User Centred Automated Composition in Telco 2.0

Armando Ordonez, Juan C. Corrales
Telematics Engineering Group
University of Cauca
Popayán,
{jaordonez,jcorral}@unicauca.edu.co

Paolo Falcarin
ACE School
University of East London
London, UK
falcarin@uel.ac.uk

Jaime Guzman
Intelligent Web System Group
Universidad Nacional de Medellín
Medellín, Colombia,
jguzman@unalmed.edu.co

***Abstract*— Services composition in Telco 2.0 is known as convergent or unified composition. This composition is a very particular process with special features and complexity associated to technical differences between Web and Telco. As available services grow exponentially and are updated on the fly, it has become impossible for human capacity to analyse all of them and generate manually a composition plan. This paper presents a service architecture for user centred automated composition in Telco 2.0. Our approach is based on artificial intelligence planning considering the context information from the user and its access device using a cost function. Finally, a prototype of the user centred planning module is presented which takes as input the request based in the user in natural language and returns the service to execute.**

Keywords-Convergence; End User Service Composition; Automated Composition.

I. INTRODUCTION

Telco 2.0 can be described as a model that relates concepts, services and technologies of Web 2.0 with traditional telecommunication services. This combination is known as convergent or unified composition. Most common approach for services composition is performed manually at design time. As available services grow exponentially and are updated on the fly, it has become impossible for human capacity, to analyse all of them and generate manually a composition plan [1]. In the other side, using new technologies in services composition field (like semantic annotations and AI planners) allow visualizing an automated composition.

Previous works deal with this problem and present some techniques and architectures; some of these approaches, coming from European projects like SPICE [2], OPUCE [3] and OMELETTE **¡Error! No se encuentra el origen de la referencia.**, do not automate the whole service composition process. Besides, these approaches lack of ways for expressing user request through voice and do not include user context information in plan generation. In the other side, some AI Planning approaches propose customize the planning process extending planning languages like PDDL (Planning domain definition language) like [5]. From Telco 2.0 perspective, this approach makes it very complex to generate planning domains. In this context, the main contributions of this paper are (1) service architecture for automated composition in Telco 2.0 considering issues associated with convergent composition for the whole process, (2) a technique to customize the planning process using cost function using the LSP (Logic Scoring preference) method [6].

This paper is organized as follows: Section 2 the motivating scenario. Section 3 presents the issues related to convergent composition. Section 4 describes the proposed approach. Section 5 presents an evaluation of the planning technique, Section 6 presents the related work and Section 6 draws the conclusion and future work.

II. MOTIVATING SCENARIO

To illustrate our proposal, we present a case study: an environmental management system (see Fig. 1). The Environmental manager is on charge of decision making about environmental alarms and crops. In order to do so, the manager has information from sensor networks; equally, he can use Telco and Web services to process basic data and send information to all the farmers and sensors. Reuse of functionalities is a very important issue for some developing countries where budgets for technologies are limited.

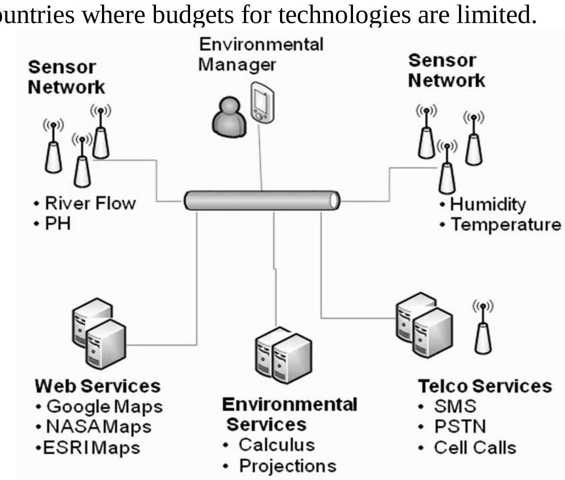

Figure 1. Telco and Web services interaction in environmental management systems.

Usually, environmental manager is an ecologist or biological expert. Therefore, his knowledge about underlying technologies is usually low. Commonly the user expresses his request in an informal way; we will illustrate this with two examples:

- "I need calculate hydrological balance of zone one and receive the resulting map to my mobile".

