E-Conf: A Conference Management System

João Luís, *Instituto Superior Técnico, Lisboa, Portugal*

***Abstract*—Conferences are an essential channel for researchers to exchange information. Nowadays, there are hundreds of thousands of conferences every year, each with its own characteristics. However, organizing one is a difficult and time-consuming process, involving hundreds of people. There are currently many Conference Management Systems to help with this process, some of them analysed in this document, but they can be too complex or lack some needed features.**

This work introduces E-Conf, a web based Conference Management System designed and developed to help an organizer planning a conference, automating a great part of the organizing process. The features of E-Conf are divided in four main categories - main app, submissions, reviews and program management - which became the system modules.

E-Conf also includes two algorithms for automatic assignment: AREA, to assign submissions to reviewers, and Chronos, to assign submissions to sessions. When evaluated, both algorithms proof themselves capable, with AREA achieving pretty good results in the used metrics.

***Index Terms*—Conference, Conference management, Paper management, Abstract management.**

I. Introduction

CONFERENCES are an important channel for the exchange of information between researchers. As such, it should not be a surprise that nowadays conferences are numerous.

There are many types of conferences - small and big, themed and general, with many types of events - and they all need to be planned and organized.

Planning is the key to a conference success, so early planning is essential for everything to run smoothly. To begin, the conference team must be formed and the conference parameters agreed - title, themes, topics, committees, dates, venue, sponsors, etc.

With this initial planning complete, the first call for papers is issued (usually on the conference website, which is launched around this time), preparing the submission phase. Some organizations opt for a two-phase submission, with a preliminary submission first, to estimate the amount of submissions and the number of reviewers needed.

During the submission phase, researchers submit their work, either the full paper or only the abstract - according to what the organization decided. By this time, submissions can be accepted to all the types of sessions the conference will held (talks, posters, workshops, panels...) or only to some of them, the rest being accepted later.

When the submission phase closes, the submitted work is evaluated. For this phase, the organization invites a number of reviewers, according to the number of submissions, so that each submission is reviewed by (usually) at least three reviewers. The submissions are assigned to reviewers, who evaluate them with the criteria the organization decided. Reviewing may be anonymous, hiding the author information for the reviewers.

After the reviewing phase, the conference team makes the final decision on which submissions to accept. The authors of the accepted submission must then submit the full paper, if only the abstracts were submitted, or resubmit the paper making small corrections the reviewers suggested.

Around this time, registration for the conference is open, with options for early bird and standard rates, workshops, meals, etc.

Afterwards, it is time to create the conference timetable, creating sessions and assigning events to sessions. The timetable is then published on the conference website.

Finally, the conference proceedings are prepared and printed and/or digitally published.

Table I, from Ex Ordo free conference planning eBook [1], illustrates the process, showing the major milestones.

Month	
1	**Start planning**
2	**Planning complete**
3	Launch conference website
	Issue call for papers
4	Invite reviewers
6	Close call for papers
7	Open early bird registration
	Begin peer review
8	Peer review ends
	Accept/reject abstracts
9	Receive copyright, final versions
10	Close early bird registration
	Create conference timetable
11	Print hardcopy book / Publish digitally
12	Close registration
	Conference day
13	Post-conference meeting
	Final report / Handover to next chair

TABLE I
CONFERENCE ORGANIZATION TIMELINE [1, FREE CONFERENCE PLANNING EBOOK]

This whole process can involve hundreds of people, thousands of e-mails and weeks of an organizer's time. However, most of the tasks can - and, therefore, should - be automated, saving hours of administration time.

As a result, there are currently many Conference Management Systems (CMSs) in the market, with varying sets of features and different business models, which will be presented in section II. However, its difficult to find an ideal system

among them: some of them are too complex, others lack some important features, and the majority they do not offer a pleasant user experience.

A. Objectives

The main goal of this project is to design and develop a software that can automate as many of the required tasks as possible.

To do this, several existing CMSs will be studied and their features analysed, along with other related work. Based on that analysis, the system's set of features will be decided and the architecture of the new system will be designed, trying to answer to the existing problems. With the system architecture defined, the system will be implemented. Finally, the created system will be tested.

B. Document structure

This thesis is organized as follows: Section I describes the process of organizing a conference, exposing the background needed for the rest of the document. Section II lists and explains the features present in most CMSs, along with a comparison between them. Further related work is also discussed in this chapter. Section III describes the requirements for this system, obtained from the analysis in the previous chapter, and the architecture resulting from these requirements. Section IV lists the features implemented on this thesis and details their implementation. The tests of the project and its results are presented on Section V. Section VI contains the conclusions of this thesis, along with further work that can be done on E-Conf.

II. Related work

For the study required for this work, several commercial and open-source CMSs were analysed, both for features and business models. Unfortunately, for the vast majority of the analysed systems there is not a demo or trial publicly available for the common user to get in touch with the system, so the study had to rely on the feature list available on the systems' websites. Additionally, their architectures are also unavailable, so they can not be analysed.

In general the studied systems have the same overall set of features, helping on most of the tasks described on section I - set up, submission, reviewing, program management and registration - with the main difference being the level of automation done by the system.

