Project Awareness System – Improving Collaboration through Visibility

Daniel Kadenbach and Carsten Kleiner

University of Applied Sciences and Arts Hannover, Germany
{daniel.kadenbach,carsten.kleiner}@hs-hannover.de

Abstract. This paper proposes and describes the Project Awareness System (PAS) which is designed to improve the awareness of projects and project participants within and beyond organizational borders. The aim of this system is to increase the visibility of projects, so that users can easily find interesting ones and contact their participants to increase communication, collaboration and reuse of project results. The system enables an organizational unit to easily store project information at a central place. It does not impose strict rules regarding what data about projects can be stored. In this way strongly heterogeneous project environments can be mapped. The system offers its users extensive search mechanisms to find the projects they are looking for. The PAS additionally supports federation of multiple instances. The user can browse through the projects of multiple organizational units and organizations at one place, while each unit stays in full control of its data.

Keywords: CSCW, collaboration, project awareness, virtual communities, education.

1 Introduction

Projects and their results are the essence of our work. They help us to proceed, build upon established work, improve it and create something new. They serve as sources of inspiration. They help us to learn and to develop, to gather and share knowledge. In the best case, they create a durable value for the community. Projects can only achieve most of these benefits if they are visible, accessible and sustainable. With these attributes, the value of a project rises or falls. If it is not possible for others to become aware of a project, find it and access its contents or at least be able to contact its authors, it looses a great deal of its real potential.

For example, how do you find a master thesis, a scientific project or a capstone project in your own department? What if you only have a few keywords for your search? How can you find out if such or a similar project already exists at a different institution so you could share knowledge, build upon existing results or even work together? Only a fraction of active and finished projects have got a website which could be indexed by a search engine. So how are we able to avoid carrying out the same small projects again and again instead of working together on something big and new, if there is no reliable way to find similar projects?

A.A. Ozok and P. Zaphiris (Eds.): OCSC/HCII 2013, LNCS 8029, pp. 164–173, 2013.

Why do most projects start from scratch instead of building upon something which already exists or improving it? How much more could we accomplish if we knew what other institutions, staff members or students are working on, if we had the possibility to work together and learn from each other?

Our paper describes the so-called Project Awareness System (PAS) which we designed to answer these questions. The proposed system mainly focuses on academical and/or open-source environments for research or software engineering projects, but it may also be applicable to other environments and projects. Its primary aim is to make projects visible, accessible and sustainable to protect and enhance their value. The PAS therefore stores information about projects of an organizational unit at a central place and offers this information through well-defined interfaces. The PAS acts as an easy to use, uniform project portal, offering a project portfolio of the organizational unit independently of its internal project culture and infrastructure. It offers interfaces to store and access project information either manually or automatically. In its core we define a project description language to store the different project attributes. Using this representation, projects can be managed, assessed and found.

1.1 Motivation and Aims

There were three main motivations to create the PAS, partially contained in the aforementioned questions of the introduction:

1. Keeping Track of Your Own Projects. Over the years you will probably carry out a multitude of projects and it may be hard to keep track of them all. For example, just creating different folders in your file system to store your project information limits you in the way you sort your projects and therefore, in the way of finding one by special criteria. Usually, you will only be able to sort them by one criteria in different folders, e.g. their creation date. But you will have to search through all of them if you are searching a project which uses a special technology, or in which you worked with a specified person. Additionally no one else will be able to get to know what you have worked on.

2. Awareness for Projects of Your Organizational Unit. Apart from knowing and finding your own projects, there is probably a multitude of projects carried out at your organizational unit. Most of the time, apart from the project teams themselves and their "customers" there are not many persons who are aware of these projects, at least this has often been the case in our department. Projects may exist in quite different environments: Some may only exist on paper, some may use a Subversion or Git repository somewhere, some may use a project management service like Trac or Redmine and some projects may even be hosted on the Internet. Consequently it is impossible to be aware of many of them. But by not knowing which other projects exist or existed, great chances are lost. Being aware of the other projects may lead to more communication and collaboration. It may lead to more motivation, because projects are visible to others. It may also help to shape the spirit of the organizational unit, because everyone is able to see the achievements of her or his colleagues.

