ON INTEGRATING AGENTS INTO GDSS

Constantin-Bălă Zamfirescu[1], Ciprian Cândea[2], Mihai Luca[3]

[1] *Katholieke Universiteit Leuven, Dept. of Mech. Engineering, Division P.M.A., Celestijnenlaan 300-B (Leuven) - Belgium*
[2] *Artificial Intelligence Research Group Reconstrucției 2A (Sibiu) - Romania*
[3] *Commerzbank AG - GHP Team Mosel 40 (Frankfurt am Main) - Germany*

Abstract: By nature, group decision cases present an immense diversity of goals, tasks, applicable processes and factors affecting them. The rigidity and limited scope of many decision models rise several critics from the practical perspective of GDSS. In this paper an integrated agent-based model for GDSS will be proposed and discussed as a complementary approach to undertake these problems. Supporting explicit representation of the decision-makers role, the procedural and contextual settings along with the group commitment to share a plan of actions as a way to achieve a common goal, results in improved capabilities, range and flexibility of GDSS.

Keywords: GDSS, MAS, group decisional process, shared plans.

1. INTRODUCTION

Group Decision Support Systems (GDSS) are interactive computer-based environments which support concerted and co-ordinated team effort towards completion of joint tasks. DeSanctis and Gallupe (1997) defined GDSS as a combination of computers, communications and decision technologies working in tandem to provide support for problem identification, formulation and solution generation during group meetings. The *commitment* of team members to achieve a common goal is considered implicit and mandatory during the functional use of the system. Anyway, this is not the case in most of the current organisational settings, where group members pose different goals, agendas and personal interests. Many studies investigated these contextual dependencies as external variables of group behaviour (DeLone and McLean, 1992). Besides supporting information access, GDSS can, at the same time, radically change the dynamics of group interactions by improving communication, by structuring and focusing problem solving efforts and by establishing and maintaining an alignment between personal and group goals.

Supporting in a flexible way a wide range of group decisions for the latest emerging organisational phenomena (i.e. work group autonomy, responsibility of professional roles, the flattening out and decentralisation of organisations) is still a challenge for GDSS research. As Gray and Mandviwalla (1999) observe the history of GDSS in real organisations is not encouraging. The inability of systems to survive beyond the whims of an individual champion may imply that they do not do anything that is sufficiently important for an organisation to maintain its investment. Extensive employment of Simon's model (Simon, 1997) becomes an obstacle for the evolution of decision support systems theory and practices

(Angehrn and Jelassi, 1994). Due to their incompleteness, the *rigidity of decisional models* employed in GDSS has been criticised on a number of grounds (Whitaker, 1992). The inequality of work and benefit discourages group members from adequately using the system. Low use of the system will lead to the replacement or cancelling of the GDSS. Ad hoc arrangements that are for the best interest of the group may not be compatible with the discipline imposed by the system. The system will be withdrawn because it is not compatible with the group operations. Consequently, GDSS do not fit well into organisational settings.

Moreover, in classical organisations decisions are often related to each other, creating a chain of temporal dependencies that are currently managed at a higher decisional level in the organisational structure. Thus, activities can never be exactly planned and cannot even be accurately stated, inducing a degree of *uncertainty in meeting planning*. In a recent survey, Kuo (1998) shows that knowledge for intuitive decision-making is often socially constructed and that perception and action play a critical role for real world problem solving. What appears to others to be intuition is actually an instance of well-trained cognitive ability to handle ill-structured problems. Thus, GDSS has to exhibit sufficient flexibility to support decisional process in very dynamic settings.

Due to the inadequate support from GDSS to model group members commitment to achieve a common goal, the incompleteness and rigidity of decisional models used, and the uncertainty carried out in meeting planning, it becomes inevitable that: 1) GDSS design is complicated enough to discourage wide spreading of the system as long as users are different in background, roles and interest; 2) group dynamics is difficult to understand and consequently to support in an adequate way; 3) group behaviour is not generalised to other groups being highly dependent by the context of use. Fortunately, the multi-agent system (MAS) paradigm represents one of the most promising approaches to address such kinds of problems. It offers a new dimension for GDSS integration with complementary services making easier to build complex and flexible architectures suitable for organisational settings. MAS are software systems composed of several autonomous software agents running in a distributed environment. Beside the local goals of each agent, global objectives are established committing all or some group of agents to their completion. Some advantages of this approach are: 1) it is a natural way for controlling the complexity of large and highly distributed systems; 2) it allows the construction of scalable systems since the addition of more agents become an easy task; 3) MAS are potentially more robust and fault-tolerant than centralised systems.

