Scoop : A complete methodology for cooperative systems modeling and analysis

David Saint-Voirin, Christophe Lang, Hervé Guyennet
Laboratoire d'Informatique
de l'Université de Franche-Comté
CNRS FRE 2661
16, Route de Gray,
25030 BESANCON Cedex FRANCE
Email: [saint-voirin,lang,guyennet]@lifc.univ-fcomte.fr

Noureddine Zerhouni
Laboratoire d'automatique
de Besanon
24, Rue Alain Savary,
25000 BESANCON FRANCE
Email: noureddine.zerhouni@ens2m.fr

***Abstract*— In this paper, we present our methodology proposal for cooperative remote systems modeling. Its aim is to develop models of existing or planned cooperative systems. These models are used to specify systems or to create dynamic working simulations. Comparative performances are deduced from the latter. The associated meta-model proposal is based, among other things, on the use of multi-agent systems, Petri nets and stochastics Petri nets. After having described the whole methodology concepts, we show a set of general results extracted from simulations.**

I. Introduction

Cooperation is a complex activity known since the beginning of humanity. It allows other activities that would not have been possible without cooperation. It is then known as an efficiency factor for many activities.

At present, the industrial world is becoming more and more enhanced and developed by the use of new technologies. Cooperative systems take advantage of this situation. New technologies of communication, mobile terminals and data access modes, for instance, improve cooperation capabilities. But, the other side of the coin is that the complexity of the distributed system which is the support for cooperation is added to the complexity of cooperation activity.

It is then difficult to quantify the profit generated by the use of one technology or another, or the use of a particular organization in a given cooperative system. There is a clear need for an efficient cooperative system meta-model.

This paper presents our methodology, based on a meta-model of cooperative remote systems whose aim is to develop models of existing or planned cooperative systems. These models are used to specify systems or to create dynamic working simulations. Comparative performances are derived from the latter using our criteria.

Several research projects have been conducted in computer and information sciences (computer-supported cooperative work, group decision support systems, knowledge engineering, human-computer interaction, distributed artificial intelligence and multi-agent systems, etc.) and in social sciences (organizational and management sciences, sociology, psychology, ergonomics, linguistics, etc.). After defining general cooperation concepts found in the literature, we propose a synthesis of existing work in cooperation modeling and cooperative systems modeling. Cooperation use in multi-agent systems is also described in this part of the paper.

Our methodology is then described step by step. The associated meta-model is also presented in this part.

The last part shows the results obtained using stochastics Petri nets and multi-agent simulations.

II. Models, cooperation and cooperative systems

A. Definitions

Cooperation is a complex human activity. The studied literature gives us different definitions revolving around cooperation :

- Cooperation requires communication, with a view to exchange information, inform the other of its intentions, interpret and understand the partner interactions, especially for current and future actions [5].
- A system is considered as a cooperative one when it considers all situations and interaction formalisms between partners. It needs to define objectives and goals to be reached by the group, to break down and allocate tasks so as to reduce conflicts, disagreements and redundancy as much as possible, and to execute tasks with synchronization and convergence of goals to be reached [12].
- When cooperation occurs, we can distinguish three very distinct conceptual universes [5] : the data world, the users world (which has access and operates data world components), and the organization world (which structures the previous two).
- Another classification is based on functional group definitions. Three functional groups are defined : production space (designating results from a group activity), coordination space and communication space [10].

B. Cooperation models

Cooperation models have been proposed using different viewpoints : activity, communications and interaction among others.

A cooperation activity model is proposed in [5]. It is based on oriented graphs representing cooperative interactions. As a

formal cooperation model, it is interesting; however, it does not allow dynamic working simulations of cooperative systems. It could better be used in cooperation mode analysis on an existing simulation.

Collective cooperative behaviours are studied in [14] [4]. In this work, six primitive collective behaviours for cooperation are defined : equivalence, transfer, specialization, redundancy, complementarity and concurrency. This study could also be used to detect specific cooperation behaviours in an existing simulation, and then draw conclusions on its efficiency.

C. Cooperative systems modeling

The border between cooperation models and cooperative systems modeling is difficult to determine. Cooperation models are often used in cooperative systems modeling. However, several points of view are considered in cooperation system modeling (decision-making process, problem to solve modeling, physical system modeling...).

