Autonomous Software Agent for Localization

Gopal Sakarkar
MCA Department,
GHRIIT
Nagpur, India
g.sakarkar@gmail.com

V. M. Thakar
Computer Department
S. G. B. Amravati University
Amravati, India
vilthakare@yahoo.co.in

Abstract— **Today alone with the popularization of Agent technologies, software Agent technology is being applied into many systems, like ERP systems, CRM systems, network management, air-traffic control, telecommunication, and electronic commerce. The paper present a model of autonomous agent architecture and proposed the implementation approach for Autonomous Counseling Agent within educational organization, which is capable of automatic detection of proficient students for particular course and identify challenges that must be met for its implementation using the concept of localization.**

Keywords - Software Agent, Localization, Autonomous software agent, Agent Technology, Agent Communication Languages.

I. Introduction

Because of the popularization of internet technology and applications, the entire world is become a globe village. As facing furious and fluky competition day by day, enterprises have to make suitable fine-tune and reaction on time to adapt themselves to the market changes. Autonomy is one of the main features of agents in Agent System. It refers to the ability of an agent to exert control over its own goals and actions.

This paper presents the development of a Localization model that aims to develop the user-friendly online graphical user interface (GUI). For that localization might not differ much from a dynamic concept of translation. Localization can be describe as "the concept [which] involves the convergence of language and technology, necessary to adapt any kind of digital content (software, websites, portals, online support systems, etc.) to a given target local ",

i.e localization involved the translation of *one* text into *one* multicultural local. John Graham (BDU, Stuttgart) defined localization as "Localization involves taking a product and making it linguistically and culturally appropriate to the target local (country/region and language) where it will be used and sold"[1].

The system is proposed in this paper is an open system where autonomous agents are capable of flexible autonomous action in order to meet their desire objectives. The agent exhibits pro-activity, sociability and learning capability. Pro-activeness is attributed to the goal directed behavior of each agent. Sociability is referred to the interaction with other agents to get its objective done and learning ability attributed to the capability to adapt to user's language preferences and new languages. As describe in previous work an autonomous agent is define as "An Autonomic Software Agent means that the agent has its own behavior control mechanism and it can base on its internal status and working environment information to carry out its own tasks. Alone with the task changed, it can automatically upgrade its own configuration to complete the new tasks" [2].

II. PROPOSED SYSTEM ARCHITECTURE

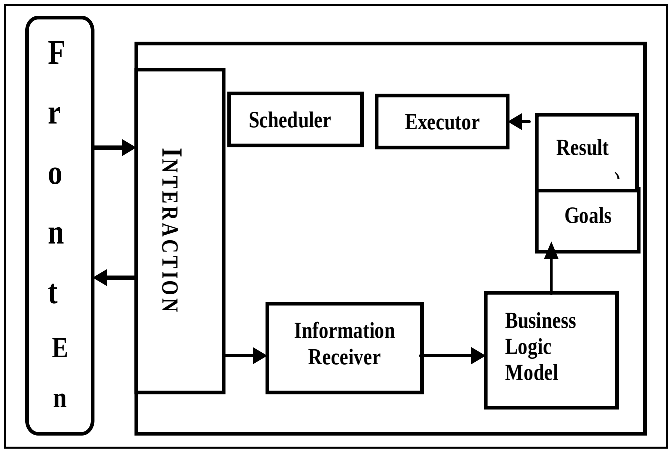

Figure. 1. System architecture

2.1. *Overall Architecture*

The architecture of the agent system comprises three primary components:

1. Front-end: A web interface handles all of the user interaction. It will be use any scripting language for GUI.

2. Back-end: The system engine is where the agents actually "live" and interact with one another. The back-end is implemented in Java.

3. Auxiliary components: There are also several auxiliary components that generate the visual display files, send out pages to users and implement the login process.

The architecture was designed to be fairly well modularized, so that changes to one component would cause at most minimal changes to be made to the others. The interfaces between components were designed to be as general and flexible as possible. The architecture of the system is illustrated in Figure 1. The agents collaborating in the system are

978-1-4244-4711-4/09/$25.00 ©2009 IEEE

information receiver agent, business logic module and executor and scheduler agent. A brief description of each component is as follows.

