

# Context-Aware Computing: The CyberDesk Project

Anind K. Dey

Graphics, Visualization, & Usability Center,  
Georgia Institute of Technology  
Atlanta, GA 30332-0280 USA  
anind@cc.gatech.edu

## Abstract

The CyberDesk project is aimed at providing a software architecture that dynamically integrates software modules. This integration is driven by a user's context, where context includes the user's physical, social, emotional, and mental (focus-of-attention) environments. While a user's context changes in all settings, it tends to change most frequently in a mobile setting. We have used the CyberDesk system in a desktop setting and are currently using it to build an intelligent home environment.

## Keywords

Dynamic integration, mobile computing, ubiquitous computing, intelligent environments

## Introduction<sup>1</sup>

The CyberDesk system is a software architecture that has been developed for the domain of context-aware computing. Context-aware computing is an area of research where the human computer interface leverages knowledge of the user's context. Context includes, but is not limited to, information the user is attending to, emotional state, focus of attention, location and orientation, date and time of day, objects and people in the user's environment. With greater consumer acceptance of ubiquitous and mobile computing, users are often in a position where their context is changing rapidly and the amount of information that they have access to at any time can grow quite large. This dynamically changing context can be used to predict what information the user wants at a particular time.

The CyberDesk architecture uses contextual information to dynamically integrate software modules together. This has the effect of reducing programming burden in identifying and defining the integrating behavior of software modules, while at the same time, retaining as much user freedom in determining how integration is to occur. This context-aware integration moves away from the present situation in which the user must seek out

relevant software functionality when he wants it, to a situation where the infrastructure seeks out the user when and where he wants it.

## Background

There hasn't been a lot of work in the area of context-aware computing, mainly due to the fact that there was not enough hardware infrastructure to support and create a compelling application. Most of the work in this area has leveraged off of knowledge of a user's location: Xerox Parc's PARCTAB [14], Olivetti's Active Badge project [13], Schilit's thesis work in context-aware mobile computing [12], and the CyberGuide project [1]. More recent work has dealt solely with knowledge of the information a user is attending to: OpenStep's notion of facilities [2], Apple Data Detectors [3], and Intel's Selection Recognition Agent [9]. The Microsoft Office Assistant uses knowledge of a user's history to help predict a user's goal and to help achieve it. Picard [10] and Essa and Pentland [7] have used various techniques to infer a user's emotional state to help guide the user.

In the area of integrating software services, much work has been done. This ranges from the use of simple UNIX pipes, to more complex systems like Field [11], Smalltalk's Model-View-Controller [8], and the Common Lisp Object System [4]. Our underlying infrastructure allows dynamic integration of isolated services at run-time. This mediation consists of two basic steps: registration of components and handling of events. While the more complex systems listed above provide for these two basic steps, CyberDesk appears to be more flexible for the domain of context-aware computing, mainly because it was designed for this domain.

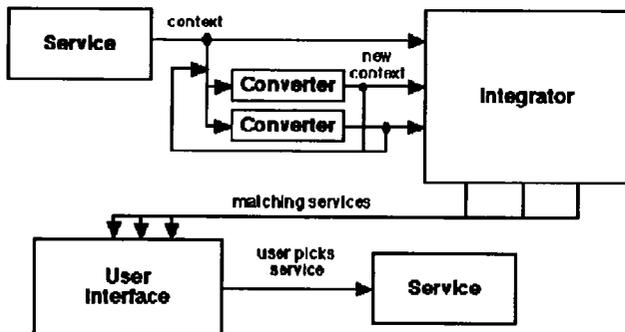
## Architecture

The CyberDesk system consists of five main components: a Registry, information services, type converters, an Integrator, and a user interface. The Registry maintains a list of components in the system and the interfaces that each supports. The information services are the tools and functions the user ultimately wants to use, such as an e-mail reader, a contact manager, or a Web-based search engine. These services register their interfaces with the

<sup>1</sup> Copyright © 1997, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

Registry and announce events that provide data/information to the rest of the system (e.g., the name selected in the e-mail message in the scenario). The type converters accept announced data from the system and convert it recursively to other forms of data that may be useful (e.g. a string being converted to a URL). The Integrator uses the Registry to automatically find matches between user data and the services that can use the data, a task that would normally be performed by the system designer. The matched services are then displayed to the user through the user interface for integration.

The run-time relationship between the components (not including the Registry) is depicted in Figure 1.



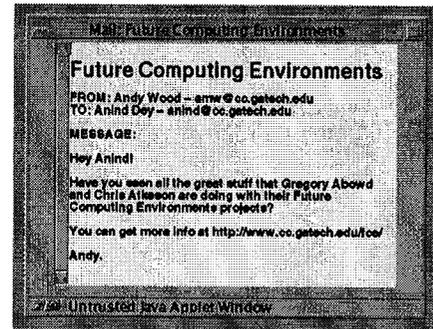
**Figure 1. The run-time architecture of CyberDesk.** Arrows indicate the flow of information and control in the system.

