Multi-Agents Corporate Memory Management System

Fabien Gandon, Agostino Poggi, Giovanni Rimassa, Paola Turci

Submitted for Publication in {**Applied Artificial Intelligence journal**}. Prepared with the T&F Journal Template.

Multi-Agents Corporate Memory Management System

Fabien GANDON
INRIA Sophia Antipolis

Agostino POGGI
AOT Lab - University of Parma

Giovanni RIMASSA
AOT Lab - University of Parma

Paola TURCI
AOT Lab - University of Parma

Address Correspondence to: Fabien GANDON, INRIA - ACACIA project, 2004 route des Lucioles, BP 93, 06902 Sophia Antipolis. E-Mail:Fabien.Gandon@sophia.inria.fr

Abstract. *This paper presents an approach to design a multi-agent system managing a corporate memory in the form of a distributed semantic web and describes the resulting architecture. The system was designed during the CoMMA European project (Corporate Memory Management through Agents) and aims at helping users in the management of a corporate memory, facilitating the creation, dissemination, transmission and reuse of knowledge in an organisation. The implementation integrated several emerging technologies: multi-agents system technology (using the JADE FIPA-compliant platform), knowledge modelling and XML technology for information retrieval (using the CORESE semantic search engine) and machine learning techniques. Here, we describe the agent roles and interactions, we explain the design rationale for the agent societies and we discuss the configuration and implementation issues.*

keywords: multi-agents system, software architecture design, corporate memory, semantic web.

Materialising and managing a distributed corporate memory

With our entrance in the information society, organisations had to adapt to the shift in the economy and market rules that followed. Information systems are becoming backbones of organisations and therefore their interest in corporate knowledge management grows stronger. Corporate knowledge management aims at facilitating creation, dissemination, transmission and reuse of knowledge in an organisation (Rabarijaona *et al.*, 1999). It can rely on an organisational memory *i.e.* an explicit, disembodied and persistent representation of knowledge and information in an organisation, in order to facilitate their access and reuse by members of the organisation, for their tasks (Dieng *et al.*, 2001). The stake in building a corporate memory management system is the coherent integration of this dispersed knowledge to promote knowledge growth, knowledge communication and in general to preserve knowledge within an organisation (Steels, 1993).

The work presented here was achieved in the context of the CoMMA European project (CoMMA, 2000) aiming at implementing a corporate memory management framework in the context of two application scenarios:

- Assist the integration of new members in the organisation by providing them with the information they need to become operational as quickly as possible.
- Support the technology monitoring processes by diffusing newly available information to relevant members.

This first part presents the motivations for our design rationale and technical choices. It then explains how the memory was materialised so as to be manageable by agents. Finally we compare our project to related works.

Engineering choices and motivating observations

The multi-agent approach, relying on loosely-coupled software components, is naturally prone to facilitate integration of different technologies in one system. This was an important need for CoMMA, since several emerging technologies have been chosen for the system implementation to address initial observations:

(1) The memory is, by nature an heterogeneous and distributed information landscape. Organisations take advantage of internet technologies to simplify the diffusion of knowledge, leading to the setup of intranets. Web technologies are used to setup corporate webs to distribute information in a uniform way independently of the information storage (Corby and Dieng, 1997). These web-based corporate memories are facing the same problem of precision and recall as the Web. The semantic Web (Berners-Lee *et al.*, 1995) is a promising approach where the semantics of documents is made explicit through metadata and annotations to guide later exploitation of the online resources. Thus, we studied the corporate memory as a corporate semantic Web where RDF (Resource Description Framework) is used to semantically annotate the corporate resources.

(2) The population of the stakeholders of the memory is heterogeneous and distributed in the corporation. The system has to interface users with the content of the memory and enable them to exploit it or contribute to it. The CoMMA system relies on machine learning techniques to make agents adaptive to users and context. This goes from learning user's preferences, up to information push technologies.

(3) The tasks as a whole to be performed on the memory are, by nature, distributed and heterogeneous. Moreover, both the corporate memory and its population of users are distributed and heterogeneous. Therefore, it is interesting to consider a distributed and heterogeneous system such as a Multi-Agents System (MAS) to manage and exploit this information landscape. MAS are a new stage in abstraction that can be used to understand, to model and to develop a whole new class of distributed systems (Wooldridge *et al.*, 2000). Agents can help the users supporting retrieval of the relevant information from the corporate memory and adapting the interaction with the system to the user’s preferences (Berney and Ferneley, 1999). This new paradigm is well suited for designing and deploying a software architecture above distributed information landscapes of corporate memories: on the one hand, individual agents locally adapt to users and resources they are dedicated to; on the other hand, thanks to co-operating software agents distributed over the network, the whole system capitalises an integrated view of the corporate memory.

An annotated world for agents

The article "Agents in Annotated Worlds" (Doyle and Hayes-Roth, 1998) explains that "knowledge can literally be embedded in the world as annotations attached to objects, entities and locations"; we obtain "annotated environments containing explanations of the purpose and uses of spaces and activities that allow agents to quickly become intelligent actors in those spaces". Thus annotated information worlds are, in the current state of the art, a quick way to make information agents smarter: if the memory becomes an annotated world, agents can use the semantics of the annotations and, through inferences, help users exploit it.

RDF (Lassila and Swick, 1999) uses a simple data model based on triples and provided with an XML syntax. It enables us to represent properties of Web resources and their relationships. We describe the content of documents through semantic RDF annotations and then use and infer from these annotations to efficiently exploit the corporate memory. RDF annotations can be either internal or external to documents, thus existing resources of the corporate memory may be kept intact and annotated externally. Annotations are based on the O'CoMMA ontology (Gandon, 2001) described and shared thanks to RDF Schema (Brickley and Guha, 2000).

Keyword-based search engines are limited to terms occurrences, the introduction of ontology-based annotations enables agents to access the semantic level and use it to improve their efficiency. O'CoMMA is the keystone of our system: it is a full resource of the memory and it provides the building blocks for models (user profiles, corporate model), annotations and agent messages, with their associated semantics.

