Advanced Interface Productization: Lessons Learned

David Kurlander
Industry Consultant
633 14th Avenue E.
Seattle, WA 98112 USA
+1 206 295 5618
david@kurlander.net

ABSTRACT
Over the years I have managed several research projects relating to advanced interfaces and have turned them into shipping products. Microsoft Comic Chat uses automatic illustration generation and the visual language of comics to present online conversations. Microsoft's Mobile Internet Toolkit renders web interfaces on a variety of mobile devices. Microsoft Surface is a multi-touch, object-sensing table-top display. More recently I have been advising start-ups in the mobile content / ubiquitous computing area. Juggling the competing interests of research and product development has been often interesting, sometimes painful, and always challenging. In my invited talk, I will discuss lessons learned while productizing interface technology, including selecting the product to ship, balancing research and product requirements, navigating management's whims of the day, setting goals, and evaluating the results, as well as what has worked and what has not, and why certain efforts have been more successful than others. Here I present a sampling of these lessons.

Categories and Subject Descriptors
H.5.2 [Information Interfaces and Presentation]: User Interfaces – evaluation and methodology, graphical user interfaces, input devices and strategies, interaction styles.

General Terms
Human Factors, Design, Experimentation, Management.

Keywords
Productization, user interfaces, research, experience, automatic illustration generation, comics, internet chat, mobile devices, mobile web, adaptive interfaces, table-top displays, multi-touch interfaces, object sensing interfaces.

1. RESEARCH, PRODUCTS & LESSONS
The first research project that I developed into a shipping product was Comic Chat [3]. It was an experiment in how automatic illustration techniques might be applied to presenting a conversation, in this case internet chat, using the visual representation of a comic strip. I had this idea in reaction to another product being developed at Microsoft, v-chat, a 3-dimensional virtual chat program. There were aspects of v-chat that I thought could be improved upon. People had to position their avatars in a 3-D space such that their conversational partners were visible, they had to explicitly specify gestures and expressions, and there was no visual history of the conversation. I developed Comic Chat to address these issues, and the program ended up shipping with Internet Explorer, several versions of Windows, and it became the official chat program of the Microsoft Network. It shipped millions of copies and was localized into 24 languages. I also ended up managing the v-chat team. So the very first lesson I learned is to pay attention to other products, and look for windows of opportunity.

In coming up with Comic Chat, I built upon an idea that I had generated years earlier for my doctoral thesis, in which I used linear panels to represent histories, as well as derived macros, in a graphical editor. These editable graphical histories [2] and comic chat look nothing alike, but they are both motivated by the need to represent history in graphical user interfaces. This led to a second lesson – that there are broad opportunities to build upon prior research in non-obvious ways to develop new ideas and products.

A third lesson that I learned shipping Comic Chat is that people will use your product in ways that you never envisioned when building it. Let them do it. Build in extensibility of various kinds to facilitate this broader use – the product will become more popular, and you will learn quite a lot in the process. Although Comic Chat was developed as a graphical internet social chat program, it has been used in military communications [1], comics construction, education, generating visuals for internet radio productions, as a communications channel during conference presentations, and much more.

Related to that, initially we shipped with a fixed set of character art (or avatars), using a particular black and white comic aesthetic. Users demanded the ability to create their own characters (and in color), which we initially interpreted simply as a need for more characters, so we distributed additional black and white character art. Eventually we got the message that people really wanted greater artistic self-expression, so we released a character editor and allowed color art. This was a difficult decision – we loved our initial aesthetic, and this violated it – but it made the program more popular and a large subset of our users happier, so it was the right decision.

Subsequently I managed Microsoft's Adaptive User Interface Group, and we had the research goal of building an interface management system that would automatically adapt to the user and device. This is in many ways one of the Holy Grails of user interface research, and while we made steps forward, we also realized that there were limits to what we were building.
Firstly, one of our goals was to automatically generate interfaces for different size displays. In showing our work to designers that built client-side application interfaces for small screens, we were devastated when they demanded the ability to control every pixel. Screen real estate is particularly valuable on small devices, and our designers were not willing to give up any control. This led to two lessons: get user feedback earlier in the process, and be flexible in charting new courses.

