Design Propositions Evaluation : Using Viewpoint to manage Conflicts in CREOPS2

Christophe Cointe, Nada Matta, Myriam Ribiere
INRIA (projet ACACIA)
2004 route des Lucioles BP. 93
06902 Sophia-Antipolis Cedex
e-mail: {ccointe,nmatta,mribiere}@sophia.inria.fr

ABSTRACT: Concurrent Engineering is a difficult task in which several designers collaborate together to build a system satisfying some requirements. In this task, individual design propositions are communicated between designers and a consensual solution is determined, by managing conflicts. Requirements and knowledge about states of the artefact are also shared between designers. Viewpoint can be used to represent shared knowledge and to help to manage conflicts in Concurrent Engineering. We describe in this paper our proposition to manage viewpoint in this task, basing on a propositions communication architecture (CREOPS2) and on a conflict typology.

KeyWords: Concurrent Engineering, Conflict, Knowledge Representation, Viewpoint, multi-agents architecture.

INTRODUCTION

In Concurrent Engineering (CE), several participants (designers, managers,...) in different specialities, collaborate in order to construct a system (also called artifact), given the customer's requirements. Each designer has a particular perspective on the design process, depending on his post (who is he in the enterprise organisation?), his domain (context of work), and his knowledge learned from his experience (know-how, competence). This concept of particular perspective called viewpoint is already used in Object Oriented Representation and CE to allow a designer to use shared knowledge from his collaboration with the designer group.

Conflicts can be revealed from disagreements between designers about proposed designs (called propositions) [12]. Such conflicts are detected in CE and negotiation methods are used to solve them.

To handle CE, different types of knowledge will be distinguished and represented, e.g: Task organization, communication protocols, conflict management, knowledge sharing, task realization,...

Our aim is to:

- study CE task modelling and representation to define needs in CE task. We define also generic models to help the modelling of this task and specially the conflict management one. These models will provide generic components to be used (by a knowledge engineer[1]) as building blocks in CE and conflict management task modelling for a real application [11]. We determine a structure to support this task and to represent its particular components. This work is a part of the Genie[2] project, in which the CE task is analysed in order to provide guides for defin-

1.A Knowledge Engineer acquire knowledge and define a model for real applications.

ing a corporate memory.

- offer a knowledge representation, that take aware multiple perspectives. The viewpoint architecture provides excellent representation to structure this variety of perception and also shared knowledge between designers. We studied this type of representation to describe this type of knowledge and also to guide conflict management by emphasizing contributors characteristics of conflicts which can appear in CE, as views.

After presenting a general definition of viewpoint (I.) and its interest to represent shared knowledge in CE (II.), we describe our viewpoint evaluation structure and its utility to manage conflicts between designers (III.) in our propositions communication and evaluation support system (CREOPS2) (IV).

I. VIEWPOINT NOTION

But above all, we have to give a definition of viewpoint because this notion can appear very abstract. Indeed there are a lot of definitions of viewpoint in the literature. They are more often dependant to the context of their application or on the contrary especially generic to avoid the characterization of viewpoint.

In Object Oriented Representation, developers center their research on accessibility and dynamism of models constructed with viewpoints. In TROPES [11] for example, a viewpoint is "a perspective of interest, from which an expert examines the knowledge base". We have to define our own definition of viewpoint, and we want to take into account the way we use them. So we prefer the definition of GARLAN [9] who says that "a viewpoint can be defined as a simplifying abstraction of a complex structure,..., suppressing information not relevant to the current focus", or of Easterbrook [6]: "viewpoint represents the context in which a role is performed".

<u>So our notion of viewpoint must permit:</u>

- simple search of information in a model,
- visualisation of part of information relative to a given process,
- comparison of information between viewpoints.

Our definition of viewpoint can be defined as a subset of information concerning the description of an artefact. This viewpoint is characterized by a context, that allows to locate the part of information that we describe, and a person or a group of person, that let us know what degree of importance we can give to the information of the viewpoint.

II. VIEWPOINT AND CE

In order to emphasize needs to a multi-perspective structure, we describe in the following a model of the CE task.

II. 1. The Concurrent Engineering task

In order to define a library of generic conflict management methods [12], we studied a generic model for the CE task. The CE task model that we defined, is represented at knowledge level [14], in which knowledge characteristics and nature are described. There is a distinction between this level and symbolic level in which implementation choices are made.

