

REPORTER A STRAIGHTFORWARD WEB TOOL FOR COLLABORATIVE DOCUMENT PRODUCTION

M. ARTIOLI, R. GUIDA

ENEA – Unità Tecnica Efficienza Energetica
Servizio Analisi e Valutazione di Sistemi per l'Efficienza Energetica
Centro Ricerche "Ezio Clementel", Bologna

G. FERRO

ENEA – Unità Tecnica Metodi per la Sicurezza dei Reattori e del Ciclo del Combustibile
Laboratorio Sviluppo Metodi e Supporto Informatico
Centro Ricerche "Ezio Clementel", Bologna



AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE,
L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

REPORTER A STRAIGHTFORWARD WEB TOOL FOR COLLABORATIVE DOCUMENT PRODUCTION

M. ARTIOLI, R. GUIDA

ENEA – Unità Tecnica Efficienza Energetica
Servizio Analisi e Valutazione di Sistemi per l'Efficienza Energetica
Centro Ricerche "Ezio Clementel", Bologna

G. FERRO

ENEA – Unità Tecnica Metodi per la Sicurezza dei Reattori e del Ciclo del Combustibile
Laboratorio Sviluppo Metodi e Supporto Informatico
Centro Ricerche "Ezio Clementel", Bologna

I Rapporti tecnici sono scaricabili in formato pdf dal sito web ENEA alla pagina
<http://www.enea.it/it/produzione-scientifica/rapporti-tecnici>

I contenuti tecnico-scientifici dei rapporti tecnici dell'ENEA rispecchiano l'opinione degli autori e non necessariamente quella dell'Agenzia.

The technical and scientific contents of these reports express the opinion of the authors but not necessarily the opinion of ENEA.

REPORTER A STRAIGHTFORWARD WEB TOOL FOR COLLABORATIVE DOCUMENT PRODUCTION

M. ARTIOLI, R. GUIDA, G. FERRO

Abstract

Producing periodic reports on activities carried out in large projects that involve teams geographically distributed is usually a really time consuming task that, despite modern technologies, often is still performed making a manual collage of all needed contributions. Even though web based solutions expansion into areas once dominated by desktop applications offers some online collaboration tools, it is still difficult to find simple to use collaborative web tools designed to support preparation of common documents among people belonging to large communities . In order to support periodic progress reports production in the frame of the European Joint Program on Nuclear Materials, it has thus developed Reporter which, thanks to its flexibility, can also be used to prepare other types of documents where a composition of contributions coming from several people is needed.

Keywords: collaborative tools, co-editing, project reporting

Sommario

La produzione di report periodici sulle attività eseguite in grandi progetti che coinvolgono gruppi geograficamente distribuiti è solitamente un compito che richiede una grande quantità di tempo e, nonostante le moderne tecnologie, viene ancora frequentemente eseguito manualmente facendo un collage dei contributi necessari. Nonostante l'espansione delle soluzioni web in aree una volta dominate da applicazioni desktop abbia reso disponibili alcuni strumenti collaborativi online, è ancora difficile trovare semplici strumenti collaborativi web studiati per supportare la predisposizione di documenti comuni tra persone appartenenti a grandi comunità. È stato quindi sviluppato *Reporter* al fine di fornire un supporto alla produzione di report periodici nell'ambito del programma europeo JPNM (Joint Program on Nuclear Materials). Tale strumento, grazie alla sua flessibilità, si presta ad essere utilizzato anche per predisporre altri tipi di documenti in cui è richiesta la composizione di contributi provenienti da più persone.

Parole chiave: strumenti collaborativi, coedizione, relazioni progettuali

INDEX

1 – INTRODUCTION	7
1.1 - Scope and user requirements	7
1.2 - State of the art	8
1.3 - A new solution almost from scratch	9
1.4 - Use cases	9
2 – ARCHITECTURAL IMPLEMENTATION	11
2.1 - Presentation logic	11
2.1.1 - Client side	12
2.1.2 - Server side	12
2.2 - Business logic	12
2.2.1 - Content management	12
2.2.2 - User management	18
2.3 - Data logic	18
2.3.1 - Data structures	18
2.3.2 - Data Base Management System	19
3 - CONCLUSIONS	20
ACKNOWLEDGEMENTS	21
REFERENCES	22

1 – INTRODUCTION

1.1 - SCOPE AND USER REQUIREMENTS

Producing periodic reports on activities carried out in large collaborative projects that involve teams geographically distributed is usually a really time consuming task, especially for coordinators that have to:

- Request a contribution from each partner involved
- Collect all the partial progress reports and assemble all the information in a single document

In order to accomplish this task dozens of e-mails are usually exchanged, the mailbox of the coordinator clogs up and too many “last” revisions of documents circulates.

