Natural Language Based User Interface for On-Demand Service Composition

Marcel Cremene, Florin-Claudiu Pop, Stéphane Lavirotte, Jean-Yves Tigli

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2009
Cluj-Napoca (Romania), July 2–4, 2009, pp. 25–28

NATURAL LANGUAGE BASED USER INTERFACE FOR ON-DEMAND SERVICE COMPOSITION

MARCEL CREMENE[(1)], FLORIN CLAUDIU POP[(2)], STEPHANE LAVIROTTE[(3)], AND JEAN-YVES TIGLI[(4)]

Abstract. The aim of this research is related to services on-demand creation, by assembling other existent services, using a natural language based interface. The original aspect of our approach is the use of a joint approach: semantic-based (connects the natural language with the concepts associated with the services) and pattern-based (insures that the resulted services are always reliable). The application field is related to intelligent buildings.

1. Introduction and problem statement

Service composition is a mean to create new services by composing existent services. Usually, the service composition task is solved by a human expert because it requires an understanding about the services semantics. This means that, the creation of a new service requires usually the human expert intervention, either before, by prediction, or after the user requirement was expressed. A priori service creation should be based on the prediction about all possible user request but such an approach is practically impossible. Even if the human expert will try to predict only the most probable user requests, this task will be an extremely costly one. A posteriori service composition by the human expert, after the user request is known, requires an important time, costs, and usually is not an option because the user wants the service as soon as his request was expressed.

The objective of this paper is propose such a system, that is able to create services on-demand, starting from the natural language based user request and assembling existent services according to this request. In particular, our application field concerns the intelligent environments. The services are offered, in this case, mainly by the various devices spread in the intelligent space (ex. a building) but we can also use remote web services.

2000 *Mathematics Subject Classification.* 68T50 Natural language processing.
Key words and phrases. Natural Language, User Interface, Service Composition.

2. Proposed solution

The proposed system is called Natural Language Service Composer(or NLSC). System architecture is depicted in Figure 1. This architecture is composed by two main parts: a) the Natural Language Processor (NLP), which is composed by a set of tools necessary for user request analyze and b) the Service Composer (SC). The Natural Language Processor transforms the user request into a machine readable, formal, request. This formal request will be used as input for the Service Composer. An existent English dictionary, WordNet[3], is used instead of creating our particular ontology. The Service Composer is based on a middleware platform called WComp [2]. This platform is targeted mainly for intelligent environment applications. WComp was designed for supporting dynamic assembling of services provided by hardware devices. Web services and UPnP services in general may be used through this platform also. The AoA (Aspects of Assembly) mechanism that comes with WComp allows the developer to create composition patterns and use them at runtime in order to modify the service architecture. The formal request (the NLP output) will be used in order to select the services and also the AoA patterns. Once we have selected the services and the patterns, the WComp platform is able to create almost instantly the new, composite service. Also, WComp platforms is responsible for devices/services discovery and dynamic architectural reconfiguration support. Contrarily to other pattern-based approaches, AoA patterns can be combined, superposed. Thus, a large number of valid combinations (services) may be created.

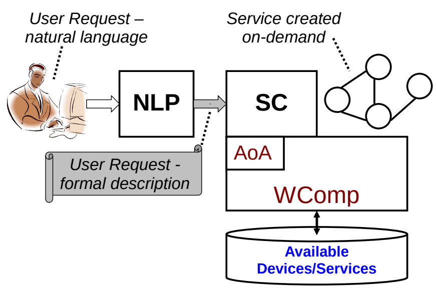

Figure 1. NLSC system architecture

The NLP input is a sentence that is either, written by the user or obtained from an existent voice recognition system. The sentence is decomposed in a word collection. The link words are eliminated. In order to do that, we use a list of link words. For each word, we apply a stemming procedure: the verbs are passed to infinitive, the nouns are passed to single form. Finally, we obtain a list of words that we will call further "request concepts". The semantic matching is based on what we have defined as the "conceptual distance". The conceptual distance is a measure of the similarity between two concepts. A specialized dictionary called WordNet [3] was use on this purpose.