2.Genie project is a collaborative project between INRIA and DASSAULT AVIATION.

A number of researchers have studied CE tasks and provided models of this task, like Brazier et al. [3] and Bond [2]. Bond's and Brazier et al's models offer a global view upon the main particularities of CE. From such models we can distinguish some features of CE like:

- The existence of shared models and of private models [2] in CE. Each one of the private models belongs to a participant of the CE task and describes his knowledge. In fact, a private model is the expertise model (i.e. an explicitation of the model of his knowledge) of a participant. These expertise models are private and they are not shared. However, a shared model describes shared knowledge about the system to be designed.

- Modifications can be made in requirements as well as in the artifact [3]. So conflicts can be revealed and managed in requirements as well as in artifact model modifications.

After studying such models of CE task, we propose a generic model for the concurrent engineering task (Figure 1.). We distinguish three main subtasks in this model:

- Design task: Relying on his private knowledge (Private model), each designer generates some propositions to satisfy given requirements. Conflicts between design and requirements appear in this task [12]. We don't consider this type of conflicts in our study.

- Argue task: To promote the acceptance of his propositions by the group, a participant justify them with a number of potential arguments.

- Evaluate task: The group evaluates the integration of propositions in the artefact. Propositions may not satisfy participants' needs and conflicts can appear. So, the principal subtask in this evaluation consists of detecting conflicts and selecting methods to solve them, helped by the structuring of propositions with viewpoints.

New knowledge can be learned from this experience. Then, some private models need to be modified by their respective owners. In the same manner, the Shared Model can be reorganized and modified, because other shared knowledge can be explicited at each cycle. The representation of this shared model must allow modifications and evolution of knowledge.

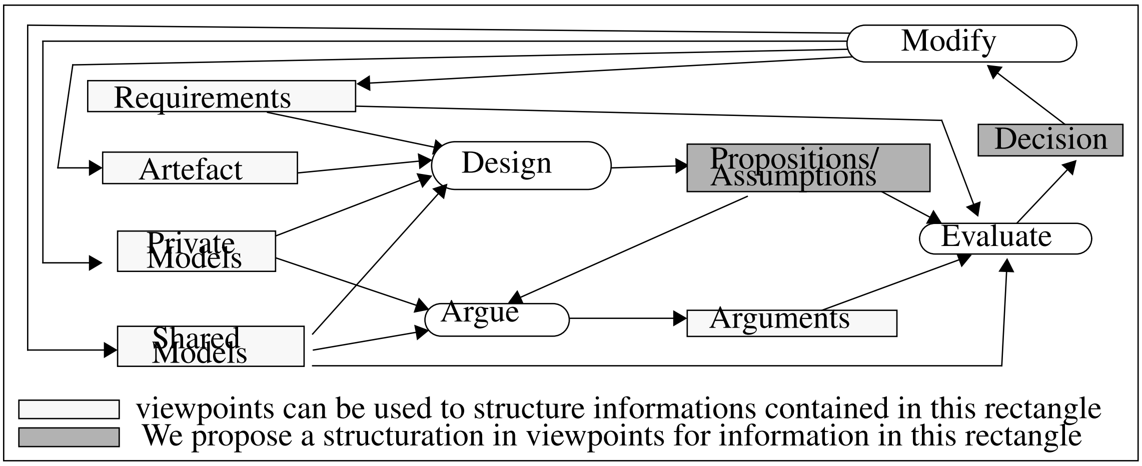

Figure 1. Concurrent Engineering Task Model. Rectangles show tasks input/output, rectangles with rounded corners present tasks.

In (Figure 1.) we suggest the use of viewpoints in structuring requirements, artefact, private models, shared models, arguments... However, we propose only in this paper, the structuring of propositions and decisions with viewpoints.

As we can see, CE is a complex domain, because it involves many actors each of them has his own experience, his own knowledge and his own appreciation of the design situation. Furthermore the designers represent their respective perspectives in different ways. So comparing or finding intersecting knowledge is difficult, because of the difference of representation (terminology, representation schemes). So in CE the designers must cope with problems of:

- Diverse knowledge domain,
- Different representation schemes,
- Different development strategies.

<u>Based on definition of viewpoint, we can define our needs as:</u>

- **individual propositions** (of a person or a group of person),

- **collective decisions** (about the description of a part of the artefact, defined in the context a the viewpoint),

- a **description of a current state** of the artefact.

