The Modelery: A Collaborative Web Based Repository

Rui Couto, António Nestor Ribeiro, and José Creissac Campos

University of Minho,
Braga, Portugal
{rui.couto,anr,jose.campos}@di.uminho.pt
http://di.uminho.pt

Abstract. Software development processes are known to produce a large set of artifacts such as models, code and documentation. Keeping track of these artifacts without supporting tools is not easy, and making them available to others can be even harder. Standard version control systems are not able to solve this issue. More than keeping track of versions, a system to help organize and make artifacts available in meaningful ways is needed. In this paper we review a number of alternative systems, and present the requirements and the implementation of a collaborative web repository which we developed to solve this issue.

Keywords: Model Driven Development, Models repository, Web collaborative repository

1 Introduction

Research into software development processes typically produces a large amount of artifacts, from documentation and different kinds of models to the actual code. Organizing and sharing those artifacts has shown to be somehow a difficult task, due to the lack of effective support. We are particularly interested in the development of tools and techniques to support software engineering and re-engineering (c.f. [1–3]), and the problems faced by teams applying them. The amount of produced artifacts when using these tools, and (in many cases) the distributed nature of the teams, begs the question of how to adequately store, catalog, archive and share such artifacts. It becomes all too easy to lose track of existing versions, the relations between artifacts, and even the artifacts themselves.

The use of standard version control systems (such as Subversion (SVN)) has shown to be inadequate [4]. In fact, it is not our objective to have a system with version control capabilities, as delta updates. Instead, we aim towards a repository for a diversity of artifacts. By artifacts, we are referring to the inputs and outputs of a software (re)engineering process, but mostly models. Example of artifacts include different types of models, test cases, pattern catalogs, processes descriptions, software prototypes, meta-models, or database schemes.

Three main functionalities are considered relevant in this context: repository functionalities (archive, catalog, categorize, search, explore and share capabilities); social functionalities (research groups support, associating groups with artifacts); scientific publications support (management and association with scientific publications). We classify such platform as a collaborative Web repository. On the one hand, it allows multiple researchers to collaborate in a project through a Web environment. On the other hand, it provides archiving capabilities (i.e., a repository). We consider a Web portal to be the best solution to access this type of system. It ensures that the users will be able to access it from almost any device with a Web browser, without the need to install any software. Some Web 2.0 functionalities, such as dynamic content and user supported contents (i.e., forums), improve both the interaction of the users with the platform, and among them.

In this paper we present and discuss the implementation of the proposed platform. Section 2 reviews related work, with the analysis of a number of similar tools. Section 3 builds on that to present the requirements for the platform. In section 4 the tool is described. Finally, Section 5 presents some discussion about what has been achieved, and Section 6 concludes the paper with some pointers for further work.

2 Related work

A study was carried out to analyze the state of the art for collaborative repository tools.

It covered, not only software oriented repositories, but also other platforms, such as business process repositories and books cataloging systems. The analyzed repositories can be categorized into two main approaches. First, there are the data repositories, common among the database research communities. They are the extension of a database management system, with emphasis on metadata management. The repository consists in a "shared database of information about engineered artifacts produced and used by an enterprise" [5]. Model management systems are also related with data repositories, addressing problems of models representation and processing [6]. Second, there are the process model repositories, based in workflow and conceptual modeling. They provide a repository and execution environment for those models [7].

The analysis of the related work produced two major outcomes. First, it allowed evaluating how suitable for our purposes existing software systems were. Second, it provided valuable input regarding the requirements for this type of platform. A contribution of this work is the table presented in Section 5. It presents the comparison of the discussed platforms regarding their functionalities. This section presents the most relevant tools.

2.1 Repository for Model Driven Development (ReMoDD)

ReMoDD[1] is a Web platform developed by the Colorado State University Department of Computer Science and Engineering [8]. This platform aims to support the Model Driven Development (MDD) community by providing an easy and complete way to share models, informations, case studies and knowledge among multiple audiences, for instance teachers, researchers and students.

The tool provides a Web portal to interact with the repository. It supports browsing the repository by listing the models sorted by multiple attributes: name, description, categories, author(s) or update data. The only organization criteria is this sorting functionality. Other than that, it is only possible to open a model and view its full description (there is no search feature). By opening a model it is possible to visualize its details, post comments and download it. To perform further interactions user registration is required. This platform provides also a group (or forum) functionality, where registered users are able to interact.