Aiming to foster better networking and collaboration among researchers belonging to the large community (32 research centers and universities from 15 different countries) of EERA-JPNM (European Energy Research Alliance - Joint Program on Nuclear Materials) [1], a simple collaborative web tool to support preparation of common documents has been developed. Even though it has been named *Reporter*, it can also be used to prepare other types of documents.

The main goal has been to have a tool:

- Simple to use (no long manuals to read or heavy courses to follow)
- Interoperable (users not tied to a single operating system and/or word processor)

In *Reporter* a user with an “Editor” profile

- Creates a new report
- Provides a template
- Defines title and structure (index) of the document
- Assigns each index entry (chapter/paragraph) to another user (“Writer”)
- Oversees drafts
- Prints the final document

All other users have usually a “Writer” profile and they have only to produce and attach a pdf file for each assigned chapter following the given template.

Some focused automatic e-mail are foreseen in order to inform a user (Writer) that a chapter/paragraph has been assigned to him or give evidence to the Editor that a Writer accomplished its task or to remind a Writer a deadline.

Reporter is available only to registered users. A normal user (Writer) can see only those chapters/paragraphs assigned to him, if any.

1.2 - STATE OF THE ART

Collaborative software is application software designed to help people involved in a common task to achieve goals and therefore they transform the way documents are shared to enable more effective team collaboration.

Client-server solutions generally allow a better integration (or complete replacement) of office suite applications, requiring, on the other hand, that the user installs the application on his workstation.

Web based solutions are generally less performing, but virtually accessible and operable from everywhere.

Collaborative software is a rich market share for office application vendors. Open source solutions are available as well, in growing number, and a complete and definitive list does not exist.

With cloud computing continuing its reach into areas once dominated by desktop applications, teams geographically distributed could benefit from online collaboration tools for producing documents. Whether it's a proposal, press release, manual, or other type of communication or documentation, there are some tools to help prepare it.

Wiki software is, for example, an application that allows multiple users to collaborate on interlinked web documents relatively easily using a WYSIWYG editor or a lightweight markup language. The most famous example is [Wikipedia](#), a user-editable knowledge base that has quickly grown into the world's largest encyclopedia.

Wikis are an excellent choice for creating complex documentation, since pages and topics can be very easily linked together to make a navigable document structure.

Wikis can also often be a good choice for collaborating on single standalone documents, particularly if those documents need to live online. Some wiki systems have export features as well that allows to convert an online wiki page into a PDF, text file or other portable document format.

If, as in our case, needs are simpler than extensive documentation or information gathering and it is just needed a way to collaborate on essentially standalone documents, it is possible to consider collaborative online document editors. Perhaps the most famous among them is [Google Documents](#). Users can make use of a rather robust WYSIWYG visual editor to prepare documents that are shareable in either view or edit levels with whoever needs to have access. With the addition of a shared folders feature, it's even easier to manage collaboration and access among one or several sets of team members.

Google is not the only choice, however, and there are other free alternative options but when more than 2 or 3 people have to build a document together all these tools becomes quite impractical.

Other notable applications in the online document preparation category include solutions provided, for example, by Adobe or Microsoft although all the team would actually use the corresponding desktop apps to view and edit documents.

1.3 - A NEW SOLUTION ALMOST FROM SCRATCH

Since all tested solutions didn't satisfy our very basic requirements in terms of simplicity, interoperability and, last but not least, freeware licensing, a new solution almost from scratch has thus been developed.

The tasks to be done, like writing and reviewing reports, are tedious but conceptually simple and the software tool has to reflect the ease of use not only in the friendliness of the interface but also in the effectiveness of the workflow.

Customizations of packaged solutions revealed to be tricky and difficult to maintain even if designed with modular plug-ins. On the other hand, widespread reliable software components like pdf manipulation libraries in PHP or MySQL connectors enable a relatively fast "in-house" development of such an application.

The design has been kept modular to guarantee future expansions or customizations.

1.4 - USE CASES

Users have to signup. Once registered they receive the default *Writer* profile. Only an *Administrator* can give the *Editor* privileges to a user, which still remains also a contributing writer.

