いる。代わりにマルチ入力用 WndProc() となるコールバック関数を用意し、入力の処理はそちらで行なう。以上の流れを図示したものが図 6.2-8 である。なお、上記以外についてはシングル入力の場合とほぼ同一である。

このミドルウェアは、イベント処理や子ウインドウの管理などを盛り込み、シングル入力のアプリケーションを開発するのと同様に SDG アプリケーションを開発できるようにした。今後一般公開を行なう予定である。

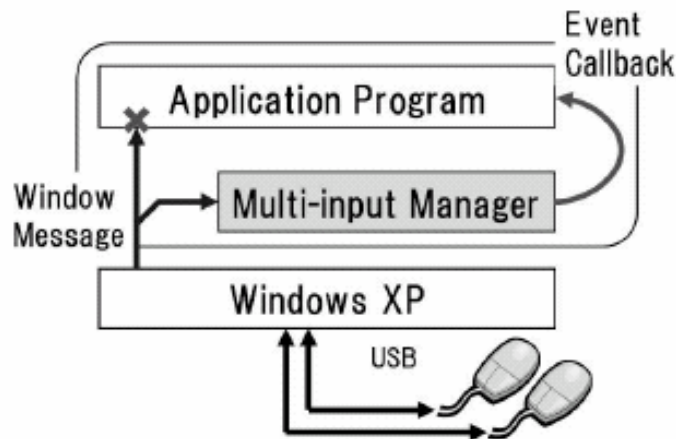


図 6.2-8 ミドルウェアの概念図

### (3) 情報提示・入力ガイドラインに基づいたシステムのフィールド実験

前述した情報提示・入力ガイドラインに基づいて、災害対策本部、現場、避難所の3通りのシステムを開発し、実験を行って実際の使い勝手等について検証を行った。ひとつはマルチマウスの応用例である避難所安否入力システムの実験、もうひとつは DaRuMa を経由した災害対策本部と現場の情報共有を支える報告・指令のシステムの実験である。

これら3つのシステムを、2005年11月の愛知県豊橋市での予備実験、12月の岩手県滝沢村でのリスクコミュニケーション実験、2006年3月の新潟県見附市での予備実験に持ち込み、その都度、ログデータ等を集積し、改良を進めた。特に見附市の予備実験では、それまで想定していた震災とは異なる水害を対象とした。

#### a) 避難所安否入力システム

これは災害発生時に避難所で住民の避難状況や安否状況を把握するためのシステムである。

災害発生時に設けられる緊急避難所では、避難民がどこから避難してきたか、避難元の被災状況などを、白地図や名簿を用いて把握する作業がある。本システムはこれを電子白地図上で行なう。図 6.2-9 に豊橋市飽海地区における避難所安否情報入力システムの画面を示す。

本システムの特徴は次のとおりである。

- ・情報を柔軟に利用できる。電子地図の情報を使い、入力された避難情報を即座に管理できる。

- ・ 並列作業ができる。（避難情報の地図への入力と比較的小規模な作業なので、複数人で同時に作業したほうが効率が高い）
- ・ 運用が容易である。低コストの SDG であるので、どのような場所であっても、PC が一式動きさえすれば運用できる。

