Context Aware Service Composition

Maja Vukovic and Peter Robinson

University of Cambridge Computer Laboratory
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK

E-mail: {firstname.lastname}@cl.cam.ac.uk

Abstract

A major challenge in building context aware applications is the large variety of contextual types and the range of their values. The proposed approach employs AI planning technology to assemble context aware applications as dynamically composed sequences of calls to Web services. Contextual changes may trigger further recomposition during the execution of the services, causing the application to evolve dynamically. We present initial results in using existing planning technology for the selection of the required services and the sequence of their execution.

1. Introduction

The development of context-aware applications has become a complex task due to the need to accommodate for the potentially vast variety of – possibly even unanticipated – context types and their values that may be encountered. A commonly followed approach has been to simply hard-code mappings between all possible combinations of context values and the corresponding application behavior. This is not only impractical, but also makes such systems difficult to later extend to take into account new values of existing context attributes and new context types.

To address this problem we proposed building context aware applications as dynamically composed sequences of calls to Web services, using AI planning technology. Initial work on this has been presented in [5]. Different service compositions of such sequences will result from different contexts such as resources available, time constraints, user location requirements and profile. Further recomposition of the service during its execution may be triggered by the change in context (values and types), causing the application to evolve dynamically. The proposed architecture, shown in Figure 1 is described in detail in [4].

![Figure 1. Overview of entities and interactions in the proposed architecture.](image)

2. AI Planning for Context Aware Service Composition

Planning technology offers a solution to Web service composition because the possible actions can be made explicit and their preconditions and effects can be specified using expressive representation formalisms. The planning goals
(requests for composite services) are descriptions of the goal state(s) to be reached. For example, user, while driving, requests a route plan in a foreign city. This may map to the following goal description

\[(\text{route\_plan\_found}) (\text{route\_plan\_speech\_out})\].

The knowledge about existing Web services, initial and goal state(s) is used to devise a composite service.

The initial system deployment indicated that the process of Web service composition requires modelling a number of sophisticated features, including concurrently executing actions, sensing, uncertainty, and resource constraints. We now present two further challenges for successful planning-based, context aware, Web service composition.

2.1 Support for context dependent goals

Returning to the above example, as the user gets out of the car, he requests a further route to the local restaurant. In this case the literal \((\text{route\_plan\_speech\_out})\) will be removed from the goal description, as the user is no longer driving and thus may not require speech interface, assuming he is using his PDA out in the street. The literals used to describe the goal depend on the context, whilst the user’s high level goal is in both cases \(\text{get route plan}\).

At present the system associates goal descriptions with specific contexts. We foresee two possible solutions to supporting context-dependent goals: (1) precompiling different subsets of services for different (sub)goals and (2) imposing constraints on the preconditions of the service.

2.2 Handling plan failures

Failures may occur both during the plan generation and plan execution stages. In our work we focus on failures arising during the execution of the plan. Planned services will be executed in unpredictable and dynamic environments, thus it is anticipated that plans may fail.

To handle plan failures we use monitoring procedures, as proposed by Haigh et al. [1]. Monitoring procedures determine: (1) What to monitor (service or event), (2) Methods that can be used to monitor it and (3) Some common recovery procedures (for service monitors).

3. Research context

Saif et. al [3] proposed an approach to dynamic assembly of \textit{Pebbles}, platform-independent distributed software components, based on the GOALS planning mechanism in response to evolving system requirements in pervasive environments. This architecture is based on the notion of generic plan customization, rather than on-line planning.

Planning approach to Web service composition has recently gained much attention to support enterprise application integration as Koehler et al. analyzed [2]. Previous research projects have addressed dynamism in the composition, primarily from the perspective of unavailability of selected Web services.

4. Conclusion

We are tackling the increasing complexity required for context awareness by building context aware applications through the dynamic composition of Web services\footnote{This work has been supported by IBM Zurich Research Laboratory.}. We employ planning technology to select the required services and the sequence of their execution. In this paper, we discussed practical challenges in using classical planning technology for context aware service composition. Our future work will involve implementation of a method for dealing with plan failures and addressing scalability issues.

REFERENCES


