Toward a new definition to term Software Engineering based on the E-Services

Seyyed Mohsen HASHEMI*, Mohammadreza RAZZAZI**
*Computer Eng. Dept., Science and Research Branch, Islamic Azad University, Tehran, Iran
**Computer Eng. Dept., AmirKabir University of Technology, Tehran, Iran
hashemi@srbiau.ac.ir, razzazi@aut.ac.ir

Abstract: *This is time to extend the definition of the term 'Software Engineering' based on the term 'E-Services'. Thus, the definition of the term E-Services must be revised with the concern of future technologies and trends such as Grid and Globalization. Paper has presented a UML-based Ontology, which called ISRUP SOSA to realize pragmatically E-Services in the Global Village Governances. In addition, to present a new definition of E-Services instead of E-Government, E-Business, E-Commerce, and E-Governance Services, the globalization trends have specified that to realize the suggested ISRUP SOSA Ontology in the world, it is necessary to propose the specification of ISRUP SOSA to promote the role of the E-Services in the commercial and governmental unbundling. Meanwhile, the main features of the ISRUP SOSA are Grid-Centric, X2Y-Centric, E-Service Centric, Collaborative-Centric base on the next generation Governance Frameworks, and they are Data and Service Integration-Oriented. Therefore, paper represents the ISRUP SOSA Ontology to the term 'E-Services' in a new definition of Software Engineering in both way of the representation of an Ontology; UML and XML.*

Keywords: *software engineering, e-services, ontology, ISRUP, collaborative.*

1. Introduction

Nowadays, Software development projects are based on TPS, MIS, and DSS and organizations are maturing to establish the baselines of their Business, Governance, and Technology architectures (and entirely Reference Architectures) based on the Service Oriented Strategies and Architectures (SOSA). The traditional software development projects set up Business, Governance, and Technology baselines on the Business Processes or Business Usecases. That Business Processes must be unbundled through the Electronic Services to realize the ISRUP SOSA. This is time to extend the definition of the term 'Software Engineering' based on the term 'E-Services'. Thus, the definition of the term E-Services must be revised with the concern of future technologies such as Grid and future trends like the globalization. This paper has presented a UML-based Ontology, which called SOSA to realize pragmatically E-Services in the Global Village Governances.

ISRUP stands as Integrated Services-Information Systems based on the Rational Unified Process (RUP®) terminology. ISRUP is an E-Services Framework for agile governance architecting through Unified Modeling Language (UML), SOSA Ontology, GVGRM Reference Model, and RUP terminology to improve the governance architecture of the Governances. ISRUP E-Services Framework has 40 Governance Patterns to apply an iterative process for radical improvement by way of information system architecting (Hashemi, 2006).

Chapter 2 specifies the Background for Software and Service Engineering Ontology. This chapter emphasizes to use UML to Ontology modeling in Software Engineering. Two types of ontology in software engineering have been studied in this chapter. Meanwhile a new definition of software engineering suggested in this chapter, which it is the extension of IEEE's definition plus two new considerable subjects: the Globalization and E-Services. Finally, the Background for E-Services Ontology is reviewed to find the different concepts and aspects from some definitions, which would be needed to model the Ontology of E-Services in the Reference Model of the Global Village Governance.

Chapter 3 suggests E-Services Ontology called SOSA for the Global Village Governance. In addition, the features of SOSA as a UML-based Ontology to E-Services in GVGRM are proposed in this chapter. Meanwhile, SOSA is proposed in both UML Class diagram and XML format to realize a new Reference Model for E-Services in the Global Village Governance. Furthermore, SOSA Ontology to X2Y Structure (G, B, C), E-Services Resources, E-Services Actors, and to E-Services Actions and Associations are elaborated in this chapter.

