

CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services

Anind K. Dey, Gregory Abowd, Mike Pinkerton

Graphics, Visualization & Usability Center
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
+1-404-894-7512

{anind, abowd, mpinkert}@cc.gatech.edu

Andrew Wood

School of Computer Science
The University of Birmingham
Edgbaston, Birmingham, B15 2TT UK
amw@cs.bham.ac.uk

ABSTRACT

Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviours and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, reducing integrating decisions to be made by the tool designers and giving more control to the user. Simple extensions to CyberDesk have been made to obtain powerful integrating behaviours.

Keywords

Adaptive interfaces, automated integration, dynamic integration, software components, context-aware computing, future computing environments, ubiquitous services

INTRODUCTION

Users are tired of using monolithic application suites that allow little to no customization, just because they are industry standards. Tightly integrated suites of tools/services currently available are unsatisfactory for three reasons. First, they require designers to predict how users will want to integrate various tools. Second, they force users to either be satisfied with design decisions or program their own additional complex relationships between the tools. Finally, users must be satisfied with the available services themselves, because they are often given no opportunity to replace or add services.

CyberDesk [2] is a component-based framework, written in Java, that relieves most of the burden of integrating services from both the designer of individual services and the end user. It provides greater flexibility to the user, and automatically suggests how independent services can be integrated in interesting ways. The components in CyberDesk treat all data uniformly, regardless of whether the data came from a locally running application or from a service running on the World Wide Web (WWW). The services and applications themselves can be running anywhere, meeting CyberDesk's goal of providing ubiquitous access to services.

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee

UIST 97 Banff, Alberta, Canada

Copyright 1997 ACM 0-89791-881-9/97/10...\$3.50

User Scenario

The user selects which applications/components they would like to use by adding them to a Hypertext Markup Language (HTML) page and loading the page in a web browser. The user receives an e-mail message (see Figure 1) with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d).

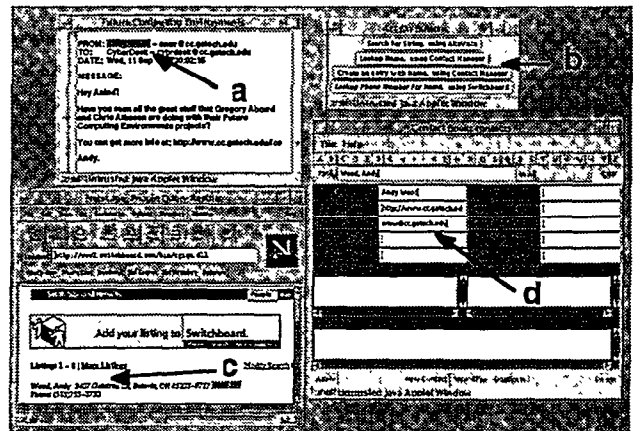


Figure 1: Screenshot of contact manager, e-mail tool and Switchboard service being used with CyberDesk.

WHAT DOES CYBERDESK GIVE US?

CyberDesk provides a simple framework for adding new services and integrating them in intelligent ways. It relieves burdens from both the individual service designer and the end user. The individual service designer can develop a generic service, with a usable API, and not have to worry how it will be integrated into CyberDesk. The designer does not have to design specifically for the CyberDesk framework. The designer also doesn't have to think of all possible ways a user may want to integrate this service with another service, because the integrating behaviour is inherent to the CyberDesk framework. CyberDesk creates a dynamic mapping at runtime from user actions to possible user actions, saving the designer from

constructing this map at design time. More information on CyberDesk's architecture can be found in [1].

CyberDesk makes things easier for the user as well. The user has the ability to easily add and remove services from the framework and does not need to hunt for ways to integrate various tools. The user is often supplied with integrating suggestions that they do not expect or had not thought of, but are appealing nonetheless.

EXTENSIONS

The CyberDesk framework was designed to be easily extensible. Simple extensions to CyberDesk include adding additional types, type converters, desktop services and network services. We have had several students develop simple extensions for CyberDesk, including desktop and network services, type converters and types. As a testament to the ease of development, at last count there were 6 desktop services, 68 network services, 7 type converters, and 6 data types. The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behaviour of CyberDesk to individual use and creating more interesting integrating behaviour. These advanced extensions are possible without any modifications to the basic CyberDesk structure.

