HAL
open science

Context-Aware Service Discovering System for Nomad Users

Pathathai Na Lumpoon, Mu Lei, Isaac Caicedo-Castro, Marie-Christine Fauvet, Ahmed Lbath

To cite this version:

Pathathai Na Lumpoon, Mu Lei, Isaac Caicedo-Castro, Marie-Christine Fauvet, Ahmed Lbath. Context-Aware Service Discovering System for Nomad Users. 7th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2013), 2013, Chiang Mai, Thailand. hal-00953083

HAL Id: hal-00953083

https://inria.hal.science/hal-00953083

Submitted on 18 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Context–Aware Service Discovering System for Nomad Users

Pathathai Na-Lumpoon*

*Corresponding author

Mu Lei
Isaac Caicedo-Castro[1]
Marie-Christine Fauvet
Ahmed Lbath

LIG Laboratory, MRIM Group,
University of Grenoble,
41 Rue des Mathématiques, Grenoble, Isere, 38041, France
Phone: +33476514473 E-mail: Firstname.Lastname@imag.fr
[1]University of Córdoba (Colombia) and National University of Colombia

Abstract: This paper presents an architecture for a system that provides nomad users, context-aware personalised services. Users might need any sort of services: information about the weather forecast for the next day, or about a museum in the neighbour worth to visit. These services are known as stateless services. More complex situations ocurre when services are stateful. Such services are, for example those which need users to be logged in (e.g. booking a room in a hotel). The question discussed in the text are those related to: i) user's privacy, ii) recommendation and discovery of services, iii) composition of recommended services into a composite service, and iv) execution of the resulting composite service.

Keywords: User privacy; Recommender systems; Web Service Composition; Semantic-based Orchestration; E-Tourism.

1 Introduction

With the development of pervasive computing and mobile technologies, mobile applications are getting more and more attention. These applications, running on mobile devices, are challenged to locate and deliver the right service to the right person, with the appropriate rendering.

The work reported in this paper is a part of a broader project which aims at designing and implementing a privacy-sensitive framework which provides context-aware personalised

Copyright © 200x Inderscience Enterprises Ltd.

services for nomad users according to their needs, profile and context (see Na Lumpoon et al. (2012)). The issues to be addressed in this project are related to system design, software architecture, distributed and heterogeneous resource access and integration, information retrieval and recommender systems. Specifically, the following research questions are raised: i) How to protect user's privacy using her profile and context? ii) How to provide highest quality recommendation of services to be composed? And iii) How to orchestrate and execute the resulting recommended composite service?

The rest of this paper is organised as follows: Section 3 sketches the architecture and components of the system proposed in this work. Section 2 presents two illustrating examples that motivate the project. Section 4 discusses users' privacy respect and management. Section 5 introduces basic concepts and design of the mechanism to retrieve services which fulfil users' needs. In Section 6 we show how to compose the recommended services, then how to orchestrate and execute the resulting composite service. Eventually, we conclude and propose some perspectives in Section 7.

2 Motivating examples

We illustrate our approach by the two following scenarios. Bob is from the UK, he likes and practices a lot of sports. He is travelling for work in Chiang Mai, Thailand. He is seeking for some tourism activities he can do during the next two days and which depend on the weather forecast. As, Bob didn't give neither where, nor when he wants to do these activities, this latter information will be taken from his query context (i.e. these values are given by the location and time sensors embedded in Bob's device, see Section 4).

According to his profile, preferences and privacy model (see Section 4) two services are found by the recommendation system (see Section 5): one, named WF, which exposes one operation op1.WS which returns the weather forecast depending on the given location and time, the other service, named AC, with one operation as well op1.AC which returns a list of activities, given a location and weather forecast. These two services are then composed in order to build a new *composite* service (see Section 6). In this example, when executing the composite service, first the operation exposed by WS is enacted, and second the operation exposed by AC. This execution is orchestrated by our system, and executed in the behalf of Bob, then its result is returned to Bob's mobile device.