The corporate model is an oriented, focused and somewhat simplified explicit representation of the organisation. It gives the system an insight in the organisational context and environment to tune its interactions and reactions. Likewise, the users' profile captures all aspects of the user that were identified as formalisable and relevant for the system behaviour. It contains administrative information and topic interests. It positions the user in the organisation: role, location and potential acquaintance network. In addition to explicitly stated information, the system derives information from past usage by collecting the history of visited documents and possible feedback from the user, as well as the user's recurrent queries. From this, agents learn some of the user's habits and preferences (Kiss and Quinqueton, 2001). These derived criterions are used for interface or information push.

The whole structuring and models of the memory being based on a shared ontology, the semantic corporate Web provides an annotated world for agents in charge of the management and exploitation of the information resources.

A Multi-Agents Information System

CoMMA adopted the weak notion of agency (Wooldridge and Jennings, 1995) but we do not claim that in our current prototype all our agents are one hundred percent compatible with this definition. The information agents are part of the intelligent agents. A MAS is a loosely coupled network of agents that work together as a society aiming at solving problems that would generally be beyond the reach of any individual agent. A MAS is heterogeneous when it includes agents of at least two types. A Multi-Agents Information System (MAIS) is a MAS aiming at providing some or full range of functionalities for managing and exploiting information resources. Based on these notions, the software architecture of CoMMA is an heterogeneous MAIS; the application of a MAIS to corporate memories means that the co-operation of agents aims at enhancing corporate knowledge capitalisation.

So far a large number of MAIS projects focused on the problem of dynamically integrating heterogeneous sources of information. It comes from the fact that it was one of the problems being addressed in the field of information systems when MAS came to meet them and also because the decentralised nature and local adaptability of agents were assets for wrapping heterogeneous sources. Some of these systems as, for example, SIMS (Arens *et al.*, 1996) and InfoMaster (Genessereth *et al.*, 1997), rely on agents that wrap these information repositories (*i.e.* wrappers), and combine and translate information within mediator agents using mediation techniques based on a local schema. Other systems as, for example, RETSINA (Decker and Sycara, 1997) and InfoSleuth (Nodine *et al.*, 1999) are based on wrappers having in charge the translation of information.

Another related type of systems are agent-based digital libraries, for example, SAIRE (Odubiyi *et al.*, 1997) and UMDL (Weinstein *et al.*, 1999). These systems use agents to manage decentralised digital libraries and enhance scaling, flexibility and extensibility using market-based or other distributed artificial intelligence techniques for agents co-ordination.

Finally, an important set of systems is specialising in making easy the gathering of information in an organisation; see, for example, CASMIR (Berney and Ferneley, 1999). In these systems, agents help the user to gather and retrieve relevant information from one or more information repositories and adapt the interactions with the system to the user's preferences.

In CoMMA we do not stress the heterogeneous sources reconciliation aspect: documents may be heterogeneous but annotations are represented in RDF and based on a shared ontology formalised in RDFS.

Agents are used for wrapping annotation repositories indexing the memory, for enhancing scaling, flexibility and extensibility of the corporate memory and to adapt the system interface to the users. One of the points that makes CoMMA system different from most multi-agents information systems is that agents are not only used for the retrieval of information, but also for the insertion of new information in the corporate memory.

The CoMMA system helps the users in performing three main tasks: insertion of RDF annotations of new documents, search of existing documents, and autonomous document delivery in a push fashion to provide them with information about new interesting documents. The design focused on engineering an architecture of co-operating agents, being able to adapt to the user, to the context, and supporting information distribution. The duality of the word 'distribution' reveals two important problems we wanted to address: (a) 'distribution' means dispersion, that is the spatial property of being scattered about, over an area or a volume; the problem here is to handle the naturally distributed data, information or knowledge of the organisation. (b) 'distribution' also means the act of spreading; the problem then is to make the relevant pieces of information go to the concerned agent (artificial or human). It is with both purposes in mind that we designed the CoMMA architecture as presented in the following section.

Multi-agents software architecture design

Our approach shares with the A.G.R. model used in AALAADIN (Ferber and Gutknecht, 1998) and GAIA (Wooldridge *et al.*, 2000) methodologies, the concern for an organisational approach where the MAS architecture is tackled, as in a human society, in terms groups, roles and relationships. The manifesto of (Panzarasa and Jennings, 2001) goes even further, advocating the application of modern organisation theory. This type of approach is even more attractive in the context of an organisational memory.

The MAIS architecture is a structure that portrays the different kinds of agencies existing in an agent society and the relationships among them. The architectural description was studied and fixed when designing the MAS. This part will detail the architectural analysis, starting from the highest level of abstraction of the system (*i.e.* the society) and going down by successive refinements (*i.e.* nested sub-societies) to the point where the needed agent roles and interactions could be identified.

From macroscopic to microscopic

The functional requirements of the system do not simply map to some agent functionalities but influence and are finally diluted in the dynamic social interactions of agents and the set of abilities, roles and behaviours attached to them.

Organising sub-societies

Considering the system functionalities we identified four dedicated sub-societies of agents: (1) The sub-society dedicated to ontology and corporate model (2) The annotation-dedicated sub-society (3) The user-dedicated sub-society (4) the interconnection-dedicated sub-society. These societies were analysed as described in the following parts. Analysing the resource dedicated sub-societies (ontology/model; annotation archives; white and yellow pages for interconnection) we found that there was a recurrent set of possible internal organisations for these sub-societies:

A *hierarchical organisation* distinguishes between two kinds of roles:

- The representative role: the agent is a mediator between its society and the rest of the MAS. It deals with the external requests. If needed it breaks them up into several sub-requests. It contacts and delegates sub-tasks to resource-dedicated agents. Finally, it compiles the possibly partial results to report to the external requester.
- The resource-dedicated role: the agent is dedicated to a local resource repository and contributes to solve the requests it receives from the agents playing the representative role, as much as it can with its local resources.

In this organisation, workload is greatly distributed because resource agents only work with the resource they have locally and they leave the fusion work to representative agents that can be placed on another machine that does not necessarily hold a repository of information. However this organisation is quite network-consuming.