As artifacts' discovery is relevant to us, the lack of search and list functionalities presents a big limitation. We consider that viewing the models' informations is also one of the most relevant functionalities, which is very limited in the ReMoDD, providing only general information and lacking the authoring tool, scope and version (among other informations). The group functionality is a forum-like functionality, but lacks a deep integration with the rest of the framework. Finally, at the moment, the platform is not accepting registrations.

2.2 ECOBAS

ECOBAS is an information system which supports online modeling and simulation. It is designed for ecology and environmental sciences [9]. This tool offers some interesting repository functionalities. It provides both a Web interface and local client. The Web interface allows users to search a model by name, by subject or by free-text. Viewing the models' informations is similar to other repositories. It is possible to select a model from a list, and its details are presented.

The focus of this platform on ecological and environmental context makes it unsuitable for our purposes. However, analyzing the tool made us aware of the importance of having an open platform. A flexible platform should provide support for a large variety of models, regardless of their application area. Another limitation of ECOBAS is the information shown about each model, which despite being detailed misses some relevant informations such as a visual representation.

2.3 Apromore

Apromore is a Business Process Model repository [7]. While it was possible to test a first version of the repository, that version has since then been deprecated and taken offline. A new version of the tool is under development but is currently unavailable to test. Hence the current analysis referes to the deprecated version.

[1] `http://www.cs.colostate.edu/remodd/v1/` (visited January 30, 2014).

Apromore provides model storage and management funcionalities (both view and create/edit). The models' discovery functionalities are adequate, as they support listing, searching and filtering of models (by criteria). All the models' details are available, and it supports rating the models. This tool provides an intuitive user interface for model management. However, groups are not supported, and all models exist at the same level, being available to all users (there is no visibility concept). This platform is closer to a repository than to a collaborative environment. Additionally, it support only the storage of models created directly in the platform. Hence, the tool is too restrictive to be considered a generic collaborative platform.

2.4 Shelfari

We consider model organization, storing and categorization as the core of the a model repository. Such functionalities are found in books managing systems, as is the case of Shelfari. This platform provides a digital library to store and organize books. Book entries can be searched, listed, added, removed and rated. Cataloging is done through several aspects, such as subject, author and tags. The concept of group is also present, where a set of users sharing the same interests about a particular subject can discuss it. While not directly usable for our needs, the tool provides useful hints for developing a new platform, as the task of cataloging artifacts shares some concepts with cataloging books.

2.5 Other tools

A number of other tools were identified and analyzed. Due to space constraints they are not discussed in-depth here. Briefly, from these tools, we may highlight the following ones.

- ATL Zoo[2], which presents a list of artifacts in a Web page, accessible also from the eclipse Integrated Development Environment (IDE).
- ARIS[3], an enterprise architecture management tool with repository functionalities.
- Adonis, a commercial platform focused in business process management [10].
- Colex, a model repository that targets model versioning and versioning conflicts [11].
- ModeleR, a model knowledge base for experts, targeting environmental model execution [12].

2.6 Discussion

None of the analyzed tools was found suitable for our purposes. Briefly, it is possible to say that the tools are either for a specific domain, for a specific

[2] `http://www.eclipse.org/atl/atlTransformations/` (visited February 26, 2014).
[3] `http://www.aris.com/` (visited January 31, 2014).

language, are closed (for registration), or are too limited in functionalities. A platform that seems promising is ReMoDD. This platform could fulfill our needs and solve our problems. However, a set of limitations (not the least of which is the fact that it is currently not accepting further registrations) made this platform inadequate for our objectives. Additionally, the platform lacks Web 2.0 functionalities to encourage collaboration between researchers [11].

While we found no suitable tool to cover our need for a collaborative artifacts repository, the analyzed tools were able to provide us with insights on what functionalities to implement to achieve an usable and adequate artifacts repository. This is discussed in the next section.

3 Requirements for a collaborative Web repository

As none of the analyzed tools is adequate for our purposes, we propose to create a new collaborative Web repository. Combining our need with the informations extracted from the tools' analysis allowed us to define a set of requirements to guide us in the development of a new tool. We present in this section these requirements, considered essential for our repository. Our objective goes towards the development of a Web platform supported (i.e., the artifacts are provided) by the community.

To start with, the platform will require what in [7] is designated as the standard repository functionalities, which include data storage, access control, and simple search queries. Those requirements are not enough when developing a new system, if we want it to be better than existing solutions. We decided to include some other functionalities, such as advanced search functionalities.

