Invisible computing: automatically using the many bits of data we create

BY GAETANO BORRIELLO*

Department of Computer Science and Engineering, University of Washington, Box 352350, Seattle, WA 98115, USA

As we go about our work and our daily lives, we leave a trail of bits behind. Every electronic device we interact with can keep a record of our actions. Even the devices themselves can keep track of their location and radio interactions, even without user involvement. The challenge of invisible computing is to make this wealth of data useful. This paper presents two examples of what has come to be known as ‘invisible computing’, namely, devices recording, distilling and rendering these many bits of data without unduly taxing human users. The first example is focused on a work environment. LABSCAPE automates the record keeping required of experimenters in a cell biology laboratory. The second example looks at more ad hoc interactions. RFID Ecosystem is a collection of radio-frequency identification (RFID) readers and databases that collect the sightings of passive RFID tags, attached to people and objects, as they move throughout a large building. It provides services such as people and object finding as well as diary keeping.

Keywords: ubiquitous computing; proactive applications; data mining; monitoring systems; work-flow systems; radio-frequency identification

1. Introduction

The term ‘ubiquitous computing’ was coined by Mark Weiser in 1991 as he embarked on a project at Xerox’s Palo Alto Research Center to realize a future where computing devices would be all around us and be so common and seamlessly integrated as to fade into the background (Weiser 1991). This vision has inspired many directions of research based on different interpretations of the phrase ‘the most profound technologies are those that disappear’ (Norman 1998). Computing has gained myriad adjectives to describe the many aspects of this vision that different research groups have emphasized, including such terms as: invisible; mobile; pervasive; tangible; immersive; palpable; embodied; embedded; ambient, etc. Each of these has already served to define the topic of research conferences and academic journals. Although there is substantial overlap among these terms, there are also some important differences. We can...
discern several important themes that focus on three aspects of invisible computing: connectivity; user interfaces (UIs); and automatic inference.

**Connectivity.** Data communication is one of the most commonly referred to aspects of ubiquitous and pervasive computing, namely, that devices are everywhere and communicate with each other to provide users with the information they need when and where they need it. Devices that we carry such as personal digital assistants, laptops and phones are moving among other users’ devices and devices in the infrastructure such as displays, servers and desktop machines. An example utility in this space is mesh networking that permits devices with shorter-range radios to extend their range over an area populated with similar devices that can serve as relay stations for a more distant user’s communication. The goal for those focused on this aspect of invisible computing is that the connections between these devices should be managed as automatically as possible so as not to burden the user with the details of authentication, security and privacy protection. Choosing the appropriate medium for communication among the many available is also an important research problem in this space. There are many communication technologies available ranging from long-range systems such as cellular data networks and SMS to short-range media such as Bluetooth radio and infrared, with Wi-Fi somewhere in between. The choice among them is motivated not only by bandwidth requirements and power consumption but also by the security and privacy implications of widespread communication. As devices emit signals that contain authenticating or identifying information, they reveal the presence of their owners and permit monitoring and tracking of that user. As they collect data, they may download malicious viruses.

**User interfaces.** The great majority of the processors around us do not have traditional UIs characterized by windows, icons, menus and pointing devices. Instead, they often have highly specialized UIs that provide an appropriate affordance to the job at hand. For example, an automobile’s anti-lock braking system uses a simple brake pedal for user input as well as the data from the drive train and wheels to prevent skidding as braking occurs. An automatic heart defibrillator has two sensor paddles and a single button that starts the device’s processing to determine the best time to issue a life-saving electric shock. Sensors often provide the input to these highly specialized interfaces and try to connect the physical world to our computing devices. Pedometers offer an interesting example of an accelerometer being used to count a user’s steps during a run to help them keep track of how well they are meeting their fitness goals. Game controllers such as the Nintendo Wii now enable physical gestures to control the movements in a video game rather than the pressing of buttons, thereby providing the player with a physical and more immersive experience. This is the objective of the UI focus on invisible computing, namely, to provide more appropriate affordances in our computing and control devices that are a better match to the task at hand, thereby lessening the cognitive burden on the user to map their actions to the desired results.

