Using Group Models to support Group Planning Interactions in MArCo

Patricia Azevedo Tedesco
Centro de Informática - UFPE
Email: pcart@cin.ufpe.br

Resumo: The emphasis on building co-operative/collaborative environments has brought out the issue of group interactions and conflicting situations, inevitable in group problem-solving. MArCo,our prototype, has an artificial conflict mediator that tries to provoke *reflection* and *articulation* on the solution process. In order to intervene more effectively, the mediator takes the group characteristics into consideration. In this paper, we show how the group model provides information for the mediator to use when deciding which intervention strategy to adopt.

Palavras-chave: Inteligência Artificial Aplicada à Computação, Modelagem Cognitiva, Aprendizagem Colaborativa apoiada por Computadores

1. Introduction

People are small group beings. In fact, we spend most of our lives as part of one group or another (e.g. our family, friends). In doing so, we have the opportunity to receive feedback, to discuss ideas, and to get support for our endeavours. It is also the case that we are faced with the (sometimes overwhelming) task of making decisions every day. Research has shown that when making decisions, groups tend to be more effective (they produce better quality solutions) whereas individuals tend to be more efficient (they produce solutions faster).

The realisation that we can be more productive as groups has caused a shift in the way we build computer systems. From building stand-alone, used in isolation systems, we are now more interested in building collaborative/co-operative environments. This change of paradigm has brought the issue of communication to light. Without a means to share ideas, we cannot work together. As evidence shows [1,2] conflicts happen through dialogue, and are inherent to group problem solving. Moreover, they can also be beneficial to it [1].

The inevitability of conflicts and their potential to help increase group productivity has generated a lot of research in different areas. For example, there are different lines of work in Education that consider conflicts to be inherent to group problem solving and to trigger cognitive change. For instance, Joiner's work [3] presents a model of conflicting interactions based on Doise and Mugny's theory of socio-cognitive conflicts [4]. Other authors (e.g. [5]) have also used conflicts in CSCL interactions to analyse the collaboration process.

In our research, we have explored the idea that it is possible to build an artificial conflict mediator that intervenes in conflicting situations with the aim of provoking *reflection* and *articulation.* Consequently, we have implemented MArCo, our prototype that embeds the artificial mediator. In its current version, MArCo's domain is PERT/CPM graphs. MArCo's reasoning mechanism is based on a Beliefs-Desires-Intentions (BDI) model of conflicts (described in [6]) in group planning situations. The mediator diagnoses the changes in the group's plan, and points out courses of action that may lead to more refined solutions (the changes model is presented in [7]). In order to be more effective, the mediator also takes the history of the dialogue as well as group and individual characteristics into consideration. In this paper, we argue that a model of the group can help a computer environment to provide better support for its users, and show how this idea has been applied in MArCo.

This paper is organised as follows: section 2 presents our model of the group (and of the individuals) in the context of research in this area. Section 3 introduces MArCo, our computer system. Section 4 presents an example interaction, showing one situation where the group model's components have been used. Lastly, we present some conclusions and extensions for the work presented here.

2. The Model of the Group and of the Individuals

In this section, we describe how we have provided MArCo with information about group characteristics and knowledge, in order to make the mediator more effective. Even though we need group models to cater for collaborative/co-operative interactions, we cannot forget that groups are made of individuals, whose needs must also be considered. Thus, in MArCo, we have both models – empty when the interaction starts and updated as it proceeds.

Although this situation is changing, it is still true that the vast majority of *CSCLEs* (Computer Supported Collaborative Learning Environments) do not have a well-defined model of the group interaction. Hoppe [8] remarks that individualised Artificial Intelligence in Education has been criticised for not catering for the social aspects of learning. According to him, there are different ways to solve this problem – ranging from embedding our individual systems into social contexts to building software that can follow social interactions. With the high demand for collaborative/co-operative systems, especially in geographically distant situations the second case needs to be better explored. This is the case in MArCo.

