Automatic report generation with Web, TeX and SQL

Boris Veytsman
ITT, Advanced Engineering & Sciences
12975 Worldgate Dr, Herndon, VA 20170
`boris dot veytsman (at) itt dot com`

Maria Shmilevich
ITT, Advanced Engineering & Sciences
12975 Worldgate Dr, Herndon, VA 20170, USA
`maria dot shmilevich (at) itt dot com`

Abstract

One of the most time-consuming tasks of a manager for a federal contractor is the creation of reports: weekly, monthly, quarterly and yearly as well as special reports at the end of a project or on any given date. Such reports are usually made by copying and pasting the daily reports of subordinates.

The system described here makes these reports automatically. The members of project team file their daily work results using a Web interface. These entries are kept in a SQL database. The report generation utility is launched through a Web interface. It creates a LaTeX file by selecting the data relevant to the given set of contracts and tasks, employees, time periods, etc., and collating the individual reports. The result is then run through pdftex or latex2html or latex2rtf to create either a PDF report or an editable (e.g., in Microsoft Word) file.

1 Introduction

In the last several decades applied science and technology in the US have seen unprecedented breakthroughs. Internet, GPS, space missions, a complete change in the civil aviation field and many advances in the military area are just a few examples of rapid technological progress. Of course there are many reasons for this, but it seems that one reason is the unique and fortunate method of technological cooperation between the government, universities and private contractors. In this scheme, the government agencies set the technological goals and solicit bids to achieve them. The winners of the bids get contracts for development of high-end technology with important military and civil uses.

Government agencies in this scheme are gate keepers of the people's money. They are obligated to control spending and check that the contracted research and development work is proceeding properly and the milestones are to be met on time. Therefore most agencies request detailed reports of the contractor's activity at regular intervals. These reports, however, pose the following problem. Obviously the taxpayer is interested only in the *results* of the contracted research and development. The time and money spent on the intermediate reports does not contribute to the value and should be minimized. This is true both for the agency, which spends effort on the analysis of the reports, and the contractor, which spends effort on their preparation, and eventually passes the costs to the customer, thus increasing the total cost of the bid.

A report of high quality (including typographic quality!) is easier to analyze, so the report must be good. On the other hand, a good report might take a considerable effort to prepare. The goal is to make good reports with minimal effort and costs.

The traditional way of making intermediate reports is the following. The contractor's employees send e-mails to their managers describing their accomplishments. A manager copies and pastes these data into a Microsoft Word file and sends the file to the next level manager, who collates the received reports together. The task is repeated regularly, and each piece of information is copied and pasted several times: in weekly, monthly, quarterly and yearly reports. If the contract involves many tasks and subtasks, the work is overwhelming. This is unproductive work, since the real task of managers is management, not copying and pasting repetitive chunks of text.

Since most of this work is purely routine, it is

possible to teach a computer to do it, thus freeing engineers and managers for more creative tasks. This is the main idea of the system which was created at ITT in 2000–2001 and successfully used ever since.

2 Analysis

The first thing in the creation of a system for automatic report generation is to understand the structure of reports. A report is separated into *contracts.* The contracts are separated into *subcontracts,* and these are in turn separated in *tasks* and *subtasks.* Each individual report covers a subset of a hierarchy: it can include several contracts or just one contract, or several tasks from a subcontract, etc. It also covers a certain time period: week, month, quarter, year, etc.

The actual contents of the report are collated from the individual work by the engineers. Each of the engineers describes her or his work made during a particular week under each subtask, task, subcontract and contract. Sometimes a report includes the names of the engineers, and sometimes not, depending on the style chosen.

This structure is well suited for a SQL database. Each individual entry can be a record in the database, indexed by the subtask or task it belongs to, the engineer who made the entry, the time covered and the time it was made. The hierarchy "Contract-Subcontract-Task-Subtask" can easily fit into a SQL table with the usual parent-child relations. SQL operators can be used to extract from the tables the information that relates to the given task and time period.

We wanted the report to be available in several forms: a high quality PDF file as well as editable RTF and HTML formats. We chose LaTeX as the base format for the report because it can be used to produce beautiful PDF output, and the tools to transform it into HTML and RTF are widely available.

The interface to the software should be available from different computers: engineers' and managers' workstations. This makes an internal Web server a natural choice.

