Collective Service Intelligence Management in Mobiquitous Systems

Evgeniya Ishkina
Information Systems department
Astrakhan State University
Astrakhan, Russia
e-mail: ishkina@aspu.ru

***Abstract*—This paper describes a Self-adaptive Middleware for Mobiquitous information system and its underlying model based upon a multidimensional service representation and management. The service dimensions refer to the "4W": 1) Who?, i.e., user profile (implicit knowledge deduced from user interaction history and explicit knowledge in form of preferences); 2) Where and 3) When?, i.e., "external" (physical) interaction context (location, time, device, etc.); 4) What?, i.e., "internal" interaction context (user ad-hoc task: goal(s), expectations and optional requirements). An original approach for dynamic adaptive service generation (on-the-fly composition) based on collective service intelligence captured in what (we call "collective services memory") is proposed. Currently a prototype of the middleware is being implemented for touristic cultural paths in the city of Astrakhan (Russia).**

Keywords - service computing; mobiquitous services; middleware; adaptive services; service composition; service mining; context-awareness; NFC standard.

I. Introduction

We are now entering a new technological era where physical spaces are becoming "smart" and objects – "tagged" thus providing all kind of (*mobiquitous*) services for mobile smart phone holders. *Mobiquity* is a recent word bearing the strategic convergence of *mobi*lity (mobile phones becoming smart) and ubi*quity* (of Internet becoming local, 2.0 and broadband) [1].

The story of computer science can be seen as a story of functional system/service layers being introduced to ease either application development (successively operating system, database server, application server, mobile server, EDGE Server) or user-friendly convenience. Our proposal consists in creating of a new functional layer on top of the EDGE server that could increase interaction efficiency of users in mobiquitous information systems by proposing them complex services fitting their goals: the interaction system realized in a middleware.

By services we understand web services and NFC (Near Field Communication) mobile services. Services could be different regarding:

- Standards: SOAP (Simple Object Access Protocol) versus REST (Representational State Transfer) standards for web services implementation;
- Interactions: web services with one simple invocation method versus NFC services with its touching paradigm and three operating modes of NFC standard: reader/writer, peer-to-peer, and card emulation mode).

SOAP and REST are two main approaches for building web services. SOAP services are operation-oriented, and RESTful services are resource-oriented. Semantic web services, in addition to simple web services, provide machine-readable semantic data. Existing semantic web service ontologies, such as OWL-S and WSMO, commonly consider only SOAP web services. However, there are proposals for extending them in order to support RESTful web services which are very widespread currently [2]. For NFC services there is not common standard yet.

These heterogeneous services should be consolidated into one integrated warehouse of complex mobiquitous services.

Adaptive information systems are becoming increasingly important especially in the field of mobiquitous information systems. In [3], Sousa *et al.* show that today users are surrounded by technology that is heterogeneous (wide variety of computing platforms, interfaces, networks and services), pervasive (wireless and wired connectivity that pervades most of our working and living environments) and variable (users can move from resource-rich environments, such as workstations, to resource-poor environments, such as a PDA in a park). Mobiquitous systems should be able to adapt to users mobility and system access ubiquity in order to reduce users overhead; they should abstract users from details of access to heterogeneous, pervasive and variable services for maintaining high-level user activities. Users no longer want to get pieces of data, information or knowledge, but they want to get the complete services [4].

Today we can make objects *smart* by tagging them (QR codes, NFC tags), but they still remain reactive only. A mobiquitous system is not adaptive by default; it provides the user with the first level of "intelligence" – the ability to communicate with objects using a mobile phone at *any time*, *anywhere* and on *any device* for obtaining different services. A new functional layer is proposed in our research to make

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-124-3

the mobiquitous system proactive, context (situation) aware, and, thus, to provide users with the second level of intelligence – the ability to obtain an *appropriate customized service*, adapted to the current user *situation* in a transparent manner.