2.2 *Information receiver agent*

In which Information receiver agent receives the information from interaction section of the architecture. It accept the IP address of request user for finding out the localize format out put.

2.3 *Business Logic Module*

This is the most important part of architecture. It contains the user defined logic to convert the input IP address in to proper user friendly GUI.

2.4 *Goal/Result*

The goal(s) will be loaded into Business Logic module is nothing but the parameters to find out proper results.

2.5 *Executer and Scheduler agent*

The Executor Module gets forward the proper result to scheduler agent and finally Scheduler agent displays the matching result to requested user.

III. SYSTEM WORKING

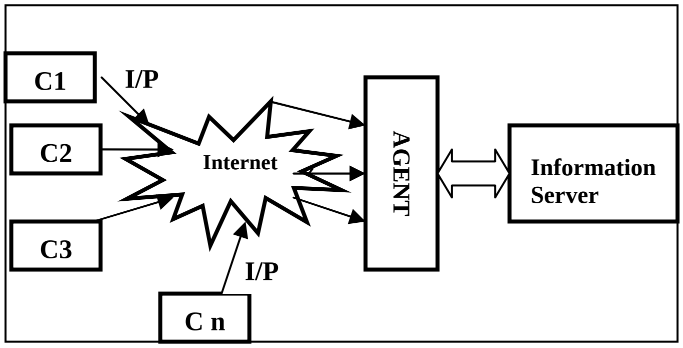

Figure 2. Schematic diagram of localization with an agent

The C1, C2,C3….Cn are the number of clients requesting for the information from the server. As the user sends the request for the information it also sends the particular IP address from it computer. As IP is unique identification number of his/her computer, from that IP address number agent can find out the location of particular client. After finding the location area of the client request, on that particular IP address the autonomous agent will be reply to the user in his/her local languages. Here, the concept of localization is used to convert the web base information into user's local languages. The next research work is to implement and find out technical aspect of these localization concepts.

IV. PROPOSED AGENT TECHNOLOGIES

Software agent technology is a rapidly developing area of research and probably the fastest growing area of information technology [3]. Following are the some proposed agent technologies for autonomous software agent.

4.1 *WOL-P*

Web Ontology Language for Protocols (WOL-P) has been recommended as a technology that will contribute to software agents' interaction with Web services and standard Web service technologies. It provides a framework in which to (1) develop, (2) specify and combine and (3) enact Commitment-based protocols.

OWL-P components include (1) an OWL ontology to support the specification of protocols elements, (2) a rule-based representation of protocols using for rule expression, (3) an algorithm for combining protocols that is based on algebra and offers a sound mathematical foundation and (4) an architecture that supports run time activities.

OWL-P may contribute to autonomous software agent Web service interactions in the cross cutting area of interactions and protocols. OWL-P can offer both a language and a framework for defining and executing the agent-Web service interactions [4].

4.2 *XML and RDF*

XML technologies are being used in the presented agent architecture. The use of agent and XML technologies in autonomous architecture can help secure the interoperability and reusability factors.

XML plays an important role in the success of a management model based on grids of autonomous agents in what it refers to the representation and passing of management information among the several components of the grid [5].

Resource Description Framework (RDF), it provides interoperability between applications that exchange machine understandable information on the Web. RDF emphasizes facilities to enable automated processing of Web resources. RDF metadata can be used in a variety of application areas, digital library by intelligent software agents to facilitate knowledge sharing and exchange etc [6].

4.3 *JINI*

Jini technology is a service oriented architecture that defines a programming model which both exploits and extends Java technology to enable the construction of secure, distributed systems consisting of federations of well-behaved network services and clients. Jini technology can be used to build adaptive network systems that are scalable, evolvable and flexible as typically required in dynamic computing environments.

4.4 *UPnP*

Universal Plug and Play (UPnP) technology is a distributed, open networking architecture that employs TCP/IP and other Internet technologies to enable seamless proximity networking, in addition to control and data transfer among networked devices in the home, office, and public spaces[7].

5. Agent Communication Languages

For successful co-operation to occur between agents, they have to be able to communicate among themselves. To enable this communication a different Agent Communication Language (ACL) are required. This paper focuses on different ACL Languages [8].

