A Generic Library of Knowledge Components to Manage Conflicts in CE Tasks

Nada Matta, Olivier Corby and Biren Prasad
Concurrent Engineering 1998 6: 274
DOI: 10.1177/1063293X9800600401

CONCURRENT ENGINEERING: Research and Applications

A Generic Library of Knowledge Components to Manage Conflicts in CE Tasks

Nada Matta* and Olivier Corby

INRIA (project ACACIA), 2004 route des Lucioles BP. 93, 06902 Sophia-Antipolis Cedex, France

Biren Prasad

Unigraphics Solutions/CERA Institute, P.O. Box 250254, West Bloomfield, MI 48325-0265, USA

Received 2 July 1998; accepted in revised form 5 October 1998

Abstract: Concurrent Engineering requires a cooperative work-group of teams in which several multidisciplinary teams of engineers, designers, process planners collaborate to build a system. Conflicts appear during this collaboration due to differences in the background and experiences of the team members (product engineers, designers, process planners, etc.) in the work-groups. To leverage the product development teams' (PDTs—engineers, designers, managers, and knowledge workers) know-how during product development, a historical trace of their experiences in a particular domain of interest can be captured into a series of digital or virtual models. In this paper the authors present some guidelines to build such library of knowledge components particularly for conflict management. How this generic library of knowledge components would model work-group tasks in CE is explained. Also the authors describe how to develop methods to manage conflicts in CE tasks, and a methodology to use this library.

Key Words: knowledge-based engineering, Concurrent Engineering, Conflicts Management, Generic Library of knowledge components, Technical Memory, knowledge management.

1. Introduction

In Concurrent Engineering (CE), several PDTs of engineers, designers, process planners—each team experienced in different fields and life-cycle domains—collaborate to build a system together (called an artifact). Product and system development is a complex undertaking in which several dimensions or perspectives must be considered. The need for collaboration between participants (engineers, designers, managers, process planners, etc.) often generates a number of conflicts, which need to be managed in order to reach the ultimate goal, that is to "build the product or the system that works and meets all stipulated requirements."

One of the objectives of this study is to define what makes up a corporate memory and, more particularly, how to capture technical memory for a CE project. In order to timely complete a CE project, individual teams must be guided by lessons learned from their past experiences. This means such experiences must be adequately captured in this technical memory or knowledge base. Such historical knowledge elements, if available in the technical memory (TM) form, can come in handy for designing similar types of new products. Examples of some knowledge bases in TM are:

- how products were initially decomposed
- how teams were designated—which tasks were designated to which team
- how teams were composed—which fields they specialize in
- how design propositions were considered
- how conflicts were solved
- when decisions were made and who made them
- how the teams arrive at those decisions, etc.

The experience of an expert and his or her right interpretations are often difficult to pin down due to an inherent complexity in decision-making—the way an expert reasons about a task and the lack of generality present in such setting. To leverage designers' know-how in CE tasks, a historical trace of their experiences can be captured into a library of knowledge components covering the domain of his or her competence. Knowledge captured in such types of models is often explicit and should be accessible to all PDT members not just a handful of experts. This way others—less knowledgeable members of the PDT—can use the knowledge models resident in TM for purposes such as professional training and decision making guides. The work reported herein employs a number of knowledge engineering concepts from literature for defining such Generic library of knowledge components.

In this paper, the authors have studied several models of conflict management and work-group tasks. Based on such studies, a generic conflict management library and a methodology to use this library are described. The generic library is

*Author to whom correspondence should be addressed.

1063-293X/98/04 0274-14 $10.00/0

intended to help guide the product development teams (PDTs—engineers, designers, process planners) to come up with a conceptual model at the Knowledge Level [18]. In such conceptual models, knowledge rules are made explicit. The work reported in this paper is a part of the Genie** project, which was partly directed towards analyzing CE tasks, to guide PDTs for modeling these task, and for defining a corporate technical memory.

In order to determine when conflicts can occur in Concurrent Engineering, the authors first define a knowledge structure of the CE work-tasks based on models studied from the literature. To manage conflicts, it was first necessary to detect conflicts and to determine their types and degree of severity [13]. Argumentation and Negotiation are common methods to resolve conflicts. In argumentation a participant tries to influence other participants so that they can change their opinions [6]. They together try to negotiate a solution for a detected conflict.

In Section 2, the authors present an overview of this library of Generic knowledge components. Section 3 outlines a methodology for managing conflicts. A recommended set of guidelines, to reuse such knowledge components from this generic library and to adapt them in building a framework for real applications, are described. An example in aeronautics engineering illustrates the use of this generic library.

2. A Generic Library of Components to Model Concurrent Engineering Work-Tasks

Some of the key problems in knowledge engineering are how to rationalize the experts' know-how, capture their contents, and how to represent them in some explicit forms. Roles such knowledge components play in problem solving, if they are part of a conceptual model, is generally not so obvious. Generic knowledge components provide a theoretical basis to define such type of models. Types of Knowledge dictate what types of knowledge acquisition approaches, analysis and inferencing will be best suited.

A number of knowledge acquisition approaches have been used in the literature. They provide KE guidelines for capturing mono-expertise task, task types and for building Generic knowledge components library. Examples include the Role Limiting Methods [17], Generic Tasks [10] and the CommonKADS library [5]. This CommonKADS library example is more complete than the others and provides a number of guides to select knowledge components and to adapt them for a particular application.

For studying the modeling guides for the CE work-tasks, the authors have extended the CommonKADS generic library to handle multi-expertise tasks. To start with, the authors define a library of Generic knowledge components for CE and for the conflict management tasks. These knowledge components are employed as building blocks to model the product development process of a particular CE project.

**Genie project is a common project of INRIA and Dassault-Aviation Co.

These knowledge components are written in the CommonKADS Conceptual Modeling Language (CML) [22] using the CommonKADS utilities.

The definition of a generic architecture is based on the analysis of a number of real CE work-tasks and abstract theories including abstract global characteristics to distinguish a task type. In order to define this generic library of knowledge components, the authors studied a number of models for CE tasks and Conflict Management methods reported in the literature. The following steps were used during this study:

1. In the beginning, the authors first represented such models in a single formalism (CML). This allowed us to determine their characteristics and to combine their similar parts/ features.
2. Then, the authors abstracted a general representation of these models.
3. Simultaneously, the authors determined circumstances and conditions under which conflict management methods can be applied and which types of conflicts these methods can manage. This allowed the team to use the index of the library as a criterion for component selection. In the sequel, we present an overview of this library.

2.1 Generic Knowledge Models

A number of knowledge models for the CE work-tasks are reported in the literature such as Bond's model [2] and Brazier et al.'s model [4], etc. The characteristics of CE projects differ from the aforementioned models in the following way:

1. There are needs to represent knowledge both as shared and private knowledge-models during a CE collaborative session. A private knowledge-model belongs to a participant, a PDT or a work-group of teams and describes his or her knowledge. Expertise model is an example of a private model (i.e., an explicitation of his knowledge onto a model) of a participant. An expert model is normally private meaning they are not shared. Shared models, on the other hand, describe shared knowledge corresponding to a system that is being designed.
2. Knowledge models can correspond to a number of life-cycle functions. Thus, with the help of such models, modifications can be made in requirements as well as in the artifact's definition. Thus, conflicts can occur anywhere during the life-cycle. The knowledge models can be helpful in revealing and managing requirements as well as modifying the artifact's definitions.

