A Ubiquitous and Context-Aware Framework for Supporting Virtual Engineering Services

Jae Yeol Lee[1], Dong Woo Seo[1], Kwangsoo Kim[2] and Hyun Kim[3]

[1]Chonnam National University, jaeyeol@chonnam.ac.kr, seodongwoo@paran.com
[2]Pohang University of Science and Technology, kskim@postech.ac.kr
[3]Electronics and Telecommunications Research Institute, hyunkim@etri.re.kr

ABSTRACT

Context-aware engineering services in ubiquitous environments are emerging as a viable alternative to traditional engineering services. In this paper, we present a **U**biquitous and **C**ontext-**A**ware computing **F**ramework for collaborative virtual **E**ngineering (U-CAFÉ) services by adopting semantic web-based context-awareness. Topic Maps are used to represent dynamically changing engineering contexts and to query and reason about the contexts for supporting ubiquitous and human-centered engineering services. We discuss how web services and JINI services are used to support engineering service federations and seamless interactions among persons, devices, and various kinds of engineering services.

Keywords: engineering web services, business process management, JINI, semantic web, context-aware and ubiquitous computing, Topic Map.

1. INTRODUCTION

Internet can be regarded as a complex system of service chains. CAX tools can be provided as distributed engineering services through the Internet, enabling in a distributed product development environment. This incorporates different engineering services and makes them available for automatic transactions[1,10]. Note that computing paradigm is also moving toward pervasive, ubiquitous environments in which devices, software agents, and engineering services are all expected to seamlessly integrate and cooperate in support of human objectives – anticipating needs, negotiating for services, acting on our behalf, and delivering services in anywhere, any-time fashion [4,13].

Several research efforts have addressed ways in which computer-network oriented design and manufacturing services will be able to support collaborative product development and suggest what an engineering service-oriented tool or system should look like in such an environment. Sobolewski presented federated P2P services in concurrent engineering environments[12]. Kim *et al.* proposed a framework for sharing product information across enterprises in a distributed environment[6]. Lee *et al.* proposed a web-enabled approach to feature-based part and assembly modeling in a distributed environment[8,9]. However, these works focused on data-centric engineering application integration rather than process-centric integration. The full potential of engineering services as an integration platform will be achieved only when applications and engineering processes are able to integrate their complex interactions by using a standard process integration model[2]. For this reason, Lee *et al.* proposed a process-centric approach to supporting engineering services[10]. However, most of them focused on computer-centered services rather than human-centered services. In a ubiquitous and context-aware computing environment, one of the most important issues is the seamless integration of devices, service components, and users by understanding contexts and by sharing and reasoning them in support of collaborative and human-centered engineering services.

In this paper, we present a ubiquitous and context-aware computing framework (U-CAFÉ) for supporting collaborative virtual engineering services by adopting semantic web-based context-awareness. Topic Maps are used to query and reason about engineering service-related contexts. By managing and reasoning engineering service contexts, it can reduce difficulty and cost in building user-oriented engineering knowledge management and sharing, which can provide relevant engineering services and information to meet service requestors on the basis of their contexts. The remainder of the paper is organized as follows. Section 2 overviews U-CAFÉ. Section 3 presents how to execute engineering web services in a ubiquitous and context-aware environment. Section 4 shows some implementation results. Finally, section 5 concludes with some remarks.