**Automatic inference.** Sensors on the physical world can generate a huge volume of data. Coupled with the data communications occurring between nearby devices, we have a plethora of information about what is happening. A simple example is the collection of global positioning system data from vehicles to infer traffic conditions from their average velocity rather than installing...
expensive road infrastructure—these sensors scale more effectively as well by providing more data where there are more vehicles—after all, if the road is devoid of vehicles, traffic data are not that important. Inference can be much more difficult when the same low-level events may be common to several higher-level activities. Home monitoring applications attempt to make sense of this kind of information to infer the many possible activities that could be occurring in a home. For example, an elder moving into a room and lying down could be someone just taking a nap or falling and not being able to get up on his/her own. Most approaches in this area use statistical machine learning to train classifiers on a set of data for which ground truth is known. Classifiers decide into which category to place new data based on this past experience. Of course, researchers struggle with the collection of well-annotated data and the interactions and segmentation of activities. Inference of higher-level activities is further complicated as long-lived activities are likely to be interrupted and interleaved with others. An example of this would be when an elder sits down to a meal, then answers the telephone or the door, and comes back to the meal some time later.

The commonality between these foci of activity is trying to get our computing devices to do more and more automatically. Invisible computing is extending the era of interactive computing where we, as users of computers, directed each step and awaited a response so that we could decide what to do next. Today, we have too many devices to manage. They are starting to dominate our time. This is the reason many people react negatively to the vision of a future of ubiquitous computing—if all those devices need to be tended to, then how will we ever have time to do the things that really matter to us? We need to make systems automatic—and as invisible as possible—if we are going to be able to scale human abilities and, most importantly, human time to deal with the millions of computing devices that will be all around us.

In this paper, I will look at two projects that have been undertaken at the University of Washington to investigate invisible computing and learn important lessons that will help us form the principles of design in the new era of ubiquitous and invisible computing. The first is LABSCAPE, an invisible computing environment for the biology laboratory (Arnstein et al. 2001, 2002a, b). This is an exemplar of a structured work-flow environment that allows us to explore how activities can be classified and data correlated into a searchable database. The second is RFID Ecosystem (Welbourne et al. 2008), a more ad hoc work environment where we are trying to explore the privacy implications of having fine-grain sensing of people’s whereabouts and associations with others.

2. 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 of what is done 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. However, there are many complications that get in the way of recording all the information in a timely and accurate manner.
For one, there are contamination risks that prohibit bringing paper or mobile electronic devices into the laboratory. This forces researchers to attempt to remember as much as they can and record it later or scribble it on small scraps of paper they take out of the laboratory. To compound the problem, many researchers are often conducting several experiments at a time because there are significant time gaps between steps in an experiment. For example, to wait for a sample to incubate, researchers will start work on another experiment in parallel. Now they must keep details—such as how much of a reagent was used or to what temperature a sample was heated—associated with the right experiment. Yet another compounding issue is that multiple researchers may work together on a single experiment. Different people often specialize in using a certain piece of equipment or are known to be better at performing certain steps. Therefore, a sample may be handled by multiple people with large time gaps in between, yet the data collected at each step must come together to document the single experiment.

In the research laboratory, as opposed to the production laboratory, 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 needed 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 these data in situ rather than having to leave the laboratory to go back to an office so that they can remain focused (figure 1).

It is for this kind 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 the place they are 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 needed—without having to be asked.

*Phil. Trans. R. Soc. A* (2008)
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 laboratory, they would usually sketch out a series of steps that they would plan to follow just to help them think through the experiment and provide a place to record the data generated. The first element of our application was a tool for specifying an experimental flow graph—a sort of schematic for the experiment (figure 2). We were able to classify all the steps in the laboratory into eight fundamental categories and defined icons for each along with a set of parameters that would be filled in, in advance, while planning the experiment, for example the type of reagent to use, or while conducting the experiment in the laboratory, for example how much reagent was actually used.