3.1 Artifacts repository

One of the main functionality that we look forward in a repository, is the artifact archiving and cataloging. Archiving artifacts will help keep track of them, store them in a centralized platform and share them with third persons. Cataloging the artifacts allows to store them in a meaningful way, and later to ease the process of finding them. The cataloging enables also the possibility of other people finding models. We consider that multiple approaches should be possible when browsing the models, namely textual search, criteria listing and criteria browsing. Also, multiple criteria for cataloging should exist in order to ease the browsing process.

Searching artifacts by text should support finding models either by name or description. This is the most direct way to perform searches, since textual forms are the most common way found nowadays. Criteria listing represents a search which allows the user to select from a set of predefined criteria. With this approach it is possible to filter the models to a subset containing only the relevant criteria selected by the user. Criteria browsing is a refined search approach, which allows searching the models by reducing the number of results

as we select criteria, related with previous selections. This approach will raise the probability of finding artifacts within the repository.

Models are prone to changes and updates, and such factor is essential when developing a repository. In order to support such behavior we propose supporting several versions of the same model, sorted in a meaningful way.

The decision of making an artifact public (accessible to everyone) or private is left to the user. Hence, the user might decide to kept an artifact private, for instance while in development, or only available to a subset of users. If an artifact is public, it should be accessible by anyone, allow to add comments, ratings, and even keep track of it. This is where the collaborative functionalities start, in the sense that other users may collaborate in the development or improvement of an artifact. If an artifact is private, only the author should be able to see and modify it. Lastly, in order to support collaboration, an artifact must be able to be restricted to a group.

3.2 Publications management

Developing tools and works in academic context results in a large set of scientific publications. The publications arise in several computer science areas, and sometimes they are related with artifacts. Hence, in this context is makes sense to manage references to scientific publications, associating them with the artifacts. As an artifact might also be referred in several articles, we propose a bidirectional relationship between artifacts and publications. With this functionality it should then be possible to search models related with specific publications, or otherwise, search publications related with specific models.

3.3 Social functionalities

It is common for the research process to involve interaction among several persons and ideas as well as previous works. The collaboration and sharing of information improves the research results. From multiple people, different approaches emerge and sometimes best results are found by combining several persons' ideas. This is the basis of the collaborative platforms [13].

An improvement on the repository would be to deeply integrate the social functionalities with the artifacts. The group concept, allied with the forum functionalities seems an appropriate requirement. By creating groups where the users could discuss ideas, and associate artifacts to them, it would allow a collaborative comportment.

In the same way that the models have a visibility option, it makes sense to have the same option for the groups. Hence, it should be possible to make a group (as well as its artifacts) restrict to a set of users. With this approach only the subset of persons related with the project would have access to the information. This is specially useful for private projects, projects in development, or simply by convenience. When an artifact is part of a group, it would be adequate to allow both the author and the members of the group to update it.

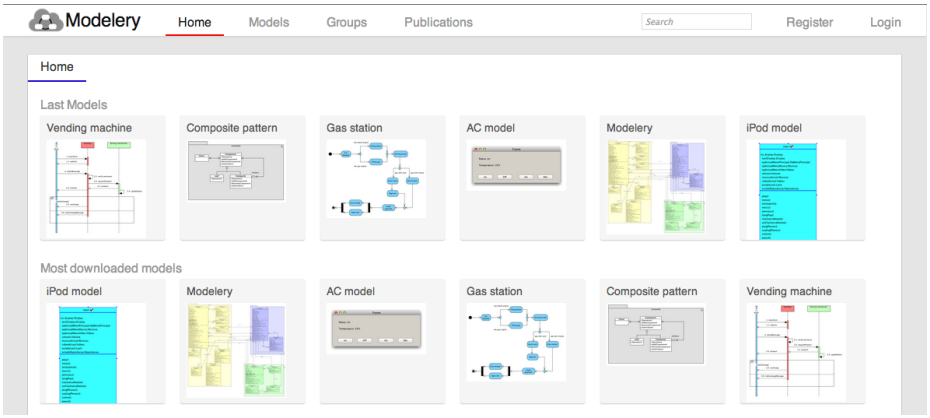

Fig. 1. Modelery main page.

3.4 Levels of sharing

Not all the artifacts and groups are developed for the same purpose. Some of them are intended to be public, other restricted to a subset of persons (and able to be updated by all these persons, or only the author) and other completely private. Also the groups may either be public or private (selecting the persons which should belong to them).