After studying many models of CE work-tasks, the authors have proposed herein a library of generic knowledge components for managing concurrent engineering tasks (shown in Figure 1). We distinguish four main sub-tasks in this model:

- *Modeling tasks:* Relying on the private knowledge (Private model), each designer generates a number of design propositions to satisfy a given set of requirements. The process is similar to a manual process of how a designer

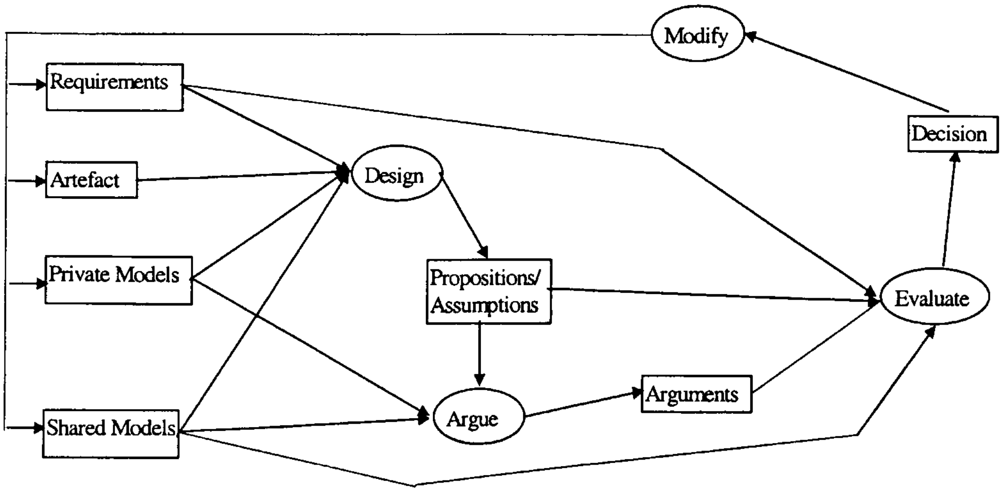

Figure 1. Concurrent engineering data flow.

would deal with a design task. Generic life-cycle knowledge models represented in the CommonKADS library [5] can guide the modeling of these tasks over the artifact's life-cycle.

- *Argue tasks:* To promote the acceptance of one's propositions by a group, a participant justifies them to the PDT with a number of arguments. Assumptions made in the "modeling tasks" stage are used to determine arguments and to define them. Argumentation tries to change other participant's opinions by justifying the utility and the necessity of a proposition [6]. It forms an important part of negotiation, which aims at solving a conflict.
- *Evaluate task:* The work-group evaluates the integration of propositions in the artifact. Propositions may not satisfy participants' (PDTs) needs and conflicts can appear. So, a principal sub-task in this evaluation consists of detecting and solving conflicts.

Propositions can concern the requirements, product definitions, process definitions or delivery and support functions or the artifact's life-cycle. Decisions about which life-cycle object (from requirement model to support models) models can be modified in the next process cycle and the order, in which life-cycle modifications can be made to the artifact, are taken into consideration as a subset of «Evaluate library» tasks.

2.2 Conflict Management in CE

Before describing a number of methods that can manage conflicts, let us determine which types of conflicts can appear in a CE process.

2.2.1 CONFLICTS IN CE

We identify two classes of conflicts that occur in CE:

1. First, *conflicts among life-cycle propositions (e.g., between a design phase and a requirement phase, etc.).* In other words, a design proposal suggested by a design team or a designer does not satisfy the requirements imposed by the specification team. This type of conflict is handled in the same way as in individual design, where a single designer tries to generate a design in order to satisfy requirements. CommonKADS library [5] and Brazier [4] offer a number of Generic knowledge models to assess the design maturity and ensure that it fulfills the requirements. This class of conflicts is beyond the scope of this paper.
2. Second, *disagreements among teams (PDTs) in the work-group.* Such conflicts normally arise either from conflicting strategies used by the team or from interactions that take place among the teams themselves:
   - *Strategic conflicts:* The incoherency between methods and tools use by different designers in a CE workgroup causes strategic conflicts. This kind of conflict also arises from an incorrect distribution of tasks within the group [12]. Other causes can be failure in cooperation between participants or a lack of well-defined responsibility [4].
   - *Conflicts about propositions:* These types of conflicts occur from:
     - ❖ (a) misunderstanding between the participants' terminology and points of view [20], or
     - ❖ (b) unacceptance of the conditions under which a proposition has been defined.

   Preconditions problems can be unsatisfied needs [25], resource contention [12] or different interpretation of requirements. Constraints imposed by the proposition and its interaction with other propositions can generate undesirable consequences. Note also that the lack of quality of a proposition can cause its unacceptance.
   - Figure 2 shows a typology of such conflicts. This typology emphasizes the nature of conflicts with problems of objects that are commonly found.

Often it is necessary to determine the nature of such con-

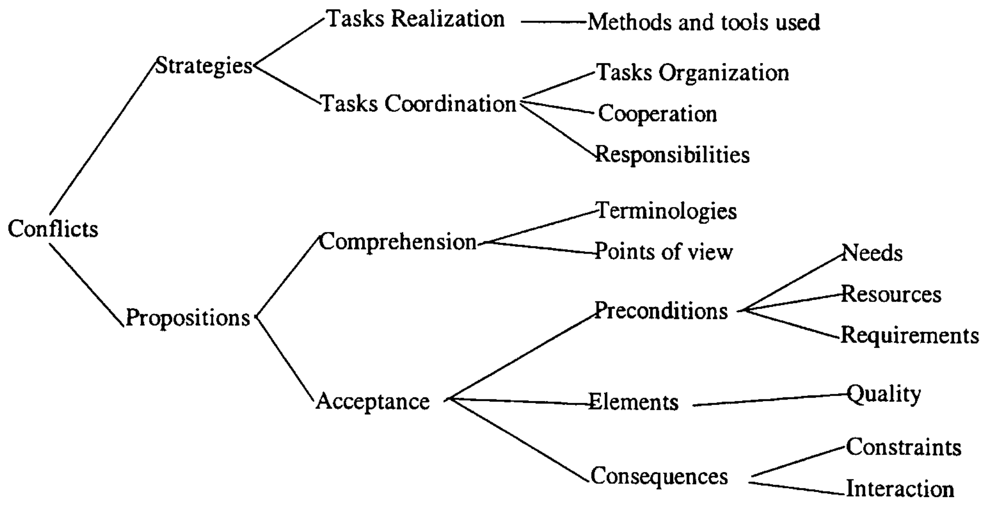

Figure 2. A typology of conflicts.

flicts, their origins and sources, to apply appropriate methods to solve them. In a library, a number of methods are presented as Generic knowledge components, which guide to manage these types of conflicts between several teams in the work-groups.