3. Finding Specific Projects in Your Organization and Beyond. Finally, the most important thing is finding projects. We cannot stress this point enough. There can hardly be any real progress if we are not able to easily find the projects of others. Every time someone starts a project, she or he has to look up if something similar has already been done or is being done at the moment. In both cases, synergetic effects could be utilized, efforts could be shared, help could be offered – but only if it is possible to find these projects. Again, some projects only exist on paper, some only on the computers of their participants, some are available on servers of their organizations and some are hosted at different Internet services[1]. How to keep track of all of them? At the moment our answer to that question is: You simply cannot. Even not in your own organization. You may conduct a search, but probably you will only find a fraction of the work, which has been done. And this has to be changed.

The aims of the PAS are therefore: Firstly, to be a central point of information for projects, offering a project portfolio of its organizational unit or organization and enabling its users to easily and extensively search for projects, to increase the visibility and sustainability of projects and to promote collaboration and reuse of results. Secondly, to be able to be used in a federation of systems, so that this single point of information may even exist for multiple organizational units, organizations or also at higher levels. And finally, to be able to be used with a minimum of effort and therefore being as simple as possible. This also means that everything has to be done to automatically keep the managed data up to date and valuable.

1.2 Related Work

Several authors have already done research in how to improve project awareness in project teams, like in [6] where dashboards and feeds are used or in [1], [2] and [3]. Our approach tries to increase the awareness also beyond project teams.

Majumdar and Krishna investigated in [4] how to utilize social computing implications for virtual teams, which is also a foundation for our work. Ohira et al. described a tool in [5] to improve knowledge collaboration on the base of project hosting sites, which is also an aim of us, while we use a different approach.

1.3 Structure

In Section 2, we are going to describe the architecture and functions of the PAS in more detail to show the challenges it faces and the benefits it may create. We will start by giving an overview of the system and its architecture. After that, we will look into the main features of the PAS – namely its data model, its interfaces and its federation support – and discuss aspects of their implementation. Finally, in Section 3, we will sum up our findings and critically review our proposed system: its weak points, potential and our future work.

[1] Services like e.g. Google Code, Sourceforge, GitHub, Gitorious, CodePlex, Alioth, GNU Savannah, Assembla and JavaForge just to mention a few of them.

2 Description of the PAS

In the following section we will describe the Project Awareness System. After discussing some general considerations, we will show an overview of the system architecture and explain the main ideas of the system in greater detail.

2.1 General Considerations

During the design of the system, we tried to follow two design goals: simplicity and usability. Like with many CSCW-systems, which are mainly used as a free choice by their users, the biggest challenge in implementing the PAS lies in the acceptance of its users. So from the beginning on, the system has to enthuse its users and give them a direct reward for the additional work they have just to use the system. Only in this way the accumulated data of projects can reach a critical mass at which it itself generates a surplus value for the users of the system. From this point on, the managed project data creates an appeal to use the PAS, as long as the quality of the data is high. We already identified two significant risks for the implementation: On the one hand, the system must not impose too much work for its users; in contrary, ideally users should not be forced to do any additional work at all. On the other hand, it has to be ensured that the managed data meets certain quality criteria. E.g. the data has to be accurate and up to date. Both points show how essential it is that project data can be managed by the system in an automated manner. Therefore it is crucial that the system comes with a well defined automation interface which can be used to automatically insert and update data.

To reach a high usability it is crucial to consider the most important factor of the system with much attention: humans. The user interface has to be simple, intuitively to learn and to use, and should return a direct surplus value to the user for every caused effort. The main user interaction, apart from creating and updating project data, will be the execution of searches. These searches must be very customizable to be able to search for projects through different criteria and considering that project data may be dynamic.

Finally, the system should not impose any constraints on the project data it manages. It has to be able to dynamically manage new data structures, because it would be impossible to describe all possible project attributes beforehand. Projects are too different by their definition. There may be useful templates for different project types (we specified a template for software engineering projects), but users of the PAS should not be limited by the use of such templates. Therefore users have to be able to add special data to their project descriptions. And users should not be forced by a template to provide data which does not make sense for their concrete projects. The system has to dynamically store and manage dynamically added data, making it as visible and easy to search for, as if it were a part of a well-defined template.

