

HAUTO: Automated composition of convergent services based in HTN planning

HAUTO: Composición automática de servicios convergentes, basada en la planificación HTN

A. Ordonez¹, J. C. Corrales² and P. Falcarin³

ABSTRACT

This paper presents HAUTO, a framework able to compose convergent services automatically. HAUTO is based in HTN (hierarchical task networks) Automated Planning and is composed of three modules: a request processing module that transforms natural language and context information into a planning instance, the automated composition module based on HTN planning and the execution environment for convergent (Web and telecom) services. The integration of a planning component provides two basic functionalities: the possibility of customizing the composition of services using the user context information and a middleware level that integrates the execution of services in high performance telecom environments. Finally, a prototype in environmental early warning management is presented as a test case.

Keywords: telecommunications, services composition, automated planning.

RESUMEN

Este artículo presenta HAUTO, un framework que permite componer automáticamente servicios convergentes. HAUTO se basa en la planificación automática basada en redes jerárquicas de tareas HTN y se compone de tres módulos: un módulo de procesamiento de la solicitud, que transforma el lenguaje natural e información del contexto en una instancia de planificación, el módulo de composición automática basado en la planificación HTN; y el tercero, es el entorno de ejecución para servicios convergentes (Web y Telecom). La integración de un componente de planificación provee dos funcionalidades básicas: la posibilidad de personalizar la composición de los servicios usando la información del contexto del usuario y un middleware que se integra con la ejecución de los servicios en entornos de Telecomunicaciones de alto rendimiento. Finalmente, se presenta como caso de estudio un prototipo para la gestión de alertas ambientales tempranas.

Palabras clave: telecomunicaciones, composición de servicios, planificación automática.

Received: March 7th 2013

Accepted: November 29th 2013

1. Introduction

Today, users are commonly interacting with smart apps in their smartphones through voice as well as with intelligent Web pages such as shopping sites or tourism platforms. Due to the latter, users are keen on using an even greater number of new and complex services. In this scenario, when a single service cannot accomplish a user request, it becomes necessary to compose many services for solving the original request.

In the telecom domain, service composition is carried out by experts, through graphical interfaces. However, these graphical interfaces require advanced knowledge of technical protocols and concepts. This approach is valid in telecommunications due to the fact that the composition uses exclusively telecom services, whose number is manageable. However, convergent composition inte-

grates telecom services with services from the Internet, commonly called Web services. These Web services may change, become unavailable and grow in number to unmanageable sizes. This means that manual convergent composition is unfeasible in practice. Furthermore, graphical interfaces cannot be used during common activities of the users such as walking or driving.

Previous works from both academia and the industry (see related works section) have revolved around this topic. However, most of the previous approaches in the area focus only on Web services and no prior work, that we are aware of, is able to combine them with telecom features. Another important limitation of these approaches is that they require that users express their requests using complex methods such as writing in formal languages. This latter issue makes the interaction of a large number of people with

¹ Armando Ordoñez. MSc, PhD (c). Affiliation: Researcher, Universidad del Cauca, Colombia., E-mail: jaordonez@unicauca.edu.co

² Juan C. Corrales. PhD. Affiliation: Full Professor, Departamento de Telemática, Universidad del Cauca, Colombia. E-mail: jcorral@unicauca.edu.co

³ Paolo Falcarin. PhD. Affiliation: Senior Lecturer, University of East London, U. K.

E-mail: falcarin@uel.ac.uk

How to cite: Ordoñez, A., Corrales, R., Falcarin, P., HAUTO: Automated composition of convergent services based in HTN planning., Ingeniería e Investigación, Vol. 34, No. 1, April, 2014, pp. 66 – 71.

little or no technical background with the composition systems unfeasible (Chen et al., 2009).

Previous works have also determined the benefits of using automated planning in service composition. Within the wide range of planning methods, HTN (hierarchical task networks) has presented high performance. One disadvantage of this approach is that in order for decomposition rules of such hierarchical structures to be sound, the user request as well as the domain must be encoded in advance using ontologies with prior expert knowledge of the domain. The latter issues imply that the expert in the domain (for example a biologist or a doctor) must specify the request using an ontology language.

In spite of the aforementioned, in some domains such as environmental early warnings, the main procedures and their associated services are relatively few. For this reason, it is feasible that some domains should be described by domain experts with the support of IT experts. The latter leaves open the issue of the specification of the user request.