A. Existing CMSs

The studied systems are presented below, with a full feature comparison available on table II. Some of the systems have different versions (or editions), described below and referenced in the comparative, each providing more features than the previous.

EasyChair: is currently probably the most commonly used CMS and according to their website, hosted more than 40000 conferences. It is the system with the most complete set of features of the study, although it lacks any registration functionality. EasyChair is provided as a service with three different kinds of licenses: Free, Professional and Executive. [2]

OpenConf: biggest lack is on program features, and as EasyChair has no registration features. OpenConf has three editions, Community (free), Plus and Professional, and is provided as a product, i.e., the user downloads and installs it on his server. OpenConf also provides a hosting service, serving the Professional edition. [3]

Ex Ordo: is provided as a service and has no free version. It has one of the most modern designs, along with OSEM. Differently from the previous systems, Ex Ordo sells a basic (Essential) package, which can be expanded with add-ons. [1]

EDAS: is also provided as a service and most of the features are included in the basic fee, the notable exception being the setup of the event public page. EDAS can also provide some additional material, like CD-ROMs and USB, for extra fees. [4]

ConfTool: has two different versions: VSIS and Pro. VSIS was designed for smaller events, has only the basic functions and is offered for local installation only, free on request. Pro is provided as a service and has substantially more features, being suitable for bigger and more complex conferences. [5]

OSEM: is an event management system for free software conferences. It is an open-source system and a single installation can manage multiple conferences. Its main lack is the reviewing process, as OSEM does not have any type of reviewer assignment. Along with Ex Ordo, it has one of the most modern designs. [6]

ePapers: provides two services: submission & review, which handles the whole process, and final collection, which handles only the collection of final manuscripts. According to its website, its main difference is that ePapers is a "full-service" system instead of a "self-service" system, meaning that ePapers team prepares the system for the specifics needs of its users. ePapers is the less comprehensive system of the study. [7]

B. Other work

There are some articles about CMS solutions that are also important for this study, proposing new systems or new algorithms for some specific tasks.

COMFy: is a conference management framework, instead of a system.

COMFy is divided in layers. The bottom layer is a relational database. The second layer is composed of three parts: the repositories, which abstract the database to the upper layers; the state machine, which is described as the core of the conference and manages the phase of each submission; and additional modules. The third layer its COMFy, the business logic of the CMS, which handles the requests. This layer is designed as an Model-View-Controller (MVC) pattern and exposes a well defined RESTful API - the fourth layer - which can be used to build applications - the top layer.

To detect conflicts of interest, COMFy uses, along other information, bibliographic data, such as co-authorships, which COMFy obtains from the DBLP API. [8]

TABLE II
FEATURE COMPARISON

	EasyChair	OpenConf	Ex Ordo	EDAS	ConfTool	OSEM	ePapers
Hosting and support							
Hosting on own server		C			V only	X	
Hosting on a high-performance server	F	P	X	X	P		X
Data backup and replication	F						
Encrypted access only	F				P		
Helpdesk and technical support	E	+	X				X
Miscellaneous							
Import configuration from other configuration	P						
Custom branding	P				P		
Authors and submissions							
Anonymous submissions (double-blind review)	F		X	X	V		
Custom submission types	F		X				X
Author notification	F	C	X	X	V	X	X
Instruction for authors	P		X				
Custom fields in submission form	P		X		P		
Multiple file uploads	P	C			P		
Audio, video and ZIP upload	P	P	X		V		
Terms & conditions for authors	E						
Program committee and reviewing							
Event log	F						
Watchlist	F						
Online discussion of submissions	F	+		X	P	X	
Flexible access models for reviewers	F						
Leading reviewer model	F						
Subreviewers	F			X			
Conflict of interest	F	C			P		
Author response phase / Rebuttal	F	+	X		P		
Program committee invitations	F		X	X	P		
Submission topics	F	C	X		V		
Customized decisions (acceptance types)	F	+			V		
Customizable evaluation criteria	F		X	X	P	X	
Statistics	F						
Custom review forms	P		X	X			
Multi-track support	P		X	X	P (Extra)	X	
AAAI model	E						
Paper bidding and review assignment							
Paper bidding	F	+			P		
Automatic submission assignment to reviewers	F	C	X	X	V		
Manual submission assignment to reviewers	F	C	X	X	V		X
Constraints on paper assignment	F						
Data export	P	C		X	P	X	
Email management							
Email to authors and program committee	F	C	X			X	
Email log	F						
Proceedings							
Proceedings creation	F	+	X				
Springer LNCS proceedings	F						
Program and page generation							
Program generation and publishing	P	P	X	X	P		
Page editing	P	P					
Talks and sessions editing	P	P	X			X	X
Assignments of talks to sessions	P	P	X		V	X	X
Scheduling constraints	P						X
Session chair handling	P	P		X			
Room handling	P					X	
Program manager handling	P						
Automatic schedule analysis	P						
Scheduling suggestions	P						
Solutions	P						
List of events	P						
Email to managers, authors and session chairs	P		X	X		X	
Multi-conference program generator	E						
Program page templates	E	P					
Program download	E	P	X				
Registration							
Registration			X	X	V	X	
Set custom prices			X	X	V		
Extras			X				
Payment handling			X	X	V		

EasyChair F - Free, P - Pro, E - Enterprise;
OpenConf C - Community, + - Plus, P - Pro;
ConfTool V - VSIS, P - Pro.

