A version of this paper has been published in: *Journal of Applied Systems Studies*, 2, pp. 100-115, 2001.

Application of a Holonic Multi-Agent System for Cooperative Work to Administrative Processes[1]

Emmanuel ADAM, René MANDIAU, Christophe KOLSKI
LAMIH UMR CNRS 8530
Université de Valenciennes et du Hainaut Cambrésis
F - 59313 Valenciennes Cedex 9 - FRANCE
{emmanuel.adam, rene.mandiau, christophe.kolski}@univ-valenciennes.fr

Abstract
Tools resulting from the Computer Supported Cooperative Work (CSCW) field, which are often centralized, are increasingly designed to adhere to the system organization using a distributed architecture. However, generally these systems have difficulty in managing data coherency at the global level. In answer to this problem, we turned towards a particular organizational model: holonic systems. This concept has guided us in the specification of a Holonic Multi-Agent System (HOLOMAS), which offers a good compromise between the distribution of knowledge and control centralization. This HOLOMAS has been specified in a particular goal: assistance to actors of cooperative processes in a department of a large company.

Keywords: Holonic System, Multi-Agent System, Cooperative Work, Modeling, Specification.

1. Introduction

The boom in Internet technology and company networks has contributed towards completely changing a good number of habits which have been well-established in companies for several decades. In a context of globalization of the economy and of serious modifications in socio-economic structures, the technical and administrative processes which underlie the activities of a Company are, in particular, the subject of considerable revision. Documents on paper, exchanged from hand to hand are progressively being replaced by electronic documents transmitted automatically by machines without taking the human factors, such as the notion of the group (the individuals are isolated at their work post), the levels of responsibility or even human-machine cooperation into account. Admittedly, tools for aid in cooperative work have already been suggested, some with success, but they do not tackle the overall organization.

This fact formed the base problem of our work. Indeed, our research aims to set up an information management assistance system in the Patent Department of a large company. This department has been dealing with an increasing quantity of digital data over the past fifteen years using old but well-proven systems. Our aim is thus to facilitate data processing by the implementation of a system which respects the roles established and played by the actors in that system.

The solutions based on Multi-Agent Systems (MAS) are extremely promising for this and they have the advantage, because of their distributed nature, of offering knowledge sharing as well as an adequate response time [Jennings (1996)]. But even though the nature of the solution seems to have been found, it is now necessary to look for its optimal form, that is to say the architecture of the multi-agent system. An over-decentralized architecture brings about a problem in managing the coherence of the data processed. On the other hand, an over-centralized architecture poses the problem of reaction speed when an actor wishes to access a specific datum.

In order to solve this problem, hybrid architectures appeared (for example [Soulié (1998)], [Georgeff (1987)]). We have positioned ourselves in this research axis and in this paper we suggest a specific hybrid architecture: the holonic architecture. The aim of the paper is therefore to suggest the specifications for a HOLOnic Multi-Agent System (HOLOMAS) which will help the actors in a human-machine system with a workflow type operating procedure.

The first part will show the general characteristics of holonic systems [Koestler (1969)], which we judged to be relevant for the specification of a holonic multi-agent system. The second part will present an application of the holonic principle for the specification of a HOlonic Multi-Agent System for assistance with COoperative Work (HOMASCOW). The third part will present a modeling method for holonic systems operating with data flow. This method makes it possible to model the "social" operation of holonic agents. Finally, the fourth part will present our approach for the specification of a HOMASCOW. This specification was only possible after the analysis, modeling and simulation of the procedures present in the patent department in question.

2. MAS point of view of Holonic System

2.1 Characteristics of a Multi-Agent System

A Multi-Agent System can be defined as being a group of agents which interact between themselves directly or indirectly. It is extremely difficult to provide a common definition for the term agent [Flores-Mendez (1999)], in the same way, as it is still very difficult at the present time to give a common definition of what an Artificial Intelligence system is. The debates currently taking place on the distribution lists are the proof of this (the international discussion group on artificial intelligence is to be found on the comp.ai forum).

An agent may be defined as being an intelligent entity which is part of a multi-agent system. The space in which the system, and therefore the agent, evolves is called the world. The agent is capable of perceiving what surrounds it, that is to say its environment, and can modify it with a given aim and with a certain degree of autonomy. Generally speaking, the agent is not unique but is part of a multi-agent system, it must therefore have capacities for communication.

As regards the notion of intelligence, the following principle should be underlined: an entity is intelligent if it is capable of learning, that is to say of adapting its knowledge. An agent can therefore be defined as follows: an agent is an adaptive, rational and autonomous entity, capable of communication and action. It may also be adaptive and its adaptivity degree may vary from an agent type to another [Adam (2000C)].

An agent generally has acquaintances; these are agents with which it communicates or interacts. Each agent has elements of knowledge concerning its environment. These elements of knowledge are also called representations or beliefs. An agent has one or several objectives, which are also called goals or desires. According to the importance of the goals, the agent can be required to plan its actions.

In fact, agents may be classified as either reactive agents or cognitive agents. A reactive agent does not have internal representation of its environment, of itself and acts according to a stimulus/responses mechanism (in that case, the adaptivity is poor and just consists in a reaction to an event). On the other hand, a cognitive agent possesses representation of its environment, of itself and is able to plan its actions and to cooperate explicitly with others agents [Nwana (1996)].

J. Ferber suggests associating six functions to the agents: the representational function linked to the representation of knowledge; the organisational function linked to the planning of actions, and to the rational and adaptive nature of the agent; the conative function linked to needs and desires, to its autonomous nature; the interactive function linked to communications; the productive function linked to the various actions which may be undertaken by the agent and the conservative function linked to the protection and conservation of the agent or the system [Ferber (1995)]. These six functions have a variety of forms based around 5 dimensions (table 1): the 'personal' dimension (concerning the agent); the 'environmental' dimension; the 'social' dimension (concerning the other agents, that is to say, the agent's acquaintances); the 'relational' dimension (concerning the relationships between the agent and the world) and the ‘physical’ dimension (concerning the implementation of the functions).

