Tools for Designing Computational Spaces

Steven Dow[1], Blair MacIntyre[1], Maribeth Gandy[2]
College of Computing[1], Interactive Media Technology Center[2]
Georgia Institute of Technology
Atlanta, GA 30332, USA
{steven, blair}@cc.gatech.edu, maribeth.gandy@imtc.gatech.edu

ABSTRACT

As the call for this workshop states, there is an increasing need for tools to support design and development of user interfaces in environments where computers are closely integrated with a user's physical space. Over the last few years we have been involved in the design of the Designer's Augmented Reality Toolkit or DART [4]. Our emphasis has been to support the existing work practices of conceptual thinkers (media designers, HCI practitioners, experience planners, etc.) in exploring the novel domain of augmented reality (AR)[1]. Through hours of interviews and observations of designers, we learned of important strategies employed during iterative design. We have worked to support diverse thought and action for a wide range of possible results, rather than simplifying problem domains through tool-imposed constraints. Presented below are four principles that guide the development of DART and contribute to a discussion of future UI software toolkits.

MOTIVATION

It is difficult to imagine a traditional software toolkit (with widgets, components, WYSIWIG editor[2], etc) that would be useful for a wide range of tangible, pervasive, or mixed reality applications. Unlike 2D graphical user interfaces where the design choices are more limited and the medium has matured to the point that there are many standards and expectations, this new world of development has infinite possibilities for input and output interfaces. Several domain-specific toolkits have been created, but generally only support a limited set of applications within the domain (i.e. augmented reality applications for assembly [5]), and even those can be difficult for expert designers to use.

FUTURE OF TOOLKITS

The common uses of a medium determine the requirements for toolkits built to support it. Until MR technology reaches the hands of designers and common metaphors emerge, we will not understand the requirements. Early toolkits must be malleable and aid designers and developers in the entire process of creating complex off-the-desktop applications from initial concept and design, through evaluation, iterative design and experience testing. Future tools for user interface design must also make considerations for working with a multitude of input/output technology in diverse physical environments. These considerations directed our development of DART and the applications we have created using DART. Although DART can be used to deploy finished AR applications, it is also a tool for envisioning, prototyping and testing a wide range of pervasive computing applications. Here we share four principles that inspire our work on DART and will hopefully be useful for developers of future software toolkits.

1) Support a wide range of designers, work approaches, and applications

Black box toolkits are not usually sufficient to enable the range of exploration that designers want and need - especially when the capabilities of the medium are relatively unknown. UI software tools should support a range of technology skills and design strategies, and should be focused on putting *materials* into the hands of designers, rather than *tools*. The technology should be conceptually layered; a good toolkit will provide easy high-level access to common functionality while exposing the internal structure to enable applications never imagined by the tool designers.

Our goal with DART has been to support *designers* and *developers*. DART has been integrated with Macromedia Director[3], an existing visual environment with many affordances commonly used by designers for interactive content. Low-level connections to devices and cameras are handled through an Xtra developed in C++. Everything else is built in Macromedia's own language, Lingo[4]. For programming novices, we have provided an initial set of Lingo behavior scripts. These high-level abstractions can be manipulated on Director's visual score and provide access to the basic functionality for creating simple MR applications. Another affordance of the Macromedia Director environment is the ease of copying and extending Lingo scripts. In our experiences, the existing DART behaviors are not exhaustive; we extend DART each time we build a new application.

Some of our DART developers download the Xtra and create their own Lingo code, using the Xtra API and disregarding our Lingo behaviors. This is a work approach that we encourage and we hope to learn from in the development of appropriate high-level abstractions for non-programmers. At this point the DART Xtra is closed source, but it is built above several

[1] We use the term augmented reality (AR) to refer to graphics and audio that are spatially registered with the physical world. Mixed reality (MR) is a broad range of technology that blends the physical world with digital information.

[2] What you see is what you get (http://en.wikipedia.org/wiki/WYSIWYG)

[3] Macromedia Director (http://www.macromedia.com/software/director/)

[4] We have found Lingo to be a fully capable programming language with very few technical limitations.

open source projects that can be improved by developers (e.g. adding a new VRPN[1] device). By supporting a wide range of skills and work practices, we enable more people to pick up the tools necessary for designing computational spaces.

2) Support the entire design process

It is understood in the CHI community that iterative, human-centered design leads to better computing applications. User interface software tools should naturally support iteration and collaboration with other designers. This approach will enable a more thoughtful exploration of emerging input/output interfaces. At each level of the design process (initial concept, rough prototyping, initial user testing, and development) the designer should be able to thoroughly explore the technology and flexibly iterate the design over time.