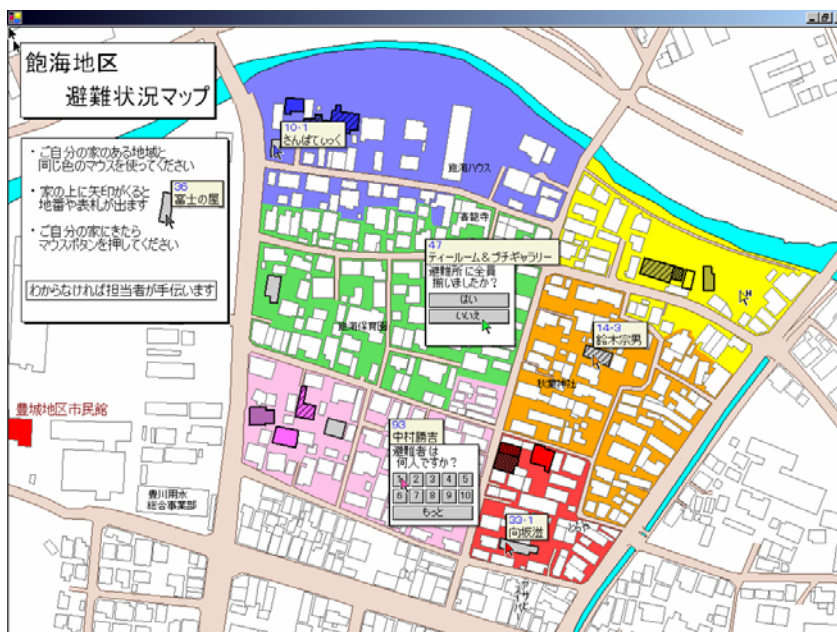


図 6.2-9 豊橋市飽海地区における避難所安否情報入力システムの画面

複数のマウスを用いる場合は、地図を小さな領域に区切り、それぞれの領域に一つずつマウスを割り当てることができる。これにより、例えば同じ町の何丁目から避難してきたかで使用するマウスを分ける、といったことが可能になる。図 6.2-9 では町内が 6 つの分会に色分けされている。

地図データ等は GML フォーマットのファイルを利用するようにした。住宅地図を用いると、入力対象となる建物の地番や表札の情報を表示することができる。集合住宅の場合は部屋毎に情報を表示することもできる。

本システムを用い、2005 年 11 月 20 日に豊橋市飽海地区で開催された災害対応訓練と、2005 年 12 月 4 日に岩手県滝沢村で開催された消防団班長研修会とで入力実験を行なった。

飽海地区では地元住民約 30 名にご協力いただいた。参加者の年齢は小学生から 70 歳以上の高齢者まで、幅広くまちまちであった。滝沢村では消防団班長たち 43 名にご協力いただいた。滝沢村での参加者の年齢は 30～50 代である。消防団班長の多くは普段から PC に触れている方々であり、並列入力が非常にスムーズであった。直後のアンケートによれば、入力が容易が 65%、やや容易も含めれば 84%で、入力インタフェースの設計として成功であったと言える。マウス操作に慣れている人たちがこの並列入力システムを使って、40 件以上の入力を 5 分で終わることができた。



## b) 報告・指令システム（災害対策本部と現場の情報共有）の技術概要

システムは2つのコンポーネントからなる。1つは災害対策本部に置くことを想定した大画面情報提示・入力システムであり、もう1つは災害現場に赴いた自治体職員や隊員が持ち運ぶ携帯端末を用いた情報提示・入力システムである。どちらも最終的にはMISPでDaRuMaに接続し、DaRuMaを経由して情報のやりとりを行なう。携帯端末はいろいろな無線通信インフラを使用してネットワーク接続をする。豊橋市ではb-mobile PHS データ通信を用い、見附市では無線LANを用いた。図6.2-10に災害対策本部と現場の情報共有システムの構成を示す。

