Towards an Adaptive and Context Aware Laptop

Conference Paper · September 2001

1 author:

Steve Loughran
Apache Software Foundation

Towards an Adaptive and Context Aware Laptop

Steve Loughran
Publishing Systems and Solutions Laboratory
HP Laboratories Bristol
HPL-2001-158
June 29th , 2001*

E-mail: slo@hplb.hpl.hp.com

laptop computing, power management, context awareness, ubiquitous computing

This paper describes a prototype application for notebook computers which infers its current context and adapts some aspects of the system -including the power policy and web proxy settings- to that context. In this manner the computer adapts to the different use situations it encounters. The prototype is limited in that it does not learn to predict user contexts, and, in the absence of good network identification technologies, context detection is fragile. We propose some improvements in the detection of context and mechanisms to support dynamic system adaptation.

* Internal Accession Date Only Approved for External Publication

Towards an Adaptive and Context Aware Laptop

Steve Loughran
Hewlett Packard Laboratories Bristol
Filton Road, Stoke Gifford,
Bristol BS34 8QZ, UK
+44 (117) 979-9910
slo@hplb.hpl.hp.com

ABSTRACT

This paper describes a prototype application for notebook computers which infers its current context and adapts some aspects of the system –including the power policy and web proxy settings- to that context. In this manner the computer adapts to the different use situations it encounters.

The prototype is limited in that it does not learn to predict user contexts, and, in the absence of good network identification technologies, context detection is fragile. We propose some improvements in the detection of context and mechanisms to support dynamic system adaptation.

INTRODUCTION

A notebook computer fulfils a multitude of roles. It is a performance workstation, a home PC, a lightweight communications tool for meetings and an energy efficient communications and computing tool for the road. A single notebook can either be a compromise between all these requirements, or it can adapt to optimise different roles. With docking stations and expansion modules a laptop can adjust its hardware configuration. However, the opportunities for software adaptation are still being explored.

In 1999 we conducted an experiment to log the use of notebook computers used as 'desktop replacements' -the users of which chose a notebook as their main computing device, even though they were rarely mobile [1]. One of the key observations of this study was that users seemed to operate in a number of distinct contexts, the four main contexts being shown in table 1.

Location	**Power**	**Network**	**Tasks**
Desk	AC	LAN	Office, email
Meeting	DC	none/LAN	Email, attachments
Home	either	modem	Email, office, play
Travel	DC	none/modem/LAN	Email, office

Table 1. Notebook operation in the four contexts

We concluded that these users would benefit from better notification of power and network state, and that they may also benefit from a notebook that could determine its operational context and adapt its behaviour appropriately.

Context awareness and adaptive behaviour in mobile devices has been explored before, such as in mobile telephones and PDAs [2], but little of this work has been transferred to the laptop domain. Notebooks do exhibit rudimentary adaptation in that Windows can be configured with multiple power schemes, each of which has different settings for AC and battery power. However, the settings are rudimentary and the selection is manual. There is no evidence that users do manually select power schemes -so effective adaptation is limited to the power source dependent options of a single policy.

Two commercial products have explored adaptive contexts in a notebook. The Phoenix PowerPanel is an application intended to maximise battery life. The user can manually select an 'application profile', and the tool tunes the system for efficient support of that profile. Different profiles are defined for word processing, spreadsheet, presentation, games and communications, and the tool will dynamically choose the policy appropriate to the application in the foreground. This can save power by disabling unneeded system components, although with the advent of notebooks with 'Advanced Configuration and Power Interface' (ACPI) BIOSes, this is now handled by the OS, which can enable and disable system devices based on demand [3]. What remains is to provide the cues as to which policy to pursue –performance or economy.

Symantec's *Mobile Essentials* [4] is an older product which adapted a laptops' network settings to different locations. It was primarily a tool to allow a notebook to move from one site to another, the user being required to reboot then indicate their location and hence configuration. This slow process of switching locations meant that it was not suited to rapid changes of context. The product is now discontinued.

Some component manufacturers have produced components with 'self-throttling' behaviour to save power when their workload is low –such as IBM mobile hard disks [5]. IBM supplies an application to control this –and disable any throttling when under AC [6]. This tool shows that controlling self-throttling devices is possible, but it does not explore when best to alter the settings.