A *peer-to-peer organisation* sets up egalitarian relationships between the roles. Roles are not distributed, but completely redundant: any agent can be contacted from outside the society to handle a request concerning the resource type its society is dedicated to. It will then have to co-operate with its peers in order to efficiently fulfil the request. Agents are specialised only through the content of the local resource repository they are attached to. The workload is less distributed than in the previous case but the network-load may be decreased. There is only one role merging the two previous roles (representative and resource dedicated). Coalitions will be formed to solve external queries.

A *replication organisation* is a subtype of the previous case: neither the roles nor the repository content are distributed. Each agent keeps up to date a complete copy of all the resources and is able to solve any request by itself. Therefore the only social interactions that exist are for content updates. The workload is even less distributed than in the previous case and the contents has to be replicated everywhere an agent sits which can be an unacceptable constraint. On the other hand the system is highly redundant, thus more robust, and the network use is minimal when dealing with a request. The only role is the resource-dedicated one.

Depending on the type of tasks to be performed, the size and complexity of the resources manipulated, a sub-society organisation will be preferred to another.

Sub-society dedicated to Ontology and Corporate Model

The agents from this sub-society are concerned with the ontology and corporate model exploitation aspects of the information retrieval activities. They provide downloads, updates and querying mechanisms for other agents, on the hierarchy of concepts and the description of the organisation. For this sub-society, the three types of organisations are conceivable:

- In a hierarchical organisation, there is a Master role in charge of resolving external queries and an Archivist role in charge of a part or a view of the ontology/model.
- In a peer-to-peer society, there is a co-operative Archivist role.
- In a replication society each agent have a complete copy of the ontology/model and can resolve queries by itself.

The last choice is acceptable when the ontology is stable and when a consensus is reached by the users so that the ontological commitment is centralised and the global ontology is updated and propagated over the agent society. This option is implemented in the current prototype of CoMMA. The agents of the ontology sub-society thus play two roles:

- The Ontology Archivist (OA) has the duty to store and retrieve information to/from the CoMMA ontology repository.
- The Enterprise Model Archivist (EMA) has the duty to store and retrieve information to/from the CoMMA enterprise model repository.

The remaining options are interesting if the ontology/model is large or changes quite often and if the system intends to support the ontological consensus process; the agent society can then support the break-up of the ontology/model and maintain the coherence between the different repositories as in the FRODO project (Elst and Abecker, 2001).

Annotation dedicated sub-society

The agents from this sub-society are concerned with the management of the annotations structuring the corporate memory; more precisely, they store new annotations, and retrieve the references matching the users' queries. Here, only the two first types of organisation are conceivable:

- In a hierarchical organisation there is an Mediator role and an Archivist role.
- In a peer-to-peer organisation there is a co-operative Archivist role.
- A replication society is not realistic because it would imply to replicate a full copy of the annotations of the corporate memory for each resource agent.

The current CoMMA system opted for the first type of society. A hierarchy is appropriate because it separates the task of maintaining annotation repositories from the task of intelligent interface towards the other agents of the system. The agents of this dedicated sub-society must play two roles:

- The Annotation Mediator (AM) typically provides its services to other societies to fulfil their requests and it requests itself the services of the resource agents to effectively solve them: (1) It breaks the requests and contacts the relevant Annotation Archivists to solve queries or allocate new annotations. (2) It compiles the partial results to build the complete answer or take a decision. Therefore, it is the interface between the archivists and the other agents of the system making transparent to the other agents the implementation of the corporate memory (*i.e.*, the number of annotation repositories, their location, etc.) and its management policy. The AM allocates new annotations to AAs in a contract-net fashion. Bids are based on a similarity measure between the new annotation and the content of the archive of the AA (Gandon *et al.*, 2002).

- The Annotation Archivist (AA) has the duty to store and retrieve annotations to/from the CoMMA corporate memory. This role is attached to a local annotation repository, and when it receives a query, it tries to obtain at least partial results from its local resources to enable the AM to handle results distributed over several information sources.

Interconnection dedicated sub-society

Agents from this sub-society are in charge of matchmaking other agents using their respective needs and service provider descriptions. Each provider must first register itself with at least one middle agent and advertise its capabilities. Requests are then matched to these descriptions to find which agent can provide a required service.

The CoMMA system is implemented using the JADE platform (Bellifemine *et al.*, 2001) that provides the agent types playing two roles defined by FIPA (FIPA, 2000):

- The Agent Management System (AMS) has the duty to maintain the system white pages where agents may register themselves and ask for addresses of other agents.

- The Directory Facilitator (DF) has the duty to maintain the system yellow pages where agents may register themselves and their capabilities and ask for services of other agents. The agents that play this role can be organised into graphs, with the possibility of searches of arbitrary depth; they are federable in order to build a peer-to-peer society in charge of managing Yellow Pages. DFs provide agent identifiers matching the service description; they are matchmakers identifying relevant providers and returning the selection of candidates to the requester. The result of the matchmaking can be further refined in a second stage. For instance the AM requires statistics from the AAs to know what types of annotations they archive and to decide when to appeal to them during the distributed query solving process.

User dedicated sub-society

The agents from this sub-society are concerned with the interface, the monitoring, the assistance and the adaptation to users. They are, typically, requester agents. Because they are not related to a resource type, they cannot be studied using the typology we proposed. However, we can distinguish at least two recurrent roles in this type of sub-society:

- The user interface management: to enable users to express their requests and to present results in a appropriate format
- The management of user's profile: to archive and make the profiles available to be used for interface purposes, learning techniques, proactive searches, collaborative filtering, etc.

In CoMMA the agents of the user dedicated sub-society must play four roles:

- User Profile Manager (UPM) is in charge of updating and exploiting the user's profile when the user is logged on to the system. This agent uses machine learning techniques to adapt the system to the user; for example, it is able to know if a document is of interest for a user and to order a list of documents on the basis of user preferences. The machine learning techniques are developed by the LIRMM [Kiss and Quinqueton, 2001].
- Interface Controller (IC) is in charge of monitoring the user interface and the interaction of the user with the system. The IC agent is the system front-end, working in close collaboration with the user. Moreover, it is the only agent in the system that is not running persistently, with its lifetime limited to a single session (IC starts when the user logs into the system and shuts itself down when the user logs out). When the IC starts up, it uses the yellow pages services provided by the Directory Facilitator agent to get acquainted with all necessary agents. During system usage, the IC interacts directly with the agents belonging to the ontology sub-society, whereas it relies on a responsible UPM agent to deal with the document sub-society. A distinguished feature of IC agent is its GUI, through which users can exploit the CoMMA MAS. The GUI makes the user look as an agent to the rest of the system, because as soon as the GUI layer is crossed, all the information passing happens by means of FIPA ACL messages.

