BDI Architecture Based on XML for Intelligent Multi-Agent Systems

Sang-wook Lee[a], Ji-hyun Yun[a], Il-kon Kim[a] and Hune Cho[b]

[a]*Department of Computer Science, Kyungpook National University*
1370 Sankyuk-dong, Buk-gu, Daegu, 702-701, Korea
Tel: +82-53-950-6376, Fax: +82-53-950-6369,
E-mail: {leesw, yjh9626}@cs.knu.ac.kr, ikkim@knu.ac.kr

[b]*Department of Medical Informatics, Kyungpook National University*
50 Samduk-2ga, chung-gu, Daegu, 700-721, Korea
Tel: +82-53-420-6051, Fax: +82-53-420-6059, E-mail: hunecho@knu.ac.kr

Abstract

Many intelligent agent systems are known to incorporate BDI architecture for cognitive reasoning. Since this architecture contains all the knowledge of world model and reasoning rule, it is very complex and difficult to handle.

This paper describes a methodology to design and implement BDI architecture, BDIAXml based on XML for multi-agent systems. This XML-based BDI architecture is smaller than any other BDI architecture because it separates knowledge for reasoning from domain knowledge and enables knowledge sharing using XML technology. Knowledge for BDI mental state and reasoning is composed of specific XML files and these XML files are stored into a specific knowledge server. Most systems using BDIAxml architecture can access knowledge from this server.

We apply this BDIAXml system to domain of Hospital Information System and show that this architecture performs more efficiently than other BDI architecture system in terms of knowledge sharing, system size, and ease of use.

Keywords:

Multi-agents; Reasoning; Belief; Desire; intention; XML; Hospital Information

Introduction

Owing to appearance of agent, many systems in complex environment are being reconstructed to agent systems. Because we expect them to achieve our purpose using intelligent characteristics of agent such as learning, reasoning, planning and so on [1].

Most Intelligent agents have an architecture based on belief, desire and intention (BDI) methodologies. The BDI model comes from artificial intelligence research over the past twenty years, and has proven to be the most robust and flexible model for intelligent agent systems [2].

Former agent systems based on BDI architectures are found to large because they have their own belief sets and intention sets without being shared.

As the growth of Internet and increase of system resource sharing, we have experienced knowledge sharing through Internet. Therefore, we are focusing on efficient knowledge sharing, which is one of the key factors for reducing of system size.

XML provides possibility for information to be transferred among multiple programs or another computers and makes it easy to store information as hierarchical structure [3]. So we use it to eliminate the complexity of knowledge implementation and to support standardization of beliefs, intentions and so on.

About knowledge sharing, research for interoperability using ACL is in progress [4]. This is due to sociability of agent. But, there are some problems in this case. Details will be discussed in section 3.

We reconstruct Belief sets and Intention sets into that of XML type for sharing and store these into the knowledge server. Other systems using this architecture can access and utilize the knowledge server via Internet.

We propose a methodology to design and implement BDI architecture, BDIAXml for composition of efficient intelligent multi-agent system using advantage of XML taken notice in the latest few years.

In Section 2, we present traditional BDI architecture. Section 3 describes the major technical characteristics of BDIAXml system, and Section 4 presents an example using this architecture in medical domain that shows an efficient implementation of this architecture.

Finally, In Section 5, we summarize advantages and disadvantages of BDIAXml system and propose the future works.

BDI Architecture

Recently, the repressed part of artificial intelligence is being activated by appearance of agent.

On the appearance of intelligent agent, people want the program to meet the requirements so that they are concerned about abilities of agent such as reasoning and learning and so on.

In the beginning, they solved this problem using Expert System or Rule-Based System. But, under the rapidly changing circumstances, the system is required to change appropriate for dynamic environment, so that PRS (Procedural Reasoning System)[5] was developed following the trend of the times.

Afterwards, on an appearance of Object Oriented Programming methodology, several important systems have been influenced by this OOP. Then UMPRS(University of Michigan PRS)[6], JAM[7], Jack[2] and so on were followed.

A common feature of such reasoning systems is the use BDI architecture. This paradigm has been proposed to model human cognitive behavior in terms of three mental states i.e. belief, desire, and intention. It is believed that one's goal-adoption behavior largely depends on its mental states [8].

