Towards Agent-based Architecture of Distributed Knowledge-driven Information Systems

Marek Kisiel-Dorohinicki, Grzegorz Dobrowolski, Edward Nawarecki

Institute of Computer Science
AGH University of Science and Technology, Kraków, Poland
{*doroh,grzela,nawar*}*@agh.edu.pl*

Abstract

A problem of knowledge management in information systems designed for open heterogeneous environments is considered in the paper. Agent approach is discussed as a design foundation and technological solution supporting the realization of such systems. A generic architecture of the system with *explicit knowledge* is proposed to aid the construction of decentralized decision-support systems. Selected implementation details of the realized platform for rule-based knowledge exchange conclude the work.

Keywords: multi-agent systems, knowledge management.

1 Introduction

Tremendous development of computer devices and software tools leads to newer and newer solutions for computer systems architecture oriented to various fields of application. Problems of knowledge acquisition and exchange (management of knowledge in general) create new directions of research and technology development, and seem to be *signum temporis*. In consequence, more and more resources are devoted to the design and implementation of systems that deal with these problems and provide adequate functionality. Effective operation of such systems requires great flexibility, in particular, ability to follow dynamic changes of demand for services. Application of artificial intelligence methods and new software techniques opens possibility to build dedicated mechanisms that consist in adaptation of a structure and organization of the system in order to widen (deepen) an offer and gain better quality of services.

In the paper some aspects related to the architecture of a system with a large variety of information resources are considered. The aspects deals with both their content as well as modes of their utilization. Such a system can be characterized briefly as an easily extensible (modified) platform that organizes co-operation of components implementing these information resources. It is assumed that the components may be built at different moments of time, under different conditions, and using computer technologies (tools) that did not keep exactly the same standards and, in consequence, are hardly integrated. In order to make the co-operation of the components possible as well as to facilitate use of the system by users of different profiles, providing possibly full knowledge about the components and platform in a symbolic and processable form is assumed. In this way a formal basis for the architecture becomes the idea of *systems with explicit knowledge* proposed in [4].

2 Knowledge management

The specific feature of decentralized information systems is the operation of information (data as well as knowledge), which not only comes from

**Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. Vol. 9 No 28 (2005), pp. 71-76.
ISSN: 1137-3601.**

different sources but simultaneously may have different thematic fields and levels of details as well as be variously structured and interpreted. This feature inflicts essential problems in stages of both designing and utilisation of such systems. Therefore, a key role in the regular work of the systems plays the management of the resources understood here as the choice of suitable for the user's needs, sources of data and knowledge as well as modes of their utilisation.

Accepting, according to the status quo, that the creation or modification of the resources can be carried out autonomously, one needs to define metaknowledge describing these resources individually and creating a basis for management procedures. A solution acceptable here is to use the ontological description of the considered application area (domain) in the form of the simplified graph of notions. The nodes of it reflect notions representative for data or knowledge available and processed in the system. The edges point at relations between the notions, which in turn occur essential for the system functioning (see fig. 1).

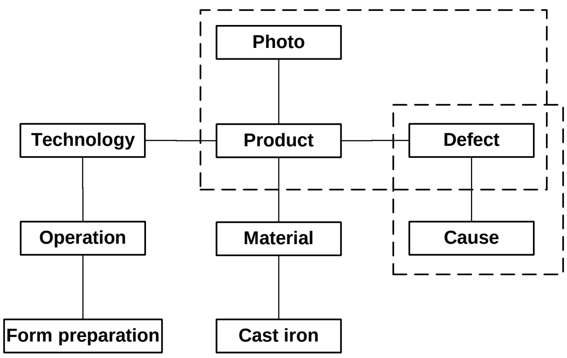

Figure 1: Graph of notions describing indicated sources of data or knowledge, related to the casting defects problem

The graph representation of metaknowledge acquires expressiveness when a concrete domain is considered. Therefore for the sake of illustration, the fragment of a graph is reproduced describing the knowledge resources of INFOCAST—the system dedicated to casting industry [2]. In particular, the areas of knowledge available through the data and knowledge resources relating to: recognizing a defect and the diagnosis of its cause respectively, are singled out with dashed rectangles.

