38

Collaborative intelligent decision systems

Jim McGovern
Department of Computer Science, RMIT
Plenty Road, Bundoora, Victoria, 3082, Australia,
Ph: 613-9468-2379, Fax:: 613-9467-8708 E-mail: mcgovern@rmit.edu.au

Abstract

This paper describes groupware that provides intelligent decision support based on the application of classical decision theory. The core of the system is essentially a group editor that supports the development of a common decision model in the form of an influence diagram. The groupware component supports flexible collaboration for a wide range of users and situations. The decision theory component provides a rigorous and widely applicable aid to decision making.

Keywords

Groupware, decision analysis, influence diagrams, group editors

1 INTELLIGENT DECISION SYSTEMS

The term intelligent decision systems refers to systems with the main features of providing intelligent support for users, and determining policy through the application of classical decision theory (Holtzman, 1989). Many problems to which such systems may be applied involve groups of decision makers.

This paper describes GuIDE (for Group Influence Diagramming Environment) a system that combines intelligent decision systems with recent advances in groupware (Ellis et al., 1991). The aim is to provide a group decision making aid that is widely applicable and supports a variety of group structure and behaviour.

Decision theory and its operational form of influence diagrams provides the basis for addressing less-well-structured problems in any domain, including valuable public sector decisions. The groupware component of GuIDE provides real-time distributed access for a broad range of participation and collaboration modes.

N. Terashima et al. (eds.), *Advanced IT Tools*
© Springer Science+Business Media Dordrecht 1996

2 GuIDE APPROACH TO DECISION SUPPORT

GuIDE differs from traditional Group Decision Support Systems (GDSS) in that it allows much more flexible access to group decision making. Group support systems have been classified according to time and place of participation (see for example Allison and Livesey, 1993). GDSS has focused on same place/same time decision making processes. Strong decision making aids, such as analytical hierarchy process (Dyer and Forman, 1992), along with other information processing aids, have been used in a relatively rigidly controlled process. The need to focus stems from the high cost of such a process. Expensive facilities are required, there is a high cost in participant's time and the emphasis for such systems is not necessarily on providing a flexible environment for participants.

GuIDE emphasises flexibility and provides a decision making environment where participants can access the system at any time from any place. Decision making can take place at the convenience of the participants. While this may not suit all decision making situations, it may be the most productive mode for some organisations and situations. In any case, a flexible environment can be constrained to provide other modes of operation such as same time and place.

GuIDE builds on the work of group editors, which support on the development of objects, such as documents (Ellis et al., 1990, Sun et al., 1996), or graphical objects (Karsenty and Beaudouin-Lafon, 1993) in applications such as software engineering. The focus of these systems is in providing functions that support collaboration (Dewan et al., 1994).

3 DECISION MODELLING IN GuIDE

Decision theory provides a sound basis for solving decision problems that are sensitive to uncertainty and preferences.

GuIDE represents decision problems as influence diagrams (Schachter, 1986). Influence diagrams provide a high-level graphical (relational) view of problems, allow a more detailed numerical specification, and can be solved directly to provide a policy that is consistent with the axioms of decision theory.

- The **relational view** of a decision problem supports the communication and development of problem structure. Figure 1 shows an influence diagram for a simplified decision problem model used to determine the quota for a course, given an early indication of demand for the course (the popularity poll).

- The **numerical specification** provides events, decision alternatives and associated numerical probabilities and utilities. Events that are influenced by other events will have conditional probabilities. The numerical level of an influence diagram is stored internally as a tree structure.

- The **solution algorithm** determines a policy that maximises expected utility. Exact solution algorithms are N-P hard problems. GuIDE deals with this by using the decision analysis process to keep decision models small, and their solution tractable.

Essentially, GuIDE is a group environment that aims to support the building of influence diagrams representing real world decisions. User commands to build influence diagrams consist of adding/deleting nodes, links and node events, and the setting of values representing probabilities and utilities. GuIDE also supports the manipulation of the model to produce solutions, and includes the methods and heuristics of decision analysis to help users develop and explore a model until they are comfortable with its solution.

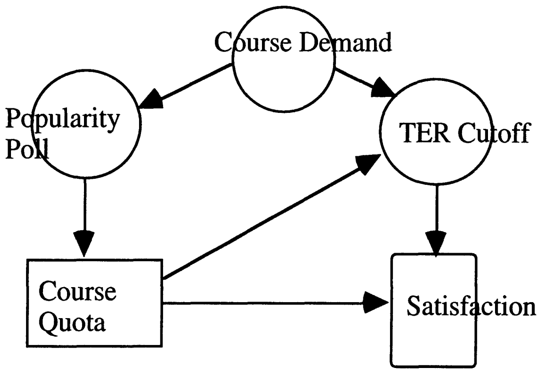

Figure 1 Influence diagram for setting of course quota decision.

