SemanticAgent, a Platform for the Development of Software Agents.

Percival Lucena, Dilvan De Abreu Moreira, Carlos Alberto Estombelo Montesco

Instituto de Ciências Matemáticas e Computação - Universidade de São Paulo
Av. do Trabalhador São-Carlense, 400 - Centro - Cx. Postal 668
São Carlos - São Paulo - Brasil - 13560-970

{percival,dilvan}@icmc.usp.br, estombelo@universia.edu.pe

Abstract: *This paper presents a new software agent platform that is able to manipulate knowledge and execute actions based on requests made on restricted natural language. A new model for representing knowledge and actions, based on extended ontologies, is presented and a prototype that implements the ideas is discussed as well.*

1. Introduction

The construction of systems based on intelligent agents is a daunting task that involves aspects such as agent communication, planning, work division, cooperation, representation and manipulation of knowledge, among other activities. In order to ease the creation of agent based systems, agent platforms provide some services that allow developers to build solutions without the need of worrying about every implementation detail.

This paper is divided as follows: section 2 discusses agent communication and knowledge representation, section 3 discusses the *SemanticAgent Platform*, the main focus of this work; at last section 4 draw some conclusions about the presented work.

2. Agent Communication and Knowledge Representation

A well-adopted paradigm of agent communication consists in the use of message exchange. Agent communication languages (ACLs) that support this paradigm usually rely on content languages to represent the knowledge logics and ontologies to represent knowledge semantics.

Content languages are usually based on formal logic, which is well suited for technical and expert persons, but fall short on the abilities of ordinary software agent users. Content languages that can express unambiguous natural language requests can be considered a more adequate tool for end users and developers who require dynamic interaction with software agents. The Universal Networking Language (UNL) and Universal Communication Language (UCL) appear as natural candidates for such use.

UNL is an artificial language used to represent restricted sentences expressed in different natural languages [Serasset et al 2000].Once a sentence or text is translated to UNL, it can be processed or translated back to one of the supported languages. Unfortunately, most of the UNL converters are still under development and are not publicly available. Due to this, we have developed a simple implementation of the UNL standard called Universal Communication Language (UCL) [Montesco and Moreira 2002] based on the Thought Treasure natural language processing tool [Mueller 1998].

In addition to a communication language, a knowledge model is also essential to allow agents to process information. Ontologies, which are commonly used to help determine communication semantics among different sources of information, can also work as a knowledge model itself for software agents [Fensel 2001]. We go further to propose the extension of ontologies as a way of representing agent behaviors, i.e. the actions agents can execute.

3. SemanticAgent Platform

SemanticAgent is a new platform for the development of agent applications, which exploit innovative ideas such the use of UCL as agent communication content language to express user requests. The SemanticAgent default knowledge base is a semantic network that extends the Thought Treasure common sense ontology A Knowledge Base Manager Agent (KBMA) manipulates the platform knowledge base and provides a simple information processing mechanism founded on path-based inference.

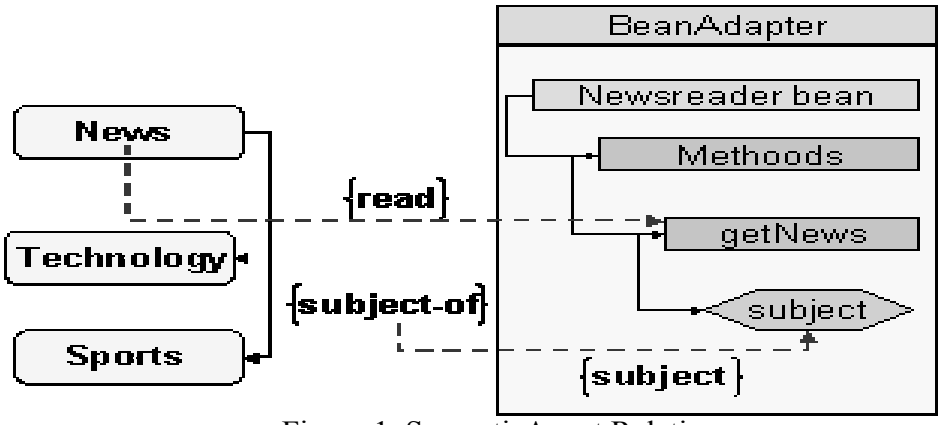

Figure 1. SemanticAgent Relations

Besides the traditional relations provided by most of the semantic network systems, the knowledge base also provides *action links*, which associate concepts to methods or parameters of software components as described in *Figure 1*. Action links (slashed lines), associate concepts and relations of the semantic network to component structures described by software component adapters.

The SemanticAgent platform can process requests in UCL generated by a English to UCL converter or manually created. The UCL interpreter is able to query and manipulate the system global knowledge base and execute actions implemented by software components as illustrated in *Figure 2*.