Hoppe [8] addresses this issue by using individualised models to inform and parameterise human-human co-operation without monitoring the interaction. For instance, the choice of problems that will challenge the group as a whole can be guided by the analysis of the differences in the individual knowledge, registered in the individual student models. This is an interesting idea, but in MArCo we need to monitor the interaction (to detect conflicts as they happen). In order to be more effective, the mediator draws analyses the differences between the group and the individual models, as well as the social roles played by group members.

Another stream of research in group modelling is concerned with finding out how we can form effective collaborative learning groups. One example of this trend is the work presented in [9]. There, groups are formed by assigning social roles to group members and a learning goal for the group. Group learning goals are determined according to the event that triggered group formation (for example, if the system has decided to form a group because student X had a misconception, the group goal is to clarify it). Differently from MArCo, social roles are static (the chosen roles are *learner, helper, presentator, observer, participant* and *debater*).

Nakamura and colleagues [10] have built a multi-agent environment that assists students discussing a logically inferable domain. Each student has a private agent, that calls learners to discuss things, and builds a difference model (based on the differences between the learner's private space and the common area). A group agent, located on the server, acts if there is a deadlock in the discussion, caused either by erroneous or missing knowledge. It is not very clear, though, how the agent uses this knowledge to intervene.

It is not the case that we need to trade individual models (IM) for group models (GM). Even in group interactions, we cannot loose track of the individual characteristics of the learners (for instance, to keep the group members motivated, we have to consider their individual characteristics and needs [11]). Paiva [12] shows that there is evidence for the assumption that "Collaboration conducts to learning under certain constraints and in certain situations which are dependent on both the individual and the group aspects of the learners involved in the collaboration process".

According to [12], when dealing with a CSCLE, we need to keep track of the group interactions, the dynamic changes of *social roles*, *beliefs*, *conflicts*, *misconceptions*, and *views of the task* within the group. This view was slightly adapted to the needs in MArCo, as we will discuss later. A group model can be used to capture a lot of information that can help to improve the coaching interaction in a CSCLE. For instance, it can describe which roles the group members are playing in the dialogue, their skills and group knowledge [5]. Our work incorporates this claim to the structure proposed by Paiva.

MArCo's GM was based on Paiva's recommendations [12]. Her GM consists of the following sets:

- • ***Group beliefs*** – are inferred from the group actions in the common shared space and from the ongoing discussion;
- • ***Group actions*** - are those assumed to have been done by the group as a whole;
- • ***Group misconceptions*** - are those diagnosed from group actions;
- • ***Group differences*** – may be a way of settling the negotiation process. For instance, people often tend to accept other's suggestions based on their expertise. *Differences* can be obtained by comparing the IMs, and used to provide feedback to the group.

- • ***Group conflicts*** – are differences in the beliefs of two or more learners. These are potential triggers of cognitive changes.

Another system that uses GMs is Vizcaíno's HabiPro (Habits in Programming). It works based on the performance of its GM [13]. HabiPro is a collaborative environment for the teaching of Java. Its interface has two windows, one with the problem to be solved and some space for the students to input their solution and a chat facility, where students can discuss their ideas. The authors claim that their system adapts its instruction via the GM.

HabiPro's GM is based on the structure proposed by Paiva, discussed above. It includes the following information:

- • ***What are the groups abilities*** – these are inferred by the exercises the group has solved;
- • ***Which exercises the group prefers***;
- • ***Which mistakes the group has made***;
- • ***The degree of motivation of the group*** – inferred by the type of help the group has used;
- • ***The number of times each student has participated in the conversation.***

While this approach does give the system some indication of the quality of the collaboration process, we believe that providing the system both the quantity and the quality of participation moves gives the system better resources to intervene in the discussion (for example the system could decide to ask the expert for his/her opinion, to ask the reflective student to elaborate on some contribution, to probe the quiet learner for more opinions, etc). Thus, the system can not only address task problems, but also help keep members interested and focused on the task. This is the approach adopted in MArCo.

