User Interface Design, Transaction Measurement, and Usability Engineering for Remote Devices

Randolph G. Bias, Ph.D. 1, Bill K. Bodin2, Samuel A. Burns1, Anuj A. Nanavati1, Ding-Yu Chung1, and Ann Marie Maynard, Ph.D.3

1 The University of Texas at Austin School of Information
2 IBM-Austin Advanced Technology and Research
3 IBM-Austin Research Laboratory

rbias@ischool.utexas.edu, bbodin@ibm.us.com, sabunome@earthlink.net, ananavati@mail.utexas.edu, alexchung@mail.utexas.edu, amg@us.ibm.com

IBM Technical Contact:

Bill K. Bodin, STSM, Director
Pervasive Computing
Advanced Technology Laboratory
IBM-Austin

bbodin@ibm.us.com

Abstract

The following summarizes our program of study addressing innovative usability engineering for remote, migratory transactions in a pervasive computing environment. The IBM-Austin Pervasive Computing Advanced Technology Laboratory (PvC Lab) is investigating and developing creative applications of remote monitoring and control capabilities. Such capabilities call for attention to advances in user interface design. Our year of study addressed how best to present the cross-device user interfaces, and how best to collect, share, and analyze associated usability data. This final report offers the theoretical and practical context for usability studies of ubiquitous computing, and goes on to present a prototype user interface for a PvC Lab usability test room, the first draft of a taxonomy for pervasive computing tasks, and a plan for the design of a PvC usability lab.

1. Introduction

We pursued a project on innovative human factors- and usability-based technology and its applicability to various devices being researched, developed, and integrated by the IBM-Austin Pervasive Computing Lab (PvC Lab). We begin with a general overview of the research approach, including a theoretical and practical context for usability studies of ubiquitous computing [10]. Then we offer the results of our study, including a prototype user interface (UI) [Appendix B] for a PvC Lab usability test room, the first draft of a taxonomy [Appendix A] for pervasive computing tasks, and a plan for the design of a PvC usability lab.

Four themes in usability engineering – four themes at various stages of emergence, but all timely – converge in the IBM PvC Lab and Director Bill Bodin’s concept of “migratory transactions.” A migrating transaction is one that is begun on one device and completed on another (with the possibility of yet other, intervening steps, on other devices).
The computing environments being researched at the PvC Lab include the car, any room of the house, plus traditional computing sites (e.g., business offices and conference rooms), pursuing traditional and innovative computing tasks (e.g., collaborative design across multiple sites).

The four converging themes are:
- Focus on task-oriented UIs,
- Device-independent UI design,
- Computer-supported cooperative work (CSCW), and
- Remote moderated usability testing.

2. Four Themes

Focus on task-oriented user interfaces

There is an inglorious history of human-computer interface (HCI) design based on the structure of the underlying technology. That is, computer scientists and electrical engineers have tended to design UIs and Web sites with the structure of the UI paralleling the underlying database structures, the organizational structure of the development teams, or otherwise reflecting the functional design, rather than having the UI maximally support the users in the completion of their tasks. In recent years the user-centered design (UCD) approach has brought focus on the users’ tasks [11, 13], and helped drive designs that maximally support users and their tasks [14]. This focus will be even more important as tasks are carried out across multiple devices.

Device-independent UI Design

That is, migratory transactions will demand focus on the user’s tasks. The design of UIs for tasks that might be finished on a different device than the one on which they were started will make obsolete technology-based structures for UIs, because the designer won’t know what technology will be involved in the completion of the task.

What shall be the elements of a UI design when the designer is uncertain of device it will be displayed on? Can there be a meaningful design one level short of the actual instantiation of the interface on a particular device? This question is especially nettlesome when that device could vary in size from a cell phone to a wide-screen TV [8].

Computer-Supported Cooperative Work (CSCW)

CSCW applications have been developed to enable computer users to gain a synergy and efficiency in working together at multiple sites [4, 5, 9]. Increasingly, there’s a move to increase the amount of information about each of the cooperating workers, or “awareness.” Applications are allowing for added social awareness (e.g., who’s gazing where?), situation awareness (e.g., ability to replay actions), informal awareness (e.g., the ability to peek into another, remote office, and to know who’s peeking in on you), and other sorts of metadata about the distributed meeting. The goal is to make the remote working interaction as much like the traditional, co-located working interaction as possible.