4 THE GROUP DECISION MODELLING PROCESS

The process of decision analysis is an iterative one including repeated model building, sensitivity analysis and discussion about model structure and solutions. The emphasis is on acquiring understanding of the problem, and establishing consensus about the solution. GuIDE supports the following broad decision modelling phases.

Problem structuring establishes the problem elements of decision choices, events and an outcome node and their dependencies.

Deterministic analysis focuses on establishing the most important problem elements. Nodes to which the problem is highly sensitive are expanded, and those that have little impact are removed.

In the **probabilistic analysis phase**, the problem structure and the numerical specification are assessed in more detail. Again the aim is to keep only relevant events and decisions in the model.

In the **basis appraisal phase**, the solution is reviewed and may be accepted and documented. Alternatively, earlier phases may be revisited to further develop the structure and/or numerical specification.

GuIDE decision groups may include a facilitator, expert decision analysts, domain experts and stakeholders. Stakeholders potentially derive the most benefit from groupware, in general and GuIDE in particular. GuIDE allows easy access by a large number of stakeholders.

Intelligent support for the process of decision analysis is achieved by applying techniques such as:

- rule-based reasoning to guide the decision analysis process,

- rule-based and case-based reasoning (McGovern et al., 1994) to develop problem structure, and

- rule-based reasoning to suggest nodes to be removed or expanded.

The intelligence component of GuIDE is restricted to those support for individual users of GuIDE and has not been extended to support for group activities, although this is seen as an important area of potential future enhancement.

5 GUIDE ARCHITECTURE

The design of GuIDE is based on a replicated site architecture, where every participant site has a copy of the GuIDE software and its working data as well as a copy of the object being developed. The replicated architecture provides better local response and higher fault tolerance.

The groupware components of GuIDE are shown in figure 2.

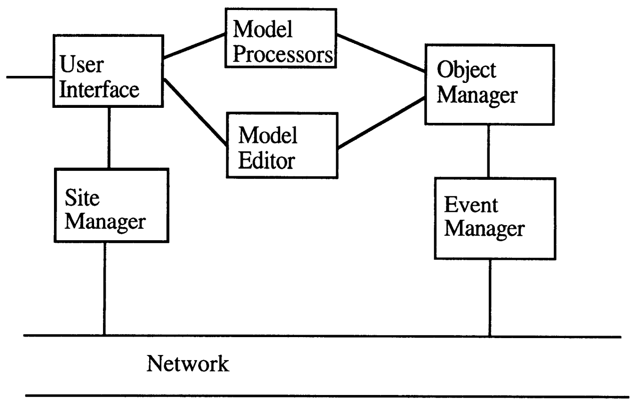

Figure 2 GuIDE as Groupware.

- The **Site Manager** keeps track of sites and users and their roles (if any) that are involved in a particular decision. Active users are aware of other active and inactive users who are involved in a particular decision.

- The **Object Manager** is responsible for the storage of decision models in the form of influence diagrams (as binary trees). In the replicated architecture the object manager must reconcile and integrate changes made at other sites with those made locally. The object manager also implements a selective undo facility that can reverse a particular user's actions.

• The **Decision Model Editor** handles all interaction between the system and the user, including user registration.

• The **Decision Model Processors** provide the facilities for exploring models including sensitivity analysis and model solution.

• The **Event Manager** consists of receiver and dispatcher components that interact with other sites. The *dispatcher* notifies the other sites of changes made at this site. The *receiver* accepts changes made at other sites and passes them to the object manager for application.

A prototype system has been developed using Java (Gosling and McGilton, 1995) a language for developing multi-user distributed applications. The prototype has been implemented on a network of Sun workstations in the Distributed Computing Laboratory at RMIT, Bundoora. Java has provided an easy environment for the development of a human-machine interface and for communication between remote processes using datagrams (Flanagan, 1996). Each site uses three threads, one to deal with the local user and one each for processing operations received from and dispatched to other sites.

Java applications can be initiated through world-wide-web browsers, and ultimately access to GuIDE will be provided through a high level interface using the world-wide web.

