TOWARDS VERIFYING SERVICE INTEROPERABILITY REQUIREMENTS FOR PERVASIVE COMPUTING ENVIRONMENTS

Yasir Malik and Bessam Abdulrazak
DOMUS Laboratory, University of Sherbrooke, Sherbrooke, Canada

Keywords: Verification, Interoperability, Design by Contract, Service Modeling, Smart Environment.

Abstract: Service interoperability verification is key to enable service continuity among pervasive computing environments. Pervasive computing is shifting the computing paradigm toward everywhere computing where number of heterogeneous and autonomous systems offer rich sets of services to assist inhabitants in their daily living. To support continuity of services these systems must provide seamless and flawless interoperability among devices and services. In this paper, We present our ongoing project on verifying the service interoperability in pervasive computing environments. We highlight the potential use of formal methods to verify the interoperability requirements and propose using design by contract technique to model and verify the semantic and pragmatic service interoperability requirements. We also present our verification requirement analysis and the benefits of using this technique with rich support from languages for implementation.

1 INTRODUCTION

Service interoperability verification is key to enable service continuity among pervasive computing environments. Pervasive computing is shifting the computing paradigm toward everywhere computing, where number of heterogeneous and autonomous systems offer rich set of services to assist its habitants in their daily living. Mark Weiser envisioned pervasive computing where computing devices will be integrated smartly in their environments to help users with their everyday living (Weiser, 1999). We foresee a future, where humans will be directly or indirectly interacting with number of smart devices and services embedded in their physical environments without knowing about them. To support continuity of such services these systems must provide seamless interoperability among devices and services. Interoperability is the capability of heterogeneous and autonomous systems to interact with each other for efficient service execution (Pokraev et al., 2006b), it can also defined as transparent & compatible execution between equipment from different vendors. The formal definition of interoperability given in IEEE glossaries is the ability of two or more networks, systems, devices, applications or components to exchange information between them and use the exchanged information. Pervasive computing environments are heterogeneous with many autonomous systems running in it. To successfully deploy pervasive computing environments, it is very important that these environments support interoperability of services and components deployed in the environments to achieve common goal(Perumal et al., 2008a). There are three types of interoperability problems that can occur in pervasive computing environments:

- **Syntactic Interoperability:** *Syntactic interoperability involves a common data format and common protocol to structure any data so that the manner of processing the information will be interpretable from the structure*[1].
- **Semantic Interoperability:** *Semantic interoperability requires that any two systems will derive the same inferences from the same information*[1].
- **Pragmatic Interoperability:** *Pragmatic interoperability is reached when the interoperating systems are aware of the methods and procedures that each system is employing. In other words, the use of the data or the context of its application is understood by the participating systems*[1].

Researchers have presented many approaches to support interoperability in pervasive computing environ-

[1]http://en.wikipedia.org/wiki/Semantic_interoperability

Malik Y. and Abdulrazak B..
TOWARDS VERIFYING SERVICE INTEROPERABILITY REQUIREMENTS FOR PERVASIVE COMPUTING ENVIRONMENTS.
DOI: 10.5220/0003907202200223
In *Proceedings of the 2nd International Conference on Pervasive Embedded Computing and Communication Systems* (PECCS-2012), pages 220-223
ISBN: 978-989-8565-00-6

ments (Perumal et al., 2008b; Maestre and Camacho, 2009), however there are very few works done on verifying service interoperability requirements for these environments. We address in this paper the problem of verifying the interoperability requirements of services in pervasive environments. We highlighted and analyzed potential use of formal methods for specifying and verifying services at run-time. We propose to use Design by Contact technique to model and verify the semantic and pragmatic service interoperability requirements at runtime. Our analysis shows promising benefits of using this technique, also there is rich support from languages for implementing this technique.

The rest of article is organized as follows. In the next section we present state of the art addressing interoperability and its verification process. In section 3 we present the potential benefits of using formal method and analysis of design by contract for semantic and pragmatic interoperability verification. Finally, in section 4, we conclude the paper and discuss future work.

2 RELATED WORK

This section presents some presentative work related to interoperability requirements and solutions in pervasive computing environments. Sullivan and colleagues highlighted the interoperability requirements and suggested the potential use of semantic techniques like ontologies in solving the key challenges of service interoperability (O'Sullivan and Lewis, 2003). Roussaki and colleagues in (Roussaki et al., 2008) suggested the use of ontologies and presented an ontology based service model for composability of various available services in pervasive computing environments. Bromberg and colleagues presented an event-based parsing technique to provide full service discovery interoperability to any existing middleware, their system act as bridge between topological networks and discovery technologies to adapt itself to the environment and its host to offer interoperability anytime anywhere (Bromberg and Issarny, 2005). Later this approach was extended by Bottaro and colleagues (Bottaro et al., 2007), they proposed a software architecture based on OSGi framework to address protocol heterogeneity, interface fragmentation and device composition among other challenges in smart home environment. All the aforementioned approaches have presented solutions for supporting interoperability requirements in pervasive computing environments; however they have not provided any mechanism or approach for its verification. Pokraev and colleagues (Pokraev et al., 2006a; Pokraev et al., 2006b) addressed the issue of interoperability verification and identified assessment requirements for semantic and pragmatic interoperability. These requirements are as summarized below:

R1: A necessary condition for the semantic interoperability of two systems is the existence of a translation function that maps the entity types, properties and values of the subject domain model of the first system to the respective entity types, properties and values of the subject domain model of the second system.

