The Invisible

One lab's
experiment
with
ubiquitous
computing

Assistant

GAETANO BORRIELLO, UNIVERSITY OF WASHINGTON

Ubiquitous computing seeks to place computers everywhere around us—into the very fabric of everyday life¹—so that our lives are made better. Whether it is improving our job productivity, our ability to stay connected with family and friends, or our entertainment, the goal is to find ways to put technology to work for us by getting all those computers—large and small, visible and invisible—to work together. Since Mark Weiser presented the ubiquitous computing vision in 1991, we have made significant progress in creating faster, smaller, and lower-power computing devices. We have just barely begun, however, to tackle the problem of how we get these devices to interact effectively with us and with each other.

In the past, computers have been tools. We have explicitly commanded them to execute the steps to get a job done. Using them in this way inadvertently turned many of us into system administrators—loading and upgrading software, configuring networks and services, and moving and storing data. In fact, it seems that the time devoted to these activities increased more than linearly as each pair of interactions between the devices required special attention. We have tried to automate some of these tasks, but automation still has a long way to go and often creates additional compatibility issues.

Today computers are an integral part of our environment, in much the same way as electric motors became embedded into appliances that targeted specific tasks and stopped being perceived as electric motors.² An ABS (anti-lock braking system) in a car now integrates data from several sensors on the wheels and pedals to help a driver apply the brakes most effectively. Of course, humans could do this, but an ABS can do it faster and better—accelerating the application of the brakes when pedal movement is first detected and then creating a feedback loop with the actual wheel behavior to prevent skids. Many sensors and computers are involved, but there is only a simple but highly specialized user interface: the brake pedal. We are spared the details of how all these devices are coordinated.

In the future, computers will be our assistants, not only helping us do things more easily but also anticipating our needs. We will want computing to be an extension

The Invisible Assistant

of ourselves—with improved senses and abilities. For example, we are starting to see cars augmented with radar systems that detect when a car is braking in front of us or that the road conditions are changing. By detecting a dangerous situation automatically, the brakes can be applied with much faster “reflexes” than any human could possibly hope for. The increased safety margin is something we will demand and be willing to pay for.

Research in ubiquitous computing is focusing on the issues that arise in building such future proactive applications using ensembles of computers and sensors. At the University of Washington, we worked on an application of ubiquitous computing that helped us begin to explore these issues.^{3,4,5,6,7,8} The project was led by Larry Arnstein, who went on to found a company, Teranode, to offer elements of this application commercially.⁹

LABSCAPE

Cell biologists perform experiments in wet labs—laboratories where samples are mixed with reagents, heated, centrifuged, separated, gelled, etc. This is a data-intensive environment where it is important to record every detail so that it may be reproduced with little additional effort and high fidelity by other researchers. To accomplish this, cell biologists mostly use paper notebooks. Many complications, however, can get in the way of recording all the information in a timely and accurate manner.

For example, risk of contamination prohibits bringing paper or PDAs into the lab. This forces researchers to remember as much as they can and record it later or scribble it on small scraps of paper they take out of the lab. To compound the problem, researchers often conduct several experiments at a time to fill in the significant time gaps that occur between steps in an experiment. For example, while waiting for a sample to incubate, researchers will start work on another experiment in parallel. Now they must make sure each detail, such as how much of a reagent was used or to what temperature a sample was heated, is associated with the correct experiment.

Yet another complicating factor is that multiple researchers may work together on a single experiment. Different people may specialize in using a certain piece of equipment or may be known to be better at performing certain steps. Therefore, more than one person may handle a sample, with large time gaps in between, yet the data collected at each step must come together to document the single experiment.

In the research lab, as opposed to the production lab, researchers often need to consult other sources to decide on what to do next in their experiments—there is not necessarily a fixed protocol to follow. The information researchers need to decide what to do can come from static resources available on the Web or even from the result of an earlier step in the same experiment. It is important for researchers to have access to this data in situ so that they can remain focused.

