Software Research Lab
Institut fr Computerwissenschaften
Universitt Salzburg
Jakob-Haringer-Str. 2
A-5020 Salzburg
Austria

User Interface Tailoring for Multi-Platform Service Access

Guido Menkhaus Guido.Menkhaus@uni-konstanz.de
Wolfgang Pree pree@acm.org

TR-C047 March 29, 2002

Due to the diversity of display capabilities and input devices, mobile computing gadgets have caused a dramatic increase in the development effort of interactive services. User interface (UI) tailoring and multi platform access represent two promising concepts for coping with this challenge. The paper presents the MUSA (multiple user interfaces, single application) prototype system that addresses both issues by introducing an event-graph as basis of a UI tailoring process.

Keywords: User interface adaptation, multi-platform support

User Interface Tailoring for Multi-Platform Service Access

Guido Menkhaus and Wolfgang Pree
Software Research Lab, University of Constance, D-78457 Constance, Germany
firstName.lastName@uni-konstanz.de

Abstract

Due to the diversity of display capabilities and input devices, mobile computing gadgets have caused a dramatic increase in the development effort of interactive services. User interface (UI) tailoring and multi platform access represent two promising concepts for coping with this challenge. The paper presents the MUSA (multiple user interfaces, single application) prototype system that addresses both issues by introducing an event-graph as basis of a UI tailoring process.

Keywords

User interface adaptation, multi-platform support

INTRODUCTION

The current trend of Web access and computing is drifting away from the desktop PC as the principal device to access services and information on the Internet to consumer devices such as mobile phones and handheld computers. The new variety of devices has a profound impact on the way UIs of Web-based services are built. The aim is to avoid a fragmentation of the Web space into spaces that are solely accessible with specific types of devices.

The paper presents one solution that helps to avoid this fragmentation. At the core is what we call the event structure code (ESC). ESC is an aggregation adaptation technique that adapts dynamically the presentation model of an UI, guided by annotations that are provided by the UI designer. The ESC adaptation process is based on event graph XML (EG-XML). EG-XML is our description language that allows Web service designers to describe dynamic interactive services in abstract and generic terms. ESC and EG-XML are part of the adaptive UI system MUSA (multiple UI, single application), which adapts the presentation structure of its UI dynamically to different contexts [2].

MOTIVATION

Being aware of the necessity of adaptive UIs the MUSA project concentrates on two activities: Multi-platform support and UI tailoring.

- **Multi-platform Support.** We argue that the introduction of an abstract description of a UI is an essential component that eases the development of UIs for a variety of computing devices.
- **UI Tailoring.** UI tailoring refers to the ability of a system to adapt to the context in which it is used [3]. Tailoring of the interface design includes the capability of adaptation of content delivery to various devices, while preserving consistency and usability of the service. The MUSA system with the description language EG-XML and the ESC process of dynamic UI adaptation allows users of mobile computing devices to access a service while the transferred content is tailored to the capabilities of the user device with respect to the display size and mapped to the UI language that corresponds to the device.

MUSA ARCHITECTURE

Figure 1 illustrates the high-level architecture of MUSA. MUSA is conceptually split into four tiers and employs an event-driven design.

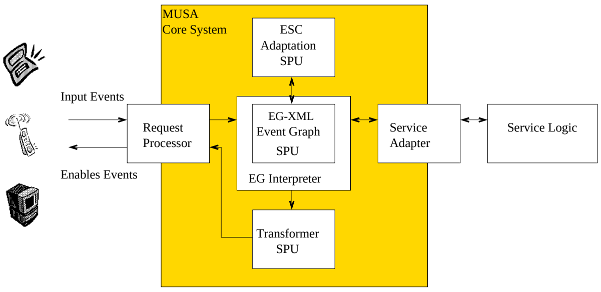

Figure 1: MUSA Architecture.

The client environment represents the first tier. No service data is installed on the client side and the client communicates via wireline or wireless Internet with the service. The request processor deals with the client's request. The communication between the services and the client UI passes through the request processor. The event graph interpreter mediates between the user interaction and the delivery of service data to the user UI through the service adapter. The event graph interpreter contains the event graph specialized processing unit (SPU), which manages the event graph, written in EG-XML, and handles the processing of elements of the event graph. The event graph implements the navigational design and the interaction of the service logic. It abstracts these aspects from the service logic. The ESC adaptation SPU adapts the presentation model by dynamically de-

IUI'02, January 13-16, 2002, San Francisco, California, USA.

signing the event graph subject to the device that connects to the service via the MUSA system. The transformer SPU assigns and maps the elements of the event graph to concrete UI objects, which trigger events that are associated with these elements. The service logic is the body of code for which the MUSA system provides the multi-platform and adaptation features.