After logging in, to the user is presented his type of panel accordingly with his profile. From the panel, the user will be able to operate as depicted in Fig. 1.1, which reflects diagrammatically the different types of users of *Reporter* and the various ways that they interact with the system.

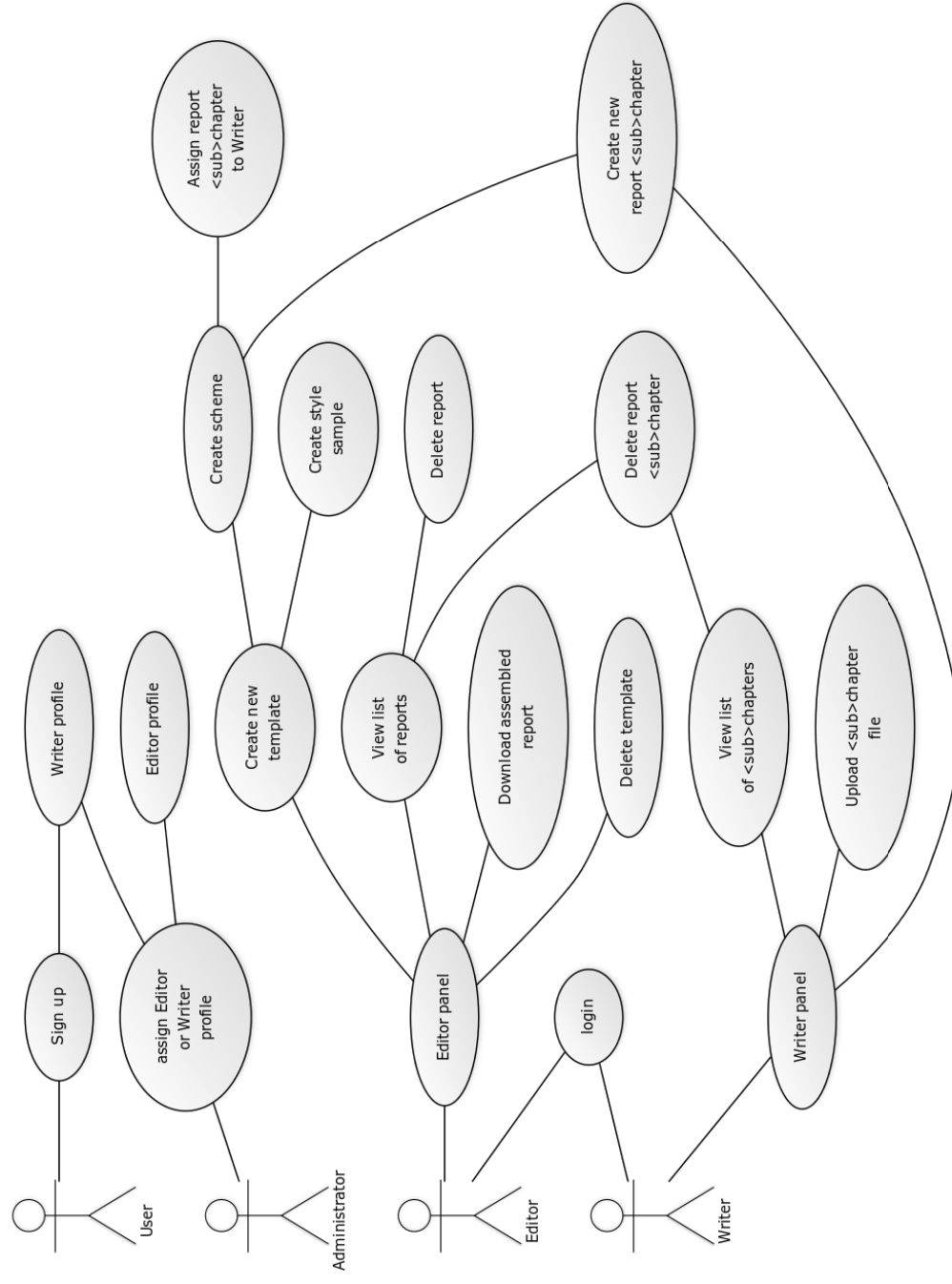


Fig. 1.1.1 : Use cases representation of users interactions with the system

2 – ARCHITECTURAL IMPLEMENTATION

2.1 - PRESENTATION LOGIC

The presentation logic is founded on a well-established web paradigm (Fig. 2.1). Pages are dynamically rendered partly in the web browser of the user and partly on the web server. The front-end is not limited to the interactive presentation function, but also implements some restrictions and suggestions on data entry to ease user operations, minimize user errors and enforce security. This checking is performed both client and server side.