For example, reasearch reported in [2] and [9] shows a Petri net modeling based on processes and activities. The system is modeled in view of activities so Petri nets represent states like decision, evaluation, negotiation... Cooperative process is known and follows a specific pattern of decision making. Colored Petri nets used in this work allow information transport and information modification on tokens. In this sense, transitions are fired only when two people generate the same type of tokens (or proposition).

D. Multi-agent systems and cooperation representation

Cooperation in multi-agent systems is an important concept. As a rule, agents interact with each other in multi-agent systems. They have the ability to communicate. They can understand the will of other agents and adapt their own behaviour to improve collective results [23] [15].

Many multi-agent systems represent existing systems composed of intelligent entities. Cooperation appears when the system is dynamically working using simulation tools. Cooperation is not explicitly represented but appears from simulation.

The principle is to represent each entity of the existing system as an agent having its own behaviour and knowledge. These agents have communication abilities and may react specifically to external constraints (environment changes, orders from another agent...).

III. *S*COOP : A COMPLETE METHODOLOGY FOR COOPERATIVE SYSTEMS MODELING

The lack of a global cooperative systems modeling approach leads us to the proposal of a complete methodology that allows us to draw models of different cooperative systems aspects. We will first describe the used hypothesis. Then, we will describe the *S*coop methodolgy step by step, linking each step with his meta-model proposal.

A. Hypothesis

In this part, we describe our modeling hypothesis, which define the limits of our representation. The whole *S*coop methodology is based on our cooperative systems definition.

We defined in [20] and [21] cooperation in a cooperative system as a complex mechanism involving the following [1]:

- *Knowledge possessed by members*,
- *Specific behaviour of the members*. Human behaviour, during cooperation represents an important part of cooperation activity. Existing multi-agent work allows behaviour specification for each individual [23]. By extension, we can define individual behaviour and observe resulting collective behaviour by simulation.
- *A certain organization of the member group*. Organization of cooperating groups has been studied considerably by different authors. This reasearch is often oriented towards group management algorithms or protocol development [6], [3], [10].
- *Communication means between members* [8]. This concerns shared data access and communication between human beings. Concerning shared data, several problems are dealt with in the literature such as data heterogeneity, shared data modes or data availability. The same can be said for communication with regards to heterogeneity and defining communication tools [3]. Much of this work has been reused to integrate multimedia flows in cooperative applications [24].

Thus, the following five categories have to be represented :

- *Cooperating member behaviour:*
  This category implies the modeling of the individual behaviours. It includes social behaviour and member reasoning. Behaviours have an influence on group organization, on fluctuations in communication and on access to shared data. Modeling behaviour includes modeling the notion of role and goals.
- *Knowledge of cooperating members:*
  Knowledge of a cooperating member is of different types : knowledge of his capacities (*skills*), knowledge of his *environment* (including his knowledge and vision of the problem and his knowledge of cooperation rules) and knowledge of his *role and goals*.
- *Organization of the cooperating member group:*
  The cooperating member group is organized on several levels : *geographical organization*, *hierarchical organization* and *legal organization*. Organization intervenes in shared data management mode, and on communication and shared data access authorizations. The static and dynamic aspect of organization has to be considered.
- *Interactions between group members:*
  These interactions can be of several types : *synchronous or asynchronous*. It may involve two or more people : *diffusion to several people*, *conference*, or *peer to peer conversations* (we suppose that the communication media

[1] In this definition, a member can be a person or equipment

is exclusive for this kind of communication). Mutual exclusion and network channel availability constraints have to be represented.
- *Shared data usable by participants:*
  On shared data, we have mutual exclusion constraints preventing several people from modifying the same data at the same time. We also find constraints on network channel availability. Access to shared data may be of 3 types : visualisation, object adding, object modifying or deleting (which represent the same constaints).

B. Scoop methodology

The *S*coop[2] methodology presents 4 main phases (see figure 1). These 4 phases are linked to the software engineering classification [18].