The distinction between all these visibility levels is crucial to cover a broader audience of developers. Also, an author might decide to keep a model private while developing it, and make it public once finished. It provides some more control over the development process.

It is easy to think in version control functionalities (e.g. for models) as adequate for such platform. However, at this point, such functionality will not be considered. Firstly, implementation of version control functionalities is known as a hard task [4]. Then, we produce models in many languages (some of them are not even standard), which results in known versioning problems [14]. By merging these two factors we face a complex problem that we decided not address at the moment. Furthermore we are more interested in cataloging artifacts (where the artifacts should be more stable and ready to be used by other users), than in a centralized development tool as is the case of control version systems.

4 The Modelery

In order to solve the inability of the analyzed tools to fulfill our needs we have developed the Models Refinery (Modelery)[4]. Our platform combines the proposed functionalities in a single Web environment, accessible through the browser as depicted in Figure 1. Here we present the decisions which lead to our tool, as

[4] `http://modelery.di.uminho.pt`

Item	Description
Name	The name of the artifact
Author	The author of the artifact, automatically associated
Date	Date of submission
Description	A description of the artifact
Institution	Institution where the artifact was produced
Tool	Tool which originated this artifact
Category	Category where the artifact belongs, i.e., the area of knowledge
Tags	A set of tags, associated with the artifact
Scope	The artifact scope, within the area of knowledge
Language	The language in which the artifact was created (for instance, programming language)
Publications	List of publications associated with the artifact
Visibility	Visibility of the artifact: Only to author, to group, or public
Updatable	Whom may update the artifact: only the author, or the group
Group	The group which the artifact may belong
Image	An image representing the artifact
File	The artifact file itself

Table 1. Meta-data.

well as the description of the functionalities. It was developed according with a model driven methodology, and used the Modelery itself to keep track of the source models.

4.1 Artifacts repository

The artifacts repository functionality was our major concern. We are interested in storing not only the artifacts, but also their meta-data. This meta-data constitutes the artifact's entry, provided by the user when submitting it to the repository, and it is essential information to provide when displaying an artifact. Table 1 summarizes an artifact's attributes. These are further discussed below. Figure 2 presents the corresponding Web page.

Collaborative functionalities are achieved by supporting interaction between the users, trough the artifacts in the platform. This interaction fosters the artifact's evolution, due to the users feedback. Indeed, registered users may interact with an artifact by adding comments (which may help the author or other users). Also, the users might rate it, expressing its satisfaction with the artifact, with a value from 1 to 5. The artifacts' author is able to both update the model (by submitting a new version - the previous version is kept on record), and to edit the artifacts' meta-data.

Our platform relies on artifacts created by the users. Hence, an artifact must always have an author. While any user might search and view (public) artifacts, registration is required in order to create a new one. When creating the artifact, the user should specify all the details, as well as group, publication and visibility options. The artifact file should be also specified, and it is then uploaded and stored online in the platform.

As the artifacts belong to a specific context, we provide two ways to specify it. First, we allow an artifact to be part of a group. This possibility enables us, not only to aggregate a set of artifacts in a specific group, allowing for their categorization, but also to restrict its access to a set of persons which may view or update it, the members of the group. Second, we provide also a means to

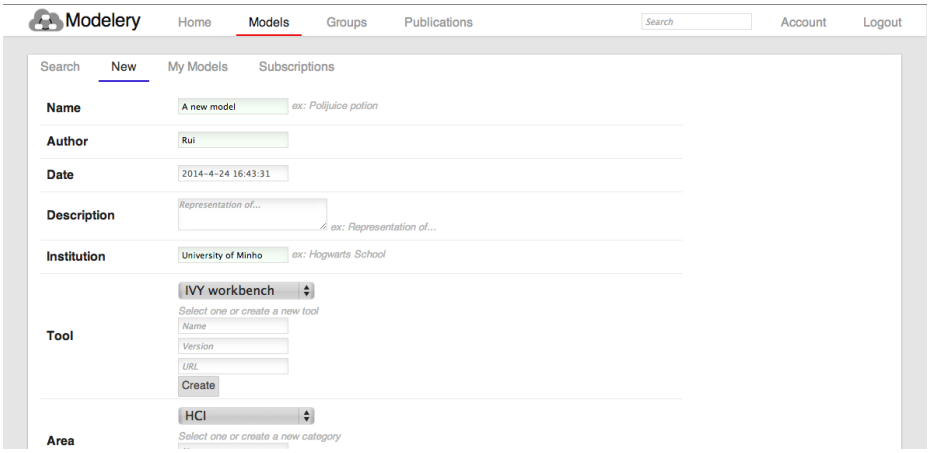

Fig. 2. Adding an artifact.

identify the publications (for instance articles, papers or posters) in which an artifact is be involved. This constitutes a further dimension though which to classify and access artifacts.