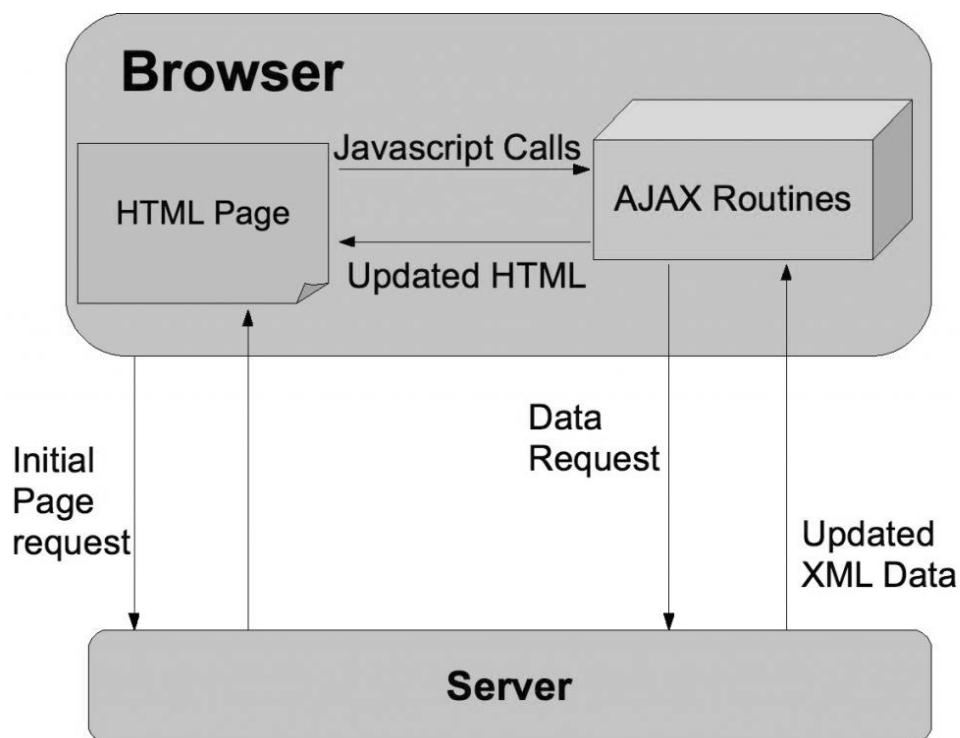


Fig. 2.1 : Deployment diagram

2.1.1 - CLIENT SIDE

The chosen technologies used to display the user interface with dynamic content and interactive controls (like buttons, menus, expandable trees, text input fields, etc.) are:

- HyperText Markup Language (HTML5 conformant);
- Content Style Sheets (CSS3);
- Java scripting (JavaScript)
 - with JQuery libraries;
 - with Asynchronous JavaScript and XML (AJAX).

HTML5 is a mandatory choice for standard compliant site and as complementary element to JavaScript and AJAX. Of prominent importance is AJAX, a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With it, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page.

2.1.2 - SERVER SIDE

The only significant technology adopted on the server is PHP, which actually bridges the presentation logic (html content generation) with the business logic like pdf documents assembly and user-profile dependent content generation.

2.2 - BUSINESS LOGIC

The business logic is simple and covers two basic functions:

1. content management and,
2. user management.

2.2.1 - CONTENT MANAGEMENT

Contents are chunks of reports in pdf format uploaded by users. Since there is no direct editing facility for the documents on the website, because they are created by the users on their own following a template, the only functions to implement are those for assembling pdf chunks in the right order and creating a table of contents for the final document.

Templates

Templates are composed by three elements:

1. document style
2. contributing writer template file
3. document cover

The document style is defined by a set of parameters, depicted in Fig.2.2, that describes formatting (font type and size, page size and layout).

