From theory to practice: cooperation models in a sustainable product life-cycle

B. David[1,] F. Tarpin-Bernard[2],
J.Poquet[1+3], K. Saikali[1], N. Boutros[1]

[1]*Laboratoire ICTT, Ecole Centrale de Lyon,*
B.P.163, 69131 ECULLY Cedex, France

[2]*Laboratoire ICTT, INSA de Lyon,*
20, av. Albert Einstein, 69621 VILLEURBANNE Cedex, France

[3]*Laboratoire d'Automatique de Grenoble (LAG) ENSIEG-INPG,*
B.P.4, 38402 Saint Martin d'Hères Cedex, France

Abstract: Our aim is to compare cooperation models and methods with the applied field of sustainable manufacturing product life-cycle. From the CSCW point of view, we examine different organizational cooperation models and present an architectural model, which is used during design, implementation and run-time of a cooperative system. This model is called AMF (French acronym for Multi-Facets Agent model). We also describe methodological aspects which must be applied to organize the adaptation process of models to a new application field. Currently, we are applying our model to the recycling of manufacturing products which today is a major concern in the majority of industrialized countries. To be effective, recycling cannot be only considered at the ultimate stage of the product life-cycle. We present an approach in which recycling is taken into account early, during different stages and mainly in the design stage. We discuss global organization of the manufacturing product life-cycle, the information flows needed and different kinds of cooperation that must be applied.

Keywords: Computer Science, CSCW, Multi-agents system, Workflow system, cooperation models, sustainable product life-cycle, System design, Architecture.

1. Introduction

In this paper we describe a convergence process between application field (sustainable product life-cycle) and CSCW theories. After a short description of sustainability and its impact on the life-cycle of industrial products we examine different contributions of CSCW to a concurrent engineering approach of design, manufacturing and noble recycling process. We explain several aspects and modes of cooperation, and we present an architectural model which is usable in design, implementation and use of a cooperative system. Back to the application field, we describe how we apply the theories presented in the previous paragraphs.

2. Application needs

The recycling of products at their end of life constitutes a world-wide concern. Increased industrialization and new markets have led to an accumulation of used technical consumer goods, which results in a higher exploitation of raw materials, energy and landfill sites. In order to reduce use of natural resources, conserve precious energy and limit the increase in waste volume, the linear progression from production through consumption and finally to landfill sites must be stopped either by product reuse, part reuse or material recycling as explained by Hiessl et al. [1]. In different countries different approaches have been proposed, namely: secondary production, green design, durable production, durable design, industrial ecology, clean production, sustainable industrial development, sustainable production, etc.

The durability principle of development is undoubtedly the richest approach, because it gives a global objective: " to satisfy the needs for the present without compromising the possibility of future generations of satisfying their own needs ". This principle received world-wide recognition when more than 150 countries signed the charter of the conference "Earth Summit" in Rio de Janeiro in 1992. At European Community level, the treaty of Maastricht contains in article 2 the principle of “durability and non-inflationary growth respecting the environment”.

Durable production can be defined as the design, production, distribution, recovery and recycling of products so that the environmental impact and the level of use of the resources match the earth's capacities. In this very broad interpretation, the aim is to reduce the use of resources and the impact associated in all the stages of the product life cycle. The evolution towards durable development implies changes in the models of design, manufacturing and use.

Two main recycling approaches can be identified: recycling by energy valorization, crushing and sorting raw materials, and noble recycling which is defined as disassembly, repairing, reconditioning and reuse of components. The first approach is in practice today. The second one rather has sustainability in view: to obtain the sufficient effectiveness, one cannot be satisfied with a recycling a posteriori, i.e. without having information on the way in which the product was designed and manufactured. The need to possess during recycling the information elaborated during the design - industrialization – manufacturing stages becomes fundamental. It is important to collect information on design choices and assembly methods in order to be able to disassemble more or less automatically and to be able to implement an efficient “noble” recycling process.

2.1 Functional organization for sustainability

To achieve noble recycling of sustainable products, the problem must be tackled at the **design stage**, i.e. to choose appropriate materials, to allow easy separability between consumed parts and reusable parts, to integrate a structure for testability into the product and to choose assembly and disassembly methods which are compatible with multiple assemblies and disassemblies by automatic, semi-automatic and manual tasks.

The manufacturing stage must apply design decisions in order to allow maintenance and noble recycling tasks.

The aim of the **noble recycling process** is to identify by testing and to extract the largest possible number of parts and components to be reused at the same level (without degradation) at the manufacturing stage for the production of new products with the same degree of reliability.

These three stages are closely related by the need for information; working on disassembly without official information seems impossible. Disassembly is an activity, which to be automated, assisted or merely shared between several actors, needs to be well known. The first solution for availability of information can be based on the traditional diagram of propagation of information in the "waterfall" life cycle. This mono directional flow is essential to allow recycling to be carried out under satisfactory conditions. This approach is necessary, but not sufficient to satisfy the objectives of durability expressed above.

The Information Management System, which is in charge of managing the information collected on the waterfall, is a PDMS (Product Data Management System). It is mainly charged with obtaining the product model (from the design stage), the manufacturing process model (from the industrialization and manufacturing stages) and possible modifications of the product during the maintenance stage. If we have this information, it effectively becomes possible to elaborate the recycling processes. However, a more appropriate approach is described hereafter.

2.2 Design for disassembling

In order to recycle conveniently, it is important to establish a closer link between the design and recycling activities. Consequently, it should be possible to take disassembling requirements (as new solutions of reuse) into account during the design stage and to have at one's disposal during the recycling stage the information on design preferences. This approach allows information feedback between at least the design and recycling stages, but also the industrialization and manufacturing stages and more generally between the product elaboration stages and recycling activities.

The new information support must be bi-directional and allow open access to information from all stages of the product life cycle. This new view on information flow is explained in Fig. 1.