2. Software and Service Engineering Ontology

Since the most of E-Services efforts have been designed and implemented through E-Commerce, E-Businesses, and E-Governments and/or E-Governances projects, it is necessary to study the detail of their projects as the references (Hashemi, 2008). These details are out of the scope of this paper. E-Xs projects (E-Government or E-Business or E-Commerce, or E-Governance) have the same behaviors by the time we are developing them as some software development projects. E-Services are a highly generic term usually referring to provide or consume of services via the computer networks. E-Services include "E-Commerce", although they may also include non-commercial services. Non E-Commerce E-Services include (at least some) "E-Government" services (WIKI, 2008). Regarding to the definitions of Turban (Turban, 2006), all E-X's Services are 'Electronic Business Services'. To make easy the using of this term, it is considered to use term 'Electronic Services' (or E-Services) instead of 'Electronic Business Services' entire of the paper.

Two types of ontology representation in software engineering have been studied in this chapter. Furthermore, it has been cleared that by the time using UML class diagram plus OCL language for ontology modeling, the represented model could be transformed in MOF XML and Z-Language formats. Meanwhile a new definition of software engineering suggested in this chapter, which it is the extension of IEEE's (IEEE, 1990) definition plus two new considerable subjects: the Globalization and E-Services.

2.1. E-X and Software Engineering

Since the term software engineering was coined by Brian Randell and popularized by F.L. Bauer during the NATO Software Engineering Conference in 1968 (Dijkstra, Edsger W, 2003), many other definitions for the term software engineering are defined like; (Pecht, Michael, 1995), (IEEE, 1990), (Pehrson, Ronald J., 1996), (Sommerville, Ian, 1982-2007), (SW,1972), (Akram I. Salah, 2002), (Mills, Harlan D., et al., 1990), and (Dennis J. Fraiey, 2008). There are two definitions of the term software engineering from IEEE to review as follows, before to express the extension of the last definition to promote the software engineering discipline.

The baseline of definitions to the term "Software Engineering" defined in IEEE in 1990 (IEEE, 1990) as:

"Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software."

The last definition to the term “Software Engineering” is presented by Dennis from Raytheon Corporation and published by IEEE in 2008 (Dennis J. Fraiey, 2008) as:

"Software engineering is the application and/or study of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, that has an impact on the lives, property, economy, or security of people or the national defense; that is, the application of engineering to software."

As the mentioned definitions, the spectrum of these definitions is restricted on a nation wise. In the real world, the terms E-Commerce, E-Business, E-Government, and E-Governance (and in a short term E-Xs) cannot be realized without the results of the software engineering projects. These results are some electronic services, which have an impact on the lives, property, economy, or security of people or the national defense as Dennis (Dennis, 2008) submitted. In other words, all E-Xs projects are software engineering projects and their differences could be distilled in their Business Architectures, Enterprise Architectures, and Technology Architectures or entirely in their Governance Architectures.

2.2. E-Services and Software Engineering

As Daniela said (Daniela, 2003), an E-Service is a software artifact (delivered over the Internet) that interacts with its clients in order to perform a specified task. A client can be either a human user, or another E-Service. Also Lu said that, E-services are likely to push the limits of software engineering in terms of analysis, design, security, and testing (J. Lu et al, 2007).

Since the Software Engineering Discipline is a multidisciplinary role and it has been derived from the Business Disciplines, Computer Science, and Information Science; it was supposed to use the Service and E-Service terms instead of each other, in the whole of this paper. Meanwhile, Web Services or Grid Services are some kinds of E-Services in this contribution. We suggested the following definition to the term of “ Software Engineering” as extension of what Dennis (Dennis, 2008) submitted in addition to the definition of IEEE (IEEE, 1990).

"Software engineering is the application and/or study of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, that has an impact on E-Services; that is, the application of engineering to software."

In addition to, it has been proposed a definition to the term of “E-Services” in the Global Village Governance in the following sections.

2.3. Unified Modeling Language

The Unified Modeling Language (UML) is a general-purpose modeling language for software-intensive systems that designed to support Meta-models that closely related to the ontology. UML and OCL are often used to describe and analyze the relations between concepts (Söderström, et al, 2001). Object Constraint Language (OCL), as a part of UML has emerged from practical needs to sharpen and tighten UML class diagram to formulate structural restrictions (Martin Gogolla, 2004).Though there are actually various "Set Theories" (J.M. Spivey, 1992), it has been supposed to use here simple concepts as defined in the Z notation. Luiz presents an empirical study in which a formal method (Z notation) was used in an articulated way with a semi-formal technique (UML diagrams). He shows the correspondence between modeling elements of UML and Z, and between relationships in Z and UML (Luiz Eduardo Galvão Martins, 2004). Respecting to Luiz's work and using OCL in addition to pure UML, the ontology of E-Services could be entirely presented.

