# An Internet System for Information about Seminars

Ireneusz SZCZEŚNIAK
*Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences*
*iszczesniak@iitis.gliwice.pl*

### Abstract

*We present a platform independent system for automatic generation of a static website which provides information about a seminar schedule. This article describes a general flexible framework that can be applied to other websites requiring moderately frequent updates. Brief instructions on how to use the system are given as well.*

## 1. Introduction

Many organizations have a website with a schedule of seminars they organize. Such websites are usually updated every week (to add a seminar announcement) and at most websites these updates are performed manually which is the bane of their maintainers. Some websites are maintained automatically, for instance [1, 2]. To the best of the author's knowledge, there is no software package available for maintaining seminar websites, except the software used to generate [2].

Therefore we developed a software package to facilitate the maintenance of seminar websites. The method used by the software is general and can be harnessed to generate other websites that require frequent updates. Because the presented approach is novel and rather unconventional, so far it has not been seen used elsewhere.

Large websites are dynamically generated (i.e. the hosting server dynamically generates pages at the time of a user request) with PHP, Python or Perl scripts. Such websites, however, are susceptible to hacking, because of security caveats in the scripts. We designed the package to minimize security threats.

The presented package is based on portable tools (e.g. Python) and, as a result, it is also portable. The package is ready for use in UNIX-like environments (UNIX flavors, Linux, Mac OS X, because it heavily uses UNIX-like tools (e.g. the Make tool). Nonetheless, the package can be used under different versions of Microsoft Windows with the Cygwin environment available.

This article is organized as follows. Section 2 discusses the approach used in the software. Section 3 offers brief instructions on how to use the package, while Section 4 concludes the paper.

## 2. Design

The system uses the following well-documented, well-tested and free tools: Automake, Autoconf, Make, Python, xsltproc, pdflatex and CVS, which are all GNU projects [3]. Data processing in the package is organized as shown in Figure 1. The main three objectives in designing the package are: simplicity, reliability and security.

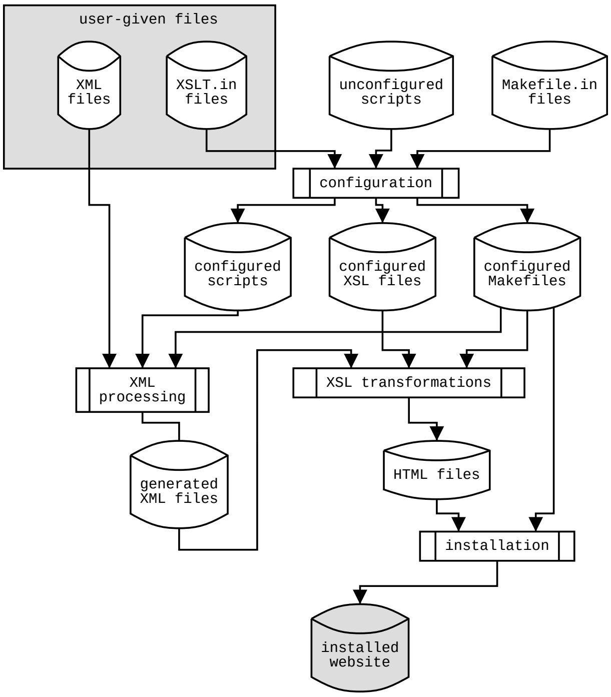

Fig. 1. Scheme of data processing in the package.

## 2.1. Simplicity

The package is entirely based on text files in order to make it simple. Configuration files, scripts, data files (XML files) of the project are regular text files. This simplicity fosters portability and reliability, because a website maintainer can fix and extend the package. No compiled programs or binary files of obscure structure are included in the package.

Separate pages are generated with the `xsltproc` command [4]. The user provides data about seminars in XML files and data about page layout in XSL files. Therefore the layout is separated from the content of the website.