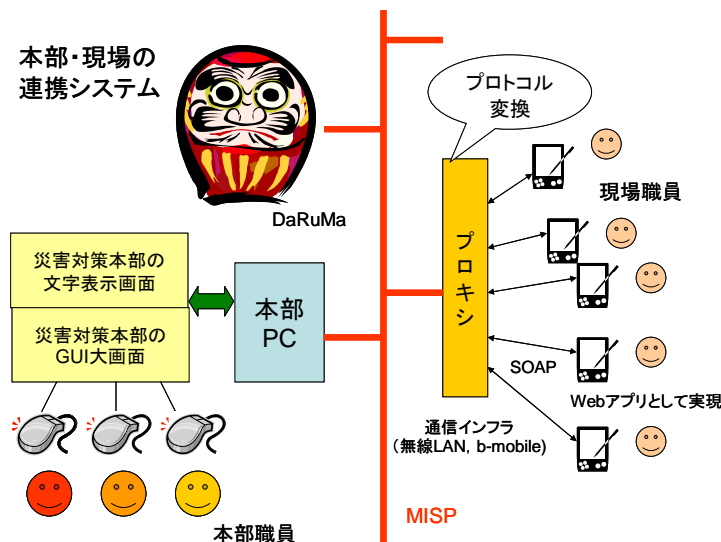


図 6.2-10 災害対策本部と現場の情報共有システムの構成（見附市のもの）

災害対策本部のシステムは、2560×1600 の解像度をもつ液晶ディスプレイ、あるいは 1400×1050 という標準より高解像度の液晶ディスプレイを用い、広い領域を細かく見られるようにした。開発言語は地図などの描画、デバッグ環境の優れた C# である。

また、複数の災害対策責任者が SDG によって画面を共有するために、マルチマウスを用いた。また、アクセス権限がユーザごと、すなわちマウスごとに独立に付与できるので、例えば、消防関係者と土木関係者が職掌範囲を越えて間違った入力をするのを防げる。

画面共有をしているときに、画面の縮尺を変えたり、画面のスクロールを行なうのは他の人に影響を与える。当面、特権をもつユーザのみが声をかけ合って画面制御を行なう方式にした。

災害対策本部システムは、DaRuMa に最新情報が入っているかどうかを定期的にポーリングする。そのため、現場からの情報や、他機関からの情報の表示には数秒～1 分の遅延が加わる。

災害対策本部システムは DaRuMa から情報を取るだけでなく、本部画面から防災担当責任者が現場に指令を出した場合、Fax や電話など電子的な手段以外で本部に入ってきた場合には、災害情報を本部で直接大画面上で入力する。そしてそれらの情報が DaRuMa に格納される。

現在、災害対策本部システムのインターフェースはマウスのみであるが、マウスによる入力から、言語情報を抽出するサブシステムを開発した。見附市の予備実験では、災害情報や対応行動情報を電光ニュースのように大画面の横に簡単なテロップで流した。地図上に描いた幾何図形から町

丁目を自動抽出し、面積とそこに含まれる建物の戸数（例えば、床上浸水家屋の戸数）を地理データベース上の空間情報演算によって求めて表示している。

現場職員が持ち運ぶ携帯端末はプロトタイプ開発の便を考慮して、Windows XP 開発環境が整っていた Sony Vaio-U を用いた。しかし、OS や実行環境に依存しない Web アプリケーションの枠組で開発することにより、携帯電話への移植性に配慮した。従来の Web アプリケーションに対して、Ajax, SVG (現在、フルセットは Mozilla Firefox のみが対応) を使用することで操作性、対話性、描画性能で大きな進化をした。

Web アプリケーションと情報共有プラットフォームをつなぐのは現場職員対応プロキシである (PHP で書かれている)。ここでプロトコル変換をするとともに、現場職員からの報告を DaRuMa に報告し、DaRuMa から最新の災害情報や指令を伝達する。なお、プロキシと携帯端末のあいだは SOAP を用いた。これは Web サービスのデファクトスタンダードのプロトコルである。

現場職員が災害対応をしながら持ち歩く携帯端末では複雑な操作が期待できない。そのため、地図を第一義とした GUI をベースに設計した。我々の提案した情報提示・入力ガイドラインにより、メニューや付加情報は必要なときのみポップアップ表示する。また、スタイラスのみですべての操作を行なえるようにするために、ジェスチャ操作を全面的に採り入れた。

災害対策本部と同様、地図上に災害情報を重畳表示し、GUI メニューを用いて災害情報を報告するとともに、周辺の現在の災害状況を他職員の位置を受信して表示する。さらに、本部からの指令も地図上に表示して、指示の了解性を高める。

プロキシはフリーの WebGIS である MapServer と PostgreSQL の拡張版である PostGIS によって地図画像を生成して携帯端末に配信する。これは通信量を要するので、携帯端末は必要な部分のタイル (200×200 ピクセル) のみをインクリメンタルに取って通信窒息することを防いでいる。

