

Using problem solving models to design efficient cooperative knowledge-management systems based on formalization and traceability of argumentation

Myriam LEWKOWICZ, Manuel ZACKLAD

Laboratoire Tech-CICO
Université de Technologie de Troyes
12, rue Marie Curie BP 2060
10010 Troyes Cédex
France

myriam.lewkowicz@univ-troyes.fr

manuel.zacklad@univ-troyes.fr

Abstract. We present here a groupware (MEMO-Net) based on a model (DIPA), which uses and simplifies the concepts of Problem-Solving methods. This model comes from a review of Design Rationale formalisms that gave rise to the ABRICo formalism. MEMO-Net enables exchange structuring in order to improve dialog quality and reusability. MEMO-Net prefigures a new KM approach, named “cooperative KM” where we propose virtual work environments for groups that will consist in new “coordination mechanisms”.

1. Introduction

In accordance with Zacklad and Grundstein [17], we can classify knowledge management (KM) approaches in three complementary categories: top-down, bottom-up and cooperative. In the first one, models are used with experts to formalize their knowledge (MKSM for example). In the second one, huge corpus are memorized and formalized afterwards (text-mining methods). And in the third one, we consider that organizations’ critical knowledge comes within a collective competence that is not enough or badly formalized.

Groupwares generally use two kinds of models to structure interactions and to manage knowledge in organizations: some of these models allow a relationship standardization instead of others that focus on know-how standardization [18].

Among models based on relationship standardization, we can quote the famous Speech-Act theory from Winograd and Flores [15] and its derivative [6]. These models allow a more efficient management of contractual processes in firms. They fit in KM strategies focused on “commitment traceability”. On the other hand, models used for decisions’ traceability in the scope of know-how standardization are closer to Design Rationale (DR) researches. In these studies, models are decision oriented. We can quote QOC [9] or IBIS [1].

In a previous study [8], we came to the conclusion about the difficulty of applying some DR models in complex collective design situations. In this paper we propose to use problem-solving methods to organize the traceability of decision-making processes, particularly in design situations. Problem-solving methods are going to help the formalization of argumentation in collective problem solving situations.

2. CSCW and Knowledge Management

Our approach comes within the scope of cooperative approaches where one considers that crucial knowledge is held by groups. If we want to capitalize on this exchanged knowledge, we have to focus on existing communication processes in the firm. Two approaches are possible faced with this problem: Memorizing all the interactions and then building a thesaurus on the base of the corpus or playing a part during the communication process by a priori structuring information.

The second option is the one that we have chosen. This interest in *a priori* structuring of problem-solving processes in order to guarantee exploitation is not recent. In CSCW (Computer-Supported Cooperative Work) research, several authors [3], [4] have already expressed the wish to switch from a “object-centered paradigm to a “process-centered” paradigm. In the latter, designers’ interaction (that is to say questions, decisions and conversations that form the elaboration environment of the objects) would be memorized as well as objects and design process results. This paradigm is the one of DR researches, of which experimentations can be classified in two groups: *A posteriori* DR whose drawbacks are the need of a project memory leader during the project and meetings with all the people of the project after the project [7]. Or DR *following the current*, with tools such as QuestMap for example [10].

Our approach is near the second group although we have shown [8] that DR models are pertinent for some situations but are sometimes far from real design situations in which we construct step by step a solution instead of choosing or sorting several options. Our aim is to propose more realistic models from a cognitive point of view but not too far from concrete design situations which people are confronted with. In order to approach the cognitive dimension of reasoning we propose to use problem-solving method concepts from knowledge engineering.

3. Problem-Solving Models and Design Rationale

Actually, we could say that Design Rationale models have neglected the “information” phase of Simon’s decision-making process [13], and have only taken into account the solutions selection phase. Models from Artificial Intelligence do not have this fault. This link with problem-solving methods seems to us a natural evolution in our researches of more realistic Design Rationale models suited to the complexity of real projects.

We have then built a model called DIPA (from the French words Données, Interprétations, Propositions, Accord, meaning facts, interpretations, propositions, agreement) where problem-solving models replace decision-making processes, even complex ones.

The model comes in two forms, according to the situations that lead the actors to give priority to either analysis or synthesis processes (for example as in KADS methodology) [14].