To summarise, past notebook adaptation has primarily controlled power settings based on the power source- AC or battery, or the nature of the foreground application. One ‘location based’ tool has focused on network and printer settings, but not power management –and no software has explored dynamically adapting the system to ‘user contexts’. We felt that there were opportunities for improvement.

THE PROTOTYPE

We set out to build a proof-of-concept program to distinguish the contexts observed in the user logging experiment, and adapt the notebook appropriately. To explore what could be supported in today’s systems, one requirement was “no new infrastructure” –an off-the-shelf notebook was to be used without adding anything to the environment to provide location data.

The core adjustment would be to the power policy of the underlying operating system, switching between predefined policies. One policy would aim for responsiveness by blanking the screen on idleness, keeping the system powered up, while another policy would save power by hibernating –copying the memory image to disk and powering off- after a few minutes of idleness.

Other system aspects that were considered were: -

- Sound: volume and mute settings.
- Screen saver: disabled for economy and presentations.
- Self-throttling devices: responsive versus economical.
- Processor clock speed: fast versus economical.
- Network settings– web proxy, frequency of mail server synchronisation.
- Printer: which printer to make the default.

Some of the settings –web proxy and printer- were more appropriate to locations than context, but we chose to support them to see whether they were beneficial or not.

We did not manipulate self-throttling devices in the prototype, but that is something a production system should consider as it could lead to power savings or enhanced responsiveness. For the prototype it would have been time consuming to implement.

Processor clock speed is an interesting example of a power/performance tradeoff –the final 25-33% of CPU clock speedup on a Intel SpeedStep™ CPU doubles the power consumption [7]. Again, this is a low level item to control, which should be considered in the future as the benefits may be great.

Five contexts were supported

- *Desk*: desk printer and proxy; focus on performance
- *Meeting:* desktop performance with silence and responsiveness.
- *Home:* home printer and proxy; focus on performance
- *Travel*: aggressive power saving.
- *Presentation*: no screen saver or screen blanking and sleep only after many minutes of idleness.

Most of the contexts focus on performance and responsiveness rather than power efficiency, as it was felt that was actually of more value than power saving. This marks a difference from classic notebook adaptation software, which seek economy whenever the notebook is mobile.

The presentation context was not explicitly noted in the original use experiment, but was added because to configure a notebook for the successful display of a presentation requires many subtle changes in configuration, changes our tool could implement.

Context Selection Algorithm

An early version offered a manual choice of context only, but after distributing it to the users of our notebook study, we saw little evidence of context selection. Automatic detection and selection of context was added, to see if this could be done discreetly enough. The rule set was quite simple, using time, network and power as the primary determinants: -

- *Home* in evenings (7pm onwards) and weekends.
- *Desk* during weekday hours when on AC and LAN
- *Meeting* during weekdays when not at one’s desk.
- *Travel* when manually chosen or when the battery level dropped below 40%.
- *Presentation* when manually chosen or when PowerPoint was detected running in presentation mode.

The context selection was made whenever the system resumed, the network state changed, and every ten minutes thereafter. A manual selection of context disabled the automatic selection for thirty minutes.

The final iteration used network assigned IP addresses as extra cues, which are often effective at differentiating work, home and travel contexts. The other cues are still essential to detecting the many in-office contexts which network addresses do not differentiate.

User Interface

The application appears as a small horizontal or vertical window, displaying power, network and context details. The power area displays power source, battery state and minutes remaining. The network state uses the signal strength metaphor, showing more strength and changing colour when moving from dial-up to LAN. A single red dot indicates no connection. The network state icon is also shown in the in the Windows 'system tray', with a popup 'tooltip' listing network address and battery minutes left.

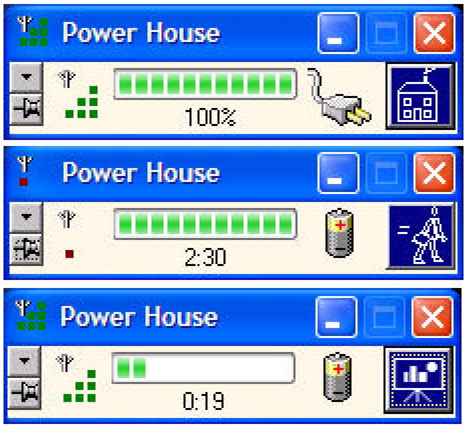

Figure 1: context application in different states

The final icon displays the current context and is also a drop-down list of all the available contexts. This enables a user to manually select a context in the event that the estimated context is not appropriate.