Several researchers studied viewpoints representation [18], the most interesting for our approach are the following works:

a) In software development

Finkelstein & al. in [8] explain that in the software development the multiple perspective problem is central to both requirements expression and elicitation. Finkelstein formalises the viewpoint as a combination of the idea of "actor", "knowledge source", "role", "agent" and the idea of a "perspective" which an actor maintains. The viewpoint is so described by slots which characterise the viewpoint. This viewpoint is principally based on the actor (person and domain of competence). The slots are:

- a style -> What is the scheme representation used?

- an area -> What is the domain? (depend on the context)

- a specification -> the statements in the viewpoint’s style describing the domain (information dependant to the style and the domain)

- a work plan -> description of the process by which the specification can be built (information about specification)

- a work record -> description of how the specification developed, and its current status

This notion of viewpoint was implemented in the application Viewer[8], but the slots as they are used in Viewer are not efficient to express the experience and the knowledge of the designer. The slot "area" allows to know the domain, but not the situation in the design process (domain but more precisely task of design and how they are related to the other task).

So this characterization of viewpoint is interesting because it is simpler to know what to express in the viewpoint. It can guide the elicitation and help to find information about the domain, but it can be more accurate.

b) In conflict management

Viewpoints are used in the approaches based on negotiation and cooperation between participants in the development process. In [6], Easterbrook aims to facilitate identification and elaboration of viewpoints, and introduces a method of representing conflicting knowledge explicitly, using hierarchies of viewpoint. The principle is that a viewpoint is a description of the world from a particular angle, but there may exist conflicts in this description. So conflicting items are placed in different descendants of the previous viewpoint. So individually, each viewpoint remains self-consistent. This principle creates a hierarchy of viewpoint where conflicts are at the bottom of this hierarchy.

The organisation of viewpoints in a hierarchy is very interesting, because it permits to see exactly links between viewpoints and if some viewpoints are under the same theme or in the same level. The organisation into a hierarchy must be guided by an aim: for Easterbrook, it is the conflict detection and management, but his organisation into a hierarchy does not allow to do simple search of information. Viewpoints are not well-defined, so it is very difficult to find a precise information through those viewpoints. Each aim give a different way to do the hierarchy, and the advantage of Easterbrook's hierarchy is that his viewpoints are all self-consistent.

III. EXPLOITATION OF VIEWPOINT NOTION TO STRUCTURE DESIGN PROPOSITIONS

As we can see, viewpoints can provide an intelligent structuring and indexation of information. It can be more efficient to structure knowledge in a design process. Viewpoints allow to obtain an accessible and dynamic structure. So we use this notion in CE, to define a shared knowledge representation, in which viewpoints defined by slots permits a sufficient characterisation and help to:

- Define the good context to submit a design proposition.

- search information through different design propositions.

- add information in a design proposition.

- Define structured access to information for facilitating the knowledge sharing.

III. 1. Viewpoint and Conflict management

Emphasizing the characteristics of elements, which cause conflicts, in views helps to detect conflicts and to manage them. In order to guide conflict detection, we defined a typology of conflicts (Figure 2.), in which objects about which conflicts can appear and their nature are explicited. Conflict types expressed provide elements to structure shared knowledge. So, we present this typology below.

Conflicts in CE can appear from disagreements between some participants in the designers' group, according to their design propositions. Such conflicts arise from problems caused by strategies and propositions made by designers: 1) Strategic conflicts ensue from the inconsistency in methods and tools used by designers and in the allocation of tasks to each designer [7]. Divergence between the participants' responsibilities and failure in their cooperation [3] cause also strategic conflicts. 2) Conflicts about propositions made can appear from: a) misunderstanding of the participants' terminology and their points of view [15], b) unacceptable conditions under which a proposition is made and of the consequences by which a proposition constrains the design cycle. "Preconditions" problems can be revealed from preferences like needs [17], from the use of the same resources [7] and from differences in the requirements evaluation by the designers. Constraints imposed by the proposition and its potential interactions with other propositions can generate problems. Note also that the lack of quality of a proposition can cause its no acceptance. Figure 2. shows a typology of such conflicts.