Files named Makefile (configuration files of the `make` command) are used to generate website files in the right sequence. The generation of Makefiles is relegated to the Automake tool, which builds them based on its configuration files named Makefile.am. Automake configuration files are simpler in comparison to standard Makefiles. Consider for instance the Makefile.am file shown in Figure 2. The Makefile file generated based on this example contains hundreds of lines.

```
SUBDIRS = data posters info archives images download
datadir = $(prefix)
data_DATA = index.html main.css
index.html: index.xsl index.xml ./data/.p/future.xml
        $(XSLTPROC) --xinclude index.xsl index.xml > index.html
```

Fig. 2. A sample Automake configuration file (Makefile.am) of the package.

### 2.2. Reliability

Aside from the fact that a static website always works (is reliable), we want to restore the website if this action is necessary, for instance after it was accidentally removed from disk. For this reason the package is stored at a CVS (Concurrent Versions System) server, from where it can be checked out and used to restore the website [5]. Storing the package with CVS is efficient because CVS was designed to handle text files well. Moreover, the CVS server stores the packet's history of changes and therefore building the past versions of the website is feasible.

CVS is not indispensable for the project for which it is only a mere convenience. In our setting regular backups can successfully replace it, because the package is developed by one developer only. Nevertheless, CVS is employed, because it is reliable, simple to use, and is frugal with using disk space.

### 2.3. Security

The most secure website is static and our approach aims at this ultimate security. The package builds a complete static website which is then copied to a given location in a file system. Therefore, an HTTP server does not run scripts for dynamic website generation and does not run the risk of being compromised by exploiting security caveats in these scripts. Interpreted languages serving this purpose (e.g. PHP, Python, Perl) are especially prone to security exploits. The package has several Python scripts, but these are run manually by a website maintainer who has no reason to exploit their weaknesses.

## 3. Brief Instructions

Using the package is standard, because it incorporates the Autoconf tool. First, run the configuration script named "`configure`" to configure the package, then build it by executing the "`make`" command, and finally install it by executing the "`make install`" command.

The configuration script searches for the location of the "`python`" and "`xsltproc`" commands, and configures the package. With the "`--prefix`" argument you can provide the script with the directory name where to install the website, otherwise the default "`/seminar`" is assumed. The building step generates all necessary files of the webpage, such as poster files in the PDF format about presentations or the HTML document listing past seminars. The installation step copies the generated files into the destination directory.

Information about seminars is stored in the "`data`" subdirectory. Each presentation is described in a separate XML file. A sample file is shown in Figure 3. The Python language [6] is used for parsing these XML files and generating XML files which list future and past seminars.

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<presentation>
<datafile>jsmith_pres.ppt</datafile>
<date><year>2003</year><month>3</month><day>14</day></date>
<time><hour>10</hour><minute>00</minute></time>
<author>Prof<dot/>Jan Wiśniewski</author>
<title>This is the title of the talk</title>
<abstract>This is the abstract of the talk.</abstract>
```

Fig. 3. A sample XML file with a presentation details.

## 4. Conclusion

The paper presents a software package for the automatic maintenance of a static website for information about seminars. The approach employed is novel and unconventional: a website is built and installed like a regular software package. The package does not build and install programs, but files that compose a website instead. The approach is reliable, secure and simple.

It must be conceded that using the package may be burdensome when the website requires very frequent updates since every update entails building and installing the website again. However, when modifications are moderately frequent (and this is a reasonable assumption for a seminar website), this administrative overhead can be accepted in return for the security of an HTTP server providing a static website.

The seminar website of The Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences [1] is produced by the package. Please visit this website, should you be interested in seeing the package's results.

## Acknowledgment

The author wishes to thank his wife Arlena for assistance with English.

## REFERENCES

1. http://www.iitis.gliwice.pl/seminaria
2. http://www.ma.utexas.edu/cgi-pub/seminar/calendar
3. http://www.gnu.org
4. http://www.xmlsoft.org
5. http://www.cvshome.org
6. http://www.python.org