구조적 시멘틱 웹 객체 기반 네트워크 데이터 모델에 관한 연구

김진홍*, 김현호**

요약

일반적으로 개념적 객체 모델은 객체 시스템에 해당하는 모델의 복잡성에 대해 다양한 관리를 제공한다. 이러한 이유로 웹 객체라 불리우는 객체의 개념은 전적으로 계산할 수 있는 능력과 네트워크 인터페이스를 제공한다. 웹 어플리케이션들은 경우에 따라 ORM, UML, 시멘틱 객체 모델과 같은 툴처럼 객체 모델에 의해 설계되는 객체 지향 웹으로 생각할 수 있다. 객체들은 모듈성, 자동성, 자원 레벨 도메인 이름과 객체 모델을 가진 강력한 웹 계산의 새로운 방법을 제공할 뿐만 아니라 로컬 또는 귀납적 추론을 만드는 플랫폼이 될 수 있다. 이런 이유로 구조적 시멘틱 웹 객체 모델은 어플리케이션 환경에 대한 더 큰 의미를 가지며 데이터 구조 능력을 더 많이 제공한다. 이 논문에서의 연구는 다음 2가지를 추구한다. 첫째는 구조적 웹 객체 모델과 개념적 데이터 모델의 이해와 문서화를 도와준다. 둘째는 구조적 시멘틱 웹 객체를 위한 엔진 커뮤니케이션 모델을 제공한다. 이에, 본 논문에서는 일반적 웹 객체의 통신 프로토콜에 나타날 수 있는 문제점을 해결하기 위해 구조적 시멘틱 웹 객체 모델에 기반한 XML-RPC 프로토콜 환경에 따른 2차원 스핀 계산을 통해 시멘틱 웹 객체 응답 속도를 빠르게 제공해 준다.

A Study on Network Data Model based on Architectural Semantic Web Object

Jin-Hong Kim*, Hyun-Ho Kim**

ABSTRACT

In general, conceptual object models provide various managements to the complexity of the model corresponding to the object system. For this reason, the concept of object, so called web object, delivers full computational power and network interface. It has thought of object-oriented web in which web applications are designed by object model, such as tools, ORM, UML, Semantic Object Model in some cases. Objects can be a platform to make logical or inductive inferences as well as to give a new way of web computation that is modular, autonomous, and robust with resource level domain names and the object models. As the case stands, Architectural Semantic Web Object Model can capture more meanings of an application environment and provide richer data structuring capabilities. The research area presented in this paper aims the followings. First of all, it assists understanding and documentation of architectural web object model and conceptual data model. Secondly, it presents the description of Engine Communication model for Architectural Semantic Web Object. Accordingly, in this paper, in order to solve the problems that can occur within communication protocol of general web, we provide faster response time with 2-Dimensional structure of spin calculation, which has architectural semantic web model based on XML-RPC protocol environment.

Key Words : Object Model, SWOM, XML Schema, Semantic

* 한성대학교 정보통신공학과(☒jinhkm@gmail.com)
** 명지전문대학 컴퓨터정보과
· 제1저자(First Author) : 김진홍 · 교신저자(Correspondent Author) : 김현호
· 접수일(2012년 7월 25일), 수정일(1차 : 2012년 8월 21일), 게재확정일(2012년 8월 23일)

Ⅰ. Introduction

A various conceptual object model has been already used, such as tools, ORM, UML, and so on. There is also Semantic Object Model that is designed to capture more meaning of an application environment, and provides richer data-structuring capabilities. In order to present various representations, nowadays, XML has become a popular data transfer language among database applications with different platform. XML Schema using a conceptual schema approach states that a XML Schema can be hierarchical or flat. XML Schema serves as both validation support and design documentation for a set of XML instance documents[1,2]. In particular, XML Schema supplies the following additional features: XML syntax, rich data type, supporting for name spaces, constraints, and type derivations.

However, the transformation of Architectural WOM into Flat XML Schema is still not addressed[3]. For example, WOM have retrieval and referral to calculate the similarity among objects is based generally. In addition, we have problems as followings; (1) A similar document is written in a synonym term (2) it is considered similarly that the term means the same as any other written document. For this reason, our research paper aims to: (1) assist understanding and documentation Architectural WOM conceptual data model and (2) propose a description for Semantic based Architectural Web Object Model.