6 PARTICIPATION AND COLLABORATION

GuIDE aims to support a wide range of participation modes. Participants may connect from any location at any time to a decision making session that may take place over days or weeks. Alternatively, the decision making session may be highly structured and involve strict control of activity with participants in the same place at the same time.

All GuIDE decisions require at least an initiator who provides an access matrix that determines who may participate, and their action and object privileges. Of course a particular decision may allow anyone to join in with full privileges.

GuIDE supports WYSIWIS (What You See Is What I See). Just as important is the need to support WYSINWIS (WYSI-NOT- WIS), so that users are able to work privately, configure their own interface and communicate messages and their private work to whom they wish when they wish. They may also accept communication from other selected users when they wish.

7 CONCURRENCY CONTROL AND UNDO/REDO

GuIDE allows fully concurrent access. Use of GuIDE in full collaborative mode requires that participants be aware of the actions of others and be able to influence each other. Users need to notify other users of their work and in return be notified quickly of the actions of others.

The replicated architecture of GuIDE requires the correct interpretation of in-coming messages representing changes at other sites. When changes are broadcast to other sites they may arrive in a different sequence and they may conflict with the operations at other sites.

GuIDE uses a modified version of the algorithm developed by Karsenty and Beaudouin- Lafon (1993) to provide concurrency control in the group development of graphical objects.

This algorithm has the main following features.

- Each operation is applied at its local site before being broadcast to other sites.

- Each operation has a timestamp and a global ordering of operations can be determined.

- Operations that are applied at a site are written to a log (AppliedLog).

- Operations that are waiting for the creation of objects (nodes, links and events) are written to a log (WaitingLog), and are applied once the object has been created. For example, the operation to create a link may arrive at a site before the operation to create one of the nodes being linked. The waiting operation is applied when the operation to create that object is received.

- When an operation arrives at a remote site it is compared to those operations in AppliedLog that have a greater timestamp. Three situations may arise:

  - The operations in AppliedLog have already been applied and if there is conflict then the operation in AppliedLog is undone (it has a greater timestamp). For example, if a new link between two nodes $Node_1$ and $Node_2$ has been created and the logically older operation DeleteNode ($Node_1$) arrives, then there is conflict and the operation CreateLink ($Node_1$, $Node_2$) is undone, and the older operation DeleteNode ($Node_1$) is applied.

  - The two operations O_t (the operation with the older timestamp, but which has just arrived) and O_{t+n} (the operation with the later timestamp which arrived first and has already been applied) commute, in which case O_{t+n} can be applied without undoing O_t. For example, the creation of two different nodes commute.

  - O_{t+n} masks O_t and O_t can be ignored. For example, if the newer, already applied operation deletes the node, and the older operation which has just arrived deletes one of the node events, the deletion of the entire node masks the deletion of the event.

- AppliedLog is periodically checked to remove old operations that have been executed in the correct sequence.

The efficiency of the partial ordering depends on the level of commutation and masking in the application. In the development of an influence diagram a significant number of operations commute or mask. Also, importantly, groups using groupware such as GuIDE in WYSIWIS mode generally make use of social protocols to allow one user to update the common decision model. Additional communication channels such as audio communication can help participants manage this type of activity.

Concurrency control in groupware is a feature that is only rarely needed. Preliminary assessments of GuIDE confirm the experience of others (e.g. Sun et al. 1996) in that the number of undo-redo sequences required is quite low and optimistic concurrency control methods are able to cope quite well.

In a creative multi-user environment users will wish to withdraw changes in the light of suggestions or the actions of other participants. This requires the ability to undo operations specific to a particular user. Methods such as that proposed by Prakash and Knister (1994) can make use of the procedures and structures that support concurrency control to provide this selective undo facility.

8 USING GuIDE.

A typical decision making scenario suitable for GuIDE is that of academic administration decisions. Academic environments often provide a culture of open and participatory decision making. Such decision making may take place through newsgroups and e-mail. GuIDE improves this process by providing a more rigorous decision making engine, while retaining the advantage of easy access at any time from any place.

An issue in the use of decision analysis is the level of expertise needed to develop models. Significant training may be required. One approach is to provide an initial model structure, which can be numerically specified as is, or edited and then specified. It is expected that participants will become bolder with problem structuring as they become more experienced. In other environments greater training effort may be required.

A typical academic administration decision is the allocation of quota to a course such as that shown in figure 1. The problem is that as the quota of students being allowed into a program (and its funding) increases the quality of students will decrease. The best information that is available when quotas are set is an early indication of student demand known as the popularity poll.