 ISBN: 978-1-61208-215-8

- "If the river flow of zone two is greater than 15% of average, emit an alarm to every farmer within a radius of 2 miles from the river"

The first request can be entered through voice, by a mobile device. Next, the system gathers information of sensors of zone one (preconfigured in the system or specified by coordinates). The system uses hydrological services from Internet, sending sensor data and maps from Google maps. Finally the resulting image is sent by MMS to the user mobile. In the second request, sensor data is evaluated, if necessary, an emergency map is generated. This map is created drawing a radius of 2 miles from the sensor. To do so, the system uses GIS (Geographical information systems) services and maps from internet. Finally the system informs about the alarm to farmers inside the emergency area; the best way to send the information is selected: SMS, Cell Phone call, fixed telephone call, voice message. In both cases, services from Web and Telco are used. These services work together and in coordination to save lives or help to make decisions about crops. Besides, take the services composition to the end user level, leads to ease of interaction with the system for non-experts users without IT infrastructure or personal required.

III. ISSUES FOR SERVICES COMPOSITION

Services composition in telecommunication networks is fundamentally different from Web services composition [7]. Next, some of the main issues for convergent composition are analysed:

Usability Creation: Convergent composition requires request specification in an easy way. Most of the users are not familiar with technologies such as service creation environments (SCE) or languages like Business Process Execution language (BPEL) [8]. In this scenario, application of natural language processing (NLP) techniques can be useful for deriving a formal specification from a request in natural language [9].

Time constraints: In telecommunications domains, there are real-time requirements in protocols and platforms, e.g., post-dial delay is typically bounded. In contrast, a best-effort response time is typically required from Web Services. For convergent composition, time is a crucial factor [7].

Services representation: Commonly end users should not know implementation details of services like URL, protocols or billing processes. In this scenario, Users should interact with abstract representation of services without implementation details [9].

User centred creation: A high-quality composed service might be one that fits to personal preferences. Besides, users can be connected using different devices with different capabilities. A composed service should include all these preferences [5].

Automatic code generation: In Telco domain, time constraints are mandatory. Heavy XML parsing and deep treatment of semantic inference can be used only for services representation. Using efficient mechanisms such as automated code generation which decrease deploy times are more suitable for execution environments [10].

Reconfiguration: Reconfiguration is a crucial factor in the convergent scenario as reliability of Web services is not as high as Telco services [7].

IV. OUR APPROACH

Next, we describe the architecture of our approach for automatic convergent composition (see Fig. 2.). This paper focuses in the user centred planning module and other modules of the architecture have been detailed by separate [11],[12]. In our approach, description of the services is different depending of the component. At the NLP Analyser, "Abstract services" descriptions are used based on OWL-S. While, at Plan Adapter Component: services descriptions are associated to implementation details and real descriptions of "Executable services" (e.g. WSDL). Abstract Services like "get", "inform" and "gather" are internally associated with real implementation of Executable services that are associated with SOAP or REST implementations. The relation between Executable Services and Abstract Services is created by domain-experts in environmental management through folksonomies [13]. Folksonomies offer a system of grouping services through a collaboratively method for creating and managing tags to annotate and categorize individual services. The architecture includes a user graphic interface where services can be registered. In this interface, domain experts can add tags to the services in order to be discovered later [14].

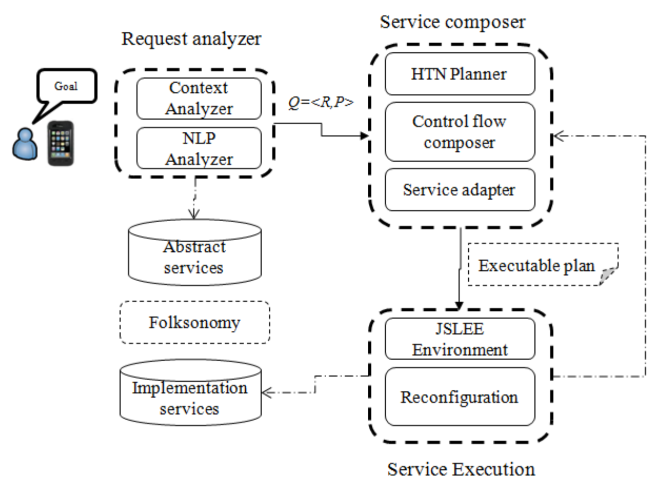

Figure 2. Telco and Web services interaction in environmental management systems.