2.4. Ontology in Software Engineering

Software engineers have more recently started using the same term to express a shared understanding of what is believed to exist. Gruber defined 'Ontology' in 1995 as "an explicit specification of a conceptualization." This definition was somewhat altered by Borst in 1997 who defined 'ontology' as "a formal specification of a shared conceptualization", As Ziv said (Ziv Baida, 2006). In both computer science and information science, ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to define the domain (WIKI, 2008). As the mentioned definition, in this paper, the domain is the E-Services in the Global Village Governance. Wongthongtham (Wongthongtham, 2005), proposed the ontology for software engineering in two layers as *Generic ontology* (is a set of software engineering terms including the vocabulary, the semantic interconnections, and some simple rules of inference and logic for software development), and *Application-specific ontology* (is an explicit specification of software engineering for a particular software development project). In addition to, Barry (Barry Robert Pekilis, 2006) specified that the Ontology for modeling software philosophically and practically support the use of platform independent knowledge at the domain-level. Paper used the general ontology in software engineering to represent the E-Services by the time of using UML.

2.5. Ontology Modeling through UML

(S. Cranefield and Martin Purvis, 1999), (KSL, 1994), (NCITS, 1998), (A. Farquhar, et. al., 1996), widely used traditional approaches for ontology modeling that called Knowledge Interchange Format (KIF) and KL-ONE style knowledge representation languages (Brachman and Schmolze, 1985). As Wongthongtham classified the Ontology Modeling Languages and Notations (Wongthongtham et al., 2005), There are generally many ontology representation languages for creating ontology including;

- Knowledge Interchange Format (KIF) (M.R. Genesereth, R.E. Fikes, et al., 1992),
- Simple HTML Ontology Extension (SHOE) (S. Luke & J. Heflin, 2000),
- ISO standard for describing knowledge structures (Topic Maps) (G. Librelotto, et. al., 2003),
- Ontology Exchange Language (XOL) (R. Karp, et al., 1999),
- Ontology Markup Language (OML) (R. Kent, 1998),
- Ontology Inference Layer (OIL) (I. Horrocks et al., 2000),
- DAML+OIL (I. Horrocks & F. van Harmelen, 2001), and
- Web Ontology Language (OWL) (McGuinness, D.L. & F.V. Harmelen, 2004), (Lacourba, V., 2004).

Most of the mentioned languages are based on the XML. They are under development and they have not good convergences to reach a ubiquity. For instance, while OIL has been developed, DAML appeared. Moreover, after two developments, the integration of them has done. It means that they must be analyzed again from the basis. e.g., OWL is being developed based on DAML+OIL. The main issue is that UML can be totally transformed to all the mentioned languages.

As Gašević cleared, a critical aspect of modeling and designing ontology is lack of graphical notation (D. Gašević et al., 2004). Stephen (Stephen Cranefield, et al., 2001) and

Wongthongtham (Wongthongtham, 2005), (S. Cranefield and M. Purvis, 1999) and (F. Bergenti and A. Poggi, 2000) use UML as an ontology modeling language. Additionally, (Weise, 2006), (Baclawski, et al., 2006), and (Stephen Cranefield, et al., 2001) have reported the UML as an ontology development environment. Pedrinaci compiled (C. Pedrinaci, et al., 2004) the benefits of using UML for ontology development have been extensively argued in (S. Cranefield and M. Purvis, 1999), (S. Cranefield, 2001), (P. Kogut, et al., 2002), (Dragan Đurić, 2004), and (S. Cranefield, 2001). Some of those benefits are:

- UML is a standard language;
- UML is a graphical notation based on many years of experience in software analysis and design, which is currently supported by widely-adopted CASE tools that are more accessible to software practitioners than current ontology tools;
- Agent-based systems will need to interact with legacy enterprise systems, which often have UML models;
- Knowledge expressed using UML is directly accessible for human comprehension and for machine processing;
- Thanks to the modular nature of object oriented modeling, the knowledge in a UML model can be changed without affecting other features.

