

# Agent mediated SOA with XML framework for Grid Computing

Gehao Lu

Hao Li

Joan Lu

Shaowen Yao

glu@ynu.edu.cn

Lihao707@ynu.edu.cn

j.lu@hud.ac.uk

yaosw@ynu.edu.cn

School of Software

School of Software

School of Computing & Engineering

School of Software

University of Yunnan

University of Yunnan

University of Huddersfield

University of Yunnan

Kunming, China, 650091

Kunming, China, 650091

Queensgate, Huddersfield, HD1 3DH

Kunming, China, 650091

## Abstract

*Service oriented architecture is applied to Grid computing in the form of OGSA. But the classical service mechanism is too sophisticated to build a large scale service oriented Grid because of varieties of homogeneous and heterogeneous services with different level of quality. On the other hand, agent communities successfully built multi-agent system theory and agent based software engineering in recent years. Autonomic agent becomes a possible solution that hides this complexity. This article reviews the current research trends in Agent based Computing. It also extends SOA into an agent mediated SOA and summarizes methods for describing agent services. It aims to propose a cooperative and competing agent based service mechanism for large scale network and virtual organization. And it also proposes to build a prototype to verify the extended architecture and use existing formal methodology to build an agent service model.*

## 1. Introduction

SOAs have been defined as a set of independently running services loosely bound to each other via event-driven messages [1]. It is an architecture that uses XML based services to facilitate loose coupling among different components of the complex system. The Grid Computing is defined as flexible, secure, coordinated resource sharing, among dynamic collections of individuals, institutions and resources [2]. SOAs have been applied to the actual large scale system, Grid Computing, in the form of OGSA [3].

However, SOA is not a silver bullet that solves all the problems of Grid Computing because the initial Web service is not designed as Grid service. Although the new standard of web service has been proposed, the web service itself is relatively static and limited. One can instantiate a service instance and publish it onto the service directory and let others to request to use it. But in a large scale

network, services are pervasive and ubiquitous. To make use of those millions of possible services without confusion and collision, the relatively simple SOA cannot afford such complexity, it is not reasonable that people know all the details about those millions of services and their implementations. A good idea is to make application intelligent enough to communicate with each other without human intervention. Then this intelligence release human users from knowing too much unrelated details about various services.

That means, we must have some intelligent agent which can substitute the role of the human users, it can visit service directory, understand service semantics, choose from competing services or even orchestrate various services according to the target assigned by the human user. With combination of service and agent, the distributed systems are increasingly viewed as collections of service provider and service consumer components, interlinked by dynamically defined workflows and web services can therefore be realized by agents that send and receive messages [4]. When multiple agents meet in the specific grid, they can cooperate to finish some specific task or they may compete to get some limited services and resources. All those processes are carrying out transparently, that is, the human users don't have to care how the service is found, which service is better, etc.

This article aims to extend service oriented architecture supporting description of the agent based web service. The agent represented service can be described under a XML framework which is confined by agent service ontology. It also applies the theory of multiple agents to Grid Computing and SOA. It aims to propose a cooperative and competing agent based service mechanism for large scale network and virtual organization. Based on the above idea, the paper also proposes a solution for building a prototype mediated by agent services and we are going to use the prototype to verify the above agent service architecture and agent service interact mechanism. We also propose to use Gaia methodology [5], Agent UML [6] and Z language to build formal

model for the agent mediated SOA and Grid Computing.

## 2. Background

### 2.1. Grid Computing and Web Services

Grid Computing is one of the most active research areas in the world of Computer Science. The Grid Computing is defined as flexible, secure, coordinated resource sharing, among dynamic collections of individuals, institutions and resources [2]. The current Grid application tends to build a virtual organization which allows geographically dispersed clients to share distributed resources more effectively. Those resources include computing power, data, hardware, software and applications, networking services, and any other forms of computing facilities. The early study of Grid focused on scheduling and allocation of the specific resources in a relatively low level, and built series of applications like Data Grid or Computational Grid. With the emergence of Service Oriented Architecture (SOA) and XML based Web Services, the service becomes a natural interface that bridges the clients and the resources as in Figure 1. Also, the loosely coupled services allow the application respond to the change quickly by adapting the service implementation while keeping the service description intact. The services also improve the possibility of software component reuse. The merging of SOA (Web Services) and Grid has led to the birth of Open Grid Services Architecture (OGSA). Now we can use WSDL to describe the service and use SOAP to transport the message containing remote procedure calls. Use WS-I protocols to enhance the capability of web services on security, integrity and addressing. The problem with the current SOA is the services are relatively static and simplistic while deploying them onto the complex large scale Internet. And people do not fully make use of the semantics aspect of the XML based services.