HAUTO (HTN Automated convergent composition) is a framework for automated convergent services composition centered in the final user. This framework is based in automated planning, and includes different modules that ranges from the user request to the execution. In the HAUTO framework, the domain and services description is performed using an annotating tool that offers a graphical interface to the experts for describing convergent services using automated planning languages. Furthermore, HAUTO includes an automatic translation of the input (user request) into a formal language (planning language) without intervention of experts.

Some of the modules of HAUTO have been presented elsewhere (Ordonez et al., 2012), (Ordonez et al., 2012a). In the present work, we focus in the *automated composition module* based in HTN. Previous works have tackled the service composition based in HTN. However, the inclusion of context information relevant to convergent scenarios has not been included in the planning mechanism. The information about context is necessary for providing the adequate service to the user according to place, device and situation. HAUTO includes the customization of convergent composition using contextual information.

In this vein, the main contributions of this work are: (1) An automated planning based framework for automated convergent composition based in HTN; (2) a mechanism for including user context information into the HTN planning; and (3) a mechanism for tagging the convergent services using HTN descriptors.

We use the environmental early warning domain as the study case, however, the HAUTO framework is applicable to any other similar domain that meets the following conditions: i) the composite services should include Web services and/or basic telecom features like calls or send SMS, ii) the number of services should be rather limited due to the specialized field of action, iii) the procedures and services follow international standards such as ISO, so these procedures and its associated services can be described in practice using semantic annotations by domain experts (with the support of IT experts), as the number of annotations needed is relatively small and the procedures and decomposition rules are shared by all the participants of the domain.

This paper is organized as follows: first, the framework architecture is described in section 2, section 3 focuses on the automated

generation module and section 4 describes a prototype of the proposed approach and its validation. Section 5 presents related works and finally, section 6 presents conclusions.

2. Architecture of the HAUTO Framework

The next considerations are the foundations for designing the HAUTO framework: i) ease of use, ii) integration with high performance platforms used in telecommunications, iii) reaction to exceptions and fault tolerant: planning systems deployed in dynamic environments must be able to react to potential failures during execution of the services and iv) personalization: users may express their preferences in any request. Typical cases include minimizing the time of the execution if it is an urgent request or minimizing the price of the service if no constraints exist.

The architecture of HAUTO is depicted in Fig. 1, and it is composed of six sub-modules grouped in two modules.

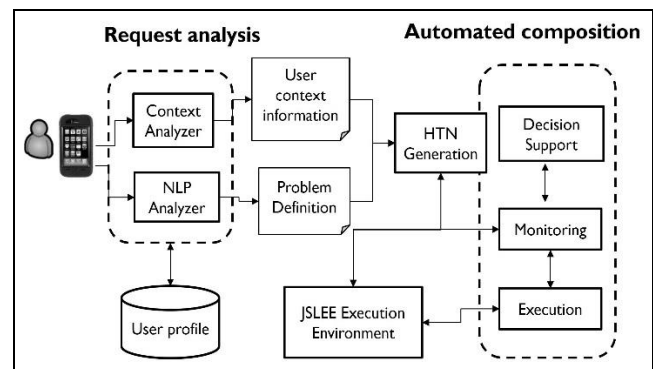


Figure 1. Architecture of the HAUTO framework

The *Request Analysis module* receives the user request in natural language (*NLP Analysis sub-module*) and translates it into a machine understandable format that includes context information (*Context Analysis sub-module*). Based on the foregoing the *HTN Generation sub-module* makes a translation from the processed request into a description in HTN. The automatically generated request in HTN is in turn the input to the *automated composition module* which performs the service composition through the *HTN planner sub-module*. HAUTO is based in HTN, however this module can be switched for including other planning algorithms (Ordonez et al., 2012b).

Finally, HAUTO uses a robust telecom execution environment called JSLEE (Java Service Logic Execution Environment). The integration between the automated composition module and JSLEE is done in the *execution sub-module*. On the other hand, the *monitoring sub-module* is aware of the execution of the composite service and triggers the reconfiguration processes as needed. Next, each module is described in detail.

2.1 Request analysis module

The Request Analysis Module is composed of two sub-modules: The NL analysis sub-module, (this is a complex task widely described elsewhere (Ordonez et al., 2012), parses and processes the user request in natural language and obtains a basic decomposition and analysis of the user request, identifying a set of work from the context: verbs, proper names, adjectives, emergency words (e.g. help, urgently, dying). Whereas the Context Analysis sub-module enriches the user request with information about user preferences, device capabilities and situational context.