2.6. Electronic Services and Service-Oriented Modeling

After reviewing some empirical definitions from E-Xs (Hashemi, 2007), and before to reveal the way of modeling E-Services Ontology, it is essential to review the background of efforts, which done to promote the role of E-Services in the world. As Ziv presented (Ziv Baida, 2006), the definitions of terms Service, E-Services, and web services are not entirely well defined in business, enterprise, and software architectures. Although, IBM defines a service as a repeatable business task e.g., check customer credit and open new account, in this paper:

"E-Service is a full-grown and fine-grained repeatable part of a Function over the Grid."

With this concern of definition, E-Government, E-Commerce, and E-Business services could be considered to dance in grid-based choreography to realize the End-to-End Electronic Processes. Elaborating of the End-to-End Electronic Processes encompass the E-Services in the Global Village Governance.

Since correspondences between ontology and UML have been established (Evermann and Y. Wand, 2005) and (DSTC. 2004), many researches present SOA-related models through UML. For instance, James (James Densmore, et al., 2007) has depicted a map between UML elements and SOA issues, and Zhang (Zhang, 2008) pointed the modeling of SOA based on UML is a hot research topic today. These models need a Reference Model within the E-Services domain.

Furthermore, Service-oriented modeling is elaborated by Bell (Bell, 2008) as a software development methodology that employs disciplines and a universal language to provide tactical and strategic solutions to enterprise problems. In addition to, Service-oriented modeling encourages viewing software entities as 'assets' (service-oriented assets), and refers to these assets collectively as 'E-Services'. These researches direct that the modeling of E-Services in any kinds of representation is being started and the suggested Ontology for E-Services (SOSA) would be considered as the baseline of service-oriented modeling in the future.

2.7. UML for E-Services Ontology

Ziv (Ziv Baida, 2006) specified the sharing as a key issue in his presented service ontology. In addition, Ziv explored that ontology for E-Services is required to enable customers and suppliers buy and sell services via the Internet, to enable joint offerings of various suppliers over the web, and to enable business analysts conduct a business analysis for networked enterprises. In addition to, Ziv elaborated that the conceptualization has to be shared in order to support communication between humans, computers, and enterprises. Humans, computers, Governments, and enterprises will be modeled also in the suggested Ontology as Service Actors.

As the result of (K. Baclawski et al., 2001), (K. Falkovych, 2002), (K. Falkovych, et al., 2004), (Elisa F. Kendall, Mark E. Dutra, 2002), and (A. Felfernig et al., 2000) researches, the main reason of using UML for E-Services Ontology Modeling is that the UML models could be transformed into OIL or DAML or DAML+OIL ontology or finally OWL ontology. OIL, DAML, DAML+OIL, and OWL are ontology languages based on XML as before mentioned. All UML model, especially Class diagrams could be transformed into each mentioned Ontology language, through XMI as a kind of XML standard.

Since the Ontology could be defined in terms of relations, classes, (and subclasses), functions, and sets (KSL, 1994). This feature makes the UML Class diagrams fit to represent the relations, classes (and subclasses), functions and sets of an ontology. Stephan also (S. Cranefield and Martin Purvis, 1999) presents the ontology modeling Based on a subset of the Unified Modeling Language together with its associated Object Constraint Language. In addition to, Luís (Luís Mota, 2003) elaborated entire differences that exist between ONTOLOGY and a CLASS, especially presented by a UML class diagram. In the mentioned Ontology Class diagram, it is obvious that the Ontology is not the Class, although it can have some classes with their properties and methods. In other words, Ontology could be realized through an *abstract singleton class* plus zero or more *classes*.

It has been shown that both UML and XML could be used to the suggested Ontology for E-Services (ISRUP SOSA), which would be considered as the baseline of service-oriented modeling in the future. Meanwhile, it has been depicted that all UML diagrams, especially Class diagrams could be transformed into each mentioned Ontology language, through XMI as a kind of XML standard.