This reference to problem-solving models allows the integration of an important knowledge category that was not taken into account in the two previous models, the “problem data”. In the DIPA model, the reasoning progresses in three major steps:

- a problem description step plus collecting of data, considered as symptoms in analysis situations and as needs in synthesis situations
- an abstraction step going from the collecting of problem data to their interpretation corresponding to a possible cause in analysis situations, and to a functionality in synthesis situations;
- an implementation step that going from an interpretation (cause or functionality) to the elaboration of a proposition that is a corrective action removing the symptom’s cause (analysis) or the means suitable for the expressed functionality (synthesis)

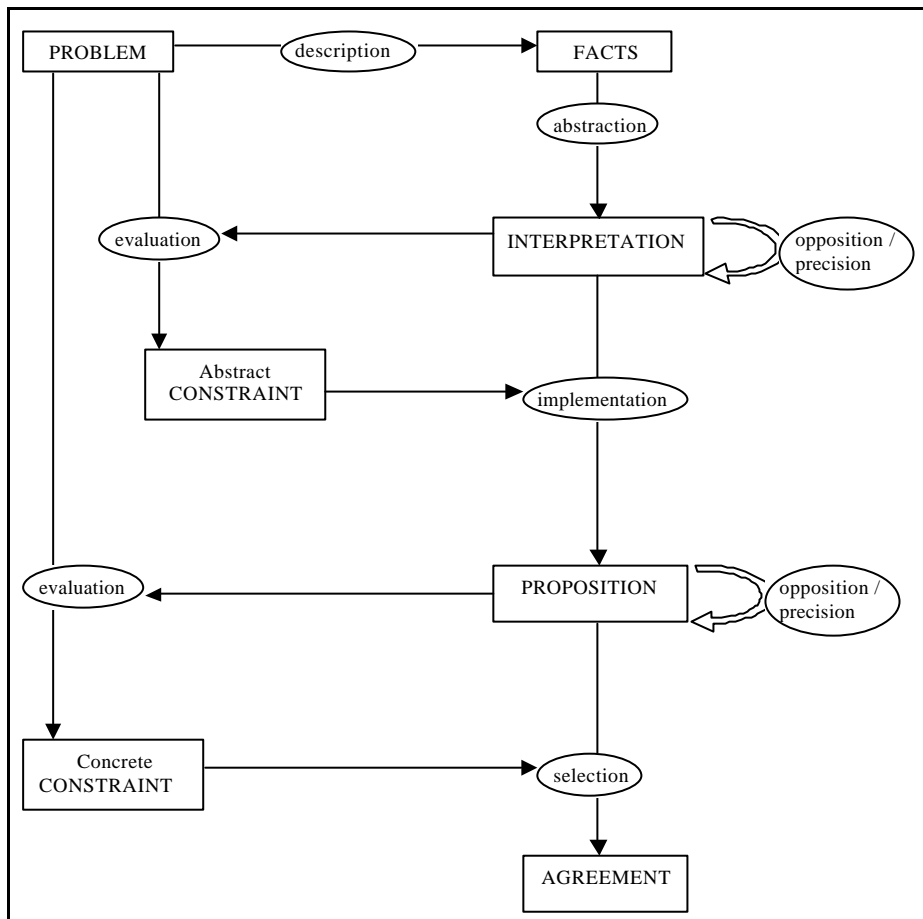
The fact that we had to present both analysis and synthesis models to designers teams may seem amazing. Actually, it might appear natural at first glance to propose only synthesis models and their variants (routine design, configuration...). But our practical experience of design meetings revealed us that analysis activities are frequent. For example, as soon as a prototype has been developed, its function analysis will give important information that will be reintroduced in the process of solution finding.

These observations are also in accordance with cognitive ergonomic psychology [5] results that teach us that design situations in the organizational sense in fact generate two distinct phases of activity: solution generation and then evaluation of these solutions. The first corresponds to synthesis problems in a KADS sense and is close to design models in this method. The second corresponds to analysis problems whose diagnosis models are well-known.

This idea of a unique model (figure 1) to represent the two types of activity is quite close to some interpretations of the heuristic classification of Clancey [2]. According to Zacklad and Fontaine [16], in both types of situation, there is both exploitation of knowledge from previous solved cases, and construction of an original solution. This solution is a proof or a justification in analysis case and a constraints-compatible approximate solution in synthesis cases.

Whereas, in the DIPA model, the abstraction and implementation inference steps are two symmetrical aspects of the same heuristic reasoning, applicable both in analysis and synthesis. In abstraction cases, for example, the point of view about any one system will change according to whether the symptoms or their causes are considered most important, or even the requirements of internal system functions.

The formalism used to describe DIPA was inspired by KADS [12], [14] but does not strictly follow the KADS conventions as to how to represent inference structures.