The screenshot displays a user interface for configuring document style parameters, organized into eight distinct sections:

- Template name:** A text input field containing the value "Annual_report_template".
- Margins:** A control bar featuring four circular icons: a double-headed arrow, a top arrow, a bottom arrow, and a left-right arrow.
- Cover:** A toolbar with icons for font face (A), font size (drip), bold (B), italic (I), text color (T), text background color (T), bulleted list, numbered list, and a diamond icon.
- Index:** A toolbar with icons for font face (A), font size (drip), bold (B), italic (I), text color (T), text background color (T), bulleted list, numbered list, and a gear icon.
- Pages number:** A toolbar with icons for font face (A), font size (drip), bold (B), italic (I), text color (T), text background color (T), bulleted list, numbered list, and a gear icon.
- Chapter titles on index:** A toolbar with icons for font face (A), font size (drip), bold (B), italic (I), text color (T), text background color (T), bulleted list, numbered list, and a diamond icon.
- Chapter titles:** A toolbar with icons for font face (A), font size (drip), bold (B), italic (I), text color (T), text background color (T), bulleted list, numbered list, and a diamond icon.
- Chapter titles when no file found:** A toolbar with icons for font face (A), font size (drip), bold (B), italic (I), text color (T), text background color (T), bulleted list, numbered list, and a diamond icon.

Fig. 2.2 : Sample screen shot of *Reporter* document style parameters

The contributing writer template file is a downloadable scheme to follow that Writers can use to provide their contributions sure that they have the structure requested by the Editor.

The structure of this file is a naked structure: a logic hierarchy of paragraphs and sub-paragraphs with relevant titles and assignments to the writers. As this logic is entirely contained in the relations between report and chapters and between Editor and Writers, there is no real code implementing this level. This function is realized by managing the data structure from the front-end (see Fig 2.3 and Fig. 2.4 for an example).

Participant¹

Concise Activity Progress Report²

Title³

<u>Contact researcher⁴:</u>	Name (+ e-mail)
<u>Scientific Staff⁵:</u>	Names
<u>Collaborators⁶:</u>	Names (+affiliation)

OBJECTIVES
Text (5-10 lines max)

MAIN RESULTS

- Text (in bullet form, concisely listed, details in relevant publications or referable reports cited below)⁷

OUTLOOK
Text (5-10 lines max)

REFERENCES

[1] *Papers/reports cited in text above, in J. Nucl. Mater. format*

1 Organisation that performed the work and produces the report

2 3 pages max. – **remove these footnotes** from the actual report: they are meant only to explain the use of the template. The report is to be uploaded on Reporter under the relevant task or subtask. Upload pdf file.

3 Freely decided, may coincide with subtask

4 Main author of the report

5 Other researchers from the same organisation involved in the work

6 Other researchers from other organisations in the JPNM (specify organisation as well) involved in the work, mention if relevant also organisations not in the JPNM

7 Tables and figures may be added if they convey a clearer message than just text

Fig 2.3 : Example of a structured template file for contributing writers

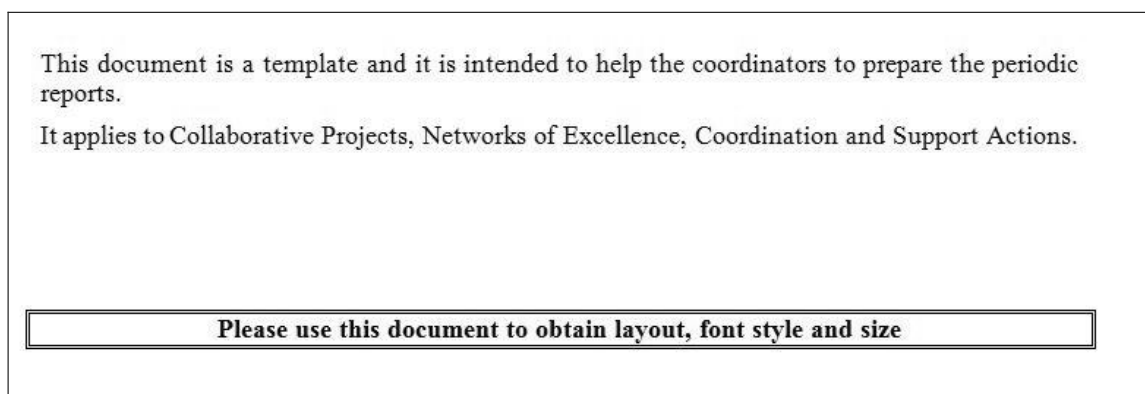


Fig. 2.4 : Example of a simple template file for contributing writers

The document cover, instead, is a file that is used by Reporter to compose the first page of the document (that, for example, could have a logo, and/or a watermark).

A template can be private or public. A private template can be used and modified only in documents created by the Editor who made it while a public template can be used, but not modified, by other Editors.