### c) 災害対策本部と現場職員の情報共有実験 — 豊橋市予備実験

上記のシステムを用いて 2005 年 11 月 20 日の豊橋市防災訓練で、災害対策本部と現場職員の情報共有実験を同市豊城地区公民館で行なった。本部は消防、救急、道路啓開担当の土木担当の 3 名である。飽海町を巡回する隊員 (現場隊員) は消防 2 名、救急 2 名、土木 1 名である。本部には 2560×1600 ピクセルの高精細ディスプレイを置く。それを本部要員 3 名がマルチマウスで共有する。現場隊員 5 名はそれぞれ携帯端末 (Vaio-U) を携えて飽海町町内を歩いてもらった。

災害シナリオは竹内が当日未明に作成し、実験開始時に隊員に封筒に入れたまま渡した。シナリオは隊員の初期位置、発見すべき災害、共通して認識すべき建物被害、道路閉塞などが書き込まれた飽海町の地図である。隊員ごとに発見できる被害現象を変えた。また、本部要員にはシナリオを与えなかった。図 6.2-11 に隊員の PDA、図 6.2-12 にその画面を示す。

隊員からの報告は、地点に関する報告、道路沿いなどの折れ線に関する報告、一つの街区全体などの領域に関する報告の 3 種類がある。実験のために用意した単体 GPS の精度と応答が非常に悪かったため利用をあきらめ、隊員は自分の位置を地図と対照しながら随時本部に報告 (一種の Keep alive プロトコル) することにした。つまり、スタイラスで自分の現在地点を叩き、そこにポップアップしてきたメニューの「報告」ボタンを叩くだけの簡単な報告である。



図 6.2-11 隊員の PDA

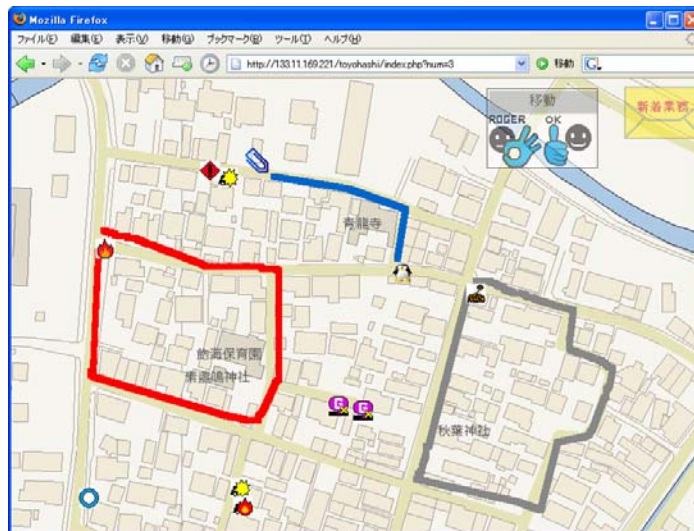


図 6.2-12 PDA の画面

すべての Vaio には PHS データ通信カードである b-mobile を装着し、本部との通信はすべてそれで行なった。念のため携帯電話による対話も必要なら行なえるようにした。

本部には飽海町の詳細な地図が表示されており、隊員から報告があるたびにアイコン、アイコン付き折れ線、アイコン付き領域が描かれる。折れ線や領域には、今回設計した簡単な文法（色、形、太さなど）を適用し、少なくとも慣れるとすぐ意味がわかるようにした。実際、アイコン上にマウスカーソルが来ると、その簡単な意味が吹き出しで出るので、意味がわからなければそこへカーソルをもっていだけでよく、習熟は容易かつ早かった。なお、隊員からの最新報告は数秒間ブリンクさせて本部の注意を喚起した。アイコンは建物被害（消防研作成）を除いて、すべて研究室内で自作した。図 6.2-13 に本部の画面（縮小版）を示す。