5.1 KQML

The Knowledge Query and Manipulation Language (KQML) [9] is probably the most widely known ACL. KQML is a message-based language for agent communication, and a message handling protocol to support run-time knowledge sharing among agents. It defines a common format for agent messages and consists of a message layer, a communication layer and a content layer [10]. It is a language that is designed to support interactions among intelligent software agents. It was developed by the ARPA supported Knowledge Sharing Effort and separately implemented by several research groups. It is a language and a set of protocols that support computer programs in identifying, connecting with and ex-changing information with other programs.

Role of KQML

a) KQML draws on work in both distributed systems and distributed AI and offer a level of abstraction that should be useful to both.

b) It provides an abstraction of a process as an information agent as knowledge –base system.

c) It's implementation have used standard communication and message protocol as a transport layer, including TCP/IP, email, Linda, HTTP, CORBA, etc

Drawbacks of KQML

a) A single research group for a particular project did not define it. It was created by a committee of representatives from different projects, all of which were concerned with managing distributed implementations of systems. One was a distributed collaboration of expert systems in the planning and scheduling domain. Another was concerned with problem decomposition and distribution in the CAD/CAM domain. A common concern was the management of a collection of co-operating processes and the simplification of the programming requirements for implementing a system of this type. However, the groups did not share common communication architecture. As a result, it does not dictate particular system architecture, and several different systems have evolved.

One drawback of KQML-based agent communication is a lack of standardization in the actual transport of messages. In addition, the semantics of the language have not been rigorously defined, which can lead to interoperability issues.

5.2 CG-KQML+

To overcome the KQML's drawbacks, the new communication protocol and communication language between agents and between agents and users, proposed a new communication language, called CG-KQML+ that is an extension of KQML Language. The authors describe the use of CG-KQML+ in MAS called POSTAGE, which aims at helping users in their correspondence task. In POSTAGE, software agents manage administrative correspondence on behalf of and in cooperation with their users. Users and agents have interactions, which respect administrative correspondence rules. A POSTAGE agent is responsible for sending the generated message to the addressee's POSTAGE agent. The authors present the second version of POSTAGE, which is implemented using the Prolog+CG language [11].

5.3 PIGEON

The author proposed the Pigeon, a specification language with customization and high descriptive power needed for specifications of mobile agent application. They describe the Customizability by representing the semantic in a logical framework called RL/R(Reflective Rewriting Logic) and making the representation changeable. The high descriptive power of pigeon is replaced by the reflection feature of RL/R. Specification of pigeon is divided in to two parts .The first one is the behavior specification part that look like programs. The Second part is the requirement specification part expressed in logical system [12].

5.4 *UCL*

The Universal Communication Language (UCL) [13] can fulfill the role of an ACL and at the same time be convertible to and from a natural language. UCL design is concerned with the description of message structures, their underlining semantic context and the support for protocols for agent interaction. The key point about UCL is that the language can be used not only for communication among software agents but among humans too. This is possible because UCL is derived from the Universal Network Language (UNL), a language created to allow communication among people using different languages. UCL was defined using the Extended Markup Language (XML) to make it easier to integrate into the Internet

V. A SPECIFIC ARCHITECTURE: AUTONOMOUS COUNSELING AGENT

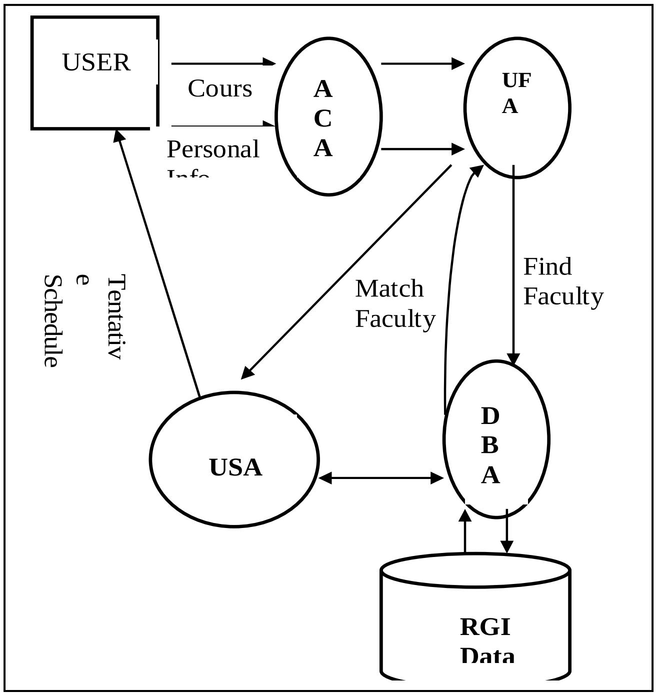

Figure 3.Interaction between User and Agents.