Table 1. Functional analysis grid [Ferber (1995)], usable for describing an agent (see §4.1)

<table>
<tr><th></th><th>Personal</th><th>Environmental</th><th>Social</th><th>Relational</th><th>Physical</th></tr>
<tr><td>Representational</td><td colspan="5">← *about Knowledge* →</td></tr>
<tr><td>Organisational</td><td colspan="5">← *about Planning* →</td></tr>
<tr><td>Conative</td><td colspan="5">← *about Goals, needs* →</td></tr>
<tr><td>Interactive</td><td colspan="5">← *about Communications* →</td></tr>
<tr><td>Productive</td><td colspan="5">← *about Actions* →</td></tr>
<tr><td>Conservative</td><td colspan="5">← *about Protection* →</td></tr>
</table>

The functional analysis grid makes it possible to classify the various types of agent and multi-agent systems. According to the agent types, the functions are developed to a greater or lesser degree, or are even non-existent. For example, the assistant agents which are seen appearing in office work suites have a behavior pattern which is essentially a reaction to events [Arafa (1999)], [van Mulken (1999)]. On the other hand, the distributed expert systems have greater capacities for reflection. It will be interesting moreover to use this grid on the agents which we are going to propose in order to constitute the Holonic Multi-Agent System for Cooperative Work in administrative processes.

2.2 Presentation of holonic systems

Holonic systems were proposed by Arthur Koestler around 30 years ago [Koestler (1969)]. The underlying principle is the fact that, in real life, an entity must be considered both as a whole made up of other entities and as being part of a set. Koestler’s ideas have already been applied in various fields, notably in an international project named “Intelligent Manufacturing Systems” [Van Brussel (1996)], in order to form one of the models on which the factory of the future could be built. These ideas are also applied in the field of robotics [Arai (1997)], as well as in the field of cognitive psychology [Young (1995)]. The work of Siekmann should also be mentioned [Gerber (1999)] on the formal description of holonic agents and his proposals for application in the fields of transport planning, flexible production systems and Robocups (a Robocup is the simulation of a football tournament in which the teams are made up of robot or software agents).

A Holon is defined by Koestler as being a part of a whole or of a larger organization, rigorously meeting three conditions: to be stable, to have a capacity for autonomy and to be capable of cooperating:

- Stability means that a holon is able to cope and react when it is subjected to high demand or to major disturbances,
- Autonomy suggests that a holon is capable of managing itself when subjected to demands in order to achieve its own aims,
- The capacity to cooperate means that the holons are able to coexist with other holons or other layers of holons, and are capable of working on common aims and projects.

Here we can find at least two of the characteristics of the agents in a MAS sense: autonomy and cooperation [Nwana (1996)]. The third characteristic, the capacity to adapt itself to an environment is suggested by stability. A holon can therefore be seen as an agent whose stability is an essential point. However, if the holons are stable, they do not have to be rigid. Indeed, the stability of the whole system is more important than the stability of each of its parts. So, it is sometimes necessary that some holons be temporarily destabilized so that the whole system can take more long-term protection strategies.

A version of this paper has been published in: *Journal of Applied Systems Studies*, 2, pp. 100-115, 2001.

2.3 Proposition of a Holonic Multi-Agent System (HOLOMAS)

It is important to study the characteristics inherent to a holonic multi-agent system. In an annex to his book [Koestler (1969)], Arthur Koestler gives a series of grouped rules defining holonic systems, which he calls Open Hierarchical Systems (OHS). Here we propose to give an interpretation of these rules according to the multi-agent point of view.

We can retain the following principles from the rules:

- A holonic system possesses a tree structure. In fact it can be seen as a set of interwoven hierarchies (cf. figure 1).
- A holon is considered both as a part of the system and as a whole, a system made up of other holons.
- A holon obeys precise principles, but is able to adopt different strategies according to its need.
- The complex activities and behavior are situated at the top of the hierarchy, the “simple” and reactive acts are to be found at the base of the holarchy.
- The communications must follow the hierarchy and, according to the direction, must be filtered or detailed. In fact, in holonic organizations, messages are only possible between a holon and its responsible or between holons on a same layer. So, in such organizations, the communication are made by message exchanges, rather than by using of a blackboard (i.e. common databases or files).

A holonic multi-agent system must therefore have a hierarchical structure as well as having a flexible organization. That is to say that each agent has aims set by a manager, but the agent chooses the appropriate strategy itself. The agents must have capacities for autonomy and cooperation. They unite to form a whole and each one can be broken down into holonic agents, this is what is called the recursive breakdown of the problem [Gerber (1999)].

The holonic agents obey fixed rules, which makes them similar to reactive agents. But they also have their own strategies, which gives them certain similarities to cognitive agents. Holons are therefore cognitive agents which obey rules. These rules are set by the hierarchy. The agent chooses its own strategy according to its environment and according to the limits imposed on it, so as not to jeopardize the stability of the system. In fact, in a HOLOMAS, an agent has a more reactive behavior than the higher level agent and a more “cognitive” behavior than the lower level agent. The top of the holarchy is considered as being the decision center whereas the base is considered as being a reactive interface with the environment (cf. figure 1). In fact, the main difference between an agent and a holon is that a holon may be considered as a holonic system that may be broken down into other holons. But, each holon may be implemented with an agent. So, a holonic multi-agent system may be considered as a multi-agent system having a recursive behavior and having explicit definitions of its autonomy, cooperation and stability behavior.

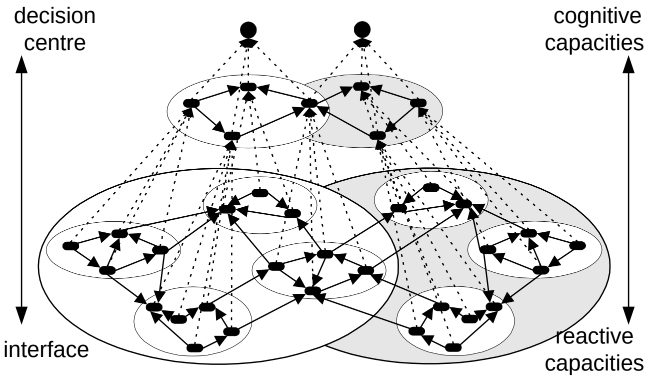

Figure 1. Holonic system made up of two hierarchies

The communication between holonic agents is carried out by the sending of messages and following the hierarchy. The messages are detailed when they are sent from the top towards the bottom of the hierarchy, and are synthesized when they are being transmitted from the

A version of this paper has been published in: *Journal of Applied Systems Studies*, 2, pp. 100-115, 2001.

bottom of the hierarchy towards the top. The communications with the outside of the system go through the base of the system first (cf. figure 2a and 2b).

