

# Conference Time-Table Management

Erik Wilde, Martin Waldbuger, and Beat Krähenmann  
ETH Zürich (Swiss Federal Institute of Technology)

## Abstract

Conference time-tables provide information that is indispensable for all attendees. Since there are a lot of reusable data structures and tasks, we have designed the *Conference Time-Table Management (CTTM)* system, which is intended to be used as a reusable component in a large diversity of conference Web sites. CTTM features a flexible concept for time-tables and provides users with personalization and notification services.

## 1 Introduction

Conference time-tables are highly structured data, they contain a lot of information and are used by a large number of users. A survey we did on the Web however revealed that many time-tables are implemented rather poorly. In almost all cases they use static content, do not allow any personalization, and are never complemented with additional services such as cross-media publishing and notifications for subscribers.

The goal of the *Conference Time-Table Management (CTTM)* system is to provide a reusable platform for services that will make conference Web sites easier to manage for conference organizers, and more comfortable to use for conference attendees. The primary design goals have been the following:

- *Reusability*: The solution should be easy to integrate into conference Web sites, only few interfaces should be required, and they should be simple and well-documented.
- *Flexibility*: Since different conferences have different styles of organizing and presenting their time-tables, the solution should be flexible enough to accommodate at least the majority of these styles. This also means that from a technical perspective, the solution should integrate easily into the large diversity of existing conference Web site implementations.
- *Simplicity*: Integrating our solution into a conference Web site should be simple, in terms of integration interfaces as well as technical requirements for the platform.
- *Functionality*: The functionality should be extensive enough to persuade conference organizers to use our solution. That is, the benefit of added features and experienced ease of use should outweigh the cost of installation and integration.

As a result of these design goals, it became clear that on the one hand the implementation platform should be kept as simple as possible. On the other hand we needed to choose a platform which made it as easy as possible to implement a solution satisfying the requirements.

## 2 Design

The design of the data model of the time-table is the most important issue of our work. Since this is the input that conference organizers must provide for CTTM, we decided to specify the data model in form of an XML schema. Since none of the available XML schema languages satisfied all our requirements, the schema is specified as a combination of XML Schema and Schematron.

As shown in Figure 1, the CTTM schema has three major parts. The first part defines the event types, the second part is the time-table itself (a set of events), and the third part is information about users, which may have special administrative access rights, or may be regular attendees of the conference. We will examine these three parts more closely in the following sections. It is important to notice that the XML document for a conference changes while CTTM is running, with events being modified and people registering and personalizing their time-table.

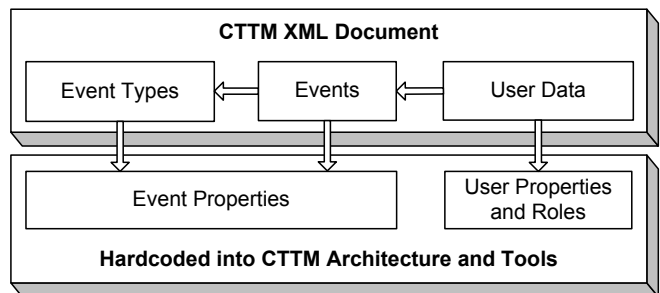


Figure 1: CTTM XML Document

We chose to implement such a file-based solution because we did not want to require every CTTM user to run a database, and because we assume that the number of users will be moderate, so that file-based operations do not become prohibitively expensive.

### 2.1 Events and Time-Tables

Before creating the actual time-table, a conference organizer must define event types, which define the way a particular conference is structured. Event types are defined by a type name and the properties (which are hardcoded into CTTM) that an event may or must have. Event may be allowed to contain events (of specified types), so that it is possible to define a hierarchy of events.

The actual conference time-table is a list of events, which are defined by referring to their type, and then defining the properties that they have. In addition, an event may refer to all events in it. For example, a track may be modelled as an event which has a name (a property) and contains references

to all the events in the track, which again may be hierarchical events (such as sessions), or single events (such as talks). The events are organized as a list of entries, which use standard XML referencing mechanisms.

Each event is described by a number of properties, which differ slightly among the types used for a particular conference. The properties that are available are *title*, *date/time*, *location*, *speaker/author*, and *links to additional materials* (such as handouts).

## 2.2 Users and User Roles

Since one of the main features of CTTM is that it supports personalization, it needs to keep track of users and their personalization settings. Users are registered with their name, a password, and any contact information (such as an email address) that they may wish to supply for receiving updates and notifications. User data is based on the user properties and roles hardcoded into CTTM.