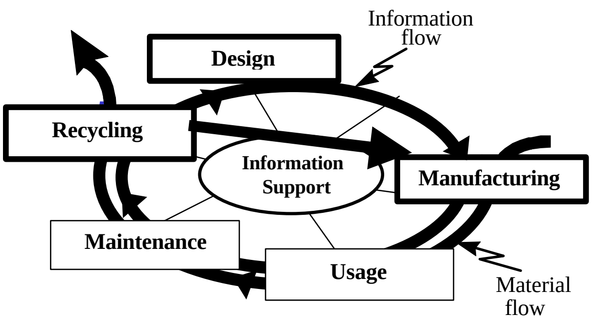

Fig. 1: Design for disassembly information and material flows.

3. CSCW Proposals

Generally, the design of complex products involves several participants with different skills and responsibilities. For many years, software did not take this aspect into account, so that the project leader had to assume unassisted the management of the design process and its multi-user aspect. CSCW (computer supported cooperative work) [2] research proposes a new type of software, called groupware, which is an interactive multi-participant application allowing participants to carry out a "joint" task working from their own workstations. It is now a question of managing not only the man-machine interface but also the man-man interface mediated by the machine. After a short introduction to cooperative work, we extract from the analysis of cooperative design four modes of cooperation and we discuss architectural aspects of groupware.

The relationship between the participants can be considered from various points of view. Ellis et al. [2] proposed a matrix which classifies the nature of cooperation in regard to synchronous or asynchronous, local or remote aspects of cooperation. Physical distances can vary in length (inside a building, local to a city, regional, national, continental, intercontinental). This classification has been extended later, introducing awareness of cooperation, foreseeability or unpredictability of collaboration and location. The possibility of bringing together geographically distant people is an important contribution of groupware. It is also possible to collaborate without imposing simultaneous work. One notes that in all cases, it is a question of managing, via the information processing system, the participation of several people who may be present only virtually (in space and/or time). The first aim of groupware is thus to propose a support for the abolition of space and time distances. Moreover, knowing and managing the interventions of the multiple participants appears necessary. In fact, the participants constitute a work group that has to be organized with respect to working conditions, time and location. The organization can lead to the definition of different roles, sub-groups and phases of project work.
The success of cooperative work can be measured by the way in which the groupware is able to create and support good group dynamics, which contributes to making the virtuality of participants' presence disappear. The project must be able to proceed at least as naturally as in collocation and without the data-processing support. It must even take advantage from an organization of more effective work based on the new possibilities offered by information processing. The technological devices used should not interfere with the work or the group dynamics needed for project accomplishment. When designing cooperative systems, it is thus necessary to be aware that the usability aspect, whose aim is to validate the environment suggested, is at least as significant as the engineering aspect.

3.1 Cooperation aspects

In-depth analysis of cooperation reveals several dimensions which must be examined i.e. new support of production, conversation, communication, coordination between participants.

The **production** dimension and more precisely co-production aims at indicating the degree of cooperation. Thus, it implies:

- Structuring and managing production data.
- Identifying shared data and their structure (complexity, dynamic characteristics).
- Modeling the support of work activities (product model upgradeability), object attributes and methods.
- Identifying sharing characteristics (single and permanent owner or dynamic change of owner, delegation).

The purpose of the **coordination** aspect is to describe the organization of the work process coordination and the workflow (flows of works) to ensure a certain dynamics of operation (in push or pull approaches). Thus, it implies:

- Identifying the activities (tasks) to be performed, the work phases and their temporal organization (concurrent operations, co-interaction, etc), modeling of the work process.
- Identifying the workflow type: static (identified and stable), dynamic (opportunist and evolutionary).
- Identifying the functional roles of participants.
- Choosing cooperation modes (see below).

The purpose of the **conversation - communication** aspect is to define communication media in order to allow exchanges between participants concerning their work. Thus, it implies :

- Choosing the conversation mode: synchronous or asynchronous conversation, in text, voice, multimedia manner and tools: textual chat, audio or videoconferencing, etc.
- Choosing the conversation protocols: point-to-point or multicast, with moderator or free group conversations, etc.

3.2 Cooperative design

These general problems concerning cooperative work are perfectly applicable to the design activity. Indeed, it is possible to build various scenarios of collaboration between designers and to model the cooperative design. This one is justified by the fact that, nowadays, the design of complex products requires very thorough multi-field competencies, so that nobody is able to accomplish this task alone. Consequently, the problem is divided into several subsets linked by relationships and constraints. These studies are conducted by several participants whose common task is to successfully produce a global product consisting of several components, parts and subsystems. The subsets can correspond to different techniques and trades. However, their interactions are sufficiently strong so that one's choices react on the other choices.

The organization of work in cooperative design gave place to the proposal for various models of work. In the CIM-ONE project presented by David et al [3], several models were proposed with, for each one, descriptions of the result and activity, schedule of conditions constituting the design contract to be completed defining the rights and the duties of each participant, to obtain a new version of the object under construction. A similar modeling was used in the IPDES ESPRIT project, presented by Brun et al. [4], with the concept of design contract and the transformation of this contract into sub-contracts.

Thanks to these studies, we can assert that a group work can be organized according to two extreme modes, namely subcontracting and co-contracting. The effective working method is often obtained by the application of these two basic modes to different parts of work to be carried out. In both cases, the project leader divides work up into parts and describes it in the form of aims and constraints. It provides each participant with the starting subset and the context. Each participant works on his subset by respecting the objectives and the constraints imposed, then provides the project leader with the result of his activity. The project leader takes delivery of the subsets, controls and carries out the merge. In subcontracting mode, all the constraints must be perfectly defined so that each participant can work independently Once the collective context is acquired, the tools used are very similar to mono-user tools. In co-contracting mode, interactions between participants take place extensively during the entire process. The work to be accomplished is less precisely defined and horizontal collaboration is needed to detect and solve progressively the problems that arise. More collaborative tools are then needed, especially in the concurrent engineering context.