The conflict management task model (shown in Figure 3) consists of first, detecting a conflict (its nature) and then solving it. Detection task can be the same as a diagnosis task, which helps to locate the problem. When the conflict is detected, a solving method is selected and applied to solve this conflict. The results display, if the conflict is resolved or not. If the conflict is not resolved, another method can be selected and applied and so on.

2.2.2 CONFLICT MANAGEMENT METHODS

We distinguish four classes of methods:

1. *Prevention methods.* In CE, several multidisciplinary teams (engineers, designers) collaborate together [19]. Each team (say a designer) advances a specific viewpoint based on his in depth knowledge of what he considers applicable to this situation. Ramesh proposes to share openly part of private knowledge to avoid terminology and interpretation conflicts [20]. Group's characteristics can be a cause of conflicts. Easterbrook recommends [12] some strategies, which consider human's characteristics and relationships in order to form a work-group.

   Such methods can be associated to the CE «Design»sub-task (Figure 1) to avoid potential conflicts amongst group of designers.
2. *Argumentation methods.* Other methods are proposed to help a PDT or its participant to justify his proposition. Argumentation is used in CE to defend a proposition and to persuade a team to choose it [6]. Arguments must be explicit to avoid conflicts. K. Sycara [24] proposes some strategies based on general principles and heuristic rules to define arguments. B. Ramesh [20] favors determining interdependencies among the teams or tasks for detecting potential differences.
3. *Conflict detection methods.* Conflict detection methods

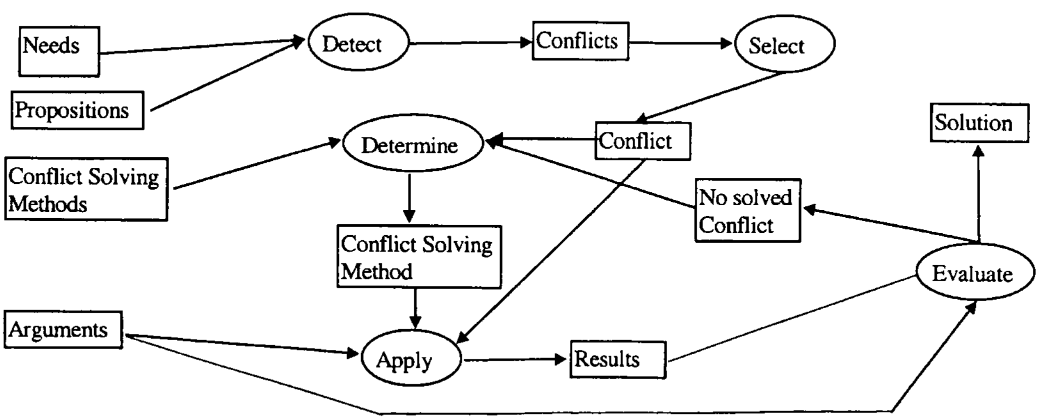

Figure 3. Conflict management data flow.

have not been thoroughly made explicit in the literature. Klein considers a conflict as an exception (in the CE cycle) and provides a typology of the exceptions nature [13]. He also recommends some strategies to handle exceptions. Conflict detection can be assimilated to a diagnosis task. So, diagnosis methods like those presented in [1] can be used to detect conflicts. Conflict symptoms can be first detected and contributors (opposed propositions and arguments) are analyzed as hypothesis in order to find conflict's causes. Our conflict typology (Figure 2) determines the nature of a conflict and helps to detect elements, which cause it to occur.

4. *Negotiation methods.* The large number of negotiation methods has be used to solve undetected conflicts. Some of them are like «Locate a consensus», that leads to forcing a consensus and to choose a solution. It can solve mixed-motive conflicts [20]. Another method is like «Introduction of a Third party», where a third party imposes a solution [12]. This method is designed to solve resource conflicts and also strategic conflicts.

Other methods are proposed in some other conditions. For example, to apply Case-Based Reasoning (CBR) [25], first a number of past cases must be stored. Second, at least one case similar to the current one must be selected. CBR recommends retrieving conflict solution from an appropriate previous case and adjust (fine-tune) it as a solution to the current conflict. Another method proposes a «Goal Graph Search» to add goals or to substitute rejected goals in order to modify tasks decomposition and attribution [26]. To be applied, this method needs goal graphs organized as influence trees. K. Sycara [25] also recommends strategies to define «Counter-propositions» or to modify rejected propositions [25].

2.3 Representing Knowledge Components in the Library

Each task is described in the library with its goals, specification Inputs, Outputs, etc. (Figure 4). For example, the «Argue» task consists of defining a number of arguments, which justify a proposition and seek its acceptance by other PDT participants. Figure 4 shows agents, who performs a task and to whom results must be communicated. For instance, the «Argue» task will be performed by one designer. Generated arguments will be communicated to the entire PDT group.

A global view of the associations between CE sub-tasks and conflict management methods can guide a knowledge engineer to identify appropriate conflict management methods in order to define a model for a CE task in a particular domain. These associations help to determine conflict management methods in a real CE application. We present trees of associations, CE-sub-tasks, and corresponding methods in our library. For example, the tree of argumentation methods (Figure 5) shows a global view of possible strategies, which can be used to justify a proposition. These trees are also used to supplement the description of a method.

The competence and the acceptance described by the method tell the knowledge engineer what the method would be able to perform and in which conditions this method can be applied. For example, the «Interruption Threats» method (Figure 6) guides PDT to formulate threats in order to incite designers to accept the rejected proposition.

To apply this method a designer may need politic and the economic state information. He will use this information to calculate the interruption cost of the design process. This cost helps to formulate threats as necessary conditions for making the design progress. Data flow (Figure 6) shows the interaction between tasks and input or output knowledge.

2.4 Conflict Types as Another Library Index

We use our conflict typology (Figure 2) as another index for selecting methods from our generic library of knowledge components. As we said before, we distinguish strategy conflicts and proposition conflicts. Associations of prevention, argumentation and negotiation methods to conflicts are used

<u>Task «Argue»:</u>
Definition:
Goal: Define arguments in order to defend propositions.
Specification: Define a number of arguments in order to defend propositions. Assumptions made during propositions generation help to evaluate the impact of arguments in the situation. These assumptions define contexts which arguments are valid.
Input: Requirements, SharedModel, PrivateModel, Propositions, Assumptions
Output: Arguments
End definition.

<u>Allocation and Communication</u>

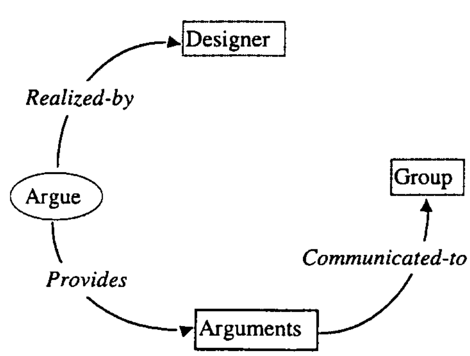

Figure 4. Tasks description.

