Representing Expectations in Spatial Information Systems

A Case Study

Graham J. Williams[1] and Steven G. Woods[2]

[1] Centre for Spatial Information Systems,
CSIRO Division of Information Technology,
GPO Box 664 Canberra 2601 Australia
Graham.Williams@csis.dit.csiro.au

[2] now with Command and Control Division
Defense Research Establishment Valcartier
2459 Pie XI Blvd., North,
Courcelette, Quebec, CANADA G0A 1R0
Woods@drev.dnd.ca

Abstract. Expectations are important in reasoning. They provide a framework within which decisions can be made. This paper describes a spatial information system which couples a large object-oriented spatial database with a graphical user interface, and incorporates ideas from artificial intelligence research, allowing the representation and use of expectations. In particular, the decision support role of spatial information systems is enhanced by the incorporation of reasoning with justifications and truth maintenance. The resulting system facilitates the assimilation, handling, and access to large amounts of dynamic and static data for decision support. Inferential information of an application can be recorded, monitored, and updated.

1 Introduction

Spatial Information Systems (SIS) are information systems which deal with the storage, manipulation, and display of spatially indexed data. Geographic Information Systems (GIS) are those SIS which particularly pertain to geographic applications. They are used by decision makers to access large amounts of stored data, to view the spatial relationships between the data, and to provide basic analyses of the data.

SIS have tended to be static tools for viewing data. The user of the system specifies data to be retrieved and displayed, specifies operations to be performed using that data, and then commits new or modified data to a persistent store. The systems are generally not actively involved in the decision making process other than to provide the data on which decisions might be made.

In this paper, we describe an interactive SIS designed to allow the input, display, manipulation, and storage of data in an object-oriented database. We view the database as an on-line system, continually being updated as new information arrives. The system described here implements a pro-active extension on top of the SIS/database which introduces the concept of Expectations. Expectations provide a decision support mechanism by which the SIS can actively monitor its own operation in order to recognise the occurrence of significant events, such as the entry into the database of important information. These expected events can either be described by the user of the SIS or automatically generated by the SIS itself. In the latter case, the concept of templates, which represent typical patterns of both spatial relationships and temporal progressions, are introduced.

A particular domain example is used here to illustrate the functionality of Expectations. The example is one of a class of generic problems which can be characterised by their requirement that they dynamically handle real-world entities whose identities are often uncertain and whose locations typically change over time. An example drawn from the military domain is used.

Section 2 describes this domain example. Sections 3 and 4 identify both the generic and domain-specific components of the basic SIS which has been developed. Section 5 introduces the concept of Expectations, and describes how they are supported in the SIS. Section 6 introduces the idea of automatically identifying expectations.

2 An Overview of the Problem Domain

TMIPS (Tactical Military Information Processing System) is a decision support system for military intelligence assessment. It has been developed jointly by the Australian Government's Commonwealth Scientific and Industrial Research Organisation (CSIRO) and the Defence Science and Technology Organisation (DSTO).

The identification and location of opposing military units is of prime concern for an intelligence officer during military conflicts. Information is received by the intelligence officer in the form of messages from field observers. These messages contain information about the location and constitution of opposing forces, and often have some amount of uncertainty and error. A particular message records information about the identity and purpose of units sighted at a *single* location.

A map forms the key tool used by an intelligence officer. The field observations are marked on a map as they arrive. The terrain, the location of roads, cities, etc., as recorded on the map, play a vital role in understanding the position of the sighted units, and in predicting their possible courses of action.

The intelligence officer's vast knowledge and experience is used to explain and predict the behaviour of the opposing forces. Detailed knowledge of the hierarchical structure of the opposing forces is used in attempting to identify the units sighted in the field. This information is specific to an opposing force and contains details about the equipment, size, and structure of units, and their relationships with other units. The intelligence officer also uses domain knowledge in the form of military doctrine which describes the behaviour of a particular opposing force. Both the hierarchical unit knowledge and doctrinal domain knowledge are non-monotonic, being subject to continual refinement as more is revealed of the opposing forces.

Mechanisation of the intelligence recording and structuring process is quite complex and presents some difficult problems in knowledge representation as well as in information retrieval and display. While there is often a great deal of information available, it may be incomplete, and sometimes inaccurate. For example, from the description of equipment contained in a message from a field observer, an intelligence officer might identify several possible units which contain such equipment and could potentially be situated at the reported location.

