CREWS Report Series 97-07

Meta-modelling for cooperative processes

Selmin NURCAN, Colette ROLLAND

Centre de Recherche en Informatique
Université Paris 1 - Panthéon - Sorbonne
17, rue de Tolbiac
75013 Paris FRANCE
tel: (33) 1 - 44 24 93 65
fax: (33) 1 - 45 86 76 66
e-mail: (nurcan, rolland)@univ-paris1.fr

Proceedings of the Seventh European-Japanese Conference
on Information Modeling and Knowledge Bases
Toulouse, France, May 27-30, 1997

Meta-modelling for cooperative processes

Selmin NURCAN, Colette ROLLAND
Centre de Recherche en Informatique
Université Paris 1 - Panthéon - Sorbonne
17, rue de Tolbiac
75013 Paris FRANCE
tel: (33) 1 - 44 24 93 65
fax: (33) 1 - 45 86 76 66
e-mail: (nurcan, rolland)@univ-paris1.fr

Abstract: Cooperative work techniques are becoming very important in organisations as well as in the information systems community. The Computer Supported Cooperative Work(CSCW) discipline makes the assumption that collaborative work and processes can be supported by software tools. This requires among others to develop models able to represent cooperative work processes. In this paper we propose a meta-modeling framework to deal with a range variety of CSCW models. We present and exemplify a meta-model from which models can be instantiated. The meta-model is taylored to support the modeling of both well-structured and ill-defined work procedures and their interactions.

Keywords: Cooperative Work - Meta-modelling- Process model

1. Introduction

We are interested in Computer Supported Cooperative Work (CSCW) which examines the possibilities and effects of technological support for humans involved in collaborative group communication and work processes. Organizations are built on the principle that groups of people can carry out tasks which are not feasible individually. Therefore in most applications, well-structured, individually performed procedures coexist with ill-structured tasks which require cooperative work processes and both of them must be managed in the final solution. Cooperative work techniques become very important in organizations as well as in the information systems community. One can note the emergence of cooperative information systems. The development of information systems is itself becoming performed in a cooperative manner (Solvberg, 1995). In the CREWS[1] project we are developing an approach for supporting cooperative requirements engineering based on scenarios. Therefore, it is necessary

[1] This work is partially supported by the european ESPRIT long term research project CREWS (Cooperative Requirements Engineering With Scenarios).

to understand the specificities of cooperative work processes in order to take them into account in models and software tools built for supporting their enactment.

Our purpose is to propose a process meta-model, which can deal with both well-defined and wickled work procedures and their interactions, so as to represent a wide range of cooperative work processes.

This paper is organised as follows: In section II, we introduce computer supported cooperative work and situate workflow with respect to groupware. We shall notice that organizational reasons justify the joint use of these technologies. In section III, we introduce our needs, in terms of models and ways of working, for modelling and guiding cooperative work processes. In section IV, we present a cooperative process meta-model which provides means to deal with secure and well-structured cooperative work processes and has the flexibility to handle ill-structured cooperative work processes.

2. Cooperative work

The cooperative work or group work is the object of a multidisciplinary research field called Computer Supported Cooperative Work. The growth of connectivity greatly expands opportunities for office workers to cooperate and work together. Most organizations acknowledge that process simplification and automation are key success factors in the present competitive environment where the watchwords are productivity and quality.

Groupware is defined in (Ellis *et al.*, 1991) as follows: "*Computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment*". A well-known categorization (Ellis *et al.*, 1991) is the division into synchronous or asynchronous activity and co-located or distributed activity (see figure 1).

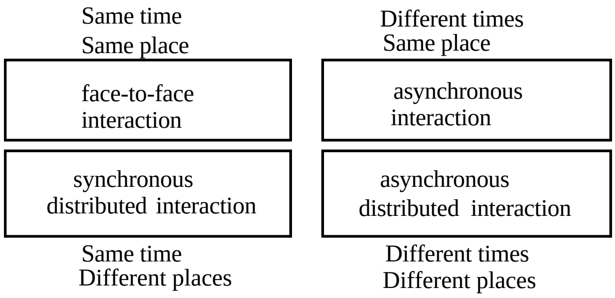


Figure 1 - Johansen's Space/Time matrix

Workflow is mainly concerned by scheduling and coordinating work between actors (Khoshafian *et al.*, 1992), (OVUM, 1991). It is defined for instance in N. Naffah (Naffah 94) as a *«cooperative work involving a number of actors which must realize tasks, in a given time span, according to a predefined procedure and having a global aim»*. In workflow applications, cooperative work means that several persons are involved in reaching a common goal, but each of them acts individually in a specific step of the work.

Based on the Ellis' definition of group work (involves a common task (or goal) and a shared environment), one might argue that workflow does not fulfill the requirements of the CSCW community because only one person executes his/her own task with his/her own data at a given time. However, taking a general view of the procedure, there is a common goal to reach by a group of people which share the same information.

Workflow is classified by J.Grudin (Grudin, 1994) in the distributed asynchronous area of the previous matrix as electronic mail systems. For many people, groupware supports unpredictable and ad-hoc interactions that occur in work group, whereas workflow automates strategies and predefined procedures. However, their global aims are the same: to increase the collective efficiency of groups of people engaged in fulfilling a common goal.

Workflow applications have been divided into two different categories depending on the nature of the processes supported (Palermo McCready, 1992). The first concerns well-structured and repetitive procedures having important coordination and automation needs (Nurcan, 1995a). In most current workflow software tools supporting well-structured processes, a procedure is a predefined set of partially ordered tasks. Each task has an assigned role corresponding to a group of actors, and the actor who actually executes the task is chosen from this group.

The second category of workflow applications deals with occasional and ill-structured (ad-hoc) work processes in organizations; a response to a call for tender in a commercial service or problem solving activities are examples of this class. The main characteristic of these applications is the information and knowledge-sharing within the work group more than the ordering of their tasks.

For many organizations, well-structured and ill-structured work processes coexist and must be managed in the final solution (Nurcan, 1995b) (Nurcan, 1996a) (Nurcan, 1996b). The integration aims to make the transition between the different types of group activities transparent. Current workflow products and their underlying control flow models require a strict respect of the predefined procedures. Therefore they cannot be used for ad-hoc workflow applications or deal with the dynamic modifiability of predefined scripts. More and more, users ask for adaptive workflow products and models which can provide the robustness and the security of the predefined scripts and the flexibility of ad-hoc applications.

Providing a single set of concepts to model both aspects of group work processes is our concern is this paper.

3. Models and way of working

Group work application development starts with the modeling of the process to automate. The implementation of this kind of application requires a preliminary analysis phase before the process may be modeled. For each stage of the work, one has to determine who does what within the task, when, after and before which other task. Information holders, types of handled documents, possible locking points…etc have also to be defined.

