Mobile Broadband Configuration Assistant

—

Application for Kesäkoodi 2008

Antti Kaijanmäki
<antti@kaijanmaki.net>

February 20, 2008

Contents

Abbreviations

GNOME GNU Object Model Environment – easy to understand desktop for your GNU/Linux or UNIX computer.
[*http://www.gnome.org*]

GNU GNU's Not Unix – The GNU Project was launched in 1984 to develop a complete Unix-like operating system which is free software. GNU's kernel wasn't finished, so GNU is used with the kernel Linux. The combination of GNU and Linux is the GNU/Linux operating system, now used by millions.
[*http://www.gnu.org*]

PPP Point-to-Point Protocol. Also, Paul's PPP Package, an open source package which implements the Point-to-Point Protocol on Linux and Solaris systems.
[*http://ppp.samba.org*]

USB Universal Serial Bus – a serial bus standard to interface devices.
[*http://www.usb.org*]

XML Extensible Markup Language — simple, very flexible text format derived from SGML (ISO 8879).
[*http://www.w3.org/XML*]

1 General

1.1 Motivation

Having an Internet access has become almost a basic commodity these days. In the past few years mobile service providers have started to offer mobile broadband solutions for their customers. Mobile broadband offers a fast and easy way to connect to the Internet anytime anywhere. Through tie-in sales customers also have inexpensive mobile broadband capable phones at their disposal.

Although service providers target their marketing mainly on laptop users, it is equally possible for desktop users to benefit from mobile broadband connection. Combined with fixed monthly fees, instead of charging per megabytes transferred, and having a wide network coverage mobile broadband can be a viable option compared to common land line broadband solutions – especially in locations lacking a land line.

1.2 Problem

Although it is very easy, fast and inexpensive to get a mobile broadband subscription, it is very difficult to configure mobile broadband connection for GNU/Linux based systems. There are no easy to use tools for creating the needed connection settings and thus the user must create nontrivial configuration files[1] by hand! What makes the situation even worse is that service providers often have support documentation only for Microsoft Windows operating systems and thus it can be very difficult even for an experienced user - let alone for a novice - to find the needed information.

1.3 Solution

An assistant software must be developed to trivialize configuration of mobile broadband connections. This tool should work out of the box with minimal or without any knowledge of technical details required from the user. Mandatory configuration files would be created based on simple information provided by the user, such as country and name of the service provider – information that everyone owning a mobile phone should be able to provide.

[1] see appendix E for examples.

2 Project Plan

This section contains the project plan for Mobile Broadband Configuration Assistant – a project for Kesäkoodi 2008. Development will be executed during the period of 3 months starting on May 26^{th} and ending on August 22^{nd}. See Appendix A for a detailed development schedule.

2.1 Goals for this Project

Project aims to address current problematics in configuring mobile broadband connections under GNU/Linux by developing an assistant software for GNOME desktop environment. Because of the tight schedule of Kesäkoodi and lack of certain hardware some requirements and features must be left out of the scope of this project. However it will be taken into account that these missing features can be added some time later.

Development will be done on Ubuntu 8.04 but there is no intention to limit the software to Ubuntu only. During development only commonly used external components and libraries will be used and thus the program should run on any GNU/Linux distribution, given that all the required external dependencies have been satisfied.

Collaboration with other components of GNOME is also a goal. GNOME's NetworkManager is introducing support for PPP connections and utilization of mobile broadband USB modems. Thus Mobile Broadband Configuration Assistant combined with NetworkManager will provide the user a complete package for a successful mobile broadband experience.

In the future GNOME's mobile phone integration can be further extended. Mobile Broadband Configuration Assistant would be only one tool amongst many. A set of tools could be developed to fully utilize the available functions from mobile phones. But developing this *Gnome Phone Suite* is out of the scope of this project.

2.1.1 Requirements

Program must to be easy to use. User must not be bothered by unnecessary technical questions. Program must be translatable to different languages. Program must support multiple configurations and multiple devices.[2] Program must look and feel like native GNOME software.

[2] e.g. John Doe has two phones - one for work, other for private use - and he wants to use them both with his laptop.

Program must support following communication methods between mobile phone and users computer:

- Bluetooth
- USB
- Serial (DE-9)

Some phones also support infrared, but because no infrared adapter is available support for infrared is not in the scope of this project.