In interpreting a single sighting the global picture plays a very significant role. The volume of information (messages, domain knowledge, maps, satellite images, etc.) which is available when interpreting a message can be overwhelming, even for an experienced officer. This information overload can easily lead to important information being overlooked, and is a significant problem for any domain involving large amounts of data.

The SIS must assist the user in dealing with competing information, particularly as it pertains to the existence and identity of sighted units. The SIS must simplify the task of collating and organising the information as it arrives rather than adding unwanted complexity to an already difficult problem. In addition, the SIS must not allow important information to become lost in the mass of data in the database or the confusion of symbols on the map.

TMIPS is a SIS which assists the intelligence officer in the task of accumulating, visualising, and interpreting large amounts of spatially oriented data. This example military domain provides a fertile environment in which the effective representation and visualisation of data, and developments in spatial reasoning, can be explored.

3 Representation and Visualisation

An important goal of any spatial information system must be to capture the domain nomenclature and symbology and to use them in the human-computer interface. The military domain has a rich set of terms and symbols which are used to capture and communicate concepts and ideas. Whilst the SIS must support these, the underlying framework must be generic, allowing different domains to be easily accommodated.

An object-oriented approach for modelling is adopted for all aspects of the developed SIS. The data required for TMIPS includes the predominantly static information about terrain, and the location of key facilities like roads, rivers, and populated areas, and the predominantly dynamic information about the location of the military units. Each of these are modelled as objects, and are persistent in an ONTOS (Ontos 1991) object-oriented database.

This section describes both the generic (object-oriented) toolkits which make up the SIS and the domain specific structures (also object-oriented) layered above the toolkit.

3.1 Underlying Toolkits

Two important toolkits for the display and manipulation of objects for map-based applications have been developed by the Centre for Spatial Information Systems. Xs (Milton and Campbell 1992) is a graphic object mapping system based on the X11 windowing system. The Spatial Object Library, or SOL (Milne, Milton and Smith in print), is a toolkit for representing spatially located objects. SOL provides a persistent representation of map objects (roads, units, etc.) through an ONTOS database, and Xs is used for the display of these objects. Intermediate between SOL and Xs is the Spatial Object Display Library, which essentially provides the interfaces for the persistent SOL objects implemented using Xs. Textual interfaces have also been developed, allowing objects to be viewed both textually and graphically.

Objects that are to be represented by symbols on a map must have attribute information describing their location. Some objects, such as those representing topology, will have static, and specific locations, but the objects of primary interest to an application will typically have locations which are dynamically updated. The location of these latter objects is often not particularly precise (referring, for example, to field observations of moving entities). A simple approach to location specification is adopted. Location is represented by specifying up to four map references together with a qualifier. The map references identify either grid squares (GS) or grid references (GR), the latter being points on the map. The qualifier is one of P (for point), L (for line), and A (for polygon area). When displaying an object on a map, some simple heuristics are used to position the symbol.

Typical map entities (roads, cities, forests, rivers, etc.) are represented by SOL objects, which in turn may contain attribute information or collections of other objects. For example, a river system may be composed of many river objects which in turn may consist of many feeder stream objects. Application information is essentially represented in the same way. Objects created by the application will similarly have relationships amongst themselves.

Objects can typically be displayed graphically (as a symbol on a map) or textually (as a text window). Figure 1 illustrates this for the TMIPS application. Each display representation of an object is identical from the point of view of the persistent object it represents. Notionally, the only difference between a text-based window interface to the data object and an Xs realised interface for this same object is implementational. Changes to the real object in the persistent database are reflected in all display objects. A military unit identified as a Tank Regiment, for example, will have a particular symbol associated with it when displayed on a map, while a text-based display might describe the unit in terms of key attribute text values. In either case, any display changes required for a persistent object are handled locally by the interface and are thus detached from the object itself.

An application built upon these toolkits provides a display for persistent object information, using the layered systems architecture illustrated in Figure 2.

Depending upon the actual application, the predominantly static type of data usually forms the base of the graphical (map) display. The dynamic data is typically displayed as a layer above the base map display. With the large amount of data available, retrieval and display of the data must be detailed enough to capture the information required by the user, and yet sparse enough to be easily comprehensible. This goal is accomplished in at least two ways: focus of attention, and smart map displays.