The project must be placed under the responsibility of a project leader. The hierarchical position of a participant within a working session is associated with his role (rights and duties) in this session. Except for the project leader who can be compared to a prescriber, we can reveal a certain number of generic roles such as: performer, appraiser, valuator, consolidator and adviser (expert). For each activity, each individual has at least one role resulting from these generic roles. Within the design project framework, the project leader has the role of manager. He is thus responsible for the coherence and progress of the project. He monitors the project while arbitrating the potential conflicts between participants, and validates the subsets by making sure that they are ready to carry out exactly the tasks defined by the contracts. If such is the case, he approves the work results and places them in a read-only reference environment. He thus guarantees the unicity of the group work versions. Each participant can nevertheless file, in a private environment, a version of his work other than the official version managed by the project leader.

3.3 Cooperation modes

From the previous analysis of cooperative design activities, we are able to define four cooperation modes that can be applied to other industrial world cooperative activities. These modes, that correspond to finer granularities of interaction, are :

- **asynchronous cooperation**: the various participants interact in the project by exchanging data and working when they can (without co-temporality).
- **in session cooperation**: the various participants work at the same time, but independently. They can communicate (in co-temporality), but cannot visually share the objects of their discussions.
- **in meeting cooperation**: clearly identified participants work and communicate in co-temporality, sharing the objects on which they work and discuss. They have identified roles in relation to the goal of the meeting and their competencies/skills. Their interventions are controlled and allocated by a metaphor of the type “to give the floor to”.
- **close cooperation**: the participants can work, communicate and interact in real time on a subset of shared objects in close collaboration with several other participants . The consequences of their interventions are directly visible to all participants.

At company level, asynchronous cooperation corresponds to the autonomous working method. Each participant works alone and submits reports/results to the relevant correspondents who will conduct their analysis later. To do this, each person has at his disposal communication tools that may be computerised such as electronic mail or the agenda for project meeting management. The main aim of asynchronous cooperation is to propose to the project members the tools that support a traditional multi-participant project accomplishment.

The concept of session corresponds to the act of presence within the company and can be brought closer to mandatory presence. Thus, at each moment, the participants know the list of their potential correspondents. In the case of obligation, this list is defined by the project leader. The telephone is a standard tool for “in session cooperation”. Indeed, the telephone call is established only if the correspondents are all present simultaneously. However, data processing makes it possible to consider far more powerful tools which make it possible to abolish more clearly the space barriers separating the various project participants (chat, videoconferencing, etc.). The main purpose of the “in session cooperation” is to reduce interaction delays between project members.

The concept of “in-meeting cooperation” corresponds exactly to that which exists within a usual project management. In fact, it is planned and it defines specific roles for each participant. Moreover, it relates to a clearly identified objective and working materials (documents, drawings, models, etc.) which must be shared by all the meeting members. Contrary to asynchronous cooperation, the “in meeting cooperation” requires active participation (e.g. immediate answer to the questions). The main aim of this mode is to increase coordination between the project members.

Close cooperation allows maximum interaction between project participants in a coherent virtual world simulating reality. It likens the concept of cooperative work to a unique worktable. It is especially within this situation that one can imagine new forms of cooperation. The power of this cooperation relies on the liberty of action it introduces, which generates the ability to act precisely and simultaneously on objects in a total virtual world. The main aim of close cooperation is to increase co-production of the various project members.

In-meeting and close cooperation requires visual continuity on the design space known as WYSIWIS (What You See Is What I See). This technique enables several users to share the same view on the objects in this design space. When views differ to a greater or lesser degree, one speaks about relaxed WYSIWIS. In the last three synchronous cooperation modes, control management and maintenance of coherence are of primary importance. On the ergonomic level, they have an impact on the reactivity of the objects handled, i.e. on the nature and the quality of the visual returns (feedback). Concurrency control, which ensures data coherence, can be explicit or implicit (transparent for the user). In the explicit context, control on the operations can be implemented by a protocol of the “round robin” type managed by a specific participant. If control relates to the data, it can lead to a segmentation of these data. In the implicit context, it can be expressed on a data-processing level by the use of more or less fine locking mechanisms, and in regard to social aspects by auto-regulation of work group dynamics.