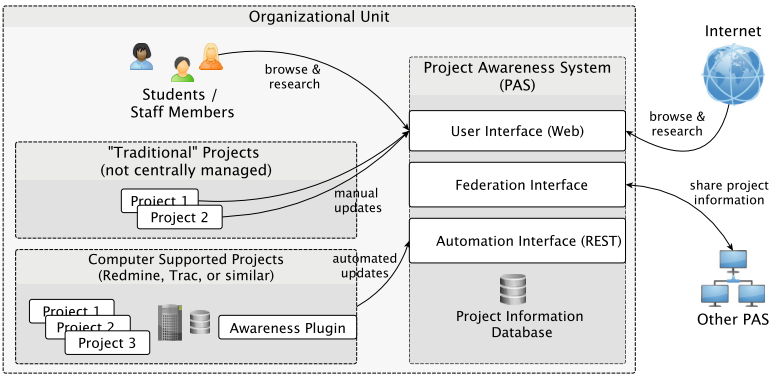

Fig. 1. PAS Architecture and Usage Overview

2.2 Overview of the PAS

The PAS can be seen as a specialized project database at a central well-known place in an organization, which offers easy to use interfaces. An overview of its architecture and usage is illustrated in Figure 1. As you can see, the system offers three different interfaces to realize its functions:

1. A user web-interface where human users have access to all the functions of the system, like searching for projects, creating new projects or updating them. The system can be accessed from inside of the organizational unit or, if desired, also from other networks or even the Internet. The web interface also acts as a project portfolio for the organization. Project teams which conduct traditional local or paper based projects can use this interface to publish and update their project data. In contrast to that, project teams which are already using CSCW support tools should use the automated approach, which is offered through the automation interface.

2. An automation interface, which enables automated clients to use the system. These clients can be used to automatically create and update project data from other project supporting systems. For example, a plugin for project management tools (like e.g. Redmine or Trac) can be written to automatically insert and update project data into the system at regular intervals, ensuring the actuality and correctness of the data while unburdening the users from all work. This interface can also be used to implement individual client software to further simplify the use of the system for special groups with common projects and to adapt even more to special requirements.

3. A federation interface which enables multiple PAS instances to be hierarchically tied together in a federation and synchronize their data. This makes it possible to offer a single point of information for multiple PAS instances while each of them retains its sovereignty.

We are currently in the process of developing a well-engineered default-implementation of the system after having finished a prototypical implementation as a proof-of-concept. The default implementation has to be ready

to be easily deployed at other institutes and therefore has to meet very high quality criteria. We use simple standard technologies and methods for the implementation. In this way the deployment, operation, maintenance, adaption and further development of the system is not burdened with complex technologies or requirements:

- The PAS follows the well known three-tier client-server architecture. The PAS server is implemented as a servlet, which uses a MySQL-database to store all project data. The servlet may run in a standard servlet container like Apache Tomcat; it does not need a full enterprise application server.
- We use HTTPS as a transfer protocol for all interfaces. This ensures that all data is transferred encrypted and that the source of information can be trusted.
- The web user interface is realized with the Google Web Toolkit, which allows to write simple Java code for the client side which is compiled into JavaScript. In this way, the potential of dynamic JavaScript at the client can be utilized while only having to deal with simple Java code in the development. Using this approach a far more sophisticated user-interface could be implemented.
- The automation and federation interfaces use REST over HTTPS for their communication, data is encoded to JSON. This makes it easy to write other software (for example individual clients or plugins to update data) which is able to communicate with these interfaces in many different programming languages.
- User authentication is done via LDAP, so that we could reuse existing user-databases of the organization. This may be a point which has to be adapted for the use in other organizational units, where LDAP is not a choice.

After this overview of the system architecture and its implementation we are now looking at the heart of the system: the data model which is mainly used to store the heterogeneous project meta data.

Data Model. To begin with, each PAS instance is attached to the domain of the organizational unit for which it is used. In this way, every project created in an instance will share its domain. As mentioned before the PAS currently does not have a user-database for authentication itself and uses an existing one via LDAP to reduce the administrative work and complexity of the application.

The most special requirement in handling the actual project data is the fact that from the point of the PAS it is unclear which data will later be used to describe a project. This is because of the diversity of projects and project environments (like supportive tools and systems). So the PAS maps project data into so called project attributes, which are named with a dot-notation to structure them. Table 1 shows a simplified list of attributes for the default project template which the PAS currently offers. This template is meant to only be a help, just very few attributes are mandatory (like project name, description and status) for a project, and further attributes can be added is necessary and can also be structured with the dot-notation. The challenge is to process this additional data seamlessly in the application and intuitively integrate it in the user interface, which we will investigate further in the next subsection.