Following information given by user must be enough to create configuration:

- communication method
- which device to communicate with
- name of the service provider

2.2 Deliverables

This section describes the deliverables that will be produced during the project. Deliverables address the requirements. Appendix B also contains a simple list as a summary of the items.

2.2.1 Assistant Program

The assistant[3] will be created to divide the configuration process into simple steps and to hide the unnecessary complex technical details from the user. In every step user has a change to go backwards to change previous selections. Assistant will guide user through every step an finally create configuration files to appropriate locations. Editing and removing configurations is left out of the scope of this project.

First step is to select a communication method between the computer and mobile phone. When user has selected the method assistant will move to next step which is selecting the device or location what to use for connection.

If user has selected Bluetooth as communication method, a list of names of available Bluetooth devices is presented. User selects the device that will be used to connect to the Internet. User has possibility to refresh the list. User also has a possibility to directly specify a Bluetooth Hardware Address

[3]on Microsoft platforms called "wizard"

for devices that do not inform about their presence. Program internally identifies Bluetooth devices by hardware address.

If user has selected serial cable as communication method, a list of available serial ports is presented. User selects the serial port where the mobile phone is plugged in. There is also a possibility to enter a custom path for the serial port. Program internally identifies the device by serial port and thus the device has to be plugged to the same port every time.

If user has selected USB cable as communication method, a list of names of available USB devices is shown. User chooses the device that will be used to connect to the Internet. Only devices that have a standard USB serial interfaces will be shown. Any proprietary USB modem will not be supported because of the lack of hardware, interface documentation and development time. Program internally identifies USB devices by HAL *Unique Device Identifier*.

After selecting the device next step is to specify the service provider. User must first select a country because service providers have different settings for different countries. The Service provider is selected from a drop down list. Service provider information is read from *Service Provider Database*. There will be a possibility in the future to enter a custom settings if needed service provider information is not available, but this functionality is left out of the scope of this project.

The final step gives user a summary of selected options and user gives a unique name for the configuration to be created. The name helps the user to identify the specific configuration when multiple configurations are available.

The program will be implemented as *GtkAssistant* and user interface will be designed with *Glade UI Designer* to have a consistent look and feel with other GNOME software. D-Bus will be used to communicate with different services when possible. Libxml will be used to process the Service Provider Database. Bluetooth device support will be implemented using Bluez libraries. Serial and USB device support will be implemented using HAL.

Some additional helper utilities must also be created in order to properly support Bluetooth devices and multiple USB devices. Bluetooth device has to be bind in order to connect to the Internet. This binding has to be requested by some helper utility, just before opening the connection, and the binding is done using Bluetooth Hardware Address. USB devices on the other hand are automatically ready for use, but because they can be plugged in to the system in any particular order, a correct device has to be discovered using HAL UID just before opening the connection. This discovery has to be done using also some helper utility.

2.2.2 Service Provider Database

A database containing service provider specific information will be created. Database will be stored on a single file containing XML presentation[4] of the data for easy editing and human readability.

Document Type Definition XML schema format specification will be provided to easily validate the correctness of the database file.

2.2.3 Documentation

User documentation covering compilation, installation and basic usage will be provided. Documentation will be written in English using plain text.

2.2.4 Translations

Program will be developed in English and a translation for Finnish will be provided as a proof that the project fulfills translation requirement.

2.2.5 Packaging

An installation package for Ubuntu 8.04 will be created containing the program, translations and the database. Package can be installed using Ubuntus package manager.

2.3 Source Code

Main programming language used to develop the program will be C. Other languages may be used for different kind of helper programs if necessary.

2.3.1 Licensing

All code will be licensed under the GNU General Public Licence[5] version 3 whenever possible. Some small portions of code, helper programs and other content may be licensed as Public Domain or under some OSI compliant license.

2.4 Mobile Broadband USB Modems

Support for dedicated mobile broadband USB modems is left out of the scope of this project because lack of hardware.

[4] see Appendix D for an example.

[5] http://www.fsf.org/licensing/licenses/gpl.html

3 Applicant

3.1 Brief Introduction

My name is Antti-Hermanni Kaijanmäki and I am a student at Tampere University of Technology. I have worked mainly with C and C++ and I have familiarized myself with Linux-kernel development, Gtk, php, python, Qt and many other technologies. Unfortunately I do not have any greatly visible contributions to Free Software but here are some projects that I have worked on that can be found using Google:

FinFlect: http://finflect.sourceforge.net
Nokia 770: http://www.mail-archive.com/maemo-developers@maemo.org/msg00762.html
Ubuntu Translations: https://translations.launchpad.net/~kaijanmaki

A Curriculum Vitae should be found attached alongside this document. If not, or for any other inquiry, feel free to contact <antti@kaijanmaki.net>. My CV also contains some information about proprietary projects that I have worked on before.