3 Nuts and bolts of the system

According to the fact and problems stated in previous section, we propose the Context–Aware Services Discovering System for mobile users. Figure 1 depicts the components of this system, which are described as follows:

- *User Interaction and Query Management*: This component aligns the user's query to her profile and context taking into account her privacy model. If Bob does not want to disclose his location, when entering the query, so the location is kept unknown, and will be treated as a parameter of the resulting composite service which will be proposed to him.
- *User Management*: This component classifies the user into user profile group while protecting the user privacy. Section 4 presents a detailed description about this component.

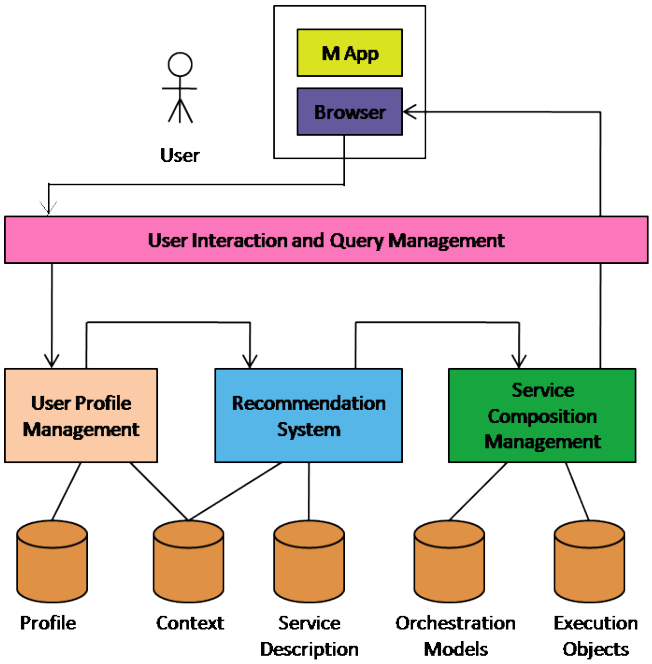

Figure 1 Components of the Context–Aware Services Discovering System for Nomad Users

- *Content–based Services Recommender System*: This component system returns the services whose execution fulfil the user's query. The output of this module is a list of web services, each of which tackling a part of the user query, but composed all together fulfilling it. The features of this component are detailed in Section 5.
- *Service Composition Management*: This component receives a set a services to be composed according to an abstract plan deduced from the user's query. Then the resulting composite service is executed on the behalf of the user (see Section 6).

4 User Management

According to Yang et al. (2005), as the foundation and core of personalised information service, the accuracy and effectiveness of the description of user preferences by user models, and the reasonableness of the recommendation algorithm, are directly related to the quality of the services.

As an information service system especially for mobile users, we take more advantage of context–based services, and also we need to pay more attention to adapt the several of different mobile devices.

Based on this consideration, we need to define the "user profile"and "context"during the research, in order to better adapt to the features of mobile devices. Regarding to our program, the better way to differentiate them, may be evaluating the usage of the descriptions in user profile and context.

In our architecture, there are 3 tables which contain user information, these ones shall be used to index the suitable services that fulfil the user queries, in order to execute the services

selected by the users, and to try to predict the potential better parameters for executing services for certain users. Firstly let's start with an example. Bob's profile is [gender: male, citizenship: UK, Occupation: Engineer, age: 28, favourite activities: hiking, museum], with his context: [Date: 25 June 2013, Local time: 12 pm, Location: 48° 52' 25.68" N, 2°17' 42" E].

The first table contains the record of Bob's user Identification (user ID) and his context: [1000, 25 June 2013, Local time: 12 pm, Location:48° 52' 25.68" N, 2°17' 42" E]. This record will be matched with the user ID that came alone with Bob's query, and sent to the system in order to retrieve in the service descriptions the available services.

