Integrating User Operations in Multichannel Hypermedia

Franca Garzotto, Vito Perrone
HOC (Hypermedia Open Center) - Politecnico di Milano (Italy)
Via Ponzio 34/5 - 20133 Milano, Tel. +39 02 2399 3520
Garzotto|Perrone@elet.polimi.it

ABSTRACT

Web Applications are progressively becoming *multi-channel* and *cross-channel*. The "same" service should be made available in different delivery environments and devices. A user may invoke a service on one device, suspend it, and complete its execution in another one. In this paper we provide the reader with the main concepts and innovative aspects of MC2 a design framework for specifying Multi/Cross Channel web application services. MC2 adopts an high-level, end user perspective and exploits the notion of *context*, to characterize who, where and how an operation can be invoked.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation (e.g., HCI)**]: Hypertext/Hypermedia – *navigation*, *user issues.*

General Terms

Design, Human Factors.

Keywords

Conceptual modelling, multi-channel, services on the Web, Web operations, context, UML, OCL

1. INTRODUCTION

Modern web applications differ from original web sites from a number of aspects. In the past, we had read-only, mono-channel web sites. Today, web applications provide functionality beyond search and navigation, offering a variety of *services* that allow users modify the application state. Moreover they are progressively becoming *multi-channel* and *cross-channel* [1]. As a consequence, the design of web applications is becoming more and more complex, and new models and methods must be explored to support the design activities. Design can be tackled at several levels of abstraction and from different perspectives. In this paper, we discuss MC2, a framework for conceptual modelling of Multi/Cross Channel web application services, that addresses design from a high-level, *end user perspective*.

The key aspects of MC2 are the following:

- We take into account the intrinsic *hypermedia* nature of services on the web. Users perceive the application as a hypertext network where navigation is the primary interaction paradigm. Services can be invoked only if the user is located in specific points of the hyperspace; users may navigate while using a service; user operations required by a service may have navigational effects, i.e., they may change the user position in the hyperspace.
- We decompose *services* into *"elementary" user interactions* that can be either operations or navigation steps and can be executed on a single device or on multiple devices, and may produce effects, which are perceived on multiple devices and by several users.
- We take into account the need of providing different "versions" of the same application depending on the *context*, i.e., on the characteristics of both the user invoking a service and the situation of use (including the device). We complement the service model with a *context model* and a *hyperview model*, which allows designer to specify how different users perceive services in different situations of use. Differently from most object-oriented web models, where operations are modelled as methods attached to information "objects", or to navigation or presentation "objects" (pages), we associate operations and services to contexts and hyperviews.
- We adopt a notation based on UML to describe the various models. In particular, we adopt OCL [2] model service constraints and effects, "extending" standard OCL with some predicates that explicitly relate to context and hyperviews.

In this paper, we shortly introduce the main concepts of the overall MC2 framework, and focus on the design of operations, which are the building blocks for designing services.

2. How to design an operation with MC2

In a user-centred perspective, we think of a service as a user *activity*: a (non linear) flow of *tasks* that the user performs within the application to achieve a given goal. Tasks can be progressively decomposed into subtasks up to elementary building blocks that we call *interactions*.

Interactions in web application services have a heterogeneous nature: they can affect the navigation position of the user (and are typically called "navigation") or the presentation state of the objects on the screen (e.g., scrolling a page, playing a sound track); they can change the application state; or they can mix all these effects. We define *operation* an interaction that causes (among other effects) a modification of the application state as perceived by at least one user category.

An operation is defined by the following ingredients [3]:

- The conditions under which an operation can be invoked, or pre-conditions. They are "evaluated" before the operation execution starts.
- The input parameters, i.e., the operation arguments. Arguments can be either provided by the user (user-arguments) or can be "calculated" by the system (system-

HT'03, August 26–30, 2003, Nottingham, United Kingdom.
acm 1-58113-704-4/03/0008.

arguments). Arguments (and operation name) are described in the *operation signature*.

- The effects resulting from executing the operation, described by *post conditions*. They express application properties that must hold after completing the operation and are evaluated after the operation execution.
- The synchronicity of the operation. This property specifies whether the user can or cannot interact with the application while the operation is under execution. The operation is synchronous if no interactions can be performed until the synchronous operation is completed. The operation is asynchronous if other interactions can be invoked and executed in parallel with the operation.