It is in this type of data-rich environment that we sought to develop a ubiquitous computing solution that would automatically record all the data associated with conducting an experiment and make data available in all the places it was needed. Our goal was to enable automatic documentation based on the researchers just doing their work, rather than through explicit additional steps as is currently the practice. Our system was intended to be an assistant that would just be over the researcher’s shoulder writing everything down as it happened and providing data as it was needed—without being asked. Think of the Radar O’Reilly character from the television series *M*A*S*H* for a vision of this type of assistant.

In the accompanying box we use a series of images gathered during our work to describe Labscape, the ubiquitous computing application/environment that we developed.

The first issue we tackled was to create a framework for organizing the data. To do this, we formalized what researchers already did. Before going into the lab, they

Labscape in Action

Touch-sensitive tablet displays at each workstation show the current state of the experiment using the flow graph originally specified by the researcher.

The flow graph is a place to collect all the data associated with an experiment. Each step includes parameters that are either pre-specified—and used as reference—or filled in by completing a step in the procedure.

would usually sketch out a series of steps they were going to follow just to help keep things straight and provide a place to record the data that was generated. The first element of our application was a tool for specifying an experimental flow graph—a sort of schematic for the experiment. We were able to classify all the steps in the laboratory into eight fundamental categories, and we defined icons for each along with a set of parameters that would be filled in advance (e.g., the type of reagent to use) or during the experiment (e.g., how much reagent was actually used).

Cell biology wet labs are organized spatially. Benches are allocated to specific procedures so that equipment can be time-shared among many researchers. Thus, each bench could be associated with a particular type of step in our flow graph. We used small infrared tags to detect a researcher's proximity to a particular bench and barcodes on samples to uniquely identify each container. Each bench was also equipped with a tablet computer to provide a touch-sensitive display surface that could be used both to view and to enter data. Depending on the equipment at each bench, we also developed some special devices to adjust the settings on the equipment or record changes made by the researcher.

We determined where a researcher was during an experiment based on several pieces of context gathered by the devices we introduced into the lab. First, we could associate a person with a bench using proximity tags. Second, we could associate a sample with a bench or piece

of equipment (e.g., refrigerator or centrifuge) by placing barcode readers at each bench or using RFID tags on the samples and reader antennas in the bench-top. Third, we logged equipment usage by adding wireless sensors to each tool, including even pipettes (in an unobtrusive way that did not change the affordances of the instrument), and recording how they were used and at which bench. Finally, any data produced by a tool was also logged and associated with a bench, researcher, and sample. This required creating some adapters, as many tools provide digital information already but in a nonstandard form that is difficult to integrate into a unified data repository. For example, we wanted the data to carry an identification number for the instrument so that it could be checked against calibration and maintenance records.

We paid particular attention to documenting how researchers filled in the many wells in a microarray tray. This was done using a camera that looked down at the work area. Markers on the pipette tip and tray allowed the vision algorithm to position and orient the objects accurately. If the pipette was triggered to dispense its contents, we could record how much liquid was dropped into precisely which of the many wells. The pipette could also be used to query the contents of a well by just hovering its tip over the well. A data projector, mounted in parallel with the camera, was used to project this information alongside the array where the researcher could easily read it. Minor parameter changes from well to well helped immensely in keeping track of how far along a researcher

Instruments in the lab are modified to provide important data. In this case, the reagent bottle includes a barcode and RFID to identify it on the bench-top, and the pipette wirelessly transmits how much liquid it has dispensed and its own ID (for keeping track of calibration).

A pipette can be used to index a microarray and record new material deposited into a particular well or query to find out what was placed in that well in the past. The information is displayed on the tabletop by a projector mounted above the bench surface.

Changes in instrument settings are automatically recorded so that the documentation is as complete as possible.

A photograph taken by an electrophoresis machine is associated into the experiment's flow graph and is available for viewing from anywhere the flow graph is displayed.