The NLP Analyser extracts names of Abstract services (e.g., "inform farmers" this service could involve send SMS or establish a voice call) from user request and translates them in a problem for the planner. At the same time, it gathers information about control flow and context. For example, if the request is: “check sensors and inform farmers”. NLP Analyser obtains: (check (Sensors)) AND (inform (farmers)). “Check” and “Inform” are translated to problems for the planner and “AND” describes a concurrent execution that will be considered latter in the Plan Adapter Component.

The context Analyser allows including user context information in the planner; as information from devices. To do so, device reference is checked in the WURFL (Wireless Universal Resource File) and the CC/PP (Composite Capability/Preference Profiles). These repositories hold all the information about capabilities of devices in the market. And provide an effective way for analysing which services can be provided to users.

With this information, a relation factor between services and users is calculated. Next the HTN Planner component obtains a plan decomposing tasks (the higher level task is an Abstract Service from NLP Analyser) in finer atomic tasks (the lowest level task or operator is an Executable Service). This plan is sent to the services discovery component to relate abstract services with implementation services. Next, the plan is sent to plan adapter component, which includes the control flow information from the user request and translate it into an executable code ready to run in Mobicents Communication Platform. Mobicents provides a robust Java Service Logic Execution Environment (JSLEE) [15]. Finally, the reconfiguration component monitors services execution; in case of failure, services selection, or re-planning is performed. Next, a description of each component is presented.

A. Request Analyser Component

This module receives user request and translate it to machine understandable language, this process can be done automatically or include user intervention at each step. This component holds two sub components context and NLP analyser.

1) NLP Analyser: In this component, User Requirements are modelled with a query *Q* specified as a couple *<R;P>*, where *R* represents the request part of the query and *P* represents the user preferences of the query (device, network, position). In turn, *R* is specified as an n-uple *R= [{s1,s2…sn}, F]*. Each *sn* denotes an Abstract Service and F represents control flow information. For its part, each s is composed of 3-uples *<I;O;C>*; where *I* denotes the input data the user provides, *O* denotes required information to be provided as a result of the query and *C* denotes a functionality (associated with Abstract Services). For example:

Q: is a request made by an environmental manager from a cell phone. Where:
P: Cell phone reference and network capabilities.
R: “I need calculate hydrological balance of zone one and receive the resulting map to my mobile”.

From R we can expand:
s1: “calculate hydrological balance of zone one”
s2: “receive the resulting map to my mobile”
F: AND (sequence of actions)

Analysing s1: “calculate hydrological balance of zone one”
I: Zone one. (An internal variable, geo-coded location or set of coordinates of Zone one)
O: Hydrological balance map,
C: Calculate hydrological balance service.

2) Context analyser: This component extracts the context information from the user and its access device. Context is any information that characterizes the situation of an entity that can be person or computational object [16]. Common types of context include the computing context (e.g., network connectivity), the user context (e.g., profile, location), the physical context (e.g., noise levels). These features are known also as Non-functional properties (NFP) and the combination of all these profiles constitutes the User Profile [17]. For our approach we use profile model from Sutterer et.al [18]; this model consider user profiles that allows deal with any parameter that could be necessary according to the domain. In the present architecture, the user profiles define alternatives for notifications delivery based on user location, computing device and network bandwidth, as summarized in Table. 1. The column criteria is extracted from the user and define some services criteria that are assigned to operators.

TABLE I. CRITERIA FOR SERVICES SCORING BASED ON USER CONTEXT

	criteria	**Values**	**Service criteria**	**weight**
User context	Network	GPRS/ WLAN/ GSM	payload size	bytes
	Device	Cell phone, Laptop	payload size	bytes
	Location	Outdoor, indoor	voice , text	integer
User preferences	Data subscription	Yes/No	require data subscription	Boolean
	Only Free services	Yes/No	cost	value
	Voice subscription	Yes/No	voice, text	Boolean
	Delivery quality	low, medium, high	delivery warranty	integer

 ISBN: 978-1-61208-215-8

The weight defines the importance for the user in an specific situation of moment. For example, in case of emergency delivery warranty is most important that cost. And for testing of applications only free services are required for specific services. All these values are included in the cost function with normalized values of the weight and define the cost of a plan as explained further.