When the work process is well-structured, the corresponding procedure is a predefined set of partially ordered tasks. Partially ordered means that tasks are not necessarily executed sequentially: loops and parallelism can appear. Each task has an assigned role corresponding to a group of actors, and the actor which shall execute the task shall be chosen among this group.

Finally, the modelling of a procedure (see figure 2) requires the identification of:

• event(s) which trigger(s) the procedure,

• tasks which compose it and their relationships with the others: these relationships define sequential, parallel (with rendezvous points) and conditional chains, and for each task:

  • events which trigger its execution,
  • resources (data+tools) which are necessary for its execution, and
  • the associated role.

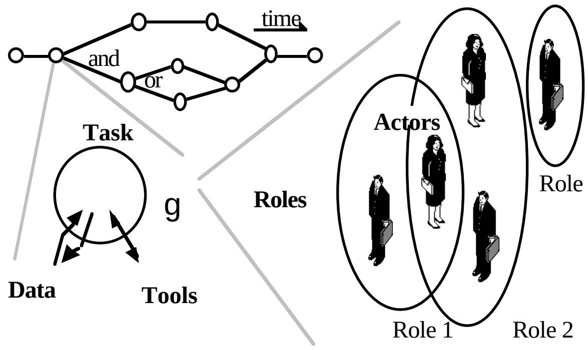


Figure 2 - Procedure representation

We have considered seven models dealing with task-workflow-agent-role representations, respectively OSSAD (Dumas, 1990) (Dumas *et al.,* 1990), ICN (Ellis, 1979) (Ellis, 1994), InConcert (McCarthy, 1993), VPL (Swenson, 1993), I* (Yu, 1994), Enterprise Modelling (Bubenko 1994), (Loucopoulos 1995) and ITHACA (Ang, 1993). This study showed a convergence on a set of concepts such as goal, procedure, task, role, actor, resource, decomposition of tasks, etc. However, an appropriate model for a large variety of cooperative work processes (going from well-structured to ill-structured) must also provide means to represent unstructured activities. We integrated these concepts in one single meta-model that we present in the following section.

4. A process meta-model for the representation of cooperative work

An approach to generate guidance centered process models has been initially proposed in (Rolland, 1993) and further developed in (Rolland, 1994a), (Rolland, 1994b), (Rolland, 1995). Authors refer to these models as "ways-of-working" since they are intended to guide application engineers in their way of working to solve a design problem. We believe that the proposed approach is applicable to any process. However the problem of distributed process guidance has not be tackled in (Rolland, 1995).

We have extended the process meta-model presented in (Rolland, 1995) in order to obtain a *cooperative process meta-model* to be used for any cooperative process.

4.1. The cooperative process meta-model

We propose a *meta-model* as a basis for process model definition. Since a process meta-model carries information about the process model, an instantiation of it shall result in a process model. Our approach introduces three levels of process modelling:

-At the lowest level, process traces are recorded.

-At the second level, ways-of-working are defined. A way-of-working is a process model i.e. a description of process. It has a prescriptive purpose and is similar to the concept of plan. A process is then, an instantiation of a process model which is executed.

-The knowledge required to design such models is related to the third level of abstraction and takes the form of a process meta-model. A process meta-model provides a set of generic concepts for describing ways-of-working which are therefore, instances of the process meta-model.

The process meta-model allows us to deal with many different situations in a flexible, decision-oriented manner. Moreover the meta-model can support different levels of granularity in decision making as well as non determinism in process performance. It identifies a decision in context as the basic building block of ways-of-working and permits their grouping into meaningful modules. Parallelism of decisions and ordering constraints are also supported.

The output of a process is a *product*, it can be requirements specification or a conceptual schema or a loan offer to a client in a bank or messages exchanged between members of a group or a set of business goals.

In the presentation of the *cooperative process meta-model* we follow a bottom up approach starting with the concept of *context*, intoducing then the concepts of *role*, *action* and *product*, and ending with an overall view of the concepts progressively introduced.

4.2. The concept of context

The central concept of the process meta-model is the one of *context* which associates a situation with a decision made on it.

A *situation* is a part of the product it makes sense to take a decision on. Situations can be of various granularity levels; they can be either atomic like an attribute of an object class or they can be coarse-grained like the whole product.

A *decision* reflects a choice that a user can make at a given moment in the process. A *decision* refers to an *intention.* An *intention* expresses what the user wants to achieve, the goal.

A *context* is the association of a *situation* and a *decision* which can be taken in this situation. A decision is not sufficient in itself, it needs to be associated with the situation in which it applies. A situation can be associated with several decisions. Acting in a context corresponds to a step in the process: in a given situation, and in order to progress in the process, the user has to take a decision (figure 3).

4.3. *The concept of role*

In procedural workflow applications tasks are individual and are performed by individual roles. Each task is assigned to a role corresponding to a group of actors (i.e. the collection of the role object) having the same privileges and obligations to achieve the task.

Now, we want to deal with cooperative processes comprising unstructured -emergent- cooperative activities. Then, acting in a context corresponds to a step in the cooperative process to which users participate having well defined roles: in a given situation, a user has an intention (because of his/her role in this process), and that makes him/her progress in the cooperative process.

To this end, we introduce the concept of *role*, and then classify it into *individual role* and *group role* (figure 3). A group role *contains* several individual roles. As we are in a cooperative process framework, each context *is attached* to a role.

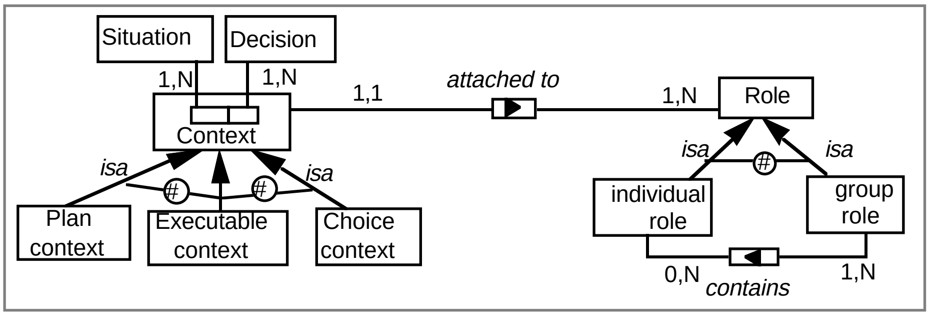


Figure 3 : The context is attached to a role

4.4. *The different types of contexts*

A situation exists at different levels of granularity. Further, decisions have consequences which differ from one granularity level to another. The different contexts are classified (figure 3) according to their consequences in the meta-model into *executable contexts*, *plan contexts*, and *choice contexts*.