Next, the HTN Generation sub-module makes a translation from the request into a description in HTN. Every time a user makes a request, an HTN descriptor of his request (goal) is generated.

2.2 JSLEE Execution

This component is based on a JSLEE that is an emerging standard specification targeted to host telecom convergent services. In JSLEE telecom and Web services, as well as the composition of such individual services, java components known as service building blocks (SBB) are represented. SBB encapsulate the logic for calling external Web and telecom services; furthermore, SBB may include workflow information for coordinating the execution of composite services.

2.3 Automated composition module

This module is based in the Planning, Learning and Execution Architecture (PELEA) (Guzmat et al., 2012) (for illustrative purposes only the most relevant modules are shown in Fig. 1). The HTN planner sub-module carries out the plan generation. This plan represents the composite service (this process is explained in detail in the following section). This sub-module generates several plans and organizes them by rank according to the user preferences that are included in the composition process.

The execution sub-module links the automated composition module with the JSLEE. This module translates the resulting plans from the HTN planner into a new SBB that represents the composite service. The new SBB contains the workflow as well as links to other SBB that represent individual services.

The composite SBB includes control points and reconfiguration information necessary for fault handlers to reconfigure in case of malfunctioning. The monitoring sub-module is aware of the execution in the JSLEE. When a service presents an unwanted behavior (network faults or service changes), the JSLEE activates an alarm that initializes the reconfiguration process. The reconfiguration detects the failure and proceeds to determine one of three actions: if the error is caused by an atomic service, the monitoring sub-module selects another equivalent service and goes on, if the problem is caused by the whole process and needs to 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 based in the current state of the execution until the point of the failure.

3. Automated composition module

The automated planning starts from a description of the system (modeling user context and the state of the services), followed by a description of the expected goals and set of possible actions. By using the above information, the planner tries to find a plan that generates a sequence of actions from the initial state that leads to one of the goal states. The initial state of the world and the desired goals conform the planning problem., The set of possible actions is called the planning domain.

Different approaches for automated planning can be found in the available literature; these planners can be divided in domain dependent and domain independent. The domain dependent planners are the most successful planning systems in practice (Gallab et al., 2004). HTN is a domain dependent technique for planning that has been widely used in service composition. HTN planning is especially promising for this purpose, because the concept of task decomposition in HTN planning is very similar to the concept

of composite process decomposition in OWL-S process ontology (Sirin et al., 2004).

Because of dynamicity of the Web services environment, the use of deterministic planners such as HTN in the automatic service composition has been questioned. Deterministic planners may be integrated within platforms with reconfiguration mechanisms (Guzman et al., 2012).

In HTN planning, a *task* represents the activities to perform. A task may be either primitive or compound (in our case, represent an individual and composite service, respectively). 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 lowest level task that cannot be decomposed further is called a *planning operator*. In HAUTO, the highest level task or goal is extracted from the user request and the lowest level planning operator represents a primitive service directly associated with implementation services.

In HAUTO, the services representation is done in levels. The operators are associated with high abstraction level and are used for generating the abstract plans. On the other hand, implementation services associated with physical services are linked technically to service descriptions such as WSDL. This model has been tested previously by Agarwal et al., (2005).

3.1 Cost function

In a standard planning process, all operators have the same cost (the default value is 1) and 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 relationship between services and user context. In this way, the cost allows the creation of a ranking of plans in which the best cost is selected. In the case of a malfunction, the second plan in the ranking may be selected. In HAUTO, the contextual information includes three dimensions: device, user profile and situation.

Each of these obtains information from different sources and defines the selection of domains. The device dimension gathers information from the user device and the network; the user profile dimension gathers information using the users ID to look up their preferences in the preferences repository in the system. Finally, the situation dimension analyses the Natural Language request identifying specific words such as “urgently” or “emergency” (Ordóñez et al., 2012).

The relationship between context and services is evaluated using the modified Logic Scoring Preference (LSP) method (Yu et al., 2008), that extends the traditional scoring techniques including the relation between criteria as shown next:

$$cost = (|\omega_1| E_1^r + |\omega_2| E_2^r + \dots + |\omega_n| E_n^r)^{1/r} \quad (1)$$

where ω is the weight of each criterion $\omega < 0$ represents that a lower evaluation result is better (e.g. for a case in which the price must be decreased), $\omega > 0$ means a higher result is preferable (e.g. a case in which for bandwidth should be preferably high). E is the defined evaluation function for providing the scores of the service for each criterion and r represents the logic relation between different criteria. The resulting cost reflects the comparison relations between services and user preferences.

