Distributed Artificial Intelligence and Knowledge Management: Ontologies and Multi-Agent Systems for a Corporate Semantic Web

Ph.D. Dissertation Summary - Fabien GANDON

Artificial Intelligence for Knowledge Management

Human societies are structured and ruled by organisations that can be seen as abstract holonic living entities composed of individuals and other organisations. The *raison d'être* of these living entities is a set of core activities that answer needs of other organisations or individuals. The global activity relies on organisational structures and infrastructures that are supervised by an organisational management. The individual work, be it part of the organisational management or the core activities, requires knowledge; a lack of knowledge may result in organisational dysfunction.

As the speed of markets is rising and their dimensions tend towards globalisation, reaction time is shortening and competitive pressure is growing; information loss may lead to a missed opportunity. Organisations must react quickly to changes in their domain and in the needs they answer, and even better they must anticipate them. In this context, knowledge is an organisational asset for competitiveness and survival, the importance of which has been growing fast in the last decade. Thus organisational management now explicitly includes the activity of knowledge management (KM) that addresses problems of identification, acquisition, storage, access, diffusion, reuse and maintenance of both internal and external knowledge.

One approach for managing knowledge in an organisation is to set up an *organisational memory management solution: the organisational memory aspect is in charge of ensuring the persistent storage and/or indexing of the organisational knowledge and its management solution is in charge of capturing relevant pieces of knowledge and providing the concerned persons with them*.

Such memory and its management require methodologies and tools to be operationalised. Knowledge resources, such as documents, are information supports which management can benefit from results in informatics and software solutions developed in the field. The work I carried out during my Ph.D. in Informatics participates to some of the multidisciplinary researches in Artificial Intelligence (A.I.) that aims at offering models, methods and tools for handling knowledge. I applied my research to the realisation of CoMMA (Corporate Memory Management through Agents), a two-year European IST project that was at the crossroad of several domains: knowledge engineering, multi-agent systems, machine learning, information retrieval, web technologies. This project designed, set up and tested a multi-agent information system managing a corporate semantic web and its users in the context of two application scenarios: (1) assistance in providing information to ease the integration of a new employee and (2) support to information management activities in technology monitoring processes.

Characteristics of Corporate Semantic Webs

If knowledge is power, then the distribution of knowledge is a form of distribution of power, in other words, a separation of powers. Inside and between organisations, knowledge is naturally distributed between artefacts and humans, and all things considered, the separation of knowledge pieces may be a salutary policy. However, the corporate memories are now facing the same problem of information retrieval and information overload than the Web. The initiative of a semantic Web [1] is a promising approach where the semantics of documents is made explicit through metadata and annotations to guide later exploitation. Ontobroker [2], Shoe [3], WebKB [4] and OSIRIX [5] are examples of projects that relied on semantic annotation based on ontologies to build centralised semantic search engines.

I was especially interested in the Resource Description Framework and its Schema - RDF(S) - and their associated XML syntax, which allows us to semantically annotate the resources of the memory. I focused on corporate memory materialised as a *corporate semantic Web*: *the memory is composed of heterogeneous changing documents distributed over an intranet and indexed using semantic annotations expressed with primitives provided by a shared ontology. RDF and RDFS provide the framework to write the annotations and formalise the ontology in a schema.*

Annotations play both central roles of persistent repository of acquired knowledge (the one that is formalised in the annotation) and persistent indexing of heterogeneous information resources; together annotations and resources constitute the persistent memory. Annotations are geared to distribution of information. Their formal structure enables systems to manipulate them and propagate them. When they annotate people and organisations, they can be used to provide the system with some awareness of its environment and tune its reasoning. *The description of the different user groups, profiles and roles uses the primitives of the ontology to make explicit, share and exploit a model of the organisational environment and user population.* Semantic annotations and models guide systems in exploiting information landscapes and allow them to simulate intelligent behaviours improving their performances.