Having described the current approaches in group modelling, let us now discuss MArCo's IMs and GM. Both models have been based on the literature, and have been adapted to cater for the knowledge the system needs to mediate *Meta-Cognitive Conflicts*. A typical IM in MArCo consists of the following sets of BDI[1[1]] attitudes:

- • ***Domain Beliefs*** – these correspond to the first level of Self's DORMORBILE framework [14]. In our PERT/CPM case, domain beliefs appear mostly when we are discussing the context of our problem, the ordering in which things should be done, or the justifications for that ordering. Because contexts are one of the defining issues in a strategy, we are marking contextual beliefs with "context". Examples of domain beliefs of Agent X[2[2]] about the problem shown in figure 3 are shown below:

⇒⇒ *belief_domain(X, parallel(produce P1, produce P2))*, meaning that X believes that P1 and P2 can be produced in parallel.

⇒⇒ *context(X, ends(Produce P1, 20))*, meaning that X believes that P1's production should end by day 20.

- • ***Intentions*** – represent the actual steps of a strategy. For instance, when a pair is cooking dinner together and one proposes to chop the vegetables, this is represented in their cooking dinner strategy as the intention to chop vegetables. Thus, agent X's intention to train workers and purchase materials in parallel is represented as:

⇒⇒ *Intend(X, parallel(train workers, purchase materials)).*

- • ***Goals*** – whenever the system realises that the group has not yet defined their goal, it will encourage them to do so, in an attempt to help the group make more objective decisions concerning the strategy being built. Suppose now that our agent X wants to finish her/his project in 30 units of time. This goal is represented as:

⇒⇒ *Goal(X, finish(project, 30)).*

- • ***Reflectors*** - are meta attitudes. They represent our thoughts about how we should set out to build/choose our strategies. For instance, during one interaction with MArCo (described in [15]), one participant suggested that they should group their intentions into the ones before, during, and after toy production. Such a reflector is represented in MArCo as:

⇒⇒ Reflector(X, group(steps, (before, production,after))).

Whenever a group member makes a contribution, her/his IM is updated. The same happens to the GM when there is an agreement by the majority. MArCo's GM consists of the same sets of attitudes as the IMs, plus the roles played by the group members during the dialogue. Roles are attributed to members

[1[1]] BDI stands for Belief, Desires and Intentions.

[2[2]] By agent X we mean any of the human group members.

according to the number and type of contributions they make. The assignment of roles is an extra resource for the mediator. MArCo's possible roles are:

- **Leader** - group member whose proposals have been accepted most of the time;
- **Follower** - members that have not really been putting proposals forward, but only accepting those made by others;
- **Outspoken** - members that have been contributing a significant amount of the time;
- **Shy** - members that have been keeping quiet;
- **Reflector** - members that although have not been making many proposals, have challenged and/or probed the proposals made by others;
- **Actor** - the "generators of ideas", i.e., the members that have been making a significant amount of proposals, without worrying about probing others' contributions.

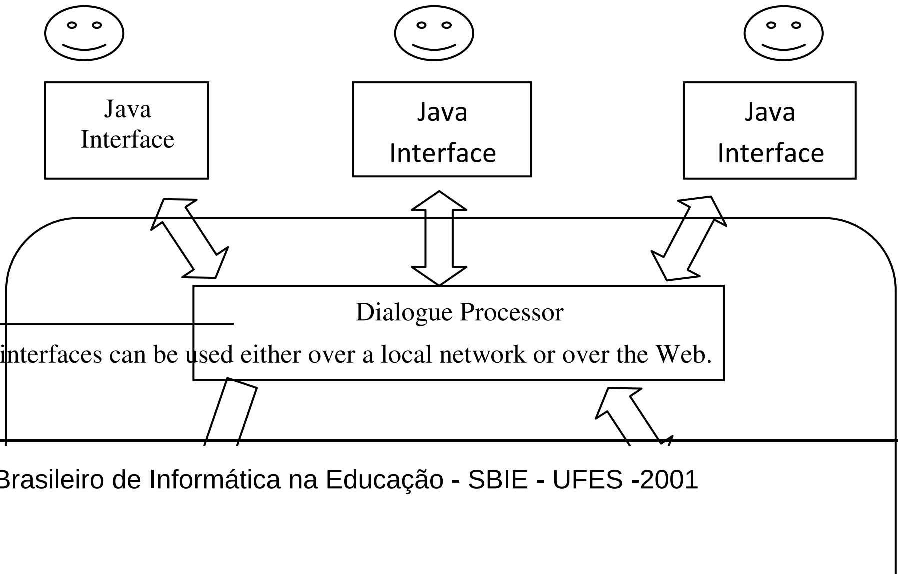
The idea of recording the roles played by the group members serves a twofold purpose: firstly, it can give the mediator some clues on how to pose its interventions (for instance, after learning that one of the participants is a *reflector*, it could ask him/her for his/her opinion about something next time the opportunity arises); secondly, it can be an indication of how well the collaboration/co-operation is going, and also indicate to the mediator how effective its strategies are. Having looked at the importance of group models and their use in MArCo, let us now present the system's architecture and analyse one example interaction.