2. OVERVIEW OF U-CAFÉ

2.1 Standards for representing engineering service-related semantics

Computer-aided design and engineering tools can be provided as distributed service components through the web service-based wrapping operations[14]. Web services are targeted specifically at providing a loosely-coupled architecture designed for exchanging information over the Internet. One of the goals of web services is to realize a distributed architecture across the web in a platform independent manner. Hence, compliance with well-established and generally accepted standards is essential. One of the most important standards is XML, which acts as the universal data format for web services. SOAP (Simple Object Access Protocol) is an XML-based protocol for service invocation. WSDL (Web Services Description Language) is an XML vocabulary to describe operational information about the service. The UDDI (Universal Description, Discovery and Integration) registry acts as a directory of available services and service providers. Moreover, there is a need for ontology when applying search and semantic matching for web services. Topic Map and OWL are typical languages for expressing sophisticated class definitions and properties[11,15].
Context-aware systems are computer systems that can provide relevant services and information to users by exploiting contexts. By contexts, we mean information about a location, its environmental attributes, and the people, devices, objects and software agents it contains. Contexts may also include system capabilities, services offered and sought, the activities and tasks in which people and computing entities are engaged, and their situational roles, beliefs, and intentions. Note that ontologies for contexts are key requirements for building a context-aware engineering service framework for the following reasons[4,13]: (i) a common ontology enables knowledge sharing in a ubiquitous and context-aware environment, (ii) ontologies with well defined declarative semantics provide a means for reasoning about contextual information, (iii) explicitly represented ontologies allow devices and agents to work together to interoperate each other, and (iv) ontologies support easy adaptation to the dynamic changes of engineering service environment. Normally, context-aware computing is very popular in the areas of building intelligent meeting rooms, supporting intelligent robots, and providing smart spaces for easy living[3,4,7]. However, there is little research work that is applied to virtual engineering services in a collaborative and ubiquitous environment, although the need for such requirements is increasing rapidly for achieving true engineering service integration and collaboration.

2.2 System architecture

The primary objective of this research is to propose a generic architecture that supports virtual engineering services, with collaborative and adaptive capabilities, in a ubiquitous and context-aware environment. Fig. 1 shows the proposed architecture for context-aware engineering services. Both web services and JINI™ services are used for flexible and easy communication to support various kinds of engineering services and devices. Each engineering service can be easily represented as a web service component. Further, engineering federation can be achieved by utilizing BPEL4WS, which can generate another new service[2]. In this way, all the engineering services can be wrapped as web service components. The JINI service from Sun Microsystems is also used to create dynamically adapting networked components, engineering applications, and services that scale from the device to the enterprise. Its unique qualities include (i) code mobility: extending the java programming model to the network, i.e., moving data and executables via a Java object over a network, (ii) leasing: enabling network self-healing and self-configuration, i.e., improving fault tolerance, (iii) integration: allowing fast, easy incorporation of legacy, current, and future network components, etc. However, in contrast to web services, there is no standard and open architecture-based module for service federations in the JINI service, just as web services cannot be used for user-oriented interfaces. To minimize disadvantages and to maximize advantages of both services, web services are used for back-end communications, whereas JINI services for front-end communications. Context-aware devices, user-friendly displays, and engineering environments are described as JINI objects, which can be easily registered, discovered, and implemented in mobile and context-aware environments. Note that code mobility can be effectively used for HCI and analyzing the results of engineering services. Similarly, each engineering computing service is represented as a web service component. Moreover, BPEL4WS is used as a web service process template for service orchestration and choreography.
The proposed virtual engineering service framework is a context-aware infrastructure that exploits semantic web technologies to support explicit context representation, expressive querying, and flexible reasoning of engineering service-related contexts. Using Topic Maps, a semantic web standard, to define context ontologies provides a foundation for interoperable engineering service environments where computing entities can easily exchange and interpret contexts based on explicit engineering context representations. Other reasons for adopting Topic Map are in the flexibility for explicitly representing contexts, easy merging of multiple contexts based on the Public Subject Indicator (PSI), multiple viewing of contexts based on the scope representation, and having standard-based querying and reasoning language of contexts.