Cell biology wet labs are organized spatially. Benches are allocated to specific procedures so that equipment can be timeshared among many researchers. Thus, each bench can be associated with a particular step type 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 a user could use to both view and 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. These were highly specialized to equipment ranging from thermocyclers to electronic pipettes.

We determined the point at which a researcher was doing an experiment based on several pieces of context that could be gathered by the devices we introduced into the laboratory. First, we could associate a person to a bench using proximity tags. Second, we could associate a sample to a bench or piece of equipment, for example refrigerator or centrifuge, by placing barcode readers at each bench or using radio-frequency identification (RFID) tags on the samples and reader antennas in the bench top. Third, equipment usage was logged by adding wireless sensors to each tool, including even pipettes (in an unobtrusive way that did not chance the affordances of the instrument), and recording how and at what bench

Figure 2. 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.
they were used. Finally, any data produced by a tool were also logged and associated with a bench, researcher and sample. This required creating some adapters, as many tools provided digital information already but in a non-standard form that was difficult to integrate into a unified data repository (figure 3). 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.

As an example, we paid particular attention to documenting how researchers filled in the many wells in a microarray tray (Hile et al. 2004). We used a camera to look down at the work area. Markers on the pipette tip and tray allowed the vision algorithm to accurately position and orient the objects. 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 (figure 4). This helped immensely in keeping track of how far along a researcher had got in doing a repetitive procedure with minor parameter changes from well to well and also made it easy to see what was different about a well when visual inspection revealed an interesting effect. This is an example of a specialized UI that blends in with the task and does not take time away from the work at hand because we use the tools of the task itself also to interact with the computing devices.

All the pieces of information associated with an experiment were brought together through the experiment’s data flow graph. Parameters in the graph were filled in as the researchers performed each step. It did not matter if multiple people were doing the work, as the samples and researchers were both identified. Similarly, a researcher could work on multiple experiments with any interleaving they wished. At each point, we would have a fairly good idea of what would need to be done based on: (i) what the sample was—and the flow graph it was associated with; (ii) what were the next possible steps—those whose inputs were already available; (iii) who the researcher was—based on bench-to-person proximity; and
(iv) what equipment was available at that bench—from the laboratory’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 laboratory 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 were 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 whether the system was on track. Invisible computing must provide feedback to the user that associations and inferences are being made correctly and provide easy ways of correcting the system when it inevitably falters—hopefully, only occasionally—and make the corrections easy to implement (e.g. by providing menus of the most likely corrections, gaps or reinterpretations of the data). In the case of LABSCAPE, it was very important that we were able to provide a work-flow for the experiment so that the sensor data could decorate an established template. It would have been a much more difficult problem if there had been no pre-established work-flow and many more errors would have been made as the data were used to build a model of the experiment.

Of course, there were some confounding elements even with an established work-flow. 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 researchers wanted to change the flow graph owing to something they observed during the experiment. This alteration of the flow graph may also require an appropriate re-association of previously collected data.

Once an experiment was completed, all the data to document its steps and results were automatically associated with the original experiment flow graph. Several stakeholders benefited from this. The original researcher had all the data they needed to document each step of their experiment—we envisioned

*Phil. Trans. R. Soc. A* (2008)
a tool that could even automatically write the methodology section of their paper. Their colleagues who might have contributed in doing some steps did not have to worry about getting that information transferred—it was done automatically through the associations recorded by the laboratory’s ubiquitous infrastructure. By having the data and experiment design in machine-readable form, the greater research community could also easily search for similar experiments based on similar samples, procedures or results—and in finding another researcher’s data, one should have an easier time either replicating the results or using them to decide on the next steps to take in one’s own line of investigation.

It is important to note that this application is not a traditional one. It is long-lived in that no one ever ‘starts’ the application or ‘quits’ it; 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 as any other piece of equipment by the laboratory’s staff. There were several implementations of the LABSCAPE software—all JAVA-based implementations. Our initial ad hoc versions were later replaced by a version built on top of systems software we had specifically developed for ubiquitous computing applications (Arnstein et al. 2002b; Grimm et al. 2004). It provided support for migrating applications from machine to machine and from screen to screen as well as providing centralized management capabilities.