3. Putting Theory into Practice: MArCo

MArCo stands for *Artificial Conflict Mediator* (in Portuguese). It has a Java-based distributed interface that communicates with a Sicstus Prolog Server.[3][3] The interface allows geographically separated group members to build a plan together. The server enables MArCo to reason about and intervene in the interaction taking place at the client interfaces. Figure 1 presents the system's architecture, showing the flow of information among the components.

When they log onto MArCo, group members see a common workspace, where their plan will be built. Users contribute to the discussion via the dialogue and/or graph tools available at the interface (shown in figure 2). Once a user finishes her/his utterance, s/he presses the *Submit your Contribution* button. The contribution is then sent to the *Dialogue Record* (at the interface) and to the *Dialogue Processor* on the Prolog Server. The *Dialogue Processor* is responsible for receiving utterances, mapping them onto the *BDI* attitudes that describe our strategies and passing the resulting information to the *Maintainer* and *Mediator.*

The *Maintainer* then updates the IMs and GM. If an individual makes a contribution that is inconsistent with what was previously held in her/his model, the system assumes that s/he has changed her/his mind, and revises the relevant set of attitudes in the IM, in order to eliminate inconsistencies. To make matters clearer, suppose *user1* declares the intention of doing activity a (having previously declared the intention of doing $\neg a$). So, when updating *user1*'s IM, the maintainer will revise its set of intentions, eliminating the resulting inconsistency. Whenever we have an agreement by the majority, the system updates the GM, including the agreement in the respective attitude set. Whenever the group finishes a dialogue game, the system updates the social roles played by its members.



[3][3] The client interfaces can be used either over a local network or over the Web.

Figure 1 - MArCo's main Components and the flow of information.

MArCo's *Mediator* observes the dialogue looking for conflicts and acts if/when necessary. It uses three levels of mediation. On the simplest level, it informs the group that some sort of conflict has occurred. In the next level, the *Mediator* informs the group that a conflict has occurred and then checks if any of the participants is saying something inconsistent with her/his (or the group) model. If that is the case the *mediator* asks the respective member to elaborate on this “change of mind”, attempting to provoke *reflection*. In the third level the system also suggests actions that can lead to more refined solutions. For example, since strategies, from our point of view are defined by their goals and context, the *Mediator* checks if the group has already defined their goals and/or context. If that is not the case, it suggests that the group should do so. Also, the mediator tries to use the social roles played by the members (as will be shown in the example presented in section 4) in order to reinforce the guidelines for supporting collaboration (discussed in [15]) and/or keep users motivated. The *Mediator* draws information from the GM and IMs, the history of the dialogue, our models of conflicts and of strategic changes and a model of the task in order to perform its reasoning. The *Mediator* sends its interventions to the *Dialogue Processor*, who broadcasts them to the clients’ interfaces.

4. An Example Interaction

Now let us suppose that agents A and B are discussing about the problem shown in figure 3. Before we proceed, it is worth noting that MArCo uses a dialogue game approach. Thus, users utterances consist of the user's identity, the chosen dialogue move (which indicates the illocutionary force of their contribution), and the contents. Contents are expressed in a formal language[4][4]. In MArCo's interface, sentence openers represent the dialogue moves, to provide for a more natural dialogue. Here we will only present the utterances as they are received by the Prolog Server. Utterances are represented as: <user>.<dialogue move>.<contents>. In this light, let us have a look at the dialogue shown in figure 4.