3. ISRUP SOSA: Suggested E-Services Ontology for the E-Services

Nowadays, Software development projects are based on TPS, MIS, and DSS and organizations are maturing to establish the baselines of their Business, Governance, and Technology architectures (and entirely Reference Architectures) based on the Service Oriented Strategies and Architectures (SOSA). The traditional software development projects set up Business, Governance, and Technology baselines on the Business Processes or Business Usecases and the Business Processes must be unbundled through the Electronic Services to realize the SOSA. SOSA is proposed in both UML Class diagram and XML format to realize a new Reference Model for E-Services in the Global Village Governance. This makes easy to use both of the Ontology representation modeling languages through XMI for E-Services. SOSA Ontology to *X2Y Structure (G, B, C)*, *E-Services Resources, E-Services Actors*, and to E-*Services Actions and Associations* are elaborated in the next sections of this chapter.

3.1. SOSA as UML-based Ontology to E-Services in GVGRM

With relation to the Suggested E-Services Ontology in the Global Village Governance, SOSA, the Next Generation Governance Frameworks offer a full range of governance patterns that include the development of comprehensive strategies to integrate internet based solutions, E-Business, E-Commerce, and E-Government disciplines, E-Service technologies, and methodologies into the software intensive organizations. In addition to, the Next Generation Governance Frameworks focus on the facilitation of the working of Service Requesters, Service Brokers, and Service Providers, in both Provider-Centric SOSA and Consumer-Centric SOSA perspectives over a Grid. The Next Generation Governance Frameworks are Electronic Governance Information Systems (or Software) to manage their Integrated Services. While SOSA is based on pure UML, all Reference Architectures that could be presented in future in software development and implementations, can be represented also in UML.

3.2. SOSA Ontology to X2Y Structure (G, B, C)

The theme of SOSA Ontology is to specify the role of E-Services to promote the commercial and governmental unbundling affairs by way of using the Next Generation Governance Frameworks were developed to leverage E-Governments, E-Businesses, and E-Commerce though just the added value E-Services. Moreover, SOSA Ontology shows the main features of the Next Generation Governance Frameworks are X2Y-Centric (figure 1), Grid-Centric, E-Service Centric, Collaborative-Centric which are based on the new Governance Architecture Frameworks, and are oriented toward Data and Service Integration.