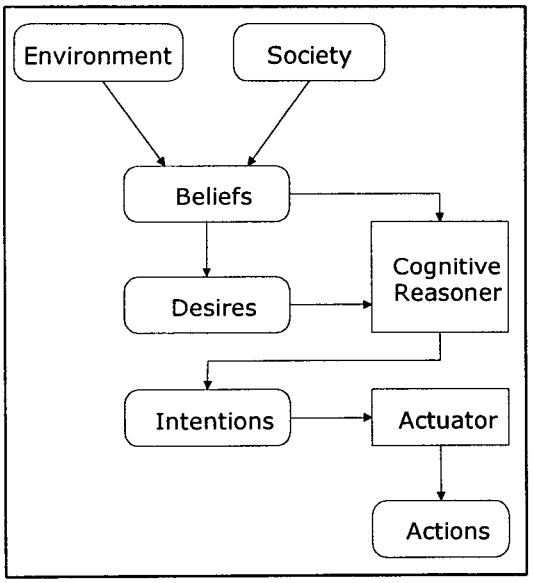

Figure 1- BDI Architecture

BDI architecture (see Figure 1) has beliefs about the world and desires to satisfy, driving it to form intentions to act. An intention is a commitment to perform a plan. The BDI agent executes the steps of the plans that it has adopted as intentions until further deliberation is required. A step in a plan must be able to add a goal to the agent itself, to change its beliefs, to interact with other agents, and to perform any other atomic action on the agent's own state or the external world.

In reaction to an event, for example, a change in the environment or its own beliefs, a BDI agent adopts a plan as one of its intentions [8].

BDIAXml System

BDIAXml

Owing to development of BDI model and multi-agents system, many systems were transformed into intelligent multi-agent systems. So they became huge systems with expertise that enables more work to do and a society of agent system became important.

Agents communicate with each other through message passing and co-operate, but theory and practice do not always coincide because of many reason, for instance, politic problems or shortage of system resource, non-cooperation of different agent systems are one of situations occurred frequently. Most multi-agent systems need to include one or more intelligent agents. However, if an intelligent multi-agents system has knowledge that includes beliefs and intention to the specific goal for all agents, the size of system must be large.

So, we designed the BDI Architecture based on XML technology for the intelligent multi-agent system. This is lightweight and needs new design in respect of knowledge sharing.

Knowledge server has all possible predefined Belief sets and Intention sets with types of XML. Each BDIAXml (We pronounce this system as Bi-Di-Aks) system accesses XML knowledge server via network and brings necessary information from it. (see Figure 2)

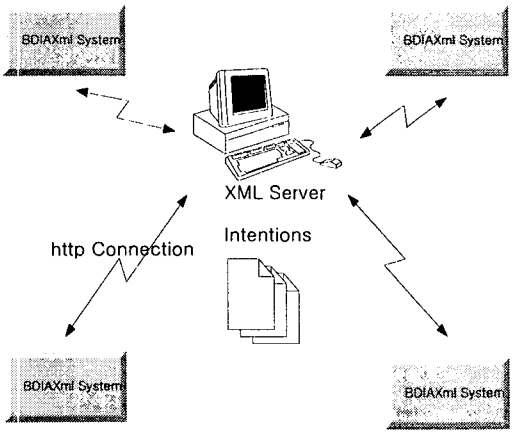

Figure 2- Knowledge Server and BDIAXml Systems

BDIAXml system that includes a XML parser utilizes knowledge stored with type of XML in order to achieve intended goal.

At first when BDIAXml system starts, this system establishes an initial state such as initial desire, initial belief knowledge server address, and so on. Then it connects to knowledge server host and makes DOM trees for the relevant files. Using this DOM trees, BDIAXml system brings out elements of XML files and uses this information to reasoning. When to progress its reasoning, BDIAXml system plans new sub goals and actuates some actions.

XML Category

In BDIAXml system, the most important parts are a DTD definition of each type, which includes belief, intention, and so on and should support a reasoning mechanism appropriate to this condition.

We classify XML documents of BDIAXml system into 6 types

(We show DTD definitions and examples of some XML files in appendix of this paper.)

They are as follows:

- Belief Set

  : Each belief element has personal knowledge about the world.

- Belief List

  : Information for the belief set.

- Intention Set

  : Each intention element has personal knowledge to achieve specific goal.

- Intention List

  : Information for the each intention set.

- Initialize XML file

  : Information to initialize the BDIAXml system.

- Result XML file

  : Verification data for the practical result of BDIAXml system.

