

Transforming the UI for anyone. anywhere.

Enabling an increased variety of users, devices, and tasks through interface transformations

Charles Wiecha

IBM T.J. Watson Research Center, and
Harvard Information Infrastructure Project
79 JFK Street
Cambridge, MA 02138
wiecha@us.ibm.com
617-495-8878

Pedro Szekely

University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, California 90292
Szekely@isi.edu
310-822-1511

THE TOPIC: ASSEMBLING USER INTERFACES FROM SEQUENTIAL AND SEPARABLE TRANSFORMS

The Challenge: what are the dimensions along which user interfaces may be decomposed and then recomposed?

Historically, application and solution developers have confronted relatively simple and stable targeting problems. When designing applications and solutions, developers would identify and develop for a single dominant design point (or "sweet spot") -- i.e., a targeted role (or business function), a targeted user, and a targeted hardware/software platform -- for the application.

The Internet and e-business transformation revolutions have made these targeting and design problems much more challenging. Application and solution developers now confront several major and rapidly changing "variety challenges" that threaten their ability to develop, implement, and evolve high performance e-business solutions.

- *A Variety of Users.*

The growth in Internet usage is causing a rapid increase in user variety. New groups of users -- with more varied capabilities, skills, and styles than technically-oriented, early adopters -- increasingly dominate the user communities targeted by business process transforming applications and solutions. In addition, technically-oriented early adopters have, themselves, become increasingly diverse as a result of their experience in using a variety of applications.

- *A Variety of Devices and Channels.*

The growth in Internet usage has also combined with rapidly evolving hardware and software technologies to increase the variety of client devices and network channels used to access application and solutions. It is apparent that this device and network channel variety is likely to continue to increase in the future.

- *A Variety of Roles and Functions.*

In addition, as e-business solutions have grown from relatively simple publishing and transactions applications to more complex, collaborative business process applications, the user roles and functions targeted for support have become more differentiated, complex, and varied. For example, multiple interaction styles (e.g., conversational and directed dialogs) need to be supported and applications will need to switch among them on the fly. As e-business solutions are extended within current marketplaces and introduced into not-yet-attacked sectors and markets, the variety of roles and functions is likely to continue to increase.

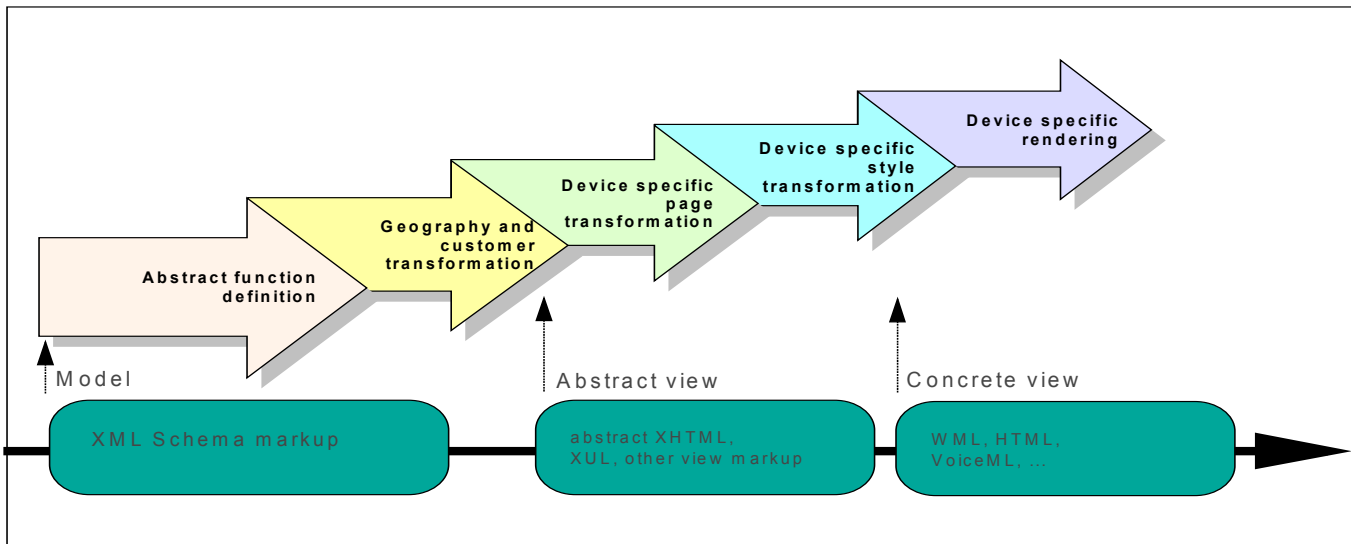
These "variety challenges" combine with several other challenges to further exacerbate the problems facing developers. For example, as user roles have become more complex, more highly interactive, supportive, and easy-to-use user interfaces are required -- particularly in domains within which users are not full-time, heads-down, computer users and when users lack the time and/or patience to participate in training.