The second table contains the record of Bob's personal information, [Bob, M, UK, Engineer, 28, hiking, museum]. This record will be sent along with the result list of services that fulfil Bob's query, to be part of the parameters needed to execute the service, and also be used to find the best user cluster for Bob (i.e. in which users have the most similar preferences to Bob). The third table contains the potential parameters to be used to execute the selected service for every cluster of users. It should be trained to optimise the parameters for any cluster of users. In order to provide personalised service suggestions, we approach to identify the K Nearest Neighbours (KNN) of the active user (users with similar interests), and calculate a prediction of the active user's potential parameters for a specific service. These parameters can be defined, for example, as the weighted average of the KNN's parameters.

To predict users' potential interest by his "nearest K neighbors", we propose the use of "Group User Profile", which is abstracted from a cluster of users with shorter distance to each other. While users are interacting with the system, their privacy has to be considered. Different users at different time and location might have different understanding or definition of their privacy, in order to meet these requirements; a component which is able to collect and summarise the "privacy rule"of user is proposed to be adopted before using any user information in the database, as shown in Figure 2.

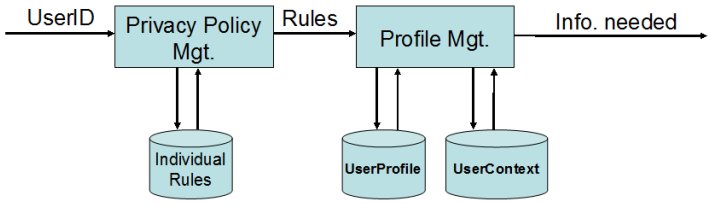

Figure 2 Process of getting individual privacy rules of user

In others words, an additional part will be added to the user's profile, which includes the description of her "Privacy".

While clustering, the profiles might be input by users directly or requested from other cooperators' database. For example, some user might login to the system using their Facebook account because they think Facebook will not leak and misuse their personal information. In order to protect the privacy of users who directly saved their profile in our system, we need a privacy sensitive distributed clustering method.

The Protecting might be done by both Data Encryption or Distance based clustering method. As we know, location is a very important parameter in mobile applications, and most of the tourism mobile applications are location–based. Let's take a look at the most–used method for mobile users to get their own location: Global Positioning System (GPS) with Assisted GPS (AGPS).

With GPS, mobile users could get their location by knowing the distance between the device and at least 3 of the 24 satellites working for the system. GPS information is considered the standard of geographical data. But in case we have a capture failure of the GPS satellites navigation signal, or a relatively longer time waiting for the response from the satellite since we were in some particular landscape such as very narrow streets, we will need AGPS module to help us. Mobile devices got their relatively accurate location from Location Provider, and request Location Based Services with these location information.

In other words, beside GPS, we will need to use AGPS module to contact to, which means we will have to send our local network information to the Location Provider, in order to get the right location information. And also we will have to contact to the Location Based Service Provider (LBSP) with the current location information.

In this case, to protect user's current location, we will have to consider both the communication between the mobile device and the Location Provider (LP), and also the communication between the mobile device and the LBSP.

For the LP, to minimum the communication between the mobile device and the LP is the only way. In case that we do not really need a very accurate location, a certain set of information needed by the Location Provider might not be sent, according to the user's preference. Or we can directly use WI–FI access to locate the user directly.

Anonymity and cloaking based methods will be useful for the LBSP. Firstly we will have to consider the detail of the data needed by a certain service, for example, a service for booking a room in hotel, usually asking only the approximate location of the user, like the name of the city. Or we can use an agent–based method in order to be anonymous.

5 Recommendation system

This system is in charge of retrieving Services whose description matches with the user's query. A Service description is given in its WSDL file (Web Service Description Language) document (Chinnici et al., 2007) . This kind of document is coded in a XML format, thereby, a WSDL document describes, in a textual format, a Service, its operations, and its parameters. To represent a WSDL document onto a vector space model is possible due to it is feasible to compute the relevance of each distinct word in this kind of document, also Hess et al. (2004) assert this assumption. Moreover, a part of a WSDL document could have documentation which is comments (i.e., unstructured text). Therefore, the aim consist of building an index with a corpus composed by WSDL documents onto a vector space model, which allow to retrieve a ranked list of Services, which is decreasing sorted in accord their similarity with the user's query in the same vector space.