4.4.1. *Executable context*

At the most detailed level, the execution of any process can be seen as a set of transformations performed on the product, each transformation resulting from the execution of a deterministic action. Such an action is a consequence of a decision made in a certain context. This leads to the introduction of the concept of an *executable context.*

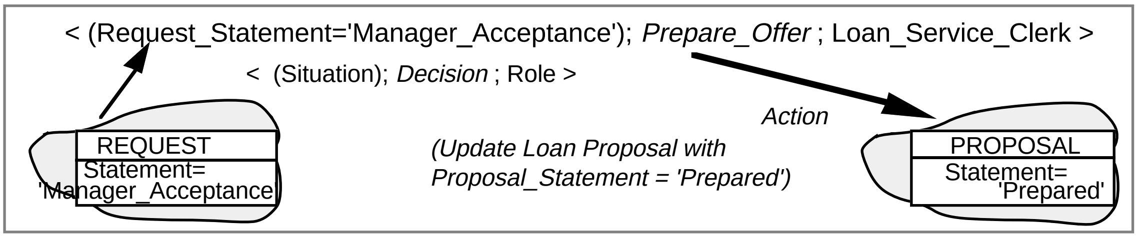


Figure 4 : Example of an executable context

An *executable context* implements a decision, its intention is realised by an action (figure 4). Therefore, in the meta-model (figure 5), an executable context is associated with an action. An *action* performs a transformation of the product, it is the implementation of a decision. Performing an action changes the product and may generate a new situation which is itself, subject to new decisions.

The concept of action

We classify actions into two types (figure 5) according to their characteristics: *individual action* and *conversation action*.

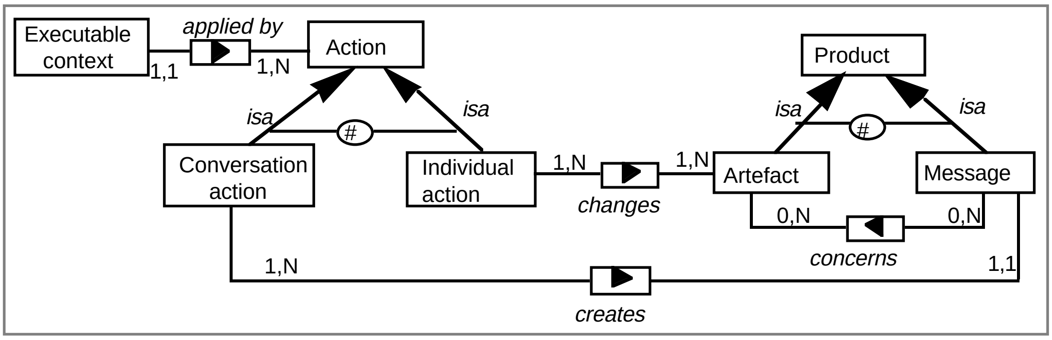


Figure 5 : Actions and products that they transform

Performing an individual action or a conversation action does not change the same kind of product. Individual actions perform transformations of artefacts while conversation actions create messages.

Therefore, we classify the concept of *product* into *artefact* and *message* (figure 5). *Artefact* represents the static component of the information system.

We need to represent also the unstructured -conversational- activities of the group work. So, we must be able to keep track of these conversations. We introduce the *message* concept as the basic component of the conversational activity. A *message* may *concern* several *artefacts.*

The *individual action* can be complex or simple. A *complex action* is composed of *actions*. A *simple action* performs a tranformation of *(changes)* an artefact by creating, updating or deleting it. An *individual action* is performed by an *individual role* (figure 6). Figure 4 shows an executable context which is applied by an individual action.

We want also to deal with group activities, in the sense that several participants can synchronously act in the same activity by exchanging messages. We represent this type of cooperation by the *conversation action.*

The *conversation action* is *attached to* a *group role*. It creates several messages, each message being *produced by* an individual role *contained by* the previous group role (figure 6).

From any *conversation action* may *emerge* new *contexts* (figure 6). These contexts can be executable and associated to actions, which might be conversational and then, triggers new contexts and so on. This feature enables the cooperative process meta-model to deal with ill-structured cooperative work processes as well structured cooperative work processes. An example of conversation action is given in paragraph IV.4.4.

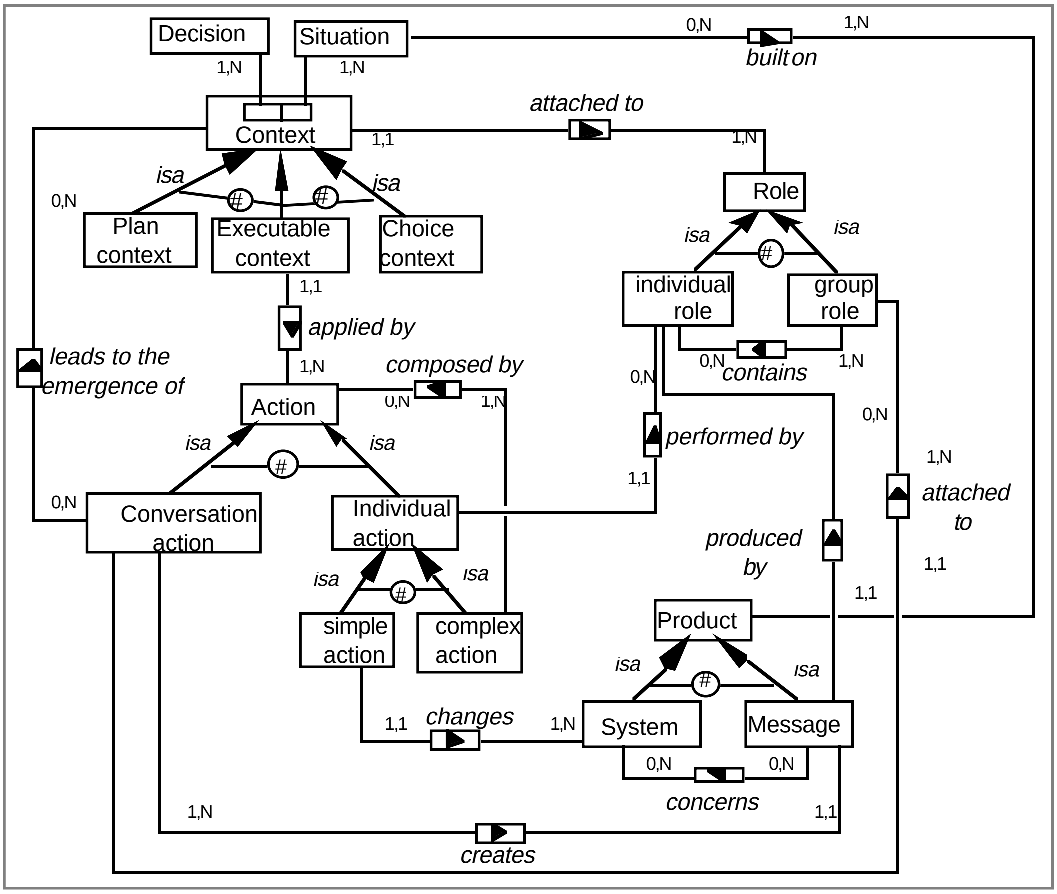


Figure 6 : The cooperative process meta-model

Executable contexts establish situation-based links among contexts, namely *correlation links*. This is modelled in figure 6 by the loop among contexts through action and situation. The term *correlation link* refers to the composition of the three following relationships: *applied by*, *changes* or *creates*, and *built on*.