## Applications

Initially, CyberDesk was only able to work with simple strings that a user was working with in a desktop application. Now, CyberDesk is also able to work with time and location, other people in the room, and objects in the room, in a desktop environment, networked environment, and mobile environment. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time, This can be particularly useful when the user's context is changing frequently, as in a mobile, intelligent environment. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected help in completing their tasks.

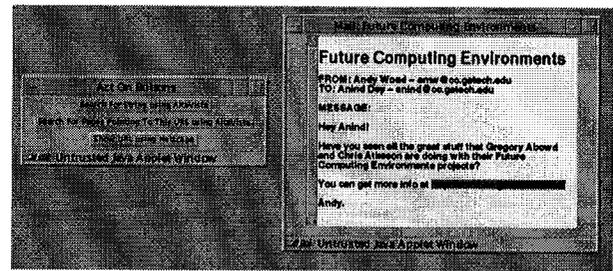
### Desktop System

To illustrate this behavior, an actual user experience follows. As seen in Figure 2, a user is checking his e-mail, and reads one about some interesting research.



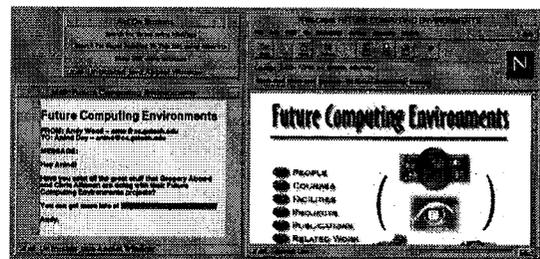
**Figure 2. Content of User's E-mail Message.**

The user is interested in the research discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface (Figure 3): search for the selected text using AltaVista, find pages that reference this URL using AltaVista, and display the URL in Netscape.



**Figure 3. User selects the URL and is offered suggestions.**

He chooses the last option and views the URL listed in the message (Figure 4).



**Figure 4. CyberDesk executes the service and displays the URL.**

The user then selects the name of the person in charge of the research and is offered the following suggestions (Figure 5): search for the selected text using AltaVista, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact this researcher so he checks to see if the name is in his contact manager, but it isn't.

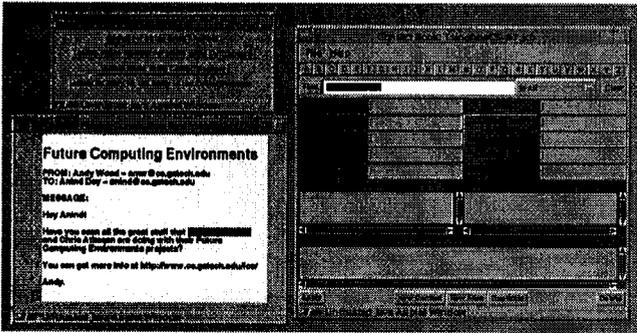


Figure 5: User selects name and chooses the Contact lookup service.

So, he selects the phone number and mailing address lookup service (Figure 6). He then creates a new entry in the contact manager with this new information.

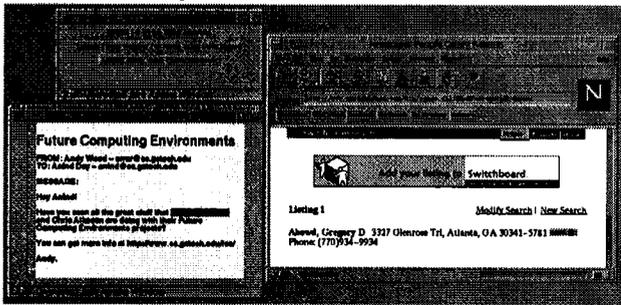


Figure 6. User selects the phone number lookup service.

After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is given a name, it returns a phone number and mailing address. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context.

So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information, but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort in locating this information themselves. As seen in Figure 7, the user can easily create a rich contact entry, simply by selecting a name.

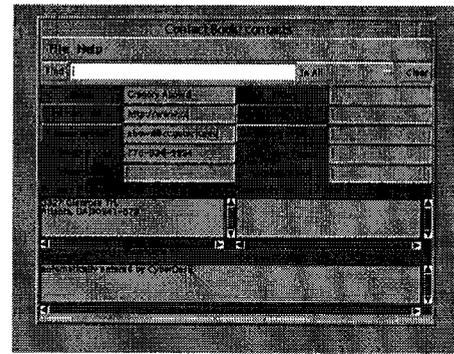


Figure 7: User creates a rich contact entry.

### Mobile System and Intelligent Environment

We have used the CyberDesk inferencing engine to create a mobile scenario as well. It behaves similarly to the system previously described, with a few extensions. These extensions include the addition of context types (user position, identification of others in the environment, identification of objects in the environment) and allowing the user to be mobile, using either a series of networked wall-based displays mounted in various rooms or a device the user can carry. This portion of the project is a work in progress.