R2: A necessary condition for pragmatic interoperability of a single interaction is that at least one result that satisfies the constraint of all contributing systems can be established.

R3: A necessary conditions for pragmatic interoperability of a service is that *R2* is met for all of its interactions and they can occur in a causal order, allowed by all participating systems.

Based on the above requirements, they proposed a method to verify composite systems. The limitation of their approach is the lack of mapping mechanism for their service model. Baldoni and colleagues in (Baldoni et al., 2006) suggested formalizing the interaction protocols using finite state automata to define and verify properties of a service for the given protocol. There approach is a static analysis of the system which can improve the confidence in the system, however it require prior knowledge of protocols description under test. A similar approach is presented by Wan and colleagues, authors presented a verification algorithm and propose to formalize the properties of service adapted to Pantagruel i.e.(language that describes and manages services), which will allow programs to be verified prior to their execution, however this approach is limited when used to verify composite systems on the fly (Wan et al., 2010).

3 VERIFICATION METHOD

Formal methods have been use to specify system and provide methods such as model checking, theorem proving for verifying and validating the functional and behavioral correctness of the systems at early design stage. Recently the research in formal methods is moved from off-line verification to on demand verification (runtime verification) which combines formal specification with programming languages to support the runtime verification of software systems. We analyzed the potential benefits of using formal methods

to verify the interoperability requirements in pervasive computing environment. In particular, we studied and identified the potential use of **D**esign by **C**ontract (DbC) technique in systems based on service oriented architectures.

Design by Contract is the object oriented design techniques introduces by Meyer (Meyer, 1992), based on the Hoare's Correctness Triplets of $\{P\}$ A $\{Q\}$. This expression can be interpreted as an execution of process A is a state where P holds and termination of process A is a state where Q holds. Here P and Q are the logical pre and post condition assertions for the process A respectively. These pre and post conditions are formally known as contracts that software designers can use to formally define precise and verifiable specification for software components. These contacts can be viewed as rights and obligations of the class and its client. DbC can be used to support the semantic interoperability. The fundamental idea of DbC is to attach assertions or "contracts" with each component of the application, which allow the runtime verification of the properties that hold. There are three different kinds of assertions that could be defined for any service, routine or application component:

1. **Pre-condition:** As the term is self expressive, preconditions are used to specify the assertion that must hold before the execution of the program, or component. In this way the programs/services can be verified before their executions. For instance it is possible to verify the arguments or inputs required for the service.
2. **Post-condition:** Post-conditions are assertions that must hold after the execution of the program. In this way services are verified after their executions against respective contracts.
3. **Invariants:** Invariants are the assertions in the contract that must hold anytime during the execution of a program.

3.1 Requirements Analysis

In this section we analyze Design by Contract technique to verify the semantic and pragmatic interoperability requirements. Contracts can be used to specify the application requirement that must hold to interoperate with other systems. A service contract can specify semantics, behavioral, functional and non functional requirements associated with the respective service. In this way, a service provider makes no assumptions about a service consumer, other than those specified in the service contract. Service contract can act as an interface through which service consumers communicate with the other services for service composition. These service contracts can be specified independently of underline technologies to support interoperability. The service model for such implementation is shown in Figure 1. The use of DbC technique will help to support and verify semantic interoperability requirements in pervasive computing environments. In the next section we present the analysis of DbC technique in satisfying the semantic and pragmatic interoperability requirements identified in (Pokraev et al., 2006b). The first requirement *R1*

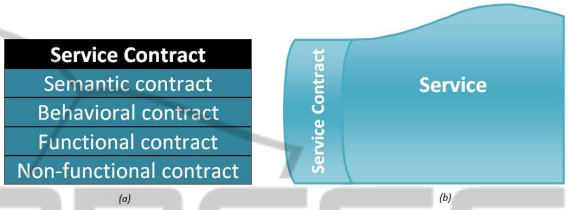

Figure 1: Service contract model.