Remote Moderated Usability Testing

Traditional usability testing involves a lab, a one-way glass, and one-at-a-time, close-up user testing. Motivated largely by the advent of the Internet, and the need to evaluate what people across the world would find usable, there has been a move towards remote usability testing [1, 3]. Such testing allows for access to customers, prospects, and employees in some city OTHER than where a usability lab exists, thus being a low-cost alternative to travel.

Technology that allows usability professionals to moderate usability testing has included tools such as NetMeeting and WebEx, sometimes in parallel with telephones. Some researchers have found that testing a user in his/her natural environment stimulates the users to talk more. Indeed, it maximizes the similarity between the testing situation and the real-world setting, thus maximizing the generalizability of the test results.

Recently, usability engineers wanting to log button pushes on an infrared remote control have used a highly configurable remote-control program like REMmitter together with an infrared receiver connected to a computer. This approaches shows promise, for remote usability testing of various devices.

Our study has driven towards validation and implementation of recommendations regarding the merger of Human Factors metrics gathering and the dynamic infrastructure of the PvC’s Advanced Technology Laboratory. The goal is to define and create a completely Internet-accessible usability
testing and validation center, void of geographic hurdles. With this venue, IBM will lead the way in which device interactions, UI assessments, and server-side recommendations are made.

3. Our Line of Study

The University of Texas at Austin School of Information provides research and instruction in various aspects of information science. We address how information is categorized, stored, archived, retrieved, and otherwise managed. Information is only of value, though, if human beings can access and process it. Thus, key aspects of information studies are usability engineering and information architecture -- how can information best be presented to users to enable them to find and process the information?

We set out to serve the PvC Lab with a two-pronged approach, addressing both a) the information content and its structure given a pervasive computing environment, and b) the innovative usability methods required to study migrating transactions.

Study of Information Content and Structure

Starting with extant research on task-oriented UIs and device-independent UI design, we worked with the PvC Lab team to identify a grammar of UI design. A dozen years ago IBM led the way, in their CUA work, in identifying elements (objects and actions) of graphical user interfaces. Migrating transactions demand another level of abstraction. In traditional language grammars, abstract concepts get instantiated in various ways. For instance, the "deep structure" concept of "President George Bush" can be instantiated accurately in a sentence as any of a variety of noun phrases: "President George Bush" or "President Bush" or "Mr. Bush" or "him" or "he."

With the proliferation of possible devices, designing for each possible device is not a scalable solution. How might we build an underlying grammar of a design that can be interpreted by and for each device?

Innovative Usability Methods

Taking off from Bodin’s migrating transactions, we pursued what we call trans-device usability (TDU). A UI might be found wondrously usable when carrying out a task on one device, and totally unusable on another. How will this drive usability engineering? How can we start with the CSCW basics, and the emerging tools of remote usability testing, and build a new approach to usability evaluations that will enable us to test TDU? The concept of two usability engineers observing a test participant behind a one-way glass can be, must be, expanded to consider multiple usability engineers in multiple locations, “observing” (maybe with video feed, maybe with simply a recording of user interactions, maybe even with simple read-outs of certain psychophysical data) a user in yet another location (maybe a car). Together with the PvC Lab team, we endeavored to set up such a prototype test environment. We identified a taxonomy of user testing settings, and detailed the technology, the usability methods (e.g., end-user testing, usability walkthrough, expert test), and the expected test data (e.g., time on task, error rates, particular navigation paths) for each.


To narrow the scope for this first phase of our collaboration, we generated specific scenarios in the arena of cross-platform copy protection and cluster management (xCP). With this focus, for which the PvC Lab has prototype hardware and software, we can recreate, test, and validate various pervasive computing scenarios specifically as they tie to eCommerce.

To begin this line of research, Phase I was to include the selection of certain migratory transactions for study, the evaluation of the current user interfaces (UIs) on various involved devices, the use of existing remote usability evaluation methods, and the identification of gaps in those methods. We will drive towards innovative methods and tools for such remote testing, and, along the way, make real usability improvements to the UIs that we study.

Test Scenarios

The specific test scenarios were generated in concert with the Pervasive Computing Lab team. We wanted to identify scenarios that:

- are possible given currently available prototypes,
- exercise a wide range of computing environments and a broad scope of computing capabilities,
- examine a wide range of human-computer interactions, and
- demand and exercise a wide range of usability engineering methods and metrics.

In general, the test scenario entails:
- the review and retrieval of digital resources, on a variety of hardware devices,
- participation in a cluster of authorized devices that are connected, disconnected, and synchronized,
- the maintenance and management of the rights management policy of a variety of providers,
- migratory transactions,
- an opportunity to review structured and unstructured user interactions with the hardware and software.