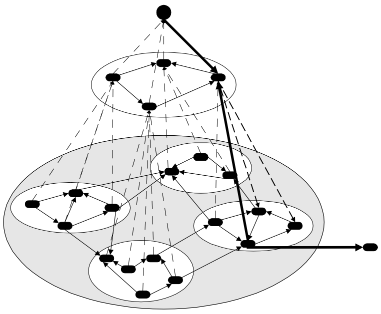

Figure 2a. Routing for a message from the top towards the exterior

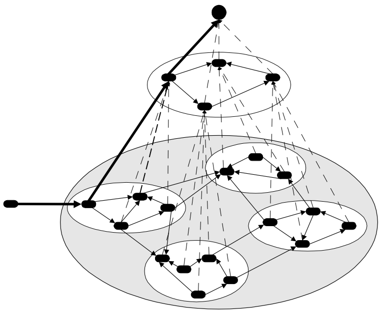

Figure 2b. Routing for a message from the exterior to the top

The previous rules, although they are general, give a concise view of the organization of a holonic multi-agent system. The interest of these rules is that they offer a control mechanism between reactive parts and cognitive parts. Research has already been carried out on this subject, such as [Chaib-draa (1996)] which suggests a control mechanism between the reactive parts, parts based on heuristics, and the cognitive parts (based on its knowledge). But this control only involves one same agent and not the whole organization. We can also mention the GEAMAS platform [Soulié (1998)] which was devoted to the modeling and simulation of complex systems according to a three-level multi-agent organization: a society level (macro-agent) which could be broken down into a secondary level made up of cognitive agents (medium-agents). This level was itself broken down into reactive agents (micro-agents). It is interesting to see that this architecture, the result of work on emergence and auto-organization, is closely akin to some concepts of the holonic approach (abstraction/specification division).

Our aim is the specification of a multi-agent organization providing assistance to the actors in a patent department. This organization must be fixed (which does not imply rigidity) in order to be able to meet the user demands as quickly as possible. This is why we have used the social rules defined in the holonic concept in order to simplify and accelerate the specification of a multi-agent society (in the [Mandiau (1999)] sense). This holonic concept is especially useable and useful in structured and cooperative fields [Gerber (1999)].

3. A Holonic Multi-Agent System for Cooperative Work (HOMASCOW)

Our aim is the specification of an assistance system for the actors in an administrative system (an organization having a hierarchical structure and whose actors correspond to specific roles). The administrative organization procedures are essentially composed of the management of documents, which brings about a high degree of cooperation between the actors (we define cooperation as being the communication, coordination and collaboration -including co-decision- of actors for the achievement of a common objective). It is possible, moreover, to put forward the hypothesis that there is now, in any administrative system, at least one workstation (PC or other) per office or area.

The assistance system must be distributed around actors in order to best advise them and to facilitate cooperation by making the group notion apparent. The aim is to increase the autonomy of the actors by appropriate assistance, along with the cooperation by an awareness of the common nature of the activities, and the stability of the actors by monitoring actions. The objective is in fact to obtain more "holonic" administrative organizations, a condition necessary for viability and durability (in the sense given by A. Koestler). We have called this Holonic Multi-Agent System for Cooperative Work "HOMASCOW" [Adam (1999)] (a HOMASCOW being dedicated to an administrative procedure which implies *k* actors on the *n* of the organization).

3.1 Assistance rules for cooperative work

In order to maintain coherence in a distributed system, much research has shown that the use of at least two types of rules is necessary: personal rules at the level of each module and social rules defining the interactions possible between modules. For example [Boissier (1996)] suggests architecture for a social and individual control of a system of agents intended for pattern recognition. In our case, the HOMASCOW has to have three levels of assistance rules:

- In order to ensure the coherence of the functioning of an administrative organization procedure, the HOMASCOW must have a first level made up of general rules describing the circulation of data between its actors (which corresponds to the social rules).
- Then, more precisely, at the level of each office, the HOMASCOW must have a set of local rules, sorts of agreements between people situated in the same space (for example: if a fax appears, the first person free deals with the fax).
- At the last level, the HOMASCOW must have the personal behavior rules of the actors associated to the workstations (personal rules). These rules are not fixed, but represent the actions generally undertaken by the actors in well-defined situations.

3.2 Structure of the organization

According to the properties of the given holonic systems, we have to find, at the base, the interface between the HOMASCOW and its environment made up of actors and documents; the first layer will therefore contain holonic agents which are responsible for interaction with the user and for document management. In addition, in order to facilitate and reinforce communications between workstations, this first layer will also contain holonic agents, which are responsible for the sending and reception of electronic messages (from now on, the symbol ◦△ will indicate an executing agent located at the base of the HOMASCOW, which has a pyramid shaped architecture). These agents have a behavior rather reactive. In fact, the messages that they receive are automatically processed and transmitted without decision making.

Each actor must be assisted according to the roles it plays in the various procedures in which it is involved. This is why the second layer of holonic agents will be made up of agents called "station managers", each one assigned to an actor within the framework of a given procedure. These agents will contain personal and local procedure operating rules (these station managers will be represented by the symbol △).

Finally, in order to maintain the coherence of the assistance given in the different procedures, the third and final layer will be made up of holonic agents which are responsible for procedures. Each holonic agent responsible for a procedure will be associated to the human actor responsible for that procedure and will contain general procedure operating rules (a procedure manager, located on the third level, will be represented by the symbol △). Figure 3 represents an example of the general architecture of a HOMASCOW. In this figure, associated to the actor responsible for the procedure, we find a sub-HOMASCOW which is responsible for the procedure and contains the global rules for data circulation; the sub-HOMASCOWs associated to the actors contain the local and personal rules associated to the actors in question.