[4][4] The dialogue games approach and the formal language defined are presented in full detail in [15].

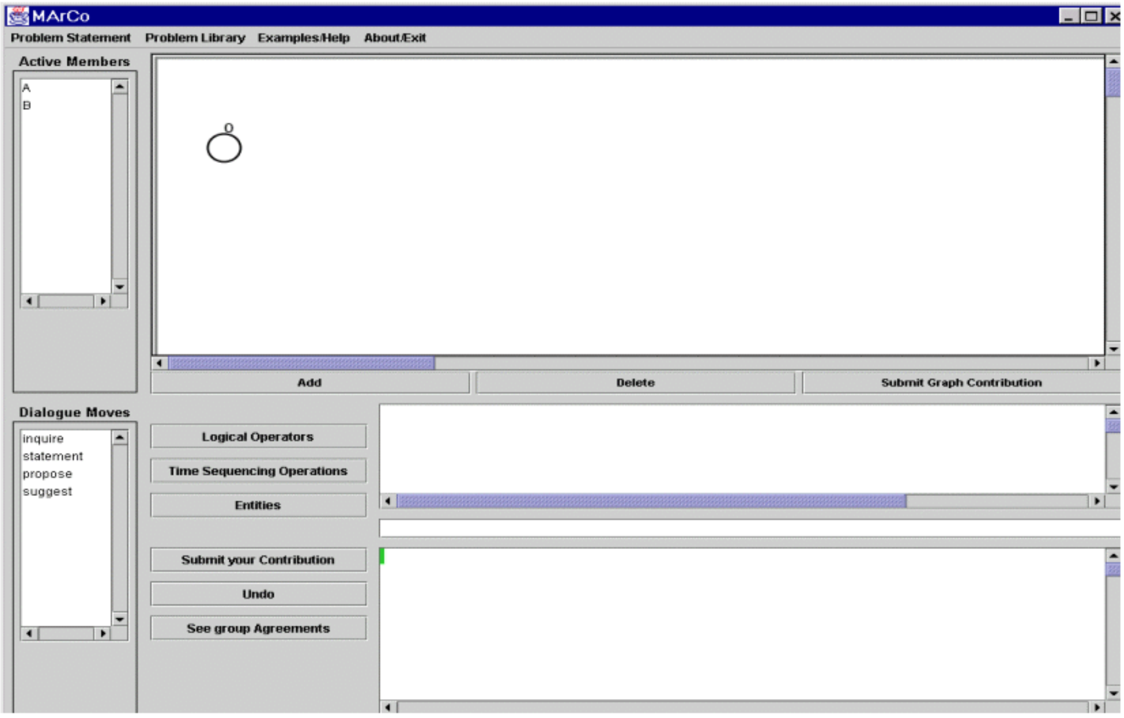
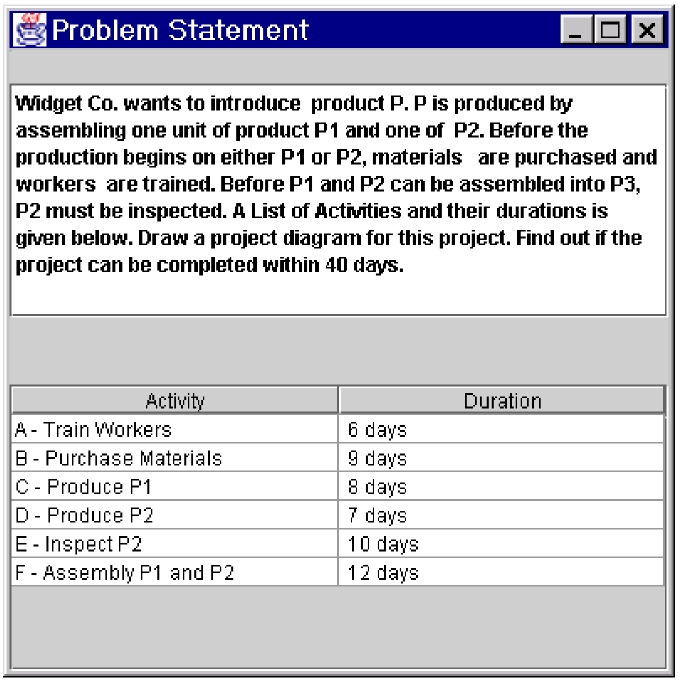


Figure 2 - MArCo's Interface.



