

Scaki - The Scaffolding Wiki

Alejandro Fernandez
LIFIA - Universidad Nacional de La Plata
casco@lifa.info.unlp.edu.ar

Abstract

Support for documenting and managing work processes is valuable for any distributed software development team. Existing tools for process documentation and enactment are complex and expensive. Scaffolding is a simple approach to describe practices. WikiWikiWeb is a simple and inexpensive collaborative publication tool. This paper proposes extensions to exploit the simplicity of WikiWikiWeb to create and tailor scaffoldings of work processes that can be enacted.

1 Introduction

Many middle-sized, distributed software development projects (e.g., young open source projects) lack of an explicitly defined engineering process. Developers learn from the practice and adopt new ways of working. Documenting important decisions regarding processes, the use of tools, and required work artifacts, contributes to assure that what has been learned is not forgotten. Moreover, in a distributed team, changes in the ways of working need to be explicitly documented and communicated.

Repeatability is an important element for quality assurance. A documented work process can be repeated. Having a system that supports tracking the execution of projects according to the documented process helps assuring that the process is performed as documented.

In education, scaffolding [5] refers to the provision of a supporting framework, for example as step-by-step instructions that students can use to complete a complex assignment. Scaffolding support is gradually removed (or ignored by the students) as students gain practice. This paper proposes the use of a groupware system based on the idea of scaffoldings to document collaboratively the software development practices of a distributed team, and to help assure that these practices are followed in projects.

2 Scaffolding for Distributed Development

A scaffolding for distributed software development describes the process of software development in terms of its activities, tools, artifacts, and team member participation, and shows how these elements come together to form the process. A groupware system to support software development with scaffoldings should provide the tools to create, edit, publish, and use the scaffoldings.

The mechanism used to describe relations among process elements must be rich and simple. The amount of detail documented about the process should be enough to serve as a manual of operation.

When the scaffolding is used for a particular project, specific information about the project must be maintained to coordinate activities and to track that the scaffolding is followed. The system makes possible to apply the scaffolding in a given project. It maintains updated information about the state of the project and the allocation of persons to the different activities, and it provides access to the artifacts used in the project.

The system documents the planned process and tracks execution of process according to this plan; however, team members must stay in control. It must be possible to use artifacts, to use tools, and to perform activities, outside the planned schedule if participants consider it adequate. The system allows tailoring a scaffolding for its use in a particular project. Tailoring can be performed even after the project has started. If a tailored scaffolding demonstrates to be effective it must be possible to reuse it. It must be possible to extract a best practice scaffolding from finished or partially executed projects.

Most process documentation and tracking systems are complex and expensive. Simplicity and low cost need to be considered in the design of the proposed groupware system. The distributed team has access to the system which is deployable over the Internet. The system does not exclude existing tools and resources; it is possible to integrate references to external tools, and artifacts from external sources.

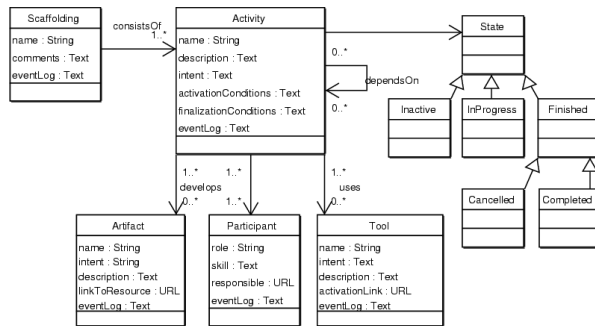


Figure 1. Object model of a scaffolding

3 Scaffolding Hypermedia

3.1 WikiWikiWeb: Simple yet Powerful

A WikiWikiWeb [3] is a server of hypermedia content rendered in HTML. A link labeled "edit this page" gives access, in the browser itself, to a form in which the source code of the page is presented and can be edited. A "save" button under the form submits the changed source code to the server, that in turn returns the newly rendered page. The source code of a page is plain text with some simple markup to do formatting and to create hyperlinks. Any group of run-together words (e.g., ScaffoldingWiki) defines a hyperlink to another page within the same WikiWikiWeb. The run-together words represent the title of the target page. If the target page does not yet exist the runtogether words are displayed followed by a question mark which is a hyperlink (e.g., ScaffoldingWiki?)¹. The user who first follows the hyperlink is given a form to edit the page. Saving the form creates the new page. Then, the runtogether words become the anchor of the hyperlink that conducts to the new page.