- User Profile Archivist (UPA) has the duty to store and retrieve user profiles to/from CoMMA user profile repository, maintaining the profile of the users belonging to a specific sub-net of the enterprise intranet. Only this agent has the access to the user profile repository of a specific sub-net.

- The User Profile Processor (UPP) has the duty to perform proactive queries on the corporate memory on the basis of the user profiles and information about new documents.

The UPM and UPP roles were merged in one role played by one agent since we did not find any reason in our scenario for distributing this two roles. Additional roles are conceivable for this sub-society can be found in (Gandon *et al.*, 2000); this exemplifies the usefulness of the modularity of MAS.

Roles and Interactions

From the architecture analysis, we derived the characteristics of the identified roles and their interactions as explained in the following section.

Agent roles

Roles represent the position of an agent in a society with the responsibilities and activities assigned to this position and expected by others to be fulfilled. In the design junction between the micro-level of agents and the macro-level of the MAS, the role analysis is a key step. The previous part identified the following roles, implemented in the first prototype of CoMMA:

- Ontology Archivists maintain and access the ontology.

- Enterprise Model Archivists maintain and access the enterprise model.

- Annotation Archivists maintain and access annotation repositories.

- Annotation Mediators manage and mediate among a set of Annotation Archivists.

- Directory Facilitators maintain and access yellow pages.

- Interface Controllers manage and monitor user interfaces.

- User Profile Managers update users' profile.

- User Profile Archivists store and distribute users' profile.

The list of characteristics that are debated in the agent community gives an overview of the actual capabilities envisaged. Their diversity reveals the influence and the integration in MAS research of results from a lot of research areas. Table 1 compiles some of them, found in the literature, and uses them to characterise the roles. Other methodologies, such as (Wooldridge *et al.*, 2000), propose a formalisation of the roles but it was not deemed necessary for our first prototype.

<Insert Table 1 here>

This table summarises some constraints and the specificity of each role; it drove the implementation of the behaviours of the agents playing these roles

Social interactions

Following the sub-societies and roles identification comes the specification of the interactions. Interactions consist of more than the sending of isolated messages and the conversation patterns need to be specified with protocols. Agents must follow these protocols for the MAS to work properly. Protocols are codes of correct behaviour, in a society, for agents to interact with each other. They describe a standard procedure to regulate information transmission between agents and they institutionalise patterns of communication occurring between identified roles. The definition of a protocol starts with an acquaintance graph at role level, that is a directed graph identifying communication pathways between agents playing the roles involved in an interaction scenario. From that we specify the possible sequences of messages.

The CoMMA behaviour may be shown through the scenarios describing how agents interact to perform the different tasks. These scenarios are: user login, creation of a new annotation, submission of a new annotation, information retrieval, update of user profile, push diffusion of information. These scenarios provide the requirements to specify the interactions. In the following, we illustrate our point with the description of one of the most relevant scenarios: push diffusion of information.

The agent roles taking part to this scenario are: Interface Controller (IC), User Profile Manager (UPM), User Profile Processor (UPP), User Profile Archivist (UPA) and Annotation Mediator (AM). This scenario is played whenever a new annotation appears within the corporate memory. The purpose of this scenario is to proactively detect and retrieve relevant information contained within the corporate memory, and present it to the user in the most comfortable way.

Even if CoMMA users connect to the system intermittently, in order to carry out the push diffusion of information functionality there is an obvious need to persistently store information about them and remembering it across all the sessions. Therefore the UPM, UPP, UPA and AM are continuously running, gathering information from and for the user to improve system performance. Moreover, the CoMMA system supports the user capability of logging into the system from different parts of the corporate intranet.

At any point in time, a UPM agent handles all the users currently connected to the system from within its jurisdiction domain. This means that the whole corporate network is partitioned in subsets and there is one and only one UPM for each subset. The UPP has the duty to perform proactive queries on the corporate memory on the basis of user profiles and information about new documents. Concerning push mode, the UPP is responsible for each user living within its jurisdiction domain; it handles AM notifications for new annotations and starts the push mode.

The pushing of information involves two phases:

1. user agents compute the list of users that might be interested in the information.
2. user agents notify the possible interested users about the information.

Figure 1 shows the protocol diagram of that scenario when the user is connected; the name Mgr denotes the concrete agent that plays UPM and UPP roles. The task is performed in three steps:

1. The AM informs all UPPs a new annotation was archived.
2. Each UPP generates a request to its local UPA, using the annotation and a search mechanism based on CORESE (Corby *et al.*, 2000) in order to receive the list of users (users profiles) that can be interested.
3. Each UPP sends information to each user interested. There are two possible cases:
   - The user is connected. The UPP informs the IC of the pending notification; the actual information is sent on explicit user demand (fipa-request interaction protocol).
   - The user is not connected. The UPP requests the UPA to store the pending information in the user profile. When the user logs in, the user will be notified of the stored pending information.

<Insert Figure 1 here>

Thus as shown here, the acquaintance connections among the roles and the protocols adopted derive from both the organisational analysis and the use cases dictated by the application scenarios. The acquaintance graphs and the ACL message traces are depicted using protocol diagrams, a restriction of the UML sequence diagrams (Bergenti, and Poggi, 2000).

The implementation of CoMMA relying on JADE (Bellifemine *et al.*, 2001), which is compliant with the FIPA specifications, the agent communication language is FIPA ACL. It is based, like its counterpart KQML, on the speech act theory and comes with already standardised protocols to be used or extended with the semantic of their speech acts specified in the FIPA specifications. In the first prototype, the content languages of the messages were SL for the FIPA ACL envelop (Figure 2-a) and the speech acts involved in the CoMMA protocols (Figure 2-b). RDF was used for the exchanged annotations and query patterns (Figure 2-c). The latest prototype uses RDF as a content language.

