

GPE – G Palmtop Environment

The Miniature Desktop

The GPE project is aimed at producing a desktop environment for Linux PDAs and addresses itself to both users and developers. Special emphasis has been placed on compatibility and compliance with current open standards. This article tells you what users can look forward to.

BY NILS FAERBER

Even though Linux is nothing out of the ordinary on today's PDAs, it is still a world apart from "normal" Linux. A small PDA just does not have the facilities that a standard desktop or server machine can offer. Its memory is limited, both with respect to RAM and to hard disk capacity. The PDA will tend to have a low-resolution display and its interfaces boil down to a few keys and/or stylus and touchscreen setup. It is hardly surprising that systems of this type need specially tailored software.

This is precisely what the GPE project [1] aims to provide, an environment that reflects both the limitations of the PDA without losing sight of the user's requirements. The project is targeted both at users and developers. Users can look forward to a software system that meets their requirements and provides genuine benefits. Developers can take advantage of a consistent system that facilitates software development.

Bottom Up

GPE is based on common standards. Well-known principles and functionality make the PDA more accessible both to developers and users. After lengthy discussions and evaluations, X11, or XFree86 [2] to be more precise, was cho-



sen as the GUI. The "kdrive" – or "TinyX" – X server proved to be powerful despite its astonishingly small memory footprint.

Some hard work on the part of the XFree86 teams, particularly Keith Packard and Jim Gettys, removed superfluous ancestral ballast from the X server and the accompanying libraries. The result is a powerful graphics server with TrueType font support that works extremely well on embedded systems. XFree86 also provided the advantage of network transparency for the complete GUI desktop – more or less as a side-benefit, and this makes using a PDA a whole new ball game.

GTK+ [3] was chosen as the toolkit and widget set. Version 1.2 of GTK+ provided a powerful approach to developing X11 applications. Due to the pervasiveness of GTK+ – and this is mainly attributable to the GIMP and GNOME projects, GTK+ has advanced to the status of a de facto standard. In the meantime GTK+ 1.2 has moved on to GTK+ 2.2.

This transition saw the addition of a wide variety of new and valuable functions. These include vastly improved font displays (anti-aliasing) and more flexible character setting which allow for later migrations to language families with completely different character sets such as Arabic, Hebrew or various Asiatic languages.

Data storage is another fundamental topic besides GUI output. PDAs normally have a fairly small non-volatile memory – typically some kind of flash memory. In contrast to hard disks or RAM this kind of memory cannot be infinitely wiped and re-written.

PDA applications need to consider this and minimize the number and scale of write operations. This is why GPE applications use the SQLite relational database [4] where this makes sense.

Using an SQL database for data storage offers a number of advantages. In SQLite's case an efficient B-Tree structure is used as a repository. This minimizes the amount of data written to the storage device in case of modifications. Additionally, applications benefit from simplified data manipulation.

As the SQL database performs complex operations, this means that they will not need to be re-invented for each new application. Finally, this separation of application logic and data storage means that the data storage method, SQLite in this case, can be exchanged with a minimum of effort at a later stage.

The last fundamental decision that needed to be taken was to apply the Linux multiuser concept to the PDA.



Figure 1: GPE login



Figure 2: GPE desktop view

This necessitates a somewhat strange login procedure that users will no doubt appreciate later (see Figure 1).

Most PDAs are used to store sensitive private information sooner or later. The login protects this data from unauthorized access, and even allows a “guest” to use your PDA without endangering your data by providing different user credentials. A profile is created for each user, and users can specify preferences, just like on a fully-fledged Linux system.

Up and Running!

The user-specific environment is launched after logging on. The Matchbox [5] window manager is the most important component here, as it expands application windows to fill out the screen – or at least almost, as a status

bar at the bottom of the screen and a header line are automatically kept clear (see Figure 2).

This behavior means that the application window can never be larger than the viewable area. This facility avoids juggling with windows – this does not make sense on a PDA anyway. In addition to the window manager, a status bar management tool and a desktop manager are also launched.

The desktop manager allows the user to access a well-organized GUI-based selection of the installed applications and to launch applications with a single click. To avoid confusion, most programs can only be launched once – again the desktop manager takes care of this. If a user attempts to launch an additional instance of a running program, the desktop manager simply switches to the active program.

This and other functions of the window and desktop manager are based on the open and free standards of the Freedesktop project [6].

What Can I Run on It?