Document structure (index) definition

Structure of a document is not fixed. This allow to produce different kinds of documents with a different organization of contents. It is therefore possible to build the classic distribution of contents of a book (chapters/paragraphs/subparagraphs) as well as, in case of reporting on a project, it is possible to build a structure that reflects allocation of responsibilities and tasks like in Fig. 2.5.

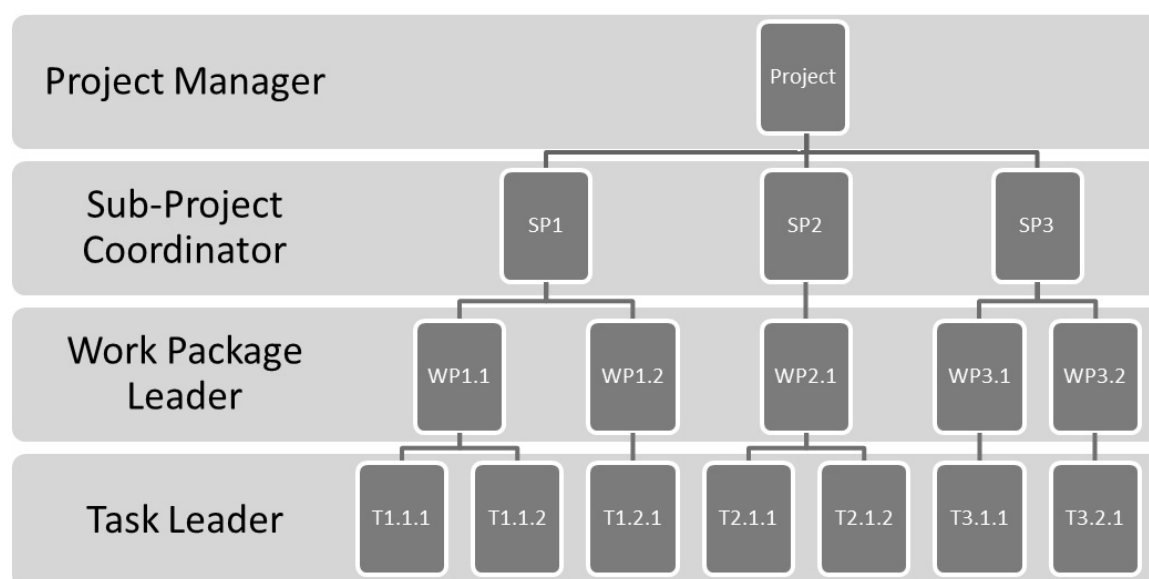


Fig. 2.5 : Example of allocation of responsibilities and tasks in a project

Each index entry must have a person in charge and may be subleveled by the assignee but actually it is not mandatory to attach a pdf file to each index entry. This feature allows having documents with a structure like that

1. First chapter
 - 1.1 First paragraph
 - 1.2 Second paragraph
2. Second chapter
3. Conclusions

where “First chapter” is only a title while the text is only in its paragraphs.

The Editor may assign a chapter/paragraph to himself. Actually, it is advisable that Editors start assigning all index entries to themselves and to reassign each item only when the structure of the document is clear/completed.

Each index entry is defined by four parameters

1. *Title*
is the name of the chapter/paragraph that will be inserted. Numbering should not included in the title since Reporter numbers all chapters/paragraphs automatically
2. *Description*
is a comment that, together with the other three parameters, will be automatically sent by e-mail to the Writer whom the chapter is assigned. The comment should have enough information to allow the writer to understand what the Editor does expect to receive
3. *Deadline*
is the due date the chapter/paragraph must be completed by (it may be different from the general deadline). This information is used by Reporter, if the report is not completed, to send an automatic reminder to the Writer three days before and three days after.
4. *Assigned to*
is the name of a registered user that should be in charge for the chapter/paragraph. The field has a smart lookup finder that, after typing two chars, shows all corresponding users.

The index, once built, if necessary, can be edited by changing titles and position of index entries that can be also deleted or inserted

Content assembly

At any moment, the Editor can request a document assembly: that is all the contributions are concatenated and a cover and a table of contents is prepended. The assembly happens on the fly and the result is not stored, to avoid misalignments with the source parts.

To this purpose a button in the front-end triggers a PHP routine which in turn calls functions to manage pdf files and then offers the final document to be downloaded.

The routine, as depicted in the flowchart (Fig. 2.6), basically queries the database for the record structure of the report, builds cover and table of content (with the given style), and then joins the file submitted for every record, numbering the pages.