The cost function allows for including different not functional properties such as availability, cost and response time that can be

very useful in customizing service composition (Agarwal et al., 2005).

3.2 planning algorithm

The planning algorithm is based in SHOP2 (Sirin et al., 2004) and it is extended for including context information as shown below:

```

PROCEDURE HAUTO (s, T, D)
P = Initial Empty plan
C = Cost of the plan
T0 ← {t ∈ T}
LOOP
  IF T = ∅ then RETURN P
  CHOOSE any t ∈ T0 (nondeterministically)
  IF t is a primitive task THEN
    A ← {(a): a is an operator in D}
    IF A = ∅ THEN RETURN failure
    CHOOSE a ∈ A (nondeterministically)
    MODIFY s by deleting DEL(a) and adding ADD(a)
    APPEND a to P
    Ca = Cost Function: (resulting of context function)
    C = C + Ca (updating total cost of the plan)
    MODIFY T by removing t
    T0 ← {t ∈ T;}
  ELSE
    M ← {(m: m is an applicable method)}
    IF M = ∅ THEN RETURN failure
    CHOOSE m ∈ M (nondeterministically)
    MODIFY T by removing t, adding sub(m)
    IF sub(m) ≠ ∅ THEN
      T0 ← {t ∈ sub(m)}
    ELSE T0 ← {t ∈ T}
  REPEAT
END SHOP2

```

The arguments include the initial state s , a partially ordered set of tasks T , and a domain description D . The algorithm chooses a task $t \in T$ that has no predecessors. At this point, there are two cases. The first case is if t is primitive, that is, if t can be accomplished using an action (i.e., an operator that represents an individual service). In this case, the algorithm finds an action a that matches t and whose preconditions are satisfied in s and applies a to s (if no action can be found, this branch of the search space fails). The second case is where t is compound, in which case, a method needs to be applied to t to decompose it into subtasks. For this situation, the planner chooses a method m that will decompose t into subtasks (if no method can be found, this branch of the search space fails). If there is a solution plan that involves m , then the actions in P will be the leaf nodes of a decomposition tree. As long as the search tree is expanded, the global cost of the planning process is increased according to operator cost. The cost of each operator depends on the user context and preferences as previously explained.

3.3 Service description

Describing the domain and the request HTN-decomposition methods is a very difficult task as mentioned beforehand. In HAUTO we address this issue through the use of a tool that allows annotating services based in domain dependent ontologies. With the annotation tool, domain experts create annotations for the services. The tool supports annotations defined by SAWSDL (Semantic Annotations for WSDL and XML Schema). Next, we show how these descriptions are translated into HTN descriptions.

4. Prototype and experimental evaluation

We have selected the environmental early warning field for a case study. The objective of environmental early warning is to create contingency plans that help resolve potentially dangerous situations, such as floods or tropical storms. The detection of these situations combines both information gathered from sensors around the monitored area and the input of relevant users (Haddow et al., 2010). Contingency plans include actions to monitor rivers, determine affected areas, warn the villagers and coordinate the logistics of the evacuations. Since the participation of a human is required in critical scenarios, we assume that all requests are triggered by users. Yet, in rural environments, the access to a device capable of sending a request in the format specified by the system, may be limited. This means that the system should be able to process natural language so it can be initiated through simple communication means like a phone call or an SMS. A sketch of an environmental management system is presented in Fig. 2. Users can express their requests informally; an example could be: "If the river flow of zone 1 is greater than 15% of the safety limit on average, send a warning to every farmer within a radius of 2 miles from the river and create an action map including emergency and rescue groups". A composition of services that solves this request could be: gathering information from the sensors, using hydrological services from the Internet to process the sensed data, obtaining the map from Google maps, getting the list of potential victims and calling or sending SMS.

In our prototype, the NLP module uses an annotation based information extraction system called GATE (<http://gate.ac.uk/>), which offers a Java based architecture that allows plugging in several kinds of NLP software, such as taggers, sentence splitters and named entity recognizers.