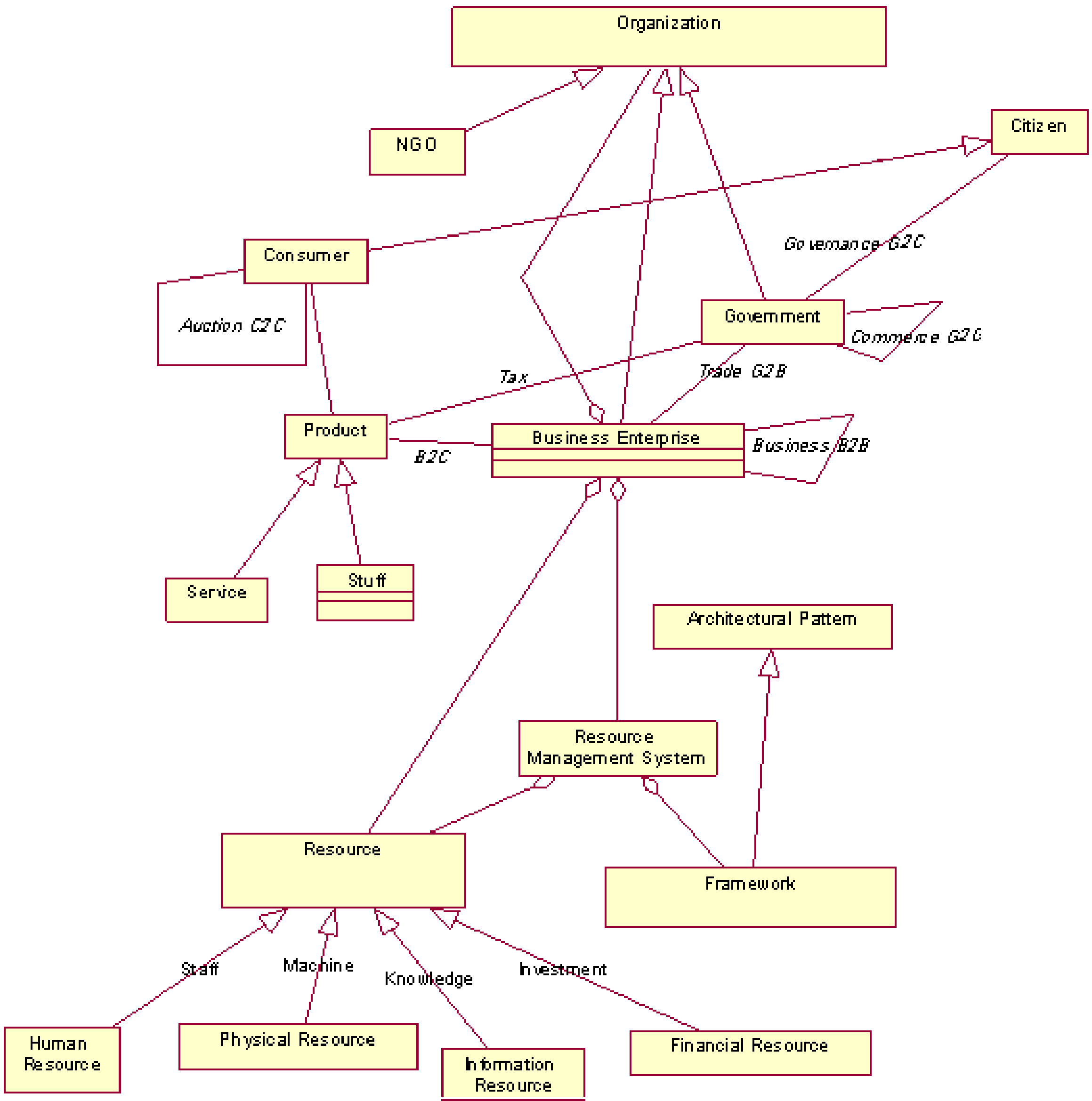

Fig. 1. SOSA X2Y Structure (G, B, C)

3.3. SOSA Ontology to E-Services Resources

In SOSA Ontology, a Service includes a Service Goal, a Service Specification, a Service Resource Collaboration model, and a Service Result as figure 2. In addition to, a Service Uses four kind of Resources; Physical Resources, Financial Resources, Human Resources, and in a general manner, Information Resources on the top of them. The Service Result would be a kind of Service Resources as an outcome. Additionally, in that UML class diagram, syntax and semantic of the relationships for Added-Value Services, Services, Service Resources, and Service Collaboration are precisely modeled. In SOSA Ontology, the added-value Service as a repeatable task could be compiled as *Simple* or *Composite*, and they could be compiled from the Public or Private Services. Furthermore, each Service has a Grid Node Identifier (Grid Node ID) to represent its association with the Grid nodes (Hashemi, 2006).