Focus of attention is achieved by using the graphical display (usually a map) as a summary or index of the objects of interest. The textual display allows detailed information about an object to be viewed and edited. This detailed information can be displayed by selecting the corresponding symbol on the map. This type of information

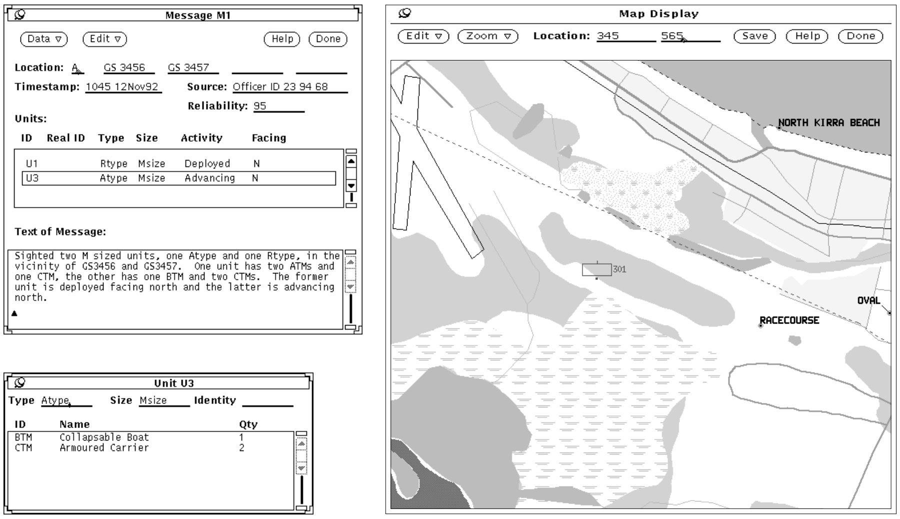

Fig. 1. Textual and graphical display of Object data. The rectangular icon near the centre of the map identifies the location referred to in the message. The boxed unit (U3) in the Message window identifies a selected Unit, which is displayed in a separate window.

hiding is effective in minimising information overload.

Smart map display is implemented in Xs using *intelligent display objects*. These objects monitor the current level of clutter around them in a particular map display, and in cases where the overlap of display objects is determined to be too high, certain objects choose to make themselves invisible. One example of where this approach is beneficial is when a map is zoomed out. Display objects possess a predetermined priority, and lower priority objects disappear. As the map is zoomed in, more detail appears as display object density decreases. The objects themselves typically determine when the map user is likely to be interested in seeing the greater detail. Typically, dynamic application objects remain displayed whilst less important labels, and other less critical details may disappear as the map is zoomed out.

The concept of multiple, object independent, and yet integrated display types, such as linked textual and graphical interfaces, transparently provides flexible support to the decision making user of the SIS. The unification of base map objects and application specific objects in a single persistent object store allows for a seamless integration and specification of relationships between objects of both types.

3.2 TMIPS and Messages

The toolkits described above form a solid basis for the development of an SIS. These toolkits provide much of the functionality often found in GISs with significant extensions based upon the object-oriented approach. TMIPS is an example of an application built upon these tools.

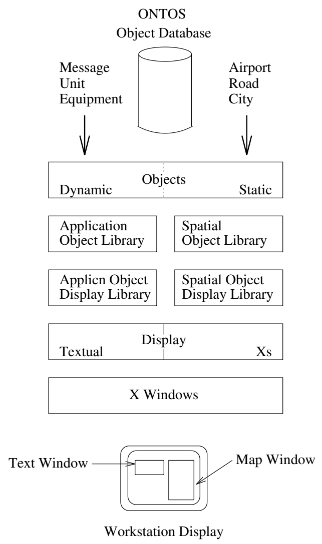

Fig. 2. Visualising persistent data textually and graphically.

For TMIPS, the static data identified above consists of map-based objects such as topology. The domain specific dynamic data consists of objects such as messages from field observers. Messages are in fact one of the primary domain objects introduced in TMIPS. They form the basis upon which the user (the intelligence officer) builds a model of the unfolding proceedings and makes predictions about future events.

Messages are reported in to the intelligence command post from field observers, and are entered into TMIPS by a clerk. Each message contains a time stamp, the identity of the sender of the message, an indication of the reliability and accuracy of the message, and details of the sighting itself, including the location, the type of activity observed, and descriptions of each unit at this location. Each Message may describe several sighted units, and each unit may be identified to be of a particular size and type, and having some set of equipment associated with it.