<Insert Figure 2 here>

Implementation: behaviour and technical competencies

We now discuss the implementation in a set of agent types of the behaviours of agents playing the identified roles and involved in the specified interactions.

From the role and interaction descriptions we proposed and implemented agent types that fulfil one or more roles. The behaviour of an agent type combines one or several behaviours implemented by the designers to accomplish the activities corresponding to the assigned roles. For instance there is currently one agent type playing both the Ontology Archivist and Enterprise Model Archivist roles. Its behaviour contains both associated behaviours which are themselves made up of sub-behaviours handling the different tasks and interactions linked to these roles. Behaviours are directly linked to the implementation choices and determine the responses, actions and reactions of the agent. The implementation of the behaviour is constrained by the role but it is also subject to the toolbox of technical abilities available to the designers.

First it uses a pre-existing software framework for the development of agent applications called JADE (Bellifemine *et al.*, 2001). JADE (Java Agent Development framework) is a software framework to aid the development of agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. JADE is an Open Source project, and the complete system can be downloaded from JADE Home Page (JADE, 1999). JADE offers a good support to the usage of content languages and ontologies designed for specific agent applications, in order to manage complex interactions between agents.

Then each agent behaviour implementation may call or 'agentise' other software. As an example, an AA has a behaviour handling its involvement in distributed query-solving. A task occurring in this behaviour, is the projection of a sub-part of the complete query to propose partial results to the AM that will build a complete answer. Therefore the behaviours of the AA and the AM include calls to the CORESE API (Corby *et al.*, 2000) a prototype of semantic search engine enabling inferences on RDF annotations by translating the RDF triples to Conceptual Graphs (CGs) and vice versa. CORESE combines the advantages of using the standard language RDF for expressing and exchanging metadata, and the querying and inference mechanisms available in CG formalism.

MAS Configuration and Deployment

A configuration is an instantiation of an architecture with a chosen arrangement and an appropriate number of agents of each type. One given architecture can lead to several configurations. In the case of a multi-agents corporate memory system, the configuration depends on the topography and context of the place where the system is rolled out, thus it must adapt to this information landscape and change with it. The architecture must be designed so that the set of possible configurations covers the different corporate organisational layouts foreseeable. The configuration description can be studied and documented at deployment time using adapted UML deployment diagrams to represent hosts (servers, front-end...), MAS platforms, agent instances and their acquaintance graph.

The deployment of the CoMMA MAS is constrained by the organisational structure of the company where it is installed. Moreover, the deployment is also driven by the network topology technical environment and its constraints (such as the topologies characteristics, the network maps and data rates, data servers location, gateways, firewalls, etc) and by interests area, where are the stakeholders (users, system managers, content providers…).

Even if the static model (agent roles) and the dynamic model (agent interactions) describe quite well the CoMMA MAS, it is useful to provide a third view that is the deployment model. This model can help a lot in focusing on the actual system, because the CoMMA KM solution is made by a highly distributed system deployed over a structured, managed corporate network.

Three-Tier Deployment

The Figure 3 shows a possible deployment, called Three-Tier Deployment because the physical computers hosting the agents are divided into clients (client tier), application servers (middle tier) and database servers (DB tier). Such a deployment strategy has the main advantage of matching one of the most popular intranet structures.

In this diagram, boxes represent physical hosts and generic components are replaced by agents (with bufferfly shape) and knowledge bases (represented using the familiar mass storage icon). An important observation to make is that this diagram contains physical agent instances and not agent roles. This means that a single butterfly icon can represent many agent roles that will be played together by a single physical component. A directed graph shows the agent acquaintance graph. Moreover, agents belonging to the same JADE platform are grouped together (notice that belonging to a platform implies the a priori knowledge of the Agent IDs for the AMS and the default DF).

According to the three-tier architecture, only agents that play the role of Interface Controller live on the client machines (performing pure presentation) and only agents that play the role of Annotation Archivist live on the database servers (performing pure data management). The middle tier, as in classical three-tier architectures, connects the client tier and the data base tier (through DF and AM agents) and is where the most interesting stuff happens (ontology and enterprise/user model management). In this deployment, the ontology, the enterprise model and the user model belong to the middle tier and not to the data base tier because they are supposed to be much smaller in size than the annotation base and they are not distributed.

In this deployment example, there is a JADE platform spanning the whole Lab A, another the Lab B and a third that contains all the agents that play the role of Annotation Archivist, managing the distributed annotation base.

<Insert Figure 3 here>

The user dedicated sub-society is deployed according to the following rules: the IC is active for a session and runs on the host where the user logs for the current session; the UPM is active continuously and runs on a server that manages the current location of the user (*i.e.* the machine he or she is logged on); the UPA is active continuously and runs on a server that manages the home location of the user (*i.e.* the machine where he or she belongs and where he or she usually logs on).

Configuration and System Flexibility

The CoMMA system was targeted right from the start at corporate intranets, so it relies on assumptions such as having a common, agreed upon ontology, or having a network whose topology can be derived from the enterprise model. However, even when requiring a global ontology or an administered network, the fact that those entities change with time needs to be taken into account. The CoMMA system needs to be adaptable to organisational changes of the corporation using it.

Several aspects of the CoMMA system work together to achieve this goal: firstly, the O'CoMMA ontology is a separate asset, materialised into RDF Schema files and completely decoupled from the other software artefacts. The ontology can be evolved after the initial deployment, with no need for system-wide operations: agents in the ontology sub-society will be able to reload the new version of the ontology. Of course, some rigidity remains due to the need for an ontology to capture semantic information, that cannot be produced automatically but needs to be extracted from humans. The CoMMA solution tries to ease this demanding task by defining a methodology to build and refine ontologies. One important point remains to be studied, that is the maintenance of annotation bases after a changes in the ontology occurred that requires an update of the annotations.