Before to index WSDL documents is necessary a preprocessing procedure to extract the comments associated with each service, its operations and parameters. This procedure also splits service identifiers and operation names, which are combinations of more than two words, e.g., "*getWeatherForecast*"should be separated in the words: "*get*", "*weather*", and "*forecast*". Next, the procedure removes symbols and stop words (e.g., the, is, at, which, of, etcetera), and transforms all words to lower–case. Service identifiers, operation names,

and documentation are retrieved as an automatic part of the preprocessing step. This can be implemented by using Java API XML Processing (JAXP), which is an Application Programming Interface (API) for XML processing. Finally, vectors with the frequencies of each. Index terms for all text descriptions are built, in order to make a bag–of–words as a matrix of term frequencies by using Inverse Document Frequency (IDF) weighting to highlight the relevance of each word along the corpus. Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be the matrix of term frequencies, such that m is the number of text terms, and n is the number of WSDL documents in the corpus. This matrix provides information about the occurrence of each term for each WSDL document in the corpus.

After this preprocessing procedure, the next step is to build an index based on latent factors. In this work the purpose of use latent factor analysis is to explain these above mentioned occurrences by characterising the term frequencies by using a set of r factors inferred from occurrence patterns. This factors measure dimensions such as how expensive is to consume a Service, or how much the Service is related to sport tourism activities. This factors can measure less well defined dimensions, or other ones completely uninterpretable.

For every term, each factor measures how much this appears in WSDL documents related with corresponding factor. These factors form new meaningful dimensions to cluster WSDL documents.

Taking into account this model, text terms can be represented in a latent factor space, their relationships may be explained by using the inner product among their corresponding representations. Let $\{\mathbf{z}^t \in \mathbb{R}^r\}_{t=1}^n$ be the set of n column vectors of the matrix $\mathbf{Z} \in \mathbb{R}^{r \times n}$, which represent the set of n WSDL documents of the corpus. Let $\{\mathbf{w}^t \in \mathbb{R}^r\}_{t=1}^m$ be the set of m column vectors of the matrix $\mathbf{W} \in \mathbb{R}^{r \times m}$, which represent the set of m text terms. The components of the vector $\mathbf{z}^i$ measure the extent to which the WSDL document i expresses latent factors. On the other hand, the components of the vector $\mathbf{w}^j$ measure the extent to which the term j appears in the WSDL documents associated to the corresponding factors. Therefore, the problem consist of learning both vectors ($\mathbf{z}^i$ and $\mathbf{w}^j$), such that the resulting! dot product between these vector is approximated to the occurrence of the term i within the WSDL document j, as follows:

$$x_{ij} = (\mathbf{w}^j)^T \mathbf{z}^i = \sum_{k=1}^{r} w_k^j z_k^i \tag{1}$$

This can be written in matrix form as follows:

$$\mathbf{X} = \mathbf{W}^T \mathbf{Z} \tag{2}$$

In consequence, the most appropriated form to solve this problem is by means of *matrix factorisation* techniques, e.g., Singular Value Decomposition (Deerwester, 1988), Non-Negative Matrix Factorisation (Xu et al., 2003), or Probabilistic Matrix Factorisation (Hofmann, 1999; Blei et al., 2003). Theses techniques can be used to compute these matrices ($\mathbf{W}$ and $\mathbf{Z}$) out of the original bag–of–terms $\mathbf{X}$, i.e., to find a factorisation of $\mathbf{X}$ in terms of $\mathbf{W}$ and $\mathbf{Z}$.