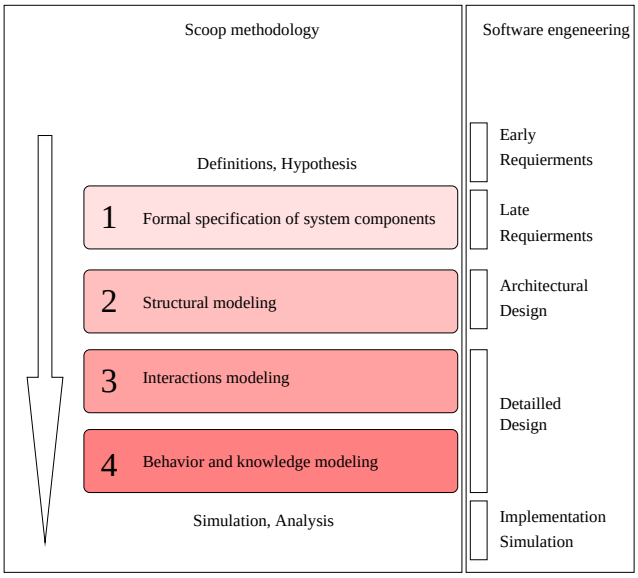

Fig. 1. The 4 main phases of *S*coop methodology

1) Formal specification of system components:

The first step of *S*coop methodology is the formal specification of system components. It uses a 5-uple :

$$Scoop_{name\_of\_the\_system} = \{P, S, G, H, L\} \quad (1)$$

Where :
- *P* is the list of human members of the system
- *S* is the list of equipment members of the system
- *G* is the list of geographical places used in the system
- *H* is the number of hierarchical levels used in the system
- *L* is the level of legal rules used in the system (0 is a system without any rules and 100 is a system where every action is ruled)

Each human member (P) is described by another 5-uple :

$$P_i = \{Beh, Know, G, H, Ch\} \quad (2)$$

Where :
- *Beh* is the class describing the behaviour of the member
- *Know* is the class describing the knowledge of the member
- *G* is the geographical place where the member is (taken from the previous list)
- *H* is the hierarchical place of the member (taken from the previous list)
- *Ch* is the work load level of the member

Each equipment member (S) is described by a 2-uple :

$$S_i = \{App, G\} \quad (3)$$

Where :
- *App* is the class describing the working mode of the member
- *G* is the geographical place where the member is (taken from the previous list) located.

This specification allows describing cooperative systems at a high level. However, a graphical representation of the cooperative system would help a lot at the specification time. This point is covered by the next step of the *S*coop methodology.

2) Structural modeling:

The second point of *S*coop methodology helps designer to draw structural models of the system. These models are easy to read and their graphical aspect simplifiy comprehension of the system.

The structural modeling we propose is based on a concrete nomenclature of members and interactions. This nomenclature is described according to the different aspects previously identified.

Human members are represented using a square containing basic information of the P_i 5-uple. Equipment members are represented using a circle containing the basic information of the S_i 2-uple.

Interactions are represented using 2 specific nomenclatures. The communication nomenclature (see figure 2) and the shared data access nomenclature (see figure 3).

In both interaction nomenclatures, the square symbol means that mutual exclusion is required for this type of interaction. The number of arrows is related to the number of members involved in the interaction.

It is necessary to draw several structural models to represent dynamic aspects. For instance, the initial state and the cooperative state of the system may be represented.

Figure 4 shows an example of the structural model for a cooperative e-maintenance system (the prototype of the e-maintenance system we developped in the european PROTEUS project [25], [19]). This model is developped in cooperative state of the system.