A Taxonomy of Tasks

One of the by-products of this line of research was a taxonomy of user tasks, designed with the idea of driving towards the systematic and algorithmic choice of design points, and usability testing approaches, for newly identified devices, tasks, environments, and interfaces. Early-identified variables that we felt would be the building blocks of such a taxonomy included:
- whether the task is user-initiated or not,
- the primary goal (either accuracy or efficiency), and
- the point the device is used in the migratory transaction (beginning, middle, and end).

In this early work, we wanted to address a variety of tasks. Ultimately, we will derive a rich taxonomy, and have some rules about what the UI should look like, and what usability testing methods/metrics should be employed. Then, when new tasks/environments are identified, we will be able to make good guesses about how to design for them and test the designs.)

We came to refer to this thread as the pursuit of a “taskonomy.” The idea of taskonomy is to categorize different user tasks into various predefined categories to facilitate UI design and user testing. Usually pervasive computing tasks are not well defined and well structured tasks. A taskonomy can drive generalized task categories to which we can assign various current and future tasks as well as devices that participate in those tasks. As there is no specific set of usability methods that can be applied to these tasks, a method can be devised later that can be applied to all the tasks into a particular category.

In the process of modeling a taskonomy we felt it was important that we consider the connected nature of the pervasive computing environment and develop a model that can work as a platform for UI design and user testing tasks. Our main aim was to formalize user behavior in pervasive computing environments. We derived the following model to approach our taskonomy. This model addresses only the current functionalities of pervasive computing environment, and we expect to have to expand it as the PvC Lab expands its purview.

Step 1: Identification – identify all the objects in the model
- All the pervasive devices – car, oven, refrigerator, CD player, etc.
- All the task categories – user initiated, system initiated, single session, multiple session, etc.
- User scenarios – downloading music, heating food, setting up temperature, etc.

Step 2: Connection or inclusion – how devices are connected to other devices or included in a particular task category or scenario
- Device ➔ device
- Device ➔ task categories
- Device ➔ scenario

Step 3: Modeling – model the system in the following way:

Scenario 1 for a sample Task category

```
D1 ➔ D2

D3

D5 ➔ D4
```

Links between devices show sub scenarios (actions) for a particular scenario of a task category.
The above figure shows all the possible devices and sub scenarios or actions (as a connection between devices) that can take part in the main scenario for a particular task category. This way we can categorize different tasks for a particular scenario.

At this point in our work with the PvC there were two main problems with our taskonomy:

- How shall we define scenarios?
- How shall we define sub scenarios or actions?

This is just the preliminary object model that we can start with to achieve more complex models at later stages. Currently we are looking to include information from other sources to help us model the system better from an object-oriented point of view [2, 12].

At the end of our first year of work with the PvC we have proposed the following variables to populate our taskonomy:

- User vs. other initiated
- One vs. multiple devices
- One vs. multiple sessions
- Same vs. different device (for multiple sessions)
- Environment-dependent versus environment-independent tasks.
- Related vs. independent tasks.
- Domain areas (shelter, food, travel, entertainment, relationships).

In Appendix A is our first attempt at a taskonomy for the migratory transactions that might occur in a car.

Our final thread in this area is to consider a task description language to describe our identified cross-device tasks. We are deriving our direction from, among other articles, “Relieving Users from the Distractions of Ubiquity: Task-Centered Architectural Framework” [11]. The following is a quote:

“Computing environments of the future should satisfy two competing requirements:

- users should be able to take full advantage of the capabilities, devices and resources available at each location.
- users should be free to focus on their real tasks, rather than being distracted by dealing with the configuration and reconfiguration of computer systems to support those tasks.”

The following points summarize our current thinking on the taskonomy:

-We have considered only one person in all the cases. We can also distinguish tasks involving more than one person.
- Tasks can have priorities associated with them so at the time of simultaneous access high priority tasks must be completed before the low ones.
- We need to consider classifying task scenarios in various devices instead of only task-wise. For example we may take one scenario and talk about all the devices that can be a part of that scenario. For incorporating different devices into a particular scenario, we will build a template “object diagram.” (For example when a person is retrieving some data from the house what if that data is being shared by another pervasive device?) We can draw higher level object diagrams which can show communication among different devices.
- From the current flow charts, we need to establish a richer context around the scenarios and then flesh out these more descriptive decision charts.
- We see the taskonomy as an ever-developing, constantly refined set of transactions that expand or get revised based on new things we attempt in the lab. And as we make decisions about usability testing (and, indeed, UI) on one task, this will help steer decisions about usability testing and UIs on subsequently identified tasks in the same “cell.”