Taking into account that the MAS paradigm represents a feasible way to address some of the problems encountered in GDSS theory and practice, the remainder of this paper is organised as follows. Section 2 will give a glance upon promising researches directed toward a flexible and a natural integration of software agents into human teams. In the section 3 the proposed agent-based GDSS architecture together with its components will be described and discussed. The theoretical representation of the co-ordination mechanism used to achieve the desired flexibility in meeting planning will be outlined in section 4. Some concluding remarks and future work directions will be given in the final section.

2. AGENTS IN TEAMS

Several agent-based systems have been developed to support a smooth integration of software agents into human teams. To achieve this desideratum two main aspects are relevant to be stressed out: the manner in which labour is distributed between agents and humans, and the underlying modelling language to explicitly encode the teamwork. Roughly, there are four main dimensions along which agents may support teamwork (Payne *et al.*, 2000): 1) team situation assessment; 2) team-supporting behaviours; 3) team leadership/initiative; and 4) communication among team members. Based on the RETSINA architecture, the authors implement a training tool for military commanders in which human operators and agents collaborate to optimise route planning of vehicles. The architecture relies on three classes of agents: interface agents, task agents, and information agents. In the same application area, Miller *et al.* (2000) developed a virtual environment for battle staff training using a knowledge-based approach to encode the roles of team members, as well as goals, capabilities, responsibilities, needs, situations, and activities of the entire team, sub-teams, and individuals in the team. To describe team structures (roles and responsibilities), teamwork process knowledge (e.g., work flows, team plans), collaborative decision making knowledge, communication strategies and protocols they use a logic-based representation language called MALLET. A complementary approach has been proposed in the ELEVES project formerly used to host a visiting researcher (Chalupsky *et al.*, 2001). It relays on SharedPlans (SP) theory (Grosz and Kraus, 1996) used in conjunction with joint intention theory (Cohen and Levesque, 1991) to make explicit the possible course of actions in which a team of agents is involved. The approach emphasises the need to adjust the autonomy of agents when acting as proxies for the corresponding humans. Concerning interaction between agents and humans, COLLAGEN was used to build a collaborative interface agent for an air travel application (Garland *et al.*, 2000). The approach relies on collaborative discourse theory describing how people communicate and co-ordinate their activities in the context of shared tasks. SP theory is used to identify the possible steps of actions in human-agent interaction context.

As can be observed, although all these approaches can support a wide range of applications, each one addresses a very specific topic, which is however not the case for GDSS. The group decision-making process involves in a large extent a mixture of contingencies (Nunamaker *et al.*, 1991) able to emerge unexpected constraints during the decisional process in their most part difficult to structure and consequently to support.

3. AGENT-BASED GDSS ARCHITECTURE

In the last decade, the approach used to solve complex problems has shifted from developing large and integrated software systems, to developing small and autonomous software components that can interact with humans, with other software components, and different services or data source. To facilitate a flexible integration of different types of software components, an agent-based GDSS architecture is proposed (Fig. 1). In this framework, two general classes of agents can be distinguished: resource agents and interface agents. This kind of taxonomy is common in MAS literature (Payne *et al.*, 2000). Usually it assumes the existences of two types of actors, referred to as facilitator and group members. The facilitator is the person responsible for presenting the decision problem, and defining the rules to be followed by participants during the meeting. The users are the group members which discuss, find possible courses of actions, create and select among solutions the pertinent alternative using specific decision tools. In some meetings, the person acting as facilitator can play also the role of group member, but every meeting usually has one and only one facilitator/initiator.