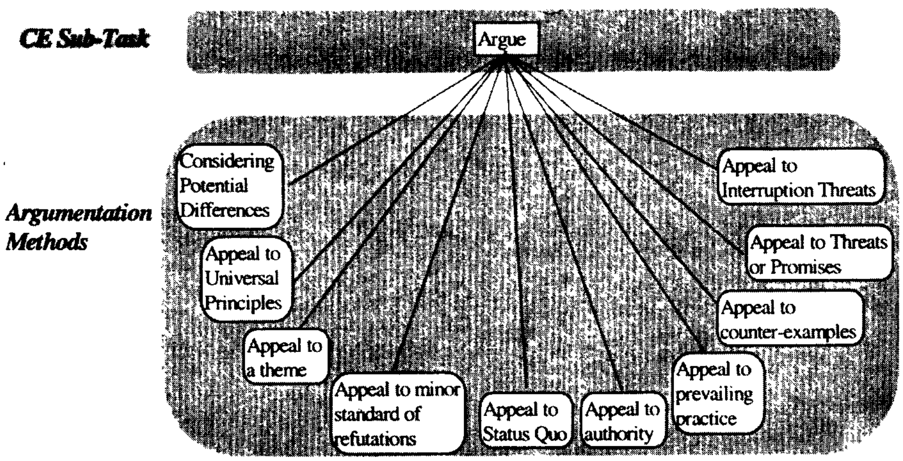

Figure 5. Argumentation methods.

Method «Interruption Threats»:

Competence: Guide to generate threats. If a conflicted proposition is not considered in the solution, the overall design process can be interrupted.

Input: Factors (Project Information), Conflict, Proposition Payoff

Output: Interruption Threats

Sub-tasks: Calculate interruption cost, Generate threats

Additional Roles:

Control:

Calculate (Factors -> Interruption Cost)

Generate (Conflict, Proposition Payoff, Interruption Cost -> Interruption Threats)

Acceptance: Information about project and company's design and economic policies.

Method «Interruption Threats»: Data Flow

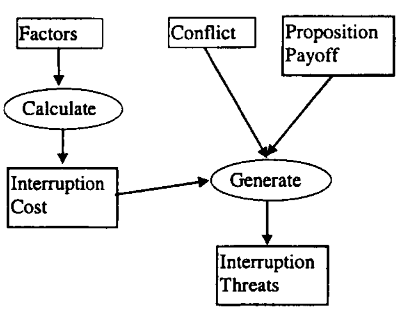

Figure 6. A method description.

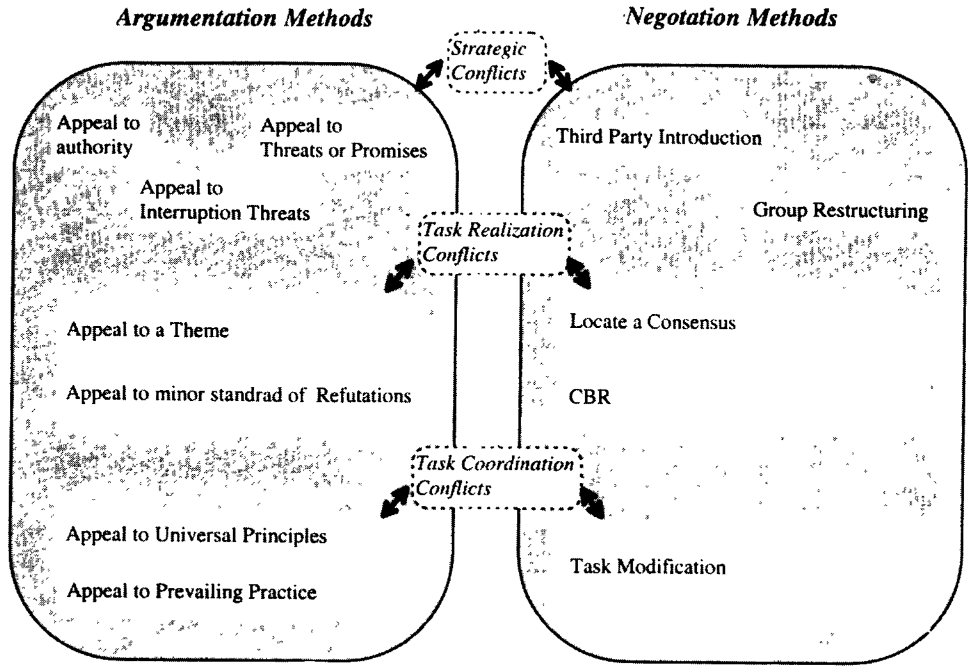

Figure 7. An example of conflict types/generic methods association: Methods to manage strategic conflicts.

as guides to determine a number of selection criteria as index. We will indicate the methods associated to strategic conflicts and to the proposition conflicts.

1. *Strategic Conflicts.* In order to manage strategic conflicts (Figure 7), the Appeal to the Authority, to Threats or Promises and to Interruption Threats can be used in the argumentation. Negotiation methods such as Group Restructuring or a Third Party Introduction will help conflict solving. Conflicts stemming from task realization

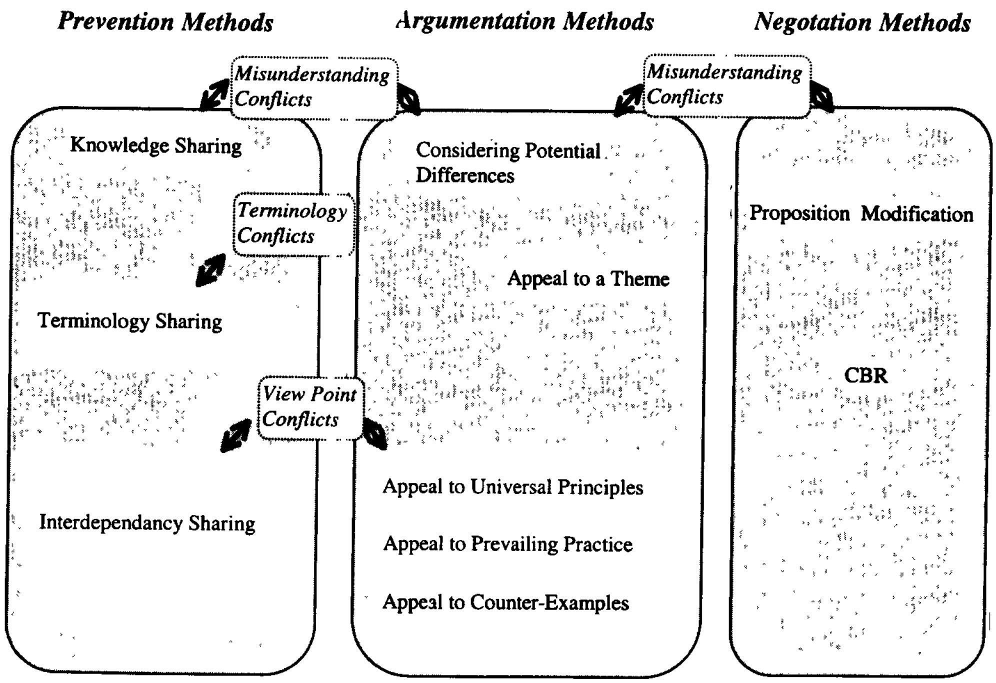

Figure 8. An example of conflict types/generic methods association: Methods to manage conflicts that appeared from misunderstanding of propositions.