DIPA	DIPA synthesis	DIPA analysis
Problem	Goal	Malfunction
Fact	Requirement	Symptom
Interpretation	Functionality	Cause
Abstract constraint	Constraint	Constraint
Proposition	Means	Corrective action
Concrete constraint	Constraint	Constraint
Agreement	Choice	Choice

Fig.1: DIPA, a heuristic model of design reasoning for analysis and synthesis and its implementation for synthesis and analysis activities (table)

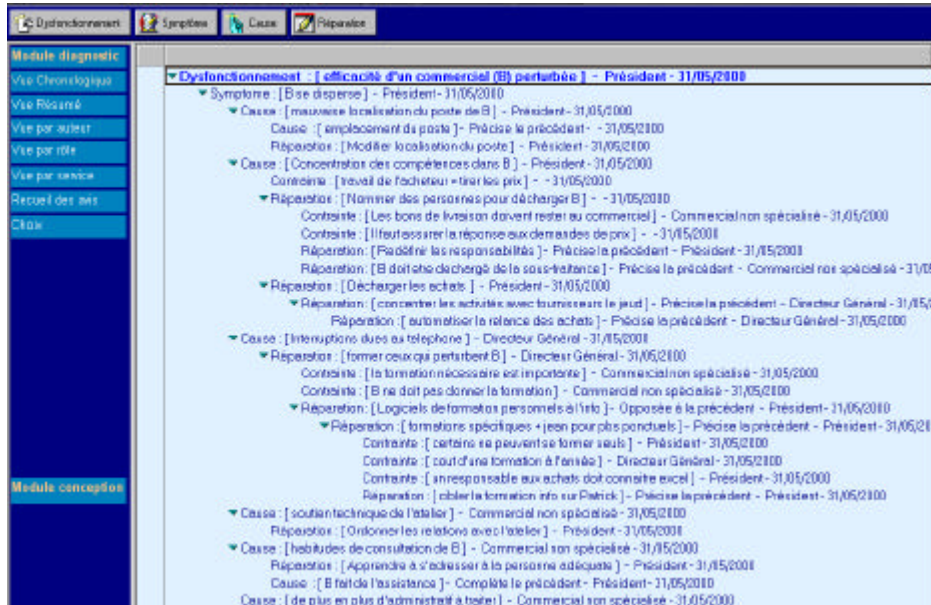


Fig.3 : A chronological view of a diagnosis phase

Experimentations

We have conducted a first experimentation of MEMO-Net with engineer students. They have used MEMO-Net synchronously to solve a familiar problem in their university: the choice of the courses each year. Some groups have solved the problem with a forum and other groups with MEMO-Net. All the students have filled in a questionnaire at the end of the experimentation. We try to prove that MEMO-Net improve collective cognitive performances. That means that the collective building of answers and solutions to a problem with MEMO-Net will be of better quality than with a forum.

We are going to compare the number of proposed solutions and the understanding of the reasoning by experts and to evaluate the quality of the solutions by showing them to experts.

We are also going to experiment soon MEMO-Net in crises meetings where quality experts debate on problems coming from Information System defaults. We will lead the problem solving process by using MEMO-Net.

Conclusion

The groupware MEMO-Net prefigures a new KM approach, based on a capture of exchanged arguments during activities following the current. In this approach, named "cooperative KM", we think that the best manner to implement a KM process is to propose virtual work environments that will consist in new "coordination

mechanisms” for these groups [11]. These virtual work environments have to be structured to offer a cooperation aid. This aid will incite actors to use these environments. They enable them to trace exchanges and to re-exploit more easily the knowledge when similar problems arise.

When one implements a “cooperative KM” step, an important stake is to identify problem-solving models which will enable the structuring of the work environment for the group. Until then, MEMO-Net used the DIPA model, which was a very general reasoning model (as generic as common KADS models). Future results would confirm that the use of general problem solving models is richer than the use of little-structured forum for the exploration of possible solutions and the re-use of these solutions.

Our future research will focus on the use of more specific models of problem-solving situations in groupware. In this way, we aim to use the synergy between traditional KM approaches and our approach. (cf. figure 4 and 5). In a traditional knowledge acquisition approach, the expert and the knowledge engineer co-construct a problem-solving model applied to the expertise domain (figure 4). Our aim is to re-use these domain models in tools like MEMO-Net to aid the cooperative problem solving between experts and/or novices.