In order to compute the semantic matching, we use a representation called "conceptual graph". The conceptual graph has as nodes the user concepts and the service concepts. The arcs connect each user concept to each service concept. The weight of each arc represents the *conceptual distance* between the concepts. In order to select the service concepts that are similar to the user concepts, we need to apply the following two transformations to the conceptual graph: a) find the minimum distance path in the graph (include all nodes) using the Kruskal algorithm that calculates the minimum spanning tree (MST); after this transformation each service description will be connected to 2 text segments; and b) for each triplet (service concept, user concept 1, user concept 2) keep the arc that has the minimum weight. Thus, we obtain a concepts pair.

Once the services are selected according to the concepts pairs, we apply all the AoA patterns that correspond to the selected services. At this moment, only the service selection was implemented. The AoA selection (based on the user request) remains as further work.

3. Implementation

Scenario. The user request is: *"I want to use my phone to turn off the light, turn on the TV and play some music on HiFi"*. All the relevant services are identified and then composed, by applying the AoA composition patterns. The system will select, among all available service, only the services that are matching the concepts from user request. Once the services are selected, the system creates the service architecture by applying the AoA patterns. The result is depicted in the figure 2. The implementation is based on the WComp platform discussed before.

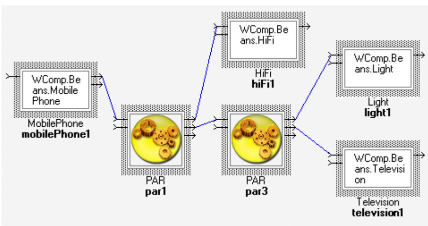

Figure 2. The dynamically composed service for Scenario 1

4. Conclusions

The original aspect of our proposal, compared with the related work (described in [1], [4] and [5] but not detailed here because of the very limited space) is the mixed approach: semantic and pattern-based. This approach combines the advantages of the both: thanks to composition patterns, it allows us to build complex composite services, that are always valid and functional. With other approaches (interface, logic, semantic based), that are not using patterns/templates, it is very difficult to create complex architectures that are are valid and work correctly. In particular, AoA patterns can be composed and this helps us to overcome the limitations of the traditional patern-based approach (not very flexible).

In the same time, the service composition is user driven, by natural language (voice) and allows the user to get the service he want on-demand. From this point of view, our solution is less restrictive than the other solutions.

Another important advantage of our solution is the reuse of WordNet [3] free dictionary, that is acting like a huge ontology. Due to this, we can relax very much the limitations for the natural language, imposed by solutions where an ontology (usually restricted) must be created by the developer. Otherwise, the creation of a rich ontology is a very costly task and our solution succeeded to avoid it. Thus, for describing the services, their developer just need to use a correct English.

References

[1] Bosca, A.; Corno, F.; Valetto, G.; Maglione, R., *On-the-fly Construction of Web services Compositions from Natural Language Requests*, JOURNAL OF SOFTWARE (JSW), ISSN : 1796-217X, Vol. 1 Issue 1, pag 53-63, July 2006.

[2] Cheung-Foo-Wo, D.; Tigli, J.-Y.; Lavirotte, S.; Riveill, M., *Self-adaptation of event driven component-oriented Middleware using Aspects of Assembly*, In 5th International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC), California, USA, Nov 2007.

[3] Cognitive Science Laboratory, Princeton University, *WordNet a lexical database for the English language*, Website, 2006 `http://wordnet.princeton.edu/`

[4] Fujii, K.; Suda, T., *Semantics-based dynamic service composition*, IEEE Journal on Selected Areas in Communications, Vol 23(12), pag 2361- 2372, Dec 2005.

[5] Larvet, P., *Web service with associated lexical tree*, European Patent, EP1835417.

(1) Technical University of Cluj-Napoca, str. Baritiu, no. 26, phone: 0264-401813
E-mail address: cremene@com.utcluj.ro

(1) Technical University of Cluj-Napoca
E-mail address: florin.pop@com.utcluj.ro

(3) University of Nice Sophia-Antipolis
E-mail address: stephane@lavirotte.com

(3) University of Nice Sophia-Antipolis
E-mail address: tigli@polytech.unice.fr