

# Ontology & Agent-Based Information Processing in Network Centric Warfare

Hui Zhang

Computer Engineering Department  
Naval University of Engineering  
Wuhan, China  
ebookisok@yahoo.com.cn

Kerong Ben

Computer Engineering Department  
Naval University of Engineering  
Wuhan, China  
benkerong08@21cn.com

**Abstract**—This paper analyzes the feasibility of agent application in the Network Centric Warfare (NCW), and points out the difficulties in this process as well. The Semantic Web-based information processing framework is constructed, so that we can process information based on agent, effectively translate information superiority into knowledge superiority and finally build the decision superiority. At the same time, this paper also propose an ontology-based Web Service description model to describe the Web Services in multi-directions, on the basis of which agent can accurately and correctly comprehend the function and process of Web Services. Thus, a general platform-independent technological framework is built to make agent automatically invoke and compose the Web Services, and the interconnection, intercommunication and interoperation of information on various platforms also can be realized finally.

**Keywords**—agent; ontology; Web Service; Network Centric Warfare.

## I. NETWORK CENTRIC WARFARE

Network Centric Warfare (NCW) is aimed at integrating the separated combating factors into the combat command system, combat power system and combat support system based on the network. Therefore, the share of situation awareness from different combat factors can be realized so as to maximally translate the information superiority into the knowledge and decision superiority [1]. NCW theories divided the battlefield into three domains, including physical domain, information domain and cognitive domain. In NCW, the information domain is responsible for creating, processing and sharing the information. In the business community, processing and sharing the information is the source of potential value, while in the battlefield, it's the source of potential combat strength [2]. From this perspective, processing and sharing the information is not only the foundation but also the core of the information domain. In this point, the features of agent make itself fully meet the need of information domain.

## II. ANALYZING THE FEASIBILITY OF AGENT APPLICATION IN NETWORK CENTRIC WARFARE

In this part, we will analyze the feasibility of agent application in Network Centric Warfare from the technological and empirical aspect.

- Agent is an intelligent entity with the ability of inference in some degree. It can search, dig, catalogue and integrate the mass information, transfer the information to the decision-maker on demand, and assist the decision-maker command and control, which leads to the great improvement of the decision-maker's ability of planning and delivering the combat strategy in response to the constantly changed battlefield situation. Furthermore, due to agent great ability of processing the information and adapting to the environment, the command post of the united army can be arranged more separately, and their survivability will be greatly improved. At the same time, their subordinate command post can become smaller ones with more flexibility and mobility.
- Since the multi-agent consisting of various agents has the ability of cognizing the whole environment, we can take use of agent interoperability to link the isolated information islands so as to realize information sharing in the whole army. Thus, we can beyond the limitation of original "chimney" system, increase the information sharing among all units of the united army and between the united army and allied army, and build the foundation for the share of situation awareness.
- Mobile ability of agent can be used to upgrade the network share of situation awareness. This kind of situation awareness is very crucial for decision-maker to respond to the situation and prepare for the combat when the enemies launch the information war. It's also useful to the network central operation and helps to improve the efficiency of NCW.

Certainly, we also can easily find there are many projects aimed at application of agents in the NCW projects of US DoD, such as CoABS/CoAX [3], DAML [4], UltraLog [5] and FCS [6].

- CoABS/CoAX: A project to develop AAMAS (Autonomous Agents and Multi-Agent Systems) technology into an integrated environment to provide adjustable interoperability between disparate information systems and to support distributed mixed-initiative decision-making with acceptable agent

---

Research project was supported by the Open Foundation of SKLSE under Grant No. 05-04.

behavior. It used the CoABS Grid which is a well developed infrastructure that has been evaluated and tested in several operational military settings.

- DAML: The DARPA Agent Markup Language. DAML is a project to develop data, information and knowledge formats for exchange between systems. The DAML+OIL [7,8] language has been extended and refined into the OWL [9,10] ontology language adopted by the W3C and used in projects like the “Advanced Knowledge Technologies (AKT)”.
- UltraLog: Utilizes agent technology to provide highly survivable information and logistics systems. UltraLog was independently verified as being able to offer a high degree of robustness while solving problems of a realistic size and nature. The Cougaar (a robust configurable multi-agent platform) agent tool kit was a result of the UltraLog program and is now open source and widely used.
- FCS Command and Control Study: FCS, the Future Combat System, analyzes the current state of agent-based technology applied to C2 functions for the Future Combat System and net-centric warfare in general. The study concludes that advancements are required in the areas of scalability, mobility and security of agent systems before the command and control problem can be suitably addressed.