**Usability Methods**

In parallel with developing test scenarios, we worked with the PvC Lab representatives to begin to set up a state-of-the-art remote usability testing lab. Our goal was to employ off-the-shelf hardware and software, to enable the non-geographically-constrained usability testing of the various UIs. (Indeed, it may be the case that some IBM usability professionals in Florida or North Carolina should be involved, at this point, to help us test the value of the mote usability test set-up.)

There were two primary, equal, parallel goals of this effort:

1. To identify and fix usability problems in each of the UIs, and

2. To identify gaps in the usability testing capabilities afforded by the current tools, and identify
and create innovative remote usability testing procedures, tools, instrumentation, and metrics.

**Meta Usability Testing**

This meant that we had as a basic goal to conduct a usability test of the usability testing. So, while we wanted to be conducting real, industrial-strength usability evaluations of the emerging PvC Lab device UIs, we needed to evaluate our ability to evaluate them.

**Iterative (Spiral) Approach**

One of the hallmarks of user-centered design is iterative design: design, test, redesign, retest. Our goal was to pursue this approach as we investigated the xCP protocol as it applies to pervasive computing. Our goal was NOT to drive towards a complete system usability test, and one major product redesign, but rather to make incremental changes to the product and UI as each successive usability test is completed. Similarly, we wanted to make iterative and creative changes to the usability tool chest we employ and build.

**Results**

We were not able to complete the proof-of-concept usability testing of any actual devices. We were able to make some progress towards identifying base tools to employ. The School of Information team reviewed a variety of extant tool sets and determined that Ovo Labs had the tools most likely to afford us the ability to conduct remote usability testing of migratory transactions. The Ovo logging tool will allow the various PvC Lab devices to communicate with the usability lab. The question at hand is the ability of their tool to interpret the signal, to parse the information coming from the device, i.e., to do real-time event interpretation and logging. Another open question is who will do tailoring of the tool to afford this event logging (Ovo, IBM, UT, someone else), and how shall the relationship be structured. That question is still on the table as we consider using the Ovo Labs tool, building our own tool, or adopting a tool set developed by Karel Vredenburg’s User-Centered Design team at IBM-Toronto.

The following are the early requirements for the PvC usability lab:

- Video (including motion detection)
- Audio (including sound initiation of timing)
- Scan Conversion
- Behavioral coding
- Event logging (including all software events and interactions with all devices)
- Time sync’ing of all data
- Mixers
- One-way glass (on three sides?)
- Editing system (real-time and post-production)
- Cabling
- Console
- Intercom
- Sound proofing
- HVAC
- Video conferencing / WebCasting
- Remote Controls (pan-tilt-zoom, plus on/off of psychophysical and other sensors, plus two-way communication with ULab and test participants).

**A Prototype**

Not awaiting the completion of the lab, we developed a UI prototype for the control console in the PvC usability lab. Screenshots appear in Appendix B. We will want to usability test this UI, just as we would any UI, to ensure that it meets the needs of the local and remote usability test engineers conducting usability testing of migratory transactions.

Other labs we have reviewed, either in person or via the Internet, include, at UT:

- Accessibility Institute
- Behavioral Research Lab
- Visualization Laboratory,

and beyond UT:

- Ovo Labs
- Usability Sciences, Inc.
- Human Factors International
- University of Michigan
- University of Arizona.

**Two Challenges**

An empirical approach to design entails usability testing. And these usability test data will generalize to the real world setting only to the extent that we test representative users on representative tasks in representative environments. This is the first challenge for the usability engineering of pervasive computing. In a ubiquitous computing setting, those users, tasks, and environments are much broader and harder to monitor than is true with traditional computing.

The second challenge for ubiquitous computing is the increased transparency of HCIs.
• For the new pervasive systems to be successful, some (most?) of the human interactions with them will have to be automatic.
• It will violate the “representative environments” stricture to stop someone in the middle of a task and say “How do you feel about silently logging into your cluster”?
• And so we will need to get creative about the types of data we collect, and how we sync them up with other data.