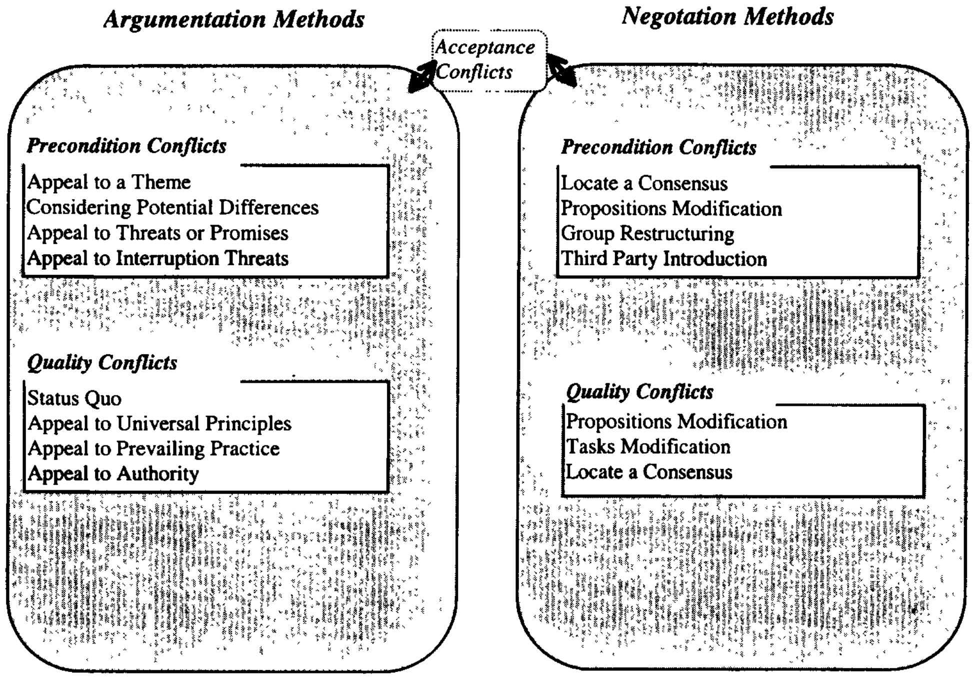

Figure 9. An example of conflict types/generic methods association: Methods to manage proposition acceptance conflicts.

can be in discordance about methods and tools or in missing to execute tasks. They can be managed with an appeal to a number of rules extracted from a base that is considered as a standard of refutation in a field. Rules related to a particular theme can help also to define arguments. Extracting solutions from past cases, or locating a consensus among designers will help to solve task realization conflicts. The modification of task decomposition and tasks attribution helps to solve coordination conflicts (like conflicts about task organization and cooperation problems).

2. *Conflicts about propositions.* Knowledge sharing is necessary to prevent problems caused by misunderstanding of propositions provided. Considering Potential Differences between designers and "Appeal to a theme" can be used in this case as argumentation methods. Proposition Modification is an interesting method to be used in the negotiation. Figure 9 shows how some methods can manage acceptance conflicts.

We have expressed a formal representation of this library in the CML language [22], using Cokace, a CML dedicated environment [11] to make easy their exploitations. An HTML format is also generated automatically using this tool. The library is accessible at the URL: http://www.inria.fr/acacia/Cokace.

In this part, we have presented our study to define a generic library of conflict management methods. In this library, we have described at the same level of abstraction, specific and general methods proposed in the literature. So, the library offers guidelines, extracted from conflict management experiences in either specific or general domains. Our library can be easily extended.

In the second part of the paper, we present a methodology for using the library to model a CE task in a particular domain.

3. How to Use the Library to Model CE Applications

Generic knowledge components library is suitable to analyze knowledge extracted from a particular domain and to define a corresponding virtual model. In the paper, the authors describe how to combine knowledge abstraction with Generic knowledge components reuse to model CE applications. Furthermore, the authors illustrate their ideas by taking examples from aeronautics engineering.

3.1 Combining Knowledge Abstraction and Generic Knowledge Components Reuse

Knowledge engineering is commonly known as a cycle of knowledge extraction, analysis and structuring [23] in knowledge acquisition community. A number of such approaches recommend Generic knowledge component reuse to model mono-expertise tasks. The CommonKADS [5] approach guides a knowledge engineer (who acquires knowledge from experts and builds a conceptual model) to select Generic knowledge components and to adapt the appropriate ones to a particular application. In this approach, a library of Generic knowledge models for several task types is also de-

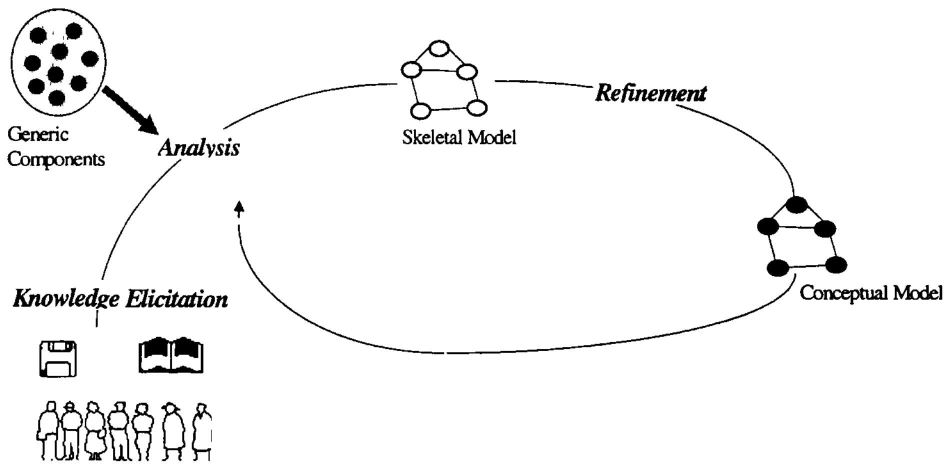

Figure 10. The knowledge modeling cycle.

fined. MACAO-II [14] proposes a methodology which combines abstraction and knowledge components reuse to build a conceptual model of the application. In this method, different levels of abstraction reduce the gap between knowledge extracted from the expertise and Generic knowledge components.

In this paper, we propose a modeling methodology based on these guides. We propose a modeling cycle based on different levels of abstraction and Generic knowledge components selection and adaptation. Dimensions like who realizes a task, interaction between participants and conflict management, must be considered to model CE tasks. Figure 10 shows the main steps of our modeling cycle.

3.1.1 KNOWLEDGE ELICITATION AND ACTIVITY IDENTIFICATION

After a feasibility study and an identification of users needs [9], the knowledge engineer is invited to familiarize with the task and to identify the general activity of the group. He has to evaluate documents and interview experts. Interviews must be focused on the activity of each expert, his interaction with others (what he needs, from whom, from where and what he communicates and to whom).

Interviews and documents can be analyzed to define an activities models of participants' sub-groups. In these models, the inputs/their origin, the outputs/their destination, and actions/their actors must be determined.