3 Assumptions and design paradigm

The basic function of a system is to provide data and knowledge about the determined, relatively wide area of application. Terms, under which this function is created, can simultaneously be the design foundations of the system:

1. Providing knowledge is carried out in the form of the realization (mutually with the user) of various kinds of technical assessments in the area under consideration. In this matter the system resambles in its function advisory or decision support systems.

2. The assessment is usually based on several data and knowledge sources. At least some of them are accessible. Primary to the assessment performing is combining adequate and suitable—for the user—components of the system.

3. The components—with regard to the both magnitude of the domain and numbers of people engaged in creating them—represent the sources of diverse character and differing maybe somewhat in the sense of knowledge—applied terms and relations among them.

4. In general, the complexity of the assessment may require the arrangement of its course (in the sense of the components used) also in co-operation with the user. Moreover, the use of the component may occur ambiguous to the user or the steering out of the mode of its utilization may be neccessary (especially for a complex component).

5. Besides the integration of data and knowledge the necessity of the components integration arises in the aspect of applied technologies.

The basic decision which is undertaken in turn is the choice of a design paradigm. As it can be taken out easily from the above mentioned assumptions, the agent oriented paradigm [6, 7] fulfils them successfully. The most important features describing a multi-agent system—MAS (the central notion of the paradigm) are expressed in the following statements.

- A task delegated to the system is decomposed by the agents themselves neither by a designer nor a user. Moreover, the reorganization of the system is possible during its work. There is no central control—the system is decentralized.

- The agents can have also their individual (local) goals. Conflicts can arise in effect of contradiction in the local goals as well as with respect to the global goal (of the system).

- Each agent has incomplete knowledge or insufficient abilities to fulfill the task on his own. Data and knowledge are spread over the system. The agents acquire information about the possibilities and goals of other agents. Even complicated patterns of communication are needed for that purpose.

- The environment may change continuously, an agent ought to notice and remember such changes using some internal representation of it.

- The agents operate asynchronously. Each agent may enter the system or give up solving the problem at any moment—the system is open.

Treating components implemented independently, in different time, and with various technologies as legacy systems, allows to capture comparatively easily the technological aspect of integration. The modularization of the system based on agents makes possible also the encapsulation of the components and, therefore, keeping autonomy due to their creators or owners. The paradigm of web services would be competitive in the discussed aspect, insufficiently capacious, however, in the face of complicated and autonomous functions attributed to the components.

The arrangement of agents into the system (here assumed as open one) requires only the elaboration of suitable interactive (communication) protocols and implementing them using a chosen agent platform [1].

The next realization question dealing with the stressed above ability to the elastic self-adapting of the system to the requirements of an assessment needs yet another paradigm, which would generate desirable organization among the agents (including the user). A framework for the subject can be the idea of multi-agent systems with knowledge expressed explicitly [2] or, generally, knowledge-driven systems. Interaction protocols in such systems supervise the exchange of data and knowledge not only in the range directly required by an assessment, but also aiming at organization and adaptation of the system internally as well as to the user's requirements.

4 Flows of information and knowledge. Overall structure of the system

The structure of the system comprises three types of agents. They constitute three layers, that process gradually the idea of an assessment, maybe not strictly defined, in order to obtain the set of acting components (see fig. 2).

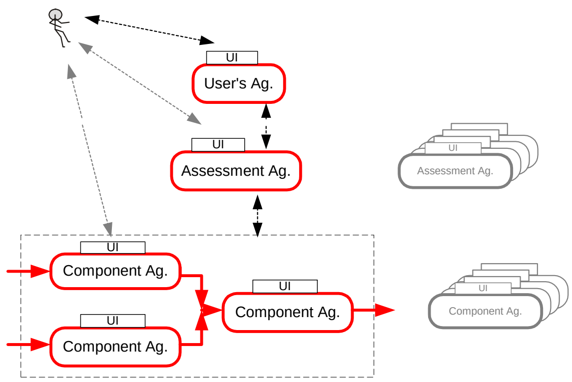

Figure 2: Flows of information and knowledge in the system

The user's agent operates in the first layer. As the result of a preliminary dialogue with the user the range of the assessment becomes agreed and access to the appropriate assessment agent is organized. If it is needed the dialogue is continued by the assessment agent, that aims to specify deeplier the range of assessment and relate its requirements with the possibilities of the system. At last the repertoire of components as well as connections among them indispensable for the assessment are established and the whole structure is set to operate. Depending on the current possibilities of the system—various assessments that can be done, several such agents occupy the second layer. Agents that encapsulate components occupy the third layer. It is obvious that in a general case a component agent can work alternatively for several different (to some extent) assessments. Moreover, the assumption about openness (possibility of adding new components to the system) is met automatically.