Widget Co. wants to introduce product P. P is produced by assembling one unit of product P1 and one of P2. Before the production begins on either P1 or P2, materials are purchased and workers are trained. Before P1 and P2 can be assembled into P3, P2 must be inspected. A List of Activities and their durations is given below. Draw a project diagram for this project. Find out if the project can be completed within 40 days.

Activity	Duration
A - Train Workers	6 days
B - Purchase Materials	9 days
C - Produce P1	8 days
D - Produce P2	7 days
E - Inspect P2	10 days
F - Assembly P1 and P2	12 days

Figure 3 - An Example Problem.

In figure 4, utterances 1 and 2 refer to adopting the goal that the project should last 35 days. In 3 and 4, A and B agree that activity A should be the first one in the project. After that, the participants move on to discuss the ordering of the activities that compose the project. Until utterance 7, B had taken the role of the *follower*, only agreeing with A's proposals. On utterance 7, B proposes that they do activity C before activity D. On 8, A disagrees, proposing to do C in parallel with D instead. At this point, MArCo signals a disagreement, and stores the following fact in its knowledge base:

disagreement([A,B], parallel(C,D), intention,[C,D]), meaning that *A* and *B* had an intention disagreement (whose last contribution was *parallel(C,D)*), and the focus was *[C,D]*.

On utterance 9, B disagrees with A. At this point, MArCo detects another intention disagreement between the same agents (A and B) about *[C,D]*. It then interprets this as the participants intending to convince each other that they are correct and signals a conflict.

At this stage, the mediator realises that there has not been a good swapping of social roles during the interaction, and decides to intervene by probing B (the *follower*) to explain her/his point of view. MArCo's Intervention is as follows:
System.Statement. An Intention Conflict was Detected.
Do you not think defining the context would help defining the problem?
B, would you like to explain why you think you are correct?

This is only an example that illustrates the use of social roles in MArCo. Other variations are certainly possible, having been used in the system. For instance, when MArCo detects that participants are now contradicting a decision taken previously by the group, it brings that up, asking for a clarification of the change of mind.

1.A.suggest.goal(lasts(project,35)).
2.B. critique.agree.
3.A. propose.starts project(A).
4.B.critique.agree.
5.A.propose(parallel(A,B).
6.B. critique.agree.
7.B. propose(before(C,D)).
8.A. counterpropose.parallel(C,D).
9.B.critique.disagree.

Figure 4 - An example dialogue between users A and B.

5.Conclusions and Further Work

In our work, we have explored the idea that conflicts and the cognitive changes they provoke can be used to support group interactions. In order to do so, we have implemented MArCo, a computational system that has an artificial mediator, capable of detecting conflicts, of diagnosing the ongoing changes on the strategy being built, and of taking action whenever deemed necessary. In this paper, we have also discussed the importance of group models in the support process, and presented MArCo's IMs and GM, together with how the mediator uses the information stored there to provide interventions that are more useful to the group.

MArCo has been evaluated (a detailed description of the evaluation process can be found in [15]), and results have shown that the approach is does help people reflect on and articulate their knowledge. Actually, a comparative study between groups working with and without the mediator has shown that mediated groups had a higher percentage of reflective/articulated dialogue moves (defined in [15]). Participants in the evaluation have been quite receptive to the idea of the artificial mediator, remarking that the mediator has indeed helped them reflect on and articulate their views. According to their answers in the debriefing sessions, participants thought that the mediator's interventions were quite adequate. Moreover, the users that worked with the mediator switched off, have indicated that they thought their answers could have been more refined with the presence of the artificial mediator.

In the short term, we intend to investigate how we can refine our group model so that the mediator can be more effective. In other words we want to investigate which characteristics could be incorporated to the GM in order to make it even more useful, specially in situations where group formation varies along time, and how we can apply the approach to other domains.