Full centralisation just as full distribution is not realistic. There will always be different applications running at different geographical points and using different data. The users will always want to be able to communicate and exchange data between these applications while remaining free to manage their own data as they want. Thus, the tasks as a whole to be performed on the corporate memory are, by nature, distributed and heterogeneous. The corporate memory is distributed and heterogeneous; the population of users is distributed and heterogeneous; therefore, it is interesting that the interface between these two worlds be itself heterogeneous and distributed. Flexible and distributed architectures are needed to assist interoperation, logical integration and virtual centralisation; the agent paradigm appeared very well suited for the deployment of software architectures above the distributed information landscape of the corporate memory. Agents were designed to assist archiving and retrieval of information from the memory: *agents individuality and autonomy enable them to locally adapt to local resources and specific users; the multi-agent systems is loosely-coupled by semantic-level message passing enabling co-operation for a global capitalisation.*

Communication relies on a shared semantic of the primitives used in the messages and captured by an ontology.

An ontology provides shared conceptual vocabulary that enables efficient and non-ambiguous communication. In a KM context, it also makes explicit the organisational jargon enabling people to understand it. *The ontology is a tool for conceptualisation, a semantic grounding for models, profiles, annotations and communication messages, as well as a full component of the memory highly relevant in itself for the stakeholders of the application scenarios*.

My thesis showed that (1) the semantic webs can provide distributed knowledge spaces for knowledge management; (2) ontologies are applicable and effective means for supporting distributed knowledge spaces; (3) multi-agent systems are applicable and effective architectures for managing distributed knowledge spaces.

Related Works & Positioning in the State of the Art

The multi-agent system paradigm adopted belongs to closed distributed problem solving systems in which the agents are explicitly co-designed to co-operatively achieve a given goal. The agents envisaged here are coarse-grained agents from the symbolic branch of A.I.: internal architectures of agents make use of knowledge-based architectures, machine learning classifiers, and concurrent finite state automata. The system can be seen as a distributed expert-systems society where the expertise is document management in this organisation.

A large number of multi-agent information systems focused on the problem of dynamically integrating heterogeneous sources of information: SIMS [15], TSIMMIS [16], InfoMaster [17], RETSINA [18] and InfoSleuth [19]. Another type of systems focused on agent-based digital libraries as SAIRE [8] or UMDL [9]. My work does share with these projects, the concern for indexing and retrieving information; however I did not focused on managing or integrating the information resources themselves but their annotations and the models of the organisation and its members to tune the system to a specific memory.

The systems which are the closest to this work are specialising in the gathering of information in an organisation. Compared to CASMIR [10] and Ricochet [11], this approach does not implement collaborative filtering; however it does try to foster communities of interest through the diffusion of annotations guided by the user profiles. It also aims at modelling the organisation to anchor the memory in its environment, the models being referred to in document annotations. KnowWeb [12] implements mobile agents to address the partial connectivity of the users to the memory, an aspect which is completely overlooked here. This project also tries to extract concepts from the documents, whereas fully automatic mining is not an issue addressed in CoMMA since the annotations can be extremely complex and must be very precise. RICA [13] maintains a shared taxonomy in which nodes are attached to documents, in our case, the engineering of the ontology was done outside the system. Moreover, it is not a taxonomic indexing of documents; it is built to provide the conceptual vocabulary to express complex annotation enabling multiple axes of querying. However we do share with the RICA project the idea of pushing suggestions to interface agents according to the user profiles. Finally, FRODO [14] emphasises the management of domain ontologies and the building of gateways between different communities and their ontology. Although in reality, multiple ontologies will coexist, I considered scenarios where only one ontology is used and deployed in the whole system. My work could not possibly address the corporate memory lifecycle in its entire complexity; it focused on the pull and push functionalities together with the archiving of semantic annotation of information resources in order to manage a corporate semantic web. The CoMMA project 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 I wanted to address: (1) distribution means dispersion, that is the spatial property of being scattered about, over an area or a volume; the problem here was to handle the naturally distributed data, information or knowledge of the organisation; (2) distribution also means the act of spreading; the problem then, was to make the relevant pieces of information go to the concerned agent (artificial or human). It was with both purposes in mind that I designed an ontology for a corporate semantic Web, and a multi-agent society in charge of archiving and retrieving distributed annotations.