This paper is structured as follows. Section Ⅱ shows the Web Object Model. Section Ⅲ explains the Architectural SWOM. Section Ⅳ discusses the XML-RPC Protocol for SWOM. Section Ⅴ shows the experiment and evaluation. Finally, Section Ⅵ concludes the our paper.

II. Web Object Model(WOM)

Web Object is intended to be a layer that can support more general classes of computations including semantic web and more[4]. Model, such as numerous tools and application, can be built upon the layer of firmly structured and well- powered utilities of interactions; it is based on concepts that are developed and published by semantic object model.

2.1 Web Object Identifiers

A web object identifier has one or more attributes that are employed to identify object instances[5,6]. It is symbolized by underline ID. So to speak again, web object is an information unit; for examples, identifier is Name[7,8]. A group identifier is an identifier that has more than one attribute.

2.2 Types of Web Object

Simple Object is an object that has a single-value of simple or group attributes but not object attributes[9,10,11]. Composite Object is an object that contains one or more multi-value of simple or group attributes but not object attributes. Archetype/Version object produces other semantic objects that represent versions, release, or editions of the archetype. Parent/Subtype object has an important characteristic called inheritance. A sub type acquires or inherits all of the attributes of its parents. Compound Object contains at least one object attribute. Association object is an object that relates at least two objects and stores data that are peculiar to that relationship. Hybrid object is a combinations of composite and compound objects. In particular, a hybrid object is an object with at least one multi-value group attribute that includes an object attribute.

III. Architectural SWOM

Web Object Model, as shown in Figure 1, manages the complexity of the model, corresponding to the object system: corporation and its environment, comprising the modeling scope. It is divided into different layers. Each layer consists of a partial model of the object system. Our model proceeds in three steps: (1) Description of the process of the model creation. (2) Distinction of different phases. (3) Each phases' usage of different model types. Partial model is provided from the external/internal perspective of a model, the organizational structure, the information systems as object oriented applications, distributed system, and the utilities (machines, premises) system. Especially, the statements of objects can contain a set of domain-based keywords (IDs) that describe the content of the objects. The keywords could be produced by one of the mechanisms that is the resident in the object or other object. The keywords also can be used to compute meaningful results by some other object mechanisms. The statements can be predicates or rules of the object apparently. Relational links, as representatives of both RDF and XML Schema, are relational statements about the object which is characterized by the context of other objects. Thus it's not a mere set of links but is a set of assertion in predicate forms or mechanism calls, involving other objects.

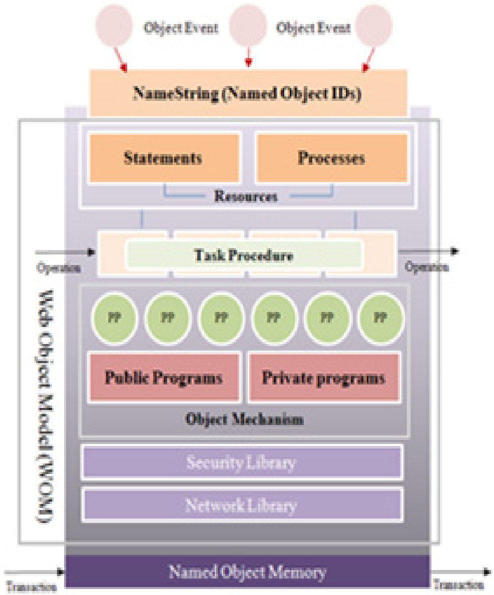

그림 1. 시멘틱 기반 웹 객체 모델 구조
Fig. 1. Semantic based the Architecture of Web Object Model

On the Semantic Web Object, Named Object Memory provides both network library and security library with object mechanism. This mechanism has two programs: (1) Public Programs, (2) Private Programs. Semantic web objects are identified by each PP, which is operated in useful task procedure and can be provided from NameString(Named Object IDs). So a widely-used method uses both local names and associated resources of semantic web objects in order to form its inverted index. Actually, in RDF, semantic links can be established from semantic web object to either objects (by using named object properties). According to the description of semantic web object, only indexing the resources in the statements and its processes may cause problems in which some potentially matched semantic web objects are missing for some task procedure, considering that pp may operate on object mechanism.

