

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/277556714>

AEMS: Towards an Advanced Event Management System

Conference Paper · October 2011

DOI: 10.1007/978-3-642-24355-4_9

CITATIONS

0

READS

2,623

2 authors:



Paulo Cristo

1 PUBLICATION 0 CITATIONS

SEE PROFILE



Ricardo Martinho

Center for Health Technology and Services Research (CINTESIS)

99 PUBLICATIONS 532 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



TeenPower: e-Empowering teenagers to prevent obesity. [View project](#)



Analysis of Business Processes [View project](#)

AEMS: Towards an Advanced Event Management System

Paulo Cristo¹, Ricardo Martinho¹

¹School of Technology and Management, Polytechnic Institute of Leiria

Morro do Lena, Alto do Vieiro, 2411-901 Leiria

cristo.paulo@gmail.com, ricardo.martinho@ipleiria.pt

Abstract. Opensource event management systems such as OpenConf, EasyChair and CiviCRM do not foresee the activities, resource allocation and costs management features necessary to each programmed event item. In this paper we present AEMS – an advanced event management system with these features. We present the requirements for such features, based on a case study carried out with a real-world customer. Then, we describe the development of AEMS and discuss its data model, architecture and the developed prototype. With AEMS, our customer will be able to manage all activities, resources and costs associated with an event. The results from applying our development to our real-world customer will allow us to derive AEMS to a generic software product.

Keywords: Event management, project management, registration, activities, participants, tasks and assignments, schedules, calendar, notifications, access levels, reporting, dashboards, teams, open source.

1 Introduction

Event management is the process by which an event is planned, prepared and produced. Like any other form of management, it involves the assessment, definition, acquisition, allocation, direction, control and analysis of time, finances, people, products, services, and other resources, in order to achieve objectives [2]. An event can also be described as an “organized occasion such as a meeting, convention, exhibition, special event, gala dinner, etc. An event is often composed of several different yet related functions” [3].

When we are referring to events, we are including a broad range of genres, like workshops, presentations, conferences, seminars, fairs and festivals, business and corporate events, government and civic events, sports events etc. Current approaches to Event Management Systems (EMS) cover a range of these kinds of events, but lack support for registering associated managing activities. For example, a conference program has sessions (program items), which in turn have a range of managing activities (management items) such as room preparations, scheduling and session chair assignments.

The most common features that we can find in current EMSs are: Attendee's registration; Event scheduling; Contact management; Automatic notifications; Event publishing (event info page) and Invoice and payment functionalities.

In this paper, we propose AEMS: an event management system which, besides considering these common features, adds the following ones (detailed further in section 3): management items (tasks) associated with event program items; participant's management and role assignment to each management/program items, and schedule management; cost management for both management/program items. Our approach includes Project Management Body of Knowledge [1] best practices regarding the scope management (presentation of work breakdown structures), time and cost management (scheduling and cost estimation processes). As a proof of concept, we also present a prototype for AEMS. For this implementation, we used scenarios from a real customer company, which events are mostly related with cultural and educational activities at a museum. Its main event kinds include guided visits, conferences, exhibitions, theater events and workshops.

Although we may have similar events occurring along the year (recurrence), each single event has very specific characteristics, so we felt the need for a generic tool that could, at the same time, not only accommodate the singularity of every event, but also respond efficiently to the identified common requirements.

This paper is organized as follows: the next section refers to most prominent related work on EMS solutions and approaches. Section 3 describes the overall architecture for our AEMS event management prototype, and section 4 presents an application example for this architecture. Section 5 concludes the paper and presents further developments.

2 Related work

During our study, we tried to understand a bit more about the EMSs available on the market. There are many commercial solutions out there, with huge differences in what concerns to the quality of the product itself and the number and relevance of the features included. Unfortunately, in almost every single case it is barely impossible for a common user to get in touch with the product, since there is no trial or demo version freely available. Most of the companies only provide a features list [4-8] that we have to rely on, and the possibility to request a demo meeting via a sales representative. Most of these tools are also extremely expensive for small businesses or individuals.

From what we were able to realize from publicly disposed information, the majority of these solutions are somewhat poor in terms of functionalities that we consider to be key features, especially in what concerns to project management capabilities for events. On the other hand, a few of them have so many "extra"

features that they become a complete mess from the user's point of view, making them inappropriate for non-skilled users [6, 7].