According to the analysis above, we can conclude that the agent is able to meet the requirements of information processing in NCW, but there are still some technological difficulties.

- The current Web mainly plays the role of collecting information, but not providing support in processing information, i.e. it’s not used as the computational device until now.
- The current Web Service description models only concentrate on the description of the service function or the description of invoking process. Lacking the mechanism of describing Web Services in an all-around manner makes it difficult for agent to completely and automatically process Web information.
- The information searching mechanism of Web is rather simple, so its recall ratio and precision ratio are also relatively lower and people themselves are often needed to choose the useful information in lots of results. These disadvantages also make it hard for agent to search and choose the most proper information resource on the Web according to the decision-maker’s requirement.
- The present researches mostly focus on how to realize the information superiority and test its role in enhancing combat effectiveness, but rarely discuss how to build the fundamental infrastructure of knowledge and how to develop knowledge superiority so as to upgrade the ability of decision-maker.

Based on the analysis above, this paper proposes the Semantic Web-based information processing framework. In this way, we can process information on the semantic level, effectively translate information superiority into knowledge superiority and finally build the decision superiority. Its major idea is to organize and abstract Web information into the Web Services so that the business process can be composed through using various Web Services. Meanwhile, in order to make agent automatically process Web information, we also develop Web Service description model with standard semantics, which can be easily comprehended by the agent. Consequently, agent can invoke, coordinate and compose the Web Services with clear semantics, and then search and choose the most proper information resource on the Web according to the decision-maker’s requirement.

### III. SEMANTIC WEB-BASED INFORMATION PROCESSING FRAMEWORK

Agent needs the support of infrastructure and semantics to run automatically in the network environment and helps decision-maker to finish the complicate tasks. The Semantic Web proposed by Tim Berners-Lee [11] provides the technological support for agent in comprehending the information [12]. We propose the Semantic Web-Based Information Processing Framework (SWBIPF) on the basis of Semantic Web structure. Figure 1 demonstrates the corresponding relation between the SWBIPF and Semantic Web.

The SWBIPF consists of four levels: Network Infrastructure, Knowledge/Semantic Web Services, Agent-based Computing and Human-Computer Interaction.

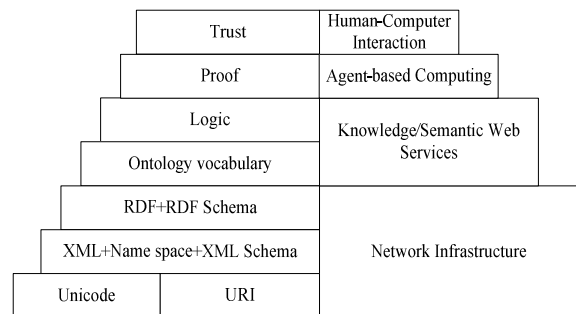


Figure 1. Semantic Web-Based Information Processing Framework

#### A. Network Infrastructure

Network Infrastructure: Including URI, XML Schema, RDF Schema and etc., network infrastructure is used to display the content and structure of the data and describe Web information and their types so as to provide syntax basis for data interoperation. However, it does not include any special semantics.

#### B. Knowledge/Semantic Web Services

Knowledge/Semantic Web Services: Based on the ontology and logic, this level organizes and abstracts Web information into the knowledge and Semantic Web Services [13].

Ontology is used to define the terminology that can be used by other elements of SWBIPF specifications. Therefore, it enables the reuse of terminology as well as interoperability between agents referring to the same or linked terminology. The ontology also defines formal semantics for Web information, so that agents can automatically process the Web information.

Web Service, as the platform for developing interoperable and distributed application program, can define its own semantics. Thus, it needs a mechanism to declare the defined conceptions when interoperating with other Web Services, otherwise, the implication of the meta-information can not be explained. Through using ontology to abstract Web information conceptually and structurally, the Web information processing can be elevated from the level of information processing to the level of knowledge processing.