Where, **ACA**=Autonomous Counseling Agent,

UFA=User Finder Agent, **DBA**= Database Agent

USA= User Scheduler Agent.

The Interaction of user with various software agents.

- The user inputs the visitor’s request to ACA.
- The ACA send the request to UFA
- The UFA then gets contact to DBA and finds the matching data from database and then passes matched data to UFA.

As soon as the matching data found the UFA forward this information to USA, then as per the users tentative timing and date and also according to availability of faculty the USA finally send this resulting information to user.

VI. CONCLUSION

Agent technology is very innovative topic for keeping the both academia and industry people quit busy. This paper is proposed and describes the concept of Localization. There are number of agent technology and agent communication languages are available for autonomous software agent, by using the appropriate combination of agent technology and agent communication languages as describe in paper, a Localization will be implement.

The visitors systems describe the autonomous software agents working in the real life environments. The Autonomous Counseling Agent architecture has been presented to describe the autonomous agent from logical view and structural view. Localization is one of the key topic of today’s agent era and it need to be researched into further.

REFERENCES

[1] [1] Kristina Mullamma, Olga Pineiro “Summary of discussion on Is localization just technology”, *Tarragona, Spain.*

[2] [2] Gopal Sakarkar and Dr. V. M. Thakare “Architectural classification and design of autonomous software agent”, ACVIT’09, IEEE Int. Conf. at Aurangabad, India

[3] [3] Gopal Sakarkar, Dr.V.M. Thakare “Revolutionary Architecture for Autonomous Software Agents” International Advance Computing Conference (IACC’09), IEEE March 6-7, 2009, India

[4] [4] Marcos Dias de Assunção, Carlos Becker Westphall, Fernando Luiz Koch“XML in Multi-agent based Applications for Network Management”

[5] [5] Ora Lassila “ Introduction to RDF Metadata”

[6] [6] *Erlin, Yusof Norazah, Abdul Rahman Azizah*“overview on agent application to support collaborative learning interaction” Jan. 2008, Volume 5, No.1 (Serial No.38) US-China Education Review, USA

[7] [7] Gopal Sakarkar, ‚Dr.V.M. Thakare“ A Road Map of Autonomous Software Agent Architectures”, Proceedings of the 3rd National Conference; INDIACom-2009,CSI, Feb.26-27, 2009, New Delhi

[8] [8] Gopal Sakarkar, Dr.V.M. Thakare, “A Survey of Various Software Agent Architectures and Agent Communication Languages”, first International Conf. On MNGSA’08,Dec.2008 CSIR and IACSIT, India

[9] [9]Tim finin , Richard fritzone, Don Mckay and R. McEntire “KQML as an Agent communication Language”, Third Int. Conf. On Info. And Knowledge Management(CIKM’94), Nov. 1994.

[10] [10]. S. Duignan and F. Kilkelly “Software agent communication language using XML”.

[11] [11]. K. Bouzouba, B. Moulian and A. Kabbaj “CG-KQML+: An Agent communication language and its use in a Multi-Agent System”.

[12] [12] Y.Tahara and A. Onsuga “Pigeon:A Specification language for Mobile Agent Application”, AAMAS’04 , July 19-14, New York,USA.

[13] [13]. C. Montesco and D. Moreira “UCL-UNIVERSAL COMMUNICATION LANGUAGE”.

[14] [14] Gopal Sakarkar, Dr.V.M. Thakare“ A Classification of Autonomous Software Agent”, Proceedings of the First National Conference; ETCSIT’09, April 17 &18, 2009, Mysore-16,pp 121-126 India