The Invisible Assistant

had progressed in doing a repetitive procedure. It also made it easy to see what was different about a well when visual inspection revealed an interesting effect.

All the pieces of information associated with an experiment were brought together through the data flow graph. Parameters in the graph were filled in as the researchers performed each step. It didn't matter if multiple people were doing the work since the samples and researchers were both identified. Similarly, a researcher could work on multiple experiments with any interleaving he or she wished. At each point, we would have a fairly good idea of what would need to be done, based on what the sample was and what flow graph it was associated with; what the next possible steps were in that flow graph (those whose inputs were already available); who the researcher was, based on bench-to-person proximity; and what equipment was available at that bench, from the lab's configuration data. It was usually possible to disambiguate where the generated data belonged.

To back up the system, we provided visual feedback on the touch screen at each lab bench—the researcher was identified on the screen and the experiment plan corresponding to the sample was brought up automatically, showing the next possible steps and when data was integrated from something that happened at the bench. This made it easy for researchers to use their peripheral vision and always be able to check that the system was on track.

Of course, there were some confounding factors. For example, two researchers might be at the same bench during the time a step was executed; a particular piece of equipment may not be instrumented to be compatible with our data gathering; multiple samples could be present on the bench at the same time. In these cases, we used the touch display to show the possible interpretations of the available data and let the person there make an explicit association. Sometimes, the researcher wanted to change the flow graph because of something observed during the experiment. This alteration had to be permitted with the appropriate re-association of any data.

Once an experiment was completed, all the data to document its steps and results was automatically associated with the original experiment flow graph. Several stakeholders benefited from this. The original researcher had all the data needed to document each step of the experiment—we envisioned a tool that could even automatically write the methodology section of the research paper. The researcher's colleagues, who may have contrib-

uted by helping with some steps, did not have to worry about getting that information transferred—it was done automatically through the associations recorded by the lab's ubiquitous infrastructure. By having the data and experiment design in machine-readable form, the greater research community could search for experiments based on similar samples, procedures, or results—and finding another researcher's data, could have an easier time either replicating the results or using them to decide on the next steps to take in their own lines of investigation.

Note that this application is not a traditional one. It is long-lived in that no one ever "starts" or "quits" the application. It is running all the time in the laboratory as it can distinguish between different researchers and experiments. It is an integral part of that environment and is maintained like any other piece of equipment by the lab's staff. It is context-aware as it uses different pieces of sensed data to classify information and decide what the user may want to do or see next.

There were several implementations of the Labscape software, all of which were Java-based. Our initial ad hoc versions were replaced by a version that was built on top of systems software we developed for ubiquitous computing applications.¹⁰ It provided support for migrating applications from machine to machine and screen to screen, as well as providing centralized management capabilities.

LESSONS LEARNED

The Labscape experience provided several important lessons about building proactive applications. These range from commonsense lessons to implementation details.

Invisibility doesn't mean there is no user interface.

There should always be a user interface that is easily seen and allows the user to check that the system is operating as expected or that can override what the system is doing proactively. In Labscape, this implied a touch screen in every work area showing the flow graph for the experiment. In addition, it showed how each piece of sensed data was used to adorn the flow graph.

Principled approach to sensor fusion. Ambiguity is always possible when using sensors to determine what people are doing. Using multiple sensors can help disambiguate difficult cases. It is important to think about sensor fusion from the start, as it is extremely difficult to add later. It is also important to involve the user to resolve more difficult ambiguities in a timely manner through the user interface.

Incremental deployment. It should be possible to install the system in pieces rather than as a monolithic entity. New equipment is constantly added to and old

equipment removed from laboratories. The system needs to be resilient to adapt to these changes and not require major reconfiguration every time an alteration is made.