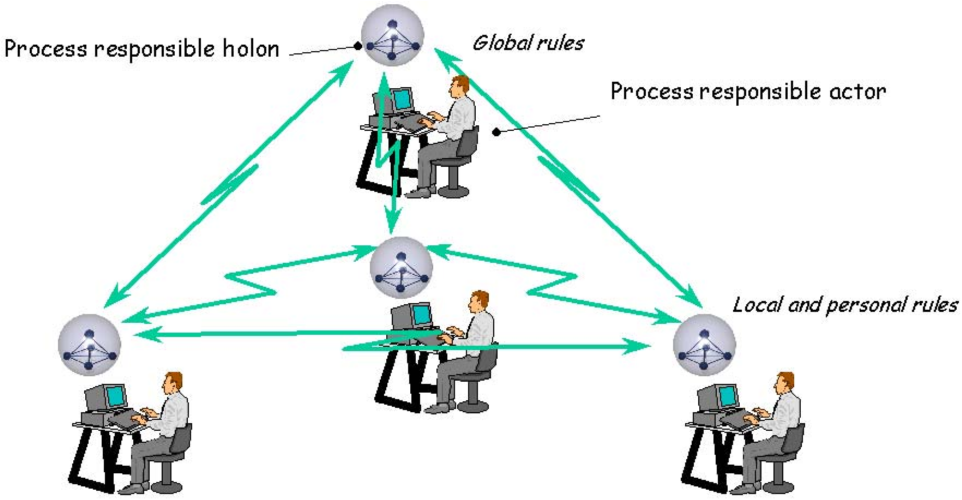

Figure 3. Architecture of a HOMASCOW

Each manager agent (the layer manager as well as the procedure manager), must help an actor. It must therefore be able to interact with the actor, help him to manage the documents, and communicate with the other manager (that is to say, send and receive information).
Each manager agent therefore has under its control four responsible executing agents which it manages interactions. We call these units *sub-homascows* (cf. figure 4).

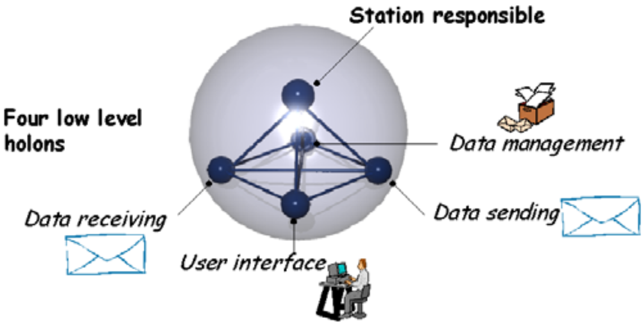

Figure 4. Architecture of a sub-HOMASCOW

The holonic model therefore makes it possible to define a multi-agent architecture which is particularly well-adapted to complex administrative organizations, that is to say, organizations in which the actors have well-defined roles and responsibilities. This holonic multi-agent system is both reactive at the interface lowest level and more cognitive at workplace level and especially at the process responsible level. So it may either react rapidly to user solicitations or, if needed, call upon all the station responsible and the process responsible. For example, a simple need (concerning the presence of a document) may be treated in local, while a more complex need (concerning the location of a the document) may come within the competence of the process responsible holon which is able to give a information in coherence with the current state of the process. When a executing agent receives a message, it may forward it to its responsible or to another executing agent, according to the nature of the message and the process type (a example of communication is given in §4.2). The following part, using this definition of HOMASCOW, suggests a model and a specification within the framework of a real application case.

4. Specifications of the HOMASCOW

Although the definition of the HOMASCOW structure has been facilitated by the use of holonic principles, the modeling of the system organization and the characterization of the functionality of the agents remain problematic. Indeed, the research published on this subject is mainly theoretical. Only a few recent research projects allow organization modeling with an application goal. In connection with this, we should mention CASSIOPEE [Collinot (1998)], a method which makes it possible to model a multi-agent system and its organization. But this method is mainly devoted to the field of collective robotics, more than to the representation of cooperation in a holonic organization in which the agents have varying degrees of responsibility. We should also mention the TELETRUCK approach [Burckert (1998)] on transportation scheduling by a holonic multi-agent system. In this approach, the holonic system represents a real system composed of trucks. Each truck is represented by a holon composed itself of holon associated to components. This work does not propose a modeling and design method of holonic system but describes principles of communication and cooperation developed in the prototype.

Here, we propose modeling and specification in two stages: the first stage concerns the individual functioning of each type of holonic agent; the second concerns the functioning of the group, describing communications in the whole system.

4.1 Individual specification of the holonic agents

In order to describe the general characteristics of various types of agent (the procedure manager, the station managers and the executing agents), we use a grid adapted from Ferber [Ferber (1995)]. This grid gives a description in three dimensions, instead of the five dimensions initially suggested (cf. table 1). The physical dimension, which is too dependent upon the system, does not appear in the description of general characteristics. The relational dimension is attached to the social dimension. Concerning the organization functions, the conative and organizational functions, dealing with planning, have been grouped together; the conative function being more oriented towards needs, desires and urges which our holonic agents do not have, at least for the time being.

This grid enables us to define the functions for each holonic agent relating to: knowledge (the representational function also describes the non-procedural knowledge); action planning (the organizational function); interactions; maintenance (the preservation function) and to actions specific to the role of the agent (the productive function). These functions are described in relation to the agent's environment, the other agents and the agent itself (cf. Table 2).

Table 2. Design grid adapted from Ferber's analysis grid [Ferber (1995)]

Dimensions \ Function	*Social*	*Environmental*	*Personal*
Representational	Representation of the group, of roles, of others	Representation of the world	Representation of itself, of its capacities
Organisational	Planning of social actions, communications	Planning of actions in the environment	Planning control, meta-planning
Interaction	Description agent-society interaction, performative	Perception and action mechanisms in relation to the environment	Auto-communication Auto-action
Productive	Management, coordination and negotiation tasks	Analysis, modification and creation tasks	Auto-modification, learning
Preservation	Preservation of the society, the relations, the network of contacts	Preservation of resources, defence and maintenance of territory	Self-preservation, repair, maintenance

In a holonic system, each agent has the same structure, only its behavior and knowledge vary according to its position, its role. The design grid facilitates the definition of the behavior of the different types of agent in the HOMASCOW, that is the procedure manager role (⛬), the station manager role (♙), and the executing agent role (△). Tables 3, 4 and 5 respectively are examples of applications of the specification grid for the definition of the general and individual behaviour of the procedure manager (⛬), of the work post manager (♙) and of the executing agent (△).