[2]Accronyme for : Cooperative systems Simulation

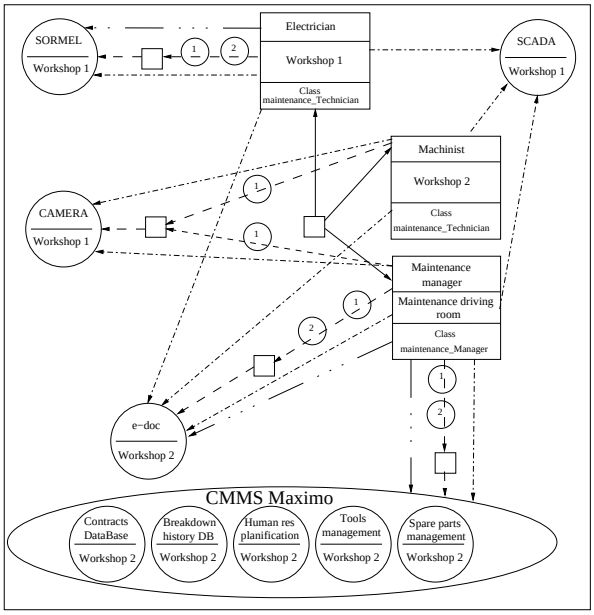

Fig. 4. Example of structural model

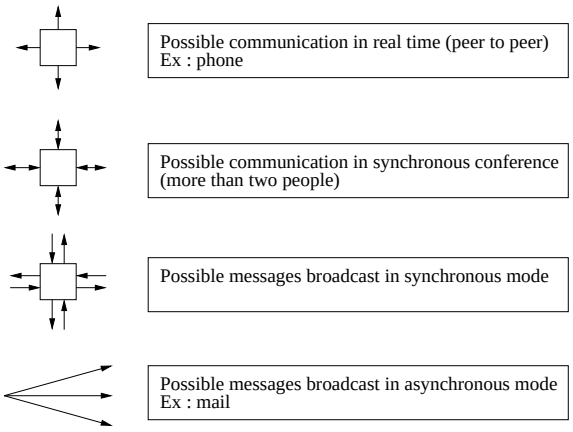

Fig. 2. Representation of the communication possibilities

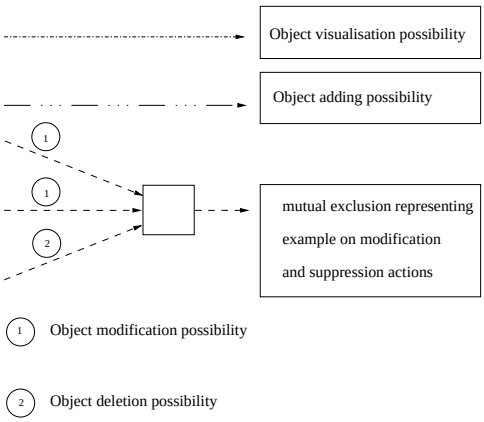

Fig. 3. Representation of data access possibilities

This structural model shows a synchronous peer to peer communication possibility (requiring mutual exclusion) between 3 members (electrician, machinist and maintenance manager). It also shows shared data access possibilities for the access to the web camera control or e-documentation reading.

The second step of *S*coop methodology helps describe the global structure of the system. It is very useful for specification. However, we cannot verify or simulate anything with this representation. That is why we developped associated Petri nets for the interactions.

3) Interactions modeling:

Each element of the interaction nomenclatures (figures 2 and 3) is then linked to a sub-method, allowing the creation of Petri nets representing interaction operating mode. Mutual exclusion is then represented. The sub-method uses number of members linked to the interaction mean, and the type of interaction mean. Conference and synchronous broadcast interaction means are similar cases in our basis hypothesis. They are modeled using the same Petri net.

For example, in the previously presented structural model (figure 4), the interaction operating mode between electrician, machinist and maintenance manager is detailled by the petri net shown on figure 5.

Petri nets created are used to verify livingness properties, to find deadlocks and conflicts in the interaction. We also developped an analysis based on stochastic simulations of these Petri nets (results are presented in section IV).

However, interaction study is just a part of cooperation aspects in cooperative systems. Human cognitive aspects have to be represented. To study this particular point, we choose to

