Ontology Based Knowledge Management and Learning in Multi-Agent System

Dariusz Choinski and Michal Senik

Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland
{dariusz.choinski,michal.senik}@polsl.pl

Abstract. This paper presents implementation of an algorithm that handles knowledge management and learning processes which are implemented in the existing Multi-Agent System (MAS). Both presented knowledge management and learning mechanisms are based on the implemented ontology mechanisms. Proposed MAS solution is based on the Java Agent Development (JADE) framework which is a Java based tool that supports creation of FIPA (Foundation for Intelligent Physical Agents) compliant agents. It integrates with various different OPC (OLE for Process Control) and database servers simultaneously. OPC is an open automation communication interface that is used to retrieve and analysis of various different real time data from the various different remote process controllers. To integrate OPC automation interface, presented MAS utilize Java Native Interface (JNI) which is a bridge between a Java and native programming languages. Presented MAS establishes database cooperation by means of the NHibernate entity framework which is a .Net platform specific, object relational mapper (ORM). Integration with .Net platform is performed by means of the highly refactored JADE Sharp module add-on which enables creation .Net agents compliant with the JADE framework.

Keywords: Multi-Agent Systems, Java, .Net, JADE, FIPA, OPC, XML, NHibernate, hybrid systems, concurrent programming, knowledge sharing, learning.

1 Introduction

In the agent managed environment reasoning capabilities are far more sophisticated than in the traditional object oriented solutions because in the MAS reasoning process is strongly supported by means of ontology [1], [2]. Conceptually ontology itself is a static description of the surrounding, dynamic environment. It is a formal explicit hierarchical representation of the knowledge about the specific domain based on which MAS will be created and because of that it should be the first element created of each MAS. It is worth mentioning that MAS can utilize various numbers of ontologies which means that it is by default open for the initially unknown modifications or other functional elements such as new agents. Fundamental feature of each MAS is the aspect of ontology based knowledge sharing. In most cases ontology is considered as system knowledge. This knowledge is modelled and
implemented beforehand MAS application and it is known prior to its runtime [3], [4]. Ontology not only describes static properties of the system but also runtime states in which the system can reside as well as the conditions of transitions between those states. Based on the ontology, MAS agents can share their knowledge and work together i.e. cooperate over the integrated system’s problems because knowledge sharing is the natural way for single pieces of MAS to achieve their goals. Technically ontology is a set of simple and more complex rules such as concepts, predicates and actions which can be created in many available programming environments such as Java or .Net. Additionally ontology can be created in the XML language which can be considered as a far more convenient way in comparison to the Java or .Net based solutions because of the fact that XML based ontology can be reused in each mentioned programming environment. Each ontological expression can be organized in a hierarchical structure which means that simpler entities can be nested in more complex entities. Agent’s reasoning capabilities reuse those ontological structures during concurrent cooperation processes over many different integrated system parts which assure high quality and accuracy of the retrieved data. Such approach allows for dynamic management of system knowledge using its static description.

2 Knowledge Ontology Based Memory Management

In many situations information is passed between sender and receiver without any prior knowledge about what is the current state of the receiver especially what is the current state of the receiver knowledge. In case of rather fast data collection the communication process itself might become inefficient. This is because of the fact that large amount of the raw static data will be passed continuously between the sender and receiver unnecessarily increasing the communication workload and message processing time. Memory management concept however enables various pieces of the integrated system to optimize and in effect reduce their need for communication (fig. 1).

![Memory management mechanism](image)

**Fig. 1.** Memory management mechanism

One approach to solve this problem would be to programmatically implement dedicated memory management mechanism and integrate it with an already existing system. Introduced mechanism must be flexible enough to handle various unexpected scenarios in the integrated system. Implementation of such memory management mechanism ought to be resistant to the possibility of dynamic integrated system