User management enables users to login, access their personalized settings, and logout. Settings affect two main areas. One is the personalized display of the time-table, making it possible for conference attendees to create their own time-table, for example only listing two parallel events at a time, offering them a choice between their favorite and a fall-back. The second area is notification services, which enable attendees to receive notifications of various events using different channels. For example, if they choose, they may receive a GSM *Short Message Service (SMS)* notification 10 minutes ahead of each session start.

For administrative purposes, we differentiate between user roles. We support regular users (i.e., conference attendees), event managers (having management rights for a limited number of events), and administrators (with unrestricted access rights). Event management rights are necessary for any modification to the time-table itself, while only administrators are allowed to change user information (such as deleting users or resetting their password).

## 2.3 User Interaction

While the data model presented in the preceding two sections is the foundation of CTTM, the purpose of the system is to provide an interface that enables conference attendees to access the data in a simple and intuitive manner. Starting from a number of use cases, we designed an interface that reflects the usage patterns, which are standard sign-up/login/logoff procedures, time-table management and personalization, and notification management.

## 2.4 Cross-Media Publishing

One of the design goals is to keep the whole CTTM solution media neutral. We achieve this by designing all interaction to be built on XML, which can then be transformed to the appropriate format. For example, displaying a custom time-table as HTML or printing it as PDF to take it to the conference should not require any modifications to the CTTM core. Naturally, supporting new output media required adaptation, but we tried to minimize that effort as much as possible.

One example are notifications, which can be triggered by event managers and administrators. Instead of integrating an email gateway into CTTM, notifications are delivered as XML. They can easily be transformed to fulfill the requirements of the gateway, which can be an email system, or a gateway for sending SMS messages.

## 3 Implementation

Our implementation platform is Apache's *Cocoon* framework [1], which basically is a servlet-based implementation of a cross-media publishing framework. Cocoon offers a flexible way for defining processing pipelines, which usually contain one or more XSLT stages or other transformations. In addition to Cocoon's XSLT, we also use *Extensible Server Pages (XSP)* technology, which is similar to the more popular JSP and enables us to execute Java code for some stages within the processing pipeline. The following picture shows an example of a typical CTTM/Cocoon processing pipeline: A new user requests the application form, fills it in, and gets a confirmation that his account has been created.

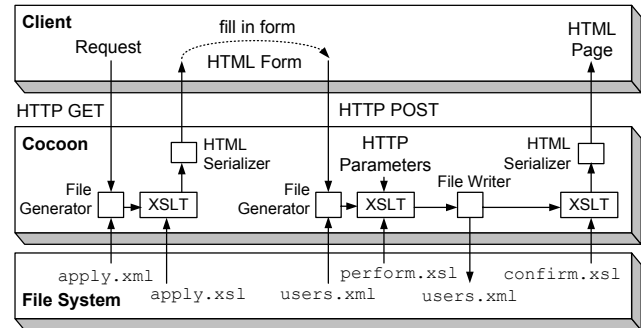


Figure 2: CTTM Processing

Since we are using a file-based approach, it is necessary to keep track of the possible sizes of the files. We estimate the file size to be  $(0.5 \times \#events + 1.6 \times \#users)$  kB, so for a rather large conference with 100 events and 2000 users, the file size would be around 3MB, which is manageable.

For a live demonstration using the time-table of the WWW2003 conference, please visit <http://dret.net/projects/cttm/www2003demo>. We would like to thank the WWW2003 organizers for providing us with the necessary data for creating this demo. On standard PC hardware (and assuming 2000 users and a 3MB XML file), the system is able to handle about 5 hits/sec which require write access (modifications to the XML file, most notably users updating their profile), while read hits do not create any relevant performance problems (Cocoon has very sophisticated caching mechanisms).

## 4 Conclusions

Since conferences usually confront attendees with a large amount of information, we tried to make this more manageable by creating a generic tool-set. Our *Conference Time-Table Management (CTTM)* system provides all the features that we believe will make conference time-tables more easy to manage for attendees as well as for organizers. The platform requirements are rather moderate, but also create some limits to the number of users that can be managed reasonably.

## References

- [1] BILL BROGDEN, CONRAD D'CRUZ, and MARK GAITHER. *Cocoon 2 Programming: Web Publishing with XML and Java*. Sybex, Berkeley, California, October 2002.