GRAPE: is a review assignment component for CMSs. GRAPE takes advantage from both the papers content (topics) and the reviewers preferences (biddings) and defines two measures to guide the system: reviewer's gratification and article's coverage, and tries to maximize both of them for all reviewers and all papers. GRAPE also accounts for conflicts of interest, both explicit by the organization and deducted by itself. [9]

III. Architecture

As stated in the introduction, organizing a conference is a process that usually follows a sequence of well-defined and well-delimited steps, a linear workflow described on figure 1. As such, E-Conf should be a state-based system - mainly the non-administration end - with some tasks confined to their respective phase.

The system architecture will, as a result, be designed around this workflow, as well as the features of the systems presented in II.

A. Functional requirements

The functional requirements for E-Conf are based on the feature set offered by existing CMSs, along with experience from conference chairs. They will be presented by phase, according to figure 1.

1) Set up: When first preparing E-Conf, an organizer shall be able to set the conference details - name, brand, dates and topics. An organizer shall also be able to configure the types of events the conference will hold (e.g., papers, workshops, panels...).

2) Submission: To set up this phase, an organizer shall be able to define the start and end dates of the submission phase. An organizer shall be able to configure the submission form, both by adding text instructions for authors and by adding/removing form fields, including deciding if the full paper should be submitted on this phase or only the abstract. The organizer shall also be able to issue a call for papers from E-Conf.

During the submission phase, authors (i.e., anyone registered on the system) shall be able to submit their work, as many articles as wanted. Authors shall also be able to edit their submissions during this phase.

Organizers shall also be able to enable or disable editing outside of this period, if they want.

3) Reviewing: While preparing this phase, an organizer shall be able to define the start and end dates of the revision phase, as well as for the bidding phase. Organizers shall be able to invite reviewers for the conference through E-Conf, using their emails.

Regarding the review form, organizers shall be able to set the marking scheme for papers, i.e., define the factors and their weights, for E-Conf to be able to calculate a score.

Reviewers shall be able to choose the topics of their interest, as well as be able to bid on submissions, choosing the ones they are willing or more comfortable to review, during the bidding phase.

Organizers shall then be able to assign the submissions to reviewers or order the system to do it, and then be able to edit the assignments. The system shall also detect conflicts of interest, both for manual and automatic assignment.

When the revision period starts, reviewers shall be able to review and mark the submissions assigned to them or delegate them to a subreviewer.

4) Pre-conference: After the revision phase, an organizer shall be able to accept or refuse submissions, as well as define how many submissions will be accepted. Both this value and the submission score shall be shown to organizers, although they shall not impose anything.

With the accepted submissions chosen, authors shall be able to submit the final work and pick which of the authors will be the presenter.

Program management: Also when the accepted submissions are chosen, organizers shall be able to define the venues and rooms (including their capacity) where the events will take place and create sessions. Organizers shall then be able to assign submissions to sessions or order the system to do it, and then be able to adjust the assignments. The program shall then be made available through the conference page.

B. Non-functional requirements

Given the functional requirements that were analysed in the previous subsection, E-Conf has specific requirements that justify the architecture of the solution.

1) Configurability: As stated in the introduction, conferences are numerous and varied. As such, E-Conf must be configurable and adapt to the needs of a specific conference and all its phases.

2) Modularity: A conference flow is clearly divided in phases, each with its own set of features. These features should be separated in modules, each module having its own responsibility, which simplifies the system while allowing for easy updating (or possibly reimplementation) of a single module.

3) Extensibility: Taking into account the previous two requirements, E-Conf must be also be extensible, i.e., E-Conf should be ready to accept new functionality in the form on new modules. Additionally, E-Conf shall also be extensible in an administrative way, i.e., new rules or policies may be applied.

4) Platform compatibility: Given the nature of a CMS, E-Conf will be accessed by many and different types of users. As nowadays there are many platforms and screen sizes, a user can access the system from any of them. So, E-Conf shall be easily accessed from any platform.

5) Reliability: Any data loss or down time could be critical for E-Conf. So, the system shall try to avoid any down time and, mainly, try to prevent data from being lost.

6) Security: As different users will be able to access a different set of features, authorization should be made according to the user role. This also implies that the system shall identify the user when accessing E-Conf.

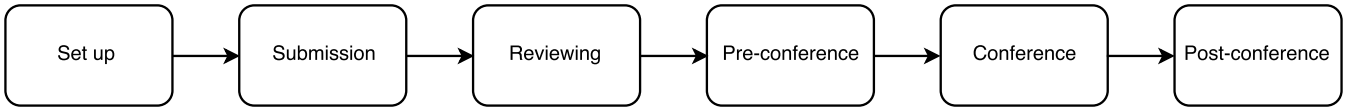


Fig. 1. E-Conf flow

7) Usability: Using the system must not require previous training from the users, not only the committees but specially the researchers, who will use the system only one or two times and should be able to submit their work easily.

C. General architecture

Some design options made from the above requirements and the analysis are presented below.

1) Web application: The best way to achieve the best platform compatibility is to develop E-Conf as a web app. Most of the devices nowadays have a web browser, allowing them to access the system, without the need to install a new software.