It uses the Free PDF Document Importer (FPDI) library, a collection of PHP classes facilitating developers to read pages from existing PDF documents and use them as templates in Free PDF (FPDF) library. Apart from a copy of FPDF, FPDF does not require any special PHP extensions.

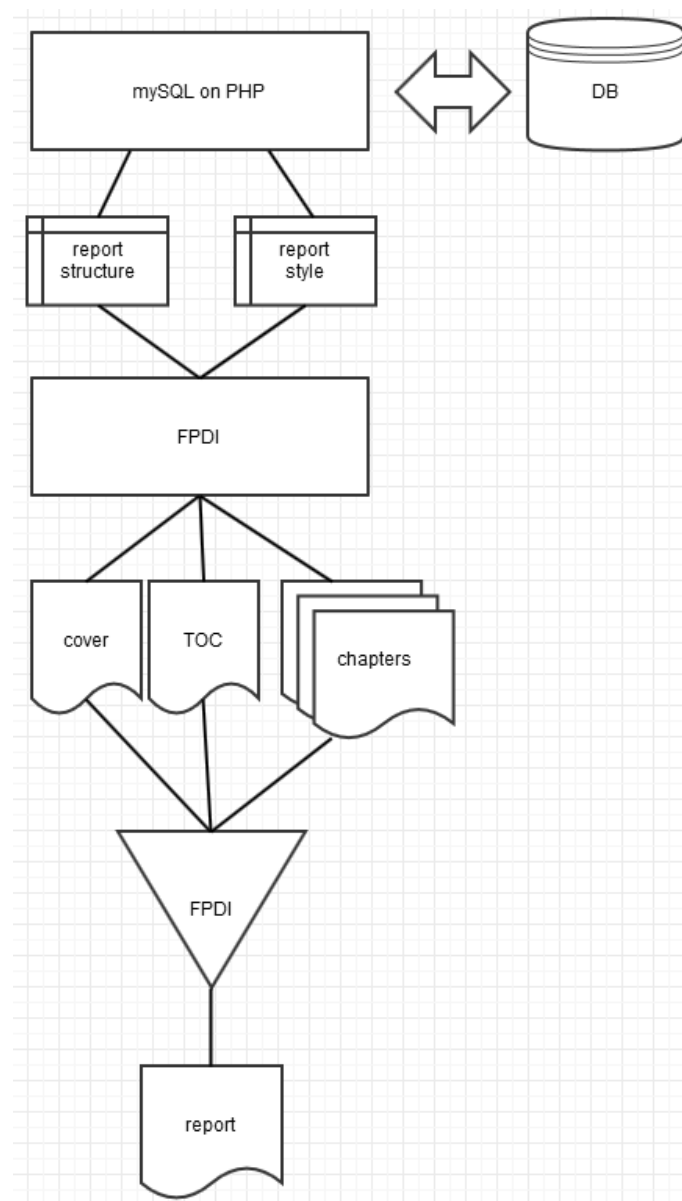


Fig. 2.6 Content assembly flowchart

As shown in Fig. 2.7, an Editor can also monitor the level of completion of the document browsing on line the index since all index entries flagged as ready by Writers are presented with a green background.

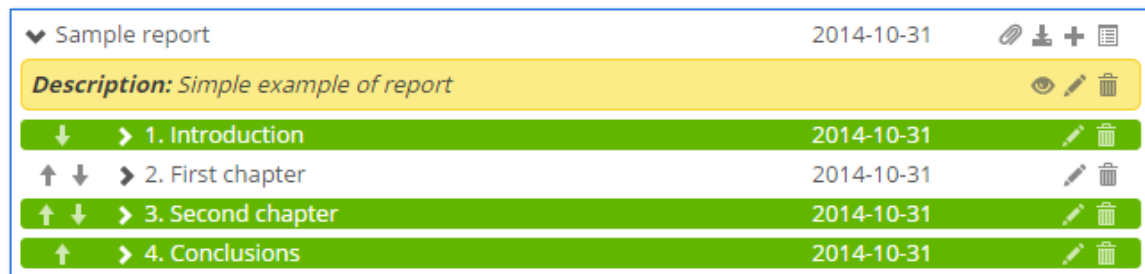


Fig. 2.7 : Sample screen shot of *Reporter* document completion monitor

2.2.2 - USER MANAGEMENT

Also user management is done from the front-end by manipulating the data structure. It is very simple, because the only function is for the Editor to assign (or re-assign) a report section to a Writer. All other functions (like profile assignments, user banning, user deletion, etc) are reserved to Administrators that operate with administration tools external to the *Reporter* web application.

