Knowledge Components Using Agent Technology for a Diagnostic System

Robert Marcjan, Grzegorz Dobrowolski
Marek Kisiel-Dorohinicki, Edward Nawarecki

Department of Computer Science
University of Mining and Metallurgy, Kraków, Poland
{marcjan,grzela,doroh,nawar}@agh.edu.pl

Abstract. The paper presents an agent-based platform dedicated for the development of decentralised knowledge-based systems. At conceptual (virtual) level such a system may be treated a set of *knowledge components*, which represent well-structured, reusable pieces of knowledge together with describing it ontology. Integration of the components is possible only if the knowledge of any component may be accessed via ontology known to the others. At physical (real) level the ralization of the system is based on agent technology, which should allow for interoperability between heterogeneous entities (built on diffrent platforms, using its own knowledge representation and reasoning strategy, etc.). The considerations are illustrated by a particular application of the above approach to a decentralised expert system for casting defects diagnosis.

1 Introduction

During the last decade the idea of an intelligent autonomous agent and an agent-based system gains more and more interest both in academic community and industry. Agent technology is used in various domains, providing concepts and tools for development of intelligent decentralised systems [3]. Also application of agent-based technology in systems with explicit knowledge representation occurs to be promising, especially when sources of knowledge have some spatial structure – often management of such knowledge bases is decentralised, and forms of representation as well as procedures of enquiry differ from place to place.

A multi-agent system (MAS) built of a set of autonomous communicating intelligent entities holds in obvious way main features of systems of this class. Moreover, such representation opens possibilities of utilisation of solutions and programming tools elaborated already in the field of agent-based technology. Having some knowledge represented somehow and communicating each to other, agents are able to follow a common thread of interactions and synthesise a solution to a problem faced by the system. Yet heterogenous character of the knowledge needs communication protocols used by agents to be flexible enough to not only carry all the necessary information but also allow for the knowledge transformation as well. The main problem in this case is to build (sustain) a shared ontology.

In the paper a two-level analysis of such systems is proposed: at physical (real) level each agent in the system communicates via well-defined interfaces providing some information and/or services to other agents, while at conceptual (virtual) level an agent is said to represent some component of knowledge available through well-defined ontology. Narrowing the scope consecutively, further considerations concentrate on systems with rule-based knowledge representation and a concrete architecture for decentralised expert systems development conforming to FIPA [1, 7] specifications is introduced. As an illustration a system for casting defects diagnosis is also described.

2 Assembling of knowledge components by means of agents interactions

Variety of admitted architectures of agents, which is the consequence of assumed different sources and representations of knowledge, makes it very hard to formulate overall description of the system. Analysis carried out simultaneously at conceptual and physical levels can be promising in this situation. The essence of the proposed approach is sketched in fig. 1: while at the lower (physical or real) level it still remains a multi-agent system providing technology powerful enough to support all implementation problems, at the upper one (conceptual or virtual) questions with respect to knowledge and ontology may be discussed.

The key notion here is a knowledge component, which may be defined as a coherent piece of knowledge together with formally described ontology that structuralizes it. A component allows for resolution of some partial task of the system, e.g. realisation of