Any number of users can simultaneously create and modify pages in the WikiWikiWeb. The philosophy of a WikiWikiWeb is to trust users in that they will edit with responsibility. Pages are versioned on every save. All versions of a page can be retrieved and compared via a link in the page footer.

3.2 WikiWikiWeb for Scaffolding Hypermedia

The idea of WikiWikiWeb can be specialized into scaffolding hypermedia. Consider the UML class diagram from figure 1 as an object-oriented model for the information contained in a scaffolding hypermedia. The elements of the scaffolding, namely, activities, tools, artifacts, and participants map to pages (i.e., nodes) of the hypermedia.

¹In this paper, underlined text represents hyperlinks

Each page is identified with a unique title. For activities, tools, and artifacts the title corresponds to the name. For participants the title corresponds to the role.

Properties of type "Text" in the diagram contain WikiWikiWeb markup. Including a reference to another scaffolding element as part of the text in a property creates the relation between the objects. The nature of the relation is detailed in natural language in the text that makes the reference. For example, the text for the intent property of the activity named "DocumentRealEffortForBuild1.0" could read "To document the time that was spent for performing activity DevelopBuild1". If a text property makes a reference to a non-existing element, after saving the changes the user is given the possibility to choose the type of the new element from a list.

The relations in the UML diagram document the intended semantics of references. References from activities to activities indicate that there is some form of execution dependency between them. References from activities to tools document usage. References from activities to artifacts document that the artifact is developed or needed in the activity. References from activities to participants document participation.

The definition of a process starts with the creation of an empty scaffolding. A cover page introduces the scaffolding with a title and a description of purpose. Additionally, it provides an index with all the elements in the scaffolding classified by type. A special section provides references to the starting points. Immediately after creation, all fields in the cover page are empty. The user can edit the description of purpose and the title. More important, the user can introduce references to starting points. These can be either activities, tools, artifacts or participants. However, the user will usually reference the initial activities of the process. When the changes to the cover page are saved, the user is presented with a list to choose the adequate type of object for each of the named starting points. Choosing from the list creates the new element and opens it in edit mode. This results in the scaffolding being populated with the first group of elements.

Each element in the scaffolding is shown to the user following the structure presented in figure 2. The details of the element are presented within the placeholder labeled "type dependent content" in the figure. The edit button is used to change to "edit mode" which allows modifying the properties of the element. In edit mode, the place-holder contains text boxes to edit the properties of the element.

In presentation mode, anchors for hyperlinks that point to scaffolding elements are followed by a small icon indicating the type of the destination element. Activities, tools, artifacts, and participants that included references to the presented element are listed in individual sections at the bottom of the page.

Figure 2. A scaffolding's

A section titled "Event Log" at the end of the page is used to list important events that occurred with the element. The system adds entries to the event log automatically. Users can create entries via the input field located under the event log. They can use this mechanism to communicate with other users about important situations regarding the element, for example, to document daily progress regarding an activity.

The form used to edit an activity includes text fields to enter the activity's name, its intent, a detailed description, and a detail of the conditions required to start and to finish the activity. In addition, there is the possibility to indicate the status of the activity. The intent part of an activity provides a short (up to two paragraphs) overview of the goal of the activity. Together with the name, the intent is used to rapidly identify activities.

The description part of an activity provides all details needed by a newcomer to be able to perform it. If an artifact is needed because it provides relevant information for the activity, or if an artifact must be created, or updated, a reference to it is included in the description. The text containing the reference makes the relation to the artifact clear. The same is true for the usage of tools and for the requirements of participants.

The activation conditions part of an activity succinctly details all required aspects of the state of the project that are precondition for the activity to be started. These preconditions are usually expressed in relation to other activities (e.g., whether they were finished, or have produced significant results). The finalization conditions part of an activity documents the conditions that must be fulfilled to be able to say that the activity has been completed. For example, the finalization conditions part of "DevelopTheCho-

senStory" could state "this activity is completed when the behavior of the system documented in the chosen story card has been implemented, and the StyleChecking tool reports full compliance of the coding guidelines".