2) Multitenancy: Most of the CMSs are provided as a service, and as a result a single installation of the software is able to handle multiple conferences, independent from each other. Also, most of the institutions / organizations that host a conference usually host more than one. So, E-Conf must be a multitenant application: each conference has a dedicated share of the instance, including its data, configuration, etc. However, there may also be cases where this is clearly too much for the user's needs. So E-Conf must also be able to host a single conference, with no further requirements.

3) Multi language: Conferences are usually open to authors (and attendants) all around the world, so it would be nice, although not required, to be able to present E-Conf to users in their own language (or in the language they prefer).

4) Hooks: To make E-Conf extensible, a hooks system will be used, like WordPress. This method places actions and hooks through the code so that the system calls package functions at specific times.

E-Conf comprises four modules, as shown in figure 2. The main app module is the foundation of the system, and as a result all other modules need it to work.

D. Main app

The main app will handle the functionality common to all the steps, and as such is responsible for multitenancy, user management and the conference setup.

The system administrator must be able to decide if the system will host a single conference or many events. If they decide for multitenancy, a special type of user must exist, the network manager, that does not belong to any conference and manages the conferences in the system.

Users shall be able to register through the conference website, and login to access their personal area, using their email address and a password to login. To ensure proper authorization, i.e., each user can only access what they are supposed to, users shall be able to be assigned to committees. To add users to committees, an organizer or a committee chair shall be able to send an invite to an email address. When a user receives an invite, they may refuse it or accept it, creating an account if needed. Both the committee and invite systems shall be made available to the other modules. The organizing committee shall be created by the main app.

Conference chairs shall be able to setup the conference details from the main app. The chair shall be able to define naming and branding details and the conference dates, along with the conference topics and session types.

E. Submissions module

The submissions module is responsible for receiving work submissions by authors. It is also through this module that chairs can configure the submission phase.

Acceptance is also handled through this module. Although most of the times acceptance is done considering the review results, that might not happen or the review process may be handled through an external service. As such, acceptance must be handled by this module and not the reviews module.

F. Reviews module

The reviews module handles the revision process, from bidding to acceptance. Because this module is so dependent on submissions, it requires the submissions module.

To manage authorization in this module, a program committee, containing all the reviewers, shall be created.

The revision phase, and thus this module, is divided in three parts: bidding, assignment and revision.

Bidding: During the bidding phase, reviewers can bid on submissions. Reviewers can also mark if there is a conflict of interest, which are stored as a specific type of bid. The dates of the bidding period and which information will be presented to reviewers during this phase shall be configurable by chairs.

The system shall be able to detect conflicts of interest between submissions and reviewers, and these conflicts need to be marked as an actual conflict by a chair to be applied.

Assignment: Assignment occurs between the bidding and revision phases, and is responsible of assigning the submissions to the reviewers. Chairs should be able to assign in one of two ways: manually or automatically. Even if assignment is automatic, a chair shall be able to adjust the results. Assignments will be stored as a review with no score, i.e., a review to be done.

E-Conf shall include an assignment algorithm and be able to receive more easily, through the extension system.

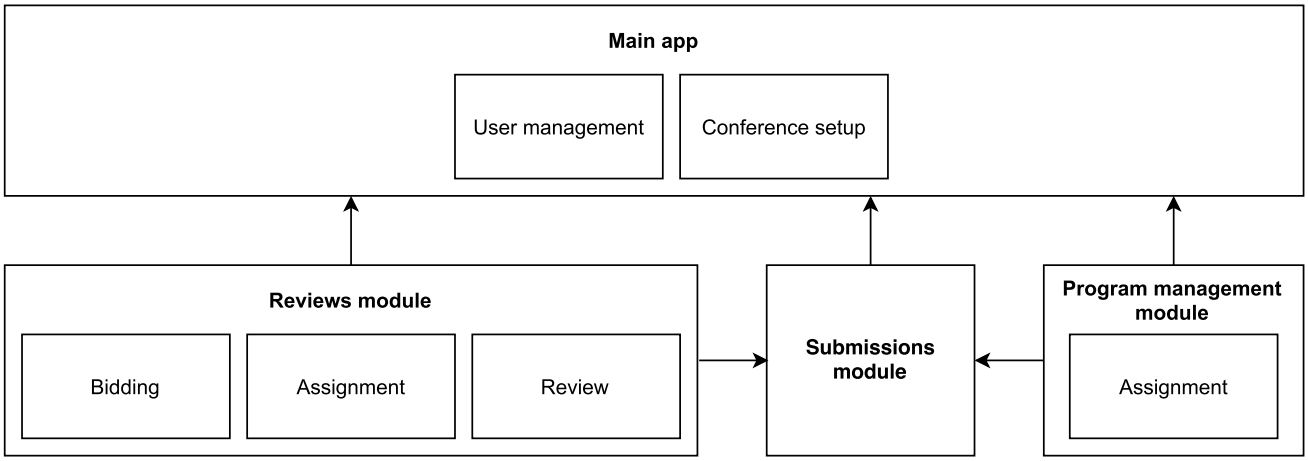


Fig. 2. System architecture overview

Revision: During this phase reviewers score the submissions, according to the parameters defined by the organization. A chair can define on which parameters submissions will be reviewed, along with their weights.