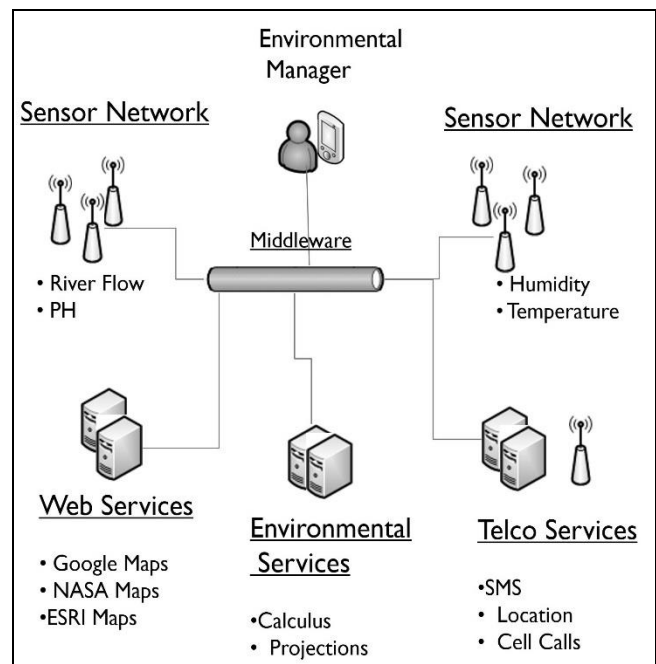


Figure 2. Elements involved in the Early Warning Domain for Environmental Management

For example, the request "I need the evacuation route" will be translated into the HTN description for requests (we only show the operators for simplicity.):

```
(defproblem (...)
  (achieve-goals
    ((calculated evacuation_route my_position) (informed user1) ) )
  ))
```

Equally, in our approach we use an annotation interface based in RadiantWeb (Guttula, 2012), a tool that allows relating in a drag and drop fashion, the concepts from one ontology (right pane in Fig. 3) to the WSDL elements of the convergent services (left pane in Fig. 3). This tool offers an instrument for experts to easily annotate the services.

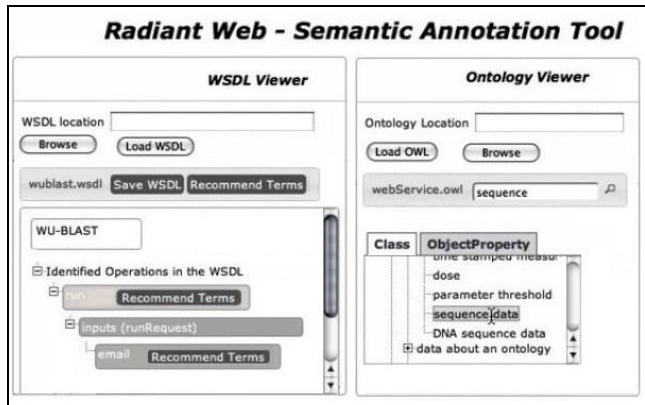


Figure 3. Annotating tool, the left panel shows the service description and the right panel shows the ontology concepts

The planner sub-module is implemented using a Java version of SHOP2 (<http://www.cs.umd.edu/projects/shop/>). In addition, the execution environment is a JSLEE based Server called Mobicent (<http://www.mobicents.org/>). The software associated with this paper is available for free download on the project site (<http://www.unicauca.edu.co/~jaordonez/auto>).

Some experimentation was carried out in order to test the planning system in a realistic scenario. The objective was to determine if the performance time is suitable for telecom environments. The experiment used a planning domain with a variable number of operators that describe telecom services in the environmental management domain; the cost function relates the criteria with the users. Each operator (service) calls the Calculate Cost function that calculates the associated cost as explained previously. The number of plans and the number of operators was modified in order to analyze response time.

4.1 Experimental evaluation

Our approach applies HTN planning techniques to convergent services. The correctness of basic HTN planning has been analyzed previously (Nau et al., 2003). In this work we model a convergent domain in HTN and include some context information during the planning process.

To check the viability of the use of automated planning in HAUTO, some experimentation was done. First, we checked whether the system was able to scale up to a request that generates a composition of services. An automatically generated instance was modified to increase the number of methods (representing services). The number of services was increased from 1 to 40. The problem was left unmodified. The metric used to customize the planning process was based in the cost function described before.

For the experimental environment JSHOP was used as HTN planner, MySQL for storing user profile and services, 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. The results of the first test are shown in Table 1.