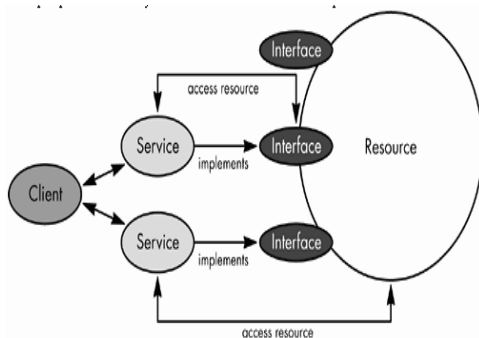


Figure 1. Agent mediated SOA architecture.

### 2.2. Agent based Computing

Agent is an encapsulated computer system, situated in some environment, and capable of flexible autonomous action in that environment in order to meet its design objectives [7]. Some process control systems and software daemons are existing successful example of primitive agent computer systems. The core characteristics about an agent include: it can control its own internal state and behavior (compare to Object in Object-Oriented Analysis); it can sense the environment through sensor and influence the environment through effectors; it is reactive in that it can respond to outside changes in time; it is proactive because it can act on the anticipation of future goals. Each agent has its own objective and tries to realize this objective. When multiple agents work together in some environment, the interactions among agents are inevitable. Those agents can cooperate to finish their design goals.

### 2.3. Levels of Agent research

With the efforts of agent communities, the agent study has evolved into an exciting research subject that may influences the future of computing. In the organizational level, multiple interacting agents form a virtual organization that mimics the human society. Such organizing mechanism could be used to organize the infrastructure of the future extremely complex systems. Yet the structure of the agent based virtual organization still faces key challenges: the trust among agents, the self organization of agents, etc. In the interaction level, traditional techniques like KIF (Knowledge Interchange Format) and KQML (Knowledge Query and Manipulating Language) [16] are enhanced by XML technologies and semantic web technologies. FIPA defined its communication language called ACL which defines 20 basic words to be specification of message. Ontology is also used to restrict terms in specific area. DAML [15] bases its grammar structures on XML and semantic web. In the agent level, theories of traditional artificial intelligence are applied onto the design of individual agent program in order to make the agent capable of reasoning and learning. Several projects aim to build such agent which can behave intelligently in the changing environment. Neural controlled Agent [17] and Evolving Agent [18] are examples that are based on artificial neural network theory and genetic programming respectively. Not only were the studies carrying out in Computer science, disciplines like Sociology, Anthropology, Economics, and Biology all have their own agent studies which may help computer scientist tackle problems from different angles.

## 2.4. Convergence of Agent, SOA and Grid

As discussed in the above sections, either Web services or SOA or Grid Computing has its own limitation. Is it possible to combine all those three major technologies for distributed computing to create a more flexible, adaptive and dynamical computing environment? The answer is YES. Since 2001, researchers have proposed solutions to explore the possibility of this combination. [9] defines an Agent Service as a service that conforms to the set of conventions for Web Services, that is, agent ontologies are defined using XML Schema components, and the agent behavior is described as a WSDL interface. [10] in the area is focused on examining how we may use multi-agent systems to develop a market place for the trading of computational grid resources such that grid scheduling is completed without the limitations of the top down approach. [11] proposes a multi-agent approach that naturally provides a solution to the service selection problem. The agents can support considerations of semantics and quality of service (QoS). In the above solutions, we took agents as the carriers of the services and the implementers of the services. Or say, we delegate services to agents to realize functionalities and make agents as service providers or service consumers for the human clients. Agent substitutes the static Object or component in the Web services paradigm to become the basic unit of the service oriented architecture.