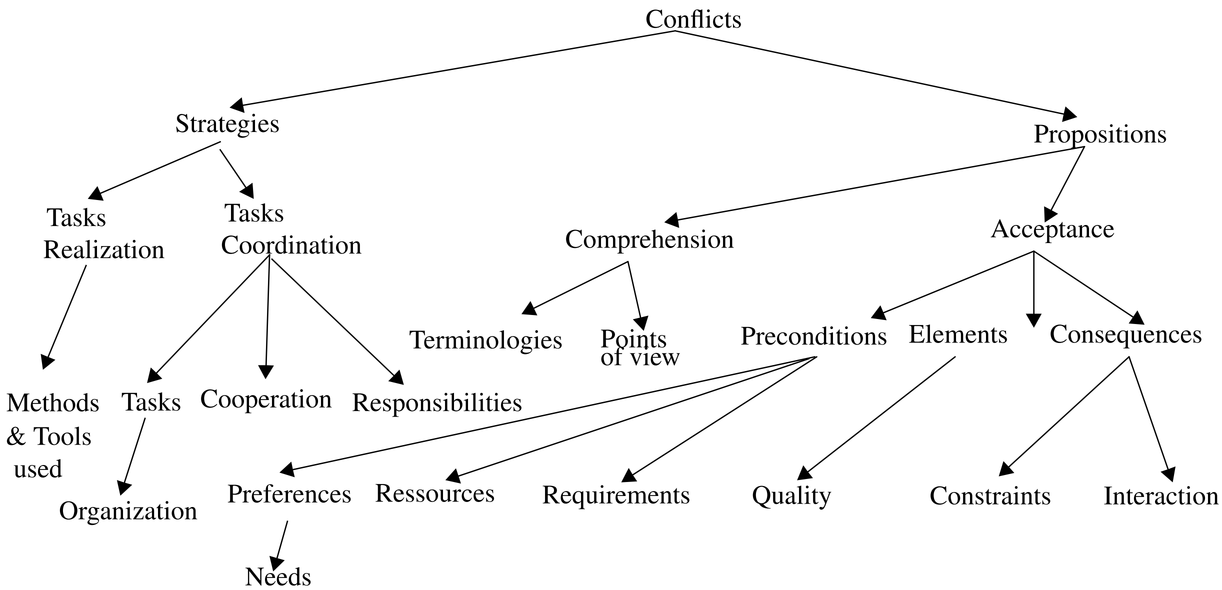

Figure 2. A Typology of Conflicts.

To facilitate the selection of methods, we use our conflict typology (Figure 2.) as index of the library of generic Conflict management methods, we defined. So, for each conflict type, several methods which can be used to solve this type of conflicts are associated (For more detail, see [12]).

We use the conflict types proposed in the typology, we defined (Figure 2.) as themes, to restructure propositions given by designers. This use of viewpoint through the typology of conflicts allows:

- to detect conflicts more easily by gathering different elements of propositions which can cause conflicts relative to a particular type.

- to be able to find, with the viewpoint, corresponding methods to resolve a conflict through the index of the generic conflict management library.

III. 1. 1. Our viewpoint structure

Our choice is guided by the fact that we aim to give help to compare propositions' interdependencies in order to detect their incompatibility. So, propositions are structured in order to have in the leaves of the structure all the proposition elements of the designers relative to a cause of conflicts (Figure 3.).

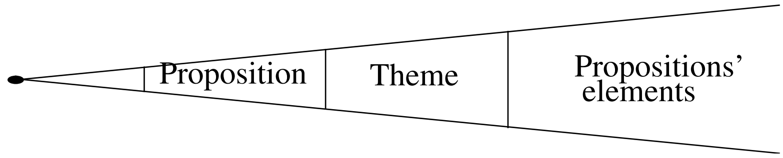

Figure 3. level of the viewpoint structure

So this type of representation (Figure 4.) with the viewpoint slot characterization allows to have a structuring adapted to our need (structuring and indexation for simple research and conflicts management between proposition elements of the designers). This structuring is dynamic (addition of theme in the second level).