Ⅳ. XML-RPC Protocol for SWOM

This protocol specifies the communication between the Configuration protocol server and other protocol entities. Protocol entities, which implement a set of methods that the Configuration protocol server can call, are referred as modules and can also act as servers in some cases. The protocol server performs the following: (1) Manages configuration, including uploading and downloading configuration information. There are two types of configuration. Universal configuration is divided into sections and is shared among all protocol clients. Configuration files are associated with a module and a path, and generate alerts when they are updated. (2) Manages modules, including registering and unregistering modules, and querying for existing modules. (3) Updates and queries a model of the system. In addition to the registered modules, this system model also keeps track of content collections and indexing components. (4) Manages alerts and sends notification to modules when changes occur, if the modules subscribed to those alerts when they registered. The following figure 2 is an example of network traffic between the protocol server and protocol entities.

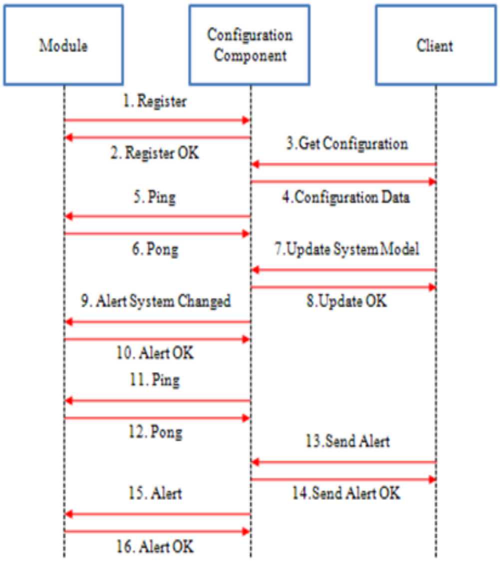

그림 2. SWOM을 위한 네트워크 액티비티
Fig. 2. Network Activity for SWOM

4.1 Data representation in XML

XML parsers and generators are available for most programming languages. An application that adopts XML parsers and generators builds an Semantic based XML document object Model, which is a tree-like data structure that closely resembles the structure of a hierarchical XML object. Internal data structures of the application have to be translated into the Semantic based DOM. For hierarchical data structures, such as lists and trees this translation is almost one to one. Many web applications providing services can be characterized by operating on such hierarchical data structures. Consider an example of

Web pages and database records. The translation of these types of objects is simple. However, applications are not necessarily constrained to hierarchical data structures.

4.2 Message

Transport. This protocol MUST use HTTP for the transport mechanism with semantic web address object mechanism over TCP/IP.

표 1. 커맨드 데이터 타입
Table 1. Command Data Type

Name	Description
CollectionName	The name of content collection
ModuleAddress	An array that contains the location of a module
ModuleRegister	Information used to register a module
AlterType	The type of alert to which a protocol client subscribes
ModuleInfo	Information about a module
AllModuleInfor	Module information for all modules in our system
CollectionDetails	Attributes of a collection
UniversalConfig	Configuration information

MessageSyntax. This message syntax describes the format of the XML body requests and responses.

CommandDataTypes. We describe it in basic unit, double, string, array and structural data types. This protocol uses the data structures specified in the following table. Each of these data structures is either a basic data type with additional restrictions, or the specification of a composite data type.

CollectionName Data Type. The name field associated with a content collection MUST contain a string that contains only the characters in [a-z A-Z 0-9] and a maximum of 16 characters. The lower-case version of the CollectionName field MUST be unique.

ModuleAddress Data Type. This MUST contain the address of the module. It is represented as an array whose values are specified in the following table.

표 2. 모듈주소 데이터 타입
Table 2. ModuleAddress Data Type

Index	Type	Description
0	String	The host name where the module is located
1	Int	The port number used by the protocol server to contact the module

표 3. 모듈레지스터 데이터 타입
Table 3. ModuleRegister Data Type

Name	Type	Description
port	integer	The port number of the module
type	string	The type of the module
name	string	The name of the module
version	string	The version of the module
alerts	Array of strings	The alters to which the module subscribed

ModuleRegister Data Type. This contains information about modules that register with the protocol server. It is represented as a structure, which contains all of the elements specified in the following table. Each module can specify additional elements.