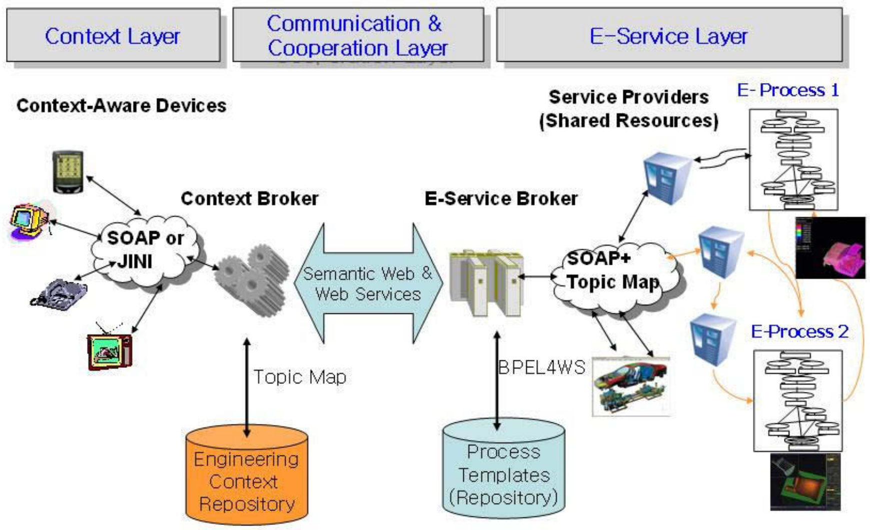


Fig. 1. Conceptual implementation of U-CAFÉ

The framework has been built on the three layers: 1) communication & cooperation layer, 2) E-service layer, and 3) context layer. The semantic web and web services-based communication & cooperation layer supports context-aware bi-directional communications between context-aware devices (i.e., service requestors) and engineering service providers. This layer that encapsulates the clients from multiple engineering web services provides an abstraction layer for available services. The E-service layer works as a service dispatching and aggregation broker. It supports dynamic engineering service composition and binding via process templates. Published web services communicate directly with their legacy applications by web service wrapping. The context layer maintains contexts from various resources such as devices, people, environment, etc. Further, the context broker facilitates reasoning and querying of contexts represented in Topic Maps. Based on these contexts, requestors or mobile devices can dynamically adapt to the most desirable situation to analyze requested engineering services. Thus, the three layered framework can support asynchronous long-running engineering service transactions, service choreography and orchestration, and context-aware adaptation to the environment and people.

For example, when a user invokes an engineering service via the public interface using a PDA or mobile device in a ubiquitous and context-aware environment, the request is sent to the E-service broker via the context broker that then performs searching and matching an appropriate engineering service template. At the same time, the contexts related to devices, location, and people are detected and stored in the context repository. When the process template is found, the broker searches and binds each activity in the process to an engineering service available. Then, it deploys the activated process template into the process engine for execution. When the requested engineering service is successfully executed, the result is sent to the requestor who can review and analyze it in either the PC or PDA according to the result of context reasoning and query.

3. ENGINEERING SERVICES ON U-CAFÉ

3.1 Engineering service federation

Engineering service integration requires more than the ability to conduct simple interactions by using standard protocols. The full potential of engineering web services as an integration platform will be achieved only when applications and engineering processes are able to integrate their complex interactions by using a standard process integration model. Thus, we propose the concept of BPEL-based engineering process templates to realize the integration of engineering applications and processes[10]. A BPEL-based engineering process template is used to orchestrate and choreograph existing engineering services to provide a new engineering service. In particular, Business Process Execution Language for Web Services (BPEL4WS) allows modeling business processes (workflows) for web services[2]. BPEL4WS represents the combination of two previously competing standards: XML business process language (XLANG) from Microsoft, and Web Services Flow Language (WSFL) from IBM. As shown in Fig. 2, when a new engineering web service instance is created, it is advertised to the UDDI by registering the WSDL description.

Well-defined process templates are stored in the process repository, from which they are fed into the E-Service broker. The WSDL descriptions of the services are queried from the repository. Finally, the broker composes the executable process and feeds it into the BPEL-based process execution engine. Readers are referred to see the Reference 10 for details.