This new functional layer is being implemented within a middleware platform which will hide details of access to heterogeneous, pervasive and variable services from users. Since interaction context is very important in mobiquitous systems, the key idea of this middleware is to manage multiple context-aware strategies (services) for fitting the same user goal, captured by the services usage analysis and then, able to be applied in an appropriate situation. These services can also be recombined in order to construct new useful services with the possibility to evaluate their quality and to infer their functional and nonfunctional characteristics. Each situation of users interactions with the mobiquitous system is unique; therefore, there should be a specific service generated especially for this situation from services fragments and knowledge about their usage stored in the collective services memory.

The remainder of this paper is organized as follows. In Section II, we describe some related research directions; then in Section III, we provide a motivating example of an illustrative scenario. In Section IV, we examine the concept of mobiquitous systems intelligence. In Section V, we present the architecture of the self-adaptive service-oriented middleware and the approach for dynamic service composition. And, finally, in Section VI, we give some research directions to enhance and validate our proposal.

II. RELATED WORK

A. Service composition and service mining

A service is an autonomous IT-asset which can be reused by an arbitrary number of consumers in contexts often unknown at design time. As a matter of fact, services must be designed to be as independent as possible from the context in which they will be utilized [5].

Service composition is an aggregation of multiple services into a single composite service providing more sophisticated functionality and creating add-on values. A user could be provided by a constructor for interactively building his own composite service but a more interesting problem is to automatically recognize user's situation for providing him an appropriate composite service.

In [6], Zheng defines two approaches for service composition. It can be:

- top-down – composition of existing web services driven by specific search criteria, or
- bottom-up – discovery of interesting and useful compositions of existing web services with no such criteria.

And the result of service composition can be:

- one-shot – it realizes a particular request of a particular end-user, or
- reusable – it realizes a generic request of a typical end-user.

In the proposed middleware the bottom-up approach consists in service mining correlated with services dependencies mining, and the top-down approach consists in analyzing goals conjunction, matching corresponding services and their composition. Final services generated by the middleware are always seen as one-shot, but the composition history is saved for further producing a reusable complex service from a set of correlated one-shot compositions.

In [7], Sousa *et al.* present two approaches for services discovery for composition, which can be:

- context-aware – given the context parameters, the suitable services are selected from the service repository, or
- goal-driven – the user makes a request and the system tries to find the most suitable service, which agrees with the request description.

The proposed middleware adopts both approaches: first, the services are selected by their goals annotations, and then the engine looks for the most suitable service composition taking into account context parameters.

Our aim is to deal with composability of different kind of heterogeneous services (mobile NFC services, SOAP/RESTful web services, semantic web services, etc.) through a middleware by integrating them at structural and semantical levels and by mining collective service intelligence.

B. Context-, situation- and task-awareness

Context awareness is referred to the capability of an application or a system to be aware of its physical environment and situation in order to be able to act and answer in a proactive and intelligent way [8].

In the field of services, context was often seen only as user location attached to popular location-based services. In our proposal we consider a larger context concept which is *situation* – combination of system-side, user-side and environment-side parameters.

In [3], the concept of task-awareness is presented. It means carrying out high-level users activities: planning a trip, buying a car, etc. In today's systems those activities and goals are implicit. In task-aware systems, users specify their tasks and goals, and it is the responsibility of the system to automatically map them into the capabilities available in the mobiquitous environment.

In our proposal the task corresponds, from one side, to a combination of goals, and from another side, to a set of service compositions. The system learns from usage how to make a reasoning on combined goals and learns about the strategies of fulfilling the goals, i.e., service scenarii.

C. Overview of existing approaches for intelligent service composition and delivery

Let consider three main groups of approaches for intelligent service composition and compare them with the proposal.

The first group is represented by approaches for context-aware service composition [7][9].

iCas [7] is a service-oriented architecture that uses an open ontological context model (SeCoM) to provide personal and contextual information and to support the composition of context-aware services on the fly. A prototype of the iCas platform is implemented. When starting services composition, user can add and remove services interactively. Possibilities for composition are returned to user based on the current context and user policies. Services are described using OWL-S.

MyCampus [9] is a semantic web environment aimed at enhancing everyday campus life. Users acquire or subscribe to a variety of task-specific agents that assist them in the context of different tasks. MyCampus supports the dynamic discovery and access of contextual information sources and the automated generation of plans by task-specific agents through the discovery of services that can be dynamically composed to satisfy one or more user-goals.