Therefore, we use ontology to formally and specifically explain the essential elements of Web Services to develop the Semantic Web Services, which can promote the agent to comprehend the properties, terminology and relations of Web Services, and then better support Web Services automatic searching, composing and interacting.

As to the logic, it is responsible for logic inference on the basis of the knowledge described by ontology, i.e., the logic can derive the implicit knowledge from the assertion of all levels of Semantic Web.

### C. Agent-based Computing

Agent-based Computing: For the common purpose or their own targets, agents need to share the knowledge of related tasks and their solutions, and then coordinate to solve the problems.

Agent computing can be regarded as a cognitive system, which composed of belief, intension, capability and commitment. Among these four elements, the belief of agent, as the foundation of agent automatic computing, describes comprehension and cognition of agent about the outside world. The intention of agent, as the inducement of agent automatic computing, embodies the choice of agent about some future behavior. The capability of agent is both the prerequisite of automatic computing and the crucial factor for agent to fulfill its intention. The commitment of agent, embodying the computing, is decision of agent about how to respond to. Agent will make commitment on the basis of its own belief, intention and capability. Once one kind of commitment is made, agent will undoubtedly fulfill it.

Knowledge/Semantic Web Services level organize and abstract Web information into the knowledge, which can be developed into the knowledge base after being formally and standardly processed. And then, on the basis of common comprehension upon the domain knowledge, agent can find the solution of the target problems through invoking, coordinating and composing different Web Services.

### D. Human-Computer Interaction

Human-Computer Interaction: Referring to logic description about facts, it can provide the related complex

proving. Based on all previous levels, there are numerous expressions about one fact. The reasonability of those expressions depends on the context. When visiting these expressions, agent can automatically decide whether trusts those expressions wholly or partly according to the context and the decision-maker's requirement, and then establish a reliable network to provide credible service, which is very important for military system in particular.

Under the SWBIPF, we can process information on the Semantic Web and make it possible to realize the interactions between agents and Web Services. However, in order to make agent automatically process Web information, we still need to develop a Web Service description model with standard semantics, which can be easily comprehended by the agent.

## IV. ONTOLOGY-BASED WEB SERVICE DESCRIPTION MODEL

Regarding the Web Services as the central part, the major idea of SWBIPF is to use the standard ontology language to describe Web Services and deliver services through announcement mechanism. Thus, it's a crucial prerequisite to realize intelligent information processing that how to make agent search the usable Web Services and automatically execute them.

Currently, OWL-S [14] is a general method to add semantic information into Web Service. However, the disadvantage of OWL-S is it just partly and incompletely describes the behavior restriction and non-functional properties of Web Services. Moreover, it does not describe QoS, which causes difficulties in choosing services. As to other Web Service description models, they only partly focus on the function of service (such as WSDL [15,16]) or service process (such as BPEL4WS [17,18]). For lacking of a special model to describe the service function and process in an all around manner, agent can not automatically search, compose and invoke Web Services.

Therefore, this paper develops an ontology-based Web Service description model to describe the Web Services in multi-directions, on the basis of which agent can accurately and correctly comprehend the function and process of Web Services.

**Definition 1** (Web Services description model) A Web Service description model  $WS$  is a triple:  $WS = \langle AD, Process, Logic \rangle$ , among which,  $AD$  means service advertisement,  $Process$  means service process,  $Logic$  means service logic. Figure 2 clearly demonstrates this triple.

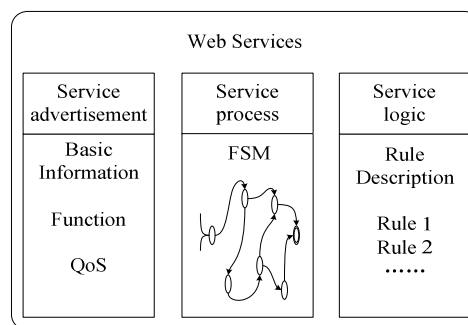


Figure 2. Web Services description model

### A. Service advertisement

Service advertisement includes the basic information of service, its functional information and QoS, which shown in Figure 3. The basic information consists of ID of Web Service, its name, URL, and the category information in related taxonomy. Functional information, mainly describing the function properties of Web Services, includes the specification of function, the data model and algorithms used by Web Services and etc. QoS, responsible for describing the quality of Web Service, contains availability, response time, reliability, security and etc. Service advertisement is created in domain ontology and described in ontology language. Based on these descriptions, the service requester (decision-maker or agent) can search the Web Services they need through syntax matching, semantic matching, multi-matching and so on to improve the recall ratio and precision ratio.