The index is used to match similar contents given a particular user's query. Let $\mathbf{q} \in \mathbb{R}^m$ be the user's query in the vector space model. All vector in this space can be projected in the latent space by computing $\mathbf{z} \in \mathbb{R}^r$, such that $\mathbf{q} = \mathbf{W}^T \mathbf{z}$. Therefore $\mathbf{z}$ can be found with the expression 3 as follows:

$$\mathbf{z} = (\mathbf{W}^T)^{-1} \mathbf{q} \tag{3}$$

Once the user's query has been projected onto the latent space, it is necessary, using a similarity measure, to compute the similarity between vectors in the latent space. So, the expression 4 is used in order to compute this similarity between the user's query and the others WSDL documents indexed in the latent space.

$$\mathbf{s} = \mathbf{Z}^T \mathbf{z} \tag{4}$$

where $\mathbf{s} \in \mathbb{R}^n$ is a vector of similarity scores. Each component s_i of this vector is the ranking recommended for each service i, when the user expresses her (or his) needs through the query $\mathbf{q}$. In other words, $\mathbf{s}$ is the recommendation, whose services with greatest rank are the most recommended, and those with the lowest are the least recommended to fulfil the user's needs.

Finally, the services retrieved by the recommender system are filtered and orchestrated by the component described in the section 6.

6 Service Composition Management

The Service Composition Management (SCM in short) has the responsibility for automatically producing the synthesis of the composed service that is built as an orchestration of existing services. The goal of service orchestration is to invoke the existing relevant services according to the designed abstract plan. The SCM involves two processes as described in the following.

1. *Service selection process*: Many existing works use service description languages such as OWL-S (Burstein et al., 2004), WSML (de Bruijn et al., 2006), SAWSDL (Farrell and Lausen, 2007), USDL (Bansal et al., 2005) to represent both functional and non-functional aspects of web services and build up application to use these semantic service descriptions. We adopt SAWSDL as service description standard. Since SAWSDL has been proposed by the recommendation of W3C in 2007, many works have been used it widely. In addition, SAWSDL is a lightweight approach to annotate service descriptions. It uses external extended attributes called ModelReference to establish a relationship between WSDL components and concepts in another semantic model. The figure 3 shows WSDL components and the SAWSDL annotations that point to semantic concepts for specifying semantics or to schema mappings for data transformations (van Assem et al., 2006).

2. *Service composition process*: Dataflow control and artificial intelligence planning approaches are two candidate approaches to compose services. BPMN (Business Process Model and Notation) (see OMG (2011)) is the major language for modeling business processes in both industry and academic. As BPEL (Business Process Execution Language) (see Fu et al. (2004)) is getting less and less attention, we haven't considered it. However, dynamic service orchestration at runtime is a difficult problem and very complicated when the maintenance time arises. For AI planning technique, the goal and requirement definition have to be explicitly specified. Pre-conditions and effects of target services are used in composition of services. For example, Sirin et al. (2004) use the SHOP2 planner to build compositions of services described in DAML-S. Similarly, Klusch and Gerber (2006) use OWLS-XPlan to build composed service in OWL-S. However, these techniques focus on selecting the right services at run-time. However, they caannot dynamic re-plan a composition at run-time.