Figure 1 illustrated both a textual and a map representation of a message. The location of the reported sighting is recorded on the map using an icon to convey information about the Message object.

4 Recording and Justifying Conclusions

As described so far, TMIPS is a flexible information display tool. Messages can be recorded and displayed on maps or viewed and edited textually. To enhance its usefulness in its support of decision making, TMIPS allows inferences made by the intelligence officer to be recorded. These inferences are termed Conclusions, and are the second primary dynamic data object introduced by the TMIPS application.

TMIPS represents and displays Messages that quite possibly describe sighted units incompletely and ambiguously. The intelligence officer is responsible for interpreting the message, in the broader context of all current activities, in order to draw conclusions about the actual identity of the reported units. Often, actual identification cannot be made, but alternate possibilities might be proposed. TMIPS Conclusions capture the intelligence officer's inferences from Messages.

With Messages as the starting point, an intelligence officer uses domain knowledge (opposing forces hierarchy and military doctrine) to transform one or more Messages into a Conclusion. Alternative inferences about the identity of the units reported in a Message are recorded in a Conclusion as a set of Configurations. Each Configuration records one possible grouping of military units which would explain the particular sightings recorded for that location. We note here the possibility of many field observers reporting messages describing sightings at common locations. The intelligence officer can fuse these Messages into a single Conclusion, containing some number of Configurations.

A Conclusion can also be derived from other Conclusion(s), or a combination of Conclusions and Messages. Such derivations reflect the intelligence officer's evolving understanding about the configuration of the opposing forces. Figure 3 is an example of the formulation of a Conclusion which indicates that within the area delineated by grid squares GS3456 and GS3457 a sighting was made, and that sighting can be explained by one of the two Configurations listed. Configuration 1 consists of Unit 1 and Unit 3, and Configuration 2 consists of Unit 2 and Unit 3. This ambiguity can be resolved later in other Conclusions as the intelligence officer learns more about the situation. A Conclusion can also suggest other information that needs to be gathered in order to clarify the situation. In this instance any intelligence gathering that would resolve the presence or absence of Unit 1 or Unit 2 would result in the removal of one Configuration, resulting in less ambiguity.

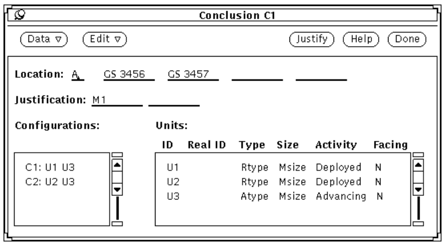

Fig. 3. The textual display of a Conclusion object.

Relationships are maintained between Conclusions and Messages. A relationship (which is a linkage) records the Messages and Conclusions which were considered significant in arriving at a particular Conclusion. These linkages define a Justification Network from which explanations of the reasoning leading to the assertion of a particular Conclusion can be composed. Figure 4 shows the explanation of a particular Conclusion, C3, consisting of the replacement of two previous Conclusions, C1 and C2. These in turn were derived from the original Messages M1 and M2. Each *replace* link can hold detailed information about the reasons or justification for the replacement. For example, the user may recognise that two Messages are in fact descriptions of the same entity, and may replace them with a single Conclusion. The links between the original Messages and the Conclusion may contain information capturing the fact that two Messages were simply fused into a single Conclusion.

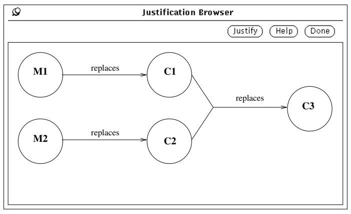

Fig. 4. The Justification Network browser. Links represent dependencies between objects. Selecting an object results in the display of its textual window.

The Justification Network can be used to monitor the validity of Conclusions. At any time, a Conclusion can be refuted on the basis, for example, of new information or insights by the user. The refuted Conclusion will have a new link to the Message(s) and/or Conclusion(s) which lead to its refutation. When a Conclusion is refuted, any other Conclusions justified by this one, either directly or indirectly, must also be reconsidered. Such dependencies between objects can easily be followed using the Justification Browser. This can lead to Conclusions being withdrawn or otherwise justified. In either case new relationships are introduced and may be displayed in the Justification Browser.