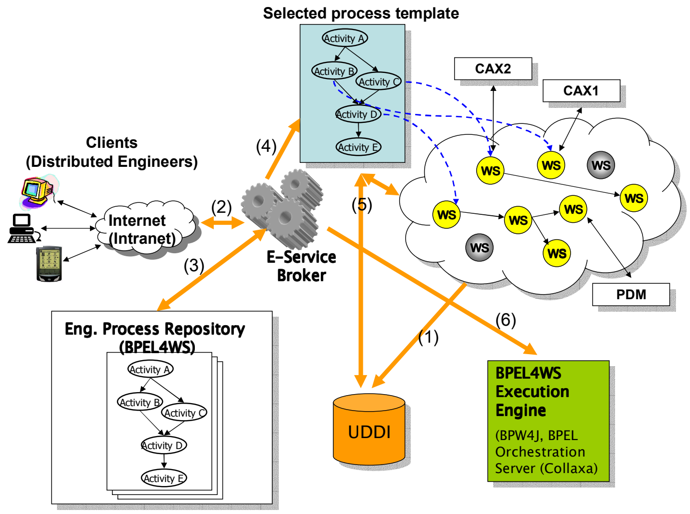


Fig. 2. Process templates and process broker

3.2 Engineering service-related context management

By representing engineering service-related contexts as easily interpreted semantic ontologies, the context-aware engineering service framework enables engineering applications to retrieve contexts using declarative queries and supports the inference of higher-level contexts from the basic contexts. Because context-aware applications must adapt to dynamically changing situations and activities, they need a detailed model of users' activities and surroundings that lets them share users' perceptions of the real world[13]. For that reason, the context-related infrastructure consists of several context-aware collaborating components as shown in Fig. 3. The context acquisition and maintenance module discovers and gathers contexts from mobile devices such as PDA and cellular phone, people, location, and engineering services, RFIDs, barcode readers, and Bluetooth-enabled sensors. Then, it asserts gathered contexts into the context knowledge base. The context knowledge base also stores context ontologies given by users or gathered from context devices. The context knowledge base links the context ontology and contexts in a single semantic model and provides interfaces for the context query engine and context reasoning engine to manipulate correlated contexts. The context query engine provides an abstract interface for applications to extract desired contexts from the knowledge base.

Topic Maps are a new ISO standard for describing knowledge structures and associations them with information resources. Dubbed "the GPS of the information universe", Topic Maps are also destined to provide powerful new ways of navigating large and interconnected corpora. Topic maps consist of three basic concepts: Topics, Associations, and Occurrences. Topics are the most fundamental concept in topic maps, which defines a subject. A topic may be lined to one or more information resources that are deemed to be relevant to the topic in some way. Such resources are called occurrences of the topic. Associations play a role in describing relationships between topics. Topic maps has a standard query language called Topic Map Query Language (TMQL)[15].

One of the most important modules in U-CAFÉ is the context acquisition & maintenance module as shown in Fig. 4 since contexts from various sources must be appropriately gathered and maintained for providing ubiquitous and human-centered engineering services. This module maintains the contexts in three different levels of details: 1) proxy generation for each context, 2) cluster generation based on the hierarchical representation of the contexts among person, device, and E-service proxies, and 3) Topic Map-based integrated context map generation. In order for each context to be easily registered, queried, and discovered over the U-CAFÉ, it is wrapped as a JINI proxy with a context wrapper. As a context proxy, it is registered to the JINI service network, and thus it is possible for the service requestor to find the registered proxy with the help of the JINI lookup service. The Topic Map-based context wrapper attached to each proxy plays another important role in matching semantics and merging new contexts with existing contexts in the context knowledge base. There are several context wrappers according to the type of contexts such as person, device, location, and E-service wrappers.