There also exist some approaches that do not allow service composition but they aim at delivering of context-aware services. For example, SOCAM [10] is a middleware architecture that supports the building and rapid prototyping of context-aware services. It uses the formal context model based on OWL to represent, manipulate and access context information.

Another group of approaches lie in the field of goal-driven service composition [11][12].

In [11], a goal-driven approach for service composition is presented. The authors propose a task-oriented semantic representation model of web services and based on this model goal-driven service composition is performed dynamically to achieve user's goal. The relevant concrete web services to complete the task are bound dynamically in the runtime.

In [12], a goal-based approach for dynamic service discovery and composition is described. The approach is based on a behavior model represented by goal modeling. Goals can be further decomposed into sub-goals, and tasks fulfill (sub-)goals. This approach is founded in a well-defined set of domain and task ontologies.

The third part of approaches concerns pattern-driven service composition. In [13], Tut *et al.* propose the use of patterns combined with the domain knowledge for facilitating the composition process of e-services. Patterns represent a proven way of doing something, "a three-part rule, which expresses a relation between a certain context, a problem and a solution". An example of instantiating of generic patterns into specific ones is presented.

Approaches for context-aware and goal-driven composition are often well separated, however there exist some mixed approaches, for example [9]. The majority of approaches for pattern-driven composition are not context-aware. Patterns represent result of an attempt to find context-free service sequences.

Our proposal consider pattern as a service: it has the same characteristics, it can be annotated by situation elements, it can make part of service composition. Patterns reveal relationships between services, and along with the context (situation) dependences represent collective service intelligence.

To our knowledge, there are not mixed approaches in the field of web services and interactive NFC services. Mobiquitous services which are being considered in this paper integrate smart objects with related web services and are consumed in a high interactive manner. However there are some research results in the field of context-aware NFC applications [14].

III. USE CASE SCENARIO

For better understanding of the remainder of this paper we will provide a use case scenario.

Consider two mobiquitous services in the city of Nice, in France (an NFC European city since May 2010). The first one is a complex NFC mobile service offering a guided tour on the "invisible historic cultural path" of Gogol, Russian writer, in Nice consisting of all places where he lived in the 1840's associated with his name and providing multimedia information attached to tags (QR Code and NFC tags) with the possibility to produce information on each point of the path and interface with social networks [15]. The second one is a web service of restaurant booking.

The complex goal of Ivan, a Russian tourist, is to make a complete tour and to have lunch in a restaurant (probably by making a break in his tour). His goal remains persistent during an interaction session and corresponds to initial interaction constraints; if Ivan would like to change his goals, a new session will start.

For the given goal we should consider two elements of the external environment: location and time. These are not constant during the interaction, their values are generated in real time.

Concerning the user himself, there are some extra parameters which could influence the system behavior. These are Ivan's meal preferences. If Ivan did not provide them to the system, the system itself can infer them based upon analysis of Ivan's interaction history. If there are no enough data for such analysis, the system can rely on the nationality of Ivan (if known) and use this information in conjunction with learned dependencies between nationalities and meal preferences. Let us consider that in this example meal preferences of Ivan are not known but the system knows that he is Russian and that most of Russian tourists prefer Russian restaurants in Nice.

Now we will describe some possible options. For example, one of the point of Gogol path concerns his preferred restaurant of French cooking. And the system has learned that whatever specific user preferences are, if the user goal is to have a lunch within this guided tour, they generally select the restaurant that Gogol preferred and not the one related with their own preferences.

So, Ivan starts his tour "Gogol in Nice"; he loads the special application or information on his mobile phone by touching a given touristic NFC tagged poster. The system proposes him to complete his goal with some specific goals, one of them is having a lunch – and he selects it. In the middle of his tour Ivan decides to have a lunch break. The system then generates the list of recommended restaurants. If the Gogol preferred restaurant is nearby to the Ivan's location, it will be the most recommended. If not – the system selection will be based upon Ivan's meal preferences primarily. If Ivan doesn't make a break but he is already in the Gogol preferred restaurant, the system provides him a proposition without his explicit demand.