The forms used to edit tools, roles and artifacts provide text fields to edit the elements' properties. Additionally, for artifacts it is possible to provide a hyperlink reference to the real resource. An artifact element in a scaffolding acts as a placeholder for meta-information about the real artifact available in another location. This level of indirection allows integrating into the scaffolding resources that originate in other systems accessible through the network. In the simplest case, the real resource is an untyped page within the hypermedia. "Untyped" means that the page does not correspond to a type of element of the scaffolding, thus it is displayed and edited as a regular WikiWikiWeb page.

3.3 Enactment of a Scaffolding

Users browse the scaffolding and learn what needs to be done. At any time they can sign-up to be notified of changes in pages of their interest. Before the project starts, users usually sign-up to be notified of changes in those activities in which they participate.

To get an overview of the work that is to be done and to check what progress has been made, users activate from the cover page the graphical overview that opens in a separate window. The graphical scaffolding overview displays the elements in the scaffolding as a two dimensional graph view. Elements are nodes in the graph and relations are the edges of the graph. The name (the role, for the case of participants) of the scaffolding element is used to label the node. An icon that precedes the label of the node indicates the type of the element. Activities are displayed using a color coding that indicates the state. The graphical overview also serves as a navigation control. Clicking on a node in the view causes the browser that launched the view to navigate to the element represented by the clicked node.

The activities in the list of starting points provide a place to start looking for work. One user, declared as the project conductor, is responsible for changing the status of these activities to "in progress". Users who have signed-up to be notified of changes in these activities receive a notification. Other users who prefer a proactive way of working periodically check the state of the project to find out if their participation is required. The scaffolding hypermedia additionally provides a mechanism to query the list of pages that changed. This history of changes can be used by proactive users as a guide of where to explore for news.

When users have identified activities that require their participation, they navigate to the hypermedia nodes that document these activities to look for the details and to do their work. Once the finalization conditions of the activ-

ity are met, they edit the activity and change the status to "completed". The scaffolding hypermedia provides no support to compute dependencies among activities. It is up to the users to decide how changes in status of activities affect other activities.

As a result of some activities users need to modify artifacts. Artifacts can be retrieved via hyperlinks embedded in the description of the activities. The event log of the artifacts can be used to share notes regarding the changes made to the artifact.

If a tool is needed in order to contribute to an activity, the node describing it can be reached via the link embedded in the description of the activity. Moreover, if the tool can be started from the web, the node describing it provides the activation link.

3.4 Instantiation of a Scaffolding

When a scaffolding is used in a project, its pages are modified to contain project specific information. The status of the activities reflects the status of the project, the artifacts reflect the content of the artifacts in the project, and the event logs reflect the events that occurred in the project. A scaffolding can describe a way of working that is valid for more than one project. Therefore, it is desired that the experience it documents can be reused. To accomplish this, an unused version of the scaffolding needs to be maintained and duplicated every time a project based on it starts. This paper refers with the term "instantiation" to the creation of a copy of a master scaffolding for its usage in a particular project.

Instantiation of scaffoldings works by creating in the same server an exact copy of a master scaffolding. The copy is accessible in a new name-space identified by a URL composed of the URL of the server followed by an identifier for the new project (e.g., <http://scaffolding.myorg.org/new-project/>). All elements maintain their original name relative to the URL of the scaffolding. Artifacts from external sources which were stored as attachments in the master are also copied as attachments for the instance. However, artifacts or other on-line resources maintained outside the scaffolding (e.g., accessible via URLs) aren't copied. The instance and the master hold references to exactly the same resource.

On instantiation, an entry documenting the time of instantiation and a reference to the master scaffolding is automatically appended to an event log presented at the end of the cover page of the instance. The log entry additionally contains a hyperlink to an untyped page where further details about the operation are provided (e.g., the list of references to external resources shared with the master). Additionally, a log entry is appended to the event log in the cover page of the master indicating that a new project based

on the master has been created, with a hyperlink to it. The references to the instances from a master serve as examples of use.

Instantiation via deep copy, as done here, has the benefit of making the copy completely independent from the master. Everything can be tailored before and during use to match the needs of the project. To foster the propagation of successful experiences, a mechanism to extract a master from any tailored instance is required.

3.5 Extraction of Best Practices

Extracting the practice contained in a scaffolding used in a project implies creating a master scaffolding that mirrors its structure and content, and that can be used for further instantiation. The status of all activities in the master should be set to inactive, all event logs should be cleared, and all modified artifacts should be taken to an initial state.