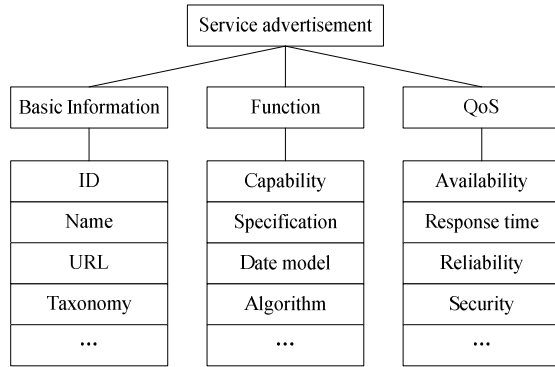


Figure 3. Service advertisement

For instance, the ordinary decision-maker only needs to input texts of describing the functions of the Web Services he required, and then the target Web Services can be searched via syntax matching. Syntax matching firstly takes use of API provided by Apache Lucene to pick off the stop-words in the decision-maker's requirement text, then, processes the key words left in standardized way, and finally computes the similarity between these key words and service advertisement description to find the proper service. Certainly, if the ontology effectively describes the level-structure and semantics of the conceptions in service advertisement, service requester (decision-maker or agent) can more accurately search the target Web Services they need through semantic matching. In the process of semantic matching, geometric-distance algorithm, property algorithm, information capacity algorithm and other algorithms are utilized to compute the semantic distance between conceptions. After finishing the syntax matching, Multi-matching launches the second semantics-based search and then filter the searching result further referring to the QoS.

However, only establishing service advertisement ontology is not enough to make agent represent the decision-maker to execute the task. It needs to know how the service runs, which conditions must be satisfied to invoke related service, and what the result is after the service runs. These problems will be solved through introducing the service process and service logic.

### B. Service process

Service process describes the Web Services behavior, so that agent can function as workflow engine to invoke, coordinate and compose Web Services. Service behavior features as the interaction between Web Services, including the steps of execution and the conjunction or disjunction of service. Usually, Common Gateway Interface (CGI) is used to develop Web information system and the system will invoke different functions of Web Services via the code of program, which means the service process on the Web is only a description in the form of code. Agent can not understand and comprehend these descriptions in the form of code. But, the model of Finite State Machine (FSM) contains finite-state sets, transition function sets and output function. Therefore, in any time, FSM always stays in one state. When receiving the input commands, FSM responds in certain behavior (invoking the corresponding transition function and output function), produces the output and induces the state transition.

Although both FSM and Petri-net can be used to describe the state transition of Web Services, and even the expression ability of Petri-net is better than FSM, Petri-net has higher space complexity in computing than FSM. However, if Petri-net has a bound, its space complexity can be decreased. But we must notice, as to a bounded Petri-net, its reachability graph can be completely described by the complicated FSM [19], so we can think the bounded Petri-net are not out of the expression ability of FSM. Therefore, we can use FSM to describe the process of Web Services, i.e. each interaction between Web Services can be looked as the state transition in FSM.

**Definition 2** (Service process model) Given a Web Service, we can use FSM to describe the service process as a sextuple:  $FSM = \langle Q, I, O, \delta, q_0, F \rangle$ . Among these six elements,  $Q$  is a finite-state set,  $I$  is a finite set of input commands,  $O$  is a finite set of output messages,  $I \times O$  is the alphabet of FSM,  $\delta: Q \times I \times O \rightarrow Q$  is a transition function, i.e. given one state, service can transit to another state in response to the input command and output message,  $q_0 \in Q$  is the initial state and  $F \subseteq Q$  is final states set.

Since FSM uses the sharing alphabet and operation set (to produce the ontology) to describe the service behavior, agent can determine whether the information is sufficient to invoke service and be aware what the result is after service running, through analyzing the service process described by FSM. If the service can not be invoked successfully, agent will inform the decision-maker what other necessary information must be added. In this process, agent will more automatically invoke service and thus reduce decision-maker's intervention to the great extent.