IV. Mobiquitous Systems Intelligence

In mobiquitous systems, there exists an augmented need of tailored application delivery for reducing user overhead in mobiquitous environment that implies the necessity of discovering implicit collective intelligence of mobiquitous services and providing multidimensional usage view.

A. Collective service intelligence

In [16], O'Reilly *et al.* present collective intelligence concept evolution. They introduce the concept of *Squared Web* which play an intermediate role between Web 2.0 (social) and Web 3.0 (semantic). Web 2.0 offers to users the possibility to generate the content. The future Web 3.0 will allow to machines the possibility to understand data but it requires a lot of work for the current Web semantization. Web 2.0 focuses on collective human intelligence, while Squared Web focuses on collective intelligence of captors, tags, etc.

In mobiquitous systems we can consider collective intelligence of services. Services cannot be fully structured at design-time because there are still many unknown dependencies of users and usage contexts. The additional functional layer we are proposing, should manage the collective service intelligence and learn automatically about services usage in particular situations for further better interaction efficiency.

B. Multidimensional usage representation

Mobiquitous services usage can be represented in multidimensional space. The two typical (minimum) usage dimensions are user profile and location. Thus, users can be provided with customized services taking into account these parameters. But each service provider can make a reasoning on it, in its own way.

The idea to tag real objects in space is not new. It has been widely used in tracking objects in logistics, etc. Users interacts with a mobiquitous system via tagged and location-based objects; the system can give some recommendations to users based upon where he is located (and propose him restaurants, museums, shops, etc.). Here, we have two levels of context: the first one corresponds to the information which is directly obtained from sensors (spatial coordinates) and the second one corresponds to the inferred information (available services in users' vicinity).

The relatively new idea is to tag real objects in time scale. Saving and analyzing interaction tracks allow to get information about when, how and by whom these objects were used.

Rules applied for generating recommendations can be static (anyone who touches an NFC poster at the bus stop receives the same list of restaurants in the neighborhood), or customizable (based upon explicit user meal preferences or implicit preferences inferred by the system from the user interaction history).

In the area of mobiquitous systems we consider three usage dimensions (dimensions of context in its global meaning which we call *situation*):

- Interaction actor – user profile (explicit and implicit knowledge), we will call it user dimension;
- "External" (physical) interaction context (location, time, etc.), we will call it context dimension;
- "Internal" interaction context (user task: goal(s) and constraints), we will call it task dimension.

Depending on the application area (m-tourism, m-marketing, etc.) the importance of different dimensions varies, but there exists one common feature – mobiquitous systems are *task-driven*. The user looks for a complex service that fulfills his goal while hiding most part of implementation details; he does not look for isolated data, information, knowledge and services fragments. User tasks and goals are hierarchical and multiple service scenarii for achieving them are discovered at run-time. If there is none on-the-shelf solution in the collective services memory, the user goals are then decomposed in order to find matching services for sub-goals. The remaining two situation dimensions (user profile, physical environment) are used to select the most appropriate scenario for a given situation.

Thus, mobiquitous systems intelligence is based upon answering the following questions:

- When, where, how and by whom mobiquitous services should be used? – *Learning about usage situations*.
- How services can interact with one another? What service compositions could be useful to users and what is the typical situation profile for this? – *Performing service mining with corresponding situation parameters mining impacts appropriate services selection*.
- How to evaluate situations equivalence in a flexible manner depending upon application areas? – *Analyzing services used in these situations with some common parameters*.

V. Self-Adaptive Middleware for Mobiquitous Systems

A. Logical architecture

In Figure 1, the global centralized logical architecture of the self-adaptive middleware for mobiquitous systems is presented. Below we describe its major components.

1) Atomic services integration.

This layer enables integration of heterogeneous mobiquitous back-end services (SOAP/RESTful web services, semantic web services, NFC mobile services) by providing a unique service metamodel. It is a service access layer: only at this level details of heterogeneous services invocations are known.