Assuming the graph representation of metaknowledge the user's agent activity leads to the definition of a subgraph where an assessment agent operates. In turn, this agent, communitating with the chosen component agents, completes the appropriate structure of inference that allows carring out the assessment.

Communication among the elements of the system (realized protocols) is threefold.

1. Between the user and agents using their user's interfaces—UI. The system receives data and knowledge indispensable to steering out realization of an assessment, introducing input data, and paying back results. Because of obvious reasons not all agents have to be equipped with UI.
2. A horizontal flow among component agents. The components configured in a simple network will pass data connected with an assessment. It is assumed that in some cases the network of components will be fed up with data from (or will direct them to) external information resources and most often in off-line mode.
3. Among agents but across the layers. The flow of knowledge and data indispensable to the composition of the network dedicated to an assessment.

Flows of type 1 and 3 are in fact the elements of knowledge and metaknowledge processing executed by agents and the user. This processing comprises also other mechanisms built-in into the system (agents) that are shortly discussed beneath.

Due to the different origin of components, the assumption about sharing ontology can hardly be adopted without the implementation of a special supporting mechanism. Differences can appear mostly in questions concerning the range of components application (including definitions of input and output data), but also the way of expression of the user's requirements. Incompatibility would be frequently observed in contact between the implementator and user at the level of a component in the former case, the latter—at the level of the whole system. Therefore, the system has to be equipped at least with the mechanism of coordinating the nomenclature or in general—ontology.

Because the configuration of an assessment depends on the mutual decision of the user and system based on the agreed ontology, yet another mechanism is necessary—inference one. Depending on the assumed knowledge domain representation the mechanism will search for and combine suitable components.

It should be noticed that both introduced mechanisms perform the function of a classic manual of the system. Their further development can lead to the new element of the system that will actively coach the user in the magnitude of the system particularly in the face of its openness (steady development).

5 Agent technology in open information systems

Search for the design methods and implementation techniques of the discussed-above class of information systems seems to be very easy at the moment. Indications that can be derived form the previous section point unambiguously at the means of application that is the **agent technology**.

Although the technology development ensued mainly in connection with the realization of information acquisition in wide-area networks, it covers the demand of the modern information systems in the significant part. There are advanced tools for flexible communication among dispersed cooperating agents or access to the information of remote databases implemented in whatever technology. At the same time there is much to do in a field that can be called *management of* the agents' *knowledge*.

It seems obvious that integration of distributed and often heterogeneous knowledge in a multi-agent system must be supported by an appropriate infrastructure – communication protocols and services providing information about knowledge possessed by agents, as well as about the structure of this knowledge (ontology). This assumes the existence of agents responsible for the management of such information and availability of specific services. Existing software tools supporting development of decentralized systems (sometimes called agent platforms) provide communication facilities and basic functionality for the identification and localization of agents and their services.

Of course, the cooperation of agents delivered by different vendors (interoperability) is only possible in open architectures, when the compatibility of technological solutions is sustained. So agent-based software cannot realize its full potential until standards supporting interoperability are available and widely used by developers [1, 5]. FIPA[1] specifications exemplify such standardization efforts, identifying the key elements of the system (*Abstract Architecture*), defining protocols and languages for inter-agent communication (*Agent Message Transport* and *Agent Communication*), as well as the framework for the registration, location, communication, and migration of agents (*Agent Management*). The same specification (part *Ontology Service*) describes ontology agents, providing services allowing for communication between agents that use different ontology.