In summary, Conclusions facilitate the intelligence officer's difficult task of assimilating and integrating the incoming reported sightings. Distinct messages, having locations in close proximity, can be quickly identified on the map, and drawn together into a Conclusion about the activities at that location, removing unnecessary redundancy and clutter. (TMIPS can be commanded, for example, to only display objects which have not been "replaced" by some Conclusion.) The dependencies between objects are visualised with the Justification Browser, providing easy access to related Conclusions, ensuring that the impact of new information can easily be propagated through the system.

5 Expectations

Often, in analysing a situation, and in particular when the situation is one that is dynamically changing, the user will identify the likely location and identity of further, as yet unreported, units. In addition, with the user's experience and knowledge, likely scenarios will be developed which will predict future movements of already reported units. In either of these situations, the user would be significantly assisted if they were able to record such predictions or expectations in the information system. The system could then automatically monitor new messages as they arrive, to determine whether the expectations have been met, or indeed, whether certain expectations cannot be met. The Expectations Model described here provides a facility for recording such speculation, enhancing the general functionality of spatial information systems.

5.1 Overview

Two types of Expectations are observed: expectations about the location of yet to be reported units, and expectations about the movement of currently reported units.

The first type of Expectation arises when a pattern in the layout of the reported units is recognised, leading to the hypothesised existence of other related units. For example, the reported sighting or conclusion of two units which typically form a larger unit when a third unit is also included, would lead the user to the expectation that the third unit will be located somewhere nearby.

The second type of Expectation hypothesises the future location of units already reported. For example, the observed relative location of a number of reported units may form a recognisable pattern which is associated with a particular strategy. Recognising this pattern would lead the user to identify one or more expectations about where particular units may be located in the future.

In TMIPS both types of Expectations record the prediction of a new Message/Conclusion entering the system. The automatic confirmation or otherwise of an Expectation must provide the user with the prompts and information needed to amend and extend their analysis of the proceedings of events occuring in the field. They will also be of assistance in keeping a check on various lines of thought. As the user's expectations are confirmed or disaffirmed further Expectations will arise.

Expectations in TMIPS provide a placeholder for the hypothesised locations of opposing forces and assist the intelligence officer in building an overall battle picture. In a hypothetical based reasoning system it is conceivable that multiple competing arguments or justifications for the location of opposing units exist. Individual Expectations are placed within the Justification Network, with links to supporting Conclusions and Messages, and to other Expectations.

Expectations are active objects—Expectations monitor the arrival of Messages and Conclusions for any that *match*. If a match occurs, the user is immediately notified with an Alert. The user will examine the Alert to determine whether a valid match occurred. If so, the evolving picture of the situation can be updated, with appropriate additions to the Justification Network.

Within TMIPS, an Expectation is represented as an object having a form similar to that of a Conclusion (and in fact inherits from the same base class object). Thus, an Expectation has associated with it a location and a description of the units expected to be found at that location. In making predictions though, time is also a factor, and so an Expectation also possesses a time range for which it is valid.

5.2 Recording an Expectation

An Expectation is entered into the system using an interface similar to that used for entering Conclusions. An example of an Expectation Edit Window is illustrated in Figure 5.

In practice, an Expectation will usually be derived from one or more Conclusions, with the relevant Conclusion data providing the initial and default values for the Expectation.

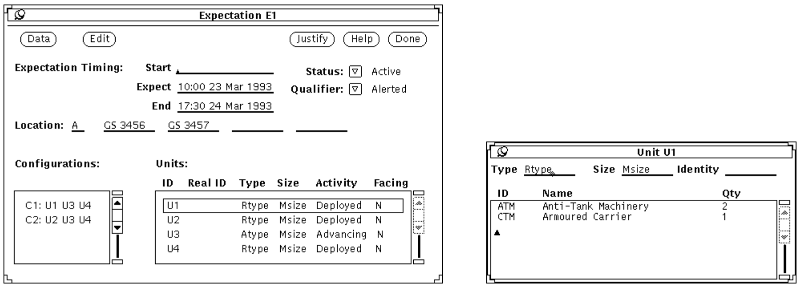

Fig. 5. Expectation Edit Window and Unit Edit Window.

As with Messages and Conclusions, the Expectation refers to a particular location described by up to four grid cell references. In common with Conclusions, the Expectation has associated with it a description of the units. Alternative configurations are permitted, identifying some degree of uncertainty of the identity of the units involved.

An Expectation, however, need not have any configurations associated with it. In such a case the Expectation anticipates some unit to be sighted at the specified location, without specifying the actual unit composition. This allows the user to identify a location which is to give rise to an alert immediately when some activity occurs at that location.