3 User interface

3.1 Authentication

A user (engineer or manager) logs in to the web server with her or his own user name and password. The database of logins and passwords is integrated with the system, so immediately after the user is authenticated, she or he is assigned a role (access level) in the system. There is a hierarchy of roles:

1. A normal user can input the information about his or her work into the database or correct it.
2. A manager can view the information and create reports, add or delete contracts and tasks.
3. An administrator can add or delete users, reset passwords and change access levels of the users.

Below we discuss these functions in more detail.

3.2 User access

A user should log in at least once a week and choose from the menu tasks and subtasks for which some work was performed by him or her. Then she or he inputs the work done under each category. There is an important option of choosing a special entry "Same as last week"; this will expand the time period of the entry of the previous week in the given category. The user can also set the priority of the tasks completed. The tasks with high priority are highlighted in the report.

3.3 Manager access

A manager can perform the functions of the user plus additional functions related to report creation and contracts and tasks changing.

A manager chooses from the menu the contracts, subcontracts, tasks and subtasks to cover, performance time, and report options: whether it should be in PDF, RTF or HTML format, whether it should include engineers' names, etc. The report is created and a link for download is presented to the manager.

A manager can add or delete contracts, change subcontracts, tasks and subtasks. This will update the menus presented to all users.

3.4 Administrator access

An administrator can change the information about users. She or he is presented with a menu, which includes changing of user personal information (name, e-mail), resetting passwords, changing access levels, etc.

3.5 Additional bells and whistles

The system generates reminders for the users to log in and enter their information, and sends lists of the procrastinators to the managers. It also generates periodic backup dumps of its databases.

4 Implementation notes

The system is implemented on a Linux computer using the Apache Web server, MySQL database, sendmail, teTeX suite and latex2html and latex2rtf programs. It is essentially a zero administration server: since it was set up, only security patches have been
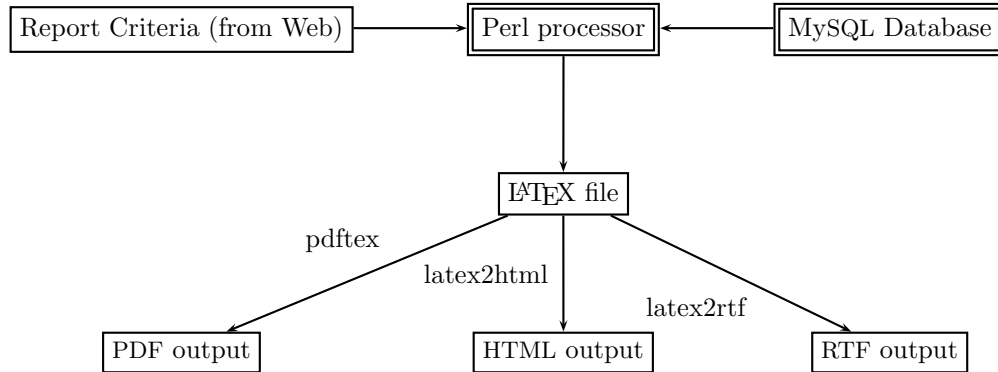
Figure 1: Report creation

applied to the machine, and everything else "just works".

The flowchart for the report creation is shown in Figure 1. The Perl program extracts from the database the entries satisfying the selected criteria. They are collated into a LaTeX file. The hierarchy "Contract-Subcontract-Task-Subtask" is mapped to the hierarchy "Chapter-Section-Subsection-Subsubsection" of the `report` document class. The entries themselves are organized into itemized lists.

An example of the result is shown on Figure 2. In this example the engineer (Archimedes) started the work of moving Earth using a lever. His completed tasks include a high-priority development of the background and low-priority work on geometry.

5 Conclusions

We developed a system to perform an important and time-consuming task of generating periodic reports by a federal contractor. The system is based on open and free software. It provides a very efficient and cost-effective solution, which has been successfully working for half a decade.

Contract 1

Moving Earth With Lever (ERTHMV)

1.1 Finding a place to stand

1.2 Mathematical Background

1.2.1 Development of series summation

- *12/23/0282:* **Developed background** *Archimedes*
- *12/30/0282:* Worked on geometry *Archimedes*

1.3 Create a lever

2

Figure 2: Example of a page from a report