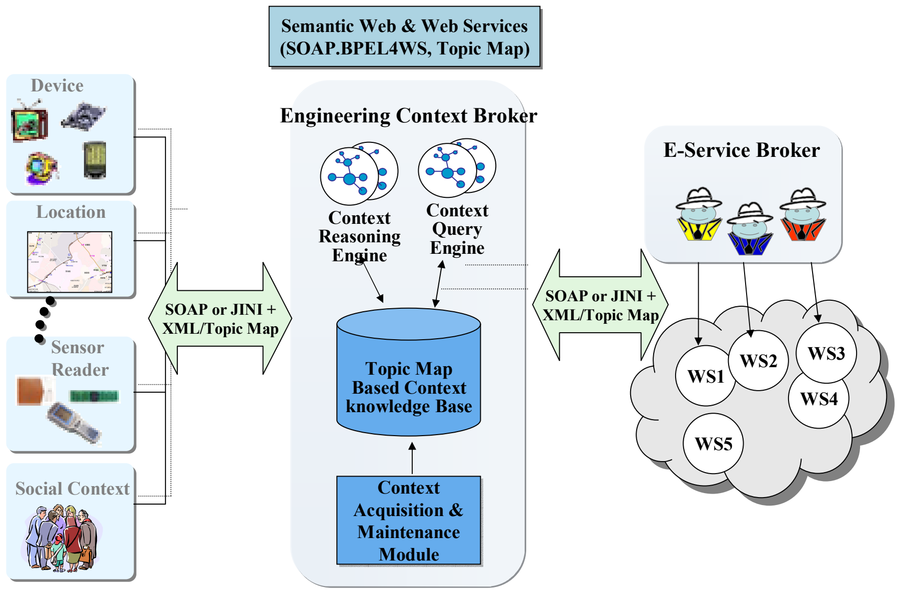


Fig. 3. Context awareness management module

Moreover, the hierarchical representation of contexts among persons, devices, and E-services is also maintained. It consists of four layers. The bottom layer includes a range of mobile and fixed devices; neither hardware architecture nor operating system must be homogeneous. The second layer contains device proxies, which every device has. The third layer is the user-proxy layer. Every user in the U-CAFÉ service network has a personal user proxy. This layer can store applications and a user's state. The fourth layer is the E-service layer, where the architecture provides shared engineering applications, utilities, and servers. All communications between layers are done via JINI lookup services and surrogate services. A surrogate is a facilitator that enables for a device that cannot run over the JINI service network to communicate with registered proxies over the JINI service network. Some sensors or devices such as PDAs and cellular phones cannot support the JINI service due to hardware and software limitations. To overcome these kinds of limitations, we implemented JINI surrogate services for such devices as shown in Fig. 4. Thus, it is possible to consistently maintain not only JINI interoperable contexts but also JINI non-interoperable contexts using surrogates regardless of device limitations. Note that the main reason in representing contexts in three different levels of details is to classify various kinds of dynamic contexts into clusters and integrated semantics for providing multiple views and adapting to a dynamically changing environment. It is also necessary to reason from and query about contexts for finding most desirable contexts and searching for conflict generating contexts. Thus, Topic Map-based integrated context map is stored in the knowledge base, from which we can query and reason about contexts using TMQL.

The Topic Map-based engineering service contexts used in this research are shown in Fig. 5. Fig. 5a shows the context hierarchy where the top node represents U-Context topic with its child nodes such as Device, Person, Geo-M, Activity, Condition, E-Service, and Location. Fig. 5b shows a Topic Map navigation about Mr. A topic whose topic type is Person. Around the Mr. A topic, there are many related topics such as requested engineering services, the service status, and location such that we can provide the right engineering service to the right person at the right time with the right analysis device. The following querying result based on TMQL shows how contexts can be effectively used for providing the right service in the given situation. The following query implies that "find Mr. A's requested engineering services and their status that can be served by the working computers located in CisLab".

Query

```
select $R-Service, $Status from
  e-requestedBy($R-Service:Requested-E-Service, Mr.A:Person),
  e-statusOf($R-Service:Requested-E-Service, $Status:E-Service-Status),
  e-instanceOf($R-Service:Requested-E-Service, $P-Service:Provided-E-Service),
  servedBy($P-Service:Provided-E-Service, $Computer:Device),
  locatedIn($Computer:Device, CisLab:Location),
  conditionOf($Computer:Device, GoodCondition:Condition) ?
```

Query Result

$R-Service	$Status
Mr. A's R-Service1	E-Service-Finished
Mr. A's R-Service2	E-Service-Processing
Mr. A's R-Service3	E-Service-Finished