For starting, BDIAXml System requires the initialize.xml file, which includes initial belief, initial goal, knowledge server host address, and so on. And then BDIAXml System can set its own Belief with Belief Set and Belief List. As in the same manner it will choose proper plans to achieve current goal with Intention Set and Intention List.

Finally, if BDIAXml System solves all desires of user, it generates a result.xml file.

Reasoning in BDIAXml

Former BDI architectures used the data structure called stack in order to achieve intended goals using push-pop methodology. They executed popped goals from the stack, and if current goal generates some sub goals, these are pushed to the stack. They iterate this kind of push-pop operations until stack is empty.

But because this causes much trouble of time and space, BDIAXml uses a GDL (Goal Direction Linked-list) based on the data structure called linked-list.

This is not a mode of push-pop for goal but a mode of appending, which adds sub-goals after corresponding goal. It is efficient to control these appending and deleting operations in consecutive order.

If initial states are suggested, BDIAXml system accesses XML knowledge server via network and verifies current knowledge state. And then system accesses the intention list in order to search an appropriate method matched with an initial state. Intention list helps to choose a suitable intention and it has information about what intention is suitable when a specific condition is proposed. Using this information, we update GDL. If current intention is a compound of actions, BDIAXml system will actuate these actions and modify the result XML file. However, if current intention is to create sub-goals, BDIAXml system will modify the result XML file and do repetitive operations just like previous job for the new sub-goals.

BDIAXml works until the initial desire will be achieved. Finally it accomplishes GDL and generates a result.xml file, and actuates all operations required to achieve initial goal.

Example

Using BDIAXml, this example demonstrates how efficient and easy it is to find a proper medical treatment of patient ID (PID) in Hospital information system.

Initial assumptions are as follows:

1. Hospital information systems have hospital information XML data files in its DBMS system or knowledge server.
2. Hospital information systems also have evidence information of many kinds of treatment as XML data files.
3. We have only Patient ID (PID) and want to select a cure for him.

Initial mental state is composed of belief that is "PID" and desire that is "Find treatment with PID".

As a way to solve this goal, we need to know what disease patient have from the patient's clinical record based on PID. To find it, we search patient information (PInfo) XML file to include patient's disease information in patient's clinical record. From this XML file we become to know patient's disease name matched with PID. Finally we search an appropriate treatment of patient's disease from initial belief "PID". This sequence of operations is performed by combination of actions. These have been generated by reasoning ability of BDI architecture.

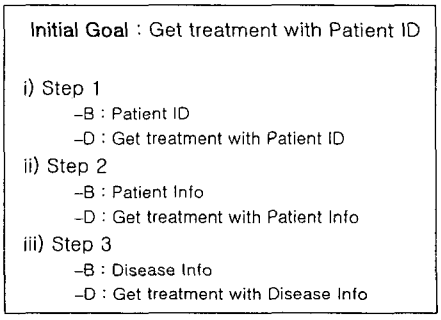

Figure 3- Step for Example

Figure 3 shows three abstract steps what desire matched with belief of medical information this example can have in order to search a treatment with PID.

Defined steps can be subdivided by what intentions desire is composed of. We can reason from any step appropriate for achieving desire matched with belief.