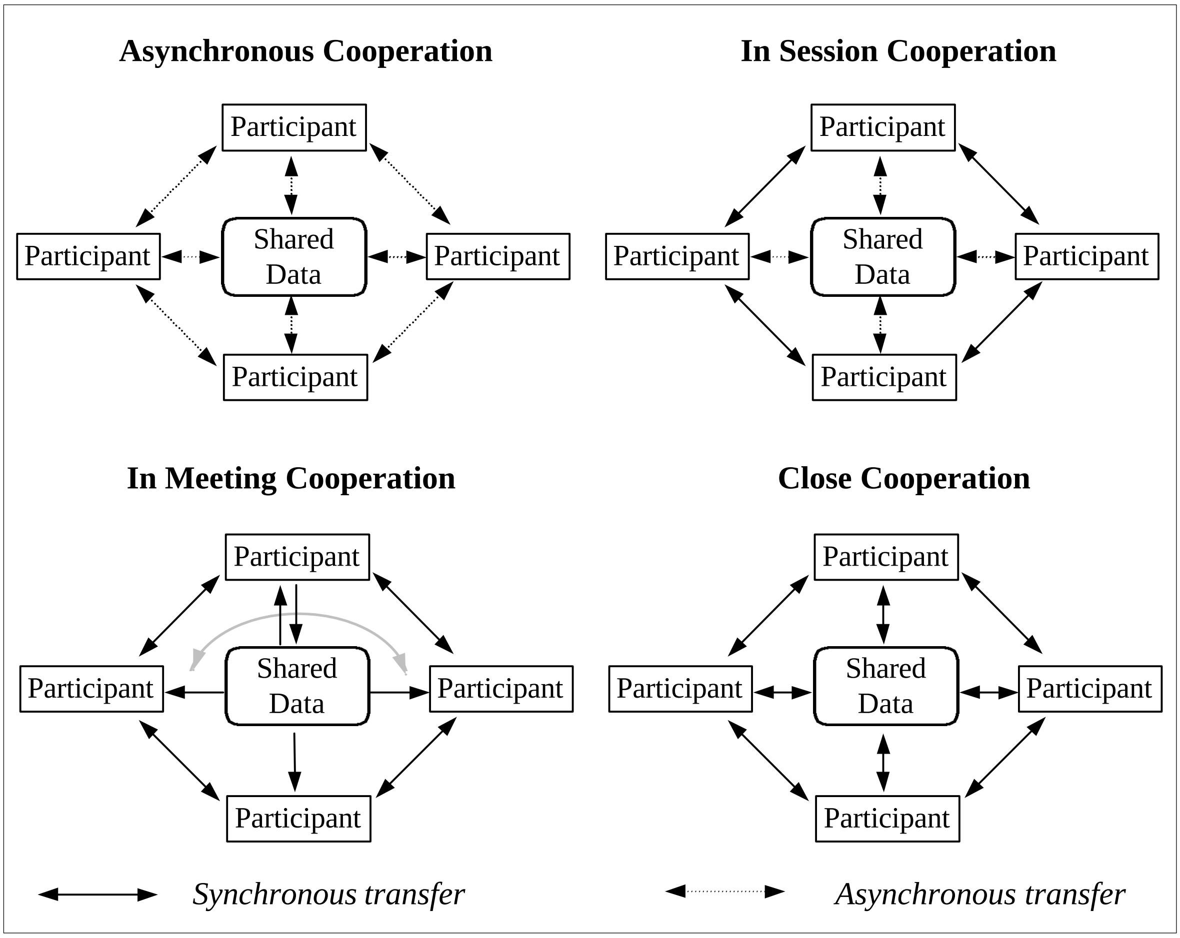

Fig. 2: Communication principles in each cooperation mode.

3.4 AMF-based ARCHITECTURE

An appropriate architectural model should fulfil three main objectives. Firstly, it organises software structure to improve implementation, portability and maintenance. Secondly, it helps identify the functional components, which is essential during the analysis and design process. Its third role is to help further understanding of a complex system, not only for designers, but also for end-users. The architectural model is one of the three key elements needed to achieve efficient and good developments: methods - models - tools.

AMF (French acronym for Multi-Facets Agent) is an architectural model for interactive software, which fulfils all these objectives [5] [6]. AMF is a generic and flexible model that can be used with design and implementation tools. It includes a graphical formalism that expresses the structures of software, and a run-time model that allows dynamic control of interactions.

The current trend in software engineering is to build frameworks and design patterns proposed by Gamma et al. [7]. Frameworks help developers share architectural principles and patterns explain new reusable configurations and behaviours. These two concepts help new developers avoid traps and pitfalls traditionally learned only by costly experience. The AMF model was initially developed for interactive software, with reactive agent paradigm. Its application in the field of groupware by Tarpin-Bernard [8] was a success. We are currently applying it to different application fields, also taking into account cognitive agent paradigm.

Architectural models for groupware have to combine the knowledge of models developed for single-user applications and the constraints introduced by cooperative work. For many years the HCI community has been very interested in designing models for interactive software. One of the most important classes of models is the multi-agent one. These models organise an interactive system as a set of agents that collaborate to support the dialogue between men and computers. Most of these agents are based on three components (facets) mapped on the HCI paradigm: presentation to the user, functional kernel, and interaction control, see PAC [9] or MVC [10]. However, these models present two main drawbacks:

1. They define very large facets which mix different thematic functions (e.g. a huge "abstraction" or "model" facet);
2. They do not provide powerful mechanisms to express interaction control (e.g. the "control" facet of PAC must be built from scratch by designers).

To solve the first drawback, AMF organises each agent in an appropriate number of facets. Naturally, among these facets we retrieve two classical components: Presentation and Abstraction (a lighter facet than the PAC one). The other facets can come either from a finer split of *control* components, from identification of new characteristics of agents (e.g.: user model management), or from duplication of classical facets (several *presentation* facets corresponding to different views). For instance, we have identified useful extra facets: *evaluation* (capture of user actions), *help* (contextual and on-line help), *user model* (information for adaptive interface), etc. To solve the second drawback, AMF expresses interaction control with two kinds of components:

1. Each facet presents several *communication ports* (allowing input, output or both) which can be seen as interfaces of real object methods. These ports avoid having a permanent binding between a function (a service) and its implementation. Moreover, it allows creation of the body of the functions in heterogeneous languages.
2. The *Control* component is a part of the agent defined by entities called *control*

administrators which have three roles:

- To *connect*, managing logical relationships between the communication ports (sources and targets) that are connected to it;
- To *translate*, converting the messages which come from the source ports into understandable messages for target ports;
- To express *behavior*, and so control strategies, using different rules of activation between a source port (A) and a target port (B). We have identified several administrators, such as simple (if A then B), sequence (if A_1, next A_2, next A_n then B), conjunctive (if A_1 and A_2 then B), etc.

These concepts are very similar to the *listener* and *adapter* concepts of *Java Beans* (in fact, the *Java* implementation of AMF uses them). However, AMF relies on a complex engine so that programmers can use predefined components, such as standard administrators, which are real objects and not only *interfaces*. Moreover, AMF provides a graphical formalism that expresses the control relationships between ports. In the following paragraphs, we will use this formalism to explain our cooperative mechanisms.

When a facet needs to trigger a distant service, it activates its corresponding output port. This port prepares a message and sends it to its associated daemon. Then, the control facet of the owner agent activates all the control administrators, which are connected to this source port. If this port is exported (connected to other agents), the activation is recursively transmitted to the parent agents. Then, each relevant administrator considers its activation conditions (see behavioral role). If these conditions are validated, the message is translated and sent to all the target ports. The activation of these ports runs their associated daemons.

