Impromptu Thin Client

Extending Desktop Applications with Pocket-size Devices

Motivation

Our research has focused on the design of systems to support usable security. By relying on event-based monitoring, visualizations, and the integration of configuration and action in the development of applications, our goal is to create a technical infrastructure which makes visible the configuration, activity, and implications of available security mechanisms, thereby allowing end users to make informed security choices resulting in increased effective security. Those principles have been used in the design and implementation of a peer-to-peer file sharing application called Impromptu (Figure 2), that works as a test bed where different visualizations have been devised.

In a collaborative scenario, file sharing is not a means in itself, but is used together with other tools. In our early trials with Impromptu, we noted that these other applications would sometimes obscure the Impromptu user interface, making it harder to notice changes and updates. To this extent, we have been experimenting with pocket-size PCs, or Personal Digital Assistants (PDAs for short) as devices for augmenting desktop applications functionality. The key insight is that those down-sized computers can be used to augment existing desktop applications, by providing downsized interfaces and visualizations.

The Impromptu desktop application

The Impromptu prototype (Figure 1) is a peer-to-peer file sharing tool designed to support co-located and ad-hoc meetings of small groups. Users participate mainly by sharing files according to different visibility levels expressed as concentric circles in a virtual table.

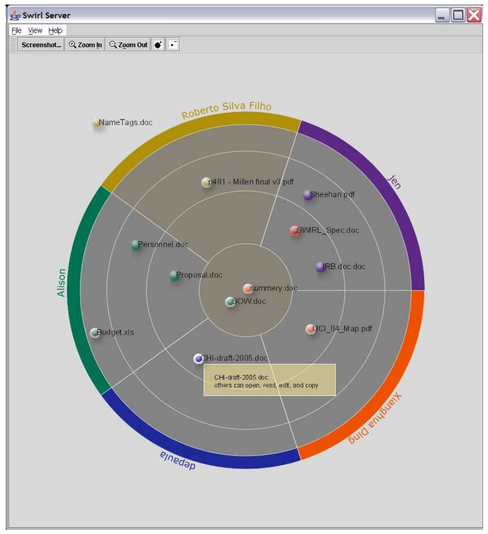


Figure 1: Impromptu desktop interface

Design Principles

As a consequence to the Pocket PC limitations, in our design, we focused on presenting relevant events form the desktop application; while direct manipulation of artifacts (reading, writing and manipulating files), available in the desktop tool, were not supported. The GUI was implemented using more conventional AWT widgets. The result is a visualization pad where events can be viewed.

The Thin Client Architecture

The thin client architecture (see Figure 3) relies on the same event-based design as the desktop tool. It uses YANCEES, the publish/subscribe infrastructure collects and routes events from the virtual group file repository. By listening to events generated by this application, the thin client builds its own view of the distributed system, and keeps itself consistent with the desktop application.

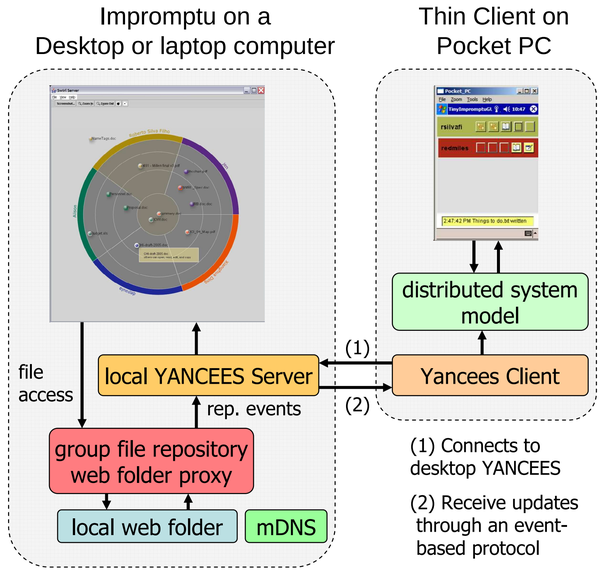
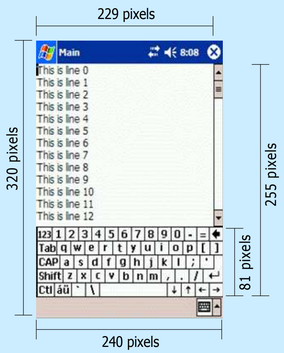


Figure 3: The Thin Client Architecture

DISPLAY AND INTERFACE:
- 3.5" Quarter VGA TFT color 16-bit, touch sensitive
- 320 pixels wide by 240 pixels high
- 229 x 255 pixels of usable screen
- 81x240 keyboard
microphone and headphone



SYSTEM:
- PXA270 single board computer
- 312MHz Intel® XScale Processor
- 64MB SDRAM plus 64MB ROM (for applications)
- Integrated 802.11b, Bluetooth and infrared
- Windows Mobile 2003 SE

Pocket PC Technical Limitations

In the design and implementation of our prototype (a.k.a. thin client), many factors impacted in the capabilities of the application and in the design of the interface. Those main factors are (see Figure 2):

- limited processing and memory
- limited input and screen size
- programming language and runtime constraints

Figure 2: Pocket PC device characteristics

Thin Client - Ticker Tape Interface

- Icons represent events generated from the Impromptu application.
- Events are displayed according to a temporal line (from right to left), as they get produced.
- By clicking on the event icon, the event detail is displayed in the bottom of the screen.
- Previous event can also be examined by clicking on its iconic representation.

This interface allows the easier visualization of the desktop application events. The goal is to drive the attention to the users back to the desktop tool, where direct file manipulation can be done.

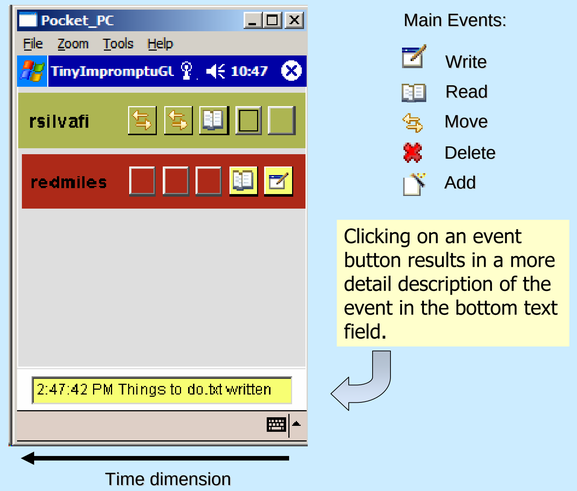


Figure 4: The Ticker Tape Interface

Contact Information
Professor David F. Redmiles
Roberto S. Silva Filho
Institute for Software Research
University of California
Irvine, California 92697-3425
{redmiles, rsilvafi}@ics.uci.edu
949-824-{3823, 4121}

To learn more about Impromptu and have access to a prototype of the system and documentation, please visit the website:
http://www.isr.uci.edu/projects/swirl

Effort sponsored by This work was supported in part by the National Science Foundation under awards 0534775, 0541462, 0133749, 0205724, 0326105, 0527729, and 0524033, and a grant from Intel Corporation.

June 2006