Each user is assisted by an *interface agent* to maintain in a consistent manner his/her distinct profiles for different roles, task contexts and/or technological environments. Interface agents have to interact with users in a friendly and convenient manner, to receive user input and to display results. At the same time, the interface agent collects metadata about users while assessing user's feedback preferences for constructing and refining user's profiles. Most decision tools provide special ways to represent data, to acquire the required information, to make consistent alternative representations and to reduce the information overload. Tool capabilities are stored as metainformation in the resource profile, but these capabilities have to be tailored and integrated

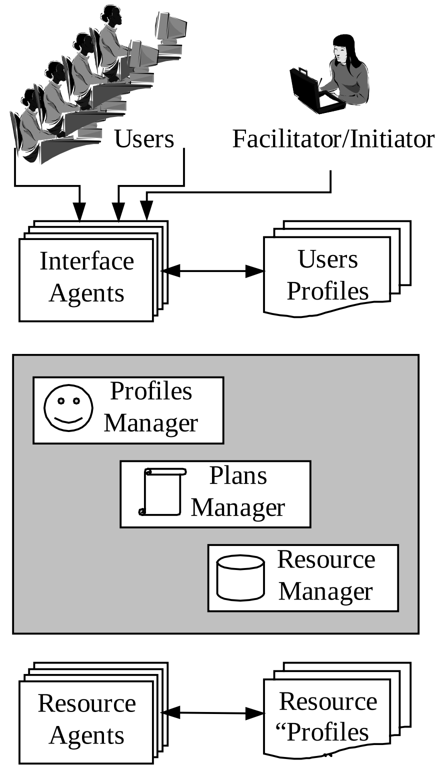

Fig. 1. Agent-based GDSS architecture

according with the contextual, cognitive, situational and possible other elements that bear upon effectiveness of task. Thus, the interface agents will have to model and utilise afterwards user preferences to support the user task and guide system co-ordination. At the same time, interface agents provide some intelligent features to manage in a mixed-interaction style the dialog with the user: 1) invitation mechanisms; 2) negotiation strategies for a chosen plan of actions; 3) provide unsolicited reports; 4) initiate dialogs with the user when the plan has been meanwhile changed to assure consistency among actions of meeting participants; 5) resolve some predefined tasks when responsibility to resolve it was transferred from the user. All these provide a suitable solution for many contingencies that constitue intrinsic factors for a successful GDSS: versatility, quality of help, adaptability, uniformity of interface, learning time and reduce information overload (DeSanctis and Gallupe, 1987).

Resource Agents perform some services for the rest of the system, providing at the same time intelligent access to a heterogeneous collection of services and data. They make functional details transparent to users, providing specialised or periodic information, or perform some task or service based on information they are given. Resource agents are generated at runtime accordingly to their stored profile in order to deal with a specific resource. For GDSS, three types of resources can be identified: communication services, DSSs and information resources. Communication services enable communication between group members (e.g. interactive video, email, news groups, etc.), facilitate information sharing (e.g. white boards, bulletin boards, collaborative editing, etc.), support collaboration and co-ordination between people (e.g. rational discourse, brainstorming, etc.) and support group decision tasks (e.g. voting, group analytical hierarchical processing, ranking alternatives, etc.). Moreover, the attention paid in the last decade to assist a web-centric perspective above DSSs gives rise to various decision tools already deployed on the Web. Inevitably, their associated counterpart agents appear to support a better way to deal in a flexible manner with a specific DSS (Gregg and Goul, 1999), since DSSs take many different forms that can be used in many different ways. Information retrieval systems is already a well-established research field in MAS (Zamfirescu, 1999). Specialised agents to exploit different types of resources (e.g. databases, digital libraries, model-bases, web-catalogues, service providers, ontology, search engines, etc.) are continuously spread out on the Web in different forms (i.e. data-driven DSSs, model-driven DSSs, knowledge-based DSS). More details concerning this kind of services in respect to GDSS context are given in (Candea, Staicu and Zamfirescu, 2000) being beyond the scope of this paper.

Plan Manager is the core-processing unit that 1) helps to formulate problem solving plans and carry out these plans through querying and exchanging relevant information; 2) maintains and monitors decisional process; 3) identifies potential conflicts and suggests possible solutions; 4) provides a framework to negotiate different approaches concerning the action steps that have to be followed during the meeting; 5) merges and integrates new plans into existing ones; and 6) tracks the plan execution, alerting decision-makers when deadlines approach. Briefly, the Plan Manager has to deal with four main functions: communication and co-ordination, planning, scheduling and execution. More details regarding these aspects will be given in the next section.