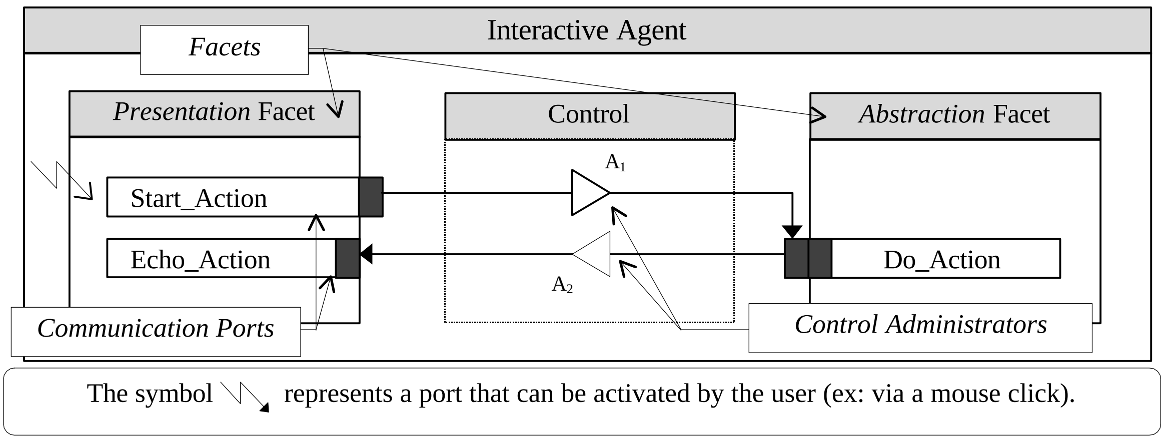

Fig. 3: An interaction on a single-user agent modeled with AMF

AMF proposes a powerful graphical formalism, which helps further the understanding of complex systems. Figure 3 shows the modeling of an elementary interaction. As a rule, two administrators manage the relationships between an action starting from the *Presentation* facet and the associated command defined in the *Abstraction* facet. In another situation, A_1 could be replaced by an iterative administrator if several activations are necessary to run the action.

When analyzing distributed system studies, we found the original concept of fragmented objects [11]. The methods and data of a fragmented object are distributed on the network and "transparent" mechanisms let it look like a classical object in a single computer. Applying fragmentation to AMF model offers an interesting approach for modeling CSCW applications. Indeed, their facets define a natural boundary for fragmentation. Thus, we can

study the distribution of the facets in the network. According to the required architecture, we can distribute *presentation*, *control* or *abstraction* facets, as well as of course the other specific facets. We called this approach the fragmented AMF-C model [12].
Figure 4 presents a centralized architecture with three shared agents manipulated by two users. Each agent is defined by four facets: the *abstraction* and *control* facets, and two *presentation* facets corresponding to specific views of each user. In this context, each *presentation* facet can be adapted to the role of each user ($P_{Ai} \neq P_{Bi}$). It is the *control* facet which is in charge of propagation of input/output events from or to the different facets, and especially between multiple *presentation* facets.

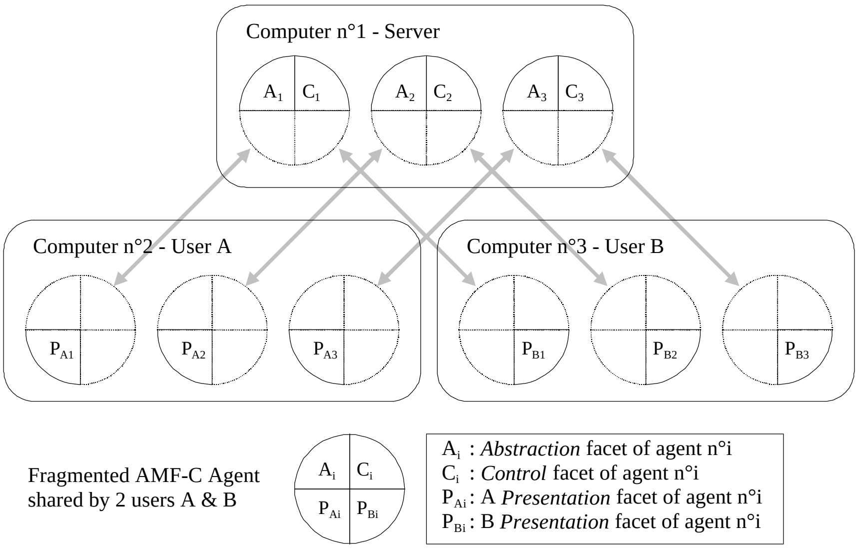

Fig. 4. An AMF-C fragmentation : centralised version.

If we try to model an elementary interaction (e.g.: a button triggers an action on an agent), we can consider a situation in which a first user is responsible for the agent, whereas a second user can just interact with its presentation. In this case, we can imagine that the agent is mainly located on the first user's workstation. To assume concurrency control and maintain the consistency of the shared agent, it is necessary to define new types of administrators. In the example given on figure 5, we have built a lock administrator that filters the access to the agent.

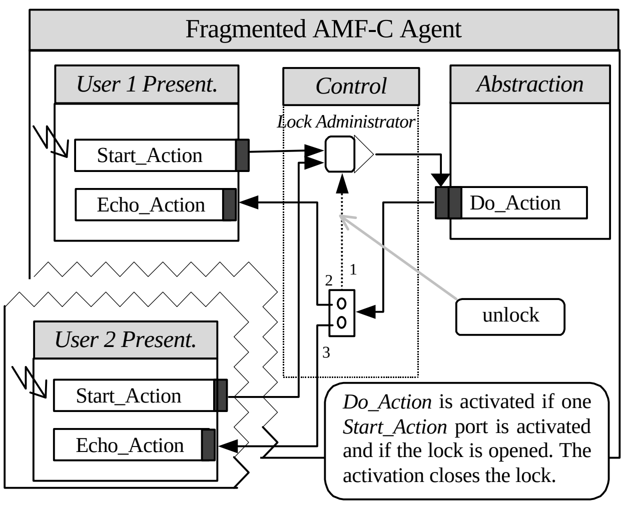

Fig. 5: An example of basic interaction on a fragmented AMF-C agent.