The artifacts have different purposes, and some of them (such as models) may belong to different development phases. As such, the artifacts' visibility level defines if it should be visible to everyone, visible to the group members, or visible only to the author. This enables users to keep artifacts private during their development phase (or permanently, of course), or opt to share them with a restricted group of people who might comment of actively collaborate on its development. The visibility level allows also to define which users might update the model. Here, the owner of an artifact may let a group update it, or restrict updates to himself/herself. The visibility level and who may update an artifact are independent properties, since it may be visible to the group, but only the author might have permission to update it.

The platform provides several ways to search and explore artifacts. By selecting the search option, a listing of the existent artifacts is presented, as depicted in Figure 3. We provide the possibility, also, for an author to view a list of his/her own artifacts. The user may then input some text, and the listing will start to be filtered, by presenting only the artifacts whose name or description match the text being input. This is the more natural search approach, common in most repositories.

Alternatively to the textual search, the user has the possibility to browse the artifacts. Browsing differs from searching by presenting the user with a set of predefined criteria: the tool in which the artifact was developed, the language in which it was written, its category and its author. With this approach it is possible for a user to select all the artifacts containing the specified properties. This supports a rigorous filtering of the artifacts, and allows also the filtering of artifacts by several criteria at the same time.

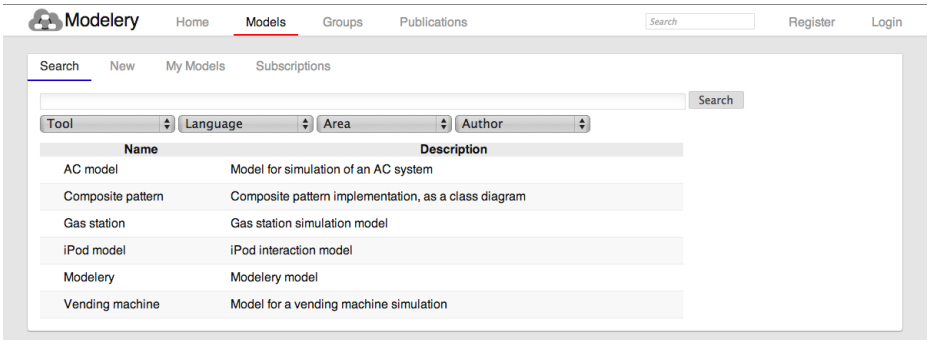

Fig. 3. Searching for an artifact.

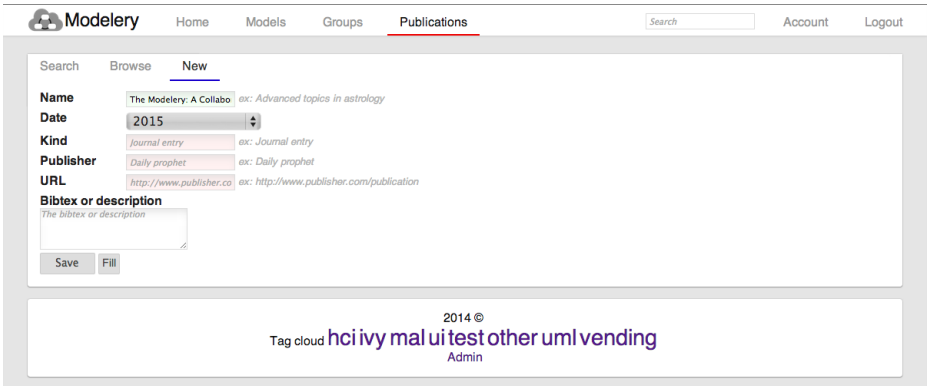

Fig. 4. Adding a publication.

A specific artifact (or set artifacts) might be of interest to a user. In order to ease the user access to those relevant artifacts, we provide the possibility to "track" an artifact. This means that a user may choose to follow the progress of a specific artifact, keeping a reference for it. It is also possible for a user to list the artifacts that he is following.