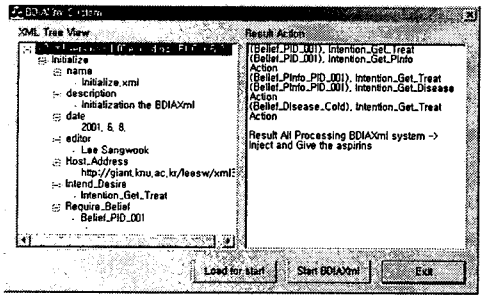

Figure 4- Result of BDIAXml system for Example

Figure 4 shows that initial information in Initialize.xml has Host address, Intend_Desire, Require_Belief, and so on from left window. From right window we can see a list of belief and desire, actions for achieving each desire, and final result generated by executing this example.

```
<?xml version="1.0" encoding="EUC-KR"?>
<?xml:stylesheet type="text/xsl" href="BDIAXml.xsl"?>
<!DOCTYPE Result_set SYSTEM "Result.dtd">
<!-- Result.xml -->
<Result_set>
  <name>Result.xml</name>
  <description>Result for Goal Intention_Get_Treat</description>
  <date>2001. 6. 8.</date>
  <editor>BDIAXml</editor>
  <Host_Address HREF="http://giant.knu.ac.kr/leesw/xml3/"
      TITLE "http://giant.knu.ac.kr/leesw/xml3/">
      http://giant.knu.ac.kr/leesw/xml3/
  </Host_Address>
  <Intend_Desire>Intention_Get_Treat</Intend_Desire>
  <Require_Belief>Belief_PID_P011</Require_Belief>
  <Process_Rule>
    <Process>
      Intention_Get_Treat
    </Process>
    <Process>
      Intention_Get_Treat->Intention_Get_PInfo->Intention_Get_Treat
    </Process>
    <Process>
      Intention_Get_Treat->Intention_Get_PInfo->Action->Intention_Get_Treat
    </Process>
    <Process>
      Intention_Get_Treat->Intention_Get_PInfo->Action->Intention_Get_Treat
    </Process>
    <Process>
      Intention_Get_Treat->Intention_Get_PInfo->Action->Intention_Get_Treat->
      Intention_Get_Disease->Intention_Get_Treat
    </Process>
    <Process>
      Intention_Get_Treat->Intention_Get_PInfo->Action->Intention_Get_Treat->
      Intention_Get_Disease->Action->Intention_Get_Treat
    </Process>
    <Process>
      Intention_Get_Treat->Intention_Get_PInfo->Action->Intention_Get_Treat->
      Intention_Get_Disease->Action->Intention_Get_Treat
    </Process>
    <Process>
      Intention_Get_Treat->Intention_Get_PInfo->Action->Intention_Get_Treat->
      Intention_Get_Disease->Action->Intention_Get_Treat->Action
    </Process>
  </Process_Rule>
</Result_set>
```

Figure 5- Result.xml for Example

Result.xml file of Figure 5 is a set of verification data as the practical result executed and reasoned by BDIAXml system. It presents all information, such as host address, initial belief, initial desire and GDL for processing

Process tags of Figure 5 show a sequence of sub-goals generated in this example and actions for achieving these sub-goals.

We have compared BDIAXml with UMPRS, JAM and Jack in terms of knowledge sharing.

Table 1- Comparison of BDIAXml and other systems

	BDIAXml	UMPRS(or JAM)	JACK
Knowledge sharing	O	X	X
System size	small	big	big
Easiness of use	easy	hard	hard
Operation processing ability	low	high	more high

Table 1 shows comparison of BDIAXml and with other systems.

As this table shows, BDIAXml is better in knowledge sharing, system size, and easiness of use, but the ability of operation processing is some low. We are strengthening actuator of BDIAXml.

Conclusion and the future works

This BDIAXml system is able to support and share the domain specific knowledge via Internet. Anywhere or anytime, if system is connected to the Internet, it accesses the XML knowledge server and retrieves relevant information.

Moreover, BDIAXml is smaller than any other BDI architecture because of using the methodology we propose, which separates knowledge from system. But it is somewhat disadvantageous that this system takes more time in doing connection via network.

Through this BDIAXml system, experimental results strongly indicate that BDI architecture based on XML technology has shown optimal performance for the task of knowledge sharing. We have confirmed that separating of mental state from system and presenting new reasoning model enable smaller size than any other one.

Finally, if so many XML knowledge servers are created over the web, it will need a BDI information broker, which plays a role in finding appropriate knowledge server and in gathering information from knowledge server.

Appendix

```
<!ENTITY % link
                ' XML-LINK CDATA #FIXED "SIMPLE"
                  HREF CDATA #REQUIRED
                  TITLE CDATA #IMPLIED
                  ROLE CDATA #IMPLIED
                  CONTENT-TITLE CDATA #IMPLIED
                  CONTENT-ROLE CDATA #IMPLIED
                  BEHAVIOR CDATA #IMPLIED
                  INLINE (TRUE|FALSE) "TRUE"
                  SHOW (EMBED|REPLACE|NEW) "REPLACE"
                  ACUATE (AUTO|USER) "USER" '
>

<!ELEMENT Initialize (name, description, date, editor,
                Host_Address, Intend_Desire, Require_Belief)>

<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT editor (#PCDATA)>
<!ELEMENT Host_Address (#PCDATA)>
<!ATTLIST Host_Address %link;>
<!ELEMENT Intend_Desire (#PCDATA)>
<!ELEMENT Require_Belief (#PCDATA)>
```