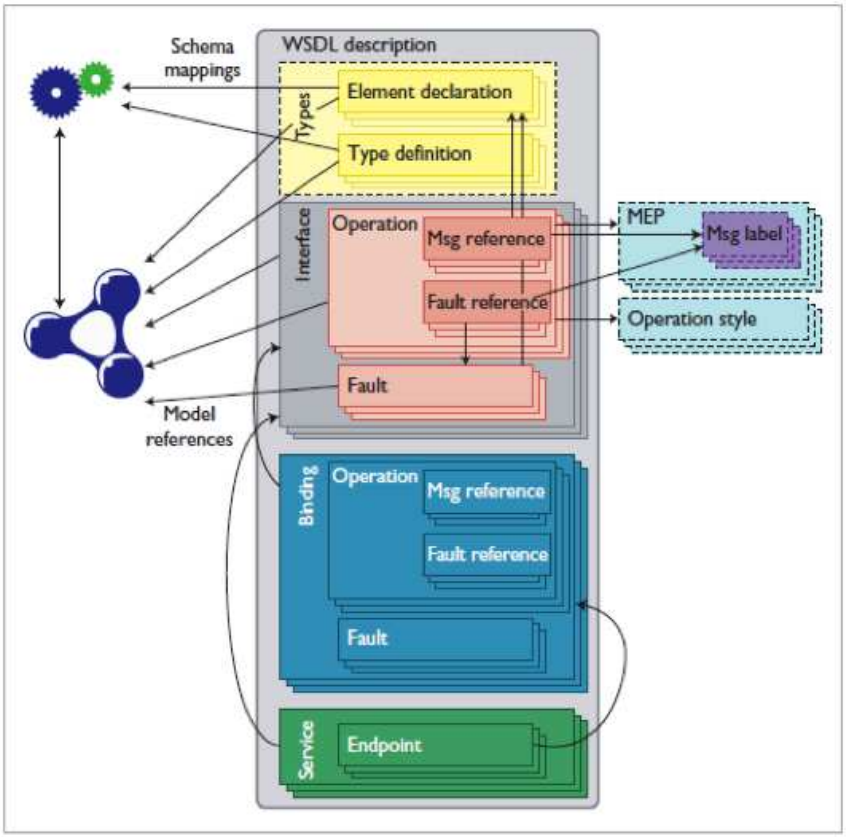

Figure 3 The Web Services Description Language (WSDL) with Semantic Annotations for WSDL and XML Schema (SAWSDL) (taken from Kopecky et al. (2007))

SCM (see Figure 4) is proposed to solve the problems stated earlier which are semantic schema matching and dynamic service composition at runtime. A key to success service composition is to match abstract service or requested service to existing concrete services. In our SCM, we use SAWSDL to represent abstract services and concrete services and apply logic reasoning using AI planning technique to orchestrate engine to orchestrate composed services. The components of our SCM approach to service orchestration are following.

- *Interpreter*: The Interpreter is the component which is responsible for retrieving the relevant abstract actions given the template id by the User Interface Management (UIM) component. The UIM component provides a template which is a workflow dictating the sequence of tasks to users. Afterward, the UIM passes user arguments to UP Management component and the template id to the Interpreter component. We propose tourist information tasks model to represent the orchestration of tourist task actions. Thus, the Interpreter looks up abstract actions given by template id in the Orchestration Model repository, and send them to the Planner Component.
- *Planner*: The Planner is the component which is responsible for synthesizing automatically an abstract plan. The abstract plan represents a sequence of abstract actions. In order to automatically build up an abstract plan, the goal and initial states