3.1.2 TASK MODEL DEFINITION

A cycle of analysis, structuring and refinement is performed to build a task model. The analyzing and the structuring can be realized by combining classifying activities selecting and adapting Generic knowledge components, (Figure 11) in order to build a skeletal model which guides knowledge extraction and model refinement. At first, we recommend to define a general model of the CE task and then

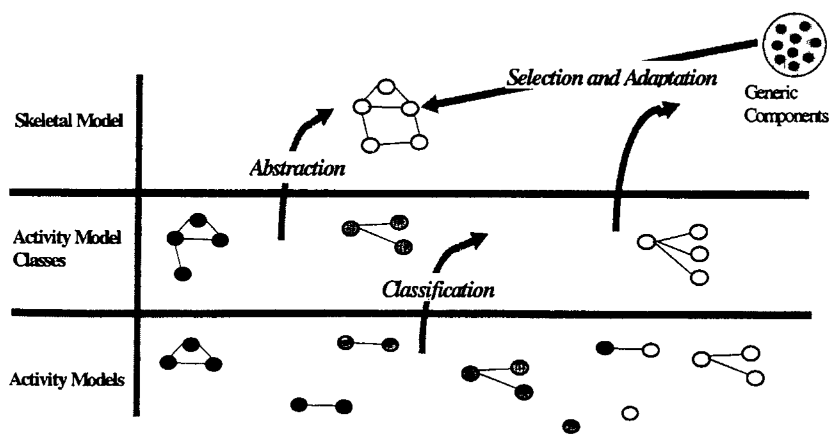

Figure 11. The definition of a skeletal model of the task by combining abstraction and generic components reuse.

focus the analysis on conflict management.

(a) *Classification of activity models.* This step is an intermediate phase of abstraction in which a classification of activity models is done. For example, all specification activities correspond to a specification class. This classification allows to identify if some Generic knowledge components can be associated to an application in a particular domain.

(b) *Generic knowledge components reuse to define a skeletal model.* A number of generic knowledge components can be selected and adapted in order to build a skeletal model. First, analysis is focused on the global CE task and then conflict management. The CE generic model, (in the library) guides to distinguish individual tasks from collaborative ones. It also guides to identify participants' needs, goals and interaction of knowledge (the input/output data flows).

In order to structure the conflict management task, the types of conflicts that appear in a particular CE task need to be first determined. Our conflict typology guides to focus the analysis on objects that contribute to a conflict. Our library index (Figures 7 and 8), in which conflict management methods are associated with conflict types is helpful to choose conflict management methods (prevention, argumentation and negotiation ones). Selected methods can be then adapted to the particular CE task.

(c) *Model refinement and updating.* The skeletal model guides knowledge acquisition in order to refine and to enrich it. Knowledge acquisition can be realized from documents, translated interviews, activities models and from meeting with experts. The refinement of the model leads to update its skeleton.

3.2 Modeling an Application in Aeronautical Engineering Domain

The application concerns the development of critical systems in the aeronautical domain at Dassault-Aviation Company. This type of system involves a number of fields like hydraulic, system architectures, computer science, etc. They define the brake of an aircraft [27]. Systems are decomposed into software and hardware parts.

The main objective of the application is to define a corporate memory of the design of such type of systems in order to capitalize design know how. This helps avoid errors that occurred in the past and to solve similar problems in current and future design of aircraft brakes.

Our methodology, (Cf. 3.1), has been used to model the experience of a brake design. We cannot describe in detail this project for proprietary reasons. So, in its absence, we will try to illustrate the modeling process and how the generic library of knowledge components is used.

Note also, that we analyzed only the activity of the Dassault-Aviation participants.

3.2.1 IDENTIFICATION OF ACTIVITIES

Sub-contracting companies collaborate with Dassault-Aviation to design brake systems. A number of experts from Dassault-Aviation have been interviewed. A manager group

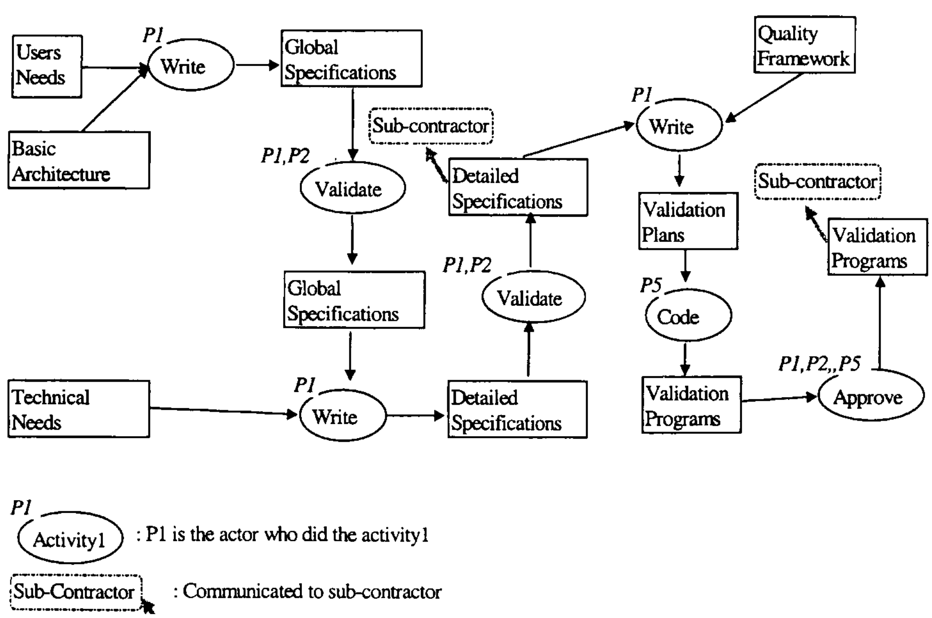

Figure 12. An example of an activity model.

is designated to supervise the design and the collaboration between subgroups. Other groups in the Contractor Company (Dassault-Aviation) define the specifications of a system. Sub-contractor groups have to satisfy these specifications by generating design propositions. Contractor groups evaluate, with sub-contractor groups, the design propositions. Modifications can be made in propositions and in specifications.

After analyzing the results of the interviews, PDT defines a number of activity models related to the Dassault-Aviation team of participants. An example of such a model is shown in Figure 12. It depicts specification activities. One PDT participant writes the global specifications, which are later validated by the head of the specification group. Detailed specifications are then obtained by the same participant, and approved by the head of that group. These specifications are communicated to the sub-contractor thereafter in order to define corresponding detailed specifications. Then, another sub-contractor's group generates design propositions in order to satisfy the detailed specifications. Detailed specifications are also communicated to another PDT, who writes validation plans and writes corresponding programs.

A large volume of information was initially gathered. Interviews were conducted in several steps. First, designers talked about their activities in general, their fields and then, they explained what types of interactions took place between design and others. It was hard to interview them about where conflicts had occurred. Their answers were almost as if there were no conflicts. But, when they talked about details of their activities and about problems they have actually encountered, a number of discordances surfaced in their discourse. So, it was easier (as we recommend in our methodology) to analyze globally the CE process and then to focus on the conflict management.