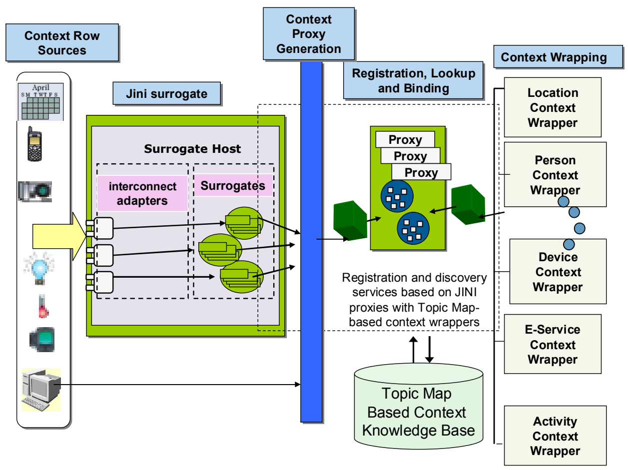
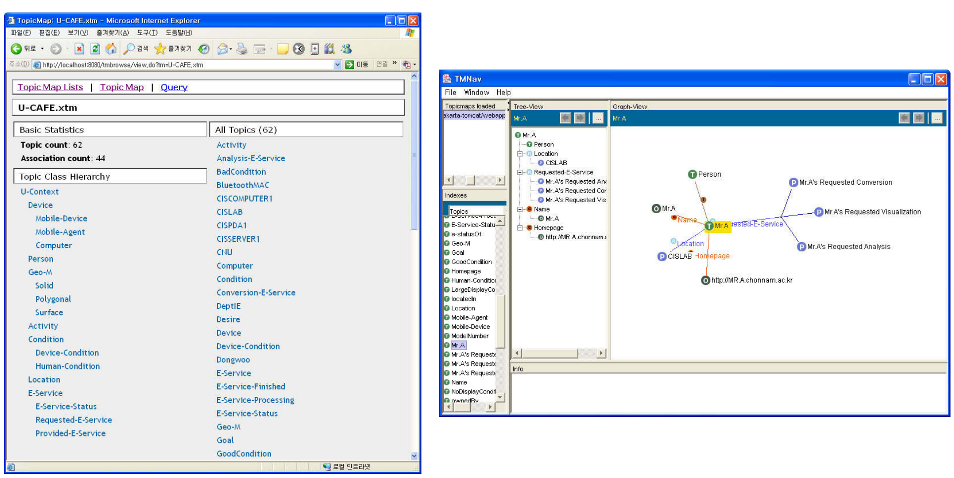


Fig. 4. Context acquisition and proxy generation



(a) Context hierarchy (b) Topic Map navigation around Mr.A Topic

Fig. 5. Topic Map-based context representation and context navigation

4. SYSTEM IMPLEMENTATION

This section explains how the proposed framework can be integrated and applied to virtual engineering services in a ubiquitous and context-aware environment. To illustrate the benefits of U-CAFÉ, we present the following product development service scenario. We assume that a user, Mr.A, requested an asynchronous product development service via mobile device at CisLab. During the analysis of the result, the U-CAFÉ service network advises him to investigate the analysis result with a large display or VR device instead of his PDA due to hardware and software limitations. During the analysis of the result, unfortunately, he finds a serious problem and, thus, he wishes to collaborate with Mr. B for resolving the problem. However, Mr. B can only access his PDA with which he cannot visualize a large CAD analysis model. Thus, they cannot collaborate with each other based on the previous engineering collaboration environment. In this scenario, Fig. 6 and Fig. 7 show a possible solution for this scenario. They show how context awareness is used to relate users with devices and to manage their relations as proxies, and show how context awareness is utilized in the ubiquitous engineering collaboration. In particular, Fig. 11 shows the screen capture of the implemented system where they can do ubiquitous engineering collaboration regardless of inhomogeneous hardware and software platforms and system limitations. In this scenario, a context aware-based application level of details is applied to suggest a possible solution. Note that this kind of collaboration is impossible in the existing concept of engineering collaboration.