A reviewer can delegate a review to another user (a subreviewer) if they intend. This invitation will use the invite system before.

Acceptance: When the reviews module is present, it shall "take over" the acceptance screen, showing the score of each submission and ordering them accordingly. In this case, the system shall highlight the top submissions according to the number defined in the submissions module.

G. Program management

This module is responsible for managing and displaying the conference program.

A chair shall be able to create venues and rooms. A chair shall then be able to create events, and set its details, as well as sessions. Each session has a corresponding event, keeping the timetable logic in one place. The program shall then be shown on the conference public page, with details on the speakers and works.

If the submissions module is enabled, and similarly to reviews, a chair shall be able to assign submissions to sessions, either manually or automatically. Like reviews, E-Conf shall include an algorithm.

IV. Implementation

This section describes the implementation details of E-Conf. All the project's code is available at https://gitlab.com/econf.

A. Technologies

As E-Conf is a web-app, its client-side needs to be built using HTML, CSS and JavaScript. LESS was used to preprocess CSS, as it extends its syntax to include variables, mixins, nested rule sets, etc.

PHP was used to program E-Conf server-side, as it currently powers the vast majority of websites [10]. To make development easier, the Laravel framework was used. Laravel is currently the most popular PHP framework [11], has a large community and includes solutions to common problems, such as authentication, authorization, multi language, routing, debugging, events, templates, form validation, etc.

B. Multitenancy

Multitenancy in E-Conf is optional, making it an uncommon case. This features needs to be abstracted, so that E-Conf could be programmed ignoring its presence. The implementation of this feature builds on Laravel features.

First, a configuration file defines if the installation will hold one or multiple conferences. Then, to know which conference is being accessed, a middleware runs before the majority of the routes, which analyses the first URL parameter - the conference slug. If it is a known slug, the system stores the conference details and removes the parameter from the query, allowing it to be parsed the same way independently of tenancy.

Knowing which conference is being accessed, the main step is to alter the queries to the database, using query scopes and model events. This behaviour is packaged in a PHP trait that is applied to the models that shall be multitenant.

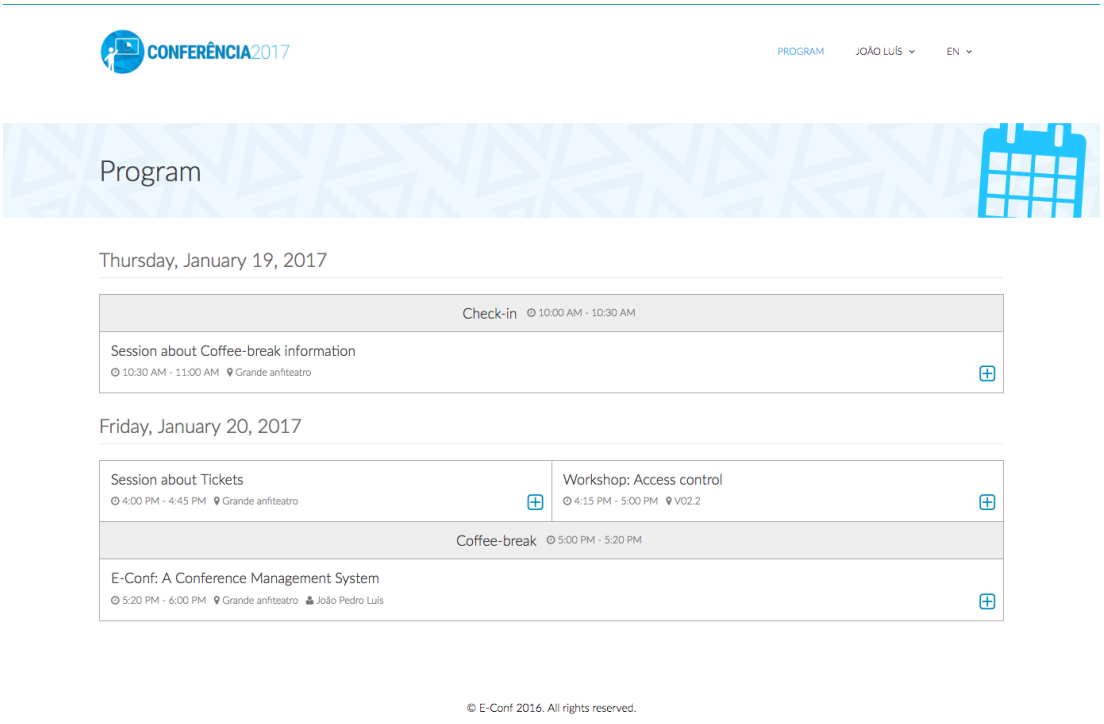
C. App module

This module is the core of E-Conf, and all other modules are installed on this app, adding them as dependencies through Composer. It is a Laravel app (built from the `laravel/laravel` repository).