4.2 Publications management

As already mentioned, Modelery supports the possibility to create publication entries. The publications are registered with their name, abstract and URL for the article location, as shown in Figure 4. Contrary to what is provided for artifacts, publications management does not allow uploading the publication itself into the platform. We consider this to be a more efficient approach, as the platform's focus is not publications' management. Since publications may have more than one author, they are not automatically associated with the user which created them. Information of the authors is in the publication document itself.

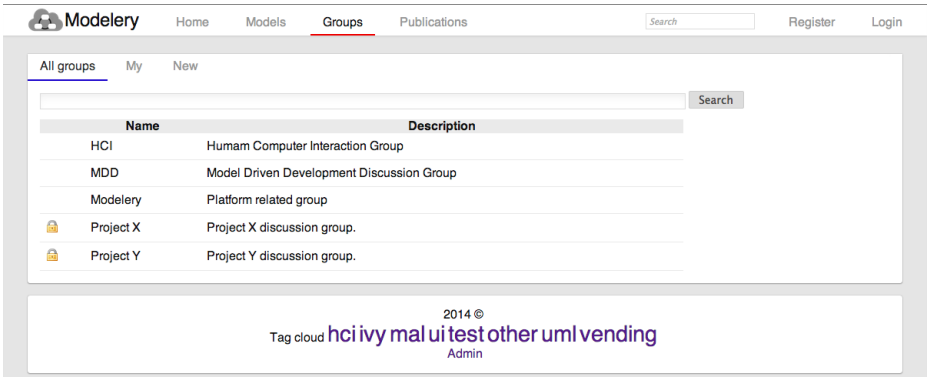

Fig. 5. Searching groups.

The relation between the artifacts and the publications can be explored starting from either entity. On the one hand, publications may refer a specific tool or artifacts, and it is possible to list the artifacts associated with a publication. On the other hand, an artifact may be referred in multiple publications, and it is possible to view all its associated publications. This functionality provides a convenient way to explore publications along with artifacts, and at the same time provides more information for a given artifact. It allows also exploration of the practical results (i.e. artifacts) of the publications.

As a large set of publications might be added, we provide also search functionalities for them. The textual search functionality is provided for publications. Along with textual search we provide the possibility to browse the publications, by their kind, date of publication or publisher.

4.3 Collaborative functionalities

Since we are developing a collaborative repository, Web 2.0 functionalities are essential to promote interaction among users [12]. Once registered, users are automatically associated with any group, artifact, comment or update that they create. This allows other users to know who is the author of a given artifact, or the owner of a specific group. Users are responsible for managing the groups that they creates, by selecting which other users should be part of the group. In Figure 5 is presented an overview of the groups search page. A functionality which is essential for promote collaborative behaviors is the possibility of users to exchange messages inside the platform. The Modelery supports both personal one-to-one messages, and more public messages in a discussion group (or forum).

Other functionalities include the dynamic main page, which presents information such as the last submitted artifacts and most downloaded artifacts, and a tag cloud. This provides an overview of the contents of the repository, emphasizing most relevant artifacts.

4.4 Usability improvements

Due to the relevance of usability considerations for the platform's success, an effort was made to create a responsive user interface (for instance avoiding to reload full pages for small requests) in order to improve the experience of the users. This was mainly achieved recurring to Ajax, by performing modular page loadings. This also enables us to provide more lightweight Web pages and reduced bandwidth usage. Resorting to a combination of HTML5[5], Cascading Style Sheets version 3 (CSS3) and jQuery, we are able to improve the user interface by, for instance, providing early error detections when filling field in the Web page, and better feedback (including animations when performing changes to the page contents).

Nevertheless, the Web interface was developed taking in mind compatibility with old browsers. Even if the visual aspect is not kept (mainly due to CSS3 compatibility), all the functionalities remain usable.

4.5 Implementation status

The Modelery was developed according to a multi-layer architecture, using a model driven approach. The presentation layer was implemented using Java Server Pages (JSP) and servlets over the business layer. Following a multi-layer approach allows us to easily improve or change some of the platform components. For instance, it would be simple to add Web-services over the business layer.

At the moment, the tool is fully functional. All the described functionalities, including access control are available. It is also possible to access public artifacts and information without registration.

5 Discussion

Regarding our initial goal of a platform to support the archiving and exchange of models and other type of artifacts, relevant to research into software engineering methods and tools, we have now achieved a fully functional prototype, which we consider implements the more relevant functionalities identified.