3. Lessons learned from LABSCAPE

Invisibility does not mean there is no UI. There should always be a UI that the user can easily see and check whether the system is operating as expected and that can be used to override what the system is doing proactively. In LABSCAPE, this implied a touch screen at 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. In using sensors to determine what people are doing, there can always be ambiguity. However, 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 on later. It is also important to involve the user to resolve more difficult ambiguities in a timely manner through the UI.

Incremental deployment. It should be possible to install the system in pieces rather than as a monolithic entity. New equipment is constantly added to laboratories and old equipment is removed. The system needs to be resilient to these changes and not require major reconfiguration every time an alteration is made.

Do not alter work practices without a good reason. At all times, we tried to fit our technology into existing work practices and devices with as minimal 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—often 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 laboratory be stateless. That is, it should be possible to turn any one piece of equipment on and off at any time without losing valuable data. Persistence was maintained in a 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 on our data formats so that an ecosystem of supporting tools can be developed. This includes the flow graph, sensed context and equipment settings.

4. RFID ecosystem

RFID tags are an interesting technology for ubiquitous computing (Want 2004). They can be attached to an object or carried by a person and provide a unique identifier that an invisible computing system can use to unambiguously identify that person or object. There are active and passive RFID tags. Active tags have their own power source and can communicate autonomously as well as collect their own sensor readings—they are a type of small wireless computing device. Passive tags do not have a power source; they harvest the energy they need from the antenna of a reader—a computing device used to interrogate the tag. Passive tags are particularly interesting in ubiquitous computing as they can be attached and forgotten—they require no maintenance throughout their lifetime (figure 5).

In the last few years, RFID technology has gained increasing attention as a flexible and relatively fast solution for tagging and wireless identification (Stanford 2003). Early successes in the asset tracking and supply chain domains coupled with the falling cost of tags have led researchers to consider pervasive, public RFID deployments that support more user-oriented services. A number of
investigations into personnel tracking and task automation using RFID (Borriello et al. 2004) have shown the technology’s potential to facilitate everyday life by seamlessly integrating the virtual and physical worlds. Unfortunately, the majority of such studies have been limited to technology and user evaluations over a short time in restricted scenarios (often in a laboratory). Furthermore, the publicity surrounding this work has revealed an intense public concern with RFID privacy and policy issues that have gone largely unaddressed.

I believe that a more holistic approach is required to effectively design and evaluate RFID-based pervasive computing systems. To this end, we are deploying a long-term, building-wide RFID-based test-bed in our department’s building that involves hundreds of RFID readers and antennas and thousands of tags. We deploy our readers throughout our seven-storey building and mount the antennas on the ceilings of our hallways where they are spaced 5–7 m apart and are the required 25 cm away from any humans walking by (a requirement of the US’s Federal Communication Commission to limit long-term exposure; figure 6).

Our intent with this ‘RFID Ecosystem’ is to explore the benefits of pervasive RFID infrastructures while identifying and addressing their challenges before such systems are adopted widely in other public settings, where problems may have more serious implications. The density of readers allows us to extract fine-grain location information for any tagged object or person. Tag-read events (TREs) are tuples that contain the antenna that reads the tag, the tag’s ID and the time of day at which the sighting occurred. From this raw information, with probably many missing reads, we want to extract higher-level events that will be of interest to the population of over 800 people who work in our building. We are dealing with a work environment where interactions have a much more limited repertoire than any arbitrary situation. However, I feel we still have much to learn about the impact that a monitoring environment based on RFID tags can have on people’s interactions and perceptions of themselves and others.
Several properties distinguish RFID infrastructures for pervasive computing from those for supply chain applications. First, pervasive RFID applications are likely to evolve and grow over time. We already see RFID in elder care and object-finding applications, each of which requires a flexible infrastructure that facilitates provisioning. Supply chain applications are typically less dynamic and apply the technology in a narrower capacity (mostly for inventory tracking). Second, because a pervasive application will typically track people and belongings rather than items in inventory, privacy issues must be considered much more carefully. Finally, people are less predictable than goods moving through established distribution patterns in a supply chain. As such, we must develop fundamentally new ways to deal with the variable-rate, partial and noisy data likely to be generated by human activity. RFID tags are notoriously unreliable when near people or metal objects, as the RF energy from the antennas is readily absorbed by water and reflected by metal. Tags carried by people have a high incidence of missed reads depending on where they are carried. Therefore, our inference must be probabilistic at its core and infer reads that should have happened but for some reason did not. For example, when a tag is read at two antennas at opposite ends of a long hallway in a short period of time, it should have also been read at the antennas in between. If these tag reads are missing, our inference methods have to be able to re-insert them (Re et al. 2008).