Table 1. Methods vs. time using HTN with cost function

Methods	time (sec)
1	0,068
5	0,06
10	0,07
20	0,053
40	0,08

Table 1 shows how the time increases with the number of services. During the tests, as the time increased, the planner behaved as expected. However, our approach is still capable of finding a plan with 40 services in less than 1 second while minimizing the metric. This performance is competitive against “ad-hoc” algorithms for services composition (Pistore et al., 2005).

The second test was oriented to test the technical feasibility of using this planner for generating many plans. Each of these plans represents a composite service. Fig. 4 shows that the time increases with the number of plans generated and that it depends on the operators involved in the planning domain. It is worth noting that convergent services usually have between 4 and 10 operators (representing services) to avoid extreme complexity and the resulting ranking from applying the HTN planning with preferences based in the user, usually can have up to 5 plans (representing 5 possibilities in the case of mal function).

In this context, despite the exponential cost of the planning module, Fig. 4 shows that this can be used for 5 plans under an acceptable execution time (less than 0.5 sec). Future work includes decreasing the time for more operators: this could be done including some heuristics in the planning algorithm.

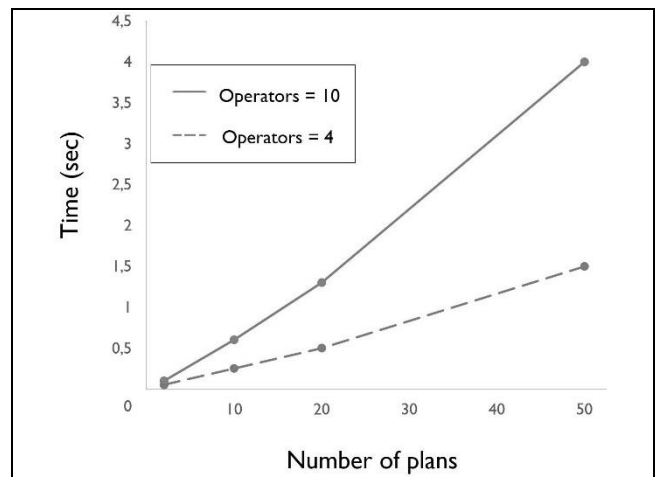


Figure 4. Time performance of the planning process

5. Related Work

Maecenas Zhu et al. (2011) present the Hybrid Service Creation and Execution Environment (HSCEE), a Template-based service creation platform with low latency service execution. HSCEE is based in on BPEL templates, but most of the design tasks must be done manually. Natural language processing is also not considered.

The OPUCE Project (Yu et al., 2012) presents a User-centric service creation environment (SCE) for the composition of Web and telecom services. This SCE includes a semantic browser, a user-centric environment and a translator to Web services Business Process Execution Language (BPEL). This work is focused on the use of semantic technologies for service description and discovery; however, it does not consider natural language interfaces and the composition is not done automatically. Kim et al. (2009) present a framework for automatic composition of services, focusing only on Web domains without including the execution phase. Da Silva et al. (2011) propose a framework for automatic composition exploiting natural language processing and semantic annotations for service matchmaking based on SPATEL language. However, they do not address context information. Our approach deals with all the phases including execution and reconfiguration based on JSLEE environments. Finally, Hatzi et al. (2012) present a semantic-based technique for discovery and approximate composition of services; yet, this work does not consider convergent scenarios and does not provide a natural language interface for accessing the system.

6. Conclusions

This work integrates HTN Planning in convergent composition as the deliberative process. We allowed the use of preferences, modeling them as the metrics of the planning instance. In the near future, we want to continue our research developing mechanisms for the automation of the generation of the planning domain. Further experimentation with different planners will be performed as well so that an increase in performance can be achieved; this includes working with Markov Decision Planners (Chen et al., 2009) and Automated Planning with extensions for including preferences. Finally, we will research ways to apply this work in domains with similar features, such as clinical history and Digital TV based learning.

Acknowledgements

The authors thank the University of East London and University of Cauca for supporting this project.