These attributes are stored in the MySQL-database for each project, along with some additional information:

- The database also stores a list of each known attribute together with a description of it. This description is shown to the user when she or he selects the attribute and enters an attribute value. For dynamically created attributes the users are able to enter descriptions themselves.
- Additionally, each project attribute instance is saved with a modification time. This becomes important for the realization of the federation function, because with this information only attributes have to be synchronized between PAS instances which actually changed since the last synchronization.
- Finally, category trees are saved in the database. They make it possible to create hierarchical ordered categories which can be used to classify the project. The user interface is then able to show which project belongs to a category or its subcategories and lets the user browse through them. As the concrete categories strongly depend on the conducted project types, these trees will probably differ from organisation to organisation.

One thing which is currently not implemented, but could be sensible for the future, would be to make it possible to store images and files directly in the PAS. At the moment only links to images and files can be stored. But it would probably improve the user experience if users would be able to upload project related images, icons and files directly. In this way, project results could also be preserved for the future, e.g. even if the project homepage goes down one day. On the other hand, it has to be weighted out if this functionality does not make the system to complex and if it is the duty of the awareness system to host files.

Table 1. Overview of the Main Attributes of the Default Project Template

Attribute-Group	*Attribute*
general	name, shortName, description, shortDescription, homepage, logo, screenshots, maintainer, team, licence, creationDate, language, news, identifier, visibility, federation
classification	domain, organization, status, tags, categories, operatingSystem, programmingLanguage, technologies, predecessorProject, successorProjects, dependencies
social	ratings, reviews, views, likes, followers
tools	vcs, downloads, wiki, forum, faq, mailinglist, tracker

In the following we will shed light on some of the interesting aspects of the three interfaces of the PAS. All of them have to deal with dynamically created data, which is especially a challenge in the web user interface to nonetheless create an intuitive user experience.

Web User Interface. The web user interface is realized with GWT to create a dynamic and rich user interface with the flexibility client-side JavaScript offers. However, this flexibility is really needed to realize different functions, for example:

- When creating a project, users are able to select attributes from the default project template and from dynamic attributes which were used in other projects. To do this efficiently, the possible attributes are ordered in a tree, which maps the structure of the dot-notation of the attribute names. In this way, the users can easily and quickly navigate to attributes. Every time an attribute in this tree is selected, its description is shown and the user is able to enter a value for the attribute and save it. Additionally, the user can use a filter through a text box if she or he knows a part of the attribute name, or can even add a new attribute to the project, which is not in the tree.
- The search also benefits from the possibilities of dynamic web pages. There the user can dynamically add filters to create complex searches with just a few clicks and keywords. The user can choose between the attributes and combine them with operators (equal, not equal, greater, lower than) and values to easily create queries. For example, she or he could search a project which contains a special keyword somewhere in its attributes, has an active status, was updated not more than a few days ago, uses a special technology and has a higher rating level than a specified amount. This can only be realized in an intuitive way with a smart and dynamic web page.
- The user has to be supported whenever possible. Therefore the system suggest possible values in a list for an attribute whenever the user is about to enter an attribute value and also tries to complete the users input. To do this in a sensible way, the PAS looks up the frequency of attribute values in managed projects and if this frequency is higher than a certain threshold, it proposes the value. This is meant to get more sensible input, because sometimes the user may have the possibility to express an attribute value in different ways. But if she or he can choose a value from this list, it is more likely to be the same when it should be.
- The system also has support for several social functions to increase the quality of the managed data and to enable more search options for its users. It counts how often a project has been viewed by users; it lets users comment, rate and like projects; and it enables users to follow a project, so that they will be informed of changes of the project meta data.

Finally, data privacy may be a concern of the web interface. To protect email addresses, they are not shown in the interface. Only a form is shown, where a message can be left. A user can only be contacted through the system, when the sender has a validated email address. The recipient may also mark messages as spam whereby the administrator of the system is contacted who can take care of further actions.