Table 3. The definition of a procedure manager agent ⛬

Dimensions \ Function	*Social*	*Environmental*	*Personal*
Representational	Representation of the whole procedure, it 'knows all the ♙ for which it is responsible.	It 'knows' the documents handled and the actors in the procedure	Has a name, a state, action plans
Organizational	It follows the modeled procedure, uses inferences or scenarios, the communications are holarchic.	It takes into account the arrival / departure of agents, it gives notification of full / empty spaces.	It must obey the holonic structure
Interaction	It interacts with the ♙ for which it is responsible.	It interacts via the user interface and by interrogating the network	It follows its action plans according to its environment
Productive	It ensures that the actions of the ♙ are following the procedure, that they are coordinated.	It analyses the functionality of the network	It 'learns', memorizes, by keeping a 'log book'
Preservation	It monitors the ♙, recreates a ♙ if necessary, monitors relations, warns its human manager in the event of network failure	It requests back-ups of data and restores faulty data.	The maintenance of the ⟁ is deferred to the ♙.

Table 4. Definition of the work post manager (♙)

Dimensions/Function	*Social*	*Environmental*	*Personal*
Representational	Possesses a local representation of the procedure, knows the direct neighbors as well as the △ it is in charge of.	Knows the documents and the actors linked to the post to which it is assigned.	Has a representation of itself and its actions
Organizational	Follows the procedure, plans the actions of the △	Plans the actions of the △ according to the documents and the actors' requests	Chooses an action plan according to the state of the procedure.
Interaction	Communicates with superior ⟁, with neighbors ♙, and with the △ under its charge	Interacts with the actors via the interface, questions the network concerning the existence of documents	Follows an action plan according to its environment.
Productive	According to the actors' requests, to the state of the procedure, and to the messages, asks the △ to perform actions.	Analyses the state of the work post and of the local network	Learns by increasing own representation of the procedure of non defined actions which the actor has performed.
Preservation	Supervises the △. Checks on the existence of its ⟁	Requests backups of the data and restoration of defective data	The preservation of the ♙ is deferred to the ⟁

Table 5. Definition of the executing agent (△)

Dimensions/Function	*Social*	*Environmental*	*Personal*
Representational	Knows its manager ♙ and neighbors △	Has a representation of the documents for the work post to which it is associated	Has a procedural knowledge of own specialty
Organisational	Meets the objectives set by the ♙	Manages own actions according to the space and the documents	Classifies own objectives according to the lack of space or lack of documents
Interaction	Communicates with its manager ♙ and with its neighbors △	According to the specialty, interacts with the user, and with the work post	The △ does not act on own initiative
Productive	According to the requests of the ♙, performs its specialty and transmits the results to ♙	According to the specialty, the △ produces, modifies and erases documents.	The △ does not have the ability to modify itself
Preservation	Checks the links with its neighbors, communicates problems to the ♙	Makes back-up copies, tries to restore defective files	Preservation of the △ is deferred to the ♙

But, even though these grids enable us to have a clear view of the agents' actions according to the environment and according to other agents, they do not allow a definition of the functioning of the whole HOMASCOW. Indeed, the main problem involving the definition and modeling of the cooperative functioning of the whole multi-agent system is still present. This difficulty is partly created by the holonic structure of the system. The roles are well defined and communications are arranged according to the holarchy. “All” that remains to be done is to find an adequate method. We have therefore designed the AMOMCASYS with this aim in mind.

4.2 Specification of the cooperative functioning of the HOMASCOW

The method used, AMOMCASYS (meaning the Adaptable Modeling Method for Complex Administrative Systems) was designed for the modeling of the cooperative functioning of the procedures of the department studied (in fact, the main procedures of the department, which implicate 30 of the 60 people working in it, were studied). The method comes from an integration of several software engineering methods, which have been the subject of critical analysis [Adam (2000A)]. The integration made it possible, in a relatively short period of time, to build a method suited to our needs, in our case: to have a clear method, allowing explicit description of cooperation (communication, coordination, and collaboration) along with the degrees of responsibility of the various actors. This method is summarized in [Adam (1998B)].

We chose to use the AMOMCASYS method again for the modeling of the holonic multi-agent organization’s activities. Indeed, the parallel between the human organization studied, which was structured according to roles with well-defined responsibilities and operating on the basis of document exchange, and the holonic multi-agent organization clearly appears.

AMOMCASYS is made up of four models: a data model (the OMT data model [Rumbaugh (1991)]); an activity model (the SADT data flow model (IDEFO) [I.G.L. (1989)]); a data processing model (the data processing model of the OSSAD method [Dumas (1990)], which allows the representation of cooperation nodes) and a dynamic model, using parameterized Petri nets [Gracanion (1994)].

This method, supported by a CASE tool (Visual Basic layer based on the commercial software VISIO), enabled us not only to reveal the key points of the procedures, but also to improve them in an organizational way (especially the time for dealing with a procedure which was halved, by improving cooperation and increasing the responsibilities of the actors).

The method includes a simulator based on parameterized Petri nets [Adam (1998A)] (using Visual C++, cf. figure 5). In our Petri net, a place represents an office in which we place actors and documents (represented by a colored token). So, documents may be transmitted from place to place and actors are also able to move from an office to another. This tool is extremely important for the setting up of an assistance system because its use makes it possible to obtain assistance rules on three levels (general, local and personal rules). Indeed, the instantiation of a simulation of a procedure based on parameterized Petri nets (PPN) involves at transition level the definition of general document circulation rules. The places contain local rules, that is the rules applying to any agreements which may exist between the actors in the same office (for example, the first person free deals with the next request). And finally, at the level of the actors (represented by objects), the personal rules will be defined corresponding to their activities in the procedure.

The rules are described in the pre-condition – post-condition form. Let us assume that the rule (“patent.ready AND actor.free”; “patent BECOMES checked BY actor”) is associated with an actor. The holonic station manager agent associated to it knows this rule. Consequently, when the patent is sent to the actor, the station manager will be able to notify the actor or remind him to check the patent.