Once the model structure has been developed, assessments of probabilities and preferences, must be obtained from those participating in the decision. Assessments of conditional probabilities of the events in the chance nodes are required. For example, the probabilities of the events of Course Demand conditioned by the events of Popularity Poll can be made. Similarly, the probabilities of the events of TER Cutoff conditioned by the Quota and the Course Demand can be made. Finally, the probability of each event in the Popularity Poll is assessed. A simple aggregating and normalising procedure is used to combine the multiple assessments. Because of the specialised nature of these assessments, estimates are made by the relatively few staff who are experienced in the selection process.

There is more interest in the input of preference information. Tenured staff may prefer better quality students while contract staff may prefer greater numbers and the greater certainty of continued funding of their positions. The participants are asked to rank preferences for student quality (through TER Cutoff) and the course numbers (Course Quota), in the range 1 to 5. Scores such as this can be easily aggregated into a simple linear utility function.

Once specified the model is solved, and if participants are happy with the outcome, the process may stop there.

9 CONCLUSION

The integration of groupware with intelligent decision systems has the potential to further extend the application of decision theory to valuable decisions. Proponents of decision theory would argue that this will result in demonstrably improved decision quality. To maximise the potential of collaboration in intelligent decision systems, the groupware component needs to support flexibility and to allow groups to address decisions in a variety of ways.

Ultimately, systems such as this may become readily available through facilities such as the world-wide-web. Employees may become used to checking out the latest organisational decisions just as they check out their mail and news each day. The ability to do this will result in greater participation in decision making with the results of greater commitment to decisions and perhaps even better allocation of resources.

10 REFERENCES

Allison, Colin and Mike Livesey (1993) Coping with Concurrency in Real Time Groupware, in *Proceedings of Symposium on Experiences with Distributed and Multiprocessor Systems IV*, USENIX Assoc., Berkeley, CA., pps 289-295.

Dewan, Prasan, Rajiv Choudhary and Honghai Shen (1994) An Editing-Based Charcterization of the Design Space of Collaborative Applications, in *Journal of Organizational Computing*, **Vol 4. No. 3,** pps 219-239.

Dyer, Robert F. and Ernest H. Forman (1992) Group decision support with the Analytic Hierarchy Process, in *Decision Support Systems*, **8** pps 99-124.

Ellis, Clarence A., Simon J. Gibbs and Gail L. Rein (1990) Design and Use of a Group Editor, in *Engineering for Human-Computer Interaction*, (ed. G. Cockton), Elssevier Science Publishers B.V. (North-Holland)

Ellis, C.A., S.J. Gibbs and G.L. Rein (1991) Groupware: Some Issues and Experience, in *Communications of the ACM*, **Vol. 34, No. 1**, pps 39-58, January.

Flanagan, D. (1996) *Java in a Nutshell*, Oreilly & Associates, Inc, Sebastopol, CA.

Gosling, James and Henry McGilton (1995) The Java Language Environment: A White Paper, Sun Microsystems Computer Company, Mountain View, CA.

Holtzman, Sam (1989) *Intelligent Decision Systems*, Addison-Wesley, Reading, MA.

Karsenty, Alain and Michel Beaudouin-Lafon (1993) An Algorithm for Distributed Groupware Applications, in *Proceedings of the 13th International Conference on Distributed computing Systems*, IEEE Computer Society, Los Alamitos, CA., pps 195-202.

McGovern, Jim, Danny Samson and Andrew Wirth (1994) Using case-based reasoning for basis development in intelligent decision systems, in *European Journal of Operational Research*, **74**, pps 40-59.

Prakash, Atul and Michael J. Knister (1994) A Framework for Undoing Actions in Collaborative Systems, in *ACM Transactions on Computer-Human Interaction*, **Vol. 1, No. 4**, pps 295-330, December.

Schachter, R.D. (1986) Evaluating Influence Diagrams, in *Operations Research* , **Vol. 34., No. 6,** pps 871-882.

Sun, C., Yang, Y., Zhang, Y. and D. Chen (1996) A Consistency Model an Supporting Schemes for Real-time Cooperative Editing Systems, in *Communication of the Australian Computer Science Association*, **Vol. 19., No. 1.**

11 BIOGRAPHY

Jim McGovern is Associate Professor of Computing in the Department of Computer Science at RMIT, Bundoora . He has a computer science degree from the Queensland University of Technology and has obtained a Ph.D. for work in intelligent decision systems form the University of Melbourne. His current interests, apart from groupware for decision support, are in databases and data modelling.