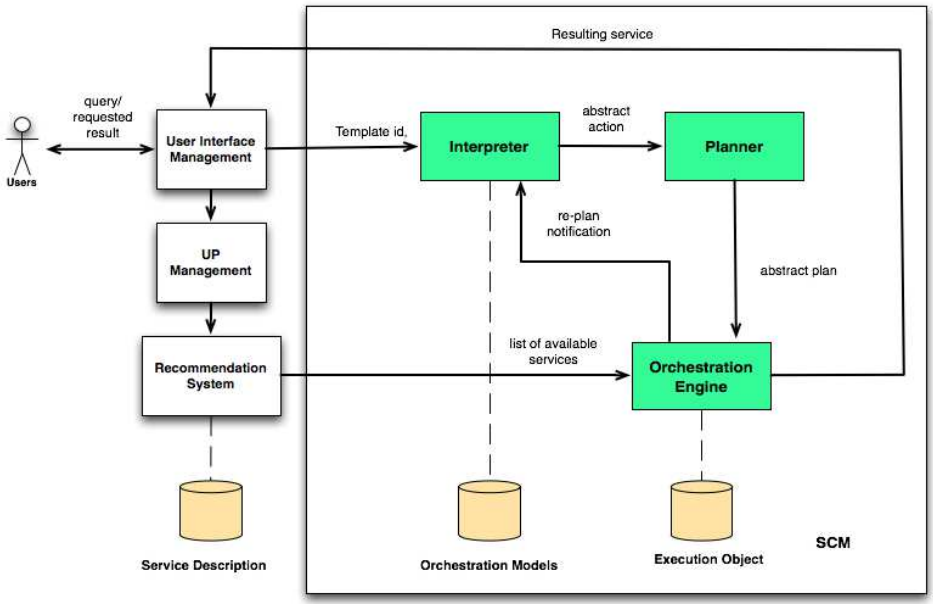

Figure 4 SCM approach to service orchestration

of abstract plan and pre-conditions and post-conditions of abstract actions have to be well defined. The Planner uses AI reasoning logic technique to approach the abstract plan. The Planner can start from the goal of the plan and go backward to the initial state and *vice-versa*.

- *Orchestration Engine*: The Orchestration Engine is the component which is responsible for synthesizing the composed service according to the received abstract plan. The Orchestration Engine will match each abstract action in the abstract plan with concrete service in the list of available services received from the Recommendation System. Afterwards, the Orchestration Engine will develop business process orchestration model and bind the model with the endpoint web services. Finally, the composed web service with the user parameters are ready to be executed. The parameters used in the service execution process are either from user specifies by himself or user preference in the user profile. The resulting service will be executed on the behalf of the user. However, we add an event handler to Orchestration Engine for monitoring unexpected situations which might happen at runtime. For Instance, when the Orchestration Engine receives a notification that a chosen concrete service is no longer available, the event handler will send this message to the Interpreter to re-plan the abstract plan. Our approach is similar to Cugola et al. (2012) who used descriptive language to self-adaptive service orchestration.

However, this approach needs the help from domain expert to set the abstract actions, which are used as the building blocks of orchestration plans, goal and initial state.

To illustrate the orchestration technique, we follow the query, from example, Bob wants to search for tourism activities in Chiang Mai in the next 2 days according to the weather forecast and his preferences. The template and user's context are used to fulfil all required

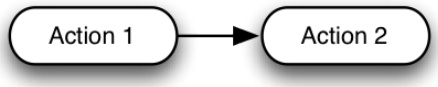

Figure 5 Abstract plan for execution

parameters, which are today date and Bob's location and travel preferences respectively. The query with all parameters are sent to the Interpreter to extract abstract actions, which are:

```
Abstract action 1: Find weather forecast
  Input: location, date(s)
  Output: weather forecast
Abstract action 2: Find travel activities
  Input: criteria, weather
  Output: travel activities result
```

The parameters are kept in object variables, which are used later at service execution time. The discovered abstract actions are sent to the Planner to generate the abstract plan using graph plan. The planner analyses the input and output of each abstract action. From Bob's example, one of inputs of abstract action 2 is weather, which is output of abstract action 1. So the Planner synthesises the graph plan like:

The abstract plan is sent back to the Orchestration Engine to match each abstract action to the concrete service and retrieve all executed parameters in order to produce the composed service. To execute the composed service, the web application is built up to display the resulting service to user.

7 Conclusion and Perspectives

The contribution of this work is the proposal of a Context–Aware Services Discovering System for Nomad Users which allows reuse of (web) services, automatic orchestration, composition and discovery of services (stateless and state–full) taking into account the profile, context, and explicit user's needs. In the design principles, the user privacy has been considered from inception of the design system. This is a very important contribution of this work, because the system cannot keep protected the user information from unauthorised access.

The perspectives for further work are: i) To integrate machine learning techniques in order to classify, cluster, and associate Web Services. ii) To assess if the learning of features algorithms based on matrix factorisation can infer import an Web service's features as quality of services. iii) To determinate or propose the best matrix factorisation technique to implement the Filtering Collaborative approach proposed in this work. iv) To evaluate the system proposed in this paper by means experiments performed over its implementation.