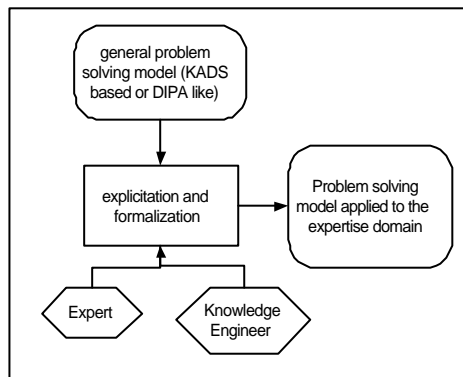


Fig. 4 : traditional KM model

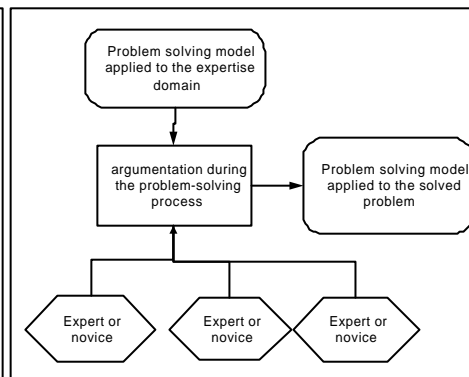


Fig. 5 : cooperative KM model

References

1. Burgess-Yakemovic, K.C. and Conklin, E.J. (1990). Report on a Development Project Use of an Issue-Based Information System, in Halasz, F. (Ed.) (1990). *CSCW 90: Proceedings of the Conference on Computer-Supported Cooperative Work*. Los Angeles, Oct. 7-10, 1990. Association for Computing Machinery.
2. Clancey, W. (1985). *Heuristic Classification*, *Artificial Intelligence Journal*, 27, pp. 289-350, 1985.
3. Conklin, E.J. (1993). Capturing Organizational Memory, in Baecker, R. M. (Ed.) *Readings in Groupware and Computer-Supported Cooperative Work*, Morgan Kaufmann Publishers, Inc.

4. Conklin, E.J. and Burgess-Yakemovic, K.C. (1996). A Process-Oriented Approach to Design Rationale, in Moran, T. P. and Carroll, J. M. (Ed.) *Design Rationale Concepts, Techniques and Use*, Lawrence Erlbaum Associates.
5. Darses, F. (1994). *Gestion des contraintes dans la résolution des problèmes de conception*. Thèse de doctorat, Spécialité Psychologie Cognitive. Paris, Université de Paris 8.
6. De Michelis, G. & Grasso, M.A. (1994); Situating conversations within the language/action perspective: the Milan Conversation Model; Proceedings of CSCW'94 Conference, Chapel Hill, North Carolina, October 22-26, pp. 89-100.
7. Karcenty, L. (1999), Capitaliser le contexte des décisions en conception : pourquoi et comment ?, in Zacklad, M., Grundstein, M. (Eds.) (1999). *Système d'Information pour la capitalisation des connaissances : tendances récentes et approches industrielles*, Hermès, à paraître.
8. Lewkowicz, M., Zacklad, M. (1998). A formalism for the rationalization of decision-making processes in complex collective design situations, *Proceedings of COOP*. Cannes, may 1998.
9. MacLean, A., Young, R.M., Bellotti, V.M.E., Moran P. (1996). *Questions, Options and Criteria : Elements of Design Space Analysis*, in Moran, T. P. and Carroll, J. M. (Ed.) *Design Rationale Concepts, Techniques and Use*, Lawrence Erlbaum Associates.
10. *QuestMap™*, Group Decision Support Systems, 1000 Thomas Jefferson St. Suite 100, Washington, DC 20007, 1999.
11. Schmidt, K., Simone, C. (1996). Coordination mechanisms : Towards a conceptual foundation of CSCW systems design, *JCSCW*, vol. 5, no. 2-3.
12. Schreiber, G., Wielinga, B. (1993). *Model Construction*, in Schreiber, G., Wielinga, B., Breuker, J. (Eds.) *KADS a principled approach to knowledge-based system development*, Academic Press.
13. Simon, H.A., (1947) *Administrative Behavior*, trad. Fçse (1983) *Administration et processus de décision*, Economica.
14. Wielinga, B., Schreiber, G., Breuker, J. (1993). *Modelling Expertise*, in Schreiber, G., Wielinga, B., Breuker, J. (Eds.) *KADS a principled approach to knowledge-based system development*, Academic Press.
15. Winograd, T., Flores, F. (1986) *Understanding Computers and Cognition*, Addison-Wesley, USA.
16. Zacklad, M., Fontaine, D. (1996). *L'acquisition des connaissances classificatoires pour les systèmes à bases de connaissances*, in Aussenac-Gilles, N., Laublet, P., Reynaud, C. (Eds.) *Acquisition et Ingénierie des Connaissances*, Cepadues-Editions.
17. Zacklad, M., Grundstein, M. (Eds.). *Système d'Information pour la capitalisation des connaissances : tendances récentes et approches industrielles*, Hermès, 1999.
18. Zacklad, M. (2000) *La théorie des transactions intellectuelles : une approche gestionnaire et cognitive pour le traitement du COS*, *Intellectica*, numéro spécial sur le COS, à paraître.