Ubiquitous Computing

"We are trying to conceive a new way of thinking about computers, one that takes into account the human world and allows the computers themselves to vanish into the background"

--Mark Weiser

"Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings" by Elin Ronby Pedersen, Kim McCall, Thomas P. Moran and Frank G. Halasz

- Tivoli: An electronic whiteboard application designed to support informal workgroup meetings.
- At the time, most of the early attempts to build computer support for meeting rooms used large whiteboard sized displays. However, these displays were controlled with a keyboard and a mouse.
- Informal workgroup meetings are collaborative in nature. Previous research has shown that the physical actions around a shared drawing surface tend to be as important as the actual marks made on the surface itself.

Goals of the Tivoli Project

- To discover and explore the user interface techniques appropriate to this new kind of technology.
- To discover and implement the functionality needed to actually support small working meetings at the Liveboard.

Nature of the User Interface

- It simulated a whiteboard by allowing freehand drawing and erasing.
- In order to be successful:
  - User interaction with the board should be unselfconscious
  - The interaction should be fluid
- The Liveboard fits in with the authors' vision of ubiquitous computing as it appears to be a familiar whiteboard.
  - It should initially behave like a normal whiteboard so that first time users can use it immediately.

Functionality for Small Working Meetings

- Small working meetings
  - The participants work closely together on a problem or idea.
  - The working group is usually less than 8 people.
  - There is no facilitator or scribe, so all participants can have access to the board.
- Core functionality:

- Thrusts
  - remote collaboration
  - Meeting management tools
  - Integration with other ubiquitous devices

Design Features and Issues

- Pen vs. Mouse
  - Gestural techniques are more natural for a pen than for a mouse.
  - A pen is a direct-control device, and a mouse is an indirect-control pointing device.
  - A cursor may not be needed with a pen input device.

- Very Large Interactive Surface
  - Adjust the size and placement of pop-up messages to make sure that the user always notices them.
  - Also, the tools such as buttons and menus have potential to be physically out of the user's reach.

Design Features and Issues (Continued)

- Multiple Users and Pens
  - The Liveboard is designed to have 3 pens to accommodate multiple users
  - Gestural actions can be localized to specific pens and do not have to rely on shared on-screen tools.
  - Pen state vs. system state
    - Undo is restricted to actions performed with a specific pen.
    - Erase can be applied to strokes performed with any pen.

- Atomic Objects
  - Drawings are represented as stroke objects.
  - As the eraser goes over each stroke, the whole stroke disappears.

Design Features and Issues (Continued)

- Generalized Wiping
  - The wiping motion dynamically selects objects as the wiper passes over them.
  - Erasing is performed by wiping over the surface with the pen in erase mode.

- Gestures
  - A gesture is a stroke that is not taken to be an element of the drawing, but it is interpreted as a command.
  - The most common gesture is the selection gesture, and subsequent gestures then operate on the selection.

Design Features and Issues (Continued)

- Gestures (Continued)
  - Design issue: How do you indicate that a stroke is a gesture?

Status and Future Directions

- Multi-Site Tivoli
  - "Our vision is to make Tivoli a true collaborative tool, supporting both multiple users standing shoulder-to-shoulder around a single Liveboard and multiple users utilizing two or more geographically-separated Liveboards."
- The Casual User
  - The feedback suggests that the current design favors increased functionality over intuitiveness.
- Support for Specific Meeting Practices
  - Tivoli is often used during brainstorming meetings. Can Tivoli be expanded to support activities such as list-making, categorization, and voting without altering its fundamental white-board-like nature?

Speech Synthesis

- Synthetic text-to-speech technology is efficient and more general than the computer playback of a recorded human voice.
  - It is generally more expensive to splice together recordings of a real human voice. However, a company may choose to use recordings if high-quality natural sounding speech is wanted.
  - A sentence written in ASCII is more compact than a recorded sentence.
  - A system that uses text-to-speech conversion is more flexible than one made up of prerecorded messages.

"Text-To-Speech Conversion Technology" by Michael H. O'Malley

- Text-to-Speech conversion transforms linguistic information stored as data or text into speech
- Applications
  - It is very useful in audio reading devices for blind people.
  - It can also be used to convert electronic mail to voice mail for access by phone.
  - It can permit access to large text databases (like parts inventories) by phone.