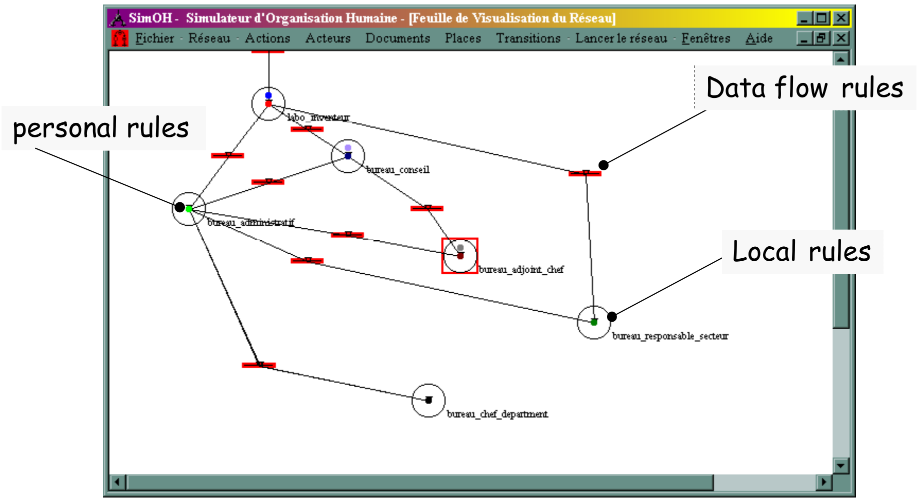

Figure 5. Extract from the simulation from a parameterised Petri network of a patent registering request procedure

The data model makes it possible to represent the principal Holon class as well as the classes associated to the representations (representation of the procedure, the actor, the workstation, the manager, the subordinates). Apart from knowledge concerning itself, each holonic agent has five main functions: to *plan* its action according to the representation of the process and its current state (corresponding to the organizational function); to *receive* and *send* to others holonic agents (corresponding to the interaction function); to *act* (corresponding to the productive function, to the specialty of the agent) and to *monitor* the links between the responsible and the subordinates (corresponding to the preservation function). Of course each holonic agent have an implicit function: to initialize (enabling it to acquire knowledge upon system start-up). The figure 6 shows an extract of the object model of AMOMCASYS, applied to the representation of a holonic agent. A main class has been defined, describing a holon in a general way. This holon possesses a name (an identifier), may be free, interruptible (by others holons). It possesses a representation of its responsible (if it has one), of its subordinates (if needed), and of its neighbors. The representation of another holon informs the holon on its name, its location (internet address, for instance: '133:28:41:14' or 'crabe.univ-valenciennes.fr') and the name of its associated human actor in the current process. The representation of the process is composed of the process name and a representation using Petri nets coming from the simulator (cf. figure 5). The workstation representation is relative to its internet address, its free space, its free memory, its pertinent files (that may be used in the process).

Class Holon
C_Holon

Attributes:
- C_String identifier
- Boolean free, interruptible
- C_RepHolon *responsible
- C_RepHolon *neighbors
- C_RepHolon *subordinates
- C_Process *whole_process
- C_Process *process_actual_state
- C_Acteur *actor
- C_Workstation *workstation

Methods:
- plan
- receive
- send
- act
- monitor

Figure 6. Extract of the object model of the HOMASCOW specification

As regards the modeling of the HOMASCOW functioning, six group behavior patterns have been identified: the *initialisation* of the HOMASCOW; the *planning* of actions according to the state of the procedure; the *sending* and *reception* of a document; the *search* for a document and the *maintenance* of the system.

Figure 7 shows an extract from the document search procedure assisted by the HOMASCOW. Overall, it works in the following fashion (we cannot give all the operating details here):

The user A makes a request to the HOMASCOW via the executing interface which is on his/her workstation. The interface transmits the request to the document's manager. If the document is on the station, it is shown. If not, the request is transferred to the station manager which tests to see if a holonic agent amongst its contacts has the document. If one of its links has the document, a request is made for it via the agents responsible for sending and receiving messages (the user B on whose post the document is to be found is then notified of the request or of the automatic sending if he doe not require the document immediately). If none of the links have the document, the request is relayed to the procedure manager, which is the only one to have knowledge of all of the documents handled in the procedure.

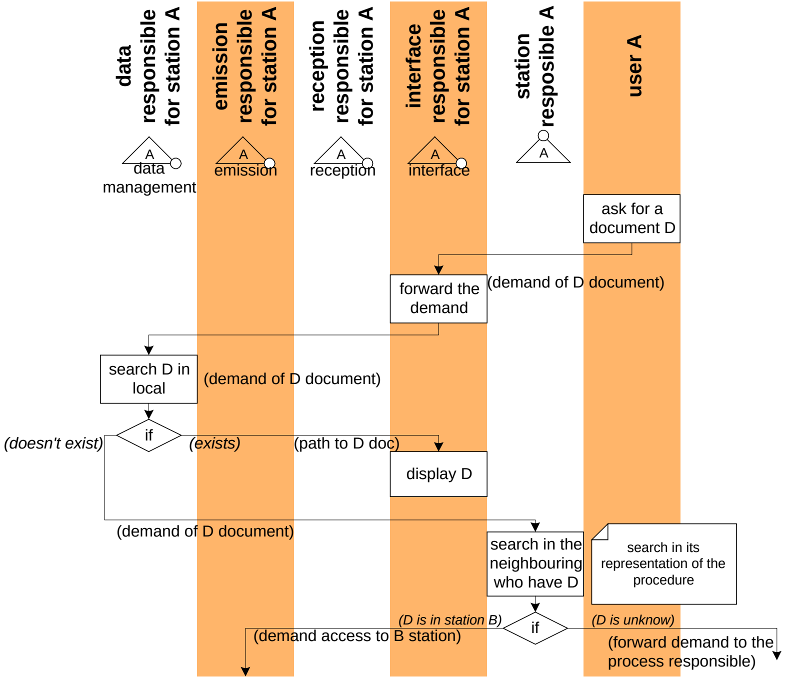

Figure 7. Document search assisted by the HOMASCOW

5. Application

The main procedures (those which were judged to be the most critical) of the company's patent department were completely modeled. They involve 30 of the 60 actors making up the department. The procedure for a patent request transmitted for technical opinion was simulated, which enabled us to obtain assistance rules. The use of this method was perceived favorably by the actors and the decision-makers. Indeed, the models used were chosen because of their clarity and they enable each actor to identify his/her own role within an overall, complex procedure.