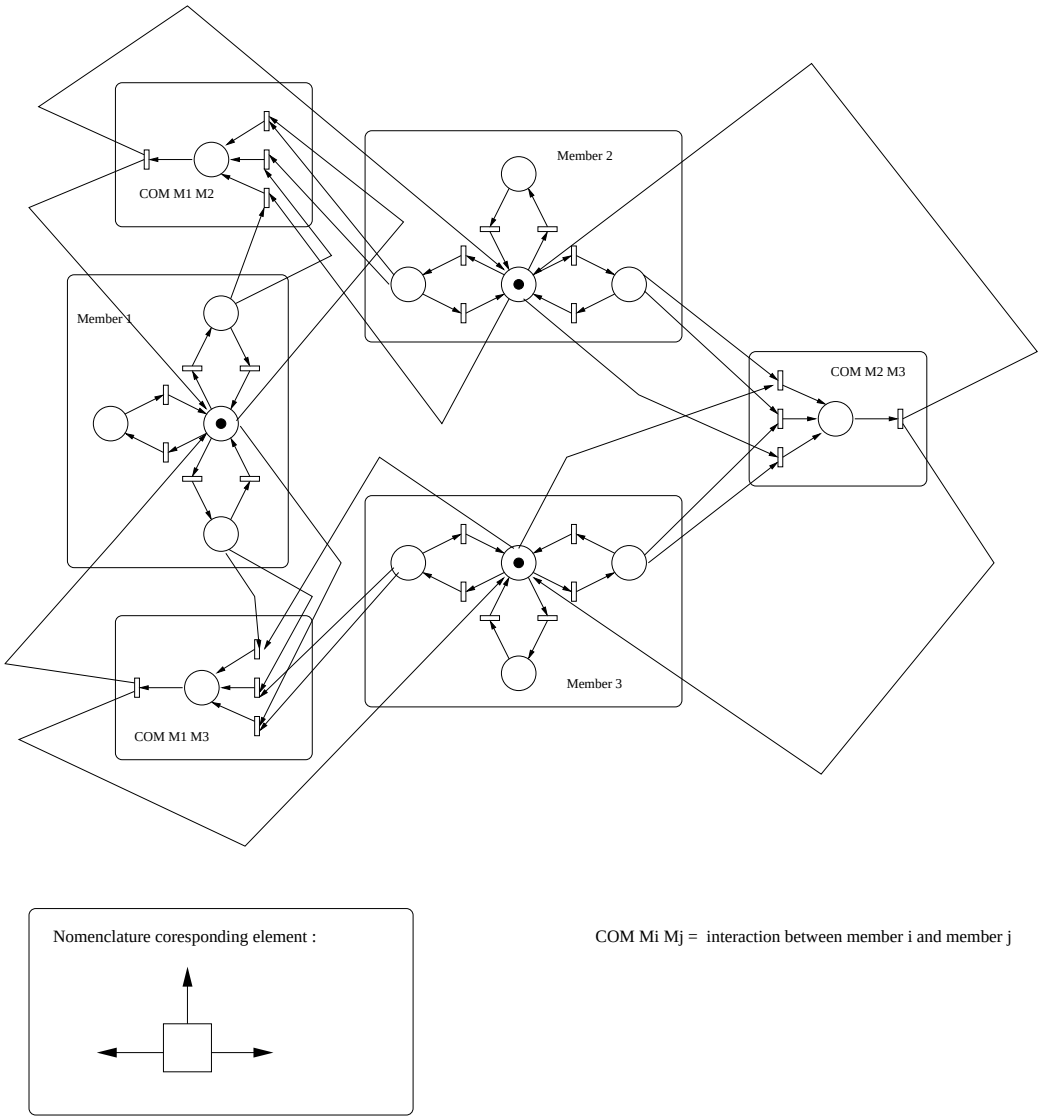

Fig. 5. Interaction operating mode for 3 members interacting in peer to peer synchronous mode (ex : phone)

create a multi-agent simulator of cooperative systems.

4) Knowledge and behaviour modeling:

We propose to define each member of cooperation as an agent. Multi-agent systems allow us to propose a good representation of human entities because of their artificial intelligence abilities. Multi-agent systems require knowledge and behaviour inherent to members. That is why we use behaviour and knowledge description formalisms.

We used a simple XML formalism based on a DTD allowing us to describe simple knowledge classified in the 3 identified parts (see part III-A) : skills, environnement and role and goals. This concept may be extended and more specific tags defined.

It is rather easy to integrate human will or desire into agents. Each agent has a goal, and in a well-managed cooperative system, these goals converge to a common goal. Agents may behave in cooperation because of their communication abilities and their initiating capacity [23]. We used an existing behaviour formalism, well suited to describe agents : PLOOM-UNITY. This one allows behaviour description with notions of goals and role.

UNITY [11] is an object typed formalism which allows agent description. We used PLOOM [7] to extend UNITY. In particular, we can use inheritance and functions in PLOOM UNITY formalism. This was not possible with UNITY formalism.

Using basic knowledge and behaviour, we developped a standard multi-agent simulator using JAVA with MADKit library[3]. This simulator allows analysis of human aspects in cooperative systems.

[3]www.madkit.org

IV. General analysis

Based on our modeling methodology proposal, we developped a general analysis of cooperative systems in two main parts :

- Interactions analysis based on Petri nets
- Cognitive human aspects based on multi-agent simulator

A. Interactions

We studied 8 typical interaction models shown on figure 6. These models have been choosen for their ability to answer the basic needs of cooperative systems. Indeed, few interaction systems involve more than 4 people. We create the associeted Petri nets for each of these 8 models.