ADAPTATION IN PRACTICE

A version of the software was distributed internally to fifteen desktop replacement users for a period of two months. Qualitative feedback was recorded after this period. This version of the software did not use network addresses as context cues, or the automated detection of PowerPoint slideshows –which meant that the detection process was of limited accuracy.

The simple display of network state and remaining time was appreciated, although some felt that the application was too intrusive. They would have preferred a version integrated with the windows task bar.

A common request was for dynamic switching of proxy server settings between home and work –this need was driven by broadband Internet access to the US homes. Supporting this feature satisfied the day-to-day adaptation needs of these users.

Context based printer selection received a mixed response. Printers are associated with a location, not a context, except when two are synonymous. Network device location technologies seem more appropriate for dynamic printer selection.

Context based adaptation appealed to the users as a simple metaphor for explaining how the notebook could adapt to their needs. However, users did not indicate they manually selected contexts on a regular basis. Either the automated detection process was adequate for most purposes, or the limited adaptation of the system between contexts meant that context selection was not viewed as particularly relevant.

The overall conclusion from the user feedback was that better state display was the primary value of the tool for many users; network and battery state being the key items. Evaluating the latest version of the software with a broader cross-section of users would help to assess the commercial value of the concept.

CONCLUSIONS AND FUTURE WORK

We have shown that it is feasible to adapt many aspects of modern notebooks to better suit the observed contexts of desktop replacement users and have begun to experiment with automatic context detection.

We have also demonstrated that users value an improved display of power and network status.

What we have not shown is that users significantly value context-based adaptation, at least within the range of customisation of our early prototypes.

The adaptation process could be expanded to alter many more aspects of system behaviour. Manipulating self-throttling devices and the CPU clock speed are both obvious options, but the concept of contexts seems to be of equal benefit to higher-level applications. For example, context could determine the message synchronisation policy of an email program or the notification policy of an instant messenger application. For a consistent user experience, the adaptation of these applications must be integrated, so that adaptation is uniform and manageable.

The other area to address is better context detection. Using more network parameters as cue is an obvious next step. Adding a location beacon to the LAN, perhaps as an extension of DHCP or UPNP services would enable roaming devices to distinguish locations and adapt to new ones. These software beacons could co-exist with physical beacons, such as CoolTown beacons [8], to provide physical and network location data to roaming devices.

A truly adaptive notebook should not only derive its context from present information, it should be able to predict future use based on past experience. Our notebook logging study showed that, apart from travel, user behaviour was often consistent on a daily and weekly basis. Logging and modelling user behaviour would provide more accuracy to context estimates and enable predictive actions to take place. For example, it could be used as a cue to determine when best to synchronise a notebook with central email and file servers, or which power policies to apply in advance of real need. Equally importantly, the system could determine what user preferences in different contexts actually were by watching the changes they manually made to the system. This would make customisation to users and contexts a background activity, and enable the system to derive new contexts that truly reflect the use of the system.

To close, notebook computers have the mobility and complexity to merit adaptation and their operating systems permit adaptation in software without significant difficulty. It is our belief, therefore, that further research in the area of context prediction and device adaptation should target these computers as a research and delivery platform.

REFERENCES

[1] Loughran, The Secret Life of Notebooks, Proceedings CHI 2000.

[2] Schmidt et al, Advanced Interaction in Context; Proceedings Handheld and Ubiquitous Computing, September 1999.

[3] Microsoft, OnNow, http://www.microsoft.com/hwdev/OnNow/

[4] Symantec Mobile Essentials www.symantec.com

[5] IBM, Adaptive Power Management for Mobile Hard Drives, http://www.almaden.ibm.com/almaden/mobile_hard_drives.html

[6] IBM, IBM Power Booster http://www.alphaworks.ibm.com/tech/powerbooster

[7] Mobile Intel Pentium III Processor featuring Intel SpeedStep™ Technology Performance Brief

[8] Kindberg and Barton, A Web-Based Nomadic Computing System, http://cooltown.hp.com/papers/nomadic/nomadic.htm