No alteration of work practices without a good reason. At all times, we tried to fit our technology into existing work practices and devices with as little impact as possible on work practices. Adoption of ubiquitous technology is held back when entirely new models for interaction are required. Users want a problem solved with measurable improvements to their work. They are much less interested if work practices are so altered as to present an entirely new set of oft-unforeseeable problems.

Fail-safe. The application needs to be easy to maintain and operate in a fail-safe manner. We learned the importance of making all of the software running on devices in the lab be stateless. That is, it should be possible to turn any one piece of equipment or display on and off at any time without losing valuable data. Persistence was maintained in a single server that contained a database of all sensed context and experiment flow graphs. Data associations could be reconstructed at any time and did not rely on file systems or in-memory data structures.

Standards, standards, standards. To make a system such as Labscape truly useful, we need to standardize data formats so that an ecosystem of supporting tools can be developed. This includes the flow graph, sensed context, and equipment settings.

CONCLUSION

Labscape was one of the first applications of its kind, but it is part of a new wave that is building. Similar ideas are finding their way into applications ranging from hospitals¹¹ to elder care¹² to oil tanker maintenance.¹³ Their distinguishing features will be the use of context to organize information for presentation to many stakeholders and the ability to predict what will be needed at any point in time. Long-lived, context-aware, proactive applications will be the way in which ubiquitous computing provides benefit to society. Q

REFERENCES

1. Weiser, M. 1991. The computer for the 21st century. *Scientific American* (September).
2. Norman, D. A. 1998. *The Invisible Computer*. Cambridge, MA: MIT Press.
3. Arnstein, L., Sigurdsson, S., Franza, R. 2001. Ubiquitous computing in the biology laboratory. *Journal of Lab Automation* 6(1).
4. Arnstein, L., Borriello, G., Consolvo, S. Hung, C., Su, J. 2002. Labscape: a smart environment for the cell biology laboratory. *IEEE Pervasive Computing Mobile and Ubiquitous Systems* 1(3).
5. Arnstein, L., et al. 2002. Systems support for ubiquitous computing: a case study of two implementations of Labscape. 1st International Conference on Pervasive Computing, Zurich, Switzerland (August).
6. Consolvo, S., Arnstein, L., Franza, R. 2002. User study techniques in the design and evaluation of a Ubicomp environment. 4th International Conference on Ubiquitous Computing, Goteborg, Sweden (September).
7. Grimm, R., Davis, J., Lemar, E., MacBeth, A., Swanson, S., Anderson, T., Bershad, B., Borriello, G., Gribble, S., Wetherall, D. 2004. System support for pervasive applications. *ACM Transactions on Computer Systems* 22(4): 421-486.
8. Hile, H., Kim, J., Borriello, G. 2004. Microbiology tray and pipette tracking as a proactive tangible user interface. 2nd International Conference on Pervasive Computing, Vienna, Austria.
9. Teranode; <http://www.teranode.com/>.
10. See reference 7.
11. Bardram, J. E. 2005. Activity-based computing: Support for mobility and collaboration in ubiquitous computing. *Personal and Ubiquitous Computing* 9(5).
12. Consolvo, S., et al. 2004. Technology for care networks of elders. *IEEE Pervasive Computing Mobile and Ubiquitous Systems: Successful Aging* 3(2).
13. Krishnamurthy, L., et al. 2005. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the North Sea. 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys 2005), San Diego, California.

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

GAETANO BORRIELLO is a professor of computer science and engineering at the University of Washington. He is known primarily for his work in automatic synthesis of digital circuits, reconfigurable hardware, and embedded systems development tools. Recently, Borriello was primary investigator for the Portolano Expedition, a DARPA-sponsored investigation on invisible computing of which Labscape was a keystone application. He was on partial leave from 2001 to 2003 to found and direct the Intel Research Seattle laboratory, which has matured into one of the premier labs for ubiquitous computing research. His research interests are location-based systems, sensor-based inferencing, and tagging objects with passive and active tags.

© 2006 ACM 1542-7730/06/0700 \$5.00