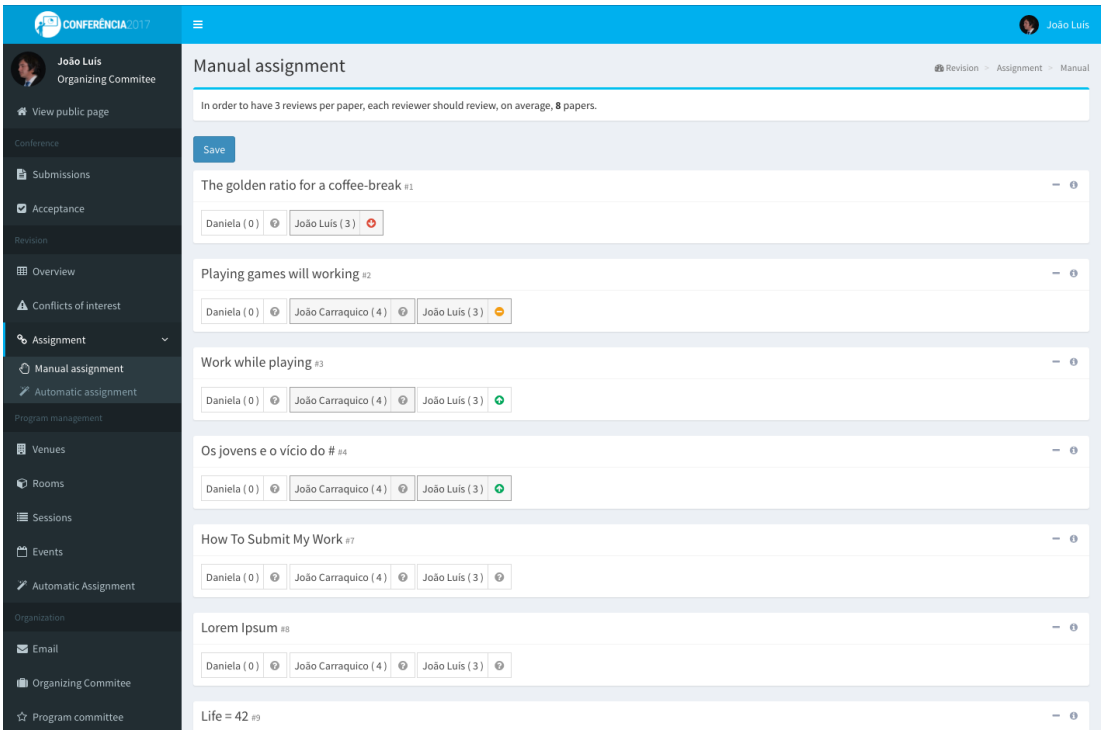
The app User Interface (UI) is split in two parts, public page and administration. The UI for both parts can be seen on figure 3.

Public page: The public page part includes the user management pages. The menu that allows to change E-Conf language is also on the public page.

Administration: The administration area is only accessible to users who belong to a committee, and it is locked through a middleware. The administration allows members of the organizing committee to configure the conference, including branding. These options are stored using the Laravel Settings package.



(a) Program public page



(b) Administration reviewer assignment pageparo

Fig. 3. App User Interface

Network administration: If multitenancy is enabled, the core app will have a third part, the network administration. From this panel the network administrator can add and remove conferences.

D. Submissions module

This module main responsibility is to receive work submissions from authors.

Submissions can be done by any registered user through the conference public page, during the defined period. The main component of this module is the submission form, which is customizable through the Submission Settings page.

A submission can have as many authors as needed, as the forms allows to add and remove authors. The list of fields available for each author is configurable through the settings page and extensible through a filter.

The submissions are stored with a sequential ID, however, showing this ID to the users can be a security breach. So, the actual ID is masked as a Hashid[1] and this is the one displayed to end users.

This module also handles acceptance. When the acceptance period is open - after the submission phase ended - the Acceptance page becomes accessible , allowing an organizer to pick the accepted submissions.

The organizer can also choose to ask for additional information when a submission is accepted, like a final document or the presenter. This data can then be filled through the public page.

E. Reviews module

The reviews module is, as stated before, divided in three parts: bidding, assignment and revision. These parts are described below.

Bidding: Reviewers can bid on submissions through the Bidding page in the administration panel. In this page, reviewers view a list with all submissions and are able to choose their bid - High, Medium, Low or Conflict.

E-Conf is, by default, able to detect conflicts between a submission authors and a reviewer by name and organization. A hook placed in the conflict detection code allows packages to add new ways to detect conflicts.

Assignment: The assignment phase occurs between the bidding and review phases. During this phase, the chairs can assign the submissions to reviewers, through two new pages in the administration panel: manual and automatic assignment.

The manual assignment page lists all the submissions and, for each submission, all the reviewers that can review it, along with the value of their bid and the number of submissions assigned to them. This page is shown on figure 3b.

The automatic assignment page lists all the available algorithms. After choosing one, its results will be presented, without saving them. A chair may then save the results and will after be able to adjust them.

E-Conf includes the AREA (Automatic REviewer Assignment) algorithm, which takes into account the reviewers' bids and the conflicts of interest and tries to maximize the reviewers' satisfaction. The algorithm is described in figure 4. New assignment algorithms can be added by external packages.

Revision: After the submissions are assigned, reviewers must review their assigned submissions. A reviewer may review the submissions assigned to them through the administration panel. The review form allows reviewers to score the submission from 1 to 5 on the parameters defined by a chair and to write text comments to the program committee and the author.

