

What do we want from a wearable user interface?

Adrian F. Clark
VASE Laboratory, Electronic Systems Engineering
University of Essex, Colchester, CO4 3SQ, UK
alien@essex.ac.uk

In work to date, an “application integration” framework for wearable computing, known as *Sulawesi*, has been developed [3]. Although it incorporates a number of innovations, and is at the limits of what can be achieved on today’s hardware using today’s systems, it is acknowledged that Sulawesi is only a first step on the road to a true wearable user interface (WUI). This document outlines the author’s ultimate goal and suggests one technology that can be exploited by applications writers to make them fit neatly into future application integration frameworks.

The desktop metaphor is dead

As we all know, graphical user interfaces employ a desktop metaphor: overlapping windows mimic sheets of paper lying on a desk; disk directories appear as folders; services are represented by icons; and so on. This is an excellent metaphor for making computers approachable to novices and casual users but is also severely limiting: experienced users tend to migrate towards typed commands or suffer frustration and the spectre of repetitive strain injuries.

As others have commented [1], this metaphor is inappropriate for wearables. The practical problems are well-rehearsed: the LCD technologies used in head-mounted displays are often too low-contrast to be used in “augmented” (see-through) mode in daylight; keyboard and mouse alternatives are cumbersome and degrade interaction speed markedly; and so on. However, most crucial in the context of this document are two other problems.

- Programs that employ graphical user interfaces almost invariably make use of graphical widgets such as buttons and menus at too low a level. Furthermore, the existence of the graphical user interface invariably permeates the entire program.
- Graphical user interfaces are intended to be guided by the user. In other words, they are *reactive*, performing operations in response to input from the user. However, the majority of researchers in wearable computing are working towards *proactive* schemes, in which the system unobtrusively volunteers information that may be helpful to the user. Rhodes’ *remembrance agent* [2] is a good example of a proactive system.

A better metaphor for a WUI

The principal aim of Sulawesi is to support *multi-modal* interactions. The system is able to receive inputs from a number of devices, including typed and spoken commands, and can render textual information in either form. The form of the rendering can be specified by the user (*e.g.*, “tell me the time” and “show me the time”) or can be determined at run-time from GPS and accelerometer data so that, for example, audible output is used when the wearer is moving. For systems such as these, the desktop metaphor is totally inappropriate.

The author believes that a better metaphor for a WUI is a *personal assistant* (PA). A good (human) personal assistant is unobtrusive, predicts what information is needed and prepares it in anticipation of its need, schedules meetings and appointments, *etc.* — precisely the *desiderata* of a WUI. Although there is no need why a “wearable PA” should be anthropomorphic, the author contends that this is desirable as human-human interaction is much more natural than human-computer dialogues. Achieving human characteristics may involve aspects of affective computing [4]; it is certainly desirable to imbue the software with the ability to adapt to the wearer’s mood. For example, immediately after the author has given a lecture, we might envisage a dialogue between him and his wearable PA of the form (where the wearer’s inputs are emboldened):

PA wakeup.

While you were teaching, there were two telephone calls to your office and six incoming emails. None of the emails were marked urgent but one of them is from the head of department. Would you like to process it now?

No. Did the telephone callers leave messages?

Yes. I can play them to you; but since that lecture was to first-year students, why don’t you have a cup of coffee first, like you usually do after seeing them?

OK. Has Neill arrived at the University yet?

Not yet; he’s half-way between his home and the University.

Note this this exchange would work equally well audibly or in visual form — but it is very different from the types of interaction that would occur with a GUI. We believe that these kinds of dialogues can just about be handled with simple natural-language parsers such as that encountered in Sulawesi. Note, however, that even this level of PA support requires a network connection, a computer in the wearer’s office capable of controlling a telephone, access to the wearer’s incoming email, and position-sensing technology in the wearable computers. It probably also requires a multi-processor architecture, so that decoding and playing out continuous media does not inhibit interaction with the user. (Indeed, such tasks may be carried out most appropriately by specialized co-processors.) Furthermore, the wearable PA has monitored the wearer’s habits and is trying to schedule things to accommodate them, something that is just about possible to achieve on today’s systems.

Controlling dialogues and exchanging information

The types of interaction in the above example involve questions and answers in both directions. It is fairly easy to write a single function that can produce yes/no dialogues

in verbal, typed and graphical forms. However, unless all these dialogues are to be hard-wired into programs, some way is needed to be able to specify them in the data communicated between services and output renderers. A scheme based around XML has attractions, particularly since it is predicted that software systems will be increasingly written in scripting languages [5] and XML-derived languages can be generated and processed on the fly.

HTML, of course, already provides tags to provide emphasis and this can apply equally well to audible and visual forms. Sable¹ uses an XML-like syntax for controlling prosody which can also be rendered in visually. VoiceXML² may also be valuable here for specifying and controlling dialogues that may be audible or typed. Equally important is the use of XML-derived mark-up languages for specifying types of content that are normally rendered graphically:

- MathML is used for specifying mathematical expressions. As with the $\text{T}_{\text{E}}\text{X}$ model from which it derives, it can be spoken almost as it is in a form that experienced mathematicians are able to visualize.
- A standard is finally being produced for vector graphics, namely *scalable vector graphics* (SVG). This allows for the production of diagrams, graphs, and even cartoons, in an XML framework.
- Some of the work taking place under the ægis of the Web3D consortium (see <http://www.web3d.org/>) is refining the existing VRML97 standard and re-casting it into an XML framework. This will allow virtual environments, including some aspects of shared ones, to be specified using an XML-derived language.

The biggest omission is in the use of an XML framework for raster data, *i.e.* still and moving imagery. Indeed, it is difficult to see how raster data could conveniently be fitted into XML — though, since HTML (and hence XML-derived analogues of HTML) provides the `IMG` tag, there is an XML-based route for passing even raster data.

References

- [1] Bradley Rhodes. WIMP interface considered fatal. In *Proceedings of VRAIS98*, <http://www.hitl.washington.edu/people/grof/VRAIS98/Rhodes.html>, 1998.
- [2] Bradley J. Rhodes. The wearable remembrance agent: A system for augmented memory. In *Proceedings of the First International Symposium on Wearable Computers*, pages 123–129, Cambridge, Mass, USA, October 1997.
- [3] Neill J. Newman and Adrian F. Clark. Sulawesi: A wearable application intergration framework. In *Proceedings of the Third International Symposium on Wearable Computers*, pages 170–171, 1999.
- [4] Rosalind W. Picard. *Affective Computing*. MIT Press, 1997.
- [5] John K. Ousterhout. Scripting: Higher-level programming for the 21st century. *IEEE Computer magazine*, pages 23–30, March 1998.

¹See <http://www.cstr.ed.ac.uk/projects/ssml.html>

²See <http://www.vxmlforum.org/>