4.4.2. Choice context

A user may have several alternative ways to fulfil a decision. Therefore, he/she has to select the most appropriate one among the set of possible choices. In order to model such a piece of process knowledge, we use a second specialisation of the concept of context, namely the *choice context* (figure 7).

A *choice context* corresponds to a situation which requires the exploration of alternatives in decision making. Each alternative is an approach or a strategy for the resolution of the issue being faced by the user in the current situation. By definition a choice context offers a choice among a set of strategies, all of them achieving the same purpose. In this sense, one can look upon the choice context as being goal oriented.

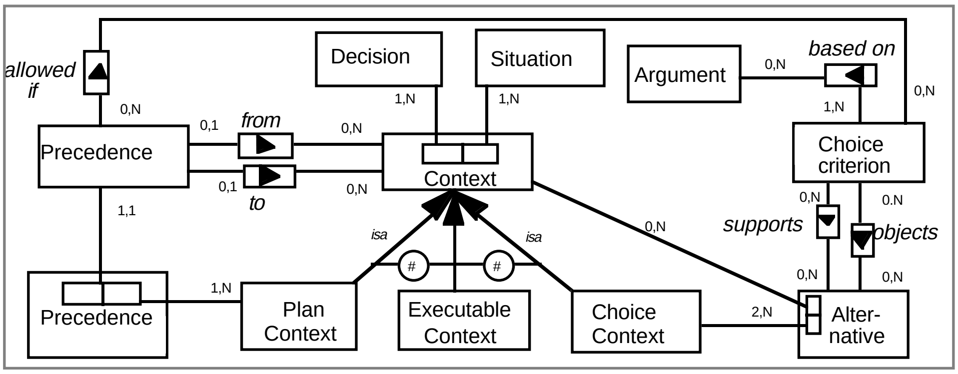


Figure 7 : The representation of the concept of context

There are two major differences between the *choice context* and the *executable context*: the first one lies in the absence of any alternatives in the latter and the second is that a choice context has no direct consequence on the product.

In the process meta-model, the various alternatives of a choice context are represented in the *alternative relationship* (figure 7). They are associated to choice criteria based on arguments.

A choice criterion is a combination of arguments which *supports* or *objects to* an alternative of a choice context. It may provide priority rules to select one *alternative* among several depending on the *arguments.*

Since alternatives of a choice context are also contexts, contexts may share an *alternative* relationship (figure 7), leading to *alternative-based hierarchies of contexts.* The alternative-based relationship among contexts allows the refinement of large-grained decisions into more fine-grained ones. This is a means by which the process meta-model handles the granularity problem (figure 8).

The introduction in the process meta-model of *alternatives* and *choice criteria* will allow the way-of-working to support the user in exploring possible strategies to resolve an issue and in selecting the most appropriate one. This alternative-based guidance leaves freedom to the user who can make a choice which is not even one of the predefined alternatives proposed by the way-of-working. This feature enables the cooperative process meta-model to deal with exception handling in workflow applications.

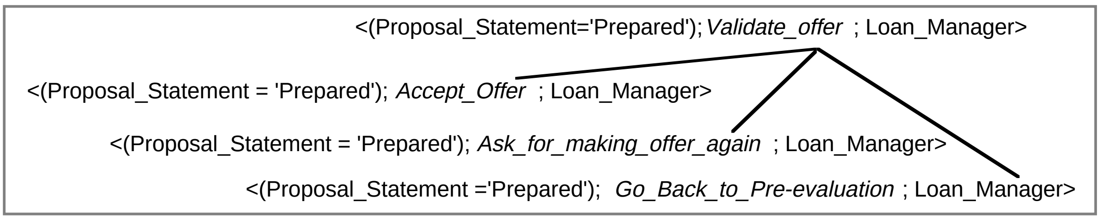


Figure 8 : Example of a choice context

4.4.3. *Plan context*

In order to fulfil an intention associated to a certain situation, a user may be required to take a set of decisions on corresponding situations; he/she has to follow a plan. To this end, a third specialisation of context, namely, *plan context* is introduced. A *plan context* is an abstraction mechanism by which a context viewed as a complex issue can be decomposed in a number of sub-issues. Each sub-issue corresponds to a sub-decision working on a sub-situation. The decomposition of context is another means provided by the meta-model to solve the granularity problem.

The component contexts can be of any type i.e. executable, choice or plan contexts. For example, for the intention named "Process_Loan_Request" to be fulfilled, the two intentions "Record_Request" and "Evaluate_Request" must be satisfied. This is modelled (figure 9) by a plan context called "<(Request_Message), Process_Loan_Request, Loan_Service_Clerk>" decomposed into two contexts:
"<(Request_Message), Record_Request, Loan_Service_Clerk>" executable context and "<(Request_Statement='Recorded'), Estimate_Request, Loan_Service_Clerk)>" choice context.

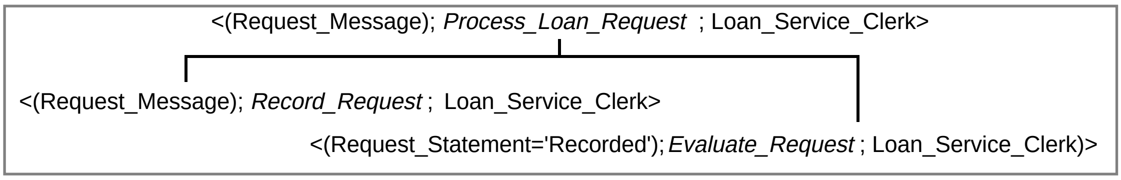


Figure 9 : Example of a plan context

In the process meta-model the decomposition of a plan context into its more elementary contexts is represented (figure 7) by the relationship *precedence graph* between *context* and *plan context*. The ordering of the contexts, within a plan, is defined by the *precedence graph*. The nodes of this graph are contexts while the links -called *precedence link*s- define either the possible *ordered transitions* between contexts or their possible *parallel enactment*. Based on arguments, a choice criterion may be assigned to a link. The choice criterion defines when the transition can be performed. Flexibility is introduced by allowing several sets of possible parallel or ordered transitions to be defined in the same graph. This feature enables the cooperative process meta-model to deal with well-structured workflow applications which require the use of a model in terms of ordered steps. The precedence graph corresponding to the previous plan context is shown by the figure 10.