A possible solution: chaining interface transforms

The UIMS community has a long history of attempting to abstract user interface functionality from its look and feel. Web developers are in fact beginning to apply some of these ideas through the use of XML, XSL, and the removal of style information from HTML using cascading style sheets. The trend in the evolution of HTML is to remove ever more style information and to treat the XHTML document as abstract UI markup. The UI architecture, shown in Figure 1 below, then consists of a model, abstract view, and concrete view produced by the application of one or more XSL style sheets to the abstract view.

The goal in extending current practice is to factor the application design further in such a way as to separate issues such as, but not exclusively, of device, navigation

Representative transforms and associated markup languages



style, geographic localization, and personal preferences into multiple transforms

Those transformations that may be desired in a given commercial application may include adaptation for differences in geography, device size or connectivity, user expertise, and user or organizational style preferences. Additional customizations could be introduced at low cost, given the abstraction present in the initial application design, when variations in end-user educational level, cultural background, need for collaborative support, and physical or cognitive disability are to be enabled as well.

Goal of the workshop: What are the desirable and feasible dimensions of interface transforms?

While this model is attractive in principle, little work has been done to understand the multiple dimensions along which it is (1) desirable and (2) feasible to factor user interface design. The goal of the workshop is to

- Understand whether the approach to interface design through chaining multiple transforms is attractive to developers, designers, and managers from the standpoint of the programming model. The benefits include increased ability to reuse interface design patterns in different contexts. The costs may include increased cognitive demands in predicting and understanding the behavior of an interface given the multiple transforms that can be involved.
- Understand whether a “fixed” set of layers can or should be identified as a standard approach to interface layering. Contrast this with an “open” architecture where the definition or selection of layers is up to application architects. The benefits of a standard set of layers may be increased confidence that rules at a given layer will interact correctly with those at other layers. The costs of standard layers may include decreased flexibility in extending the architecture.
- Whether a fixed or open approach is preferable, suggest an initial set of layers that should be developed or prototyped. What aspects of interface

design are most important to factor out? This is perhaps the most concrete output of the workshop. In addition, as described below in follow-on activities, the organizers intend to establish a web-based working group to continue discussion and code sharing, most likely around these initial layers.

- Understand, time permitting, what the performance and maintenance implications are of this approach to interface design. Are there approaches to “compiling” sets of transforms so that the run-time performance may be optimized? The benefits of compilation are, of course, increased performance. Compilation’s costs center around decreased ability to respond to run-time circumstances in selecting the transforms that should be applied.

Further Reading

- Universal Interaction, Stephen Boies, Stephen Levy, Julie MacNaught, Paul Matchen, Scott Mcfaddin, David Mundel, Bill Nagy, Rich Thompson, Submitted to IBM Systems Journal.
- Bray, T.; Paoli, J.; and Sperberg-McQueen, C. M. [editors] (1998). Extensible Markup Language (XML) 1.0. (W3C Recommendation). Website: www.w3.org/TR/REC-xml
- Clark, J. [editor] (1999). XSL Transformations (XSLT) Version 1.0 (W3C Recommendation). Website: www.w3.org/TR/xslt
- Barrett, R.; Maglio, P.; and Kellem, D. C. (1997). How to personalize the web. In *Proceedings of the conference on human factors in computing systems (CHI '97)*. New York: ACM Press.