2) Collective services memory management.

This layer enables discovery of new services compositions useful in particular situations, and enables management of all services – atomic and composite.

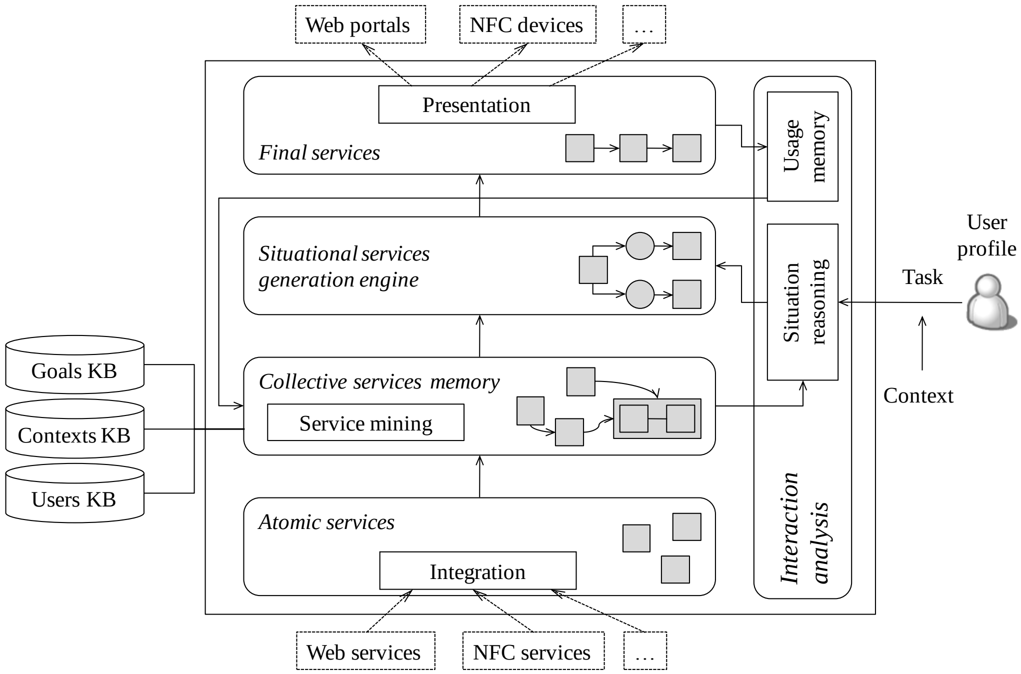

Figure 1. Logical architecture of self-adaptive middleware for mobiquitous systems

At this level the second layer of services metadata is defined, it represents situation parameters (task, user, context) learned from interaction history by applying data mining techniques.

"Collective" means that all services are stored in the memory along with explicit and imlicit relationships between them. Explicit relationship between two services is their joint use in a composite service. This use is conditional, related to the appropriate usage situation. Implicite relationship between two services consists in similarity of situations of their appropriate usage. Both explicit and implicit relationships between services are processed for pre-selecting services for final service generation.

Basic elements of collective memory are services (atomic or composite) which represent usage fragments. These fragments are used in the next layer for generating a unique service corresponding to user's situation.

All atomic services with their metadata generated at the integration level are stored within the collective memory. Then, it is enriched by compositions of existing services obtained by service mining algorithms.

Special algorithms allow evaluation of the usefulness of discovered service composition, i.e., the probability of its reuse, it also allows predicting of functional and non-functional characteristics of composite services.

Special algorithms are also used to mine key situation parameters which may influence the use of services together in a scenario. These parameters represent preconditions taken into account during adaptive end-user service generation.

Each of the three dimensions of usage situations (task, user, context) correspond to an ontology constructed during the system functioning for reflecting the set of situational parameters and their importance for a given application domain.

Service itself is independent from usage situations, it has a goal to achieve. Multidimensional service annotations based on task, user, and context ontologies enable evaluation of service relevance for the given situation.

Thus, collective services memory stores usage fragments – atomic services and useful composite services both annotated by ontologies corresponding to situation dimensions. Atomic services at this layer are considered as abstract elements described using a unique service metamodel without any details of their invocation.