In addition to the underlying functions for the GUI desktop and data storage, and the desktop manager of course, a PDA will tend to provide Personal Information Management applications, or PIM for short. However, they have been completely reworked for GPE.

The calendar provides appointment management facilities and can assign alarms to appointments (Figure 3). The

at daemon is responsible for alarm management, just like on a fully-fledged Linux system. Acoustic and optical signals occur when an alarm expires.

The contact management application follows the pattern of typical PDA address management tools. Work is now in progress on intelligent vCard import [7] facilities, which will allow the user to simply import complete phone and address books from any other device, such as cellular phones.

Let’s just look at a few more examples of applications – more are being released more or less daily. The entries in the to-do list can be categorized (see Figure 5).

Figment (see Figure 6) is a practical variant of the to-do list that allows the user to organize memos in a more usable tree structure and navigate the structure. This is an extremely practical way of getting organized.

The coding of the numerous configuration programs was equally as important as developing applications. As PDAs cannot really be managed via the console, you need a GUI application for each setting. Figure 7 shows a selection of GPE configuration applications.

Synchronization? Teleport!

The killer issue for Linux PDAs is the question of synchronization. But let’s be honest: What do we really expect to gain from synchronizing PDAs and desktops? Is it really the synchronization aspect itself – the replication of two or more potentially different data sets?

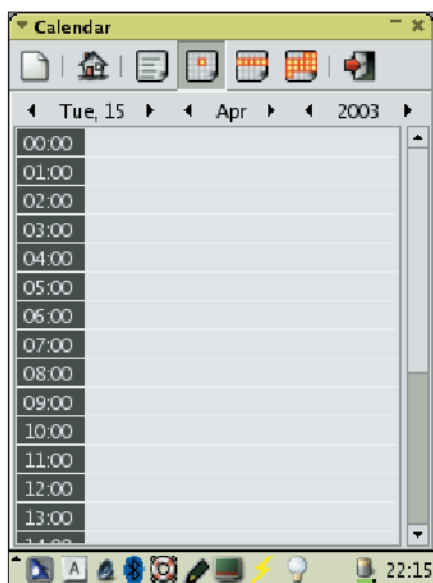


Figure 3: The Calendar Application

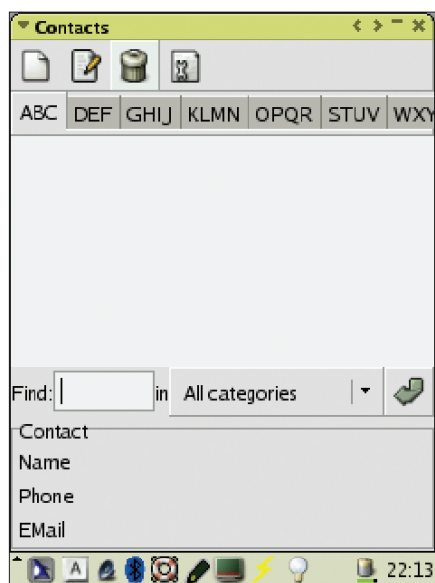


Figure 4: Contact Management on GPE



Figure 5: The to-do list

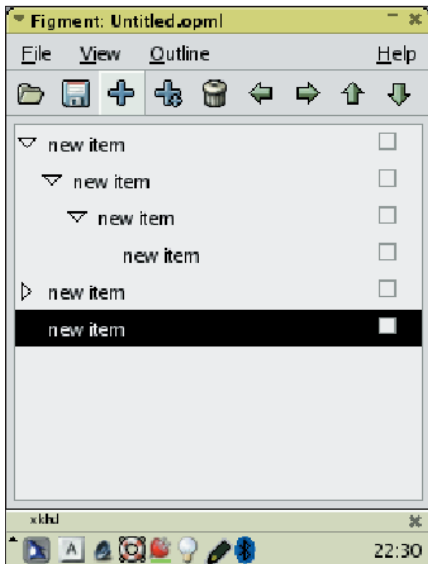


Figure 6: Figment - a super To Do list

Or is it simply wanting to avoid the insufficiencies of today's PDAs, that is, that data entry on these devices is a pain, and that one would prefer to use a PC with a larger, more readable screen and a decent keyboard?

GPE does not provide synchronization facilities at present. Current approaches are simply too disparate to allow a de facto standard to emerge. Elegant solutions to the data entry problem are a different issue altogether: why not simply migrate the PDA's display to a larger desktop system, complete your work at the desktop in comfort and then send it back to the PDA?