Even if the previous subsection presented a classical three-tier deployment for the CoMMA system, it should be emphasised that this was only a sample deployment, chosen to set up a simple example in a familiar environment. Moreover, the enterprise model is just another set of RDF annotations, so it can be augmented and modified during the system usage. Due to the highly flexible distributed component infrastructure provided by the JADE environment, the system can be incrementally deployed, new agents or platforms can be added or removed dynamically, without the need to stop normal system operations. So, the network structural changes due to corporate reorganisations can be dealt with rather seamlessly.

Above and beyond the qualities of its separate components, the flexibility of the overall CoMMA solution is achieved thanks to the system architecture that holds them together. Solid software engineering best practices were used, and adapted to multi-agent systems where needed; an iterative development process was used, which performed two complete iterations during the project, gathering user feedback through questionnaires and walkthroughs. A first version of the system was released to end users after one year of development, and a first system trial was carried out. The feedback gathered was used in building the second version of the system, featuring such non trivial changes as a complete GUI redesign driven by usability results from trial 1 and a change of the content language used by the agents from FIPA SL with RDF embedded in strings to a complete RDF-based content language, relying on a more advanced metamodel introduced by the latest JADE versions. All this happened while some major new features were introduced in the system (information push, inference rules in CORESE, new learning mechanism); nevertheless, the loosely coupled component model promoted by multi-agent systems proved itself extremely important in minimising interference between different change efforts.

Moreover, relying on a framework as JADE helped a lot in managing the content language replacement, an infrastructure change that could have affected every software component in the system, but which remained well hidden behind JADE abstract APIs.

Conclusion

In this paper we presented an approach to corporate memory management relying on the technologies of multi-agent systems, XML and machine learning. We described the methodology we followed do design such a system and the architecture we obtained and implemented in the first prototype of CoMMA. The prototype implemented in Java was evaluated by end-users from a telecom company (T-Nova System Deutsche Telekom) and a construction research center (CSTB) with archives containing up to 1000 annotations. Interface and ergonomics problems were raised by the users but the usefulness and the potential of the functionalities offered by the system were unanimously acknowledged. Even if groups and roles are not first class citizens in the JADE platform, an organisational approach proved to be successful to specify the implementation. Several methodologies now propose formal models to support a sound organisational analysis. It would be interesting to compare how these different models would represent the CoMMA architecture and capture the design rationale. 'Agentisation' was difficult for large and complex components but this would disappear if fully finalised Agent-Oriented languages were used instead of object-oriented languages with an Agent API. Therefore from a feasibility and conceptual point of view the multi-agent paradigm proved to be extremely well-suited to design a software architecture that could be deployed in a variety of configurations to adapt to a distributed information landscape. Concerning the integration phase of the development, the agent technology proved to be extremely valuable: the different agents have been developed by distant partners having the needed experience, starting from shallow agents; but since the agents are loosely coupled software components and that their roles and interactions have been specified using a consensual ontology, the integration and setup of a first prototype was achieved in less than two days.

CoMMA MAS architecture showed good modularity and flexibility. In fact, during its implementation we needed to modify some parts of the system to cope with new user requirements and it does not generate side effects on the other parts of the CoMMA system. The use of JADE increases system modularity and flexibility. The separation between the software platform infrastructure managing agent life-cycle, distribution and communication and the software implementing agent tasks decouples modifications in these two parts.

The system was developed within an international research project, funded by the European Commission IST program [CoMMA, 2000]. The project lasted two years, ending in January 2002, and it spanned two iterations of a complete development life cycle, including two field trials, carried out by end user project partners in order to validate the overall system and to provide feedback to the project partners more involved in development. As it often happens in this kind of applied research projects, the software developed within the CoMMA effort is not a mere proof of concept, but rather a prototype that is expected to be further evolved into a product or a solution after the project conclusion. An exploitation plan was part of the CoMMA project right from the start, and several avenues were envisaged, involving the whole CoMMA KM solution or separate components (e.g. the ontology, the MAS architecture). The project final report is a public document containing explicit information about project result exploitation strategies.

Acknowledgments

We thank our colleagues of the CoMMA consortium for our fruitful discussions and the European Commission that funded the CoMMA project (IST-1999-12217).

References

(Arens *et al.*, 1996) Y. Arens, C.A. Knoblock, W. Shen. Query reformulation for dynamic information integration. Journal of Intelligent Information Systems, 6(2):99-130, 1996.

(Bellifemine *et al.*, 2001) Bellifemine F., Poggi A., Rimassa G. Developing multi agent systems with a FIPA-compliant agent framework. Software Practice & Experience, 31:103-128, 2001.

(Bergenti, and Poggi, 2000) Bergenti F., Poggi A. Exploiting UML in the Design of Multi-Agent Systems. Omicidi A., Tolksdorf R., Zambonelli F. Engineering Societies in the Agents World - Springer, LNAI 1972:106-113, 2000.

(Berners-Lee *et al.*, 1995) Berners-Lee T., Hendler J., Lassila, O., The Semantic Web, Scientific American, p. 35-43, May 2001.

(Berney and Ferneley, 1999) B. Berney and E. Ferneley. CASMIR: Information retrieval Based on Collaborative User Profiling. 1999.

(Brickley and Guha, 2000) Brickley D., Guha R.V. – Resource Description Framework (RDF) Schema Specification. W3C Candidate Recommendation

(CoMMA, 2000) Corporate Memory Management through Agents, Proc. E-Work E-Business, 2000. CoMMA Home Page. Available at http://www.ii.atosgroup.com/sophia/comma/HomePage.htm.

(Corby and Dieng, 1997) O. Corby and R. Dieng. A CommonKADS Expertise Model Web Server. In Proc. Fifth Int. Symp. on the Management of Industrial and Corporate Knowledge, 1997.

(Corby *et al.*, 2000) Corby O., Dieng R., Hébert C., A Conceptual Graph Model for W3C Resource Description Framework. Proceedings ICCS'00 Darm-stadt, LNAI 1867, 2000.

(Decker and Sycara, 1997) K. Decker, K.P. Sycara. Intelligent adaptive information agents. Journal of Intelligent Information Systems, 9(3):239-260, 1997.