Thanks to the CASE tool, the modifications made to the procedures following the first readings of the models were performed quickly (half a day maximum).

On the other hand, the move from modeling to simulation requires a more considerable amount of work by the modelers. The transition is not automated and the level of detail necessary, which is much finer, requires a considerable degree of data coherence management. The validation of the simulation rules is performed by the modeler and the actors involved in the procedure. Figure 8 shows the sequence of the various phases of MAMOSACO applied to the case of a patent registration procedure.

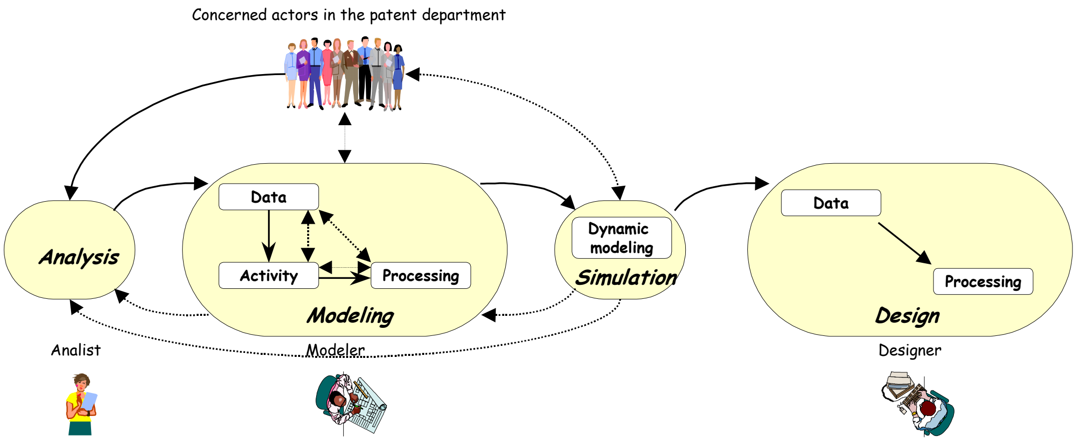

Figure 8. The phases of the AMOMCASYS method applied to the case of a cooperative procedure in the patenting department

In order to validate the architecture and the principle of communication, two HOMASCOWs were developed. For reasons internal to the company, the programming was mostly done using Visual Basic. Visual C++ language was used for the management and use of assistance rules; this was in order to reuse the inference engine developed during the development of the simulator. These implementations were facilitated by the object model, describing the general structure of a holonic agent, and the processing models, specifying the cooperative actions.

As regards communications, the agents use KQML formalism [Finin (1994)] and go through an Active X for intra-station communications (within a single sub-set) and by the actors' bulletin board for inter-station communication (transparently, these messages are automatically processed by holonic agents).

6. Conclusion and perspectives

In order to specify a multi-agent system assisting the actors in an administrative system, we turned towards a particular organizational approach: holonic systems. This approach enabled us to obtain a distribution of knowledge and roles offering a good compromise between the reactivity of the assistance system and the management of coherence in the advice given.

The specification of the assistance system (HOMASCOW) was facilitated by the development of a modeling and simulation method for human organizations bringing to the fore the cooperation and degrees of responsibility of the agent (human or otherwise).

This method (AMOMCASYS) is currently being used in a company for the modeling of human organizations. Its use for the modeling of the holonic multi-agent organization did not require any modifications because of the very similar nature of the human-machine system and the holonic multi-agent system.

As far as the functionalities of our SOHTCO are concerned, several possible improvements have been revealed. Indeed, the case studied uses a constant flow of documents. However, other application cases could have very variable flows (bringing about the management, not of two or three documents, but of approximately fifty documents, for example). It is therefore necessary to make SOHTCO adaptive, and several low-level agents must be created, which then brings about the dynamic creation of a manager for each type of low level agent generated. It is also possible to make SOHTCO adaptive through a migration of tasks (such as is suggested by [Deen (2000)]) associated to the low-level agents (a data emission agent could transform itself into a reception agent, in the event of a sudden inflow of messages). Such an adaptivity, which requires dynamic task management and distribution, should therefore be studied.

The notion of stability is very important in holonic systems. However, too great a degree of stability in the whole holonic system would, paradoxically, threaten to destabilize the system by making it rigid and less able to adapt to environmental fluctuations (it is sometimes necessary for a part of the organization to be destabilized temporarily, because of production constraints, for example, in order to allow the systems to adapt to a new environment). A management system of the stability distributed at the level of each holonic agent should therefore be suggested in order to make it possible to obtain SOHTCOs which are both stable and adaptive.

The creation of a java platform is envisaged, in order to allow a more efficient and rapid implementation of the HOMASCOW in other workflow type procedures in the company; work is currently being performed concerning this.

It is also envisaged to assess in hindsight, within the framework of a procedure whose actors are requesting the same kind of assistance, the number of communications and the impact of the HOMASCOW in cooperative work, these assessments being performed on the basis of models at the present time.

Acknowledgements

The authors would like to thank Emmanuel VERGISON for his wise comments, Jerome GRANSAC for his participation in designing the simulator, as well as Caroline GUILLAUME-BLAYDES for her help in translating this paper.

Note

(1) This paper has been extended on the basis of our HOLOMAS'00 presentation [Adam (2000B)]

References

Adam, E., Mandiau, R., Vergison, E. (1998A). “Parametrized Petri Nets for modelling and simulating human organizations in a workflow context”. *19th International Conference on Application and Theory of Petri Nets,* Proceedings of the Workflow Management Workshop, Lisbon, Portugal, June, 22-26.

Adam, E., Kolski, C., Vergison, E. (1998B). “Méthode adaptable basée sur la modélisation de processus pour l’analyse et l’optimisation de systèmes coopératifs dans l’Entreprise”. *Proceedings International Conference ERGO’IA 98,* Biarritz, France, November.