It must be stressed here that available specifications and tools (platforms) provide only generic agent templates (shells) and basic management services. In particular, they often neither define the internal agent structure nor rules of its behavior— including representation of knowledge, decision making procedures, etc. It means that the platform offers only some kind of the agent *casing* compatible with other elements of the system, which most often implies that the agent is able to send and receive messages. Yet the contents of this casing (interpretation of messages, algorithms describing the agent's behavior) lies in the designer's hands.

6 Architecture of the system with explicit knowledge

Following the above considerations a platform dedicated to the exchange of rule-based knowledge with explicitly defined structure (ontology) was designed and implemented. The main goal was to provide the functionality of distributed backward chaining inference (verification of hypothesis) based on knowledge from many sources using compatible ontology [3]. This allows the construction of decentralized expert systems combining heterogeneous knowledge components.

The proposed architecture consists of the following *component agents*:

data servers give access to data from relational databases and warehouses,

knowledge servers store the domain-specific knowledge of the same ontology, provide and use inference services,

data mining servers utilize knowledge obtained from data servers via incremental data mining procedures,

ontology agents store information about relations between ontologies used by particular agents and thus support the selection of partners for cooperative inference,

clients interface agents which allow (human) users to utilize knowledge processing capabilities of the system (give expertise in selected domains).

These elements were implemented on JADE platform and utilizing XProlog knowledge representation.

The internal structure of the agents encompasses a knowledge base and inference engine. The cooperation of the agents consists mainly in verifying hypotheses (proving goals) based on the local knowledge of particular agents. This results in distributed backward chaining, which is simply the extended version of the classical backward chaining inference technique. The difference appears if there is no rule which can be used to confirm the current goal in a local knowledge base. The agent-server (performing this stage of inference) first tries to ask the agent-client (which asked him for the verification of some hypothesis). The agent-client in turn tries to use its own knowledge base to find a solution. If unsuccessful, it may transfer questions to the "previous" agent, and so on. If there is no possibility to ask or there is no reply, the agent-server tries to locate other agents using compatible ontology (by sending a query to a directory service) and ask them to prove the current goal.

[1] http://www.fipa.org

7 Concluding remarks

Some aspects of the design and realization of information and decision support systems, in which sources of data and knowledge have decentralized character, were considered in the article. The use of agent technologies is proposed as a general solution. Special attention was turned to the problem of knowledge management. Such approach creates the possibility of intelligent knowledge processing—building metaknowledge structures suited needs generated by an assessment to be done. Following the above, the idea of architecture and implementation assumptions of the knowledge exchange platform dedicated to open environments was introduced.

It seems that the presented approach brings certain elements of novelty to methodology standing behind the given class of systems. Some of the proposed solutions are applied in an information system dedicated to casting and metallurgical industries, which is realized also by the team of Intelligent Information Systems Group at UST.

References

[1] J. Dale and E. Mamdani. Open standards for interoperating agent-based systems. *Software Focus*, 1(2), 2001.

[2] G. Dobrowolski, R. Marcjan, E. Nawarecki, S. Kluska-Nawarecka, and J. Dziaduś. Development of INFOCAST: Information system for foundry industry. *TASK Quarterly*, 7(2):283–289, 2003.

[3] R. Marcjan, M. Kisiel-Dorohinicki, B. Śnieżyński, G. Dobrowolski, and E. Nawarecki. Knowledge components of a decentralized expert platform. In M. Hamza, editor, *Proc. of the Second IASTED Int. Conf. on Artificial Intelligence and Applications.* IASTED/ACTA Press, 2002.

[4] E. Nawarecki, G. Dobrowolski, S. Ciszewski, and M. Kisiel-Dorohinicki. Ontology of cooperating agents by means of knowledge components. In V. Mařik, J. Müller, and M. Pěchouček, editors, *Multi-Agent Systems and Applications III*, volume 2691 of *Lect. Notes in Artif. Intelligence*, pages 180–190. Springer, 2003.

[5] P. O'Brien and R. Nicol. FIPA – Towards a standard for software agents. *BT Technology Journal*, 16(3):51–59, 1998.

[6] G. Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* The MIT Press, 1999.

[7] M. Wooldridge, editor. *An Introduction to Multiagent Systems.* John Wiley & Sons, 2002.