Similarly, an Expectation need not have a location specified, having only configurations and units. This Expectation anticipates some activity associated with particular units, without actually specifying where the activity is to occur. Thus, any relevant activity reported to the system as Messages or Conclusions will be brought to the attention of the user. This allows the user to monitor sightings of important units.

Additional information that an Expectation carries includes a Status and Status Qualifier recording the state of the Expectation, and a time-range within which the Expectation is predicted to occur.

An Expectation is either Active or InActive as indicated by the Status. The Qualifier identifies further information about the status. Figure 6 shows the menu associated with the Qualifier. This menu, and another associated with the Status, permit the user to update the Expectations' status. Expectations are recorded as Active if they correspond to a current valid prediction. Active Expectations monitor new or changed Messages and Conclusions, alerting the user of any matches. The Qualifier of an Active Expectation is automatically changed to "Alerted" when such a match occurs. When the user determines that the Expectation is of no further use, it will be flagged as InActive. For archival and explanatory purposes all Expectations are persistent, even if they are inactive. An Inactive Expectation can be qualified as Satisfied if it has been entirely satisfied by one or more Messages or Conclusions. It can also automatically time out, in which case the user is alerted and must determine whether or not the Expectation is of any further significance. Possible actions at this stage include leaving the Status as InActive, or editing the Expectation's period of validity, and thereby re-activating the Expectation. Alternatively, an Expectation can be recorded as having been replaced by another Expectation or as no longer required, for whatever reason.

The time specified on an Expectation has three components. The Start Time refers to the earliest time at which a sighting is anticipated. When this time arrives, the Expectation is changed to Active and it will then attempt to match any subsequent Conclusions or Messages. The Expect Time is the most likely time for this particular sighting to occur. Matching could take into account the difference between this Expect Time and the sighting time of a Message by considering that certain activities have a regular nature and sightings near certain times have some explanations that are more likely than others. The Expectation End Time indicates the latest time a sighting is expected. Once this time elapses, an Expectation alerts the user to the expiry of its validity and the user can decide whether to re-Activate the Expectation or not.

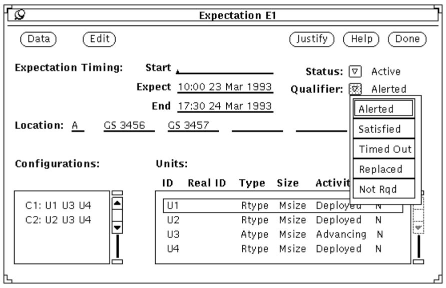

Fig. 6. Expectation Edit Window showing Status Qualifiers.

Each Expectation will be a node in the Justification Network. Links identify the supports for an Expectation, which may be Messages, Conclusions and other Expectations. Links can be of different types, indicating the different relationships between these objects. For example, the user might identify two Messages as evidence of the existence of another unit at this location. The first two Messages could be linked (via the Conclusions) to an Expectation for this as yet unsighted unit by a *doctrinal* link which may also contain a description or pointer to the appropriate military doctrine that suggests the existence of the third unit. The user can obtain an "explanation" of an Expectation (or Conclusion) by viewing the links in the Justification Network, using the browser as in Figure 7.

5.3 Monitoring Events

An active daemon associated with the Expectations has the task of monitoring new Messages as they arrive and Conclusions as they are entered or modified, looking out for evidence supporting or invalidating the Expectation. The user is alerted to such

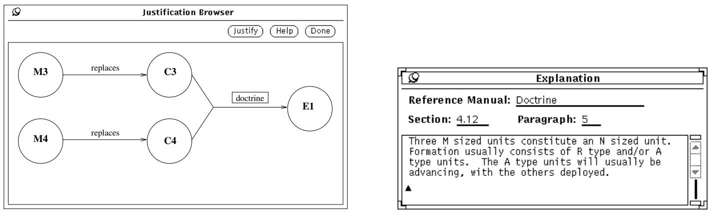

Fig. 7. The Justification Network browser. The explanation for a selected dependency (the doctrinal inference link) is shown.

changes as soon as they occur. The user then has the option of modifying the status and qualifier of the Expectation, possibly creating new Expectations and Conclusions. An Expectation that has been confirmed, for example, will usually be transformed into a Conclusion.