The dynamic property of AMF-C agents allows adaptation of each agent to be customized to the current user's role. Indeed, the number and the form of facets is not static, any change of role can lead to the substitution of a facet, and especially a *presentation* one.
The fragmented AMF-C framework is ideally suited to represent hybrid architecture in which some facets are centralized whereas others are replicated. Moreover, the use of a distributed object-oriented language can really simplify the implementation of such a model. Indeed, in the current implementation of AMF, with Java, each facet is an object. Activation of a communication port leads to invocation of a method for these objects. In a distributed context, this corresponds to a remote method invocation as defined in CORBA or Java-RMI.

4. A concrete application

The sustainable product life-cycle is the main objective of the global goal of our project, whose acronym in French is <<RESTER PROPRE>> (which means “remain clean” in English). This project relates to the study, organisation and evaluation of the transformation processes adapted to “noble” recycling, i.e. reuse, repairing, disassembling and dismantling of used products. The complexity of this problem requires a strong interdisciplinary approach. Our consortium (LAG/INPG Grenoble - ERIHST/UPMF Grenoble - ICTT/ECL Lyon), covers competencies on automatic control, information processing, human and sociological factors of working conditions and ergonomics. The main aims of this project are:

- Study of recycling approaches,
- Design of automatic, manual and semi-automatic disassembly cells,
- Design of a recycling platform (REX)
- Design of a network of recycling platforms and study logistic aspects
- Elaboration of information system for collection, and “just in time” distribution to different cells using this information
- Elaboration of a design stage support for cooperative engineering of sustainable products
- Design of multi-agent architecture, able to manage design, manufacturing and recycling activities in a distributed and cooperative manner.

The consortium work is based on a real recycling example, proposed by a non profit making association called ENVIE, whose purpose is professional reinsertion. The first recycling platform that we studied was inspired by the organization of an existing recycling unit of ENVIE, who is working on the recycling of "white" goods (refrigerators, washing machines, and cookers).

From the cooperation point of view, three main stages are taken into account: **Design, Manufacturing** and **Recycling**. For the design stage we propose a distributed cooperative environment which is able to support a concurrent engineering approach with the virtual platform metaphor with four cooperation scenarios (two asynchronous: off-line and forced presence, and two synchronous: in-meeting and closed collaboration, as described in earlier). The manufacturing stage is concerned with the finalized information derived from the design stage. We did not analyze this stage in detail, currently we are only concerned with appropriate dissemination of design and management information to manufacturing infrastructure. The recycling stage has been extensively examined in our project, with the study and design of a generic disassembly platform called REX implementing the noble recycling process.

4.1 REX platform

The recycling platform REX described in figure 6 allows manual, semi-automatic and automatic disassembly of products reaching their end-of-life phase. These products are collected upon request within collect centers which are responsible for sending them to the appropriate platform.

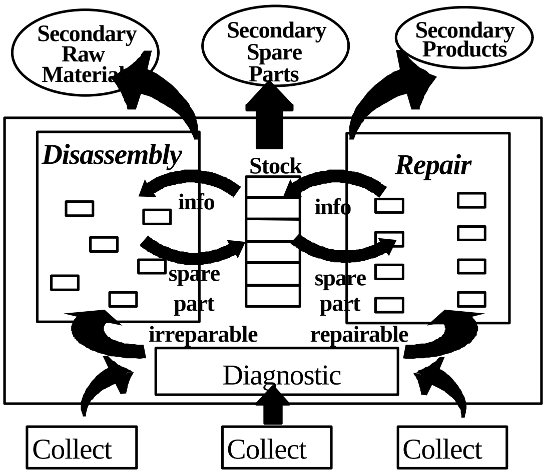

Fig. 6: Recycling platform REX.

The incoming products are analyzed in a diagnosis center whose main task is to decide whether the product is repairable or not. If it is irreparable, then the decision is made as to which spare parts are to be extracted. This information is used by the decision system of the disassembly workshop in order to determine the most appropriate disassembly sequences. We defined three categories of disassembling cells in the workshop: fully automated, semi automated and manual, depending on the complexity of the disassembly activities. In the workshop, end-of-life products are disassembled by extracting spare parts, (reused for repair of other products, component recycling, and secondary spare parts) and by obtaining raw material of high purity (material recycling, secondary raw materials). The repair workshop renews the incoming products by replacing broken components with used spare parts provided by the disassembly workshop (product recycling, secondary products). Finally spare parts and renewed appliances are stored into stocks..

At a higher level, we can consider a network of cooperating platforms, each specialized in specific types of products. The recycling process is thus dispatched among these platforms, depending on the nature of the products or parts to be disassembled or recovered and on the availability of each platform in terms of free cells and spare parts.

4.1.1 Information sharing on the platform

At each step of the recycling process, specific and detailed information about the product is needed. For example, the diagnosis phase requires complete knowledge of the functional structure of a product in order to determine which components to test and decide which parts are recoverable and can be extracted. As for the workshop decision system, it needs detailed information about a product structure, its geometry and topology, in order to generate the best sequences for disassembly operations.

As we can see, there are many views for a same product. The decision was thus made to use a Product Data Management System (PDMS) for management of product information. PDM Systems are of primary importance in CSCW as they allow grouping, referencing and sharing of all product information among its different life-cycle stages (Design,

Manufacturing, Maintenance and Recycling). In particular, the PDM System can provide all the necessary information to carry out the recycling process (Disassembly sequences, product components, technical and functional information).
However, this is not the only purpose of the PDMS. Indeed, in order to achieve the "sustainable design" objective, it is necessary to convert the traditional "waterfall" life-cycle of a product into a cyclic life-cycle (§2.2)(fig. 1). In figure 1, two flows are represented: the physical flow, which is concretized by the noble recycling approach (reuse of components), and the information flow, which consists of a feedback of information from all the product life-cycle stages to the design stage. In particular, all information resulting from the recycling stage should be taken into account when designing new "green" products. The PDMS is thus seen as the main tool for referencing and managing all this information and linking it to the design stage.