Acknowledgement

Pathathai Na-Lumpoon and Mu Lei are PhD students supported by European Union's Erasmus Mundus programme. Isaac Caicedo-Castro is a PhD student supported by

COLCIENCIAS–COLFUTURO (convocatoria 528 de 2011) scholarship, and he is also founded by the University of Córdoba (Colombia). Authors thank the anonymous reviewers for their relevant comments and their help to improve the quality of this paper.

References

Bansal, A., Kona, S., Simon, L., and Hite, T. D. (2005). A universal service-semantics description language. In *Proceedings of the Third European Conference on Web Services*, ECOWS '05, pages 214–, Washington, DC, USA. IEEE Computer Society.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Burstein, M., Hobbs, J., Lassila, O., Mcdermott, D., Mcilraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., and Sycara, K. (2004). OWL-S: Semantic Markup for Web Services.

Chinnici, R., Moreau, J.-J., Ryman, A., and Weerawarana, S. (2007). Web services description language (WSDL) version 2.0 part 1: Core language. World Wide Web Consortium, Recommendation REC-wsdl20-20070626.

Cugola, G., Ghezzi, C., and Pinto, L. (2012). Dsol: a declarative approach to self-adaptive service orchestrations. *Computing*, 94(7):579–617.

de Bruijn, J., Lausen, H., Polleres, A., and Fensel, D. (2006). The web service modeling language wsml: an overview. In *Proceedings of the 3rd European conference on The Semantic Web: research and applications*, ESWC'06, pages 590–604, Berlin, Heidelberg. Springer-Verlag.

Deerwester, S. (1988). Improving Information Retrieval with Latent Semantic Indexing. In Borgman, C. L. and Pai, E. Y. H., editors, *Proceedings of the 51st ASIS Annual Meeting (ASIS '88)*, volume 25, Atlanta, Georgia. American Society for Information Science.

Farrell, J. and Lausen, H. (2007). Semantic annotations for WSDL and XML schema. W3C working draft, World Wide Web Consortium.

Fu, X., Bultan, T., and Su, J. (2004). Analysis of interacting bpel web services. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 621–630, New York, NY, USA. ACM.

Hess, A., Johnston, E., and Kushmerick, N. (2004). ASSAM: A tool for semi-automatically annotating web services with semantic metadata. In *Proc. Intl. Semantic Web Conference (ISWC 2004)*. Springer.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA. ACM.

Klusch, M. and Gerber, A. (2006). Evaluation of service composition planning with owls-xplan. In *Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, WI-IATW '06, pages 117–120, Washington, DC, USA. IEEE Computer Society.

Kopecky, J., Vitvar, T., Bournez, C., and Farrell, J. (2007). Sawsdl: Semantic annotations for wsdl and xml schema. *Internet Computing, IEEE*, 11(6):60–67.

Na Lumpoon, P., Lei, M., Kamnardsiri, T., Lbath, A., and Fauvet, M.-C. (2012). Illustrating some issues raised when designing context-aware personalized services for mobile users. In *Proc. of Asia-Euro conference on multidisciplinary approaches to tourism, hospitality & gastronomy research.*, Chiang Mai (Malaysia).

OMG (2011). Business Process Model and Notation (BPMN), Version 2.0.

Sirin, E., Parsia, B., Wu, D., Hendler, J., and Nau, D. (2004). {HTN} planning for web service composition using {SHOP2}. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(4):377 – 396. International Semantic Web Conference 2003.

van Assem, M., Gangemi, A., and Schreiber, G. (2006). Rdf/owl representation of wordnet. World Wide Web Consortium, Working Draft WD-wordnet-rdf-20060619.

Xu, W., Liu, X., and Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 267–273, New York, NY, USA. ACM.

Yang, Y., Li, J., and Gao, H. (2005). Ontology-based preference model in digital library. *Journal of Software*, 16(12).