Due to the difficulties on deeply studying any commercial software, we focused more on the open source market, where the offer is extremely shorter and fragmented, despite the existing demand that we can realize just by spending some time browsing the terms on the Internet. The existing projects are particularly oriented to a specific type of event, and they normally target the group formed by conferences and seminars and online ticket selling. From the open source systems we analyzed, we found no EMS solution that could address the management activities of events.

Following is a short presentation of the open source tools that we took in special consideration either by its popularity or functionalities.

OpenConf [9] is a system that is used to manage online conferences, allowing paper and program submission, and reviewing features by other members. It is available in two versions; the community and the professional edition. OpenConf is used specifically for conferences, workshops and seminars. It has no project management features and it is not suitable for other type of events. Its main features are:

- Proposal submission;
- File upload;
- Documents reviewing;
- Email notifications;
- Data export (SQL E CSV) and reporting;
- Build online programs;
- Offers and reviewing proposals.

EasyChair [10] is another free and widely spread conference management system, being used in more than 4000 conferences in 2011, according to information available on the project site. Like OpenConf, it is entirely designed to the same specific types of events and does not offer support for others. Its main capabilities are:

- Automatic paper submission;
- Paper assignment based on the preferences of PC members;

- List of the latest events;
- Submission of reviews;
- Sending email to PC members, referees and authors;
- Monitoring email;
- Online discussion of papers;
- Automatic preparation of conference proceedings.

CiviCRM [11] is a totally different tool. It includes a Customer Relationship Management (CRM) set of modules, specially focused on fulfilling the needs of small, non-governmental, non-profitable and cooperative organizations. It is designed to allow data management about volunteers, activists and downers, as well as the handling of more generic contact types, like employees, customers and sales people. It integrates the CiviEvent module, which has some event management features. CiviCRM can be installed on top of a Content Management System (CMS), like Drupal [12] or Joomla [13]. This is the open source tool that we found more close to any commercial software, in terms of features, which include:

- Custom event types;
- Online registration for paid and free events;
- Contacts management;
- Automatic registration confirmations;
- Import and export participant data from other systems (.CSV files);
- Event data exporting features (iCal files);

Although CiviEvent has some of the capabilities we were looking for on an EMS, it is just a small subset of the CiviCRM application, and not purposely designed for event management. Therefore, it lacks all the project management aspects we are looking for, as well as some basic event related features we need, like the concept of event items (or sub-events), team assignments or any sort of visual agenda scheduling.

Storm [14] was another free tool that got our attention. Unlike the others mentioned in this paper, Storm is not an EMS at all, but a project management

solution, with some interesting features that we would like to see on our AEMS system, especially in what concerns to organizing human resources, distributing them along the tasks/projects, and also because it has the possibility to create work tickets and manage execution times. It is also possible to create a few reports and do expenses control (cost management). Storm has a big part of the management features that we want to build for events, and because of that it was very helpful in clarifying our concept. Storm also integrates with the Drupal platform.

At first, we thought that these last two projects could be a perfect match if they could be integrated. Therefore, our first efforts were driven to this integration, customization and the extension of CiviCRM and Storm in one single application. However, we soon realized that a process involving three different data models (Drupal, CiviCRM and Drupal) designed with such different purposes, even if sharing the same core platform, would result in a bigger development effort than developing our EMS from scratch.

3 AEMS requirements, domain model and architecture

In this section we will present the requirements for our own EMS, as well as present a brief overview over the data model and the architecture of the system.

The main requirements for our system (customer's and derived ones) are:

- Event and participant types. (we introduced the concept of “active” and “passive” participants)
- Participant roles (attendee, presenter, host...)
- Event items (management items and program items)
- Contacts management
- Costs and budget control
- Team building and personal time tables (worker time sheets)
- Automatic program building and file upload features
- Visual calendar scheduling
- Internationalization support (English and Portuguese)
- Participant's profile (with events list and activities)

- Dashboard and custom reports (for instance, monthly activities, type and number of attendees, budget/expenses tracking) with drill down features)
- Automatic email notifications
- Payment facilities
- Invoice and confirmation receipts
- Mailing lists and email marketing campaigns
- Tickets and badges printing
- Wait lists
- Registration management (participants, limits, groups)
- Application access levels
- Data import, export, and synchronization features (for contacts and events) with desktop and mobile client
- Publishing of an event profile (event website), with view and registration functionalities
- Social media networking
- Service layer for mobile applications

In the following figure we have a very brief overview of our core domain model. Events have participants (that can be of two types; active or passive, and have different roles assigned). Each event can have several event items (program or management). Every participant should be on our contact list. If the participant is internal to the company he belongs to a team and is associated with one department.