An alternative approach to achieve a similar platform would had been to conjugate several other platforms into a single environment, for instance a Concurrent Version System (CVS) (such as SVN or GIT) for artifacts management, along with an online forum (such as phpbb) for discussion issues. However, the approach taken presents advantages over the integration of multiple platforms. First, CVS system are mainly used and optimized for textual documents (such as source code). They lack model targeted functionalities, and it is harder to add functionalities (such as an online model editor) later on. Furthermore, CVS systems are not targeted for sharing and cataloging. Using an online forum for our objectives suffers from similar issues as the usage of a CVS for the models, with the inability to provide specific functionalities. Integrating visibility levels

[5] `http://www.w3.org/TR/html5/` (visited January 30, 2014).

in a CVS, or groups, managed by the users, in the forum, would have been a very hard and time consuming tasks. Combining these functionalities to collaborate together, by providing a platform as coherent and as practical as ours would have been more costly than developing this one. Finally, a poor integration of these technologies might easily lead to an unpractical platform, and result in a project failure.

Some of the repositories discussed in Section 2 offer online models' editing. That is an interesting functionality. However, not suitable for our repository at the moment. Since we allow any kind of artifact (therefore any models) in our the repository, it would require either a restriction on the type of supported artifacts (by imposing a metamodel, for instance), or selecting a subset of artifacts with online editing functionalities. We have chosen to ignore this functionality for now, since it would not lead to a solid and robust editor.

Comparing our platform against other repositories, it is possible to draw some conclusions. There are some similarities between our tool and ReMoDD, since our objectives are somehow similar. However, we provide some improvements over Modelery. First, our platform provides a larger group of functionalities without requiring registration. An unregistered user is free to explore all the public information, from groups to models and publications. ReMoDD is considerably more restricted in model browsing. The only way to search content in the site (any kind of content) is by textual search. Another possibility is to list all of the models. The platform provides also a forum, however completely disconnected from the models. Finally, it provides a workshop catalog system, once again, disconnected from the models. Viewing a model's information is very limited, since only few informations are displayed. ReMoDD claims to be a repository for model driven development, however our platform might provide a better support for model driven methodologies by overcoming some of ReMoDD shortcomings.

ECOBAS is targeted to different purposes, being aimed at a specific area and focusing on modelling and simulation. In what concerns management of models, ECOBAS is somewhat limited in terms of search functionality, since it only supports the listing of models by name, or performing a textual search. Opening a model's entry provides a large amount of information, but lacks some of the details we consider relevant, such as a visual representation of the model or the author. ECOBAS lacks also other functionalities such as publications management and discussion groups. From this point of view, Modelery provides a more complete environment as a model repository.

The Apromore platform shares some of our objectives, but is currently in a preliminary phase of development. The platform allows public models' submission only, limiting the models' scope. The model entries do not provide very complete information, since apart from its name, it is only possible to view their language, domain, ranking, version and author. The platform offers an interesting online model editor. However that editor is language specific, allowing only to edit one kind of model. Also, Apromore provides no other functionalities than a model repository. At the moment, this platform has limited browser support. Modelery provides a more usable option, since it is ready for use. Users are free

Tool	Fully Web	List	View	Comments	Download	Public access	Groups	Advanced search	Open platform	Software oriented
ReMoDD	✓	✓	□	✓	✓	×	□	□	×	✓
ECOBAS	×	✓	□	×	□	✓	×	✓	✓	×
Apromore (prev.)	✓	✓	✓	×	×	✓	×	✓	✓	×
Shelfari	✓	×	✓	✓	×	✓	✓	✓	✓	×
Modelery	✓	✓	✓	✓	✓	✓	✓	✓	□	✓

Table 2. Comparison of the analyzed repositories.

to register (contrary to Apromore), and submit any artifact (not only models), as well as their relevant informations.

Table 2 summarizes the comparison of the platforms.

6 Conclusions

In this paper we have described a collaborative repository for software artifacts, with special focus in models, patterns and catalogs. We presented the Modelery, a platform which combines an online artifact repository, publication management and social functionalities. The presented functionalities came mainly from our needs to store, manage, catalog and make the artifacts we produce during our research projects, available online. Also, with this platform we have created a new means to discuss the artifacts within discussion groups.