Profile Manager and *Resource Manager* are particularly intended for the exclusive use of system facilitators, enabling them to introduce new users and tools into the system. An user profile is build initially in respect to the organisational structure in which the user is involved, the skills and responsibilities that cover these positions, and the broad range of expertise in using technological issues. This initial profile will be refined afterwards during the system use for each subsequent used decision tool. On the other hand, a coherent description of a resource requires special skills to define and classify resource capabilities, accessing and execution protocols, input and output parameters in order to form the necessary integration with the others components of the system. A special attention must be given to the context in which these tools will be executed in a straight way. The execution context is provided in terms of pre-condition and post-condition rules that try to define tasks or group conditions that should be fulfilled before running the tool, preliminary documentation, collect additional data, etc.

4. DECISIONAL PROCESS AS A SHARED PLAN

The decision process is broadly defined as a bundle of correlated tasks that include: gathering, interpreting and exchanging information; creating and identifying alternative scenarios; choosing among alternatives; and implementing and monitoring a choice (Guzzo and Salas, 1995). Briefly, the decision process refers to some techniques or processing rules aiming at structuring the context, timing or content of communication. The challenge to model the decisional process, motivated by the prospect to offer a systematic perspective of how groups operate with certain variables within certain contexts, drives to a large set of decision models (Mintzberg, 1979; McGrath, 1991; Rasmussen, Brehmer and Leplat, 1991; Simon, 1997; Vangundy, 1997). Unfortunately, these models provide a partial perspective of the decision process, being more a contextual-based instantiation than a generic framework.

The decisional model that will be outlined below is based on the SP theory (Grosz and Kraus, 1996) which states that the participants need to have mutual beliefs about their goals and actions to be performed and the capabilities, intentions, and commitments of the participants. Inspired initially from a social sort of human collaboration, the model, which formalises these mental states of collaborative actions, was applied afterwards to model teamwork of agents. Consequently, $SP(P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$ denotes a group G 's plan P at time T_p to do action α at time T_α using recipe R_α in the context C_α. To successfully complete the collaboration, each group member must mutually believe that they: 1) have a common goal to find a course of actions in order to achieve the joint objectives (G does α over time T_α in the context C_α); 2) have agreed on a sequence of actions to accomplish the common goal (G's members mutually believe that R_α is the recipe for α in the context C_α); 3) are each capable of performing their assigned actions and intend to do their assigned actions as well (a subgroup $G_k \subseteq G$ has a subplan P_k for doing action β_i, using recipe R_{β_i}. in the context C_{β_i}); and 4) are committed to the overall success of collaboration not just the successful completion of their own parts (other members of G believe that there exists a recipe such as G_k can bring about β_I and intend to be performed by them).

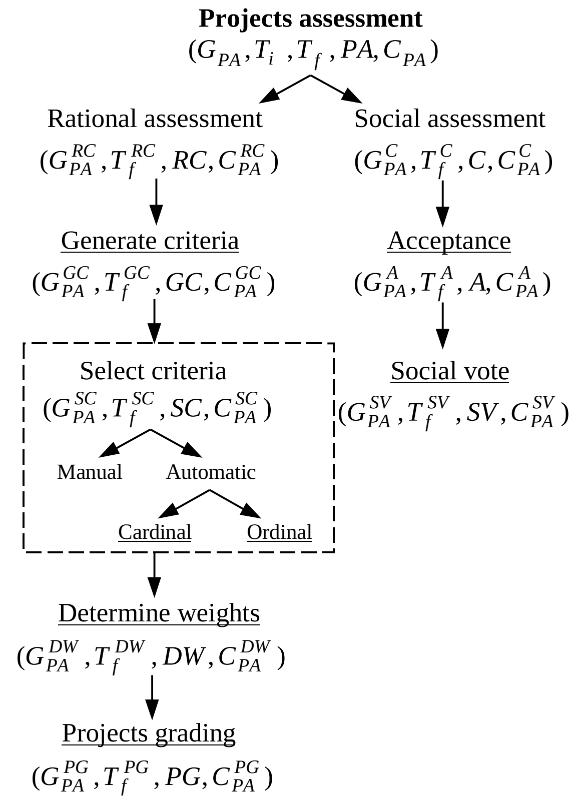

Fig. 2. A snapshot of GDSS plan trees for projects assessment.