Some prototypes has been build for experimenting the possibility of agent based SOA. Researchers deploy current generations of Agent and semantic web technologies to create a global test bed for dynamic service composition involving lots of organizations [12]. The test bed created a large-scale networked agent based environment which is used to explore the possibility to make agent technology integrate with Web services and Grids. The CONOISE-G project [13] is another project that provides mechanism to assure effective operation of agent-based virtual organization. It built a system architecture using technologies like agent decision making, auctions for allocation of contracts, and service discovery incorporating with quality of service control. The prototype system Distributed Agents for Mobile and Dynamic Services (DIAMOnDS), allows a service to send agents on its behalf, to other services, to perform data manipulation and processing. Agents have been implemented as mobile services that are discovered using the Jini Lookup mechanism and used by other services for task management and communication [19].

Combination between agent and Grid can happen in the lower resource level or in the higher service level. In Grid Computing, agent technologies can be used as one of the mechanisms for resource allocation and job scheduling so that the applications dont have to know every detail of the resource level of Grids. Researchers built multi-agent sys-

tem as Grid scheduler at this level to form a market place for trading Grid resources instead of trading Grid service in the service level [14].

## 3. Discussion

The purpose of agent based SOA or Grid is to build a service oriented architecture that supports Grid computing which is consistent with Web services and informed by Agents research. Based on the researches mentioned above, an agent-mediated semantic-aware SOA infrastructure should be built so that multiple agents are allowed to cooperatively achieve multiple services consuming, registering and provision.

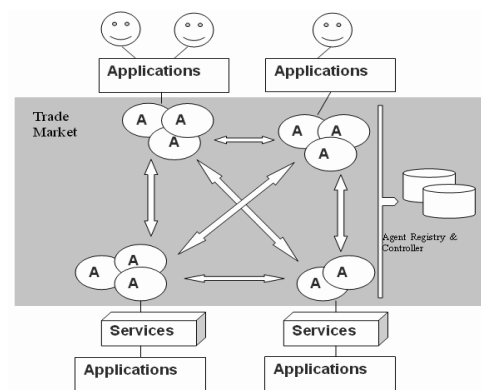


Figure 2. Agent mediated SOA architecture.

It is necessary to extend service oriented architecture to support description of the agent based web service. Find out an approach to integrate the Agent with the existing Web services standard so that the agent can be encapsulated as Web services and we do not have to build another mechanism to make use of those multiple agents. There might be two major issues to consider here: the transport layer which is responsible for communicating among multiple agents and the description layer that is used to describe, discover, publish and advertise agent services. Agent should build upon the existing Web service standards like SOAP, UDDI, WS-I, etc. If possible, semantics could be applied onto the agents as well. For example, we may build ontology to specify the concepts of an agent and use schema to formalize the construction of agent services.

A cooperative and competing mechanism for multiple agent services should be proposed. An agent represents service plus service implementation. There must be hundreds of services suppliers and consumers in a specific network or organization. Especially in large scale network such as Internet, the selection and trust of multiple services (the same services with different QoS) become a cumbersome of the

enlarging virtual organization. A good suggestion is to build every service as an agent, while multiple agents represent multiple services and the multiple benefits, desires or intentions behind those services. A service trade market is also choice that different consumer agent can search most appropriate supplier agent who can supplies the best service according to the consumers preference or predefined QoS.

Researchers need to program a prototype that is composed of agent services and use it to verify the agent service architecture and agent service mechanism. Also, to support the prototype design, a developing environment, a set of toolkits, and a series of middleware are nontrivial. Only are these supporting facilities available, the proposed agent solution can be said successful. The middleware include the agent registry middleware which is responsible for registering the agents and their relate information, the agent controller middleware which is responsible for controlling the decision process of the intelligent agents, and so on. The toolkits include the Computer Aided Software Environment that supports agent based software engineering, the graphics generating tool which allows you to design agent based software without repeating additional jobs from scratch. It is also possible to use Gaia methodology [5], Agent UML [6] and Z language to build formal model for the agent mediated SOA and Grid Computing.

#### 4. Conclusion and Future work

The merging of several cutting edge technologies always brings a much more powerful new computing model. The advanced applications evolve toward a more complex system in a more complex and changing environment. The mechanisms for controlling such a big system like Grid are difficult to design and implement due to the huge size and excessive complexity. The realization of the dynamic Grid Computing depends on the realization of a real Service oriented architecture powered by autonomous Agents. The adoption of Agent technologies in the development makes the design of the complex system become possible. The modeling and design of Grid systems using autonomous multi-agent have demonstrated a promising approach for designing and developing intelligent, open distributed systems in grid service. Agents enhance the level of abstraction and facilitate the loose coupling of the service oriented architecture.