3.2.2 CLASSIFICATION OF GROUPS' ACTIVITIES

Classifying activities into categories allowed us to distinguish the different groups, implicated in the project and to identify the task of each of them. So, we distinguished a number of categories respecting a number of task types and sub-groups (like specification, propositions validation). For instance, the Specification category model (Figure 13) represents the specification task in the application. This task is related to software (a part of the activity model shown in Figure 12) and hardware groups. Both of them follow the same scheme to define specifications. One participant analyzes needs and basic architecture in order to determine a number of specifications related to its part. These specifications are evaluated by the whole specification group. Unapproved specifications are modified by their author and so on. Specifications are detailed at different steps of the specification cycle.

3.2.3 SKELETAL MODEL DEFINITION

In CE generic model, three tasks ("Design," "Argue" and "Evaluate") are proposed. We can note that "Design" and "Evaluate" are used in our application. In fact, sub-contractor groups generate design propositions to satisfy given specifications. The group uses their private knowledge to propose a design.

A number of validation procedures are applied by the contractor to validate propositions. Decisions are made to modify specifications or propositions. Argumentation is also used in the evaluation to force the acceptance of a proposition and to justify modifications to be done.

The group is formed (sub-contractors are selected), global requirements (cost, delay) are determined and detailed specifications are defined in a preparation stage. Figure 14 shows the skeletal model for this application.

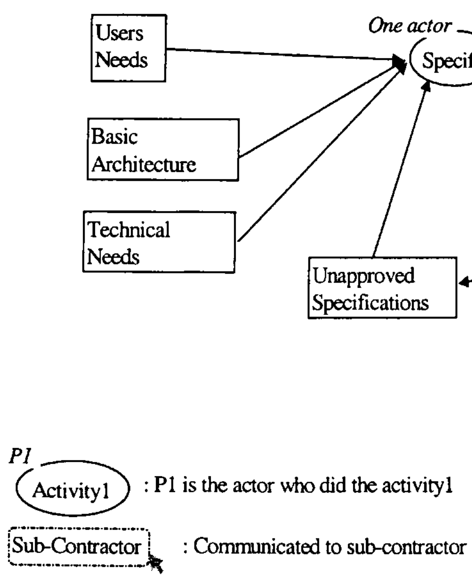

Figure 13. An example of the specification category model. A cycle of specification (by one actor) and evaluation (by the specification group).

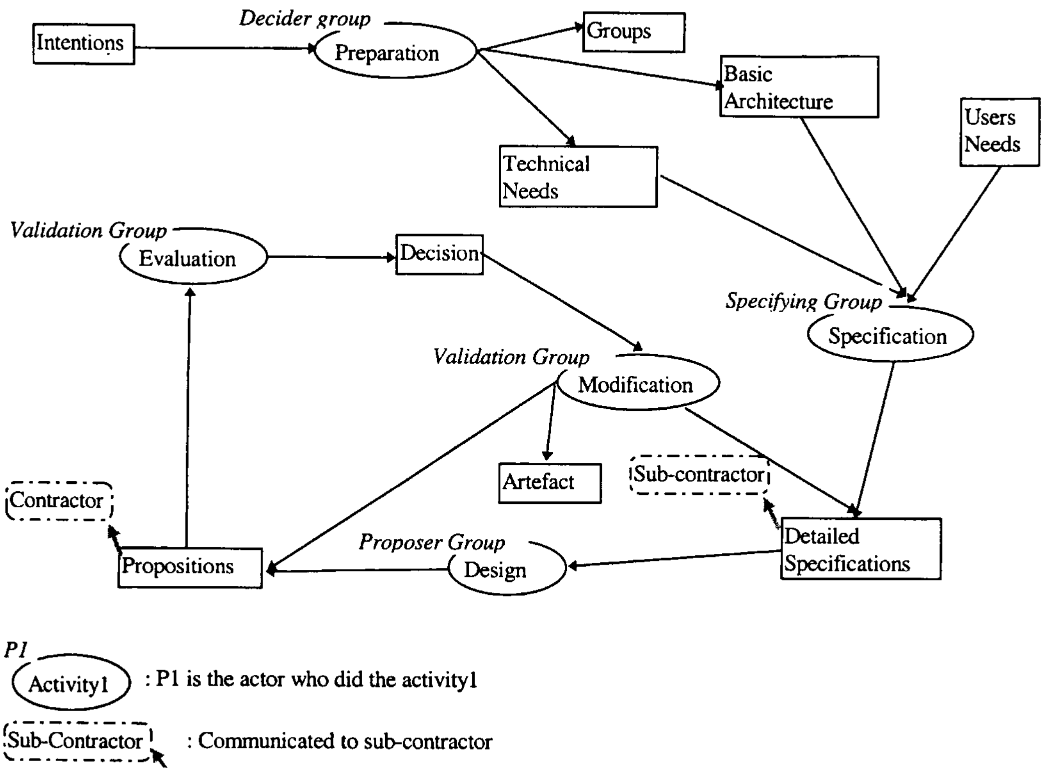

Figure 14. The skeletal model defined for the application.

It contains the global tasks performed in a brake design project. This model will be used to guide the definition of similar project organization in the future. It marks out the different steps in which problems and conflicts can occur.

At the next step we are going to identify methods used to detect and solve conflicts. Our conflict typology guided us to identify a number of conflicts that appeared between designers. There were a number of problems corresponding to resources conflicts (costs, delays). These types of problems appeared especially between the contractor and subcontractors. The principal methods used to solve this type of conflicts were to modify global requirements (cost, delay changing and/or modifying the specifications) or to impose the decision of the contractor (no changing of resources). Another type of conflict was observed when the requirements were interpreted differently by each designer and when different methods and tools were used by these designers. Designers had to modify their propositions in order to match together the interfaces between the parts of the brake system.

Cooperation conflicts were solved by restructuring teams. In some cases, the number of team members was increased to accomplish some tasks (i.e., one person was exclusively in charge writing validation plans). This allowed redistribution tasks in accordance with people skills.

The conflict analysis described before was still incomplete. The next step will allow us to interview experts following our conflict typology and generic methods.

The corporate memory resulting from this analysis will be of great use in detecting and managing conflicts in future projects.

Studying such applications showed us how difficult the conflict analysis is and arouses some sensitivities among designers. Even if designers worked in CE environment in which all of them were at the same level of responsibilities, they do not dare to mention problems occurring among them for fear of offending their hierarchy. Since conflict management takes a prominent part in CE projects it is important to define methods to make the extraction of information.

4. Conclusion

A memory of past projects constitutes an excellent aid for designers. It facilitates the definition of new projects and the management of problems already encountered. This paper presents our work in defining such memory and especially it provides guidelines to represent this memory as a conceptual model, which makes the knowledge accessible to different users (designers, deciders).

Modeling multi-expertise application (especially CE projects) is a complex task. The comprehension of knowledge gathered from experts is difficult. Several dimensions like communication and conflict management must be taken into account.