Several important features of SP should be noted here. The observations will be depicted in the case of projects assessment with multiple criteria, being a very common case study for GDSS stream of research, with multiple applications from education to business organisations. The problem is also enough complex to cover all pertinent aspects of how SP could be used to model the entire decisional process. If the process is driven by rational motivations, the whole assessment process generally includes four stages, namely generation of assessment criteria, selection of assessment criteria, weight selected criteria and finally grading the projects (Fig. 2). From a social assessment perspective, there is usually an interplay between project's acceptance and social voting. This method is indicated in cases where agreement is an important group output, or where interpersonal conflict is creating problems in meetings. Note that the model do not force meeting facilitator to adopt a solution or another. Instead, the solution to tackle the problem under debate is emerging from interaction of group members. In the example outlined above that could be either a rational or a social assessment. As notations used, underlined nods represent atomic actions or steps that have a direct correspondence with a resource agent that will support action achievement (e.g. generate criteria, social vote). The rest are either goals or actions for which a further decomposition in primitive actions is required (e.g. rational assessment, select criteria). More details regarding the allocation of tasks to a special resource agent are given in (Zamfirescu and Filp, 1999).

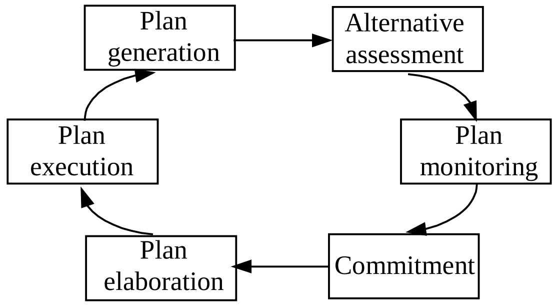

Fig. 3. Decisional process as emergent formulation.

The decisional process may start with only a partial SP (the recipe Rα may be only partially specified). A SP is either a full SP (situation in which every aspect of a collective activity α is fully determined) or a partial SP. As McGrath (1991) observes, the whole decision process often requires groups to cycle and move between multiple intertwined processes as new problems, alternatives and insights emerge. Thus, the decisional process is considered to evolve over time, as group members reactively decide the next step based on the context of the current situation. In the project assessment case, the steps how to select relevant criteria will be decided after these ones has been previously generated. To achieve this desideratum a continuos cycling between *Plan generation* (design appropriate steps to achieve the joint goal, i.e. generate criteria, select criteria, determine weights and projects grading), *Alternative assessment* (evaluate possible course of actions in the given context, i.e. rational assessment or social assessment), *Plan monitoring* (estimate the implications of new opportunities, i.e. manual or automatic selection of criteria), *Commitment* (explicitly express the willingness to assume the responsibility toward goal achievement following a curs of actions, i.e. rational assessment), *Plan elaboration* (extend or modify the plan in order to carry out own obligation, i.e. select criteria), and *Plan execution* (complete the decisional steps) is required (Fig 3.). Not all these steps should be considered as mandatory actions during the entire process, but as recursive opportunities to mediate the process itself. For example, if the relevant group already agrees to follow a social assessment of the projects, they just have to execute the plan.

The model can deal with group decision as well with individual decisions. Not all the actions are part of a SP, part of them being relevant only for a certain decision-maker. For instance, the decision-maker decides to use a hierarchical analytical processing tool to weight alternatives according to some personal criteria. This step is not pertinent for other participants, being irrelevant for the overall process. This kind of service can be found also in some GDSS architectures, in which decision-makers have exclusive access to particular types of DSSs, but it is not settled explicitly into the action plan.