Ontology Design

I proposed and applied a methodology to build an ontology; the main steps of my approach are as follows:

- In *an inventory of fixture*, end-users describe current scenarios where needs were detected and ideal scenarios they would like to achieve. Scenario grids and reports are used to initialise, focus and evaluate the whole knowledge modelling and the whole application design. Informal scenario reports provide the initial base for terminological study of the application context, to initialise the ontology.
- This first set of terms initialises the *data collection* and analysis activities such as interviews, observation, document analysis, brainstorming, brainwriting and questionnaires. It enables the designers to discover implicit aspects of the conceptualisation(s) underlying the scenarios and to be in contact with real cases. Since data collection is time-consuming and resource-consuming, scenario-based analysis is also used to guide and focus the whole collection, analysis and design processes.
- Organisations are parts of broader organisations, cultures, *etc.*; knowledge is holistic and its collection is bounded by the scope of the scenarios, and not by the organisational boundaries. The *inclusion of external relevant resources* relevant to the application scenarios is part of the collection and analysis processes.
- *Lexicons* are built to capture the terms and definitions mobilised by the scenarios; they constitute the first intermediary representation towards the final ontology and they introduce the separation and links between terms used to denote the notions and the definitions of the notions. To build these lexicons, terms are analysed in context and reuse of existing ontologies and lexicons is pursued wherever possible, using semi-automatic tools to scale up the process when they are available. Built lexicons contain one and only one instance of each definition, with an associated label and a set of synonyms used to denote it.
- Then, the structuring of an ontology consists in identifying the aspects that must be explicitly formalised for the scenarios and in refining the informal initial lexicons to

augment their structure with the relevant formal dimensions against which the notions can be formally described. However ontologies and their design must be kept explicit to both the humans and systems. I showed that both populating and structuring the ontology require tasks to be performed in parallel in three complementary perspectives: top-down (by determining first the top concepts and by specialising them), middle-out (by determining core concepts and generalising and specialising them) and bottom-up (by determining first the low taxonomic level concepts and by generalising them); performing a task in one perspective triggers tasks and checks in the other perspectives in an event-driven fashion.

- The *coverage* of the ontology is evaluated in terms of exhaustivity, specificity, and granularity against usage scenarios. Any missing coverage triggers additional collection and/or formalisation. Granularity improvements require *axiomatisation* to factorise knowledge in the ontology and detect implicit knowledge in facts that may have been described from a particular point of view; inference rules were used to operationalise this phase. Ontologies are living objects and over time their usage reveal new needs for knowledge acquisition and formalisation following a never-ending prototype lifecycle.

Applying this approach, I designed O'CoMMA (Ontology of CoMMA) containing: 470 concepts organised in a taxonomy with a depth of 13 subsumption links; 79 relations organised in a taxonomy with a depth of 2 subsumption links; 715 terms in English and 699 in French to label these primitives; 547 definitions in French and 550 in English to explain the meaning of these notions. O'CoMMA is divided into three main layers: (1) a very abstract top inspired by top-ontologies and providing the relevant sub-part of such an upper ontology on which the other layers can rely; (2) a very large and ever growing middle layer divided in two main branches: one generic to corporate memory domain (documents, organisation, people, etc.) and one dedicated to the topics of the application domain (telecom: wireless technologies, network technologies, etc.); (3) an extension layer which tends to be scenario-specific and company-specific with internal complex concepts (trend analysis report, area referent, new employee route card, *etc.*).

The equilibrium between usability and reusability of notions of the ontology varies within the ontology. The upper part is extremely abstract and the first part of the second layer describes concepts common to corporate memory, therefore they both are reusable in other application scenarios. The second part of the middle-layer deals with the application domain, therefore, it is reusable only for scenarios in the same domain. The last layer extends the two previous parts with specific concepts that are not reusable as soon as the organisation, the scenario or the application domain changes.

Multi-agent architecture and design

One may say that the software architecture envisioned for the memory management could have been done in object or component programming; in fact, technically speaking, it could have been done in byte code (since at the end of the day it runs in this format), but the whole interest of agents is to provide a high level of abstraction reducing the gap between on the one hand the conceptualisation of a distributed problem and its solution, and on the other hand, the technical specification and the programming primitives used for the implementation. It is all the more true since the organisational design approach adopted here was perfectly natural in the context of the of knowledge management.

Even if the description of the design is tedious, I showed in my dissertation each stage of the design rationale of a real experience, with every refinement stage having a dedicated documentation. The approach taken is only possible for a closed system *i.e.*, a system where the type of agents is fixed at design time. We showed that the architecture is flexible enough to accept new roles for additional functionality with a minimum rework, but it was not designed for a full open world with rapidly changing configurations and agent types.