Our early thoughts about the sorts of creative, psychophysical data we might collect from people carrying out migratory transactions include:
• Eye Tracking
• Galvanic Skin Responses
• Heart Rate
• Blood Volume
• Stress Parameters (cortisol)
• EEG
• Blood Flow

Regarding ubiquitous physiological monitoring, Dr. Emil Jovanov from The University of Alabama in Huntsville is doing research in wireless intelligent sensors [6]. He has developed a prototype for Wireless body area network of intelligent sensors (WISE). For personal medical monitoring. They are using all kind of body sensors for brain, heart, movement, etc. All these sensors (WISE clients) are controlled by a single server (WISE server), which can be connected to the internet. It is easy to imagine how we’d adapt this to collect usability test data. This would satisfy two of our requirements – testing independent from the device we are monitoring and remote testing.

5. Acknowledgements

We would like to thank Jesse Redman for help throughout our involvement.

6. References

7. Contact Information

If you have any questions about anything in this proposal, please don’t hesitate to contact:

Randolph G. Bias, Ph.D.
The School of Information
D7000
The University of Texas at Austin
Austin, TX 78712
rbias@ischool.utexas.edu
Office: 512-471-7046
Cell: 512-657-3924
Fax: 512-836-0845
## Appendix A: A Sample Taskonomy: Migratory Transactions in a Car

### Taskonomy Development - Car Focus

**UT-Pervasive Computing Usability Testing**

**AM Maynard w/Jesse Redman**

**Updated 7/14/04**

To start developing a Taskonomy, we start with the functionality of "the car..."

### Notes:
- Multiple Session - any time there is a break in which you do something else before the desired activity is completed

### Categories of Car Activities
- User Initiated
- Non-user Initiated
- Completed in
- Completed after
- Begins/Ends
- Begins on one device
- Ends on other device
- w/o aid of others
- w/ aid of others
- multiple devices implied

### Case: General Data Movement

<table>
<thead>
<tr>
<th>Data Movement</th>
<th>car -&gt; car</th>
<th>car -&gt; house</th>
<th>house -&gt; car</th>
<th>house -&gt; house</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>2.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>3.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>4.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>5.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

### Case: Car Diagnostic System

<table>
<thead>
<tr>
<th>Data Movement</th>
<th>car -&gt; car</th>
<th>car -&gt; house</th>
<th>house -&gt; car</th>
<th>house -&gt; car</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>7.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>8.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

### Case: General Data Movement

- **Requested Car Sys Diag Report while at home**
  - User Initiated: yes
  - Non-user Initiated: no
  - Completed in: yes
  - Completed after: yes
  - Begins/Ends: yes
  - Begins on one device: yes
  - Ends on other device: no
  - w/o aid of others: yes
  - w/ aid of others: no
  - multiple devices implied: no

- **Unsolicited Car Sys Diag Report while at home**
  - User Initiated: no
  - Non-user Initiated: yes
  - Completed in: yes
  - Completed after: yes
  - Begins/Ends: yes
  - Begins on one device: yes
  - Ends on other device: no
  - w/o aid of others: yes
  - w/ aid of others: no
  - multiple devices implied: no

- **Requested Car Sys Diag Report while in car**
  - User Initiated: yes
  - Non-user Initiated: no
  - Completed in: yes
  - Completed after: yes
  - Begins/Ends: yes
  - Begins on one device: yes
  - Ends on other device: no
  - w/o aid of others: yes
  - w/ aid of others: no
  - multiple devices implied: no

- **Unsolicited Car Sys Diag Report while in car**
  - User Initiated: no
  - Non-user Initiated: yes
  - Completed in: yes
  - Completed after: yes
  - Begins/Ends: yes
  - Begins on one device: yes
  - Ends on other device: no
  - w/o aid of others: yes
  - w/ aid of others: no
  - multiple devices implied: no
Appendix B: Screenshots for a Usability Test Console UI

1. The top navigation will guide the tester through the process of setting up testing preferences. The first stage of setting up a test would likely be to select room(s) for testing. During this phase the first block would be highlighted and the others would be more faint. As more and more settings are added to this navigation bar, the progress will either be by color changes or some other identifying mark.

2. These side panels will reflect the selections that the tester makes as they are setting things up. This example shows several different values as if someone had already gone in and made a number of different selections.

3. The center display is where all of the selections are made. This example shows how it might look when the tester initially goes into to select a room. The tester may want to select a primary room and then other rooms that might activate when the user selects them. Once a room is selected, the tester would use the navigation at the top to begin setting up the different cameras, devices and microphones in the rooms. For each room selected, the center panel would display graphically the different cameras, devices, microphones, etc. that could be enabled. For example, if the tester selected the living room and then moved onto devices in the top navigation, the center panel would then show all the devices in that room.