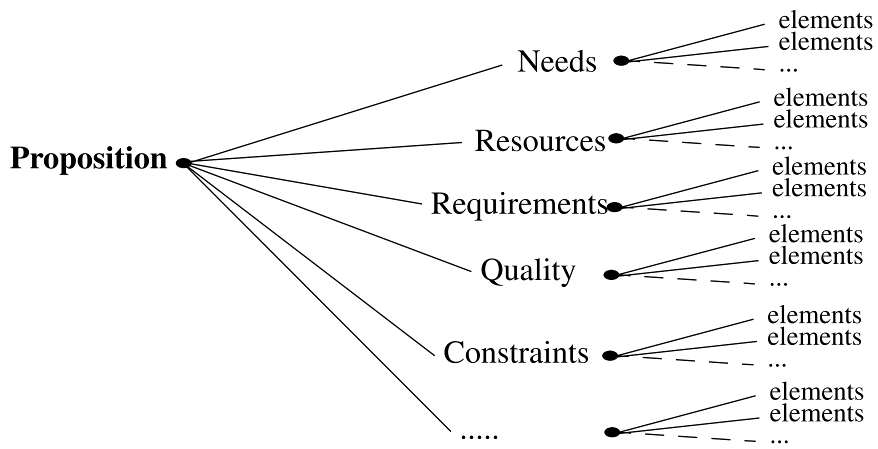

Figure 4. Restructuring design propositions with conflict types

IV. COMMUNICATION, REPRESENTATION AND EVALUATION OF PROPOSITIONS (CREOPS^2)

To handle some parts of the CE task, we propose a multi-agents architecture in the Communication, Representation and Evaluation of Propositions Support System (CREoPS^2). As we can see in (II. 1), individual propositions in CE must be communicated to the group of participants in order to evaluate their integration in the artifact. So, the representation of propositions in a single formalism facilitate the comparison of their parts and their evaluation [4]. We present in this part this single formalism and the exploitation of the viewpoint structure (Figure 4.) to evaluate the integration of design propositions.

IV. 1. A Communication Structure: a Proposition Representation

A proposition is represented by the type of elements needed to describe a design or to defend it. Each type of proposition description is associated with a set of design elements. Propositions are described with three general types: aspects, arguments and links.

- ***aspects***: they describe design elements in order to refine and to complete the description of the design statement. Aspects can be static (i.e. describe the product and the requirements) or dynamic (i.e. describe the design process) as in the design statement [3]. Those aspects can be versions of elements representing design statement. We call version of elements a modification or an addition of properties of a design elements.

- ***arguments***: As described in the CE model (I), the goal of arguments is to defend the proposition.

- ***links***: a design element (representing aspects or arguments) is linked with others. For example, a structural link (for example the link «A is sub-part of B», «C is composed of D,E,F»), a functional link (for example the link «A is equal to B», «C is greater than D»), a logical link (for example the link «A implies B»), or a composition of links. Thus, we can describe different types of links between design elements and for each link, we can have the inverse link. So, we view two kind of links, link to design elements defined in another step of design (which depend on future design decision) or already defined somewhere else.

Emphasis of elements which can be incompatible and cause potential conflicts form an excellent help to detect discordance between propositions. So we use the viewpoint structure (Figure 4.) as an evaluation structure to compare those elements.

IV. 2. An Evaluation Structure: a Structure based on Viewpoint Notion

Elements which can cause potential discordance between propositions are naturally compared in order to detect conflicts. Viewpoints in complex system modelling [6] is used to index and manage knowledge. We used a similar notion in our evaluation structure (Cf. Figure 5.) to propose a guide for a comparison design elements by indexing them in conflict types. So this indexation guides the comparison by emphasising and split elements relative to a potential type of conflict. For example, all resources (for example delay, cost...) required by the designers to perform their propositions are represented under the *Resources* conflict type. Then the incompatibility of resources required is more easily detected.