*The model supports delegation and negotiation***.** Depending on the complexity of the problem, both the goal and the task may be decomposed. The organisational model (Mintzberg, 1979) posits that the complexity of the problems and the differences among individuals require some division of labour and result in the establishment of an organisation. The model supports delegation at different levels of abstraction, from individual delegation to departmental delegation (e.g. G_{PA} could be an entire department responsible for projects assessment, but the designed group to carry out the final decision G^{PG} is only a subteam made up from the former one). Moreover, the context of each step could explicitly stipulate if the decision has to be made by a certain number of decision-makers from a designated group (e.g. C^{DW} ask the criteria weights to be assigned only by three members of the group G^{DW}). Situations where unreconciliation or disagreements regarding the commitment to follow the course of actions appear will lead to communication within the team. An important purpose of the communication between participants is to determine the appropriate course of actions to do and who should do what. This is also the case of participants involved in organisational meetings without support from a GDSS. Inside the GDSS stream of research, the participants commitment is usual treated as a range of observable variables (e.g. willingness to conlucrate in the future, actions in respect to the process, meeting scheduling) that have no influence during the process itself (DeSanctis and Gallupe, 1987). These usually serve as basic data to understand and refine afterwards meeting settings within different contingencies. The necessity to support delegation inside GDSS has been argued in (Zamfirescu, Barbat and Filip, 1998).

The model supports synchronous and asynchronous collaboration. Discussion rules (i.e. duration of the discussion, if the contributions are anonymous or not, number of interventions, type of communication, etc.) has to be defined within the context of each action. The facilitator/initiator must define conventional limits according to the decision problem characteristics and/or the group. These limits could be constantly updated during the process. An important characteristic of SP is the possibility to inherit the context of the current action from the previous ones or from the covering plan.

Four categories of users could be identified for dealing with SP: 1) *system facilitator* - the user responsible for creating resource profiles, classify available resource, grant user access, etc.; 2) *meeting initiator* - the user who initiates a new meeting plan and performs any function to the created plans, delegates responsibilities, modifies plans, etc.; 3) *participants* - the users who can only access those meeting plans by which they are directly affected, perform those functions permitted by the initiator and facilitator, etc; 4) *observer* - the user who has read-only access to the meeting plans specified by the initiator.

6. CONCLUSIONS AND FUTURE WORK

Given the fact that current GDSS approaches assume that every group member is well aware of and accepts the rules defined for the meeting, the rigidity of the decisional models used, and the uncertainty in meeting planning, an agent-based GDSS architecture inspired from SP theory has been proposed and discussed. The proposed model provides a generic framework not only for register the order in which planning and execution occur but also for deciding how to interleave them. It emphasises the itinerary to achieve a common goal rather than the goal itself. The decisional process is emerged from the group interaction during the meeting and not prescribed beforehand. Unfortunately, such patterns are predominantly rigid, bureaucratic, and static, being unable to deal with dynamic situations in which group decision making usually take place.

The key proprieties of the proposed model could be summarised as follow: integrates activities across heterogeneous environments and provides a medium in which decisional process has to take place. It provides awareness-oriented collaboration, supports participants autoreflection during the meeting, captures the decisions, the rational behind each decision, the open questions related to the decisions, the assumptions behind it, and any related supporting information. The model has not to be viewed as a substitute approach of the existing ones, but as a complementary advance able to expand *capabilities* (increasing within group support so that more group activities are supported), *range* (support more organisational task), and *effectiveness* (of groups).

The model opens several research areas: 1) the roles interplay between human decision-makers and agents in mixed-initiative interaction; 2) group norms definition as a team-oriented programming case (Pynadath *et. al.*, 1999) the relationship with other models of decisional process; 4) integration in real organisations. The authors are currently pursuing several of these areas.

REFERENCES

Angehrn, A. and T. Jelassi (1994). DSS Research and Practice in Perspective. *Decision Support Systems: The International Journal*, **12**(4-5), 267-276.

Candea, C., M. Staicu and C.B. Zamfirescu (2000). An Agent-based System for Modelling the Searching Process on the Web. In: *Proceedings of 4rd IEEE International Conference on Intelligent Engineering Systems*, pp. 103 – 106, Portoroz, Slovenia.

Chalupsky, H, Y. Gil, C. Knoblock, K. Lerman, J. Oh, D. Pynadath, T. Russ and M. Tambe (2001). Electric Elves: Applying Agent Technology to Support Human Organizations In: *Proceedings of International Conference on Innovative Applications of AI*.