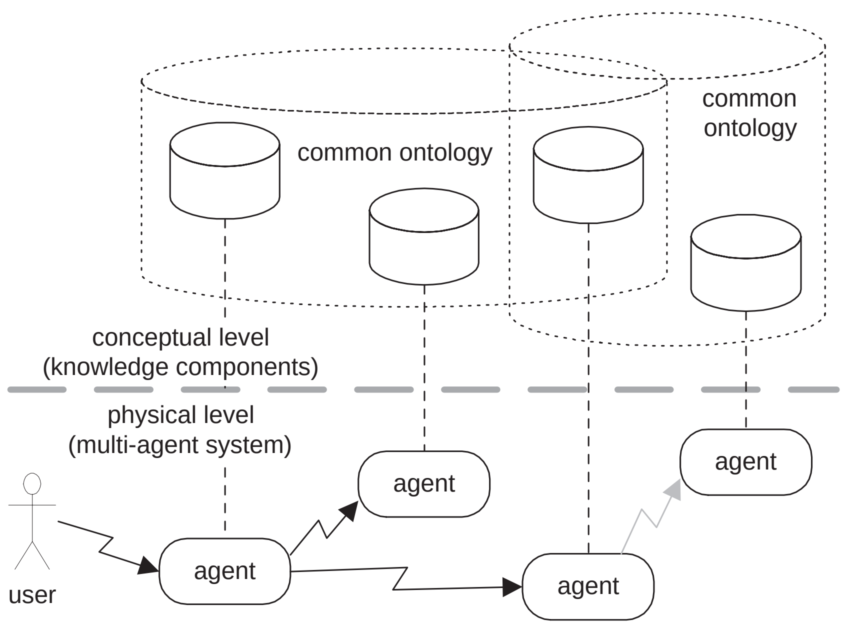

Fig. 1. Knowledge components of a multi-agent system

an aspect or stage of reasoning. From the application point of view a component may represent some knowledge about a technological unit (a part of the production process) or disease (in medical diagnosis).

Each connection between the levels in fig. 1 means that a subset of components finds its embodiment in an agent or a group of them. Agents must be able to understand one another, which at physical level is achieved by a common communication language. Yet for semantic interoperability it is not enough – it is possible *only if* the components share ontology. At the upper level this plays role of a glue that allows putting together necessary knowledge components. Of course the knowledge of a component may be available through diffrent ontologies, which means that an agent is able to communicate in different ontologies. This allows for integration of heterogenous knowledge provided by different vendors.

3 Agent technology for decentralized expert systems

The proposed approach seems a suitable tool for the development of an open architecture for decentralized expert systems since:

- it is easy to extend the system by adding appropriate agent(s), possibly agents can be developed by different vendors,
- inference engine can be specialized for given agent task – it is not necessary to build universal and thus complex inference engine,
- every agent has a knowledge base, whch is necessary for its goals so it is small and thus easy to create and maintain,
- it is possible to sell problem solving abilities, not a complete system with a knowledge base.

Yet in open environments (i.e. from and between heterogeneous information systems supplied by multiple vendors) efficient access and exchange of resources and services is possible only when standards to support agent interoperability are available and widely used by developers. Even though a key role within standardization efforts of agent-based software plays communication [2], a common communication language (agent communication language – ACL) is not enough to support interoperability between different agent systems. Also some management infrastructure seems indispensable, which should provide a unique way of agent identification, facilities whereby agents can locate each other (directory services), and a secure and trusted environment where agents can operate and exchange messages (a platform).

An example of such standarisation effort is the Foundation for Intelligent Physical Agents (FIPA), which is a non-profit organisation which purpose is *the promotion of technologies and interoperability specifications that facilitate the end-to-end interworking of intelligent agent systems in modern commercial and industrial settings* (FIPA mission statement). Contributors to FIPA may produce their own implementations of software frameworks as long as their construction and operation complies with the published specifications, which makes that the individual software frameworks are interoperable. FIPA specifications neither describe how developers should implement their agent-based systems, nor specify the internal architectures of agents. Instead, they rather provide the interfaces through which agents communicate [1, 7].

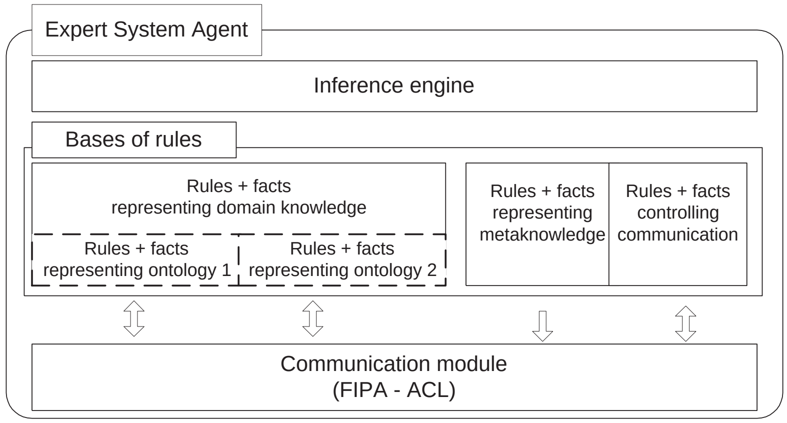

Fig. 2. Agent based realization of the rule knowledge component

4 Knowledge components of the decentralized expert system

Described decentralized expert system consist of a set of individual expert systems and knowledge bases that cover various aspects of considered domain. These expert systems inter-operate, sharing their knowledge and reasoning capabilities.