B. Services Composer

This component receives processed user request from the previous component and translate it into an executable plan for JSLEE environment. This process is performed automatically; however, this module can include user verification of the generated plan.

1) HTN Planner: Previous works have determined benefits of use AI planners in services composition [19]. The HTN planner produce a sequence of actions that perform some activity or task, this sequence is called a plan. Planning proceeds by using functions called methods to decompose tasks recursively into smaller subtasks, until the planner reaches primitive tasks that can be performed directly using the planning operators. For our framework, we use the planning engine SHOP2 [20], improving planning process according to user context through cost functions.

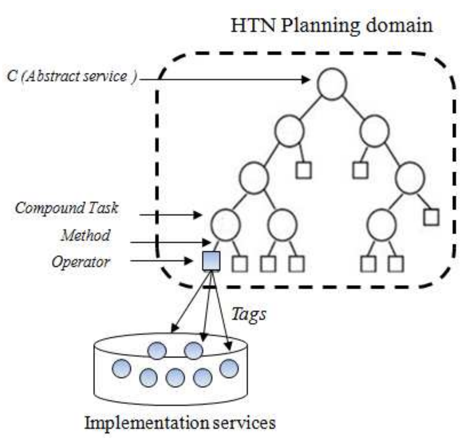

Figure 3. Relation between HTN Operators and implementation services.

A planning domain describes the context of the planning process (see Fig. 3.). This domain consists of a set of methods and operators. A task represents a service or activity to perform. A task may be either primitive or compound. A compound task is one that needs to be decomposed into smaller tasks using a method. Methods indicate how to decompose a compound task into a partially ordered set of subtasks, each of which can be compound or primitive. The highest-level task is extracted from the user request and is associated with variable *C* from *sn* as explained in the previous section. The lowest-level of task is a planning operator, i.e. a primitive service (or task) that cannot be decomposed further and is directly related with implementation Services.

An operator can be associated with many implemented services; the relation between operators and implementation services is optimized using folksonomies. A folksonomy can be described by a tripartite model of tagging activities [21], which consists of tagging entities (i.e., users, tags, and resources). *Tagging* =<*U, T, R*>; here *U* is the set of users who participate in a tagging activity. In the platform U is represented by a group of domain experts who make this tagging. *T* is the set of available tags; the available tags are the operator names. R is the set of resources being tagged; in this case, the available services. Using folksonomies improves the time for services selection, as the search space is limited to implementation services that best match with related tags. The planning domains for the platform is created at design time based on OWL-S descriptions, so that, a domain is ready at execution time and a HTN planning process can be executed with good time performance.

Each operator also has a cost (the default value is 1). The cost of a plan is the sum of the costs of the operator instances. In our approach, we assign a cost based on the relation between services and user. In this manner cost of plans lets create a ranking of plans; the lowest cost plan is selected. And in case of execution failure the second plan in the ranking can be selected.

The cost of each operator depends on the user context and preferences. For considering relation between services and user preferences, is needed to analyse the importance (weight) of each criteria for each user. For example, a user may establish that they simultaneously need low cost and MMS messages enabled service. For calculating the operator cost, is necessary to calculate a scoring technique. For the present architecture, LSP is selected; LSP is one evaluation method that extends the traditional scoring techniques to consider besides of the weight, the relation between criteria [6]. In the present architecture the cost function, receives the user preferences and returns a value that corresponds to operator cost for each service. This cost is included in the planning process as was explained before.

2) Service Adapter component: This component receives the abstract plan from the previous component, and associate services and control flow (extracted during NLP analysis) creating a composed set of Java components called SBB (services building blocks). SBB are the core components of JSLEE environment and allow create composed services referencing other SBBS. These SBB are precompiled in a repository in order to avoid unnecessary compilation processes. The executable plan includes control points and reconfiguration information included in the code necessary by fault handlers for identifying failures during execution.