The hardest problem for the operation designer is to model pre and post conditions. In multi-cross channel web applications, they are intrinsically complex because they have to predicate upon a number of variables depending on the context, in which the operation is invoked, and the state of the application. In MC2, a context comprises the characteristics of *users* and of *situations of use*. In multi/cross channel web applications the situation of use often concerns just devices and their technological features. Depending on the application domain, it may involve additional aspects related to location (in its geographical or logical characteristics), time, etc. Pre conditions mostly depend on the current state of the elements composing the hyper schema where the operation is located, and aspects like the user profile, the used device, temporal and location variables, and so on like defined in [4], but however locally to the operation. On the other hand, for post conditions the situation tends to get worst. Executing an operation may involve many sophisticated effects. The user invoking the operation directly perceives some effects, other effects are perceived also (or only) by other users. The invoker may perceive some effects in the same situation of use where the operation has been invoked, others in a different situation of use. Some effects may even change the invoker profile or the characteristics of the situation of use (e.g., some device features), implicitly changing the "view" that the invoker perceives of the application. Finally, operations may affect the user navigational state, determining a user movement in the hyperspace. To allow operation designers take all the above aspects into account, we provide the primitives to model contexts and hyperviews. In MC2 a hyperview represents what a user perceive about an application in a given situation of use. An hyperview includes aspects regarding both the information and the navigation capabilities. We also structure the operation design space in a number of dimensions as shown in Figure1.

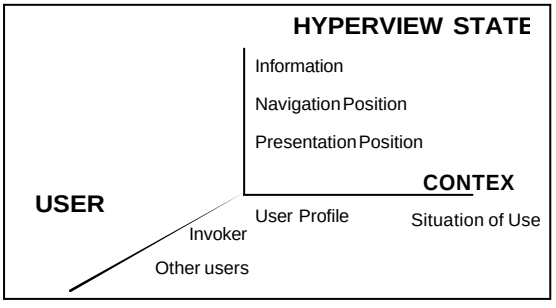

Figure 1. Design Dimensions for MC2 operations.

Using MC2 concepts, we have defined a taxonomy of Pre and Post conditions. Let's consider as example a web conference system and the operation *Submit Review*. In order to write down the Pre-Conditions, we can say the invoker must be a PC member (user profile), then the context could include that the invoker device must be a PC (device) and the review process must be in the *submitting reviews* state (time). The effect of the operation affects the hyperview perceived both from the invoker and other users. The post conditions of this operation should, for example, include statements like "the user is not able anymore to submit a review for that paper", from the invoking user point of view, and "the review has been added to the list of a paper's reviews" that can be perceived by the Program Committee of the conference (a user with a different profile and a different hyperview of the application. We can see this example as cross-context operation affecting the invoker and other users. Just to provide the reader with the idea on how the specification looks like, we report some fragments of the specification of this operation.

```
Context SubmitReview::execute()
  inv:  self.signature.operationName = "submit_review"
    Let arg1:Argument = arg1.name = "p",
      arg1.type = "Paper",
      arg1.category = "user-provided",
      arg1.value -> oclIsTypeOf(Paper)
    self.signature.arguments -> includes(arg1)
    Let arg2:Argument = arg2.name = "r",
    ... ... ...

  pre:
   Let C:HyperViewContext = HyperViewContext.allInstances -> any(c
| c.device = PC and c.user.state = "authenticated" and c.user.type
= "PCMember")
   MC2clCurrentView(C) and
   MC2clCurrentNode = ReviewsIndex and
   ReviewsIndex.type = "ReviewsIndexType"

  post:
    self.executes_on.IS.Paper.allInstances[arg1.value].reviews  ->
    includes(arg2.value) and
    IF    (self.executes_on.IS.Review.allInstances   ->   exists(r|
    arg1.value.reviews -> includes(r)))
    THEN
    self.executes_on.IS.Paper.allInstances[arg1.value].
    reviews -> includes(r)
    Let C':HyperViewContext = HyperViewContext.allInstances ->
        any(c | c.user.type = "PCMember" and c.device = PC and
        c.userID ≠ C.userID)
    MC2clCurrentView(C')
                      IF (self.executes_on.IS.myPapers ->
                        includes(arg1.value) and
    ... ... ...
```

As the previous example shows, MC2 provides a semi-formal approach to specify operations using OCL [2].

3. REFERENCES

[1]. F. Garzotto, V. Perrone. Conceptual Modelling of Services in Multi/Cross Channel Web Applications. In Proceedings of SEBD 2003 (Cetrano Italy, 23-27 June 2003)

[2]. Object Management Group. Object constraints language specification, February 2001.

[3]. L. Baresi, G. Denaro, L. Mainetti, and P. Paolini. "Assertions to Better Specify the Amazon Bug" In Proceedings of 14th International Conference on Software Engineeering and Knowledge Engineering. Ischia (Italy), July 2002

[4]. G. Kappel, W. Retschitzegger, W. Schwinger, Modeling Customizable Web Applications - A Requirements' Perspective, International Conference on Digital Libraries: Research and Practice (ICDL), Koyoto, Japan, November 2000