4.1.2 Managing the process

The REX platform and the platform network form a complex system. They group in the same (virtual) structure a variety of competencies: logistics, workshops, warehouses, etc. each responsible for separate tasks in the process during the accomplishment of which information, documents and objects are handled, exchanged and modified by different actors. Since the platform is also submitted to productivity and profitability requirements, the success of the recycling process in an acceptable period of time is a major concern.
In order to increase recycling process efficiency, the platform actors' work needs to be coordinated, and appropriate information should be made available just in time at each stage. This need is emphasized by the fact that the recycling process can be distributed over a network, thus generating communication and coordination problems between the remote actors and activities.
Workflow management technology provides solutions for satisfying these requirements. We use a Workflow tool for managing the recycling process, providing information and supporting the co-ordination and synchronization of the cooperating actors in a transparent way. The Workflow tool also allows us to combine remote actors and processes in a unique virtual structure. At operational level, it is possible to use synchronous and asynchronous groupware tools to support remote or local cooperative work

4.2 AMF-C in <<RESTER PROPRE>>

The aim of concurrent engineering is to improve time, cost, quality and technical performance by simultaneous and integrated design of products and their manufacturing processes as well as logistics support. Our “design for disassembly” approach requires various complementary competencies to obtain greater interactivity and reactivity. Without information technology tools one must co-locate all the actors to obtain such interactivity. Co-localization is today one of the keys to the success of the simultaneous model. However, information technology can contribute in an operational way to generalizing the localized platform approach. Indeed, Computer Supported Cooperative Work is an emerging field of information technology, usable in many application fields. Its purpose is to go beyond the individual man-machine relationship proposed up to now and to enable groups of people to cooperate.

In the <<RESTER PROPRE>> project we study the adaptation of AMF-C to the management of this large Design – Manufacturing – Recycling System for sustainable products. This system is a distributed interactive cooperative system and we are currently extending the AMF-C model to be able to take into account reactive and cognitive behaviors of agents. By modeling these behaviors in new facets, which must be stereotyped as design patterns, we could reuse them in new developments. We thus propose an architectural framework and a series of design patterns which formalize basic behaviors. This framework must be compatible with the CORBA framework in order to take into account distribution and cooperation requirements. The development is carried out in Java in the Internet environment.

As we mentioned earlier, REX is semi automated, which means that human and automated actors alike are involved in the recycling process. Although the process (main) tasks are well known, the nature of the actors performing them is, on the other hand, not necessary defined beforehand. Indeed most of the time, this nature cannot be known before run time as it depends on many conditions, such as for example the complexity of the disassembly tasks. Consequently, before the process starts, we can only allocate roles to the recycling tasks. Moreover, heterogeneous but complementary systems co-exist on the platform, which adds to the complexity of the cooperation problem among the platform entities.

As a solution (fig. 7), we propose allocating an AMF agent to each platform and each platform component (WMS, Diagnosis, Collect center, Workshop and Product). Thus, each entity is represented by a specialized agent that owns specific behavior and knowledge, controls internal life and manages exchanges and communication with and among other agents. Some specific facets have to be defined such as Workflow facet or Cell facet in order to encapsulate the specific management modes introduced by the workflow or the interaction with the cell (a robot with or without a human operator). This approach introduces a great deal of flexibility to the platform. Indeed, the problem is reduced to a problem of cooperation and communication among AMF agents only, independently from the underlying nature of the entity. Even the real product is associated with an agent, thus allowing the workflow agent to keep control of the product and to establish a history of its evolution on the platform.

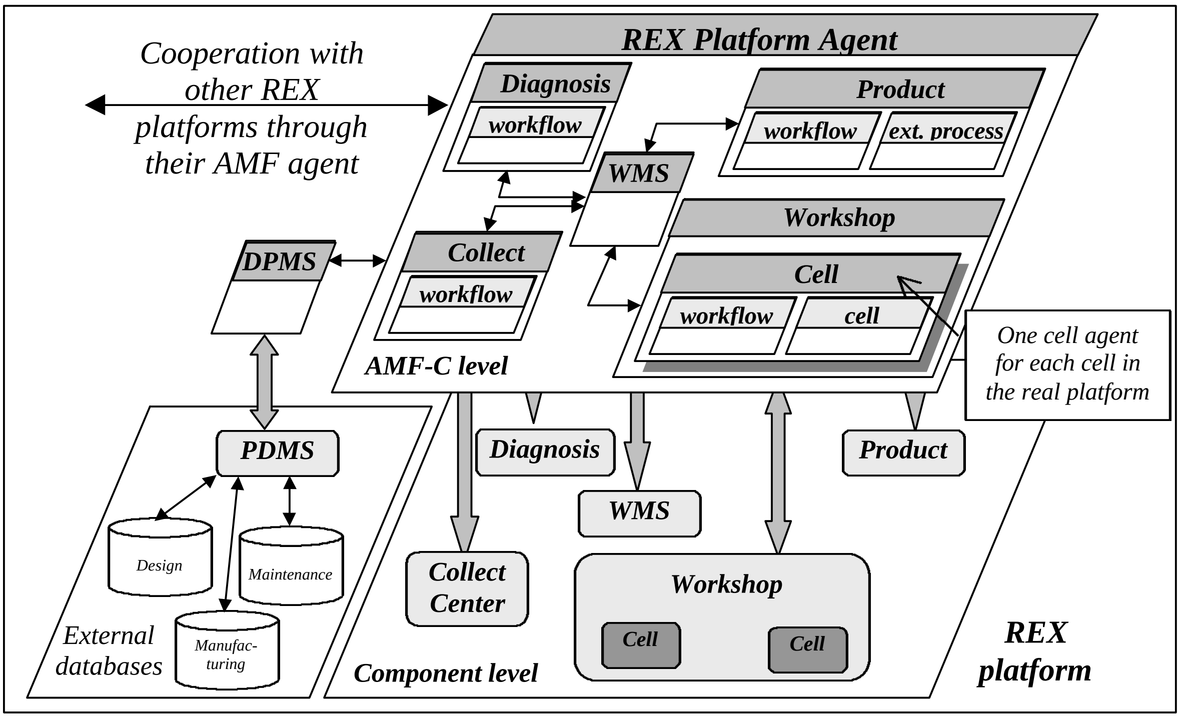

Fig. 7 : AMF-C agents in REX.