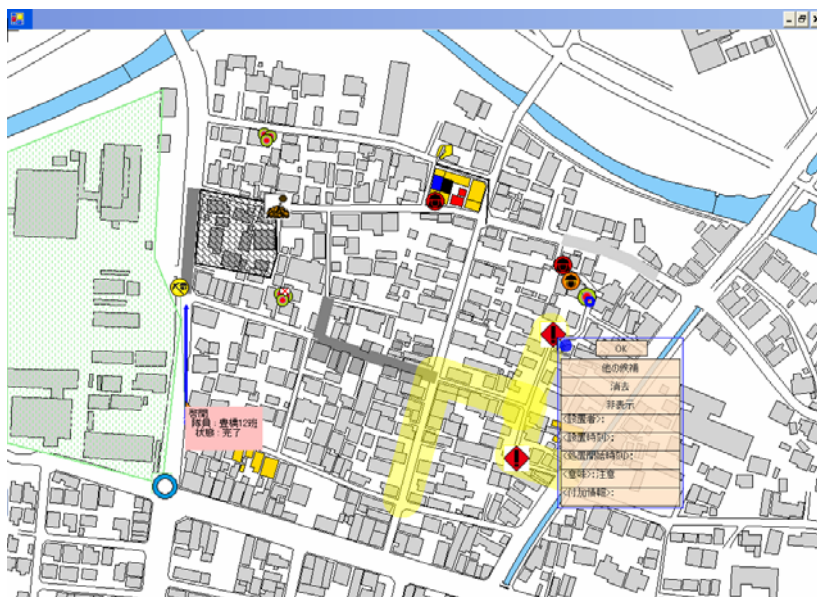


図 6.2-13 本部画面 2 (一部分) 災害対応が動き始めている。

本部は報告された状況を見て、どこそこ方面 (領域指定) を調べよ、(必要なら、折れ線を使って経路指定をして) 消火に急行せよ、この道路の閉塞を啓開せよ、などといった地物指定あるいはユーザ定義地物指定を伴って指令を現場に出す。本部から指令が来ると隊員の PDA にはメールボックスが現れ、ビープ音が鳴る。これに「指令了解」を返信し、現地に行ってから命令が完了したか失敗したかを報告する。必要な情報は他の隊員の PDA にも同報される。

本部は別のタブレット PC に装着した b-mobile を使って現場と通信する。PHS しか使わないことによって、災害現場における非常に悪い通信環境を再現することが可能になった。実験は、本部から帰着指令が出て、それを全員完了することによって終了した。

豊橋市での予備実験の結果、以下のことが達成あるいは発見できた。

- ・ 火災は 2 ヶ所とも消すことができた。つまり、消防隊員以外による発見報告、本部から消防隊員への消火指令、消防隊員の移動・消火完了報告の一連の流れが達成できたということである。
- ・ b-mobile によるパケット通信は片方向の通信で 30 秒ないしは 1 分もかかることがあった。つまり、隊員の報告が自分の画面に反映されるのに、往復の通信遅延と DaRuMa のポーリングによる遅延とで、最大 2 分かかった。b-mobile はスループットはあっても、応答時間がほとんど保証されないということであろう。

#### d) 災害対策本部と現場職員の情報共有実験 — 見附市予備実験

豊橋市予備実験のさまざまな反省を踏まえ、対象とする災害が水害という展開になったことに対応して準備を進めた。システムの基本的な枠組を変更する必要はなく、流れる情報の種類とアイコンを追加したのみである。図 6.2-10 は見附市での実験システム構成を示している。

見附市の予備実験では、時間と場所の制約があったため、本部と現場の通信は近距離無線 LAN を用いた。また、ここでは予め用意したシナリオに沿って情報共有の進行をスムーズに見せるというスタイルをとった。

デモでは高精細 (1400×1050) の本部大画面と、通常の XGA プロジェクタを用いた携帯端末画面の拡大表示を両方用いて本部と現場の挙動を対照的に示した。この結果、まったく異種の言語で書かれた 2 つのシステムが、本部と現場においてそれぞれの環境に応じた入力や情報提示を行えることを示すとともに、これらが実際に通信 (無線 LAN) を通じて情報共有できることを確認した。