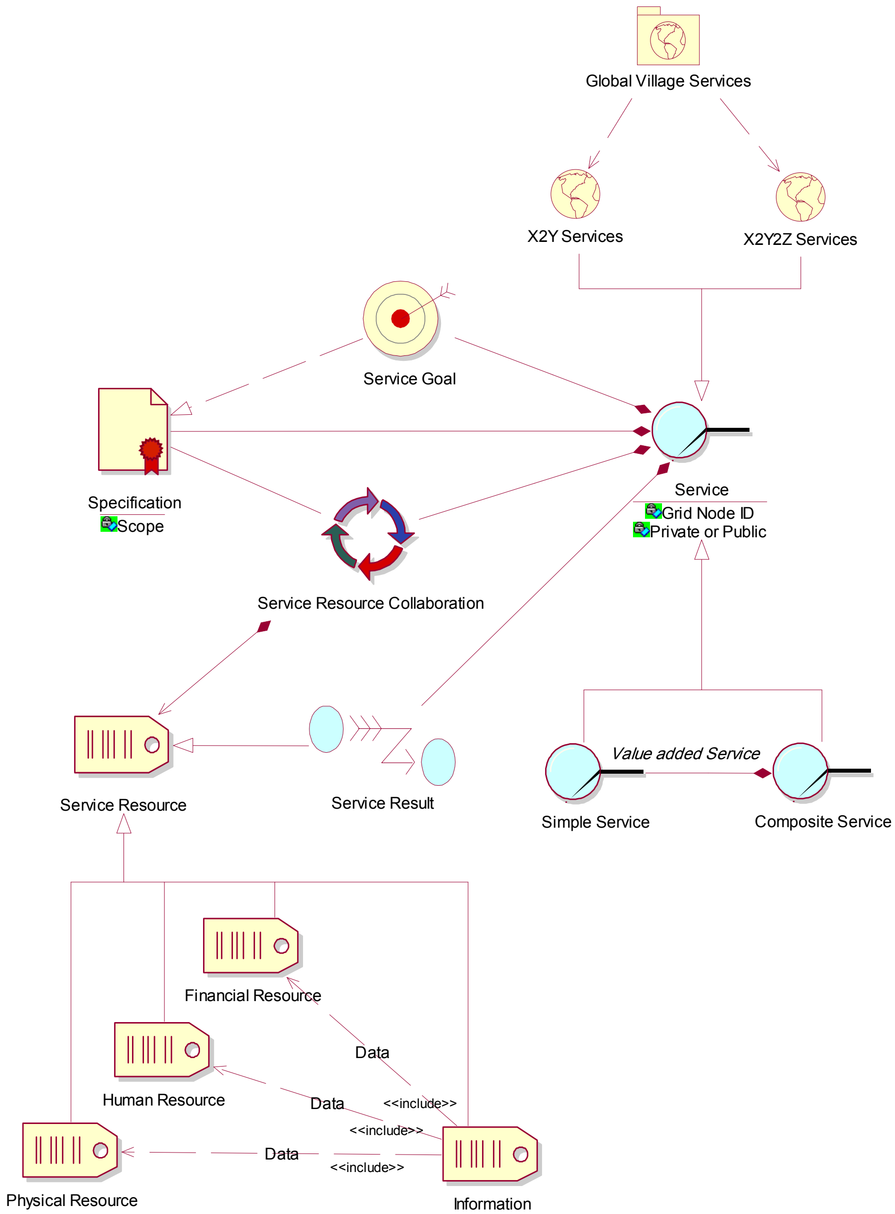

Fig. 2. UML Class Diagram; the Elements of SOSA E-Services Added-Value Services

Some researches like (West, D. M., 2006), (Stowers, G., 1999), (O. Glassey, 2004), and (G.P. Dias, J.A. Rafael, 2006) adjust the focus of existing one point to access the resources of the E-Services, this point could be the Global Village Governance portal, which serves any kind of provided and on demanded E-Services to any kind of the Services Actors that detailed in the next section.

3.4. SOSA Ontology to E-Services Actors

Any kind of collaborative agents or Actors could be related to the Global Village Governance are Government, Business and Residents (Citizens plus Consumer). SOSA Ontology to E-Services Actors is depicted in figure 3.

Regarding to the elaborated XML-based Ontology in chapter 2, the SOSA Ontology for E-Services Actors also represented in MOF/XML-based. The XML files are available to access if they would be wanted.