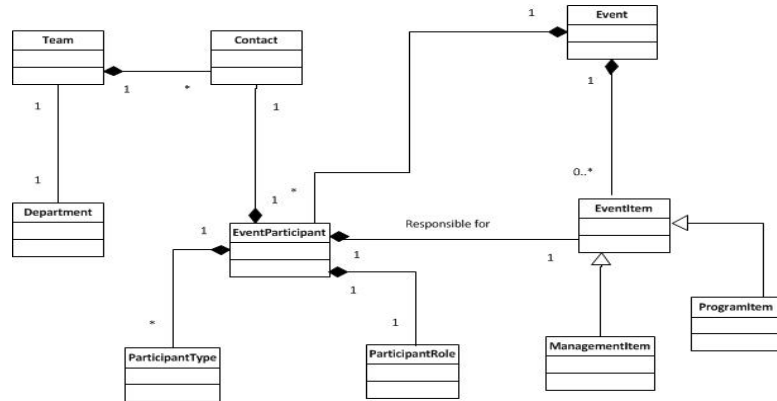


Fig. 1 – Brief overview of the AEMS domain model

The domain model of Figure 1 is implemented by following the architecture illustrated in Figure 2.

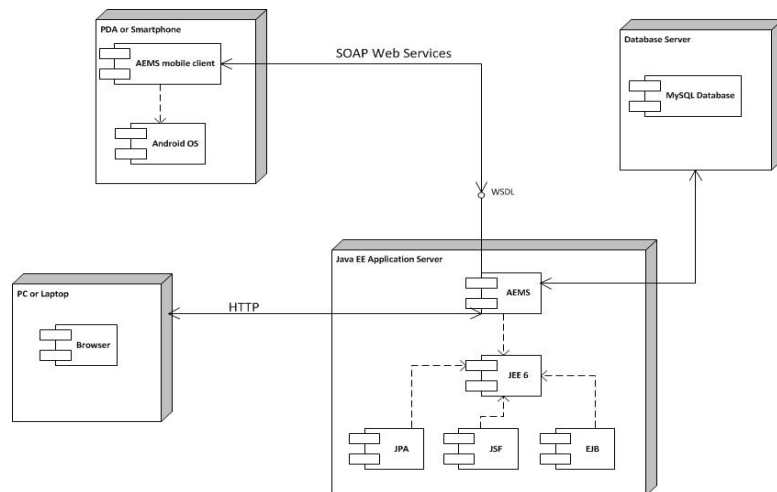


Fig. 2 – AEMS architecture

It basically depicts a three layer architectural pattern implementation [15]. The core platform is Java EE 6 [17] and the application is built mainly using EJB's (Enterprise Java Beans), JPA (Java Persistence API) and JSF (Java Server Faces – Model-View-Controller MVC-based [16]) technologies deployed on a JEE application server. We use MySQL [18] server as backend database and we provide a web services layer (SOAP) [19] for interfacing with mobile clients.

4 Application example

Following we show some screenshots of the application on earlier stages of the development process. Figure 3 shows the main screen of the application where we have an agenda to easily view/edit any current event or act as a shortcut to create new ones. The main sections navigation menu is on top of the page and on the left corner we have relevant information about current or upcoming events, and we have a list of common tasks, actions or “todo’s” waiting for input. The main screen “widgets” may also vary according to login roles.