The use of guides like methodologies and generic knowledge models was already helpful in modeling mono-expertise applications. In this paper, these guidelines are extended multi-expertise applications. It comprises a library of generic knowledge models (generic CE task models and generic conflict management methods). These models give a theoretical scheme to analyze and model CE projects. We describe a methodology allowing the use of the library. This methodology suggests different levels of abstraction (activities modeling and classification into categories) to reduce the gap between the generic knowledge models and the knowledge extracted from a given application. This facilitates the reuse of generic knowledge components from the library. We note that in CE, a general model of the CE must be first defined and only then, conflict management can be analyzed and modeled.

We have developed a knowledge server "Cokace on the Web" that guides the access to models described in CML language and facilitates the search of their knowledge components [16]. It offers different ways to access our library and facilitates the selection of methods and tasks. This knowledge server is accessible via: URL: http://www.inria.fr/acacia/Cokace.

We are currently studying the representation of shared knowledge in CE projects using viewpoints notion [21]. A multi-agent system has been developed by our team [7] and [8]. The system guides designers through the structuring of their propositions into a common formalism in order to facilitate the communication between them. It facilitates the proposition comparison in order to detect conflicts, and suggests the use of the library to solve them.

Acknowledgement

We deeply thank the French "Ministere de l'Enseignement Superieur et de la Recherche" and Dassault-Aviation (Genie Project) that funded this research. We thank also Dr. Rose Dieng, Ms Peggy Abreu and Mrs Mary Maty for their remarks.

References

1. R. Benjamins, "Problem Solving Methods of Diagnosis," Thesis Univeriteit van Amsterdam, Amsterdam, 1993.
2. A. H. Bond, "A Computational Model for Organizations of Co-operating Intelligent Agents," *Proceedings of the Conference on Office Information Systems,* Cambridge April, 1990.
3. S. Boyera, O. Corby, *Etat de l'art sur la multi-expertise, Projet Génie Thème 3,* Rapport de contrat GENIE, Thème 3, Sophia Antipolis, June 1995.
4. F. M. T. Brazier, P. H. G. Van Langen, J. Treur, "Modeling Conflict Management in Design: An Explicit Approach," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* vol. 9, n. 4, Cambridge University Press, USA, 1995, pp. 353–366.
5. *CommonKADS Library for Expertise Modeling Reusable Problem Solving Components, Frontiers in Artificial Intelligence and Applications,* J. Breuker and W. Van de Velde (Eds.), Amsterdam: IOS. Press 1994.
6. C. Castelfranchi, "Conflict Ontology," *Proceedings of ECAI'96 Workshop on Modeling Conflicts in AI,* H. J. Muller, R. Dieng (Eds.), Budapest August 1996.
7. C. Cointe, N. Matta, "Architecture to Handle Concurrent Engineering," *1st International Engineering Design Debate,* Glasgow, September, 1996.
8. C. Cointe, N. Matta, "Multi-Agents System to Support Decision Making in Concurrent Engineering," to appear in *Tools and Methods for Concurrent Engineering,* Manchester, April 1998.
9. C. Chabaud, J. L. Soubie, F. Buratto, N. Lompre, *Cycle de vie des systemes a base de connaissances: Objets et methodes de validation ergonomiques,* Rapport Interne IRIT, IRIT/91-48-R, Juin 1991.
10. B. Chandrasekaran, B. Johnson, J. Smith, "Task-Structure Analysis for Knowledge Modeling," *Communications of the ACM,* vol. 35, no. 9, September 1992.
11. O. Corby, R. Dieng, "Cokace: A Centaur-Based Environment for CommonKADS Conceptual Modeling Language," *Proceedings of ECAI'96,* Budapest August 1996.
12. S. M. Easterbrook, E. E. Beck, J. S. Goodlet, L. Plowman, M. Sharples, C. C. Wood, "A Survey of Empirical Studies of Conflict," *CSCW: Cooperation or Conflict?* S. Easterbrook (ed.), Springer-Verlag 93.
13. M. Klein, "Conflict Management as Part of an Integrated Exception Handling Approach," *Proceedings of ECAI'96 Workshop on Modelling Conflicts in AI,* H. J. Muller, R. Dieng (Eds.), Budapest August 1996.
14. N. Matta, *Methodes de Resolution de Problemes: leur explicitation et leur representation dans MACAO-II,* PhD Thesis, University of Paul Sabatier, Toulouse, October 1995.
15. N. Matta, "Conflict Management in Concurrent Engineering: Modelling Guides," *Proceedings of ECAI'96 Workshop on Modelling Conflicts in AI,* H. J. Muller, R. Dieng (Eds.), Budapest August 1996.
16. N. Matta, C. Ros, O. Corby, "A Generic Library to Guide Decision Making in Concurrent Engineering," to appear in *Tools and Methods for Concurrent Engineering,* Manchester, April 1998.
17. M. Musen, "Automated Support for Building and Extending Expert Models," *Knowledge Acquisition: Selected Research and Commentary,* S. Marcus Editor, Kluwer Academic Publishers, Netherlands 1990.
18. A. Newell, "The Knowledge Level," *Artificial Intelligence Journal,* 19 (2), 1982.
19. B. Prasad, *Concurrent Engineering Fundamentals, Volume I: Integrated Product and Process Organization,* One Saddle River, NJ: Prentice Hall, PTR, 1996.
20. B. Ramesh, K. Sengupta, "Managing Cognitive and Mixed-Motive Conflicts in Concurrent Engineering," *Concurrent Engineering: Research and Applications,* vol. 2, no. 3, 1994, pp. 223–236.
21. M. Ribière, N. Matta, C. Cointe, "A Proposition for Managing Project Memory in Concurrent Engineering," to appear in *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA '98),* Churchill, Australia, February 1998.
22. G. Schreiber, B. Wielinga, W. Van de Velde, A. Anjewierden, "CML: The CommonKADS Conceptual Modeling Language," *Proceedings of EKAW '94,* Lecture Notes in AI n.867, L.

Steels, G. Schreiber, W. Van de Velde (Eds.), Bonn: Springer Verlag, September 1994, pp. 1–25.

23. L. Steels, "Tools for Corporate Knowledge Management," *Proceedings (complement) of EKAW'93,* Toulouse and Caylus, September 1993.
24. K. P. Sycara, "Persuasive Argumentation in Negotiation," *Theory and Decision,* 28: 203–242, Kluwer Academic Publishers, Netherlands, 1990.
25. K. P. Sycara, "Cooperative Negotiation in Concurrent Engineering Design," *Computer Aided Cooperative Product Development. Proceedings of MIT-JSME Workshop.* D. Sriram, R. Logcher, S. Fukuda (Eds.), Cambridge, MA 1991.
26. "Problem Restructuring in Negotiation," *Management Science,* vol. 37, no.10, USA, October 1991.
27. P. A. Tourtier, *Rapport intermédiaire de l'étude des metiers et de leurs interactions, Génie thème 3, Phase 1,* Lot 3.1-1, Juin 1995.
28. B. Wielinga, W. Van de Velde, G. Schreiber, H. Akkermans, *The CommonKADS Framework for Knowledge Modelling,* Report of ESPRIT KADS-II project p. 5248: KADS-II/1.1/pp/UvA/35/1.0, September 1992.