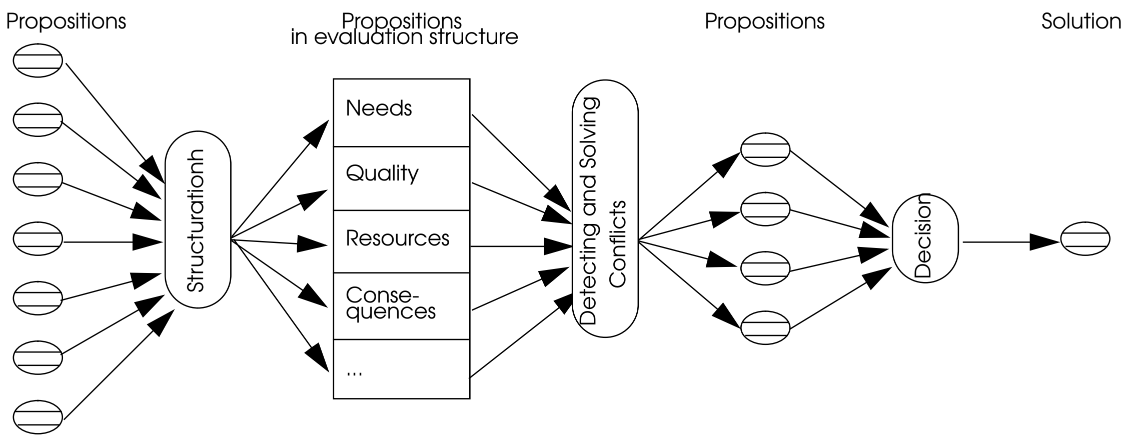

Figure 5. Use of Evaluation Structure

IV. 3. Support System to detect and solve conflicts

We focus in this part on the use of the evaluation structure (Figure 5.) in conflict management (detection and solving) in CREoPS[2]. In order to help conflict management, the architecture of CREoPS[2] is composed of three agent types (Figure 6.) and it is based on three data management system [4], in particular viewpoint structure (Figure 4.).

1) The *Representation* agent permits to designers to interact with the system in order to help to formulate and to communicate, via the Web on the Intranet, their proposition's elements.

2) The *Proposition Management* agent receives structured proposition's elements and it updates, if necessary, the artefact or the evaluation structure (Figure 4.). This agent is an automatic agent and designers do not interact with it.

3) The *Evaluation* agent helps designers to manage conflicts.

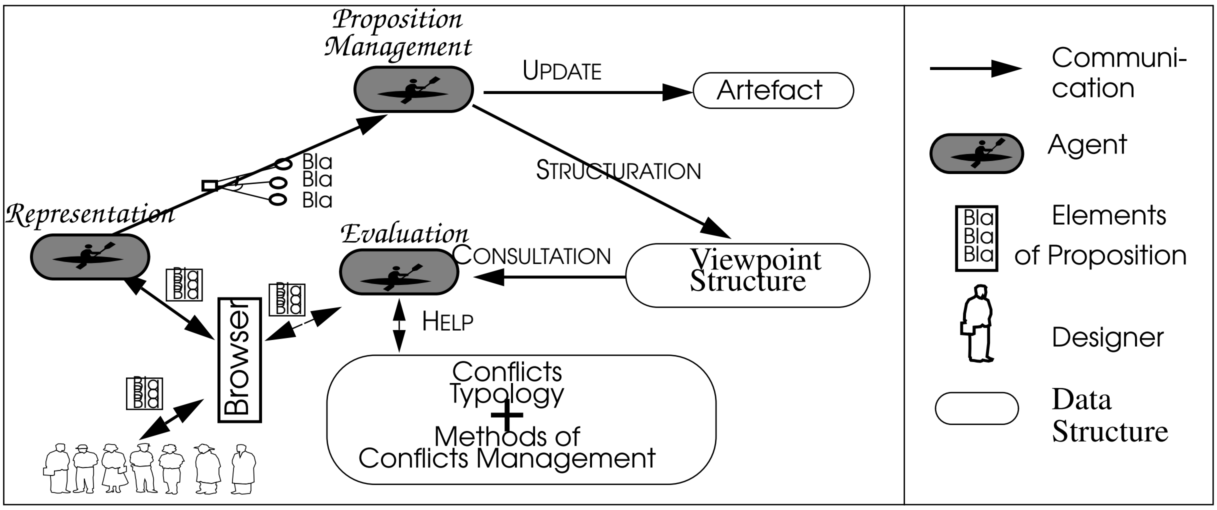

Figure 6. CREOPS^2 Architecture for Conflict Management

We suppose here that all designers have send their propositions via the *Representation* agent. The *Proposition Management* agent have structured all the elements of those propositions in our evaluation structure. Between those elements of propositions conflicts can appear.

Based on conflicts typology [12] and on the comparison of proposition elements in the viewpoint structure, the *Evaluation* agent helps to detect incompatibility between proposition elements and proposes a set of potential conflicts.