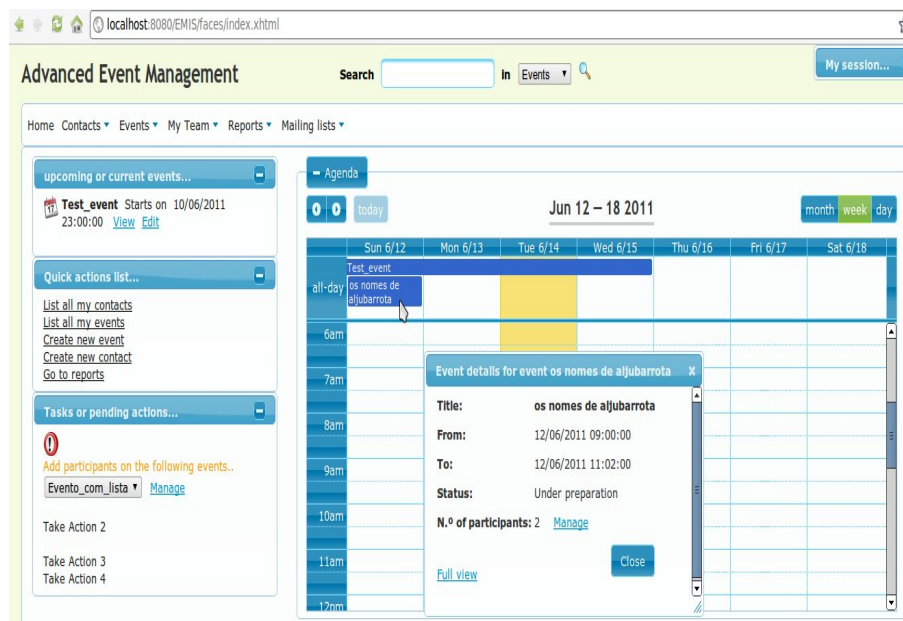


Fig. 3 – Main page of AEMS

Figure 4 shows the employees work scales design page. In our customer’s case employees do not work the normal “Monday to Friday” week so we need to know who is available to work on a given day or week and who is not, in order to assign them tasks/events. In this page we can easily choose a person from a team/department (the person is identified by name and with a different color in calendar) and then pick any date on the agenda to automatically inform the system that the person in question is working on that time period. At the bottom of the page the full year view calendar is also updated with the same information. We have other views where we can see the employee entire profile, regarding the events where he was involved (participant and events will be identified by the same color).

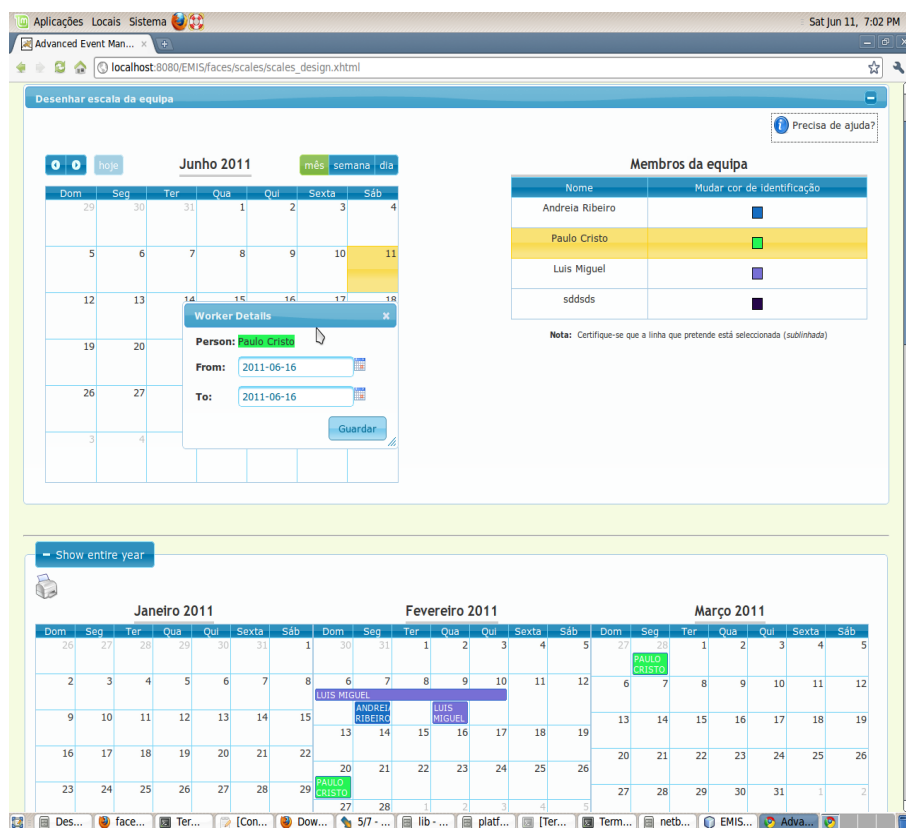


Fig. 4 – Work scales design page

The next figure shows the event items edition page. We can edit an event and create self-contained sub-events (event items). These event items can be of two types: program items and management items. Program items are part of the event program itself and will be visible to everyone (except if the event is not public), while management items are internal management activities that are not visible to the public in general. Each event item can also contain other event items. Event items are organized in hierarquical order inside an event. Every vent item will have assigned a responsible person (an active participant). At the end if have a complete tree of all the tasks and activities that are part of the event and we can automatically build an event program from the program items information.