(Dieng *et al.*, 2001) R. Dieng, O. Corby, A. Giboin, M. Ribière. Methods and Tools for Corporate Knowledge Management. In S. Decker and F. Maurer eds, IJHCS special issue on knowledge Management, vol. 51, pp. 567-598, Academic Press, 1999.

(Doyle and Hayes-Roth, 1998) Doyle P., Hayes-Roth B., Agents in Annotated Worlds, Proc. Autonomous Agents, ACM SIGART, p. 173-180, 1998.

(Elst and Abecker, 2001) Elst L., Abecker A., Domain Ontology Agents in Distributed Organizational Memories Workshop on Knowledge Management and Organizational Memories, IJCAI, 2001.

(Etzioni and Weld, 1995) Etzioni O., Weld D., Intelligent Agents on the Internet: Fact, Fiction and Forecast, IEEE Expert/Intelligent Systems & Their Applications, 10(4), August 1995.

(Ferber and Gutknecht, 1998) Ferber J., Gutknecht O., A meta-model for the analysis and design of organizations in multi-agent systems. IEEE Computer Society, Proc. 3rd ICMAS, p. 128-135, 1998.

(FIPA, 2000) FIPA Home Page. Available at http://www.fipa.org.

(Franklin and Graesser, 1996) Franklin S., Graesser A., . Is it an Agent, or just a Program ? : A Taxonomy for Autonomous Agents. Proc. 3rd International Workshop on Agent Theories, Architectures and Languages, Springer - Verlag, 1996

(Gandon *et al.*, 2000) Gandon F., Dieng R., Corby O., Giboin A., A Multi-Agents System to Support Exploiting an XML-based Corporate Memory, Proc. PAKM'00, p. 10.1-10.12

(Gandon, 2001) Gandon, F., Engineering an Ontology for a Multi-Agents Corporate Memory System, Proc. ISMICK'01, p. 209-228, 2001.

(Gandon *et al.*, 2002) Gandon F., Berthelot L., Dieng-Kuntz R., A Multi-Agent Platform for a Corporate Semantic Web, to be published in Proc. AAMAS 2002.

(Genessereth *et al.*, 1997) M.R. Genessereth, A. Keller, O.M. Duschka. Infomaster: An Information Integration System. In Proc. Of the ACM SIGMOD Int. Conf. On Management of Data, pp. 539-542, Tucson, AR, 1997.

(JADE, 1999) JADE Home Page, 1999. Available at http://jade.cselt.it.

(Kiss and Quinqueton, 2001) Kiss A., Quinqueton J. Multiagent Cooperative Learning of User Preferences, Proc. of European Conference on Machine Learning Principles and Practice of Knowledge Discovery in Databases, 2001.

(Lassila and Swick, 1999) Lassila O., Swick R. – Re-source Description Framework (RDF) Model and Syntax Specification. W3C Recommendation

(Nodine *et al.*, 1999) Nodine M., Fowler J., Ksiezyk T., Perry B., Taylor M., Unruh A.; Active Information Gathering In Infosleuth™, Proc. Internat. Symposium on Cooperative Database Systems for Advanced Applications, International Journal of Cooperative Information Systems, World Scientific.

(Nwana, 1996) Nwana H.S., Software agents: an overview, The Knowledge Engineering Review, 11(3): 205-244, 1996.

(Odubiyi *et al.*, 1996) Odubiyi J., Kocur D, Weinstein S, Wakim N. Srivastava S., Gokey C., Graham J., SAIRE A Scalable Agent-Based Information Retrieval Engine, Proc. Autonomous Agents, 1997.

(Panzarasa and Jennings, 2001) Panzarasa P., Jennings N.R., The organisation of sociality: A manifesto for a new science of multi-agent systems, Proc. MAAMAW'01, LNAI, Springer Verlag, 2001.

(Rabarijaona *et al.*, 1999) Rabarijaona, R. Dieng and O. Corby. Building a XML-based Corporate Memory. In Proc. European Knowledge Acquisition Workshop (EKAW'99). LNAI 1621, Springer-Verlag, 1999.

(Steels, 1993) Steels, L. Corporate knowledge man-agement. Proc. of ISMICK'93, p. 9-30, 1993.

(Weinstein *et al.*, 1999) Weinstein P., Birmingham W., Durfee E., Agent-Based Digital Libraries: Decentralization and Coordination. IEEE Communication Magazine, p. 110-115, 1999.

(Wooldridge *et al.*, 2000) Wooldridge M., Jennings N., Kinny D., The Gaia Methodology for Agent-Oriented Analysis and Design, Autonomous Agents and Multi-Agent Systems, 3(3):285-312, 2000.

(Wooldridge and Jennings, 1995) Wooldridge M, Jennings N., Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, 10(2):115-152.

	Ontology Archivist	Enterprise Model Archivist	Annotation Archivist	Annotation Mediator	Directory Facilitator	Interface Controller	User Profile Manager	User Profile Archivist
Reactive								
Complex Mental State								
Graceful Degradation	X	X	X	X	X	X	X	X
Temporally continuity	X	X	X	X	X		X	X
Autonomy								
Goal-oriented				X			X	
Collaborative	X	X	X	X	X	X	X	X
Flexible				X			X	
Proactive							X	
Personality								
Communication	X	X	X	X	X	X	X	X
Adaptability								
Learning							X	
Customizable						X	X	
Mobility								
Visual representation						X		
Veracity	X	X	X	X	X	X	X	X
Benevolence	X	X	X	X	X	X	X	X
Rationality	X	X	X	X	X	X	X	X