An Alert will be raised whenever it is deemed that the user should be informed of any significant events related to the Expectations. Events which merit attention include: a possible match of a new or modified Conclusion or Message with an Active Expectation; the timeout of an Active Expectation; or a change to the Conclusions or Messages which support an Active Expectation, as recorded by the Justification Network.

When an Alert occurs, the associated Expectation is added to a list of active alerts. A window is displayed informing the user of the new Alert. The user can choose to resolve the Alert immediately or to postpone this until convenient. The Alert (but not necessarily the window) will remain until it has been resolved. To resolve an Alert an appropriate course of action must be decided upon by the user.

If an Expectation matches a Message or Conclusion the Expectation status can be changed from Alerted to either Satisfied (by identifying one of the matching Messages or Conclusions as satisfying the Expectation, and replacing the Expectation and the "satisfier" Message or Conclusion with a new Conclusion based upon the "satisfier" Message or Conclusion), or InActive (simply causing the Expectation to be ignored by the system at present—no matching is performed on InActive Expectations). If an Expectation times out, the status of the Expectation can be changed from TimedOut to either: Active, (by modifying the Expectation to extend its time range), or InActive. If any of the Messages or Conclusions which support the Expectation in the Justification Network change, the Expectation status can be changed to either: Active, (indicating that the support changes do not affect the Expectation in question), or InActive, (indicating that the support changes invalidate the Expectation as it exists). New Expectations can be constructed at any time by the user.

Of central importance to the monitoring of Expectations is the idea of matching Expectations with Messages and Conclusions. Matching is the automatic process of identifying a Message or Conclusion which has some similarity to an Active Expectation. The process of bringing the match to the attention of the user, as described above, is an Alert. There are two scenarios under which matching is initiated. If a Message or Conclusion is created or modified, all Active Expectations will be searched to determine if this new or modified object is *expected*. If an Expectation is created or modified, all

Messages and Conclusions will be searched to determine if the expected situation has already been identified.

In TMIPS, matching is based upon location and identity. Location matching involves determining if the two regions intersect. Identity matching involves determining whether the configurations, and in turn the units of the two objects, match. In determining this the type, size and identity fields can be used.

In comparing locations and identities, imprecise matches may arise. Two types are distinguished: *possible* matches, and *necessary* (or exact) matches. A possible match describes the case where, for example, two locations overlap but do not coincide. An exact match occurs when the two values of an attribute in different objects are identical. A number can also be associated with the match, as an alternative means of specifying the degree of match.

6 Automatic Generation of Expectations

Conclusions, then, provide the user with the ability to capture inferences about how a particular situation is evolving as Messages arrive over time. This capability of revisiting earlier Conclusions and replacing them with new ones based upon new information is enhanced by the ability to explain current beliefs about the situation via the Justification Network. Expectations support the user with the additional ability to identify situations expected to arise in the future. Since the user creates these expectations based upon experience and domain knowledge, further support can be provided by identifying expected events automatically. This automatic generation of Expectations is driven by an understanding of the motivations and strategies of the opposing forces. A limited form of plan recognition is required.

Plan recognition is the process of determining how a set of perceived events in some domain are related, with the underlying desire of predicting what future events are likely to occur. Events can take the form of the sighting of new domain objects, or possibly of the movement over time of a single domain object, or perhaps even the non-movement or non-sighting of some object or objects. The definition of an event is domain dependent.

Traditional plan recognition involves fitting currently perceived events to a library of known domain plans. The process of identifying which plan is currently occuring consists of constructing a mapping from the “perceived” situation to one of the known plans. Of considerable interest as well is the explanation of the hypothesised mapping(s).

One way in which to address plan recognition from the standpoint of spatial information systems is to attempt to express the known (or common) domain plans in very general terms, typically as a pattern. An actual partial plan will be an instantiated pattern which may match some part of any one or more of the stored common plans (un-instantiated patterns). Matching a partial plan with a common plan will result in the identification of expectations of future events. Matching multiple common plans will result in multiple (or alternative) expectations. A general model of expectations, borrowing ideas from Truth Maintenance Systems (Winston 1984) and Possible Worlds (Charniak and McDermott 1985), is devised.

This general approach is being developed in TMIPS. Currently, all objects are archived so that partial plans which describe military unit movements are available.

These partial plans can be matched against patterns (or templates) derived from military doctrine relating to general strategies of the opposing forces. When matching occurs, TMIPS Expectations can be automatically generated, using the Expectations Model already developed. Also, new strategies adopted by the opposing forces can be "learnt" by the system from its observations, employing common machine learning approaches (Kodratoff and Michalski 1990).