The future wok is to investigate a better approach to build intelligent autonomous agent in the agent level. And a standard framework should be built upon the existing service oriented architecture to adopt the agent based services. This framework can fully make use of Web services and XML technologies. The software development methodology needs to be transformed from the traditional Object-oriented paradigm to the agent-oriented paradigm and ap-

propriate toolkits that support the new paradigm are also required. Before widely adoption of agent based computing in SOA and Grid, lots of experimental work should be done to test the availability and stability of the agent service systems. The test beds or prototypes need to explore different combinations of technologies and different frameworks. It is reasonable to believe that the use of multiple agents will expand rapidly in the future Grid and SOA based systems once the above key future work have been done.

#### References

- [1] Business Integration, <http://www.bijonline.com/default.asp> [12.01.2004]
- [2] Ian Foster, Carl Kesselman, and Steve Tuecke. The anatomy of the grid. enabling scalable virtual organizations. International Journal of Supercomputer Applications, 2001.
- [3] Joshy Joseph, Craig Fellenstein. Grid Computing. Prentice Hall PTR. December 30, 2003
- [4] Agent technology roadmap, Compiled, written and edited by Michael Luck, Peter McBurney, Onn Shehory, Steve Willmott and the AgentLink Community
- [5] Jones, C.B. (1990) Systematic Software Development using VDM, 2nd edition. Prentice-Hall, Englewood Cliffs, NJ.
- [6] Odell, J., Parunak, H. V. D. and Bauer, B. (2001) Representing agent interaction protocols in UML. In Agent-Oriented Software Engineering '00 Proceedings of the First International Workshop AOSE-2000 (eds P. Ciancarini and M. Wooldridge), LNCS Volume 1957, pp. 185-194. Springer, Berlin
- [7] Wooldridge, M. and Jennings, N. R. (1995) Intelligent agents: theory and practice. The Knowledge Engineering Review, 10(2), 115-152
- [8] NICHOLAS R. JENNINGS, A Roadmap of Agent Research and Development, Autonomous Agents and Multi-Agent Systems, 1, 7:C38 (1998)
- [9] Arturo AvilaRosas, Luc Moreau, Vijay Dialani, Simon Miles, Xiaojian Liu. Agents for the Grid: A comparison with Web Services (part II: Service Discovery). Technical report, Intelligence, Agents, Multimedia, Department of Electronics and Computer Science, University of Southampton, United Kingdom
- [10] Peter Gradwell, Grid Scheduling with Agents, Data Knowledge Engineering, 41(2-3):229:C245, 2002.
- [11] E. Michael Maximilien, Munindar P. Singh, Toward Autonomous Web Services Trust and Selection, ICSOC04, November 15:C19, 2004, New York, New York, USA.
- [12] Steven Willmott, David Bonefoy, Ion Constantinescu, Simon Thompson, Patricia Charlton, Jonathan Dale, Tianning Zhang. Agent Based Dynamic Service Synthesis in Large-Scale Open Environments: Experiences from the Agentcities Testbed. AAMAS04. July 19-23, 2004, New York, USA.
- [13] Conoise-G Project <http://www.cs.wisc.edu/conoise/>
- [14] Junwei Cao, Daniel P. Spooner, Graham R. Nudd, et al. Agent-based resource management for grid computing, 2002. Available at <http://www.dcs.warwick.ac.uk/?hpsg/html/downloads/public/docs/CaoJ>.

ARMGC.pdf

- [15] DAML (2001) The DARPA agent markup language . See <http://www.daml.org>
- [16] Mayfield, J., Labrou, Y. and Finin, T. (1996) Evaluating KQML as an agent communication language. In *Intelligent Agents, II*, LNAI Volume 1037, pp. 347-360. Springer, Berlin
- [17] R.D. Beer and J.C. Gallagher. Evolving dynamic neural networks for adaptive behavior. *Adaptive Behavior*, 1(1): 91-122, 1992.
- [18] J. R. Koza. *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge, MA, 1992.
- [19] M Aamir Shafi, Maria Riaz, Saad Kiani, Anjum Shehzad, Umer Farooq, Arshad Ali. Iosif C Legrand, Harvey B Newman. *Computing in High Energy and Nuclear Physics*, 24-28 March 2003, La Jolla, California