No problem with GPE. The Teleport tool exports a running application from

the PDA to any other display, just as if you had launched the application on the other system. The major difference is the fact that you can export an application that is already running.

The future bodes well thanks to migration support in GTK 2, and you can expect to hear a lot more about this feature. The application is informed when you switch to another display. If you move from the PDA to the desktop, the application can use a whole range of new facilities – the screen is larger, you have a real keyboard, and a mouse with multiple buttons, instead of a touch-screen, more color-depth on the display, and lots more.

In future, applications will be able to react to any newly discovered resources automatically, by displaying extended menus with more functions, and allowing completely new views of your data. The whole desktop will then correctly re-configure automatically!

As the USB port is used as a network device, this automatically happens when you attach a PDA to the desktop. This would allow any applications that happen to be running on your PDA to automatically migrate to the desktop, and to move back when you detach the PDA! This functionality would also work on Wireless LAN or Bluetooth technologies.

This does not mean that work on synchronization has come to a standstill; you can look forward to having more than one way to move your data to your PDA in future.

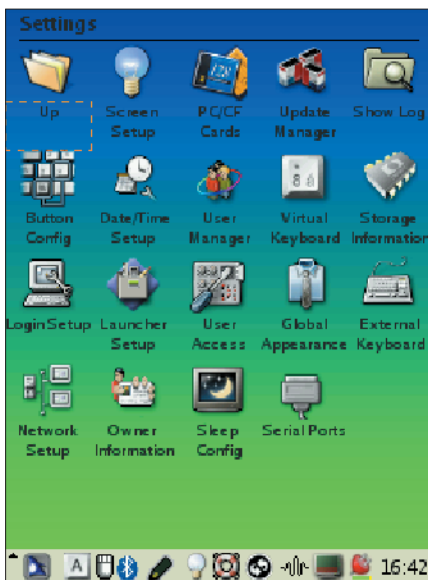


Figure 7: GPE Configuration

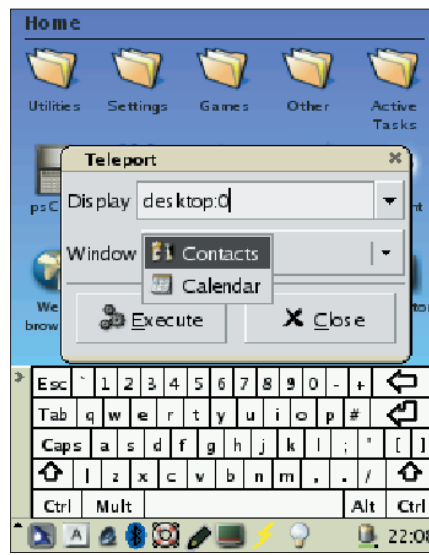


Figure 8: Teleport exporting a program window to a PC display

Future

At present a lot of hard work is going into completing version 1.0. The iPAQ Linux distribution, "Familiar" [8], contains a current version of GPE. Some enhancements can obviously be expected in future versions. The dynamic configuration of applications at runtime by means of so-called X settings [9] is under development.

Applications that use the SQL database are due to be ported to a GPE specific SQL server to provide more security. SQL servers could even allow applications to access your PDA data via the network at some stage.

This opens up a whole new realm of possibilities: In the near future D-Bus [10] will assume responsibility for the interprocess communication between GPE applications, thus providing for seamless integration of the programs involved; a typical example might be an appointment with someone in the contacts database. ■

INFO

- [1] GPE, "G" Palmtop Environment: <http://gpe.handhelds.org>
- [2] XFree86: <http://www.xfree86.org/>
- [3] GTK+, The GIMP Toolkit: <http://www.gtk.org/>
- [4] SQLite, An Embeddable SQL Database Engine: <http://www.sqlite.org/>
- [5] Matchbox Window-Manager: <http://handhelds.org/~mallum/matchbox/>
- [6] Standards for the desktop, Freedesktop.org: <http://www.freedesktop.org/>
- [7] vCard and vCalendar standard: <http://www.imc.org/pdi/>
- [8] Familiar PDA Linux Distribution: <http://familiar.handhelds.org/>
- [9] X settings standard: <http://www.freedesktop.org/standards/xsettings.html>
- [10] D-Bus Interprocess communication: <http://www.freedesktop.org/software/dbus/>

THE AUTHOR

Nils Faerber has been an avid PDA user for many years now, and even more so since Linux entered the PDA arena.

You can contact Nils at nils@kernelconcepts.de.