3) Situational service generation.

Taking into account the situation description and services metadata stored in the collective memory, this layer generates a unique service corresponding to the given situation and representing a composition of services fragments from collective memory.

Some details of our approach for situational service generation are given in the next subsection.

4) Final services presentation

The purpose of this layer is to interact with atomic services integration layer for invocation of composition elements. This layer also provides the user with an appropriate interface.

5) Interaction analysis

Usage history component allows saving system usage logs and their preprocessing for further use of service mining algorithms.

Situation reasoning component hides from all other middleware components details of capturing 1st level situation data, i.e., information which is directly obtained from sensors. It constructs the 2nd level of situation representation, i.e., high-level information inferred from the 1st level situation data and related with task, user and context service dimensions operated by the middleware. It is further processed by the situational service generation engine.

B. *Approach for situational service generation*

Our approach for dynamic service generation is based upon the use of ontologies (Figure 2).

First of all, there is a domain ontology for sharing vocabulary between all middleware components. Then, there are three ontologies corresponding to *users*, *contexts*, and *goals* using the domain ontology. Goal ontology is totally domain-specific; for user and context ontologies some domain-independent elements can be defined.

Services in the collective memory are annotated by all the three dimensions ontologies. The situation is composed with task, user, and context descriptions. Task represents a set of goals indicated by the user. All useful parameters of user profile and physical context are preprocessed at the interaction analysis layer thus constructing the 2nd level of situation. User's and context's descriptions are based on the corresponded ontologies.

Situational service generation engine receives situation description and annotated services from collective memory.

Service discovery for composition is task-driven, it is based on task goals matching. In case of no services matched, task goals are decomposed for trying to find services corresponding to sub-goals.

Candidates are tested for correspondence to the given situation and for syntactic and semantic compatibility. The final service is then composed of the most appropriate services fragments.

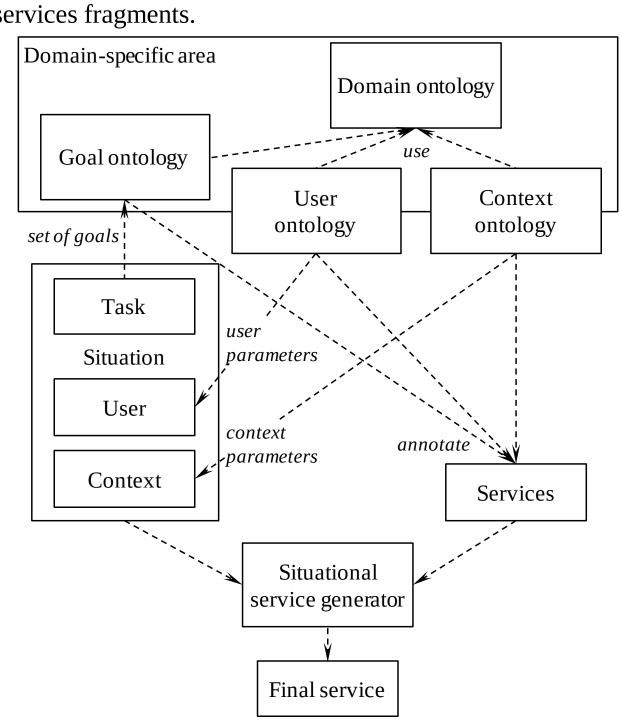

Figure 2. Situational service generation

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented the self-adaptive middleware for mobiquitous systems which manages the intelligence of mobiquitous information systems along with a description of its major components. We also described our approach for situational service generation, i.e., adaptive composition. The basic rationale is that, in distributed heterogeneous systems, it is necessary to collect knowledge about usage. This knowledge should not be the simple facts, but it should represent strategies of goals achivement in particular situations. Users no longer want to get isolated information; they look for complex services. Current systems are becoming task-aware: their goal is to match high-level user task in a transparent manner. Services could then be recombined for producing new ones with the possibility to evaluate the important characteristics of service composition, its usefullness in particular situations.