Cohen, P.R. and H.J. Levesque (1991). Teamwork. *Cognitive Science and Artifical Intelligence*, **25**(4), 487-512.

DeLone, W.H. and E.R. McLean (1992). Information systems success: The quest for the dependent variable. *Information Systems Research*, **3**(1), 60-95.

DeSanctis, G. and B. Gallupe (1987). A Foundation for the study of Group Decision Support Systems. *Management Science*, **33**(5), 589-609.

Garland, A., N. Lesh, C. Rich and L. Sidner (2000). Learning Task Models for Collagen. In: *Proceedings of AAAI Fall Symposium on Learning How To Do Things*, pp. 24-29, Falmouth, MA, AAAI Press, Menlo Park, CA.

Gray, P. and M. Mandviwalla (1999). New Directions for GDSS. *Group Decision and Negotiation*, **8**, 77-83.

Gregg, D.G. and M. Goul (1999). A Proposal for an Open DSS Protocol. *Communication of the ACM*, **42**(11), 91-96.

Grosz, B. and S. Kraus (1996). Collaborative plans for complex group action. *Artificial Intelligence*, **86**, 269-357.

Guzzo, R. and E. Salas (1995). *Team Effectiveness and Decision Making in Organizations*. Josey-Bass.

Kuo, F.Y. (1998). Managerial intuition and the development of executives support systems, *Decision Support Systems*, **24**, 89-103.

McGrath, J. (1991). Time, interaction and performance (TIP): A theory of groups. *Small Group Research*, **22**(2), 147-174.

Miller, M., J. Yin , R.A. Volz, T.R. Ioerger and J. Yen (2000). Training teams with collaborative agents. In: *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems*, pp. 63-72.

Mintzberg, H. (1979). *The Structuring of Organizations: A synthesis of the research*. Englewood Cliffs, NJ: Prentice-Hall.

Nunamaker, J.F., A.R. Dennis, J.S. Valacich, D.R. Vogel and J.F. George (1991). Electronic meeting systems to support group work: Theory and practice at Arizona, *Communication of the ACM*, **34**(7), 40-61.

Rasmussen, J., B. Brehmer and J. Leplat (1991). Distributed decision making. *Cognitive models for cooperative work*. Wiley. England.

Simon, H. (1997). *Administrative behavior: a study of decision-making processes in administrative organizations* (4th edition), Simon & Schuster Inc.

Payne, T., T.L. Lenox, S.K. Hahn, M. Lewis, and K. Sycara (2000). Agent-Based Team Aiding in a Time Critical Task. In: *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Vol. CD-ROM, IEEE Computer Society, Los Alamitos, CA.

Pynadath, D., M. Tambe, N. Chauvat and L. Cavedon (1999). Toward team-oriented programming. In: *Proceedings of the Agents, theories, architectures and languages workshop*, Intelligent Agents VI, Springer Verlag.

Vangundy, A. (1997). *Techniques of structured problem solving*, Van Nostrand Reinhold, New York.

Whitaker, R. (1992). *Venues for Contexture: A Critical Analysis and Enactive Reformulation of Group Decision Support Systems*. Research Report UMADP- RRIPCS 15.92, Umea (Sweden): Umea University, Dept. of Informatics.

Zamfirescu, C.B., B.E. Bărbat and F.G. Filip (1998). The "coach" metaphor in CSCW decision making systems design. In: *Intelligent Systems for Manufacturing: Multi-Agent Systems and Virtual Organizations* (L. Camarinha--Matos, H. Afsarmanesh and V. Marik, Eds.), pp. 241-250, Kluwer Academic Publishers, Boston.

Zamfirescu, C.B. and F.G. Filip (1999). An agent-oriented approach to team-based manufacturing systems. In: *Proceedings of the Second International Workshop on Intelligent Manufacturing Systems* (H. Van Brussel and P. Valckenaers, Eds.), pp. 651-658, Katholieke Universiteit Leuven Press, Belgium.

Zamfirescu, C.B. (2000). *Agent-based computing for information retrieval*. Technical Report IWAS-2000-01, Klagenfurt University, Department of Informatics, Austria.