3.2 Mobile Broadband Configuration Assistant

I have planned this Mobile Broadband Configuration Assistant for over two years now. Unfortunately things like military service and becoming a father got into the way of making it a reality. I even had a working prototype six months ago, but an unfortunate typing error destroyed all what I had accomplished so far. Hopefully now I have a change to finally create a new and better version of the software that most probably will benefit many people.

A Development Schedule

This appendix contains development schedule for the project. Project will be executed during the period of 3 months starting on May 26^{th} and ending on August 22^{nd}. For project plan, see section 2.

wk22	project kickoff set up development environment register project at SourceForge.net
wk23	design UI
wk24	specify XML schema format for service provider information develop parser for service provider information
wk25	implement UI navigation implement general UI functions integrate XML parser to UI
wk26	implement serial communication method implement configuration file generator
wk27	implement USB communication method
wk28	implement Bluetooth communication method
wk29	testing time for unfinished tasks
wk30	localization
wk31	usability testing
wk32	reacting on usability testing feedback
wk33	writing documentation
wk34	project ending

B Deliverables

This appendix contains a list of different deliverables. See section 2.2 for detailed description of each item.

1. Assistant
   (a) source code
   (b) translations
2. Service Provider Database
   (a) XML schema format specification
   (b) database
3. Documentation
   (a) compilation
   (b) installation
   (c) basic usage
4. Installation package

C Resources

This appendix contains lists of resources that are available for development and testing.

C.1 Software

Operating System	Ubuntu Hardy 8.04
Development Tools	Anjuta IDE Glade UI Designer GNU gettext libxml, HAL, BlueZ..

C.2 Hardware

C.2.1 Mobile Phones

Model	BT	IR	USB	Serial
Ericsson R600s	-	-	-	x
Nokia 6021	x	x	x	-
Nokia 6151	x	x	x	-
Nokia 6630	x	-	x	-
Nokia 6280	x	x	x	-
Nokia 9300i	x	x	x	-
Nokia N70	x	-	x	-

BT = Bluetooth, IR = infrared

C.2.2 Cables & Adapters

Type	Availability
Bluetooth Adapter	available
Infrared Adapter	not available
Serial Cable	available
USB Cable	available

C.3 Other

Service Subscriptions	Saunalahti *Saunapaketti3G* Saunalahti *Paketti Plus* Elisa *Puhepaketti 250*
Voluntary Helping Hands	1 tester 1 proof reader (non-tech) 2 usability testers

D Service Provider XML

This appendix contains an example how information about service provider specific settings could be presented in XML.

```
<serviceproviders>
    <country code="fin">
        <provider>
            <name>Elisa</name>
            <apn>internet</apn>
        </provider>

        <provider>
            <name>Saunalahti</name>
            <apn>internet.saunalahti</apn>
        </provider>

        <provider>
            <name>Sonera</name>
            <apn>internet</apn>
            <dns>192.89.123.230</dns>
            <dns>192.89.123.231</dns>
        </provider>
    </country>

    <country code="gbr">
        <provider>
            <name>Orange</name>
            <apn>orangeinternet</apn>
            <dns>158.43.192.1</dns>
            <dns>158.43.128.1</dns>
        </provider>

        <provider>
            <name>Orange (PAYG)</name>
            <apn>payginternet</apn>
            <dns>158.43.192.1</dns>
            <dns>158.43.128.1</dns>
        </provider>
    </country>
</serviceproviders>
```

E Configuration Files

This appendix contains examples of some of the mandatory configuration files that has to be provided in order to establish mobile broadband connection.

E.1 PPP

Connection options.

```
115200
/dev/ttyACM0
connect 'chat -v -e -f /etc/chatscripts/saunalahti'

hide-password
nodetach
persist
holdoff 3
crtscts
modem
maxfail 0
noauth

defaultroute
noipdefault
usepeerdns
```

E.2 CHAT

Automated conversational script with a modem.

```
ABORT BUSY
ABORT 'NO CARRIER'
ABORT VOICE
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
"" AT+CGDCONT=1,"IP","internet.saunalahti"
OK-AT-OK ATDT*99***1#
CONNECT ""
```