One of current tasks consists in analysis of the middleware architecture using available software engineering methods such as ATAM (Architecture Tradeoff Analysis Method).

The proposed approach is being implemented and validated for touristic cultural paths in the City of Astrakhan (Russia). Finally, our proposed architecture described here, will be formally described in companion research papers.

ACKNOWLEDGMENT

The author would like to express her appreciation to two Professors of Computer Science from the University of Nice Sophia Antipolis (France) and I3S Research laboratory, Serge Miranda for his valuable comments and remarks during this paper preparation, and Nhan Le Thanh, for his research guidance and impetus.

REFERENCES

[1] S. Miranda and E. Ishkina, "NFC mobiquitous information service prototyping: overview, lessons, state of the art, innovation and research directions", 3rd International Conference on Information Systems and Economic Intelligence (SIIE'2010), Sousse, Tunisia 18-20 February 2010.

[2] O.F. Ferreira Filho and M.A. Grigas Varella Ferreira, "Semantic Web Services: a Restful Approach", IADIS International Conference WWW/Internet, Rome, Italy, 2009.

[3] J.P. Sousa, V. Poladian, D. Garlan, B. Schmerl, and M. Shaw, "Task-based Adaptation for Ubiquitous Computing", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Special Issue on Engineering Autonomic Systems, 36(3), 2006, pp. 328-340.

[4] G. Zheng and A. Bouguettaya, "A Web Service Mining Framework", Proc. of the Int. Conf. on Web Services (ICWS'07), Salt Lake City, Utah, USA, 2007.

[5] J.J. Dubray, WS-PER: An abstract SOA framework http://www.wsper.org/ (2007)

[6] G. Zheng, "Web Service Mining", PhD Thesis, Virginia Tech, 2009.

[7] J.P. Sousa, E. Carrapatoso, B. Fonseca, M.G.C. Pimentel, and R.B. Neto, Composition of context aware mobile services using a semantic context model, International Journal on Advances in Software, 2:2-3, 2009, pp. 275-287.

[8] A.K. Dey and G.D. Abowd, "Towards a better understanding of context and context-awareness", Proc. of the Workshop on the What,

Who, Where, When and How of Context-Awareness, ACM Press, New York, 2000.

[9] M. Sheshagir, N. Sade, and F. Gandon, “Using Semantic Web Services for Context-Aware Mobile Applications”, in MobiSys 2004 Workshop on Context Awareness, Boston, 2004.

[10] T. Gu, H. Pung, and D. Zhang, “A service-oriented middleware for building context-aware services”, Journal of Network and Computer Applications, 28(1), 2005, pp. 1-18.

[11] K. Zhang, Q. Li, and Q. Sui, “A goal-driven approach of service composition for pervasive computing”, in Proc. of the 1st International Symposium on Pervasive Computing and Applications, 2006, pp. 593–598.

[12] L.B. da Silva Santos, L.F. Pires, M. van Sinderen, “A Goal-Based Framework for Dynamic Service Discovery and Composition”, International Workshop on Architectures, Concepts and Technologies for Service Oriented Computing, Porto, Portugal, July, 2008, pp. 67-78.

[13] M.T. Tut and D. Edmond, “The Use of Patterns in Service Composition”, Proceedings of the International Workshop on Web Services, E-Business, and the Semantic Web, Toronto, Canada, 2002, pp. 28-40.

[14] P.C. Garrido, G.M. Miraz, I.L. Ruiz, and M.A. Gómez-Nieto, “A Model for the Development of NFC Context- Awareness Applications on Internet of Things”, 2nd International Workshop on Near Field Communication, Monaco, 2010.

[15] E. Ishkina and S. Miranda, “NFC Mobiquitous Ecosystem for Information Services of the Future: Applications to M-tourism and M-learning”, Proc. of International scientific conference “Electronic culture. Information technologies of the future and modern e-Learning”, Astrakhan, 6-8 October 2009.

[16] T. O’Reilly and J. Battelle, “Web Squared: Web 2.0 Five Years On”, Web2.0 Summit Conference, O’Reilly Media, San Francisco, 2009, pp. 1-13.