Every time a review is saved, its score is computed and stored to the database, so they are not computed each time it is viewed.

Reviewers may also delegate their reviews to other users (subreviewers) through the reviews list. These users do not need to be a part of the program committee.

F. Program management module

The program management module is responsible for managing and showing the conference program. Organizers can manage venues and rooms through the Program section of the administration panel. Organizers can then create events, which have the start and end times, the room where they take place and other details.

Organizers can also create sessions. Sessions are special events where submissions will be presented. Each session has a session type and a topic, along with its submissions. As it was said before, each session has a corresponding event, allowing queries to the program to be more efficient.

E-Conf also creates a program page on the conference public page, shown on figure 3a.

Assignment: Organizers can assign submissions to sessions though this module. This assignment can be done manually or automatically.

The manual assignment can be done from the administration session page. From this page organizers can add submissions to a session, and order them.

Automatic assignment works similarly to the reviews automatic assignment. When the Automatic assignment page is accessed, organizers are presented with a list of algorithms and, after an algorithm is picked, the results of that algorithm are presented.

E-Conf includes the Chronos algorithm, described in figure 5, which takes into account the scores and the topics. New assignment algorithms can be added by external packages.

V. Evaluation

Like every other system, E-Conf needs to be tested and evaluated, to check if it is ready to use in a real world scenario.

A. Algorithm tests

Both the algorithms created, AREA and Chronos, were tested and evaluated.

To make the results more accurate, the algorithms were tested with real data, from the 12th Conference on PhD

[1] http://hashids.org

```
Function AssignForValue (bidValue)
    valueReviewers ← Bids.withValue(bidValue).reviewers
    .sortBy(numBids) foreach valueReviewers as reviewer do
        reviewerBids ← reviewer.bidsWithValue(bidValue).submissions
        .sortBy(numBids);
        foreach reviewerBids as submission do
            subs_revs[submission.id].add(reviewer.id);
            revs_subs[reviewer.id].add(submission.id);
            if subs_revs.length ⩾ reviews_per_submission then
                submissions.remove(submission);
            if revs_subs.length ⩾ submissions_per_reviewer then
                reviewers.remove(reviewer);
                break
Function AssignRemainingSubmissions
    foreach submissions as submission do
        submReviewers ← ProgramCommittee.canReview(submission)
        .sortBy(topicMatch);
        foreach submReviewers as reviewer do
            subs_revs[submission.id].add(reviewer.id);
            revs_subs[reviewer.id].add(submission.id);
            if subs_revs.length ⩾ reviews_per_submission then
                submissions.remove(submission);
            if revs_subs.length ⩾ submissions_per_reviewer then
                reviewers.remove(reviewer);
                break
submissions ← Submissions.all();
reviewers ← ProgramCommittee.members();
bids ← Bids.all();
subs_revs ← [];
revs_subs ← [];
reviews_per_submission ← getFromSettings();
submissions_per_reviewer ← ceil(submissions.length×reviews_per_submission / reviewers.length);
AssignForValue(High);
AssignForValue(Medium);
reviewers ← reviewers.sortBy(numAssignments);
AssignRemainingSubmissions();
if not submissions.empty then
    reviewers ← ProgramCommittee.members();
    AssignRemainingSubmissions();
return subs_revs;
```

Fig. 4. AREA algorithm

```
submissions ← Submissions.accepted().groupBy(topic);
sessions ← Sessions.all();
sessions_submissions ← [];
while not submissions.isEmpty or not sessions.isEmpty do
    submissions.sortByDesc(length);
    sessions.sortByDesc(remainingSubmissions);
    session ← sessions.shift();
    chosen_submissions ← submissions.first
    .slice(0, session.remainingSubmissions);
    submissions.remove(chosen_submissions);
    if session.full then
        sessions.remove(session);
foreach sessions_submissions as session do
    session.shuffle();
return sessions_submissions;
```

Fig. 5. Chronos algorithm

Research in Microelectronics and Electronics (PRIME 2016). To import the data and compute the values, a fifth module was created (Evaluation), which imports data from a spreadsheet, exported from EasyChair.

Reviewers assignment algorithm (AREA): Two metrics were used to test this algorithm: reviewer satisfaction and topic matching. These metrics were computed after importing all the data from PRIME and using AREA to assign submissions to reviewers, configured to assign three reviewers per submission.

The results were then compared with the assignments EasyChair made for PRIME.

Reviewers satisfaction: Reviewer satisfaction describes how much a reviewer's assignments match their bids. The Evaluation module computes this metric for each reviewer and then uses that data to present some statistics.

Taking into account only the reviewers who have done their bidding, the average reviewer satisfaction is 84.69%. While this is a good rate, there is clearly space for improvement.

Topic matching: After assigning submissions to the users who bided, AREA assigns the remaining submissions matching the submission and the reviewer topics.

Considering all the reviewers who have not done their bidding, the average topic matching is 8.12%. However, if only reviewers who chose more than one topic are considered, the average topic matching raises to 82.56%. These results show that the algorithm performs better if the reviewers choose more than one topic of interest, because submissions have usually more than one topic.