Text-To-Speech Conversion Technology

- Two major elements
  - A module with text as the input and with an output that drives the sound-generating mechanism
  - The sound-generating mechanism
- Difficulties
  - The vocal tract is incredibly complex.
  - The current models are able to produce intelligible speech. However, they cannot produce speech that is indistinguishable from human speech.
  - Models that are based on the physical shape and the physiology of the sound production mechanism are difficult to use.

Text Normalization

- Almost all texts contain some sort of symbolic material such as numbers, abbreviations, acronyms, and information signaled by graphic layout.
- The symbolic information needs to be converted into a standard text format.

Word Pronunciation

- "English has by far the most complex relationship between spelling and word pronunciation of any alphabetic language."
- A dictionary needs to be combined with a sophisticated set of pronunciation rules.
  - A dictionary is needed to handle exceptions. However, a large dictionary is not enough on its own.
  - An accurate text-to-speech conversion system needs to be programmed with several thousand rules in order to handle English.

Prosodies

- Prosodic component of speech: rhythm, intonation, and the emphasis or de-emphasis of particular words.
  - There may be many valid prosodic interpretations. However, there are also many that would not be considered natural.
  - Often emphasized words carry the information focus of the message
  - Word stress patterns may indicate that a word is a part of a multiword compound
  - Conversations usually have an elaborate prosodic component, while speech read from a text generally provides more regular prosodic information.
  - Prosodies prove to the be most difficult part of language for a text-to-speech conversion system.

Phonetic Rules

- Phonemes are the symbols used in dictionaries which represent word pronunciation.
- The phonemes are the inputs of the phonetic rule module, and the output is a detailed description of the sounds of the utterance.
  - In this module, the duration of each phoneme is assigned according the the context of the text.
  - Other rules might remove parts of phonemes.

Voice Tables

- The voice table module converts the detailed phonetic description of an utterance into numeric targets for use by the voice model.
- The voice tables handle differences in bandwidth.
  - Speech researchers usually assume a minimum speech bandwidth of about 5-6 kHz
  - Telephone system designers traditionally use a bandwidth of 3.5 kHz
- Therefore, the author suggests using different sets of voice tables for different frequency ranges.
- High-quality text-to-speech systems usually score around 95% for voice intelligibility.
- Low-quality text-to-speech systems usually score around 60-75%.

Development Tools

- Development tools are "tools that provide programming support for implementing interactive systems."
- The toolkit should minimize the effort that it takes to translate the interface design to the format necessary for the toolkit.
- A tool should be integrated into the interface development process.
- An alternate definition:
  - "tools that help a developer convert interface specifications into an interactive system, and that support all phases of system refinement including prototyping, implementation, testing, maintenance, and system enhancement."

"GROUPKIT: A Groupware Toolkit for Building Real-Time Conferencing Applications" by Mark Roseman and Saul Greenberg

- Conventional toolkits are often inadequate for groupware.

- GROUPKIT is a groupware toolkit for real-time distributed work.
- GROUPKIT assists in constructing real-time work surfaces.
  - A work surface is a shared visual environment where one user's actions are made immediately visible to other users.

- Toolkit Design Requirements
  - Human-centered design requirements
  - Programmer-centered design requirements

Human-centered design requirements

- Supporting multi-user actions over a visual work surface
  - Provide support for gesturing
  - Provide support for graphical annotation

Human-centered design requirements

- Structuring group processes during a meeting
  - Provide various floor control policies
    - Floor control policies provide a way to mediate access to shared work items.
  - Support different registration methods
  - Support latecomers into the conference

Human-centered design requirements

- Integration with conventional ways of doing work
  - Integrate other forms of communication
  - Allow use of single-user applications
    - Groupware should not create a barrier between "individual" and "group" ways of doing work.

Programmer-centered design requirements

- Technical support of multiple and distributed processes
  - Provide processes for basic conference management
  - Provide a robust communications infrastructure
  - Provide support for persistent sessions

Programmer-centered design requirements

- Technical support of a graphics model
  - Provide primitives to a shared graphics library
  - Provide object concurrency control
  - Separate the view of an object from its underlying representation

Wednesday, July 28th

- Assignment 4 Parts I & II due at the first break of class.
- Readings
  - Article 22: Integration of Inter-Personal Space and Shared Workspace: ClearBoard Design and Experiments by Hiroshi Ishii, Minoru Kobayashi and Jonathan Grudin
  - Agents that Reduce Work and Information Overload by Pattie Maes