From the state of the art, the approach reused the general idea of a top-down functional analysis along the organisational dimension of the multi-agent system using the roles as a turning point between the macro-level of societal requirements and structures, and the micro-level of individual agents and their implementation. This is an organisational approach in the sense that the architecture is tackled, as in a human society, in terms of roles and relationships.

An architecture is a structure that portrays the different families of agents and their relationships. 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 and a given configuration is tightly linked to the topography and context of the place where it is deployed; thus, the architecture was designed so that the set of possible configurations covers the different layouts foreseeable.

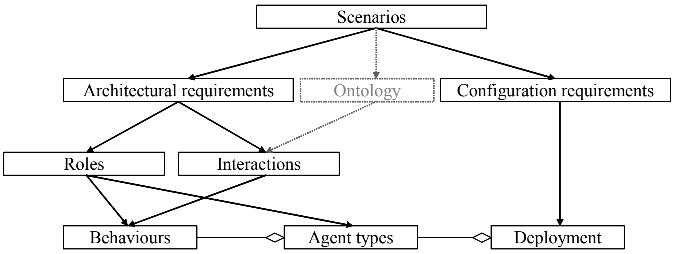

Fig. 1 Top-down organisational and functional analysis

As shown in figure 1, the architectural design comprises the following steps:

- Considering the functionalities requested for the system at the social level, we identified *dedicated sub-societies* of agents to handle the different general facets of these general functionalities.
- Considering each one of the sub-societies, we identified a set of *possible organisational structures* for them (in the present case: hierarchical, peer-to-peer, replication). Depending on the type of tasks to be performed, the size and complexity of the resources manipulated in each a sub-society, an organisational structures is preferred to another.
- From the organisational structure analysis, we identified *agent roles*, that is, the different positions an agent could occupy in a society and the responsibilities and activities assigned to this position and expected by others to be fulfilled. We studied the different role identified using role cards and comparing them along the agent characteristics usually identified in the literature.
- In parallel to role descriptions, we identified *interactions* among agents playing the roles. The role interactions are specified with *protocols* that the agents must follow for the

MAS to work properly. The documentation of an interaction starts with an acquaintance graph at role level that is a directed graph identifying communication pathways between agents playing the considered roles. Then, we specified the possible sequences of messages. The acquaintance network and the protocols are derived from the organisational analysis and the use cases dictated by the application scenarios.

- From the role and interaction descriptions, the different partners of CoMMA proposed and implemented *agent types* that fulfil one or more roles. Behaviours come from the implementation choices determining the responses, actions and reactions of the agent. The implementation of a behaviour is constrained by the associated roles and interactions and is subject to the toolbox of technical abilities available to the designers. Some roles were merged in one agent; some were split in two because part of the role corresponded to another existing role.
- For a given instance of the architecture, the configuration is studied and documented at deployment time using adapted *deployment dia*grams.

Thus, the architectural design started from the highest level of abstraction (*i.e.*, the society) and by successive refinements (*i.e.*, nested sub-societies as shown in figure 2) it went down to the point where agent roles and interactions could be identified. The user-dedicated society comprises three roles:

- The *Interface Controller* manages and monitors the user interface; it makes the user looks just like another agent. G
- The *User Profile Manager* analyses the users' requests and feedback to learn from them and improve the reactions of the systems, especially the result ranking.
- The *User Profile Archivist* stores, retrieves and queries the user profiles when requested by other agents. It also compares new annotations and user profiles to detect new documents that are potentially interesting for a user and to proactively push the information.

Precise querying on user profiles is handled by another agent role (AA) from the annotation-dedicated society.

CoMMA uses the JADE platform, thus the agents of the connection sub-society play two roles defined by FIPA:

- The *Agent Management System* that maintains white pages where agents register themselves and ask for addresses of other agents on the basis of their name. G
- The *Directory Facilitator* that maintains yellow pages where agents register themselves and ask for addresses of other agents on the basis of a description of the services they can provide.

The society dedicated to ontology and model relies on:

- The *Ontology Archivist* that stores and retrieves the O'CoMMA ontology in RDFS.
- The *Enterprise Model Archivist* that stores and retrieves the organizational model in RDF. G

The annotation-dedicated society comprises two roles:

- The *Annotation Archivist* that stores and searches RDF annotations in the local repository it is associated to.
- The *Annotation Mediator* that distributes subtasks involved in query solving and annotation allocation and provides a subscription service for agents that wish to be notified of any newly submitted annotation.