EVENT GRAPH

The event graph (EG-XML) is an abstract description of a service, which is presented to the user who interacts with it through a UI. Elements in the event graph do not specifically define UI objects. However, they are assigned and eventually mapped to UI objects that are able to trigger the associated events. The UI objects trigger an event either when the UI objects are displayed or in response to user interaction. The event graph is designed with the following concepts:

- **Events.** An event is assigned to a concrete interaction object from the presentational and the behavioral point of view.
- **Dialoglet.** A dialoglet consists of a number of events, which belong logically to a specific subtask.
- **Dialog Profile.** A dialog profile represents a dialog adapted to a specific device profile.
- **Dialog.** A dialog consists of a set of dialoglets and is designed to represent a task or a subtask of a specific Web-based service.
- **Service.** A service is composed of a sequence of dialogs.

EVENT STRUCTURE CODE

The ESC adaptation can be classified as aggregation adaptation [1]. ESC adaptation creates dynamically a presentation model for a UI by constructing a dialog profile and clustering events into a set of dialoglets, which form the resulting dialog profile. Instead of clustering events of a dialoglet into a difficult to navigate, linked list of small dialoglets (Figure 2), the ESC builds a hierarchical structure (Figure 3).

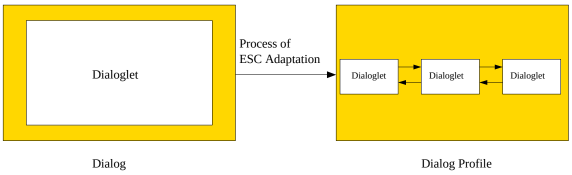

Figure 2: Linear structure of dialoglets.

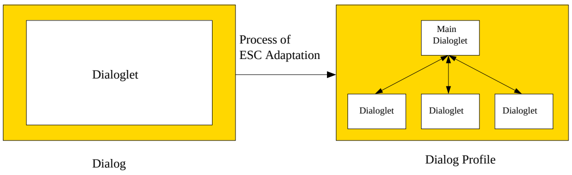

Figure 3: ESC hierarchical structure of dialoglets.

The process is guided by annotations provided by the EG-XML designer. The ESC adaptation consists of the building and the construction phase:

Building Phase: The building phase consists of three steps:

- **Clustering.** Event clusters are clustered into a hierarchically linked structure. At the beginning of the clustering process, each event cluster contains a single event.
- **Separation.** Event clusters that cannot be clustered are separated.
- **Linking.** Distinct event clusters that belong logically together are mutually linked. This linking phase creates additional main dialoglets (Figure 3), which are the building blocks for the hierarchical structure. The linking step occurs concurrently with the clustering and separation step.

Construction: The final step of the clustering process consists of transforming event clusters into dialoglets and the creation of a dialog profile.

The ESC is a technique that builds pyramidal structures by clustering event clusters. If a new event cluster that has tentatively been built does not satisfy a specific clustering criteria, it is separated into two event clusters. The clustering and separation process depend on the device profile and is different for each class of device. The result of the clustering process is a presentation model that is individually adapted to a device. In contrast to model-based approaches, the designer does not determine the presentation model of a UI for different platforms during design time, nor does the designer explicitly need to represent possible adaptations in the presentation modle. Rather the presentation model is dynamically created at run-time.

CONCLUDING REMARKS

MUSA supports multi platform service access and UI tailoring. The ESC adaptation is based on EG-XML, an abstract description language. The ESC adaptation process consists of the building and the construction phases. The building phase forms a hierarchical structure by clustering, separating and linking event clusters. The construction phase transforms the resulting event cluster into a dialog profile. A set of dialoglets presents a dialog, specific to a device profile. The ESC adaptation process builds dynamically a presentation model of a dialog.

REFERENCES

1. D. C. Bulterman. User-centered abstractions for adaptive hypermedia presentations. In *Proceedings of the sixth ACM interbational conference of Multimedia*, pages 247–256, Bristol, United Kingdom, September 1998.

2. G. Menkhaus. Architecture for client-independent web applications. In *Tools Europe*, pages 32–40, Zuerich, Switzerland, March 2001.

3. P. Szekely. Retrospective and challenges for model-based interface development. In F. Bodart and J. Vanderdonckt, editors, *Design, Specification and Verification of Interactive Systems '96*, pages 1–27, Wien, 1996. Springer-Verlag.