The core of the system is based on agents utilizing knowledge components, agents operate as an expert systems that possess a part of the domain knowledge and are able to utilize and serve that knowledge to other agents. The knowledge component should express a well structured fragment of knowledge together with an interface describing how to use that knowledge. Presented approach is based on rule knowledge representation. Rules are used both for domain knowledge representation of the knowledge components (representation of the domain knowledge that come from human experts) and the for the control knowledge (knowledge that control agents operation and behavior).

A scheme of an agent based realization of the rule knowledge component is presented fig. 2. At the conceptual level a knowledge component can be treated as a set of facts and rules. Facts represent the knowledge about terminology, objects (their structure, attributes etc.) and rules represent knowledge on relations that hold on objects. Build in inference engine can be used to reason (make inference) on that knowledge and produce conclusions (new facts or even new rules). The facts and rules together with the inference engine represent all the knowledge possessed by the component. A subset of that knowledge can form external interface of the component (external ontology). The component operates as a knowledge server that can serve knowledge services at a level described by the external ontology. Any question to the component as well as the answer given by the component must be expressed i terms specified by the external ontology (terminology, facts and rules). Decision what part of the whole ontology of the component should form the interface available for others (the external ontology) is an arbitrary one and should be taken by the knowledge engineer at a design time. Lets remember that knowledge component can introduce itself using several different exter-

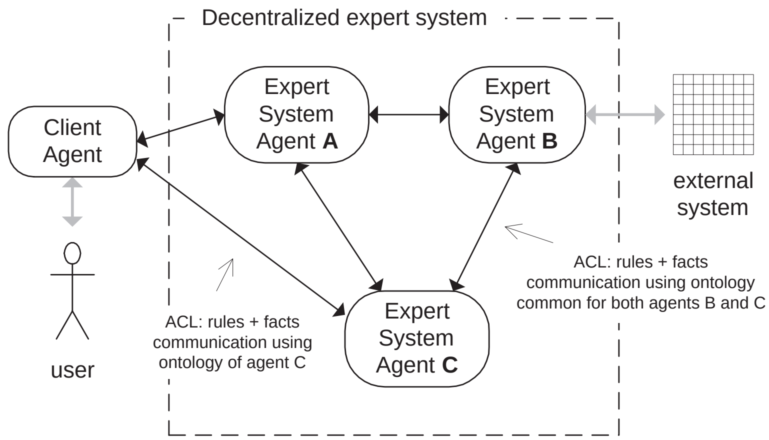

Fig. 3. FIPA based decentralized expert system architecture

nal ontologies (e.g. representing different level of abstraction of the served knowledge, or different terminology for various purposes etc.).

At the physical (real) level the component based decentralized expert system is realized by is realized by a set of inter-operating agents. The main element of the framework is an expert system agent, which serves expert services from given domain and communicates with other expert system agents if necessary. Client agents (agents that communicate with the end user) may use knowledge processing abilities of selected expert system agents, which provide expertise in a particular domain. Depending on the task, they may be used by a human user (e.g. in a diagnostic system) or some external system (e.g. in monitoring or control system). Cooperation of agents is achieved by using a common communication language based on FIPA-ACL. A general architecture of a decentralized expert system system based on FIPA specifications is presented in fig. 3.

5 Decentralised expert system for casting defects diagnosis

The technology described above is used to design a decentralized expert system dedicated to diagnosis of casting defects and aid quality control procedures during the realisation of the foundry production process [4, 6].