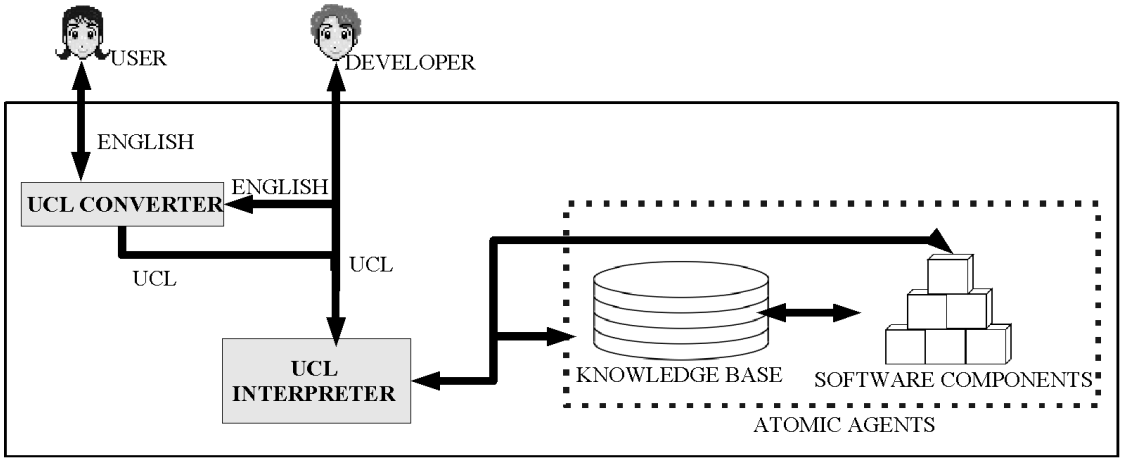

Figure 2 – Semantic Agent Platform Architecture

The SemanticAgent Server accepts messages which content language is UCL or English. In later case, the message is converted to UCL before it can be processed by the system. The UCL Interpreter Agent is responsible for receiving UCL requests, activating the proper atomic agents as needed. A simple pattern matching mechanism tries to associate the incoming messages to the activation conditions of the UCL interpretation scripts that are available on the platform. If a script is executed successfully, its result is sent back to the user who originated the request.

The UCL interpretation scripts describe the actions that should be processed to satisfy UCL requests. Those scripts act as templates for a certain category of event. The information processing used on the scripts usually benefit from generic services provided by the knowledge base manager and the component manager. Some of the scripts developed for the SemanticAgent Platform are listed on Table 1.

Table 1. UCL Interpretation Scripts

Category	UCL Script	Sample Case of Use
QUERY KB - Queries the Knowledge Base.	Query definition of a concept	*What is an apple?*
	Query person information	*Who is "Bugs Bunny"?*
	Query elements that match a criteria	*Agent, list red planets.*
VALIDATE – Validates an assertion.	Validation of facts based on ontology structure.	*Banana is a fruit.*
	Validation of facts based on KB rules.	*Do birds fly?*
INFORM – Inserts new information on the KB.	Insert a new Concept on Knowledge Base	*Insert student "Paulo Cunha"*
	Insert a new Assertion on Knowledge Base	*Grades are greater than zero.*
ACTION - Executes an Action.	Executes a component method.	*Get last mail from Monica*

The UCL interpretation scripts currently developed aim to be as generic as possible so they can cover frequent use cases. The *query definition script*, for instance can be used to obtain information about any given concept that is stored on the knowledge base and the *action script* can be used to activate any action that can be matched to the concepts embedded in the UCL request message. In addition to the existing scripts, new ones can be easily constructed to match a specific situation needed by the users of the SemanticAgent Platform.

When the UCL Interpreter receives a behavior execution request, it asks KMBA whether there are actions associated to the concepts in the message. If any of the message concepts is associated to a method from a software component, the KBMA can find the associated method and the necessary parameters. After that, the UCL Interpreter Agent asks the Component Manager Agent to instantiate the given software component. and to execute the desired method with the appropriate parameters.

In order to obtain results from the capabilities implemented by SAS, some tools were developed to allow users and developers to interact with the platform. TalkAgent is an application aimed to help end users while SAS IDE is targeted at software developers.

TalkAgent is a web-based application that allows users to interact with the platform through requests in restricted natural language, one sentence at time. TalkAgent user requests are sent to the SemanticAgent Server, which converts them to UCL and later process them using the available UCL interpretation scripts. Currently, the TalkAgent application allows users to query the platform knowledge base, insert simple information and request the execution of useful actions.

The SemanticAgent IDE is a client server application that allows developers to connect to SemanticAgent Server and manipulate its knowledge base and repository of software components (Javabeans). The Component Editor allows the control of the software components stored in the SemanticAgent Server. The tools provided by the IDE can easily edit relations between the components, methods and parameters. New components can be aggregated to the Component Repository dynamically and their execution can be tested directly from the IDE.

5. Conclusions

The SemanticAgent platform provides a set of tools to create agent applications that process knowledge and execute actions. Much work is still needed to fully explore this paradigm and investigate its usability. For those interested in testing the prototype and return feedback its source code and binaries are available at http://talkagentfw.sf.net.

References

[Fensel 2001] Dieter Fensel. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer Verlag, Berlin, 2001.

[Montesco and Moreira 2002] Carlos Montesco and Dilvan Moreira, UCL - Universal Communication Language, Proc. of the First International Workshop on UNL, other Interlinguas and their Applications, Las Palmas Spain, June 2002, pp. 33-37.

[Mueller 1998] Erik T Mueller, Natural Language processing with ThoughtTreasure. New York, 1998, on-line: http://www.signiform.com/tt/book

[Serasset et al 2000] Gilles Serasset, Christian Boitet- On UNL as the future "html of the linguistic content" - Coling 2000 - The 18th International Conference on Computational Linguistics - Saarbrücken, Germany, 2000.