Table 1. Roles Characteristics

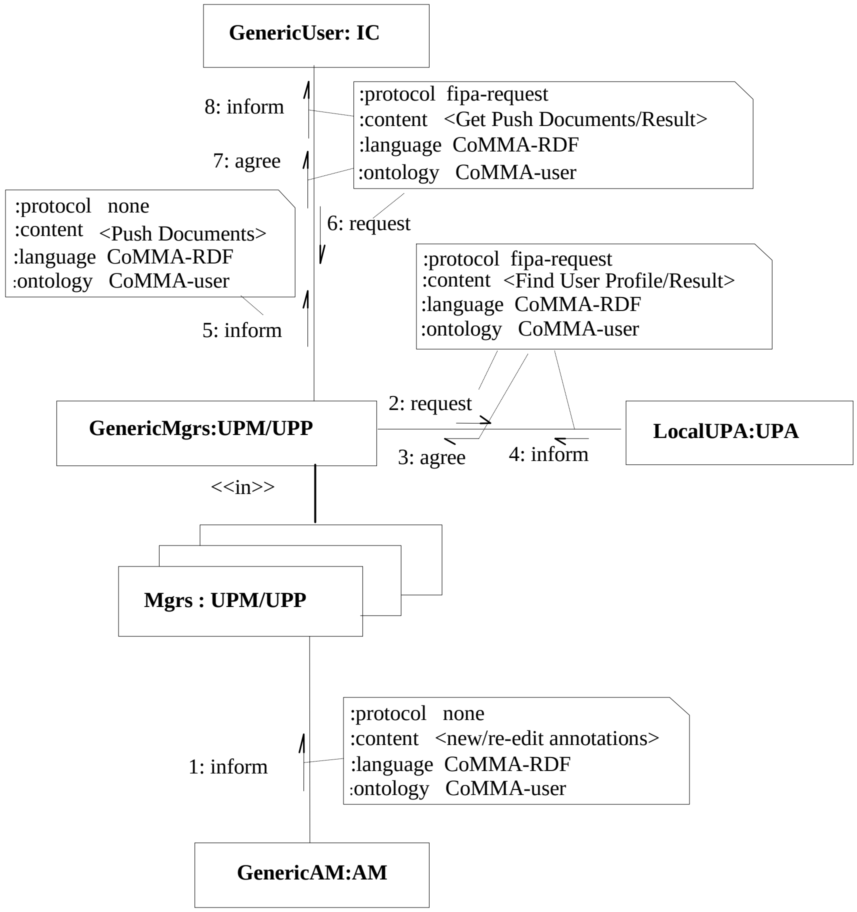

Figure 1. Push Diffusion of Information (user connected)

```
   (QUERY-REF
    :sender(agent-identifier :name localUPM@apollo:1099/JADE)
 a  :receiver(set(agent-identifier :name AM@apollo:1099/JADE))
    :content
      ((all ?x (is-answer-for
 b                 (query
        :pattern
       <?xml version ="1.0"?> <rdf:RDF xml:lang="en"
       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 c     xmlns:comma="http://www.inria.fr/acacia/comma#">
       <comma:Memo><comma:Designation>?</comma:Designation>
       </comma:Memo>
       </rdf:RDF>
 b                 ) ?x ) ) )
    :reply-with  QuerylocalUPM987683105872
    :language  CoMMA-RDF
 a  :ontology  CoMMA-annotation-ontology
    :protocol  FIPA-Query
    :conversation-id  QuerylocalUPM987683105872 )

   (a)FIPA ACL Envelop - (b)CoMMA SL0 Request - (c)RDF Pattern
```

Figure 2. A message asking for the title of Memos

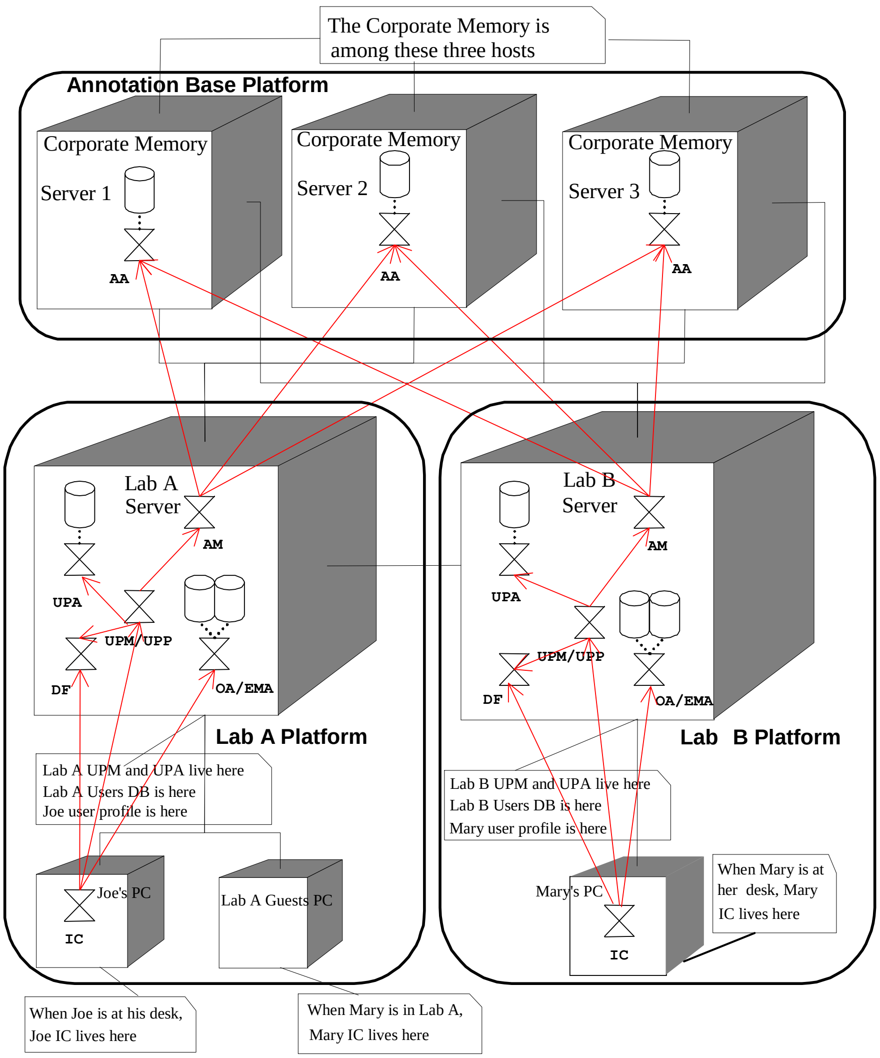

Figure 3. CoMMA Three-Tier Deployment