The deployment of a pervasive RFID-based infrastructure in an everyday environment allows us to experiment with a wide variety of applications. These can be broken up into the following three rough classes: personal; social; and enterprise. Personal applications include a diary that automatically populates a Google calendar with persons encountered and meetings attended so that one can search on their everyday work activities (figure 7). Our goal is to connect this to activities on individuals’ laptops and phones so that a more complete record of communications and file creation/edit can be integrated. The idea is to extend the concept of desktop search to include additional context such

![Figure 7. A Google calendar automatically filled in by the RFID Ecosystem to represent how a user spent their day, including activities and encounters with others.](image-url)
as location and the presence of others. Another example of a personal application is an object finder that simply helps find where or with whom an object owned and tagged by the user may have been misplaced (figure 8). Social applications include the obvious friend-finder augmented with information about that friend’s current context, for example in a meeting, in a casual hallway conversation, or on their way to lunch. In addition, we are also imagining applications that help research groups study the communication habits of group members and work to instil appropriate working practices. Finally, enterprise applications are those closest to what is being done for the supply chain. These include inventory management—where different pieces of equipment are and who installed them or moved them to their current location—and building security. Enterprise applications serve an important purpose as they serve the technical support staff who are likely to be charged with maintaining the infrastructure of readers, antennas and TRE databases, and thereby help ensure their longevity. For all of these applications, there is leverage in exploiting the history of tag reads and the statistical distribution of tag sightings. This is fundamental in dealing with the inherently noisy nature of our sensors. The metaphor of search extends in this direction as well because search results can be ranked in terms of their probabilities.

Figure 8. A map of the fourth floor of our building showing the last known location for a tagged object. We can also add the last direction of travel from the last two sightings.

Phil. Trans. R. Soc. A (2008)
Enabling these classes of applications, however, presents a significant challenge to the system design (Welbourne et al. 2008). Not only must the reader/antenna infrastructure offer enough accuracy and granularity to enable these applications, but it must also support many individuals, groups and the enterprise as a whole, simultaneously. To enable the monitoring of large spaces with a possibly large number of tags, the system must scale to handle high-volume streams of tag reads. The architecture of our system has been designed to separate different concerns at different levels. For example, our reader infrastructure reports tags seen at each antenna with 1 s granularity (readers could generate thousands of TREs per second because they continuously read the tags in the environment, including the same tags over and over again). Applications are serviced by a layer of servers that can be provisioned as needed by the volume of users. Thus, we bound the possible amount of data generated and transmitted to the (possibly replicated) database while permitting additional application servers to come online at any time and begin accessing those data.

Finally, because such a system will manage large amounts of (potentially sensitive) personal data from multiple users, it must be secure and support an appropriate privacy model. An active area of research is developing privacy models that are easily understood by users. For example, our RFID Ecosystem only allows visibility to data that correspond to what a user probably saw by being physically present. We call this the physical access control (PAC) model (Kriplean et al. 2007). The basic idea is to make people’s memory closer to perfect but not give them the power to see through walls or into spaces they are nowhere near. PAC works by computing the visibility users would have had given where their tag was seen by our readers. Those users’ applications then only have access to tag reads at the same antenna or others within sight of where the users were standing. Thus, a query of how many people were near a particular room would be answered with a number corresponding to those who were within sight of the person asking the question and not the actual number. The exchange would be ‘how many people were in the south section of the fourth floor?’ with the answer ‘you saw four people in that area’.