The interaction process between agents and Web Services includes four steps (shown in figure 4):

- Step 1: Receives the decision-maker's order via interface agent;
- Step 2: Scheduling agent produces the service sequence and the method of service composition, and

the facilitator agent is responsible for the network load balancing;

- Step 3: Task agent invokes Web Services through sending the input command, which inspires the internal computation;
- Step 4: Task agent gains the output message from the Web Services, and then sends the result back to the decision-maker via interface agent.

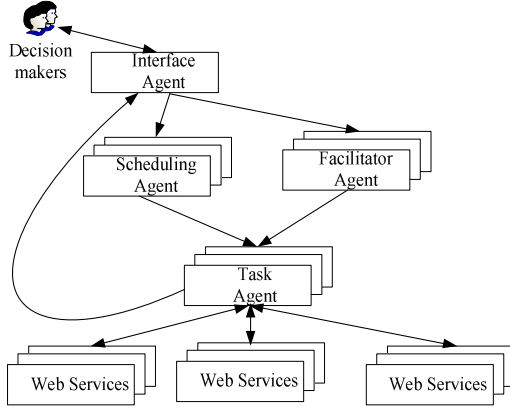


Figure 4. Interaction process between agents and Web Services

### C. Service logic

On the basis of the general service ontology, Service logic takes the expression of rules into account, which are the prerequisites for agent to make inference, so that the infrastructure of knowledge contains more knowledge that can be comprehended by agent.

We think that the expression of rule will play important role in NCW. On one side, the rules can inject more semantics, ranging from the common sense in human's mind to military strategy. On the other side, the expression ability of the rules can save the storage space to a large degree, because some facts can be inferred dynamically from the existing facts and rules in the knowledge base and needn't be stored.

Figure 5 simply shows a fragment of military rules, which is described directly via conceptions and relations in the domain ontology. The *antecedent(body)* represents the prerequisite of inference, and the *consequent(head)* represents the result of inference. This fragment means when  $x$  is aimed at  $y$ ,  $x$  will hold fire at  $y$  if  $x$  and  $y$  belong to the same side and  $x$  will fire at  $y$  if  $x$  and  $y$  belong to the opposite sides respectively. Based on this fragment, firing each other between the warships in the same side and the related accidental injuries can be prevented.

```

Redside(x) ∧ Redside(y) ∧ Aim(x,y) → Holdfire(x,y);
Blueside(x) ∧ Blueside(y) ∧ Aim(x,y) → Holdfire(x,y);
Redside(x) ∧ Blueside(y) ∧ Aim(x,y) → Fire(x,y);
Blueside(x) ∧ Redside(y) ∧ Aim(x,y) → Fire(x,y).

```

```

Redside(x):  warship in red side
Blueside(x): warship in blue side
Aim(x,y):   x aimed at y
Holdfire(x,y): x holdfire at y
Fire(x,y):  x fire at y

```

Figure 5. A fragment of military rules

## V. CONCLUSION

Based on the particular properties of NCW and the requirement of information processing, this paper proposes the SWBIPF, the superiority of which is it takes the semantic information of Web Services into account. Thus, in the open network environment, the automatic search and composition of Web Services can be realized via the semantic information based on the Web Service. The SWBIPF also provides the technology platform to solve the agent-based computing problems in a distributed environment. However, the existing researches upon Web Services mainly regard Web Service as a non-state action, which actually is incomplete. Therefore, the Web Service description model developed in this paper not only considers the service advertisements and rules, but also describes the Web Service as a finite state action sequence. Furthermore, this model also uses FSM to describe service process so as to make agent automatically invoke and compose services and help decision-maker to execute the task.

This paper only develops a technology framework and a description model, and there are still many problems need to be further researched. For instance, when describing service process, the FSM can be further extended [20] according to different requirements. On this basis, the verification of service composition and other works also can be carried out. Moreover, agent, as a work flow engine, needs a proper agent computing model corresponding to the description model proposed in this paper.

## ACKNOWLEDGMENT

This research project was supported by the Open Foundation of SKLSE (State Key Laboratory of Software Engineering) under Grant No. 05-04. Great thanks to Professor Kerong Ben and all my classmates for the sincere helps they provide in the work of this project.