4.3 RPC Stub

The stub performs basic support functions for remote procedure calls. For instance, stubs prepare input and output arguments for transmission among systems with different forms of data representation. The stubs use the RPC runtime to send and receive remote procedure calls. The client stub can also use the runtime to find servers for the client. When a client application calls a remote procedure, the client stub first prepares the input arguments for transmission. The process for preparing arguments for transmission is known as “marshalling”. Marshalling converts call arguments, a stub unmarshals them. Unmarshalling is the process by which a stub disassembles incoming network data and converts it into application data using a format that the local system understands. Marshalling and Unmarshalling both occur twice for each remote procedure call; that is, the client stub marshals input arguments and unmarshals output arguments. Marshalling and unmarshalling permit client and server systems to use different data representations for equivalent data. The stub compiler we use generated stubs by compiling an RPC interface definition written by application developers. The compiler generates marshalling and unmarshalling routines. To build the client for an RPC application, a developer links client application code to the client stubs of all the RPC interface the application uses. To build the server, the developer links the server application code to the corresponding server stubs.

4.4 Object Serialization

Object serialization is the ability of an object to write its complete state and the complete state of any objects that it references to an output stream; and then, at some later time, to recreate itself by reading its serialized state from an input stream. The stream may be a file(document or content), a byte array or a stream associated with a TCP/IP socket.

Ⅴ. Experiment and Evaluation

In order to test the our solution described in this research, we used the Random Field Semantic Web Object Model on the 2-dimensional 70 x 70 structure of spins. Within this approach, which is typical in the Web Object type calculations, the spin dynamics with the two +1 and －1 states were tested. The local condition of a spin flip from one state into another is fulfilled, when the numerical value of the total magnetic field acting on this spin changes its algebraic sign. What we simulated here is the hysteresis loop of the (n x n) structure (n=70), thus the dependence on magnetization. The calculations were carried out on the computer cluster built from 16 PC-type computers, each equipped with the 8G of RAM and x processors. An Ethernet network at the 1000Mb/sec. speed connected clustered computers. Table 4 provides the times of computation the values of calculated exchange-bias and co-activities along with their standard deviations.

표 4. 계산에 의한 시간 값

Table 4. The times of computation the values

Num of repetitions	Exchange-bias (a.u.)	Coercivity field (a.u.)	Time of computation
64	0.427±0.009	4.228±0.014	580 sec.
128	0.419±0.006	4.242±0.009	940 sec.
256	0.412±0.007	4.186±0.031	1960 sec.
512	0.397±0.007	4.169±0.025	3552 sec.
1024	0.415±0.003	4.208±0.011	7175 sec.

The results possess a statistical character, while values of uncertainties with a tendency to minimize. The time of computation is more or less proportional than the number of repetitions. Some deviations from an exact line dependency are caused by network performance. The simulations were controlled in a local network as well as presented here are extremely useful due to their remote character with the independence from the operating system applied.

The distributions of the transaction times for XML-RPC-based in general and Semantic Web Object (SWO) based on XML-RPC-based implementations for tests over the local network are shown Figure 2. From the figure we observe that the performance does not vary much between the runs. The result is expected due to relatively static condition of the local network. For the smallest transaction, IsPrime ( ), the SWO based on XML-RPC transaction take roughly 70 ms. while this is slightly more than the time required by the XML-RPC (which takes roughly 60 ms) the difference is not likely significant in systems involving human interaction.

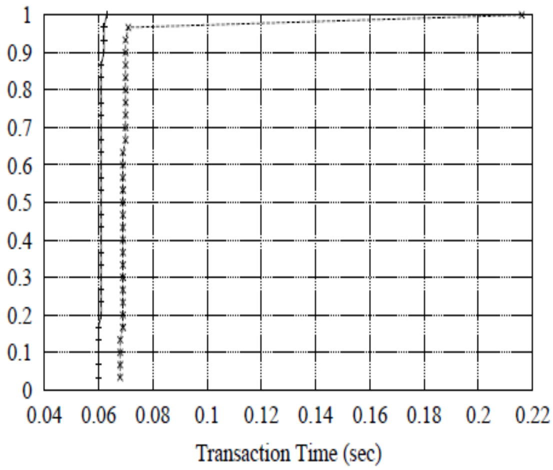

그림 3. IsPrime ( ) 트랜잭션 시간
Fig. 3. IsPrime ( ) Transaction time

표 5. 트랜잭션 크기
Table 5. Transaction Size

Transaction	Request Size (bytes)	Reponse Size (bytes)
IsPrime - XML-RPC IsPrime - SWO based on XML-RPC	6 ≈339	1 ≈332
Average - XML-RPC Average - SWO based on XML-RPC	400,006 ≈2,513,756	8 ≈334
GRNums - XML-RPC GRNums - SWO based on XML-RPC	6 ≈346	400,000 ≈2,513,812