References

- Agarwal, V., Chaffe, G., Dasgupta, K., Karnik, N., Kumar, A., Mittal, S., Srivastava, B., Synth: A system for end to end composition of web services., *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 3, No. 4, 2005, pp. 311-339.
- Chen, K., Xu, J., Reiff-Marganiec, S., Markov-htn planning approach to enhance flexibility of automatic web service composition., *Proc. of IEEE ICWS*, Los Angeles, July, 2009, pp. 9-16.
- Guzman, C., Alcazar, V., Prior, D., Onainda, E., Borrajo, D., Fernandez-Olivares, J., Quintero, E., PELEA: a domain-independent architecture for planning, execution and learning., *Proc. ICAPS*, Vol. 12., Sao Paulo, Freiburg, 2012, pp. 38-45.
- Da Silva, E. G., Pires, L. F., Van Sinderen, M., Towards runtime discovery, selection and composition of semantic services., *Computer Communications*, Vol. 34, No. 2, , Feb., 2011, pp. 159-168.
- Ghallab, M., Nau, D., Traverso, P., *Automated Planning: Theory & Practice.*, Morgan Kaufmann, 2004.
- Hoffmann, J., Bertoli, P., Pistore, M., Web service composition as planning, revisited: in between background theories and initial state uncertainty., *Proc. 22nd national conference on Artificial intelligence*, Vancouver, July, 2007, pp. 1013-1018.
- Haddow, G., Bullock, J., Coppola, D. P., *Introduction to emergency management*, 4th ed., New York, Butterworth-Heinemann, 2010.
- Hatzi, O., Vrakas, D., Nikolaidou, M., Bassiliades, N., Anagnostopoulos, D., Vlahavas, D., An integrated approach to automated semantic web service composition through planning., *IEEE Transactions on Services Computing*, Vol. 5, No. 3, 2012, pp. 319-332.
- Huang, C, Lee, G. M., Crespi, N., A semantic enhanced service exposure model for a converged service environment., *IEEE Communications Magazine*, Vol. 50, No. 3, March, 2012, pp. 32-40.
- Kim, A., Kang, M., Meadows, C., Loup, E., Sample, J., A framework for automatic web service composition., Washington D.C., US Naval research Lab., Apr., 2009.
- Nau, D. S., Au, T. C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., Yaman, F., SHOP2: An HTN planning system., *J. Artif. Intell. Res. (JAIR)*, Vol. 20, 2003, pp. 379-404.
- Ordonez, A., Corrales, J. C., Falcarin, P., Natural language processing based Services Composition for Environmental management., *Proc. SoSE 12*, Genoa, July, 2012, pp. 497-502.
- Ordonez, A., Alcázar, V., Borrajo, D., Falcarin, P., Corrales, J. C., An Automated User-Centered Planning Framework for Decision Support in Environmental Early Warnings., *Proc. IBERAMIA*, Cartagena, November, 2012, pp. 591-600.
- Pistore, M., Traverso, P., Bertoli, P., Marconi, A., Automated synthesis of executable web service compositions from BPEL4WS processes., *Proceedings from WWW (Special interest tracks and posters)*, ACM, 2005, pp. 1186-1187.
- Sirin, E., Parsia, B., Wu, D., Hendler, J., Nau, D., HTN planning for web service composition using SHOP2., *Journal of Web Semantics*, Vol. 1, No. 4, 2004, pp. 377-396.
- Yu, H. Q., Reiff-Marganiec, S., A Method for automated Web Service Selection., *Proc. IEEE SERVICES '08 Conf.*, Honolulu, July, 2008, pp. 513-520.
- Yu, J., Sheng, Q. Z., Han, J., Wu, Y., Liu, C., A semantically enhanced service repository for user-centric service discovery and management., *Data & Knowledge Engineering*, Vol. 72, Feb., 2012, pp. 202-218.
- Sohrabi, S., Prokoshyna, N., McIlraith, S., Web service composition via the customization of Golog programs with user preferences, *Conceptual Modeling: Foundations and Applications.*, Springer-Verlag Berlin Heidelberg, 2009, pp. 319-334.
- Hatzi, O., Vrakas, D., Nikolaidou, M., Bassiliades, N., Anagnostopoulos, D., Vlahavas, I., An integrated approach to automated semantic web service composition through planning., *IEEE Transactions on Services Computing*, Vol. 5, No. 3, 2012, pp. 319-332.
- Zheng, J., Harb, O., Stoeckert, C., Ontology Driven Data Collection for EuPathDB., *Proc. ICBO*, Buffalo, July, 2012, pp. 95-100.
- Zhu, D., Zhang, Y, Cheng, B., Wu, B., Chen, J., HSCEE: A Highly Flexible Environment for Hybrid Service Creation and Execution in Converged Networks., *Journal of Convergence Information Technology*, Vol. 6, No. 3, March, 2011, pp. 264-276.