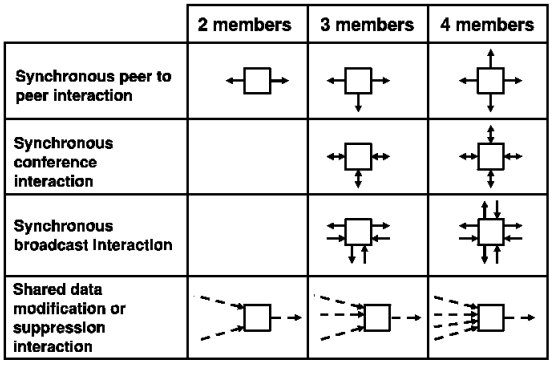

Fig. 6. Standard interaction models chosen for study

We formally analysed these 8 Petri nets models, proving that there is no deadlock. The 8 models are live. Structural conflicts that may be detected draw our attention to the points to be managed. For instance, a conflict may appear when 2 members want to acces the data at the exact same time. In this case, an algorithm or a human manager would have to choose the member which would be allowed to access data using some criteria. This underline aspects to be managed while interaction is running.

We realised a set of experiments, using stochastic parameters on these Petri nets. We realised these experiments with STPNPlay[4]. We present in this paper the results we obtained when using differents interaction durations (from 1 to 10 time units, considered as minutes to obtain readable results), for the following communication means :

- Synchronous peer to peer interaction (ex : phone)
- Synchronous conference and synchronous broadcast (same modeling)
- Shared data modification or suppression (same modeling)

We studied the effect of this duration variation on the cooperative system efficiency. This criterion is defined as the time the user takes to use the system without obtaining any interaction. It is a “lost time criterion” which is better when it goes lower.

[4]http://dce.felk.cvut.cz/capekj/StpnPlay/

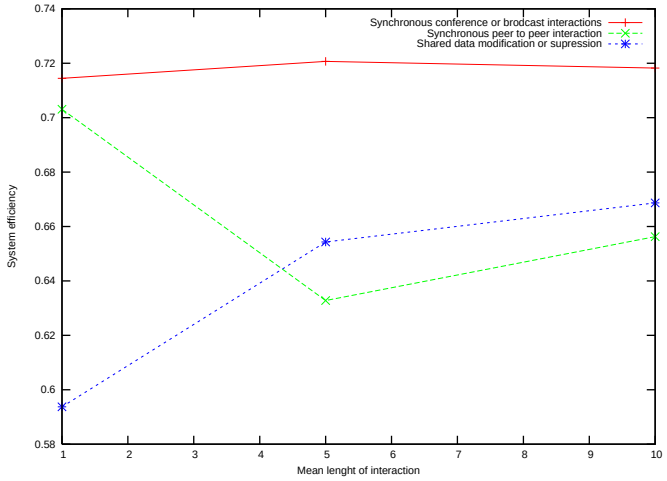

Fig. 7. Effect of interaction duration variation on cooperative system efficiency

Results are presented on the figure 7. Our modeling shows synchronous conference or broadcast systems efficiency is fewly affected by the mean duration of interaction. At the opposite, a shared data modification or deletion system is loosing its efficiency when mean interaction duration is growing. In a last point, the results of synchronous peer to peer interaction systems shows that a short interaction duration (1 minute) gives bad results, a long interaction duration (10 minutes) gives bad results, whereas a intermediate interaction duration (5 minutes) gives the best results.

B. Cognitive human aspects

Cognitive human aspects have been studied throughout our cooperative system multi-agent simulator. We studied several parameters as the multi-agent simulation is rather versatile : number of members that constitute the cooperating group, number of available members to constitute the cooperating group, vision quality of the cooperating members, type of members behaviour, members knowledge, type of management group algorithm...

Criteria we used to present results are :

- The mean time to solve a problem
- The mean number of exchanged messages
- The percentage of messages that get lower the group trust in it solution

Results of this study shows that the number of available members to constitute the cooperating group is a very important parameter compared to the number of members that constitute the cooperating group which influence fewly on the cooperation quality (see figures 8 and 9). This is caused by the “knowledge disponibility” provided by a great number of available members.

V. Conclusion and prospects

This paper proposes a general methodology and a metamodel which allows us to specify and to define cooperative remote systems.