The main idea is to integrate heterogeneous data and knowledge sources (sets of decision-diagnostic rules, parameters of the production process, characteristic of used materials, datasets with previous cases describing the history of the production, etc.) that can help an engineer in the realisation of the quality control process, including identification of the defects, search for the causes and point out how to eliminate such situations in the future.

The system follow FIPA standard guidelins, agents communicate in terms of FIPA-ACL to interoperate and exchange or share their knowledge.

5.1 Diagnosis of casting defects

A task of a diagnostic expert system is to assist a process engineer in identification of a casting defect (the type of defect) and to enable him to find the causes of occurrence of this defect with indication of the possible remedy (repair of casting) or measures that should be undetaken to avoid occurrence of such defects in the next run of the process of making the same products [5].

Effective diagnosis should be integrated with a technological process used by a foundry and with applied system (procedures) of quality control. All the information for the diagnosis can be obtained from the quality control system. On the other hand, conclusions produced by the expert system can be used to improve quality control procedures. At the stage of quality control the expert system is used as a tool aiding the decision-making process. The diagnostic system is continuously fed with information on the course and output of the technological process. The diagnosis can be done either on final product or a semi-finished product manufactured at a given stage of the production process.

The knowledge acquired by the system describes the relations that exist between the concepts of TECHNOLOGIES – PRODUCTS – DEFECTS – CAUSES – PROCESS PARAMETERS and are represented by a set rules.

As an example lets consider two types of rules. Rules of the first type enable, at the instant when a defect is noted to occur, concluding about the possible causes of its occurrence. The reasoning is as follows: if defect $DEFECT_i$ has occurred, then its cause can be $CAUSE_1$ or $CAUSE_2$ or ... This knowledge is of a general character and, in same sense is independent on specific features of the production process. Rules of the second type interrelate parameters characteristic of a technological process with potential causes of the defect formation. The reasoning is as follows: if certain configuration of parameters of values ($PARAMETER_1, PARAMETER_2, ...$) has occurred, then this fact can be regarded as a potential threat (or potential cause) $CAUSE_j$ which may finally result in the occurrence of a defect.

These sets of rules forms two knowledge components that can be used to design (by the process of composition) the complex diagnostic system that can be used for different kinds of diagnosis or production process controll and evaluation.

Application of the diagnostic system and its integration with procedures of quality control can be made at following levels:

- The diagnosis of final product. This is a typical diagnostic procedure applied at the stage of making a final product. The system supports the work of a process engineer at the stage of defect identification and searching for its causes. Generally, the reasoning is done using rules of the first type. An output is ”statement of occurrence of a defect” or the decision about possible repair of a product indicating also the causes of the defect.
- Monitoring of production process and aiding the process of decision making. The rules of an expert system are used at the several control and measuring points. Appearance of an undesired configuration of the measured parameters actuates the relevant rules and starts a process of reasoning on possible consequences of such situation. The recommendations of the system may concern further possible corrections in the manufacturing process.

– Recommendations for possible corrections in procedures of quality control. The recommendations may prescribe some modifications of control and monitoring procedures and indicate the stages of a technological process which, when monitored properly, may enable elimination of some causes and reduce the severity of defects. They may also indicate the parameters which, having no significant effect on product quality, can be left unmeasured, reducing cost of measuring procedures.

Extension of the system can be accomplished by introduction of new knowledge components providing knowledge services in another areas (e.g. knowledge about previous cases of the produced goods, knowledge about norms to be in force, makketing and bussines information etc.)

5.2 An example of the knowledge representation and operation of the system

Bellow we show an example how the knowledge of the distributed expert system is represented and how agents can inter-operate and realize reasoning processes.

As the consequence of two kinds of rules present in the system two kinds of agents can be implemented.

– Expert System Agents that possess general knowledge of the casting defects of some kind and their possible causes (rules that describe relation between sets: DEFECTS and CAUSES.
– Expert System Agents that possess knowledge of a fragment of the production process (rules that characterize the production process and describe relation between sets: CAUSES and PROCESS PARAMETERS). This kind of knowledge is specific for the given production process (different parameters can be important and their different values and configurations are acceptable or not). These agents can communicate with the appropriate knowledge and data sources (directly of through wrapper agents) to obtain the actual values of the needed parameters).