C. Services execution component

This component is based on Java Service Logic Execution Environment (JSLEE). JSLEE is an emerging standard specification targeted to host convergent services. In JSLEE: Telco and Web services, as composition of these services can be represented by SBBs. In order to support changes in flow and eventual reconfiguration during execution, a dependency injection method is used to manage changes on the fly without recompilation. Service reconfiguration, is implicitly related to monitoring process, and implies to be aware of services status. When a service has an unwanted behaviour, Mobicents environment activates an alarm. These alarms initialize the reconfiguration process, which would detect the failure and proceed to determine one of three actions: if the error is caused by an atomic service, so the system selects another service and goes on. In the other way, if the problem is caused by the whole process and it can be changed: a new plan from the generated ranking is selected. Finally, if there is a problem in the middle of the executing process, a new planning process is initiated, in order to complete the task beginning from the actual state of the world, i.e., a set of values in the variables that define a system in a given moment.

The goal of the present architecture is to offer a coherent and sound framework for automate the steps in the services composition. This architecture can be used with user intervention at each step in order to verify the automation process. For example, after the Natural language analysis, the user can review the request transformation. Equally, after the composition process, the user may verify the composed plan. Thus the present architecture aims for ease the service composition process.

This architecture is oriented to environmental management domain. Most of the principles and functionalities presented here are appropriated for different domains.

V. EVALUATION OF THE PLANNING APPROACH

In terms of functionality, the planning approaches can respond to a planning problem and returns a plan. However, to provide a suitable mechanism for automated plan generation in convergent Telco 2.0 domain. This mechanism must include user context information and must be fast enough. Therefore, the prototype implementation was used to carry out some simple experiments designed to test the planning performance in a realistic planning scenario. The objective is to determine if the performance time is suitable for Telco environments.

The experiment used a planning domain with variable number of operators that describe Telco Services in the environmental management domain, the cost function relates the criteria presented in Table 1. The basic pattern of the operators is:

(:operator (!send_sms) () () () (call CalculateCost))
(:operator (!generate_map) () () () (call CalculateCost))

Each operator calls the *CalculateCost* function that calculates the related cost between service and user; to do so, the function extracts information of the user and service and apply the scoring technique. The number of plans and the number of operators was modified in order to analyse time response of the prototype. For planning, JSHOP is used, which is a Java version of the SHOP2 [22] planner. Equally the prototype uses MySQL for storing user and services, information and JDK. 1.6.0. All the experiments were performed on a machine equipped with a Pentium Dual Core T4400 2.2 GHz processor, with 4 GB of main memory and running Windows as OS.

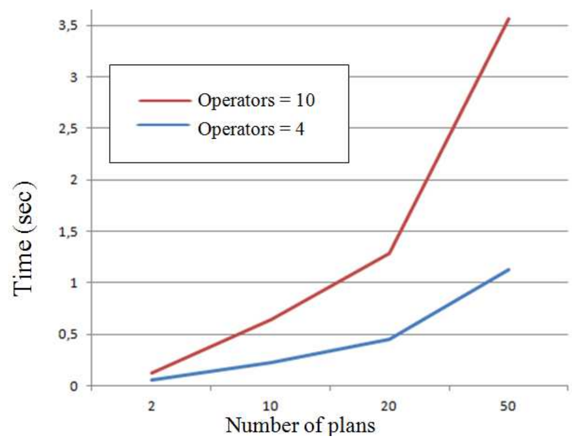

Figure 4. Time performance of the planning module

As expected, Fig. 4 shows that the time increases with the number of plans generated and depending of the operators involved in the planning domain. Convergent services usually have between 4 and 10 operators to avoid extreme complexity, and the resulting ranking usually can have up to 5 plans. In this context, despite the exponential cost of planning module, the figure shows that this one can be used for 5 plans with acceptable execution time (0.5 sec). Future work include decreasing the time for more operators, the last could be done including some heuristics in the planning algorithm.

VI. RELATED WORK

Our approach is focused on automation of user centred service composition, and we intend to apply our framework to the environmental management. Our approach considers all three phases or components for automated service composition: request analysis, services composition and service execution. Previous works have proposed frameworks for automated services composition, like Kim et.al [23] and Rao et.al [24]. They both present phases for