Fig. 6. User awareness in the U-CAFÉ service network

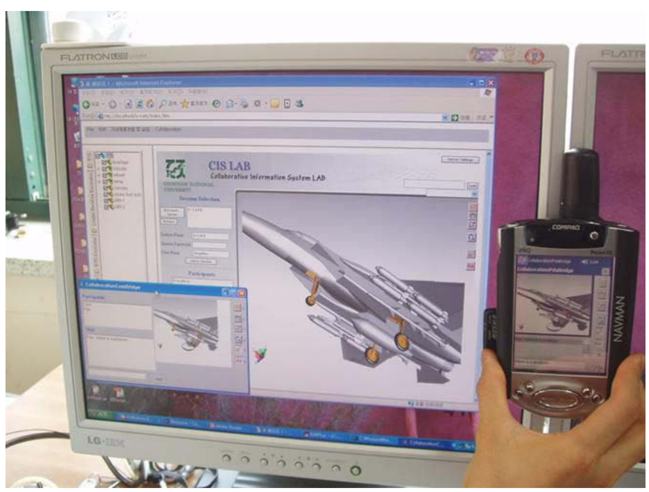


Fig. 7. Ubiquitous engineering service collaboration based on engineering contexts

4. CONCLUSION

A ubiquitous and context-aware computing framework for supporting virtual engineering services has been proposed. In this framework, a service-oriented approach to engineering services and service federations can be readily evaluated against engineering service customer, provider, and delivery objectives by considering contexts. By managing and reasoning engineering service contexts, the proposed approach can greatly reduce difficulty and cost in building user-oriented and context-aware knowledge that can provide relevant engineering services and information.

Several areas of research related to engineering service remain. Research is still needed to develop an understanding of various process templates for typical organizational structures and service acquisition strategies. There is a need to develop a formal representation of engineering service related contexts and test it for supporting more realistic engineering services.

5. ACKNOWLEDGEMENT

This work was supported by grant No. (R01-2003-000-10171-0) from the Basic Research Program of the Korea Science & Engineering Foundation.

6. REFERENCES

[1] Abrahamson, S., Wallace, D., Senin, N. and Sferro, P., Integrated design in a service marketplace, *Computer-Aided Design*, Vol. 32, No. 2, 2000, pp 97-107.

[2] BPEL4WS, http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/, 2003.

[3] Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S., EasyLiving: technologies for intelligent environments, *Proc. 2nd Int'l Symp. Handled and Ubiquitous Computing(HUC2000)*, LNCS 1927, Springer-Verlag, 2000, pp.12-29.

[4] Chen, H., Finin, T., Joshi, A., Kagal, L., Perich, F., and Chakraborty, D., Intelligent agents meet the semantic web in smart spaces, *IEEE Internet Computing*, Vol. 8, No. 6, 2004, pp 69-79.

[5] Jini, http://www.jini.org/, 2004.

[6] Kim, H., Kim, H.-S., Lee, J.-H., Jung, J.-M., Lee, J.Y. and Do, N.C, Framework for sharing product information across enterprises, *10th ISPE International Conf. on Current Engineering: Research and Applications*, Portugal, 2003, pp 497-504.

[7] Kindberg, T., *et al*., People, places, things: web presence for the real world, *Mobile Networks and Applications*, Kluwer Academic Publishers, 2002, pp. 365-376.

[8] Lee, J.Y., Kim, H. and Kim, K., A web-enabled approach to feature-based modeling in a distributed and collaborative design environment, *Concurrent Engineering: Research and Applications*, Vol. 9, No. 1, 2001, pp 74-87.

[9] Lee, J.Y., Shape representation and interoperability for virtual prototyping in a distributed design environment, *International Journal of Advanced Manufacturing Technology*, Vol. 17, No. 6, 2001, 425-434.

[10] Lee, J.Y., Lee, S., Kim, H., and Kim, H., A service-oriented approach to engineering web services, *Proc. 11th ISPE International Conf. on Concurrent Engineering: Research and Applications*, Beijing, China, 2004, pp.81-88.

[11] OWL, http://www.w3.org/TR/owl-guide/, 2004.

[12] Sobolewski, M., Federated P2P services in CE environment, *Advances in Concurrent Engineering, A.A. Balkema Publishers*, 2002, pp 13-22.

[13] Wang, X., Dong, J.S. and Chin, C.Y., Semantic space: an infrastructure for smart spaces. *IEEE Pervasive Computing*, Vol. 3, No. 3, 2004, pp. 32-39.

[14] Web Services, http://www.w3.org/2002/ws/, 2002.

[15] XML Topic Maps 1.0, http://www.topicmaps.org/xtm/1.0/, 2001.