## REFERENCES

- [1] Jun Yu, Kerong Ben, "Network Centric Warfare and Complexity Theory", Beijing, Electric Industrial Press, 2005.
- [2] Patrick Beautement, David Allsopp, Mark Greaves, et al, "Autonomous Agents and Multi-Agent Systems (AAMAS) for the military-issues and challenges", S.G. Thompson and R. Ghanee-Hercock (Eds.): DAMAS 2005, LNAI 3890, 2006, pp. 1-13.
- [3] D.N. Allsopp, P. Beautement, J.M. Bradshaw, E.H. Durfee, M. Kirton, C.A. Knoblock, N. Suri, A. Tate, C.Q. Thompson, "Coalition Agents Experiment: Multiagent Cooperation in International Coalitions", IEEE Intelligent Systems, 17(3), 2002, pp. 26-35.

- [4] Ian Horrocks and Peter F. Patel-Schneider, "The generation of DAML+OIL", Proc. Description Logic Workshop, 2001, pp. 30-35. Available at: <http://ceur-ws.org/Vol-49/>.
- [5] Mike Dyson, Sue Rho and David Wells, "Extending the limits of DMAS survivability: The UltraLog project", IEEE Intelligent Systems, 19(5), 2004, pp. 53-61.
- [6] T. E. Potok, L. Phillips, R. Pollock, A. Loebl and F. T. Sheldon, "Suitability of agent technology for command and control in fault-tolerant, safety-critical responsive decision networks", Proc. 16th Int'l Conf. Parallel and Distributed Computing Systems, Reno NV, Aug. 2003, pp. 13-15.
- [7] Ian Horrocks. "DAML+OIL: a description logic for the semantic web", Bulletin of The IEEE Computer Society Technical Committee on Data Engineering, 25(1), 2002, pp. 4-9.
- [8] Deborah L. McGuinness, Richard Fikes, James Hendler, Lynn Andrea Stein, "DAML+OIL: An ontology language for the Semantic Web", IEEE Intelligent Systems, 17(5), 2002, pp. 72-80.
- [9] W3C, "OWL Web Ontology Language Reference", 2003. Available at: <http://www.w3.org/TR/owl-ref/>.
- [10] Peter F. Patel-Schneider, Patrick Hayes and Ian Horrocks, "OWL Web ontology language semantics and abstract syntax", W3C Working Draft, 2003. Available at: <http://www.w3.org/TR/2003/WD-owl-semantics-20030331/>.
- [11] Tim Berners-Lee, James Hendler, Ora Lassila, "The Semantic Web", Scientific American, 12(5), 2001, pp. 35-43.
- [12] James Hendler, "Agents and the Semantic Web", IEEE Intelligent Systems, March/April 2001, pp. 30-37.
- [13] Payne T, Lassila O, "Semantic web services", IEEE Intelligent System, 19(4), 2004, pp. 14-15.
- [14] D. Martin, et al, "OWL-S: Semantic markup for Web Services", 2004. Available at: <http://www.w3.org/Submission/2004/SUBM-OWLS-20041122/>.
- [15] W3C, "Web Service description language (WSDL) 1.1, W3C Note", 2001. Available at: <http://www.w3.org/tr/wsdl>.
- [16] Chinnici R , Moreau J J , Ryman A , et al, "Web services description language (WSDL) Version 2.0 Part 1: core language", 2005. Available at: <http://www.w3.org/TR/WSDL20/>.
- [17] Petia Wohed, Aalst Wil M P van der, Malon Dumas, et al, "Analysis of web services composition languages: The case of BPEL4WS", Proc. 22th Int'l Conf. Conceptual Modeling, Chicago, USA, 2003, pp.200-215.
- [18] BEA Systems, IBM, MicroSoft, et al, "Business process execution language for Web Services version 1.1", 2005. Available at: <http://www-106.ibm.com/developerworks/WebServices/library/wsbpel/>.
- [19] Wombacher A, Fankhauser P, Mahleko B, Neuhold E, "Matchmaking for business processes based on choreographies", Int'l Journal of Web Services Research, 1 (4), 2004, pp. 14-32.
- [20] Wombacher A , Fankhauser P , Neuhold E, "Transforming BPEL into annotated deterministic Finite State Automata for service discovery", Proc. IEEE International Conference on Web Services (ICWS2004), Los Alamitos , IEEE Computer Society Press, California, 2004, pp. 316-323.