smaller companies).

For all these reasons, the main focus of this report will be on the raw data that constitute “public” PSI as defined above. This is the reason why in this report the terms “raw data” and “PSI” are practically interchangeable. We will also focus on the local dimension of Open PSI, that is raw data directly produced by, or directly relevant for, local communities (City and Regions), and on their direct impact on local government and local economy.

Chapters 2 and 3 summarize the importance of data in the modern society and some recent developments on the Open Data front in Europe. Chapter 4 explains why raw PSI should be open, while Chapter 5 shows the potential of such data with a few real world examples from several (mostly EU) countries. Chapter 6 looks at some dangers that should not be ignored when promoting Open Data and Chapter 7 proposes some general practices to follow for getting the most out of them.

Why and how FOSS scripting languages are important at school and at home

Marco Fioretti

M. Fioretti, freelance writer/teacher/researcher (http://mfioretti.com) Email mfioretti@nexaima.net

FOSS scripting languages like Unix shells, Perl and Python are extremely productive, flexible and powerful programming tools for everybody. They are much easier to learn and deploy on any operating system than programs in compiled languages like C or C++, and yet they can automate or greatly speed up many tedious computer activities in any area of daily computing.

The topic of this paper is at the intersection of two tracks of the FOSS Lviv 2011 Conference: FOSS in Education and FOSS at home.

The central point of the paper is that scripting languages like Perl or the shell are not relevant only for system administrators and other software professionals and much easier than they look. These languages have no license costs and can be very efficient and useful tools even at school and at home, in at least two ways: on one level, they can save lots of time when studying or in normal home computing tasks, even on older/limited computers.

On another level, they are a very efficient, easy to set up and extremely portable way to teach programming, from primary school to adult, professional training classes. Besides, unlike other languages used to teach programming, these are tools directly usable and valuable when looking for a job (not just in the software industry, but in any business sector).

This paper discusses in depth all these theses and the fact that scripting languages are much easier to learn than normally thought. The paper also provides practical, immediately useful, real world examples of how non programmers (especially, but not exclusively teachers, students and small business owners!) can quickly learn these tools and can benefit from them for their daily, non programming computer-related needs.

Promoting scripting languages in this way can also be an indirect, but very effective way to promote larger adoption of FOSS in all branches of society. The paper also discusses this point and suggest some strategies for effective promotion and teaching of these tools in schools and small businesses. In spite of its topic or, we should say, just because of it, the paper is non-technical and its intended audience is the general public (but especially teachers).

References:

OpendDocument Format scripting examples, M. Fioretti, 2010: http://freesoftware.zona-m.net/tag/odf-scripting/

How to survive the GNU AUTOTOOLS

Rob Savoye (RS)

Open Media Now <rob@openmedianow.org>

Surviving the GNU Autotools is a technical talk for anyone that has to use the Autotools, namely Autoconf, Automake, and Libtool.

After a brief discussion of the history behind Autoconf and configuration tools in general, the basics of using the Autotools are discussed. This is oriented towards anyone who has to configure and build a free software package using the Autotools. Then I dive into the gory details of how to debug configure scripts and Makefiles. Some coverage of basic Unix shell utilities and bourne shell programming will also be covered.

Otwarta infrastruktura przetwarzania rozproszonego berkeley – wspomaganie międzynarodowych obliczeń naukowych

Swierczewski L.

College of Computer Science and Business Administration in Łomża,

University of Warsaw, lswierczewski@pwsip.edu.pl

Wraz z ciągłym rozwojem nauki i technologii informatycznych środowiska naukowe coraz bardziej odczuwają potrzebę wykorzystywania olbrzymich zasobów obliczeniowych. Nie wszyscy jednak mogą uzyskać dostęp do wysokowydajnych komputerów dużej mocy, których zakup i eksploatacja jest bardzo droga. Alternatywnym rozwiązaniem może być platforma BOINC, która umożliwia przetwarzanie rozproszone danych na komputerach połączonych siecią Internet i udostępnionych przez wolontariuszy.