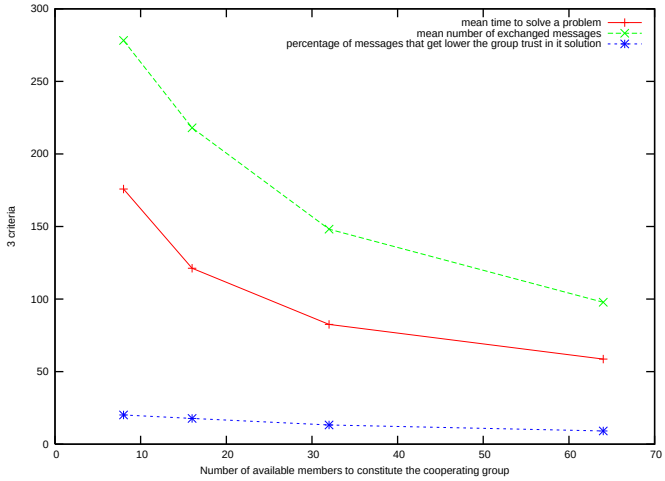

Fig. 8. number of available members to constitute the cooperating group effect on the 3 criteria

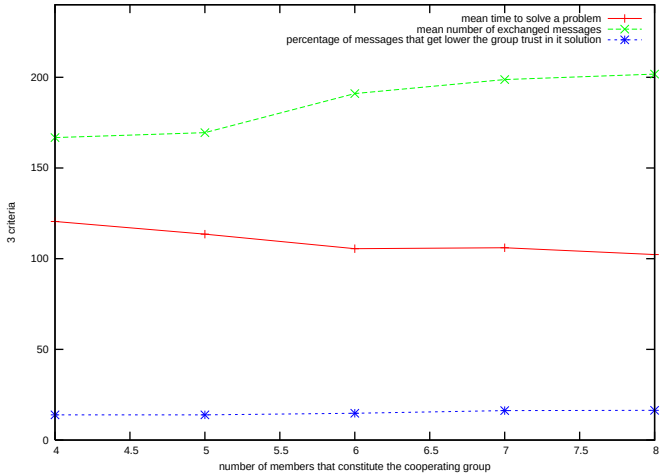

Fig. 9. number of members that constitute the cooperating group effect on the 3 criteria

After having presented the existing work both in cooperation and cooperative systems, we developed our own analysis of cooperative systems, leading us to the description of a meta-model.

This meta-model is based on a concrete and unified nomenclature of people and systems and also of their communication and resource access possibilities. It can therefore be used to draw the structure of a given system with a flexible level of description.

For each communication and resource access possibility type, we represent mutual exclusion for connexion by Petri nets. Communication and shared data access deadlock or starvation are then detected on Petri nets without simulation.

XML knowledge file definition and PLOOM - UNITY behaviour specification contribute to implementing a dynamic working simulation using multi-agent architecture. Multi-agents have been chosen for their ability to represent human behaviour, integrating role, belief and desire among other features.

Based on Petri nets and multi-agent simulator, we propose two ways for analysing cooperative systems :

- The first uses stochastic Petri nets to represent individual behaviour with probabilistic rules of transition.
- The second method of analysis is to compute a multi-agent system using the meta-model specifications. This method is more realistic because of the artificial intelligence aspect.

In a new project, XML knowledge representation will allow us to develop an ontology for agent communication and data requests. ACL (Agent Communication Language) research will assist us in this task. We then hope to be able to draw qualitative results based on the semantic content of messages exchanged by agents.

A semi-automatic generation of models would be an important step forward. It would simplify the computer model creation of existing or planned cooperative systems.

Representation of the different cooperating members' mobility is an important contribution to new cooperative system modelling. Our meta-model will have to take mobility representation into account by adding mobility behaviour and position knowledge.

Finally, adaptive agents could be used in our representation to improve simulation realism by allowing agents to acquire new abilities.

REFERENCES

[1] Abrilian, Buisine, Rendu, and Martin. Specifying cooperation between modalities in lifelike animated agents. In *Working notes of the international workshop of lifelike animated agents : tools, functions and applications, Held in conjunction with the 7th Pacific Rim International Conference on Artificial Intelligence (PRICAI'02)*, pages 3–8, Tokyo, Japan, August 19 2002.

[2] Aoumeur and Saake. A component-based petri net model for specifying and validating cooperative information systems. *Data and Knowledge Engineering Journal*, 44(2):143–187, August 2002.

[3] Borghoff and Sclichter. *Computer-supported cooperative work, introduction to distributed applications*. Springer, 2000.

[4] Buisine and Martin. Design principles for cooperation between modalities in bi directional multimodal interfaces. In *CHI'2003 workshop on principles for multimodal user interface design*, Florida, 5-10 April 2003.