Next, we examine the overhead incurred by the two versions that happens before or after the transaction is sent across the network. Table 5. shows the median transaction time for the XML-RPC measured by the application, as well as the median difference process and the length of the TCP connection measured from the packet traces. From this table, we can make several observations: (1) There is little pre- or post-processing overhead involved in the IsPrime ( ) method call. (2) The Average ( ) and GetRandNums ( ) methods take roughly the same amount of time to execute when measured by the application.

Ⅵ. Conclusion

Architectural Semantic Web Object Model can be designed by our research project. In short, the steps for WOM architecture processes are (1) Name Object IDs with object event and a multi-value attribute are operated into complex type using resources. (2) Named object from the operating task procedure is carried out object mechanism in both public programs and private programs, and is assisted by each library between security library and network library. (3) Finally, Named Object Memory is carried out. Through more advanced and improved research, we believe simple and small elements can produce a collective intelligence when their interactions are well designed.

References

[1] D. J. Armstrong, "The quarks of object-oriented Development," *Communications of the ACM*, Vol.49, No.2, pp.123-128, 2006.

[2] C. Bizer and R. Cyganiak, "D2R Server-Publishing Relational Database on the Semantic Web," *In Proceedings of the 5th International Semantic Web Conference(ISWC)*, 2006.

[3] X. Dong, A. Halevy, and J. Madhavan, "Reference reconciliation in complex information spaces," *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp.85-96, 2005.

[4] E. Oren and R. Delbru, "ActiveRDF: Object -oriented RDF in Ruby," *In Proceedings of the ESWC Workshop on Scripting for the Semantic Web*, 2006.

[5] L. Sauermann et, al. "Semantic Desktop 2.0: The Gnowsis Experience," *In Proceedings of the International Semantic Web Conference (ISWC)*, pp.887-900, 2006.

[6] D. Vrandecic, "Deep Integration of Scripting Languages and Semantic Web Technologies," *In Proceedings of the ESWC Workshop on Scripting for the Semantic Web,* vol.135, 2005.

[7] A. Kalyanpur, D. Jimnez, S. Battle, and J. Padget, "Automatic mapping of OWL ontologies into Java," *In Proceedings of the International Conference on Software Engineering & Knowledge Engineering(SEKE)*, 2004.

[8] Diane Kelly, "Evaluating Personal Information Management Behaviors and Tools," *Comm- unications of the ACM*, Vol.49, No.1, pp.84-86, 2006.

[9] Leo Sauermann, "The Semantic Desktop - a Basis for Personal Knowledge Management," *In Proceedings of the I-KNOW2005*, pp.294-301, 2005.

[10] L. van Elst and M. Kiesel, "Generating and Integrating Evidence for Ontology Mappings," *In Proceedings of the 14th EKAW*, Vol.3257, pp.15-29, 2004.

[11] H. Xiao and I. F. Cruz. "A Multi-Ontology Approach for Personal Information Manage- ment," *In Proceedings of Semantic Desktop Workshop at the ISWC,* Vol.175, 2005.

Acknowledgments.

This research is supported by HanSung University and center for Smart Address Standardization of excellence.

저자소개

김진홍(Jin-Hong Kim)

2006년 성균관대학교 컴퓨터공학과 박사

2009년 VODAFONE Co. Senior Researcher

2010 년 성균관대학교 정보통신공학과 연구조교수

2011년~현재 한성대학교 정보통신공학과 겸임교수

※ 관심분야 : 웹 객체 모델링, 소프트웨어공학, 스마트 플랫폼

김현호(Hyun-Ho Kim)

1987년 성균관대학교 전자공학과 (공학사)

1997년 연세대학교 전자계산전공 (공학석사)

2006년 성균관대학교 컴퓨터공학 (공학박사)

1988~2000 LG전자 컴퓨터사업부 SE

2003년 ~ 현재 명지전문대학 컴퓨터정보과 부교수

※ 관심분야 : 웹 객체 모델링, Mobile Computing, Ad-Hoc Networks, 스마트 플랫폼