Decomposition of contexts can be made iteratively leading to hierarchies of contexts. This hierarchical link is refered to as a decomposition link. Notice that this link corresponds in figure 7 to the composition of the precedence graph relationship with the *from* and *to* relationships.

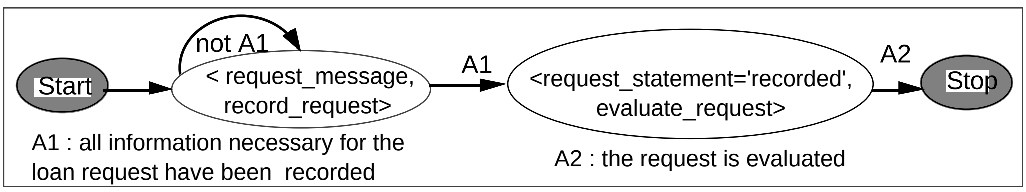


Figure 10 : Example of a precedence graph

Plan contexts provide a different type of guidance than executable and choice contexts do. They support the user in performing long term transactions, providing advice on the ordering of component activities, whereas *choice contexts* help in making the appropriate choice in the situation in hand and *executable contexts* tell how to implement the decision taken.

Each type of context influences the on-going process in a different manner: an executable context affects the product and generates a new situation, which itself becomes the subject of decisions; a choice context does not change the product but helps to further the decision making process through the refinement of an intention; a plan context provides the means to manage the complexity of an intention by providing a decomposition mechanism. Performing decomposition and refinement iteratively allows the users to reach executable intentions and thus, to act on the product.

IV.4.4. *Conversation action*

In this paragraph we examplify the use of the conversation action for a non structured group activity. Let us take an example from the Air Traffic Control case study and assume that the context C < G1 "minimize risks of accidents", Operationalise G1> requires to call a group of experts (we name it the "risk elucidation group") for a brainstorming session.

In other words, the strategy selected for context C is "brainstorm". The guidance provided by this strategy (Rolland, 1997) suggests the following tactics:

(1) to define the group role required for this cooperation,

(2) to execute a conversation action within the previously defined group role and having the initial input context as situation.

The *Risk elucidation group* is a group role which contains the following individual stakeholders: Airport manager, ATC center manager, a representative of airlines managers, a representative of pilots, and a local autority.

The executable context is applied by a conversation action leading to the creation of several messages (figure 11).

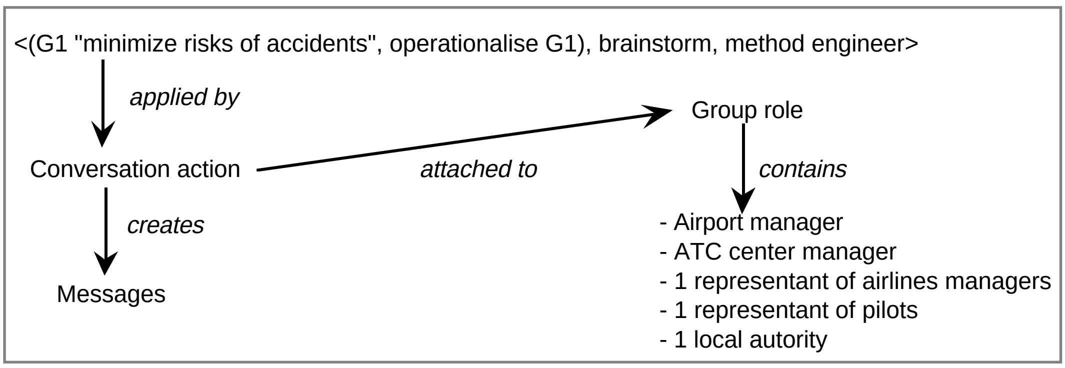


Figure 11 : An executable context leading to the execution of a conversation action

Let assume as an example that the flow of messages is the following:

Message 1: (ATC center manager)

Have we got a report about reasons of accidents happened during the five last years in the world ?

Message 2: (Airport manager)

No, we don't. But we have some informations about the last three major accidents.

Message 3: (ATC center manager)

So, what about the reasons ?

Message 4: (Airport manager)

In Strasbourg in France, it was a human error.

At Delhi, the reason was twofold; the accident was partly due to the heavy air traffic and partly to a human error, due to his poor knowledge of the english language, the pilot misunderstood the message of the control tower.

In the US, it was a confusion about the airport. The pilot made an error in typing the airport and the computer understood the airport code as Bogota in South America while the aircraft was to land in California.

Message 5: (Representative of airline managers)

So if we want to minimize risks of accidents we have to *decrease risk of human error*.

Message 6: (Pilots representative)

Sometimes what is called human error is not. How to decrease the human error in the accident occurred in US. You must rather review computer systems.

Message 7: (ATC center manager)

It's more convenient to talk about Human-Computer interface for this accident. So, our goals are to *decrease risk of human error and to review all human-computer interactions.*

Message 8: (Pilots representative)

And what about the accident in Delhi ? The human error was not the unique reason, isn't it ?

Message 9: (ATC center manager)

The number of aircrafts allowed to cross the controlled airspace is too high in Delhi.

Message 10: (Local autority)

Precisely, since 2 years local autorities argue that this number must decrease in our city too. People living near the airport are disturbed because of the noise and late/early take offs and landings. In order to minimize risks of accidents we must *limit the number of aircrafts allowed to cross the controlled airspace*.

As a conclusion of this message exchange, the conversation action generates three emerging contexts :

- < message 5, create G2 "decrease risk of human error">
- < message 7, create G3 :"review human-computer interactions">
- < message 10, create G4 :"limit the number of aircrafts allowed to cross the controlled airspace">. The new contexts are inserted in the contexts pile for further processing.

IV.4.5. *The concept of way-of-working*

It should be clear now that due to the meta-model concepts, the basic building block of a way-of working is an instance of context that we call also *context*. Contexts in the meta-model have hierarchical relationships of two different types, decomposition and refinement. In the way-of-working, we suggest a grouping of contexts based upon these links. The modules resulting from this grouping are hierarchies of contexts called *trees*. Finally, a way of working can be composed of several trees. This leads to the final vision of a way-of-working as a *forest* of trees (figure 12).

