Context-Sensitive User Interfaces for Semantic Services

WANITA SHERCHAN, Monash University
SURYA NEPAL, CSIRO ICT Centre
ATHMAN BOUGUETTAYA, RMIT University
SHIPING CHEN, CSIRO ICT Centre

Service-centric solutions usually require rich context to fully deliver and better reflect on the underlying applications. We present a novel use of context in the form of customized user interface services with the concept of User Interface as a Service (UIaaS). UIaaS takes user profiles as input to generate context-aware interface services. Such interface services can be used as context to augment semantic services with contextual information leading to UIaaS as a Context (UIaaSaC). The added serendipitous benefit of the proposed concept is that the composition of a customized user interface with the requested service is performed by the service composition engine, as is the case with any other services. We use a special-purpose language (called User Interface Description Language (UIDL)) to model and realize user interfaces as services. We use a real-life e-government application, human services delivery for the citizens, as a proof-of-concept. We also present a comprehensive evaluation of the proposed approach using a functional evaluation and a nonfunctional evaluation consisting of an end user usability test and expert usability reviews.

Categories and Subject Descriptors: H.5.2 [User Interfaces]: User-Centered Design
General Terms: Design, Human Factors, Management
Additional Key Words and Phrases: Semantic service, user interface, context-aware

ACM Reference Format:
DOI = 10.1145/2078316.2078322 http://doi.acm.org/10.1145/2078316.2078322

1. INTRODUCTION

Context is typically defined as a user's surrounding characterized by attributes such as location, environment, identity, date and time, season and temperature [Schilit et al. 1994; Brown et al. 1997; Ryan et al. 1997]. Other works have included the cognitive aspects of the user to define context in terms of the user’s physical, emotional, social and informational state [Dey et al. 1998]. The definition of context relevant to this article is more in line with Abowd et al. [1999], in which context is considered to be any information that can be used to characterize the situation of an entity relevant to the interaction between a user and an application. If we apply this definition to services computing, context can be interpreted as the information that characterizes the operation of a service, specifically, information relevant to the interaction between a
user and the service. Services typically represent complex real-life entities. Therefore, contextual information is paramount in interpreting and representing service output to users. Web services are the technology of choice for realizing service computing [Curbera et al. 2002]. Services are typically delivered through the Web; therefore, interfaces are a significant part of the interaction between the user and the Web service. As a result, information pertaining to user interfaces, such as user profiles, is a source of context in addition to the typical user location, identity, time, etc.

A Web service is considered to be context-aware if it has the ability to detect and respond to changes in the environment [Maamar et al. 2006]. For example, a flight booking service is context-aware if it can adapt the currency of the ticket price to the location of the client. Context-aware Web services should be easily accessible and adaptable so that they have potential to enrich user experiences and make daily life more productive, convenient, and enjoyable. Such a context-aware perspective also brings new challenges in the architecture, design, and implementation of existing Web services’ infrastructures and applications. Some questions that need to be addressed are how to deal with large amounts of services based on a variety of users’ personalized needs; how to handle context-aware service composition in a dynamic environment; and how to customize business processes according to users’ preferences.

**Issues.** In many applications, such as e-government services, user interfaces are considered to be among the most important contextual information for the provisioning of services. In that respect, the use of appropriate user interfaces is an important part of making government services more easily and effectively accessible to a wider range of constituencies, including those with little computing skills, physical and mental impairments, and young people who are tuned into the latest in social network technologies. Therefore, user interfaces are key to providing rich context to many e-government services. Research in accessibility and usability has traditionally focused on typical users and domain specific applications. It is assumed that users’ abilities do not change over time, and a system is designed for a specific application. The accessibility research has mostly focused on webpage content and how it can be made more accessible using features like text-to-speech, page magnification, enlarging menus and scroll bars, colour and font size changes, and keyboard-based navigation. The techniques for adaptive user interface usually depend on a user selecting the most appropriate options for layout. The challenge is how to provide an adaptive user interface in a way that is transparent to the user. Designing a good interface in general is challenging. A new approach is needed that will provide mechanisms to adapt to both short-term and long-term changes in user’s abilities, as well as the requirements of different services accessed by the user.

**Our Approach.** To address the aforementioned issues, we propose an approach to design *User Interface as a Service* (UIaaS) to create interfaces that automatically adapt to the conditions of the users as well as the applications. The UIaaS leverages users’ context information to allow for easy access to services while adapting the presentation medium (i.e., user interface (UI)) to suit their context. The UIaaS takes two types of input—the user’s context from the user profile, and the service context from the service being provisioned—to generate the UI, which is itself a service. This considers user interface as a service with context as input, enabling the composition of UI services, assignment of quality attributes, and querying them based on the contextual information. This is a novel view of context which leverages and extends our previous work on aspect-oriented approach, weaving context and services [Li et al. 2009]. In our current proposal, we consider the User Interface as a Service, which is part of the service ecosystem. Since the UIaaS already has contextual information embedded in it, it can also serve as a source of context for service composition leading to UIaaS as a Context