In reaction to this feedback and exploring how our work could better fit within Microsoft’s developer platform, we built the Microsoft Mobile Interface Toolkit [5]. The design environment and run-time is focused exclusively on mobile web content, because in contrast to mobile client-side applications, which are more associated with specific devices, mobile web pages must often work on hundreds of different devices, so our experience is that designers are willing to give up more control.

We needed to swallow our pride, and do smaller adaptations – all targeted at small screens browsing the web. We had to focus on mundane things, like generating the appropriate markup language for the device-specific web browser (HTML, cHTML, WML), pagination and widget selection for the browser, and building a catalog of different devices, making the device-specific information available to the developer. We allowed developers to make arbitrary customizations and overrides for a given device if they wanted. It was not the grand research that we had embarked upon, but that research would have taken years to reach fruition, and there was an immediate need for our product – which still relied upon our earlier research. Microsoft Mobile Internet Toolkit consisted of a design environment and run-time that shipped with Microsoft Visual Studio .NET and ASP.NET, and it was widely distributed and used.

A third research project that we shipped as a product, at least in some sense of the word, was Microsoft Surface [4]. Microsoft Surface is an object-sensing, multi-touch, table-top display. I initiated and managed this project as General Manager of Microsoft’s New Consumer Products group, which had the charter of developing new product concepts that would excite consumers about using personal computers.

One of my favorite aspects of the project was working with people throughout the company to develop the idea and build the first prototypes. I put together a virtual team of people from Microsoft’s consumer division, hardware team, and research group to get to the point where we could excite upper management enough to fund the project. We also met many times with great collaborators from Lego Corporation and discussed possible partnerships, which helped us be creative, and develop something very much out of the ordinary for Microsoft. The lesson here is to work with the best people inside and outside of your organization to get things done.

On the negative side of things, Microsoft Surface was very much a showpiece for the company, and never a mass market product. It was released and did appear at select locations (like Disneyland and the Smithsonian) where the idea was to wow passersby – not be a ubiquitous product. Some of the issues here were form-factor and cost. But we were constrained by Microsoft’s organizational structure and our charter. We were the first to develop multi-touch interfaces for a Microsoft product – in fact years ahead of any other Microsoft product. But we could not develop a phone because we were not in the phone group. We could not develop a tablet because we were not in the tablet group. So we ended up with a table! Part of this was our own doing – we thought (and continue to think) that the table has lots of great scenarios – but were also limited by corporate boundaries.

Potentially we should have spent more time evangelizing multi-touch displays to other groups in the company, but we were a very visible project, one of Bill Gates’ favorites, and the idea did get out, which was partially our doing, but also partially the doing of Microsoft’s competitors. Now some of the ideas and expertise in multi-touch at Microsoft has moved into the phone, tablet, and operating systems groups. Windows 7 and to a greater extent, Windows 8 are, and will be, bringing multi-touch interfaces to millions of Microsoft’s customers. It was certainly more gratifying to ship millions of copies of research reflected in my own product, but there is still great value in influencing other products.

2. BIOGRAPHY

David Kurlander joined Microsoft Research as one of its charter members in 1992. In 1996 he followed one of his research projects into a product team, and found himself leading the chat and communities technology effort. Subsequently David managed Microsoft’s Adaptive User Interface team, which created Microsoft’s mobile web technology for the .NET Platform, and shipped it along with Visual Studio.NET. As General Manager of the New Consumer Technology Group, he led the effort to propose and build exciting new product concepts, including Microsoft Surface, which has strongly influenced Microsoft’s future Windows interface development. David left Microsoft at the end of 2005, and has been working with several technology start-up companies, as well as advising a venture capital firm.

David received a bachelor’s degree in Applied Mathematics from Harvard, and a Ph.D. in Computer Science from Columbia. Prior to Microsoft, David worked at a diverse collection of places, ranging from Xerox PARC to the Pentagon. He has been active in the research community, and was chair of the ACM UIST Conference. Over the years, David has done research or worked on products related to visual languages, graphical editing, interface agents, automated illustration, adaptive interfaces, ubiquitous computing, table interfaces and more.

3. REFERENCES