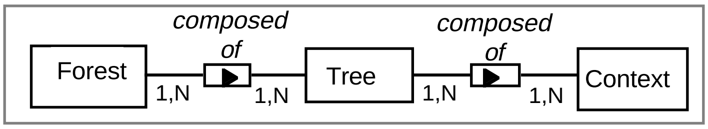


Figure 12 : The way-of-working structure

IV.4.6. *An example of way of working as a forest of trees*

We plan to model the loan process in a bank (figure 13). The working rules are given below:

When a customer applies for a loan, the bank clerk in charge of his banking account analyses the loan request according to its nature.

He/she can decide to accept or refuse the request himself, or to ask for a deeper evaluation. In the third case, first a pre-evaluation is made by the financial department (ill structured task carried out synchronously by a group of experts), then the request is examined by the loan manager in order to accept or to refuse it.

The study of the request by the loan service clerk must be validated by the loan manager who has the possibility to either :

. accept the loan offer prepared by the loan service clerk, or

. ask the loan service clerk to review it, or

. ask the financial department for a complete re-evaluation of the loan request.

When the decision is favourable, a proposal of loan is sent to the customer by the clerk's assistant. When the decision is unfavourable, a refusal letter is sent by the same person.

Four different roles are involved in the loan process :

- The loan service clerk which is in charge of the client account,
- The loan manager,

- The work group constitued by the financial manager and three experts in the financial departement,
- The clerk's assistant.

The information systems' objects that we defined are: REQUEST, PROPOSAL, CLIENT.

Request and *proposal* have noticeable states during their life cycle, respectively represented by *request_statement* and *proposal_statement*.

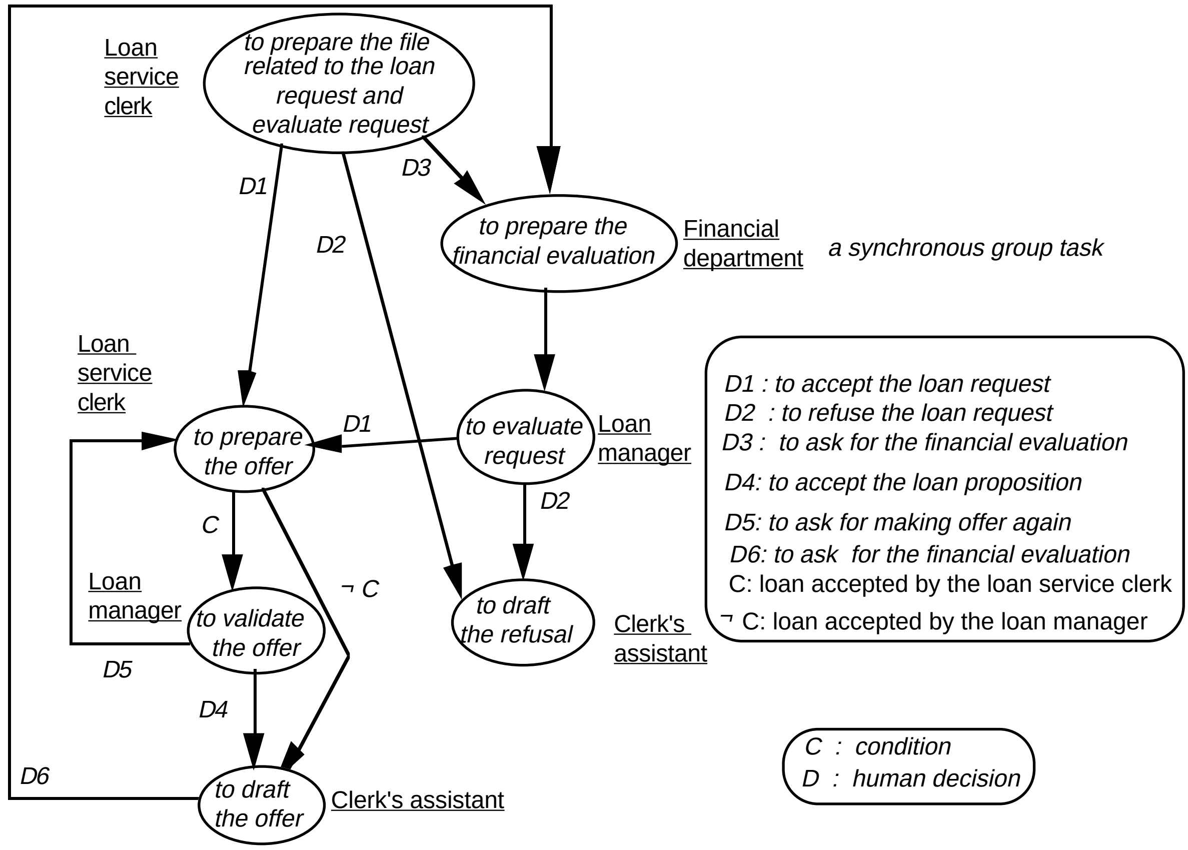


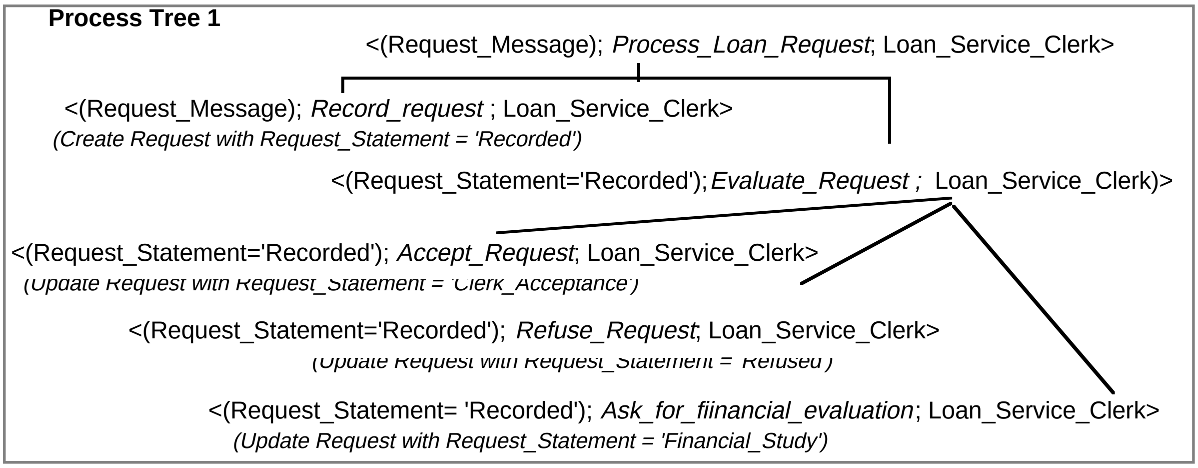
Figure 13 : The graphical representation of the Loan Request case study

The approach consists of instantiating the concepts of the meta-model. We have to define the executable, choice and plan contexts and their decomposition and refinement links.