Comparison with PRIME: Another interesting evaluation to consider is the comparison between the AREA results and the actual assignments for PRIME, both for the metrics above and the assignments done.

55.86% of the assignments done by AREA were also done for PRIME. This value, by itself, does not say much, although it is still interesting that both methods share more than half of the assignments.

To get the values for reviewers satisfaction and topic matching, the assignments from PRIME were imported and the same computations were done. The reviewers satisfaction, 81.96%, is pretty close to the value obtained by AREA. However, the topic matching value is greatly inferior: 2.19% for all reviewers with no bids and only 10% for the reviewers with more than one topic.

Session assignment algorithm (Chronos): The Chronos algorithm is a very simple algorithm that mainly takes into account topics. So, the metric used to evaluate this algorithm was the topic grouping, i.e., whether the submissions in a session shared a topic.

To test the algorithm, the acceptance data from PRIME was imported, a program similar to PRIME's was created on E-Conf and the submissions assigned using Chronos.

The average topic grouping for these assignments were 100%. Closer inspection of individual sessions shown that, for the majority of the sessions, the common topic was the same topic. This shows that there is clearly room for improvement, even if taking only in account topics. For instance, topic pairs could be considered.

Other improvements can be done to this algorithm, as stated in VI-A.

B. Functionality tests

To test if the system could fulfil its main goal, it was run for the whole organization process, from the set up to the program generation.

To do these tests a mock conference was created - TEST 2016 - and organized using only E-Conf. All the needed data was generated with the help of the Faker package [2] and inserted automatically, but all the remaining interactions, like assigning submissions to reviewers or creating sessions, was done through the E-Conf UI.

Besides a struggle with time when storing the assignments, E-Conf was able to handle the whole process with no problems, and requiring minimal input from the organizer.

VI. Conclusions

Conferences are an essential channel for the exchange of information, and as result there are hundreds of thousands of conferences every year. Organizing a conference is a long and hard process, involving hundreds of people. Most of the tasks are repetitive and, therefore, ideal to delegate to a computer. There are already several systems to solve this problem, Conference Management Systems, although they have some restrictions. The work shown in this document proposes E-Conf, a new CMS.

To find which features were required, other CMSs were analysed. They share the same overall set of features, which can be grouped in five main categories - submission, reviewing, pre-conference, registration and public page. The set of features on E-Conf, based on this analysis, covers three of this five areas.

E-Conf was implemented as a web-app, using the most used web technologies today, making E-Conf accessible to anyone with a web browser. The clear separation of the features in categories was used to group the functionality in modules, one for the main app and one for each category.

The evaluation performed showed that E-Conf can fulfil its purpose. The AREA algorithm evaluation also showed positive results (similar or better than PRIME's) for the used metrics. The Chronos algorithm, however, despite the good results in the metrics used, clearly needs improvement - or to be replaced.

A. Future Work

The addition of new modules focusing on registration and public page would make E-Conf a CMS able to cover the whole organizing process, with no external tools required.

The assignment algorithms are another area that could be enhanced in the future. E-Conf includes two basic algorithms, but a thorough study on these aspects could produce much better algorithms.

Regarding UI and customization, E-Conf could benefit from a theming engine, allowing chairs to easily change their public page appearance. This is already possible if a module overrides most of the templates, but this process could be made easier.

Another feature that would open many doors for E-Conf is the creation of an Application Programming Interface (API). This could allow the integration of E-Conf in other systems and many other creative uses, e.g. a complete new interface for the system.

[2] https://github.com/fzaninotto/Faker

References

[1] "Ex ordo — award winning abstract management software," http://exordo.com/, (Visited on 14/Dec/2015).

[2] "Easychair," http://www.easychair.org/, (Visited on 14/Dec/2015).

[3] "Openconf peer-review, conference and abstract management software system," http://www.openconf.com/, (Visited on 14/Dec/2015).

[4] "Edas: Editor's assistant," https://edas.info/doc/, (Visited on 14/Dec/2015).

[5] "Conftool: Conference and event management software," http://www.conftool.net/, (Visited on 14/Dec/2015).

[6] "Open source event manager," http://osem.io/, (Visited on 14/Dec/2015).

[7] "Welcome to epapers!" http://www.epapers.org/main/, (Visited on 14/Dec/2015).

[8] C. Caldera, R. Berndt, and D. W. Fellner, "Comfy –a conference management framework," *Inf. Serv. Use*, vol. 33, no. 2, pp. 119–128, Apr. 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2596858.2596867

[9] N. D. Mauro, T. M. A. Basile, and S. Ferilli, "Grape: An expert review assignment component for scientific conference management systems." in *IEA/AIE*, ser. Lecture Notes in Computer Science, M. Ali and F. Esposito, Eds., vol. 3533. Springer, 2005, pp. 789–798. [Online]. Available: http://dblp.uni-trier.de/db/conf/ieaaie/ieaaie2005.html#MauroBF05

[10] "Usage of server-side programming languages for websites," https://w3techs.com/technologies/overview/programming_language/all, (Visited on 05/Oct/2016).

[11] B. Skvorc, "The best php framework for 2015: Sitepoint survey results," http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/, Mar. 2015, (Visited on 02/Jan/2016).