4.3 Other extensions

In view of the extreme complexity of the process, we also try to address two different problems, namely management of various degrees of automation and on-line learning of workers.

Total automation of recycling is easy. Some of the risks can be solved only by human intervention. Between fully manual recycling (which we studied with ENVIE) and fully automated recycling, various man-machine collaboration situations were identified by Skaf et al. [14]. All these situations can be supported by AMF agents, which are able to adapt their behaviors to them. Complete sequences can be elaborated in advance and stored in the PDM System, contextual sequences can be either stored, if they are commonly known, or calculated “on the fly”.

To allow the workers to receive information on disassembly processes not only during the learning sessions conducted by a platform supervisor, but also during the disassembling activity, we propose multimedia help or tutorials. Such help takes the form of an informational support to assist the worker in his choice of operation sequence. The help can become a tutorial if it also guides the worker during a sequence, and explains which tool can be used. This tutorial will be completed by explanations on identified disassembling problems and their solutions. A special AMF agent is in charge of this help-tutorial. If the user requires more help, he can contact an expert during an “in session cooperation” using synchronous communication tools.

5. Conclusion

In this paper we discussed cooperation theories and practices. We then applied this study to the sustainable industrial product life-cycle. We identified the information needs of sustainability of having access at each stage to all information). The design stage is more complicated because, in comparison to conventional design, the aim is to choose reversible assembling methods and to imagine the testability of the product, required during repairing and recycling. To take into account these new aspects, we propose the use of the cooperative design approach based on different cooperation modes and based on AMF architecture. In the recycling stage, access to design and manufacturing information is mandatory. This information can be used either for recycling process planning or for worker instruction. The AMF multi-agent model, based on frameworks and patterns, is an appropriate approach mainly through its adaptability. The PDM System is constantly accessible, workflow allows “just in time” information distribution through REX platforms.

The cooperative approach of design activity is the foundation for concurrent engineering, the on-line learning tool guarantees permanent worker information and cooperation allows virtual contact between workers and instructor. In this way, the generic recycling platform implementing the “noble recycling” concept for manufactured products integrated in a platform network can become a reality. All these aspects are progressively integrated into the <<RESTER PROPRE>> project to be validated. Only the cooperation aspect of the project has been described in this paper.

Acknowledgements

This project presents the results of the collaborative research program <<RESTER PROPRE>> financially supported in part by the French Rhone-Alpes region. Many thanks to our project partners from LAG- INPG and ERIHST – UPMFG with who we are working on this project.

References

[1] Hiessl H., Mayer-Krahmer F., Schön M. (1995). Auf dem Weg zu einer ökologischen Stoffwirtschaft. *Gaia – Ecological Perspectives in Science, Humanities and Economies*, Gaia 4 n°2.

[2] Ellis C.A., Gibbs S.J. & Rein G.L., (1991), Groupware : some issues and experiences, *Communication of ACM*, Vol 34. n°1, 39-58.

[3] David B.T., Marty C., Vandorpe D. & al. (1991), *Environnement informationnel pour le CIM: Projet CIM-ONE*, 23ème CIRP, Séminaire International sur les Systèmes de Production, Nancy 6-7 juin 1991

[4] Brun J-M. & al. (1993), *IPDES: an approach to the integration of design and manu-facturing*, in Realizing CIM's Industrial Potential, IOS Press p.190-200.

[5] Tarpin-Bernard F., David B.T. (1999) AMF : un modèle d'architecture multi-agents multi-facettes. *Techniques et Sciences Informatiques.* Hermès. Paris. Vol. 18. No. 5. Mai, 555-586.

[6] Tarpin-Bernard F. & David B.T. (1997b), "AMF a new design pattern for complex interactive software ?", *International HCI'97 Proceedings*, San Francisco, 24-29 August 1997, in Design of Computing Systems, 21 B, Eds Elsevier, 351-354.

[7] Gamma E., R. Helm, R. Johnson & J. Vlissides (1995), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, Mass. 1995.

[8] Tarpin-Bernard F., *Architectures logicielles pour le travail coopératif*, PhD Ecole Centrale de Lyon, France, 1997.

[9] Coutaz J. (1990) "Architecture Models for interactive software: Failures and Trends", in G. Cockton (eds.): *Engineering for Human-Computer Interaction*, Elsevier Sc. Publ., 137-153

[10] Krasner G.E., Pope S.T., (1988) "A cookbook for using the model-view controller user interface paradigm in Smalltalk-80", *Journal of Object-Oriented Programming*, 1(3), 26-49

[11] Gourhant Y., Makpangou M., Le Narzul JP., Shapiro M. (1994), Fragmented objects for Distributed Abstractions, in *Readings in Distributed Computing Systems*, Eds Casavant & Singhal, IEEE Computer Society Press, 170-186.

[12] Tarpin-Bernard F., B.T. David, P. Primet (1998) Frameworks and patterns for synchronous groupware : AMF-C approach, *7th IFIP Working Conference on Engineering for Human-Computer Interaction*, EHCI'98, Greece, september, 225-241

[13] David B., Poquet J., Saikali K., Boutros N., Skaf A., Vial I., Binder Z., Multi-Agent Architecture for Design-Manufacturing- Noble Recycling System for Sustainable Products. *IFAC 1st International Workshop on Multiagent Systems In Production - MAS'99.* Vienna, Austria. Décembre 1999

[14] Skaf A., H. Tian, B. David, B. Descotes-Genon, Z.Binder (1999) Actions and activities in Man-Machine Disassembling Systems, *HCP'99*, Brest, France