[5] Diaz. A logical model of cooperation. Technical Report 92016, LAAS, January 1992.

[6] Drira, Diaz, Villemur, Jmaiel, Ben Hamadou, and Hadj Kacem. Cooperative systems for information sharing and exchange. In *IEEE 10th International workshops on enabling technologies : infrastructure for collaborative enterprises WETICE'2001*, pages 313–314, Cambridge USA, 20-22 June 2001.

[7] Ferber. Conception et programmation par objets. *Technologies de pointe - informatique, Hermes,*, 1990.

[8] Fisher. Communication requirements for cooperative problem solving systems. *Information systems*, 15(1):21–36, 1989.

[9] Frausto, Camargo, and Ramos. An application of an expressive coloured petri nets modeling methodology to a business to business environmen. In *MOCA'01 (Workshop on Modelling of Objects, Components and Agents) / Daniel Moldt (Ed.)*, pages 37–54. DAIMI PB-553, Aarhus University, August 2001.

[10] Greenberg. *Real time distributed collaboration in Encyclopedia of distributed computing*. Kluwer academic, 2002.

[11] Gruia-Catalin, McCann, Plun, and Jerome. Mobile UNITY: Reasoning and specification in mobile computing. *ACM Transactions On Software Engineering And Methodology*, 6(3):250–282, 1997.

[12] Guyennet and Lapayre. The group approch in distributed system. *Journal of parallel and distributed computing practices PDCP, Nova science publisher*, 2 (3):285–297, 1999.

[13] Karsenty. Cooperative work : the role of explanation in creating a shared problem representation. *Le travail humain*, 63(4):289–309, 2000.

[14] Martin. six primitive types of cooperation for observing, evaluating and specifying cooperations. In *AAAI Symposium on Psychological Models of Communication in Collaborative Systems*, Sea Crest Conference Center on Cape Cod, North Falmouth, Massachusetts, USA, November 1999.

[15] Odell, Van Dyke Parunak, and Bauer. Extending uml for agents. In *AOIS Workshop at AAAI 2000*, 2000.

[16] OMG. Document formal/02-04-03 meta object facility (mof) specification v1.4. Technical report, OMG, April 2002.

[17] Rodriguez Peralta, Villemur, and Drira. An xml on-line session model based on graphs for synchronous cooperative groups. In *International conference on parallel and distributed processing techniques and applications PDPTA'2001*, pages 1257–1263, 25-28 June 2001.

[18] Perry and Wolf. Foundations for the study of software architecture, 1992.

[19] Rebeuf, Blanc, Charpillet, Cheve, Dutech, Lang, Pélissier, and Thomesse. Proteus, des web services pour les systèmes de maintenance. In *NOTERE 2004 - NOuvelles TEchnologies de la REpartition*, pages 163–178, Saidia, Maroc, June 2004.

[20] Saint-Voirin, Lang, and Zerhouni. Distributed cooperation modeling for maintenance using petri nets and multi-agents systems. In *International Symposium on Computational Intelligence in Robotics and Automation, CIRA'03*, pages 366–371, Kobe Portopia Hotel, Kobe, Japan, July 16-20 2003. IEEE.

[21] Saint-Voirin, Lang, and Zerhouni. Distributed cooperative systems metamodel for maintenance using petri nets and multi-agen systems. In *International Conference on Distributed Frameworks for Multimedia Applications (DFMA'05)*, pages 254 – 261, Besancon, France, February 6-9 2005. IEEE.

[22] Saint-Voirin and Zerhouni. Cooperative systems modeling, example of a cooperative e-maintenance system. In *DFMA'06, Procs of the 2nd IEEE Int. Conf. on Distributed Frameworks for Multimedia Applications*, pages 83–90, Penang, Malaysia, May 2006.

[23] Shoham. Agent oriented programming. *Artificial Intelligence*, 60:245–252, 1993.

[24] Szymanski, Bangemann, Thron, Thomesse, Reboeuf, Lang, and Garcia. Proteus - a european initiative for e-maintenance platform development. In *9th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA2003*, Lisboa Portugal, 16-19 September 2003.

[25] Szymansky, Thron, Thomesse, Rebeuf, and Lang. Web service driven integration platform for industrial maintenance oriented applications. In *2nd IEEE int. conf. on Industrial Informatics, INDIN'04*, pages 114–119, Berlin, Germany, June 2004.