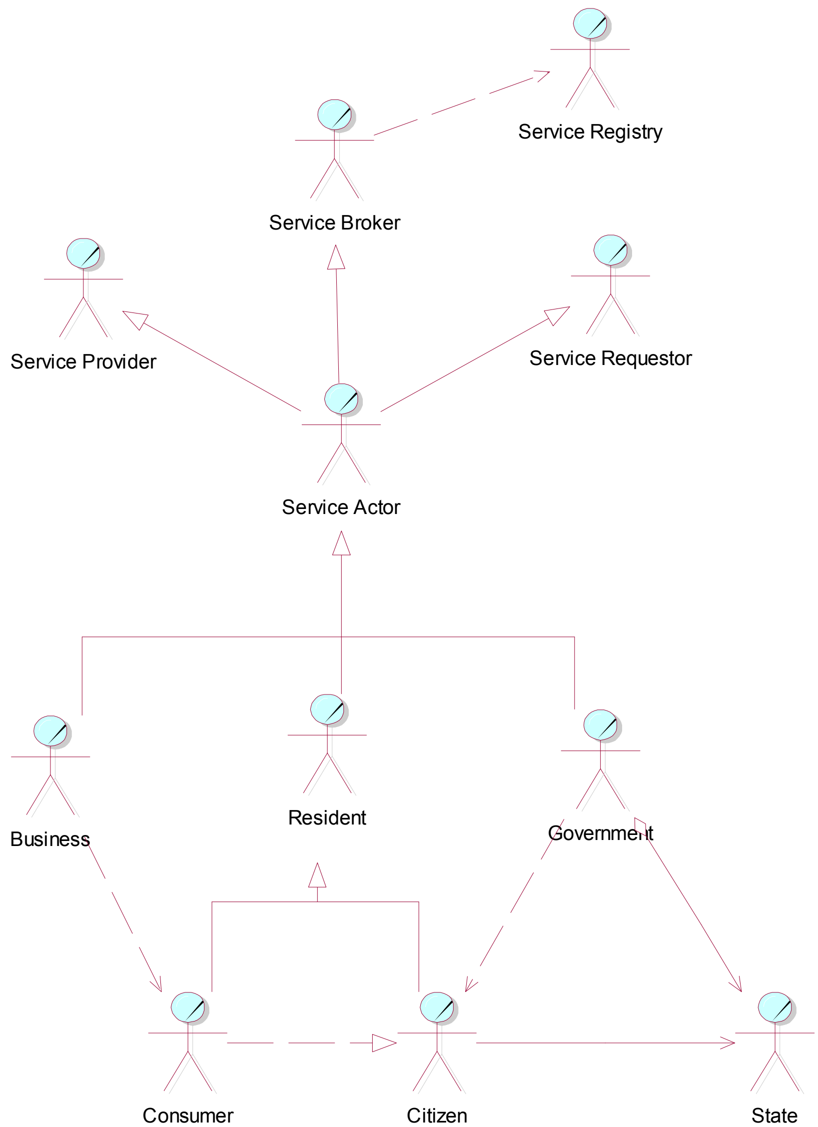

Fig. 3. UML Class Diagram; SOSA Ontology to E-Services Actors

References

[1] IEEE, "IEEE Standard Glossary of Software Engineering Terminology," *IEEE std 610.12*, 1990.

[2] J. Lu et al, *E-Service Intelligence: An Introduction, Studies in Computational Intelligence (SCI)*, 37, 1 Springer, 2007.

[3] E. Turban et al, *Electronic Commerce: a Managerial Perspective*, Prentice – Hall, 2006.

[4] Ziv Baida, *Software-aided Service Bundling Intelligent Methods & Tools for Graphical Service Modeling, PhD thesis*, Universiteit Amsterdam, 2006.

[5] T. Zhang et al, *Enterprise Information Systems Journal*, Vol. 2, No. 3, August 2008, pp. 239-257.

Authors

Seyyed Mohsen HASHEMI received the M.S. degree in Computer Science from Amirkabir University of Technology (Tehran Polytechnic University) in 2003, and the PhD degree in Computer Science from the Azad University in 2009. Both of them ware under supervision of Professor Mohammadreza Razzazi. Moreover, he is currently a faculty member at Science and Research Branch, Azad University, Tehran. His current research interests include Software Intensive Systems, E-X systems (E-Commerce, E-Government, E-Business, and so on), Global Village Services, Grid Computing, IBM SSME, Business Modeling, and Agile Enterprise Architecting through ISRUP.

Mohammadreza RAZZAZI received the MS degree in Computer Science from Stanford University and the PhD degree in Computer Science from the University of California at Santa Barbara. Currently he is an Associate Professor in the Department of Computer Engineering and IT at Amirkabir University of Technology, Tehran. His primary areas of research are Distributed Processing on Mobile Phones, Software Development on Mobile Phones, Grid Computing, Software Engineering Best Practices, Electronic Commerce, Computational Geometry and Computer Graphics.