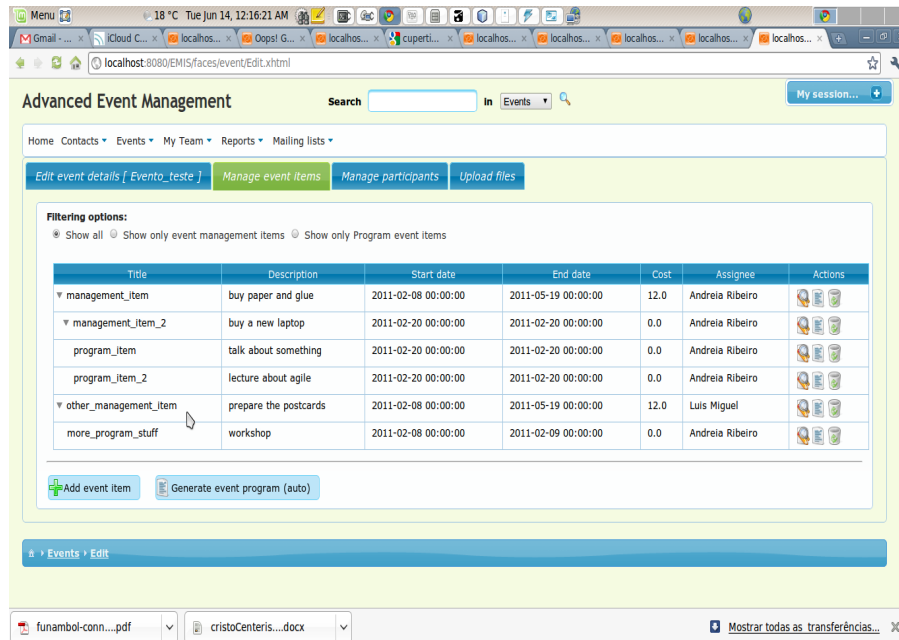


Fig. 5 – Edit event / add event items

5 Conclusions and Future Work

We have developed a tool that allows end users to easily integrate and attach project management tasks and activities to events life cycle, from configuration to planning and management processes. We have presented the architecture, the domain model, and a functional prototype based on a real customer's request. By performing several iterations and acceptance tests, we are validating each of the common and advanced features proposed. First acceptance tests revealed needed improvements on the following features:

As future work we are going to develop a mobile client for the android platform and add a service layer for other future clients. We are also considering the integration of a data synchronization server (Funambol [20] DS Server), that would allow synchronization of contacts and events between our web application and our mobile client(s), and also with Microsoft Outlook, therefore providing a support for exchanging updated and real time data with other systems.

References

1. Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK). s.l. : Project Management Institute (PMI), 2008.
2. Silvers, Julia Rutherford, Event management Body of Knowledge (2003)
(<http://www.juliasilvers.com/embok.htm>)
3. Accepted Practices Exchange (APEX) Industry Glossary of terms (CIC, 2003)
4. Online Registration Center, (<http://www.onlineregistrationcenter.com/features.asp>)
Symphony Event Management Software,
(<http://www.symphonym.com/Product/Features/>)
5. EventPro Event Management Software, (<http://www.eventpro.net/features/General.pdf>)
6. MISION Event Software, (http://event-master.com/regis_key_features.html)
7. Ennect Web Marketing Tools, (<http://www.ennect.com/Event/features.aspx>)
8. OpenConf Peer Review & Conference Management System (<http://www.openconf.org>)
9. EasyChair Conference System, (<http://www.easychair.org>)
10. CiviCrm Community Site, (<http://civicrm.org/civievent>)
11. Drupal CMS, (<http://drupal.org>)
12. Joomla CMS, (<http://www.joomla.org>)
13. Storm project, (<http://drupal.org/project/storm>)
14. Fowler, Martin, Patterns of Enterprise Applications Architecture, (2002)
15. Reenska, Trygve, "MVC XEROX PARC 1978-79
" <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>
16. JAVA Enterprise Edition
(<http://www.oracle.com/technetwork/java/javaee/overview/index.html>)
17. MySQL database (www.mysql.com)
18. SOAP Simple Object Access Protocol) version 1.2, W3C, (<http://www.w3.org/TR/soap/>)
19. Funambol Open Source Mobile Cloud Sync , (www.funambol.com)