Mobility

One extension we've made to CyberDesk is to add services that allow it to access data from mobile devices. In particular, CyberDesk can access name and string data from an Apple Newton PDA (Personal Digital Assistant). This provides ubiquitous access to information, regardless of where it is stored.

Chaining

Chaining is another example of a complex extension. It extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behaviour. In the user scenario, the user highlighted a name and received a list of actions relating to names that she could perform. Chaining creates more interesting suggestions for the user, as is shown below. The WhoWhere network service can take a name as input and return a WWW browser showing a list of possible e-mail addresses corresponding to that name. By making the assumption that the first e-mail address returned in the list is the correct one, we can now use this service to convert the name to an e-mail address. The service now creates a related e-mail address selection event, and the user is supplied with all possible suggestions for both a name and an e-mail address.

Combining

Along the same line of thought, chaining can be used along with the concept of *combining* to make services more powerful. The services previously described were designed to only operate on a single data type (at a time). With data being converted to multiple types via chaining, the idea is that services, both network and desktop, should be able to take advantage of these multiple types. They can, through a process we call combining. Combining, in CyberDesk terms, is the ability to collect multiple data types for a piece of selected data, and bind them together, as needed, to create multiple meta-objects. These meta-objects could be used to perform substantially more powerful actions. Again

referring to the user scenario, with multiple services capable of chaining, selecting a name can generate multiple selection events, so a URL selection event is created, a date selection event is created, a mailing address selection event is created, etc. There is now enough information to create a rich new entry in the Contact Manager. The ability to assimilate this widespread information into a compact entity is very powerful for a user.

Other Forms of Context

CyberDesk contains some simple notions of context. It knows the application a user is working with and the data (both type and content) the user is interested in (via explicit selection with the mouse). But CyberDesk has shown the potential for supporting other forms of context. We have implemented prototype services that accept position and time information for use with CyberDesk. These services do not require explicit user action for events to be generated. The position service is for the campus of Georgia Tech. The service coarsely maps GPS (Global Positioning System) coordinates to buildings on campus.

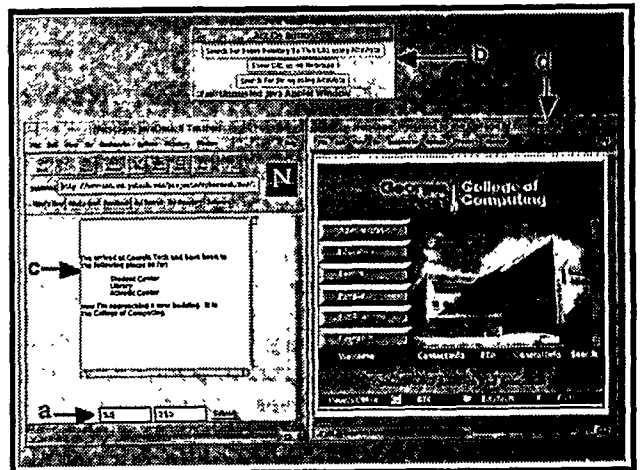


Figure 2: Screenshot of position service. (a) is where GPS coordinates are being input, causing changes in the ActOn Button Bar (b) when the coordinates correspond to a different Georgia Tech building. The user is keeping track of his trip in the scratchpad (c), and is able to view the building URLs in the web browser (d).

A demo version of CyberDesk is available at <http://www.cc.gatech.edu/fce/cyberdesk>.

REFERENCES

1. Dey, A., Abowd, G., Wood, A., Pinkerton, M., CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services. Technical Report, GVU Center, Georgia Institute of Technology. GIT-GVU-97-10. June, 1997.
2. Wood, A., Dey, A., Abowd, G. CyberDesk: Automated Integration of Desktop and Network Services. Technical Note In Proceedings of CHI' 97 (Atlanta, GA, March 1997), ACM Press.