2.3 - DATA LOGIC

2.3.1 - DATA STRUCTURES

Data structures (Fig. 2.8) basically model two entities: users and reports. Users have several details associated, as usual. Reports are represented in table containing all chunks and related parent-child relationships to reconstruct the whole hierarchy outlined in the report template.

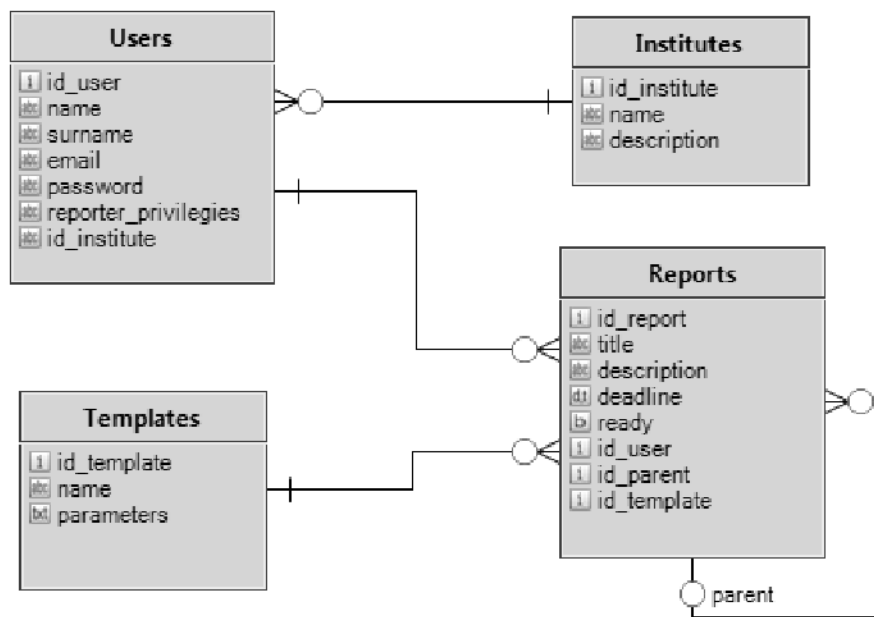


Fig. 2.8 : Structure of the database and relationships between collections of data

2.3.2 - DATA BASE MANAGEMENT SYSTEM

Due to convergence of MySQL and PostgreSQL in terms of acceptance, features and performance, the former has been adopted mainly as matter of taste. The application is not "data-critical" and the Data Base Management System has no special requirements.

3 - CONCLUSIONS

Reporter has been employed to support report production activities of the Joint Programme on Nuclear Materials (JPNM) of EERA (European Energy Research Alliance). During this preliminary test has been possible to collect and assemble in a single document about 120 different contributions. More than 90 % of the Writers were able to use *Reporter* without any assistance, about 5% needed more information while the remaining 5% presented some simple technical issues concerning a specific version of their web browsers that have been easily solved. The total amount of time needed to the Editor to build and manage the report until its completion has been estimated about 20% of the effort necessary to obtain the same result manually.

Reporter has proven to have the potential to become a valuable tool to foster better networking and collaboration among researchers belonging to a large community in the important task of preparing common documents and it is under evaluation its adoption also in other joint programmes of EERA.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge Dr. S. Pagnutti for many fruitful discussions.

This work has been developed in the frame of the project MatISSE (Materials' Innovations for Safe and Sustainable nuclear in Europe) that is supported by the European Commission under the Euratom Theme of the 7th Framework Programme for Research and Technological Development under grant agreement n.604862.

This work also contributes to the Joint Programme on Nuclear Materials (JPNM) of the European Energy Research Alliance (EERA).

REFERENCES

- [1] C. Fazio, “Nuclear materials research in EU: present status and future perspectives”, 2012; 28 p, Nordic-Gen4 seminar, Risoe, Roskilde (Denmark), 29-31 Oct 2012, available at <http://nordic-gen4.org/seminars/nordic-gen4-riso-2012-2/>

Edito dall' **ENEA**
Servizio Comunicazione

Lungotevere Thaon di Revel, 76 - 00196 Roma

www.enea.it

Stampa: Tecnografico ENEA - CR Frascati
Pervenuto il 28.1.2015

Finito di stampare nel mese di febbraio 2015