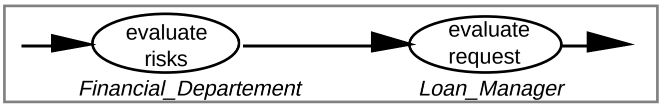
The loan process is then represented by a forest composed of six trees.

Each tree describes a piece of knowledge about the process associated to a given role. The trees describe the process in a workflow style but, in addition, encapsulate guidance to support the agents performing their tasks.

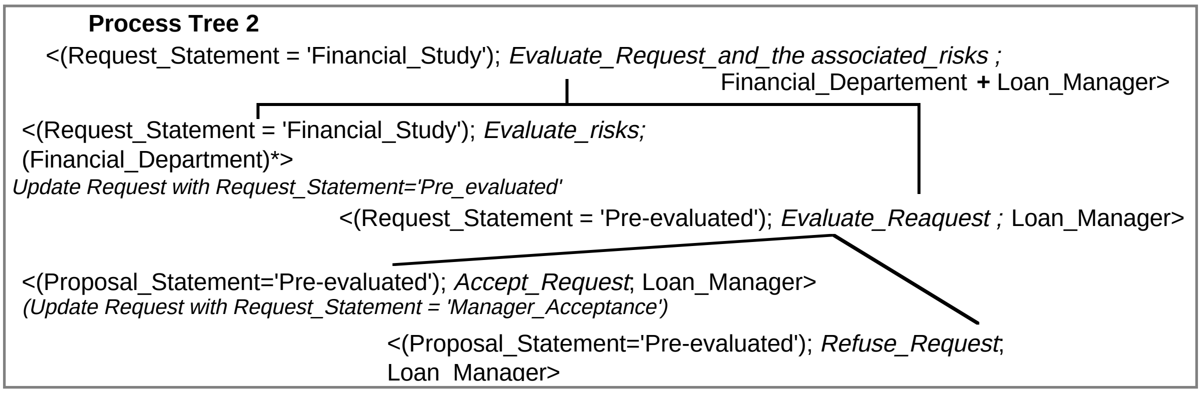
The **first tree** describes the evaluation of the loan request by the loan service clerk. The first component context of this plan is an executable one associated to an individual action: Create Request. The second component of the plan is a choice context with three alternatives. It provides an alternative-based guidance to the clerk.



The **second tree** describes the way-of-working for the group work *processed by* the financial departement and the loan manager. The root of the tree is a plan context which represents the risk evaluation by the financial department, then the request evaluation by the loan manager. The predefined decomposition of this group work is described in the following precedence graph.



The risk evaluation is a group work synchronously processed by a group of experts in the financial department. It is represented by an executable context associated to a conversation action. The evaluation of the request by the loan manager is defined by a choice context with two alternatives, to accept or to refuse the request, each of them being described by an executable context.

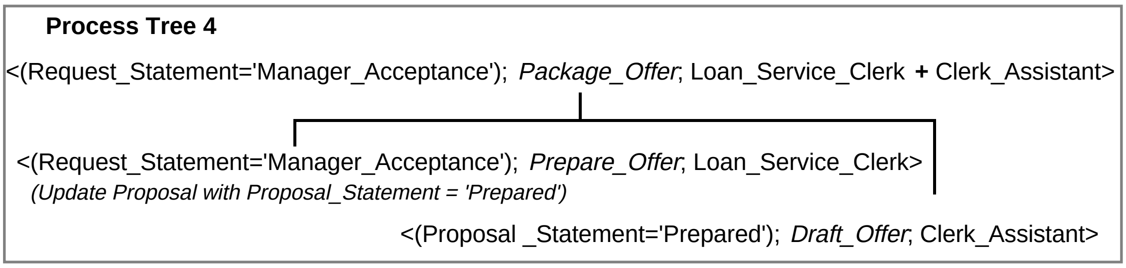


Tree 3 represents the drafting of a refusal letter by the clerk's assistant when the request is "refused". It is an executable context associated to an individual action.

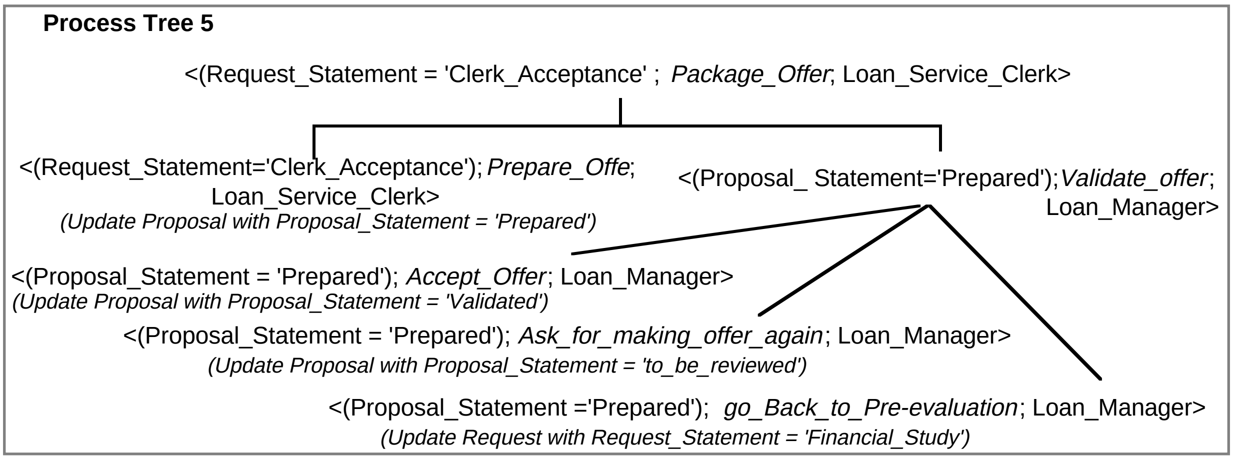
Process Tree 3

<(Request_Statement='Refused'); *Draft_Refusal_Letter*; Clerk_Assistant>

Tree 4 describes the package of the loan offer by the loan service clerk and his/her assistant when the situation corresponds to manager acceptance. This is a plan context composed of two executable contexts affected to individual roles.



When the situation corresponds to clerk acceptance, the process can be performed according to **tree 5**. The first component context of the plan is an executable one and corresponds to the clerk's individual action in order to prepare the loan offer. The second component context of the plan represents the validation of the offer by the Loan Manager with three alternatives.



Tree 6 represents the drafting of the offer by the clerk's assistant when the proposal is "*validated*".