We took inspiration from the research community and built in support for sketch-based content into DART [3]. This allows designers to quickly create rough content and work on placing it either on distributed displays in the space or, for AR applications, in a virtual world that matches the physical world. The rough content is flexible and open to adaptation, and in DART can be easily swapped for more finished content, such as Video-based content or 3D graphics.

Another key feature supporting iteration and collaboration in DART is integrated capture and playback [2]. These facilities enable a designer to store a time-synchronized set of sensor data so that designs can be iterated in a static condition, independent of the actual physical space. In AR this is useful for capturing video and tracking data and developing 3D content for a particular scenario. Later in the design process the pre-captured data can be quickly replaced by live data. Since captured data is stored in external Director cast files, designers can share pre-captured scenarios and applications with other designers who have Director and the free DART scripts. Using DART's capacity for video capture, we have essentially enabled *editable video scenarios*. Our editable video scenarios afford high-bandwidth communication and collaboration, but are much easier to create and modify than traditional video editing software like Final Cut Pro. We have moved beyond what is afforded by Director, to support key aspects of the entire design process: concept sketches, collaboration, iteration, and evaluation (discussed below).

3) Deal with the physical environment

Designing the input sensors and output displays for physical environments is more complex than the desktop constraints of the monitor, mouse, and keyboard. There are important decisions regarding the placement of output displays that are contingent on the specific conditions in the physical space. Continuous, ambiguous or multi-user input can be confusing and requires many different design strategies. Realistic and simple simulation of input sensors allows designers to explore input methods and sensors without having to rely on a deployed hardware infrastructure.

One of the strategies with sketch content created in DART is to roughly place the content into the physical world and to evaluate it under real conditions. In a physical space, conditions such as lighting, placement of other objects, and ergonomics can influence designers' ideas.

In DART, we have integrated tools for Wizard of Oz (WOz) simulation of sensors and trackers [1]. This technique is used to simulate real conditions (including realistic sensor error) without having to set up a full sensor infrastructure. Previously, the use of WOz by researchers for user testing has been limited due to the difficulty of constructing a second, networked interface. The WOz tools we created in DART are very flexible so that designers can employ WOz interfaces at many stages of the process for a variety of conditions and user interfaces. We are working on concepts that make the physical environment more manageable while supporting a thorough design practice.

4) Explore new "materials" for communication

The underlying goal of design work, particularly in the early stages, is to communicate ideas about an experience. When designing computational tools for the physical world, designers must communicate about the combined physical/digital forms. Paper is often used to communicate very rough concepts, but for detailed design exploration and communication about a physical environment, there should be a broad range of communication materials. Because we do not yet understand the breadth of MR experiences, the tools themselves should be exposed and modifiable so that good designers can create the building blocks they need.

DART was created to enable rapid prototyping in AR. As the tool matures and reaches the hands of more designers, we anticipate AR will be used as a communication medium for designing a broader class of MR applications, including ubiquitous and tangible computing spaces. AR can be used for simulation of displays and user interaction, communication of concepts, and possibly rough evaluation of an interactive space before it is transitioned into a final physical form. Input and output methods can be explored through supported physical devices (VRPN and Phidgets) or using Wizard of Oz techniques. Applications built with DART can be distributed across any computer that runs Macromedia Director and can result in a wide variety of end-user interfaces from AR to simple 2D graphics to audio only.

[1] Virtual Reality Peripheral Network (http://www.cs.unc.edu/Research/vrpn/)

REFERENCES

1. Dow, S., Lee, J., Oezbek, C., MacIntyre, B., and Bolter, J.D., "Wizard of Oz Interfaces for Mixed Reality Applications," Submitted to *ACM Conf. on Computer-Human Interaction (CHI'05)*, 2005.
2. Dow, S., MacIntyre, B., Gandy, M., and Bolter, J.D., "Prototyping Applications for the Physical World Using Integrated Capture/Playback Facilities," In *Conf. on Ubiquitous Computing (UBICOMP'04)*, 2004.
3. Landay, J., Myers, B., "Interactive Sketching for the Early Stages of User Interface Design," In *ACM Conf. on Computer-Human Interaction (CHI'95)*, 1995.
4. MacIntyre, B., Gandy, M., Dow, S., Bolter, J.D., "DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences," In *ACM Symp. on User Interface Software and Technology (UIST'04)*, 2004.
5. Zauner, J., Haller, M., Brandl, A., "Authoring of a Mixed Reality Assembly Instructor for Hierarchical Structures," In *ACM International Symposium on Mixed and Augmented Reality (ISMAR'03)*, 2003.