Spatial and temporal templates, derived from doctrine, can be used to guide the intelligence officer's assessment. Spatial templates describe the relative spatial positions of the opposing forces. Temporal templates provide indicators of the progress of the opposing forces over time. Key to the concept of spatial and temporal templates is the concept of expectations. Spatial templates give rise to expected locations of units yet to be sighted given sightings of other related units. Temporal templates give rise to expectations associated with troop movement.

Uncertainty needs to be represented in templates. The distance between the expected units described in a spatial template, for example, must contain some flexibility. A rubber sheeting type of approach is envisaged whereby the units of the template can be visualised as having stretchable and shrinkable rubber bands between them. The amount of stretch and shrinkage permitted is specified as part of the template. If the current layout of actually sighted units falls within these bounds, then the template is applicable. Direction and time can similarly be rubber banded.

Introducing the idea of plan recognition into the spatial information systems arena through the conception of templates is an important step in the incorporation of more powerful spatial reasoning capabilities. Such enhanced systems can play an even more supportative role in the decision making process.

7 Summary and Conclusions

The problem of formulating a decision support system based upon manipulation and display of spatially oriented, complex and dynamic information provides many different challenges, both technical and conceptual. Data representation, technical mapping approaches, multiple simultaneous views of complex and uncertain objects, and the complications at all levels of interleaving hypothetical information with factual information are only a few of the difficulties that provide ample opportunity for investigation in this domain. While each of these problems have been addressed in the course of the TMIPS project, the focus of this paper has been upon the representation, explanation, and future evolution of predictions of change for data that has a spatial orientation. Expectations are a hypothetical representation of spatial data that may or may not exist, and may or may not be confirmed or dismissed at some future time. Evidential relationships exist and need to be recognised explicitly in the process of developing individual Expectations and in fact, in meeting those Expectations.

The support role of a decision support system can be enhanced by allowing the user to record their expectations, and have these expectations automatically monitored as the system evolves. Further, patterns can be identified and brought in to play a further supportative role in the identification of possible future events.

While the problem domain presented in this paper has been that of an interactive military intelligence system, many other domains with spatially oriented aspects offer different applications of these same ideas. Where there is a need to form a general picture of an evolving situation based upon intermittent and sparse observations, the ideas introduced here are relevant. One example is the task of tracking sightings of rare animal species. Here an Expectation could serve to help resolve many sporadic sightings into a general prediction of future sightings or perhaps even help understand other patterns such as migration. Domain knowledge about animal likes, dislikes, and habits could drive a plan recognition system that effectively combined spatial information such as vegetation, topography, water availability, etc., with sighting information over time.

In general, many environmental and infrastructure type applications could benefit from the use of an Expectations Model. Other potential applications in spatial information systems include weather observation and prediction, water quality measurements and inferences, and traffic flow measurements and consequent diagnosis of congestion over a network.

Acknowledgements

The seeds for the ideas which emerged as Expectations were planted by our colleague John Smith of the CSIRO Division of Information Technology. Early discussion with John helped to nurture the seedling. Discussion with our DSTO colleagues (particularly Ronnie Gori and Peter Calder) provided many insights into the application of the ideas to the military domain.

The original design and implementation of TMIPS as described here was done by Peter Milne, Scott Milton, and David Campbell, all of CSIRO Division of Information Technology.

References

Charniak, E. and McDermott, D.: 1985, *Introduction to Artificial Intelligence*, Addison-Wesley, Reading, Massachusetts.

Kodratoff, Y. and Michalski, R. (eds): 1990, *Machine Learning: An artificial intelligence approach*, Vol. 3, Morgan Kaufmann Publishers, Inc., Palo Alto, California.

Milne, P., Milton, S. and Smith, J. L.: in print, Geographic object-oriented databases, a case study, *International Journal of Geographic Information Systems*. Accepted for publication 1992.

Milton, S. and Campbell, D.: 1992, A graphic object mapping system for Xwindows, *Tools Pacific '92*, Sydney, Australia.

Ontos: 1991, *ONTOS Developers Guide*, Ontos Inc., Three Burlington Woods, Burlington Massachussetts, USA 01803.

Winston, P. H.: 1984, *Artificial Intelligence*, 2nd edn, Addison-Wesley, Reading, Massachusetts.