If there is conflict(s) in a viewpoint (the detection can be automatic or not) the *Evaluation* agent suggests to the designers method(s) (automatic(s) or not) to manage conflict(s) [12]. After this conflict management step, the designers use the *Representation* agent to communicate to the system modified proposition(s). This modified propositions can be evaluated again to manage new conflicts. Or, if a solution is chosen for the design task, the *Proposition Management* agent updates the artefact.

V. CONCLUSION

CE is a complex domain, in which individual knowledge is confronted with a common one. In fact, design propositions generated by each designer are communicated to all participants in order to integrate them in a shared representation (the artefact). Knowledge about participants' preferences, methods, etc. and about propositions' quality, constraints, interaction, etc. must be shared to manage conflicts between designers.

In CE, the use of viewpoint in the structured proposition of design shows how the viewpoint notion can provide real help in the extraction, treatment and consultation of shared design information. Our viewpoint description and multi-perspective management are more efficient to structure propositions in CE because of a more accurate characterization of viewpoint, which allow an intelligent indexation of design propositions.

The application in CREoPS^2 of this viewpoint description highlights contribution in conflicts management (detect and solve) and knowledge sharing between designers. Easterbrook [6] helps also to detect conflicts in a hierarchy of viewpoints, but the structuring of CREoPS^2 with viewpoints permits simple research of information to facilitate knowledge sharing between designers too. So, it is important to detect conflicts, but also to access easily to information in the structure.

We have described a specific application of viewpoint based on structuring proposition and conflict management with viewpoints. The use of viewpoints would be generalized to propose a generic help to distribute and define tasks relative to a design process in CE and to designate designers who participate to a group. Real applications in CE will help us to evaluate our architecture.

REFERENCES

[1] L. Acker, B. Porter, Extracting Viewpoints from Knowledge Bases, *in Proceedings of AAAI* 1994.

[2] A.H. Bond, A Computational Model for organizations of cooperating intelligent agents, *Proceedings of the Conference on Office Information Systems*, Cambridge April, 1990.

[3] F.M.T. Brazier, P. H.G. Van Langen, J. Treur, Modelling 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.

[4] C.Cointe, N. Matta, Architecture to handle Concurrent Engineering, 1*s International Engineering Design Debate*, Glasgow, September, 1996.

[5] *CommonKADS Library for expertise modelling Reusable problem solving components, Frontiers in Artificial Intelligence and Applications,* J. Breuker and W. Van de Velde (EDS), Amsterdam: IOS.Press 1994.

[6] S. Easterbrook, Domain Modelling with Hierarchies of alternative viewpoints, *in Proceedings of IEEE International Symposium on Requirements Engineering*, January 4-6, San Diego, California, 1993.

[7] 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 1993.

[8] A. Finkelstein, J.Kramer,B.Nuseibeh, L.Finkelstein, M.Goedicke, Viewpoints: A Framework for Integrating Multiple Perspectives in System Development, *International Journal on Software Engineering and Knowledge Engineering, Special Issue on Trends and Research Direction in Software Engineering Environments,* 2 (1):31-58, March 1992.

[9] D. Garlan, Views for Tools in Integrated Environments, *Proceedings of TOOLS'87*, pp. 313-343, 1987.

[10] M. Klein, Conflict management as part of an integrated exception handling approach, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* Vol.9, p.259-267, Cambridge University Press USA 1995.

[11] O. Marino Drews, *Raisonnement Classificatoire dans une représentation à objets multi-points de vue*, PhD Thesis, University of Joseph Fourier Grenoble 1, 4 October, 1993.

[12] 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.

[13] R. Moyse, Knowledge Negotiation implies Multiples Viewpoints, *in Proceedings of the 4th International Conference on AI and Education*, 24-26 May 1989, Amsterdam, Netherlands, pp 140-149.

[14] A. Newell, The Knowledge Level, *Artificial Intelligence Journal,* 19 (2), 1982.

[15] B. Ramesh, K. Sengupta, Managing Cognitive and Mixed-motive Conflicts in Concurrent Engineering, *Concurrent Engineering: Research and Applications* Vol.2, N.3, 1994, pp.223-236.

[16] M. Stefik, D.G. Bobrow, Object-Oriented Programming: Themes and Variations, *The AI Magazine*, vol 16, nº 4, pp 40-62, 1985.

[17] 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.

[18] J. Vanwelkenhuysen, *Cooperative Design,* Research Report of INRIA RR-2855, Sophia-Antipolis, 1996.