Process Tree 6

<(Proposal _Statement='Validated'); *Draft_Offer*; Clerk_Assistant>

5. Conclusion

In this paper, we have presented a *cooperative process meta-model* which provides means to deal with secure and rather well-structured work processes and provides the flexibility to handle ill-structured cooperative processes . It allows us:

- to represent cooperative work processes,
- to integrate conversations between agents,

• to guide and keep track of what happened in cooperative brainstorming sessions,

• to model the emergence of new contexts;

all these being made in an homogeneous manner.

An instantiation of the cooperative process meta-model results in a cooperative process model allowing to deal with a large variety of situations in a *decision-oriented* manner.

The concept of plan context enables the cooperative process meta-model to deal with well-structured cooperative processes which require the use of a control model. In fact, the corresponding precedence graph defines the ordering of the component contexts (the possible ordered transitions between contexts or their possible parallel enactment).

The alternative-based guidance of the choice context leaves freedom to users who can make a choice which is not even one of the predefined alternatives proposed by the way-of-working. This feature allows the cooperative process meta-model to deal with exception handling in cooperative processes.

The concept of conversation action allows us to represent emergent cooperative activities. It enables the cooperative process meta-model to deal with ill-structured cooperative work processes and the emergent component of globally well-structured cooperative work processes.

Our current work consists of building a cooperative environment which supports the definition of cooperative process models (in terms of ways-of-working) and provides the flexible guidance of groups in well-structured and/or ill-structured cooperative work processes. This environment is an extension of the MENTOR process centred environment (Si-Said, 1996).

Bibliography

Ang, J.S.K. and Conrath D.W. (1993): The ITHACA Office Object Model: Modeling and Implementation, *Data Base*, November 1993, p. 5-14.

Bubenko, J. (1994): Enterprise Modelling, *Ingénierie des Systèmes d'Information*, Vol 2, N° 6, 1994.

Dumas, P. and Charbonnel, G. (1990): *La méthode OSSAD - Pour maîtriser les technologies de l'information - Tome 1: Principes*, Les Editions d'Organisation, Paris.

Dumas, P., Charbonnel, G. and Calmes, F. (1990): *La méthode OSSAD - Pour maîtriser les technologies de l'information - Tome 2: Guide pratique*, Les Editions d'Organisation, Paris.

Ellis, C. (1979): "Information Control Nets, A Mathematical Model of Office Information Flow", In *Proceedings of the ACM conference on Simulation, Measurement and Modelling of Computer Systems*, p.225-240.

Ellis, C.A., Gibbs, S.J. and Rein, G.L. (1991): "Groupware: some issues and experiences", *Communications of the ACM*, 34(1), p.38-58.

Ellis C.A., Wainer J. (1994): "Goal-based models of collaboration", *Collaborative Computing*, Volume 1, Number 1, March, p.61-86.

Grudin, J. (1994): Computer-supported cooperative work: History and focus. *IEEE Computer, Special CSCW*, May 1994, p.19-26.

Khoshafian, S., Baker, A.B., Abnous, R. and Shepherd, K. (1992): "Collaborative work and work flow in intelligent offices",*Intelligent Offices: Object-Oriented Multi-Media Information Management in Client/Serveur Architectures*, Wiley.

Loucopoulos P., Kavakli E. (1995): "Enterprise Modelling and the Teleological approach to Requirements Engineering", *International Journal of Cooperative Information Systems*, Vol. 4, N° 1, p. 45-79.

McCarthy D.R, Sarin S.K. (1993): "Workflow and transactions in InConcert", *Bulletin of Technical Committe on Data Engineering*, Vol. 16, N° 2, IEEE, june, *Special Issue on Workflow and Extended Transactions Systems*. p. 53-56.

Naffah, N. (1994): "Workflow: Etat de l'art et évolution", In *Proceedings of the Conference IT FORUM'94,* Télécom Paris.

Nurcan, S. and Trolliet, J.Y. (1995a): "Une méthode d'analyse et de conception pour les applications workflow", In *proceedings of the 13th INFORSID congress*, May 31-June 2 1995, Grenoble, p.453-472.

Nurcan, S. and Chirac, J.L. (1995b): "Quels modèles choisir pour les applications coopératives mettant en œuvre les technologies de workflow et de groupware ?", In *Proceedings of the AFCET 95 congress,* October 25-27 1995, Toulouse, p.593-602.

Nurcan, S. (1996a): "A method for cooperative information systems analysis and design: CISAD", In *Proceedings of the Second International Conference on the Design of Cooperative Systems (COOP'96)*, 12-14 juin 1996, Juan-Les-Pins, p.681-700.

Nurcan, S. (1996b): "Analyse et conception de systèmes d'information coopératifs", *Numéro thématique "Multimédia et collecticiel" de Techniques et Sciences Informatiques*, Vol. 15, N° 9.

Nurcan, S., Gnaho, C., Rolland, C (1996): "Defining Ways-of-Working for Cooperative Work Processes", In *Proceedings of the First International Conference on Practical Aspects of Knowledge Management (PAKM) Workshop on Adaptive Workflow*, October 30-31, 1996, Basel, Switzerland.

OVUM (1991): *Workflow Management Software*. Ovum Ltd., London, England.

Palermo, A.M. and McCready, S.C. (1992): "Workflow software: A primer", In *Proceedings of the Conference GROUPWARE'92*, London, p.155-159.

Rolland C. (1993): "Modeling the Requirements Engineering Process", *Information Modelling and Knowledge Bases*, IOS Press.

Rolland C., Prakash N. (1994a) "A Contextual Approach for the Requirements Engineering Process", *Proc. Int. IEEE Conf. on Software Engineering and Knowledge Engineering (SEKE94)*, Riga, 1994.

Rolland C. (1994b) "Modeling the evolution of artifacts", *1st IEEE Int. Conference on Requirements Engineering*, Colorado Springs, Colorado, 1994.

Rolland, C., Souveyet, C., Moreno, M. (1995): "An approach for defining ways-of-working", In *Information Systems Journal*, Vol. 20, N° 4.

Rolland C., Nurcan S., Grosz G. (1997): "Guiding the participative design process", Association for Information Systems 1997 Americas Conference, August 15-17, 1997, Indianapolis, Indiana.

Si-Said S., Rolland C., Grosz G. (1996): "MENTOR : A Computer Aided Requirements Engineering Environment", *in the Proc. of the 8th CAISE Conf. Challenges In Modern Information Systems*, Heraklion, Crete, Greece, May 1996.

Swenson K.D. (1993): "Visual Support for Reengineering Work Process", In *Proceedings of the Conference on Organizational Computing Systems*, ACM, Milpitas, California, p. 130-141.

Yu E.S.K., Mylopoulos J. (1994): "From E-R to "A-R" - Modelling Strategic Actor Relationships for Business Process Reengineering", In *Proceedings of th 13th Int. Conference on the Entity-Relationship Approach,* December 13-16, 1994, Manchester.