Figure 6- A DTD for initialize.xml

```
<!ENTITY % link
                ' XML-LINK CDATA #FIXED "SIMPLE"
                  HREF CDATA #REQUIRED
                  TITLE CDATA #IMPLIED
                  ROLE CDATA #IMPLIED
                  CONTENT-TITLE CDATA #IMPLIED
                  CONTENT-ROLE CDATA #IMPLIED
                  BEHAVIOR CDATA #IMPLIED
                  INLINE (TRUE|FALSE) "TRUE"
                  SHOW (EMBED|REPLACE|NEW) "REPLACE"
                  ACUATE (AUTO|USER) "USER" '
>

<!ELEMENT Belief_list (name, description, date, editor, Belief_sets)>

<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT editor (#PCDATA)>

<!ELEMENT Belief_sets (Belief_set*)>
<!ELEMENT Belief_set (#PCDATA)>
<!ATTLIST Belief_set %link;>
```

Figure 7- A DTD for Belief list

```
<!ENTITY % link
                ' XML-LINK CDATA #FIXED "SIMPLE"
                  HREF CDATA #REQUIRED
                  TITLE CDATA #IMPLIED
                  ROLE CDATA #IMPLIED
                  CONTENT-TITLE CDATA #IMPLIED
                  CONTENT-ROLE CDATA #IMPLIED
                  BEHAVIOR CDATA #IMPLIED
                  INLINE (TRUE|FALSE) "TRUE"
                  SHOW (EMBED|REPLACE|NEW) "REPLACE"
                  ACUATE (AUTO|USER) "USER" '
>

<!ELEMENT Intention_list (name, description, date, editor,
                Intention_sets)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT editor (#PCDATA)>
<!ELEMENT Intention_sets (Intention_set*)>
<!ELEMENT Intention_set (Intention_Filename, Require_Belief*,
                Intention_goal)>
<!ATTLIST Intention_set %link;>
<!ELEMENT Intention_Filename (#PCDATA)>
<!ELEMENT Require_Belief (#PCDATA)>
<!ELEMENT Intention_goal (#PCDATA)>
```

Figure 8- A DTD for Intention list

```
<?xml version="1.0" encoding="EUC-KR"?>
<?xml:stylesheet type="text/xsl" href="BDIAXml.xsl"?>
<!DOCTYPE Intention SYSTEM "Intention.dtd">
<!--  Intention_Get_Treat_use_Disease.xml  -->

<Intention>
   <name> Intention_Get_Treat_use_Disease </name>

   <description>
      intention to get Treatment using Disease name.
   </description>

   <date> 2001. 5. 9. </date>

   <editor> Sangwook. Lee. </editor>

   <Request_Belief>Belief_Disease_Cold</Request_Belief>
   <Request_Belief>Belief_Disease_Fracture</Request_Belief>

   <Intention_goal>
      Intention_Get_Treat_use_Disease
   </Intention_goal>

   <Rules>
      <Action>
         <Describe>Search Information</Describe>
         <Instruction>Search_Info</Instruction>
         <Parameter>Treatment</Parameter>
      </Action>
      <Action>
         <Describe>Show Information</Describe>
         <Instruction>Show_Info</Instruction>
         <Parameter></Parameter>
      </Action>
   </Rules>
</Intention>
```

Figure 9- An example of Intention set

References

[1] Gerhard Weiss, Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence, Massachusetts Institute of Technology, 2000

[2] Paolo Busetta, Ralph Ronnquist, Andrew Hodgson & Andrew Lucas, Using Java for Artificial Intelligence and Intelligent Agent Systems, Agent Oriented Software Pty. Ltd., 1999

[3] Frank Boumphrey, Professional XML Applications, wrox, 1999

[4] Foundation for Intelligent Physical Agents, Specification, 2000, http://www.fipa.org

[5] Michael P. Georgeff and Amy L. Lansky, Procedural knowledge, Proceedings of the IEEE, 1986

[6] Jacho Lee, Marcus J. Huber, Edmund H. Durfee, Patrick G. Kenny, UM-PRS: An Implementation of the procedural reasoning system for multirobot applications, The University of Michigan, 1994

[7] Marcus J. Huber, JAM Agents in a Nutshell, Intelligent Reasoning Systems, 1999

[8] Hrushikesha Mohanty, Manas Ranjan Patra, K Sagar Naik, Influencing: A Strategy for Goal Adoption in BDI Agents, University of Hyderabad and University of Aizu, 1997