This platform corresponds to a prototype developed in order to fulfill our need for a collaborative model repository. The implemented functionalities represent a first approach, and as such there are many planned improvements as future work. Our mainly outreach with this platform is the academic community. This is due to our platform nature, developed within an academic and research context. In long term, we intent to outreach other areas, such as general research purposes (including research in business contexts), and even to support model driven development and artifacts sharing for enterprise contexts.

We have started using the repository for our own needs[6]. This has allowed us to test the repository and made possible minor adjustments. Our immediate next objective is to make it publicly available and encourage other research groups to adhere to it.

In the longer run. we take also in account the possibility to include other functionalities in the platform. Namely, the possibility of integrating editors or the generation of graphical representations for particular modelling languages, and also integration with verification and validation tools (e.g. for certification purposes). The integration with other tools can be achieved by means of Web

[6] `http://modelery.di.uminho.pt`

services. We will study the possibility to include such functionality, through Simple Object Access Protocol (SOAP) or REpresentational State Transfer (REST) technologies.

Acknowledgments

This work was carried out in the context of project LATiCES: Languages And Tools for Critical rEal-time Systems (Ref. NORTE-07-0124-FEDER-000062) is financed by the North Portugal Regional Operational Programme (ON.2 - O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

References

1. Couto, R., Ribeiro, A.N., Campos, J.C.: A Patterns Based Reverse Engineering Approach for Java Source Code. In: Software Engineering Workshop (SEW), 2012 35th Annual IEEE. (2012) 140–147
2. Campos, J., Saraiva, J., Silva, C., Silva, J.: GUIsurfer: A Reverse Engineering Framework for User Interface Software. In Telea, A., ed.: Reverse Engineering - Recent Advances and Applications. InTech (2012) 31–54
3. Campos, J.C., Harrison, M.D.: Interaction engineering using the IVY tool. In: ACM Symposium on Engineering Interactive Computing Systems (EICS 2009), New York, NY, USA, ACM (2009) 35–44
4. France, R., Bieman, J., Cheng, B.: CRI: Collaborative Project: Repository for Model Driven Development (ReMoDD). Colorado State University (2006)
5. Bernstein, P.A., Dayal, U.: An Overview of Repository Technology. In: Proceedings of the 20th International Conference on Very Large Data Bases. VLDB '94, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1994) 705–713
6. Dolk, D.R., Konsynski, B.R.: Knowledge Representation for Model Management Systems. Software Engineering, IEEE Transactions on **SE-10**(6) (nov. 1984) 619 –628
7. La Rosa, M., Reijers, H.A., van der Aalst, W.M.P., Dijkman, R.M., Mendling, J., Dumas, M., García-Bañuelos, L.: APROMORE: An advanced process model repository. Expert Syst. Appl. **38**(6) (June 2011) 7029–7040
8. France, R., Bieman, J., Cheng, B.H.C.: Repository for model driven development (ReMoDD). In: Proceedings of the 2006 international conference on Models in software engineering. MoDELS'06, Berlin, Heidelberg, Springer-Verlag (2006) 311–317
9. Cavalcanti, M.C., Mattoso, M., Campos, M.L., Llirbat, F., Simon, E.: Sharing scientific models in environmental applications. In: Proceedings of the 2002 ACM symposium on Applied computing. SAC '02, New York, NY, USA, ACM (2002) 453–457
10. Karagiannis, D., Kühn, H.: Metamodelling Platforms. In: Proceedings of the Third International Conference on E-Commerce and Web Technologies. EC-WEB '02, London, UK, UK, Springer-Verlag (2002) 182–

11. Brosch, P., Langer, P., Seidl, M., Wieland, K., Wimmer, M.: Colex: a web-based collaborative conflict lexicon. In: Proceedings of the 1st International Workshop on Model Comparison in Practice. IWMCP '10, New York, NY, USA, ACM (2010) 42–49
12. Pérez-Pérez, R., Benito, B.M., Bonet, F.J.: ModeleR: An enviromental model repository as knowledge base for experts. Expert Syst. Appl. **39**(9) (July 2012) 8396–8411
13. Wang, H., Johnson, A., Zhang, H., Liang, S.: Towards a collaborative modeling and simulation platform on the Internet. Adv. Eng. Inform. **24**(2) (April 2010) 208–218
14. France, R., Rumpe, B.: Model-driven Development of Complex Software: A Research Roadmap. In: 2007 Future of Software Engineering. FOSE '07, Washington, DC, USA, IEEE Computer Society (2007) 37–54