for the semantic interoperability is that the system must have a translation function that maps the entity types, properties and subject domain model of the systems being integrated. This requires the complete knowledge of the system/service and its relationships with other components in the environment. This requirement can be satisfied by associating a contract with each component of the system. Thus when systems are interacting, they can be verified w.r.t to their contracts. This could be the pre or post-condition that must hold for any communication. To enrich the contract with the domain knowledge and relations with other components, ontologies can be used. Ontologies are the description of some shared concepts and their relationships for the given domain and they support semantics interoperability based on the domain knowledge. Requirement *R2* and *R3* address the problem of pragmatic interoperability. As discussed above, pragmatic interoperability can be reached when the inter-operating systems are aware of each other methods and procedures also that necessary conditions are met for at least one result that can satisfy constraint. As a result the context of its application is understood by the participating systems. This requirement can be satisfied by defining the invariants of the contract and can be verified any anytime during the execution to know the current state of the system. DbC can also be applied to address the service composition and verification in pervasive computing environment. Authors in (Chen and Huang, 2009) has presented an approach to model a service behavior represented by concurrent regular expression and then translate it to labeled transition system to form a finite state process notation trace

to verify the service composition. This approach is not automatic and does not support service interoperability if the composition is between two different systems. DbC approach can be applied to support such interoperability and verification of service composition. The idea is similar as stated above, every service has its contract which specifics its behavior, functional and semantics requirements.

4 CONCLUSIONS

In this paper, we presented the potential benefits of using formal method for verifying the interoperability requirements in pervasive computing environments. In particular, we highlighted the potential benefits of Design by Contact technique to verify the service interoperability. We analyzed the DbC to satisfy the requirements of semantic and pragmatic interoperability. We have also presented a service contract model that can be used to specify the contracts of the service in OSGI framework. The model include behavioral, semantical, functional and non functional contracts associated with respective service. The potential benefit of DbC can be applied at runtime for service verification and supporting interoperability at the same time. In upcoming publications, we will present the implementation of service contacts developed in OSGi framework to show the results of our proposed method.

REFERENCES

Baldoni, M., Baroglio, C., Martelli, A., and Patti, V. (2006). A priori conformance verification for guaranteeing interoperability in open environments. In *In Proc. of ICSOC 2006, volume 4294 of LNCS*, pages 339–351. Springer.

Bottaro, A., Gerodolle, A., and Lalanda, P. (2007). Pervasive service composition in the home network. In *Proceedings of the 21st International Conference on Advanced Networking and Applications*, AINA '07, pages 596–603, Washington, DC, USA. IEEE Computer Society.

Bromberg, Y.-D. and Issarny, V. (2005). Indiss: interoperable discovery system for networked services. In *Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*, Middleware '05, pages 164–183, New York, NY, USA. Springer-Verlag New York, Inc.

Chen, J. and Huang, L. (2009). Formal verification of service composition in pervasive computing environments. In *Proceedings of the First Asia-Pacific Symposium on Internetware*, Internetware '09, pages 19:1–19:5, New York, NY, USA. ACM.

Maestre, J. M. and Camacho, E. F. (2009). Smart home interoperability: the domoesi project approach. *International Journal of Smart Home*, 3:31–44.

Meyer, B. (1992). Applying "design by contract". *Computer*, 25:40–51.

O'Sullivan, D. and Lewis, D. (2003). Semantically driven service interoperability for pervasive computing. In *Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access*, MobiDe '03, pages 17–24, New York, NY, USA. ACM.

Perumal, T., Ramli, A. R., Leong, C. Y., Mansor, S., and Samsudin, K. (2008a). Interoperability among heterogeneous systems in smart home environment. In *Proceedings of the 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*, SITIS '08, pages 177–186, Washington, DC, USA. IEEE Computer Society.

Perumal, T., Ramli, A. R., Leong, C. Y., Mansor, S., and Samsudin, K. (2008b). Interoperability for smart home environment using web services. *International Journal of Smart Home*, 2:1–16.

Pokraev, S., Quartel, D. A. C., Steen, M. W. A., and Reichert, M. (2006a). A method for formal verification of service interoperability. In *Proceedings of 2006 IEEE International Conference on Web Services 18-22 September 2006, Chicago, Illinois, USA*, pages 895–900. IEEE Computer Society.

Pokraev, S., Quartel, D. A. C., Steen, M. W. A., and Reichert, M. (2006b). Requirements and method for assessment of service interoperability. In *Proceedings of the 2006 4th International Conference on Service Oriented Computing, ICSOC '06, December 4-7, 2006, Chicago, USA*, volume 4294 of *Lecture Notes in Computer Science*, pages 1–14. Springer.

Roussaki, I., Papaioannou, I., Tsesmetzis, D., Kantorovitch, J., Kalaoja, J., and Poortinga, R. (2008). Ontology based service modelling for composability in smart home environments. In Mhlhuser, M., Ferscha, A., and Aitenbichler, E., editors, *Constructing Ambient Intelligence*, volume 11 of *Communications in Computer and Information Science*, pages 411–420. Springer Berlin Heidelberg.

Wan, H., Drey, Z., You, Z., and Liu, L. (2010). Formal Modeling and Verification of Services Managements for Pervasive Computing Environment. In *Proceedings of The 7th International Conference on Service Systems and Service Management*, Tokyo Japan.

Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11.