6. References

1. Putnam, L.L., (1986). Conflict in Group Decision-Making. In: Hirokawa, R.Y. and Poole, M.S. (eds.): *Communication and Group Decision-Making*, Sage Publications, 175-196
2. Robbins, S.P., (1996). *Organizational Behaviour: Concepts, Controversies and Applications*, London, Prentice-Hall,
3. Joiner, R. (1995). The Negotiation of Dialogue Focus: An Investigation of Dialogue Processes in Joint Planning in a Computer Based Task. In: Claire O'Malley (ed.): *Computer Supported Collaborative Learning*, Nato ASI Series, 203-222
4. Doise, W., and Mugny, G. (1984). *The Social Development of the Intellect*, International Series in Experimental Social Psychology, vol. 10, Pergamon Press,
5. Soller, A., Goodman, B., Linton, F. and Gaimari, R., (1998); "Promoting Effective Peer Interaction in an Intelligent Collaborative Learning System", in Proceedings of the 4th International Conference on Intelligent Tutoring Systems, ITS'98, Goettl. B.P., Halff, H.M., Redfield, C.L., and Shutte, V.J. (eds.), Springer Verlag, 186-195

6. Tedesco, P. A. and Self, J.A. (2000a) Using Meta-Cognitive Conflicts to Support Group Problem Solving. In Proceedings of the 5th ITS'2000, Gauthier, G., Frasson, C., and VanLehn, K. (eds.), Lecture Notes in Computer Science, 1839. Springer Verlag, 232-241
7. Tedesco, P.A. and Self, J.A. (2000b) "MArCo: using Meta-Cognitive Conflicts to Provoke Strategic Changes". In Sichman, J. and Monard, M.C. (eds.) Proceedings of the International Joint Conference 7th Ibero-American Conference in AI 15th Brazilian Symposium on AI - IBERAMIA-SBIA'2000, Lecture Notes in Artificial Intelligence 1952, Springer Verlag, 186-195
8. Hoppe, H.U. (1995). "The Use of Multiple Student Models to Parameterise Group Learning". In Greer, J. (ed.) Artificial Intelligence in Education 1995, AACE
9.Ikeda, M., Go, S. and Mizoguchi, R. (1997). "Opportunistic Group Formation", in du Boulay, B. and Mizoguchi, R., (eds.), Artificial Intelligence in Education: Knowledge and Media in Learning Systems, IOS press, 167-174
10. Nakamura, M., Hanmoto, K., Tsutsumi, Y., Fukuda, M. and Otsuki, S. (1999) Models and Negotiations for Group Learning Support. In Cumming, G., Okamoto, T. and Gomez, L (eds.) Advanced Research in Computers and Communications in Education: New Human Abilities for the networked Society -Proceedings of the International Conference on Computers in Education ICCE'99. IOS press, pp. 677-684
11. Issroff, K., and Del Soldato, T., (1996). "Incorporating Motivation into Computer-Supported Cooperative Learning" in Proceedings of the European Conference on Artificial Intelligence in Education, P. Brna, A. Paiva & J. Self (eds.), Edições Colibri, 284-290
12. Paiva, A., (1997). "Learner Modelling for Collaborative Learning Environments". In Proceedings of the 8th World Conference on Artificial Intelligence in Education (AIED'97). Du Boulay, B. and Mizoguchi, R. (eds.). IOS Press, 215-222
13. Vizcaíno, A., Contreras, J., Favela, J., and Prieto, M. (2000). "And Adaptive, Collaborative Environment to Develop Good Habits in Programming", in Proceedings of the 5th International Conference Intelligent Tutoring Systems (ITS'2000); Gauthier, G., Frasson, C., and VanLehn, K. (eds.) Lecture Notes in Computer Science, 1839, Springer Verlag, 262-271.
14. Self, J.A. (1995) "DORMORBILE: a vehicle for metacognition" in T.W. Chan and J. A. Self (eds.) Emerging Computer Technologies in Education, Charlottesville AACE
15. Tedesco, P.A. (2001) Mediating Meta-Cognitive Conflicts in Group Planning Situations, Unpublished Doctoral Dissertation, Computer Based Learning Unit, University of Leeds, UK.