 ISBN: 978-1-61208-215-8

automatic composition, focusing only on Web domain without concern on execution. Shia et al. [25] and da Silva et al. [9] present frameworks for automatic composition. These frameworks exploit natural language processing and semantic annotations for services matchmaking based on SPATEL language [25]. They do not address the validation of the non-functional properties and are focused only on the request analysis and plan generation. Our approach deal with all the phases including execution and reconfiguration based on JSLEE environments. Sirin et al. [26], provide an algorithm to translate OWL-S service descriptions to a SHOP2 domain and makes planning based on services. Although, this works lack of details of implementation in real environments and focus only on Web services. Other authors present approaches for Web service composition based on AI planning with preferences [27]; however they propose extensions to standard planning language and adaptation of planners, adding a new level of complexity to the automated composition (other levels include semantic annotation of telecommunication and Web services and basic planning domain definition). Table 2 outlines a comparison of related work, based in the following criteria: first, if the work deals with all the phases in the service composition process including reconfiguration. Second, if the approach for service composition is user-centred. Third, if the approach takes in account convergent considerations.

TABLE II. COMPARISON OF RELATED WORKS

works	Include all phases	User centred	Consider Convergence
[23][24]	No, just the request analysis and the service creation	No	No, they focus on Web domain
[25][5]	No, just the request analysis and the service creation	No	Yes
[26]	No, just the OWL-S based planning	No	No, focused on Web domain
[3]	No, just the planning with preferences	Yes	No, focused on Web domain

VII. CONCLUSION AND FUTURE WORK

Automated convergent composition is a very intensive research area; previous works have worked on some aspects of this process. However, there is not a complete framework to develop this process. None of the above proposals presents details to apply proposed frameworks to Telco environments in a real environment, the works presented by Shia et al. [25] [9] are the most relevant for us in the literature, many elements are similar to our work but our approach has a different direction. They deal with complex treatment of SPATEL language and ontologies in order to reach automated User centred composition. We deal with user profile information to customize AI planning. Equally, our approach is tending to reach low execution times and include mechanisms for automated reconfiguration. Our future works include the development of mechanisms for automation of planning domain creation, and experimentation with other planners in order to consider reconfiguration in different phases of the process and better execution times. Equally we are extending the preferences criteria in order to get a better personalized experience for the user.

ACKNOWLEDGMENT

The authors thank University of Cauca, University of East London and Colciencias for supporting this research.

REFERENCES

[1] S-C. Oh, D. Lee, and S. Kumara, “A Comparative Illustration of AI Planning-based Web Services Composition,” ACM SIGecom Exchanges,vol. 5, Jan. 2006, pp. 1-10, doi: 10.1145/1124566.1124568.

[2] P. Falcarin, “Service Composition Quality Evaluation in SPICE Platform,” High Assurance Services Computing, Springer US, pp. 89-102, 2009.

[3] J.C. Yelmo, J.M. del Alamo, R. Trapero, P. Falcarin, Y. Jian, B. Cairo, and C. Baladron, “A User-Centric service creation approach for Next Generation Networks,” Proc. Innovations in NGN: Future Network and Services (K-INGN), May 2008, pp. 211-218, doi: 10.1109/KINGN.2008.4542268.

[4] O. Chudnovskyy et al., “End-user-oriented telco mashups: the OMELETTE approach, “ Proc. of the 21st international conference companion on World Wide Web (WWW '12 Companion), Abril 2012, pp. 235-238, doi: 10.1145/2187980.2188017

[5] J. Baier and S. McIlraith, “Planning with Preferences,” AI Magazine, vol. 29, 2008, pp. 25-36.

[6] H.Q. Yu and S. Reiff-Marganiec, “A Method for automated Web Service Selection,” Proc. IEEE Congress on Services, July 2008, pp. 513-520, doi: 10.1109/SERVICES-1.2008.8.

[7] G. Bond, E. Cheung, I. Fikouras, and R. Levenshteyn, “Unified telecom and Web Services Composition: problem definition and future directions,” Proc. 3rd International Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm’09), ACM Press, pp. 1-12 ,doi: 10.1145/1595637.1595654.

[8] A. Arkin et al., “Web Services Business Process Execution Language Version 2.0. Committee Draft,” WS-BPEL TC OASIS, December 2005.

[9] E. da Silva¸ L. Ferreira, and M. van Sinderen, “Towards runtime discovery, selection and composition of semantic services,” Computer Communications, vol. 34, Feb. 2011, pp.159-168, doi: 10.1016/j.comcom.2010.04.003.