Other privacy mechanisms we are investigating include giving each person a way of knowing who is asking about him/her and how he/she is showing up in others’ calendars and other applications. We do this to exploit the social pressures that prevent someone from asking about another’s whereabouts many times for fear of being perceived as prying or nosy. Potential embarrassment can be a powerful force in limiting anti-social behaviour. In addition, we are assigning costs to various pieces of information so as to force trade-offs between cost and usefulness. For example, asking whether someone has been seen in the building is of very low cost as this is not very specific information. On the other hand, asking a person’s precise present position is significantly more costly. The idea is to limit access to the more detailed information for when it really matters rather than just for idle curiosity.

5. Lessons learned from the RFID Ecosystem

Scalability at all levels. Issues of scale have to be considered from the ground up when building such a large infrastructure. Although this is not surprising, it has substantial implications for how we distributed the functionality of our system. For example, to minimize the number of TREs and limit privacy concerns, we do
not keep any TREs for tags that are not issued by us. However, this requires the readers to have access to the tag database. We accomplish this by caching tag registrations daily at the reader in hashed form so that, even if a reader is compromised, the tag database is not.

*Direct representation of uncertainty*. We must represent uncertainty in our application as a first-class concept. We use particle filters to represent a tag’s likely geographical position as a probability distribution. This is propagated throughout our inference process all the way up to the applications. It was challenging to recast the output of most of our applications as ranked search results.

*Varied applications to provide value to many users*. There are many types of people in any large organization with different roles and responsibilities. We purposely created a set of applications within which each group would find something of value to its work or social life.

*Privacy considerations are highly nuanced*. Even the smallest amount of information can have privacy implications. We were conscious of not putting readers near the building’s restrooms to alleviate any fear that personal habits would be tracked. However, it was fairly obvious just from people leaving and re-entering their offices when they might have gone on a bathroom or coffee break. When we made it possible to see when someone asked about another person’s location, we were surprised that the absence of someone asking could also raise concerns.

*Opting out may be difficult*. Although we have a vetted process for recruiting participants, it is clear that with such a large-scale deployment even people who choose to opt out may have some information gathered about them. For example, a faculty member pointed out that, even if he did not carry a tag, his students’ presence near his office would allow inference as to his whereabouts at least part of the day. This is a complex challenge we are still pondering as to how to best handle.

*Aesthetics matter*. The large amount of equipment we wanted to install in the hallways raised concerns and objections regarding the inaesthetic appearance of RFID readers and antennas. We found, however, that no one noticed antennas attached to the ceiling as long as the cables were hidden. Similarly, readers near the building’s entrances were placed in cabinets that matched the décor. Other useful camouflage included non-metal ceiling tiles, glass windows and non-metallic paint.

### 6. Conclusion

Invisible computing is an emerging field based on the concepts of context awareness and the mining of data coming from myriad sensors and devices. LABSPACE was one of the first applications of its kind, but it is part of a new wave that will be building. Similar ideas are finding their way into applications ranging from hospitals to elder care to oil tanker maintenance. The RFID Ecosystem follows in the same direction as Sentient Computing (Addlesee et al. 2001) and PARCTAB (Want et al. 1995) in creating a large-scale monitoring environment in which people work and find utility from a system that can automatically gather information for them and make it easier to retrieve data when needed. The distinguishing feature of these systems is the use of context to organize
information for presentation to many and varied stakeholders. Long-lived, context-aware, proactive and mostly invisible applications will be the salient features of our advance from the era of interactive to ubiquitous computing.

Special thanks are due to Larry Arnstein who led the LABSCAPE project and took part of it to commercial venture in the form of Teranode, Inc., and to Robert Franza who generously provided his laboratories at the Cell Systems Initiative for our experimentation. Evan Welbourne and Magda Balazinska have been instrumental in making the RFID Ecosystem a reality. There are also the numerous students, both graduate and undergraduate, whose hard work and creative efforts made possible the two projects described in this paper.

References


*Phil. Trans. R. Soc. A* (2008)