In our test environment, we have 3 separate rooms: a kitchen, an office, and a living room. The mobile system is a work in progress. We are using face recognition to identify the primary user and others in the room. We use an infrared positioning system to track both users and the wireless mobile computing devices. Some objects in the environment are active and intelligent. For example, the system knows when the user approaches the fridge and the fridge lets the system know about its contents. As the user enters a different room, the CyberDesk system is fed with new context information: type of room and identification of objects and people in the new environment. For example, when the user enters the kitchen, he may be provided with the following list of suggestions:

- Obtain a list of fridge items
- At mealtime, create and display a list of recipes that can be made with available ingredients and knowing what kind of food the user likes
- View important reminders and notes.

When the user enters the living room, the list might be this:

- View a list of today's headlines
- Look at recent sports scores
- Read personal e-mail.

Or when the user enters the office, the list might look this:

- Load the last document being worked on
- View business-related reminders
- Load latest set of meeting notes (or if someone was in the room with the user, load notes for last meeting that both the user and the other person had attended).

The software development for this scenario is complete. Only the integration of the various hardware components, the pieces of the system providing the context information to CyberDesk, is left to complete.

## Conclusions

The CyberDesk system provides a software architecture designed to support context-aware computing. It leverages knowledge of a user's context to dynamically integrate software services, in support of a user's task. A desktop system has been built which uses knowledge of the information a user is attending to. A mobile scenario in an intelligent environment is currently being built. Future work includes increasing the use of context and using statistical analysis to determine which of the possible choices a user is likely to choose.

## Resources

For more information on the CyberDesk project, including Web-based demos and other publications, see <http://www.cc.gatech.edu/fce/cyberdesk>

For more information on other related projects at Georgia Tech, see <http://www.cc.gatech.edu/fce>

## Acknowledgments

This work is supported by Motorola Corporation through the University Partnerships in Research (UPR) Program, sponsored by Dr. Ron Borgstahl. I would like to thank Andy Wood from the University of Birmingham and Gregory Abowd from Georgia Tech for their efforts in building the original CyberDesk infrastructure with me. I would also like to thank the Future Computing Environments Group and the numerous other undergraduate and graduate students at Georgia Tech who have provided much inspiration and support in the development of the initial CyberDesk prototype.

## References

1. Abowd, G.D. et al. CyberGuide: A mobile context-aware tour guide. ACM Wireless Networks, 3, 1997. To appear. Available at <ftp://ftp.gvu.gatech.edu/pub/gvu/tr/96-27.ps.Z>
2. Apple Enterprise Software. Topics in OpenStep Programming. Available at <http://devworld.apple.com/dev/SWTechPubs/Documents/OPENSTEP/Programmin gTopics/>, 1997.
3. Apple Research Labs. Apple Data Detectors homepage. Available at <http://www.research.apple.com/research/tech/AppleDataDetectors/>, 1997.
4. Bobrow, D. et al. Common Lisp Object System Specification X3J13 Document 88-02R. ACM SIGPLAN Notices 23 (special issue). September 1988.
5. Dey, A.K., Abowd, G.D., & Wood, A. CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services. In the Proceedings of Intelligent User Interfaces '98, January 1998.
6. Dey, A.K., Abowd, G.D., & Wood, A. CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services. Technical Report, GVU Center, Georgia Institute of Technology. GIT-GVU-97-20, 1997.
7. Essa, I. and Pentland, A. Facial expression recognition using a dynamic model and motion energy. In Proceedings of the International Conference on Computer Vision, pages 360-367. IEEE Computer Society, Cambridge, MA, 1995. Available as MIT Media Lab Perceptual Computing Techreport #307 from <http://vismod.www.media.mit.edu/vismod>.
8. Krasner, G.E. and Pope, S.T. A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80. Journal of Object Oriented Programming 1(3) (August/September 1988), pp. 26-49.
9. Pandit, M. and Kalbag, S. The Selection Recognition Agent: Instant Access to Relevant Information and Operations. In Proceedings of Intelligent User Interfaces '97. ACM Press, 1997.
10. Picard, R. Affective computing. Technical Report 321, MIT Media Lab, Perceptual Computing, November 1995. Available as MIT Media Lab Perceptual Computing Techreport #362 from <http://vismod.www.media.mit.edu/vismod>.
11. Reiss, S.P. Integration mechanisms in the FIELD environment. Technical Report CS-88-18, Brown University, October 1988.
12. Schilit, W.N. System architecture for context-aware mobile computing. Ph.D. Thesis, Columbia University, May 1995.
13. Want R. et al. The active badge location system. ACM Transactions on Information Systems, 10(1): 91-102, January 1992. Available at <ftp://ftp.cam-orl.co.uk/pub/docs/ORL/tr.92.1.ps.Z>
14. Want R. et al. An Overview of the PARCTAB Ubiquitous Computing Experiment. IEEE Personal Communications 2 (6). 1995. 28-43.
15. Wood, A., Dey, A.K., & Abowd, G.D. CyberDesk: Automated Integration of Desktop and Network Services. Technical Note in the Proceedings of CHI'97, pp. 552-3.