[10] A. Lehmann et al., “TeamCom: A Service Creation Platform for Next Generation Networks,” Proc. Fourth International Conference on Internet and Web Applications and Services (ICIW’ 09), IEEE Computer Society, May 2009, pp.12-17, doi: 10.1109/ICIW.2009.10.

[11] E. Pedraza, J. Zúñiga, L. Suarez, J.C. Corrales, “Automatic Service Retrieval in Converged Environments Based on Natural Language Request,” Proc. The Third International Conferences on Advanced Service Computing, Sep. 2011, pp. 52-60.

[12] A. Ordonez, J. Corrales, P. Falcarin, “Natural language processing based Services Composition for Environmental management”, Proc. 7th International Conference on Systems of Systems Engineering. July 2012, in press.

[13] I. Peters, “Folksonomies: Indexing and retrieval in Web 2.0,” De Gruyter Saur, Berlin, 2009.

[14] A. Ordonez, J. Corrales, P. Falcarin, “Automated context aware composition for convergent services”, Proc. 7th International Conference on Systems of Systems Engineering. July 2012, in press.

[15] D. Ferry et al. JSR 240, JAIN SLEE (JSLEE) v1.1, July 2008.

[16] A. Dey, “Understanding and Using Context,” Journal of Personal and Ubiquitous Computing, vol. 5, Feb. 2001, pp. 4-7, doi:10.1.1.31.9786.

[17] S. Panagiotakis, M. Koutsopoulou, A. Alonistioti, and S. Thomopoulos, “Context sensitive user profiling for customised service provision in mobile environments,” Proc. 16th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC 2005), IEEE Press, Sept. 2005, pp. 2014-2018, doi: 10.1109/PIMRC.2005.1651793.

[18] M. Sutterer, O. Coutand, O. Droegehorn, K. David, and K. Der Sluijs, “Managing and Delivering Context-Dependent User Preferences in Ubiquitous Computing Environments,” Proc. International Symposium on Applications and the Internet Workshops (SAINT-W '07), IEEE Computer Society, Jan. 2007, pp. 4-4, doi: 10.1109/SAINT-W.2007.60.

[19] J. Xu, K. Chen, and S. Reiff-Marganiec, “Using Markov Decision Process Model with Logic Scoring of Preference Model to Optimize HTN Web Services Composition,” International Journal of Web Services Research, vol.8, 2011, pp. 53-73, doi: 10.4018/jwsr.2011040103.

[20] D. Nau, O. Ilghami, U. Kuter, J. William, D. Wu and F. Yaman, “SHOP2: An HTN Planning System,” Journal of Artificial Intelligence Research, vol.20, Dec. 2003, pp. 379-404, doi: 10.1.1.72.188.

[21] H-L. Kim, J. Breslin, H-G. Kim and J-H. Choi, “Social semantic cloud of tags: semantic model for folksonomies,” Knowledge Management Research & Practice, 2010, pp. 193–202. doi:10.1057/kmrp.2010.10.

[22] D. Nau, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman, “SHOP2: An HTN planning system,” Journal of Artificial Intelligence Research, 2003, vol. 20, pp. 379-404, doi: 10.1613/jair.1141.

[23] A. Kim, M. Kang, C. Meadows, and J. Sample, “A Framework for Automatic Web Service Composition,” Technical Report, Naval research Lab USA, 2009.

[24] J. Rao, and X. Su, “A Survey of Automated Web Service Composition Methods,” Proc. First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), July 2004, pp. 43-54, doi: 10.1.1.122.6700.

[25] M. Shiaa, P. Falcarin, A. Pastor, F. Lécué, E. Silva, and L. Ferreira, “Towards the Automation of the Service Composition Process: Case Study and Prototype Implementations,” Proc. ICT Mobile Summit, June 2008, pp. 1-8.

[26] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, “HTN Planning for Web Service Composition using SHOP2,” Web Semantics: Science, Services and Agents on the World Wide Web, vol. 1, Oct. 2004, pp. 377-396, doi: 10.1016/j.websem.2004.06.005.

[27] S. Sohrabi, N. Prokoshyna, and S. McIlraith, “Web Service Composition via the Customization of Golog Programs with User Preferences,” Conceptual Modeling: Foundations and Applications, Springer-Verlag Berlin, Heidelberg, pp. 319-334, 2009.

.