Lets consider two Expert System Agents:

– Agent1 which is capable to point out what are causes of cast iron defects. Its knowledge base is a set of rules of a type: *IF* $CAUSE_i$ *and ... THEN* $DEFECT_j$ for each defect it can recognize and reason about. An example of such a rule be: *IF metal-temperature is unacceptable THEN defect can-be shrinkage-cavity*.
– Agent2 which has knowledge of the course of the production process. Its knowledge is a set of facts describing the measured parameters, their values and a set of rules of a type: *IF* $PAREMETER_i$ *and ... THEN* $CAUSE_j$, (e.g. the rule that say what does it mean (in a given production process) that the metal temperature is unacceptable and what are the exact values of a parameter *metal-temperature*).

To prove the hypothesis that defect is *shrinkage-cavity* and the reason is problem with *metal-temperature* Agent1 can ask question directly to the user or can search the directory service to find out if any other agent can prove the fact that *metal-temperature is unacceptable*. If it knows that Agent2 is capable to do so (Agent2 should earlier register in the directory service the entry describing its capabilities) it passes the appropriate FIPA-ACL communicate to prove the desired fact. Agent2 starts it's reasoning process, find out the actual value of the parameter sends back answer with the content describing that the fact was proved or not. Of course there can be situation that Agent2 cannot prove fact directly (the fact can be proved by proving other facts and Agent2 has not appropriate information or reasoning capabilities). So it can ask for a missing fact by sending communicate back to Agent1 or search the its meta-knowledge base or directory service and propagate the distributed reasoning process by requesting another agent to prove the given fact.

The above example describe the case when user via its Client Agent communicates with Ekspert System Agent responsible for solving problem on DEFECTS – CAUSES level. The system starts with the hypothesis about defect and then try to find its causes. The system can also work in other way. Client Agent can communicate with agents responsible for solving problems on CAUSES – PROCESS PARAMETER level. The system can prove the hypothesis about potential causes basing on actual information on process parameters (with no information about defect). If it prove the cause it can communicate with other agents to find out possible defects.

6 Concluding Remarks

Interoperability between knowledge-based systems seems to be of vast importance for development of intelligent information systems working in the global network environment. The proposed architecture of decentralized expert system based on knowledge components may be considered as a step towards an open framework supporting cooperation of expert systems produced by different vendors and providing information and the services in agent-oriented manner.

References

1. J. Dale and E. Mamdani. Open standards for interoperating agent-based systems. *Software Focus*, 1(2), 2001.
2. T. Finin, Y. Labrou, and J. Mayfield. KQML as an agent communication language. In J. M. Bradshaw, editor, *Software Agents*. AAAI/MIT Press, 1997.
3. N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.
4. S. Kluska-Nawarecka, G. Dobrowolski, and R. Marcjan. INFOCAST - a system for quality control procedures and diagnosis of casting defects. *Acta Metallurgica Slovaca*, 7:441–446, 2001.
5. S. Kluska-Nawarecka, T. Wjcik, G. Dobrowolski, and R. Marcjan. Learning analysis of casting defects using the expert system. In S. K.-N. et al., editor, *Simulation Designing and Control of Foundry Processes*, pages 165–174. Foundry Research Institute, 1999.
6. S. Kluska-Nawarecka, T. Wjcik, G. Dobrowolski, R. Marcjan, and E.Nawarecki, editors. *Od pasywnych do aktywnych zrodel danych i wiedzy. Zdecentralizowany system informacyjno-decyzyjny dla wspomagania technologii odlewniczych. (From pasive to active sources of data and knowledge. Decentralised information-decision system to aid foundry technologies*. Institute of Computer Science AGH, Foundry Research Institute, Krakw, 2002.
7. P. O'Brien and R. Nicol. FIPA – Towards a standard for software agents. *BT Technology Journal*, 16(3):51–59, 1998.