Adam, E., Mandiau, R., Kolski, C. (1999) “Approche holonique de modélisation d’une organisation orientée workflow: SOHTCO”. In : *"Ingénierie des sytèmes multi-agents. Actes des JFIADSMA'99",* (Gleizes M.P., Marcenac P., eds.), Editions Hermès, Paris, 121-134.

Adam, E., Kolski, C. (2000A) “Etude comparative de méthodes du Génie Logiciel en vue du développement de processus administratifs complexes”. *Génie Logiciel*, in press.

Adam, E., Mandiau, R., Kolski, C. (2000B). "HOMASCOW : a Holonic Multi-Agent System for Co-operative Work". *Proceedings of the first First International Workshop on Industrial Applications of Holonic and Multi-Agent Systems in conjunction with the Eleventh International Conference on Database and Expert Systems Applications,* (6-8 September 2000, Greenwich, London, UK), Holomas’2000.

Adam, E. (2000C). "Model of multi-agent organization to assist cooperative work in company processes: application to complex administrative systems" (in french), Ph.D. thesis, University of Valenciennes, September.

Arafa, Y., Charlton, P., Fehin, P., Mamdani, A. (1999). "Personal Service Assistants with personality". *Human Computer Interaction – Communication, Cooperation and Application Design*, (Bullinger H.J. & Ziegler J., eds.), London : LEA, 147- 151.

Arai, T., Aiyama, Y., Sasaki, Y. (1997) “Holonic Storage: an assembly and storage cell by manipulation using environment”. In *New Manufacturing Era - Adaptation to Environment, Culture, Intelligence and Complexity , Proceedings of the 29th CIRP International Seminar on Manufacturing S*ystems, Osaka, 221-226.

Boissier, O., Demazeau, Y. (1996). "ASIC, an Architecture for Social and Individual Control and its Application to Computer Vision”. *Lecture Notes in Artificial Intelligence, 1069,* Springer Verlag, 135-149.

A version of this paper has been published in: *Journal of Applied Systems Studies*, 2, pp. 100-115, 2001.

Burckert, H.-J. and Fischer, K. and Vierke, G. (1998). "Transportation Scheduling with Holonic MAS The TeleTruck Approach", *Proceedings of the Third International Conference on Practical Applications of Intelligent Agents and Multiagents* (PAAM'98).

Chaib-draa, B. (1996). “Interaction Between Agents in Routine, Familiar and Unfamiliar Situations”. *International Journal of Intelligent & Cooperative Information Systems*, 5(1), 1-26.

Collinot, A., Drogoul, A. (1998) “Approche orientée agent pour la conception d'organisations: application à la robotique collective”. *Revue d'intelligence artificielle*, 12 (1), 125-147.

Deen, S. M., Fletcher, M. (2000). “Temperature Equilibrum in Multi-Agent Manufacturing Systems”, *Proceedings of the first First International Workshop on Industrial Applications of Holonic and Multi-Agent Systems in conjunction with the Eleventh International Conference on Database and Expert Systems Applications*, (6-8 September 2000, Greenwich, London, UK), Holomas’2000.

Dumas, P., Charbonnel G. (1990). *La méthode OSSAD, pour maîtriser les technologies de l’information*. Tome 1 : principes. Les éditions d’organisation, Paris.

Ferber, J. (1995). *Les systèmes multi-agents, Vers une intelligence collective*. InterEditions, Paris.

Finin, T., McKay, R.F.D., McEntire, R. (1994). “KQML as an Agent Communication Language”. *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94),* November, ACM Press.

Flores-Mendez, R.A. (1999). "Towards the Standardization of Multi-Agent System Architectures: An Overview". In ACM Crossroads. *Special Issue on Intelligent Agents, Association for Computer Machinery*, 5.4, 18-24.

Georgeff, M. P., Lansky, A. L. (1987). “Reactive reasoning and planning”. *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, Seattle, WA., 677-682.

Gerber, C., Siekmann, J., Vierke, G. (1999). “Holonic Multi-Agent Systems”. *Research report*, RR-99-03, March, DFKI GmbH, Germany.

Gracanion, D., Srinivasan, P., Valavanis, K.P. (1994). “Parameterized Petri nets and their application to planning and coordination in intelligent systems”. *IEEE Transactions on Systems, Man and Cybernetics*, 24, 1483-1497.

I.G.L. Technology. (1989). *SADT, un langage pour communiquer*. Eyrolles, Paris.

Jennings, N. R., Faratin, P., Johnson, M. J., Norman, T. J., O'Brien, P., Wiegand, M. E. (1996). “Agent-based business process management”. *International Journal of Cooperative Information Systems*, 5 (2-3), 105-130.

Koestler, A. (1969). *The Ghost in the Machine*. Arkana Books, London.

Mandiau, R., Le Strugeon E. & Agimont G. (1999). “Study of the influence of organizational structure on the efficiency of a multi-agent system”. *Networking and Information Systems Journal*, 2(2), 153-179.

Nwana, H. S. (1996). “Software Agent : an overview. *Knowledge Engineering Review”*, 11(3), 205-244.

Rumbaugh, J., Blaha, Premerlani, W., Eddy, F., Lorensen, W. (1991). *Object-oriented modeling and design*. Prentice-Hall.

Soulié, JC., Marsenac, P., Calderoni, S., Courdier, R. (1998). “GEAMAS V2.0 : an Object Oriented Platform for Complex systems Simulations”. *Proceedings of TOOS-28’98*, Santa-Barbara, Californie, IEEE Press.

Van Brussel, H., Valckenaers, P., Wyns J., Bongaerts, L., Detand, J. (1996). “Holonic Manufacturing Systems and IMS”. In *IT and Manufacturing Partnerships,* (J. Browne et al., eds.), IOS Press.

Van Mulken, S., André, E., Müller, J. (1999). "An empirical Study on the Trustworthiness of Life-Like Interface Agents". *Human Computer Interaction – Communication, Cooperation and Application Design, Vol 2.* (Bullinger H-J. & Ziegler J., eds.), Londres: LEA., 152-156.

Young, M. J. (1995). “Human Error and the holon cognitive architecture”. *Proceedings 6th Symposium on analysis, design and evaluation of Man-Machine systems*, Massachusetts Institute of Technology, Cambridge, USA, june 27-29.