Automation Interface. The automation interface is offered via REST and has largely the same functionality as the web user interface. Its main purpose is to be

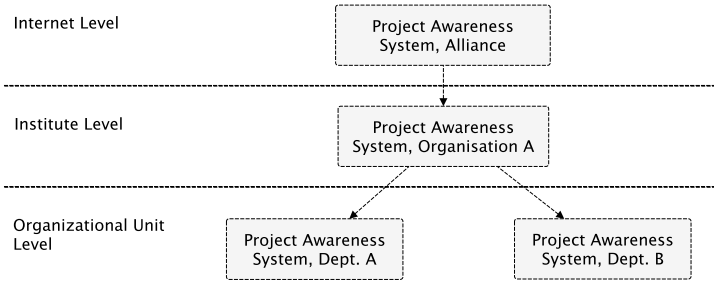

Fig. 2. Federation of Multiple PAS

used from plugins, which could be written for and installed into project supportive software like Trac, Redmine, Allura and others. These software tools already manage projects for their users and could easily and automatically update the project data at regular intervals to the PAS with such plugins.

Additionally, special clients for the PAS could be written using this interface to further adapt the system to specific organisational needs.

Federation Interface. The federation interface allows multiple PAS instances to hierarchically share their project information at different levels, which is illustrated in Figure 2. For example, a PAS at the institute level federated with the PAS instances of its organizational units would be able to list all projects from all its units, enabling its users to share a single point of information for all projects. Ideally, the vision is that these federations grow as much as possible, so that all projects of a kind are finally available at a single point.

The federation is realized through regular synchronisation of the project metadata databases. For example, if a PAS from a department A like in the figure is the child of the PAS from organisation A, then a federation user has to be configured in the department PAS. With this user, the organisation PAS regularly connects to the department PAS and synchronises a special part of its database with it. This data is used to offer information about the department projects. In the synchronisation process only projects which permit federation are processed. From these projects only attributes are transmitted which have changed since the last synchronisation to minimize bandwidth.

3 Conclusion

In this paper, we identified the need to be aware of other projects. We drafted the possible benefits from being able to be more aware of the projects which are carried out at your own organisation unit, organisation or even world wide. This is especially true if the projects are academically or open-source and therefore, want to create common values. Only if we have a reliable way to find these projects, we will really be able to build upon each others work and to realize even higher aims together.

To solve this problem, we introduced the PAS, which would enable organisations to collect their project metadata at a central place and even share them with others. It mainly depends on the acceptance of its users if the PAS is able to realize the aforementioned benefits. There still has to be done work to finish the default implementation and start to use the system. Many aspects of the system have to be further investigated and optimized using feedback from its users, so that everything is done to reach a high acceptance and to create a really useful and usable application. Usage studies have to be conducted with the first version. Other project supporting systems which offer portal functionality have to be investigated more closely, to further optimize the system and develop it into a form where it could become appealing for great accepted project supporting systems like GitHub, GoogleCode or SourceForge. Hopefully they will want to share their project metadata via a common interface, making them all together easily searchable instead of creating separated islands of information. And if that aim is finally reached someday, it probably will not matter if the used system is an successor of PAS or something else.

References

1. Anderson, K.M., Bouvin, N.O.: Supporting project awareness on the www with the iscent framework. SIGGROUP Bull. 21(3), 16–20 (2000), http://doi.acm.org/10.1145/605647.605650
2. Bharadwaj, V., Reddy, Y.V.R.: A framework to support collaboration in heterogeneous environments. SIGGROUP Bull. 24(3), 103–116 (2003), http://doi.acm.org/10.1145/1052829.1052852
3. Gutwin, C., Penner, R., Schneider, K.: Group awareness in distributed software development. In: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, CSCW 2004, pp. 72–81. ACM, New York (2004), http://doi.acm.org/10.1145/1031607.1031621
4. Majumdar, A., Krishna, S.: Social computing implications for technology usage and team interactions in virtual teams. In: 2011 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), pp. 443–450 (October 2011)
5. Ohira, M., Ohsugi, N., Ohoka, T., Matsumoto, K.: Accelerating cross-project knowledge collaboration using collaborative filtering and social networks. In: MSR 2005: Proceedings of the 2005 International Workshop on Mining Software Repositories, pp. 1–5. ACM, New York (2005)
6. Treude, C., Storey, M.A.: Awareness 2.0: Staying aware of projects, developers and tasks using dashboards and feeds. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, ICSE 2010, vol. 1, pp. 365–374. ACM, New York (2010), http://doi.acm.org/10.1145/1806799.1806854