The deep copy function used for instantiation supports the specification of replacement rules for element properties. For each property of type "Text" of each element type, the user can indicate a replacement value. A replacement value can be also indicated for the state of the activities. For untyped pages, the user can indicate whether to keep or to clear the structure and the content.

External resources pose a limitation when using copy as a mechanism for extracting practice. It is not possible to specify replacement rules to handle these elements. Resources from external sources stored as attachments are copied without any change. Artifacts or other on-line resources maintained outside aren't copied, but referenced.

After extraction, an entry documenting the operation is added to the event log of the new master scaffolding. The entry contains a hyperlink to the project scaffolding that served as the model. A page with additional information about the operation lists the resources that may require manual processing, and the resources that are shared with the instance.

4 Implementation Overview

Swiki [4], an implementation of WikiWikiWeb written in Smalltalk, has been chosen for the implementation of the system described here (which was named Scaki). Swiki, additionally includes markup elements to create heading, tables, and HTML links that point to anchors inside pages. Additionally, Swiki allows the definition of page templates. Each template defines how pages are seen in edit and in presentation mode. The richness of the editing and presentation facilities of Swiki simplified the creation of the edit and presentation page of the different types of scaffolding elements. Each of the types of elements in the scaffolding is defined as a template in the swiki.

The graphical overview was implemented as a Java Applet running on an independent browser window. The swiki server was extended (using the programming facilities provided by Swiki) to dynamically generate the web-page containing the applet. A HTTP request is sent from the applet to the server to retrieve a XML representation of the scaffolding. The XML contains information about the type and label for each of the elements in the graph, the relations among elements, and the URL of each of the elements.

5 Comparison to Related Work

Related work can be found in the area of workflow systems. Workflow systems aim at strictly defining work processes whose correct execution can be enforced and partly automated by a workflow engine. Most workflow management systems require expert knowledge to document processes, impose a rigid structure in how to describe and execute processes, are expensive and complex to deploy, and do not usually aim at providing detailed work guides.

Wang and Haake [6] used graphical two-dimensional hypermedia to model workflows. The result of their work is visible in the XCHIPS system. The XCHIPS system can be highlighted for its flexibility to change process definitions during execution. However, it does not provide mechanisms for extracting or cloning processes. Moreover, XCHIPS does not foresee the need of providing detailed descriptions of work.

Gruetzner et.al [2] documents an attempt to integrate the XCHIPS system with Spearmint. The spearmint System aims at providing detailed work guides accessible over the Internet. However, Spearmint has no provision for instantiation (support for tracking during work performance). The combination of both systems brings together the capability of Spearmint to document processes with the capability of XCHIPS to support execution. Processes can be tailored. Although this combined system is powerful it lacks of the possibility of extracting practice from tailored processes.

The Endeavors [1] system provides support for modeling and tailoring process definitions with the plus of allowing incremental adoption and deployment. However, it does not foresee the need of discussing and modeling the processes in collaboration. Moreover, Endeavors does not provide support for instantiation.

6 Usage Experience

Early prototypes of the system have been used to document the development process of a co-located development team at our institute. The scaffolding was created in an attempt to document the chosen internal engineering process by a team of managers. Each of the managers contributed

to the scaffolding in the spare time. They met regularly to review and discuss the changes. Managers found the capability to look for the latest changes useful for review meetings. In a later meeting with the developers of the system, the managers expressed they were happy about the simplicity of the tool. They regarded with special interest being able to navigate the resulting process as they were conducting the reviews. Nodes describing activities were completed in detail, whereas nodes describing tools were usually left incomplete. Managers indicated that they were not the adequate ones to describe how to use the tools.

Experiences regarding tailoring and enactment of the documented scaffolding are planned for the following months. Simulations with users from our team indicate that the system is usable and has the potential to play a central role as the mechanism to coordinate the work of the team. If the experiences with our co-located development team succeed, the system will be used to coordinate the cooperation projects between our institute and partner institutes in other countries.

7 Conclusions and Future Work

Existing tools for the creation, tailoring, and enactment of distributed development processes are complex, expensive, and do not provided the needed support. This paper discussed the requirements for such support, and described a groupware system which is simple to deploy and use.

Usage experiences have just started. As a next step, Scaki will be integrated as part of a system to support collaborative tailoring of groupware. In this context, Scaki will serve to document a method of collaborative tailoring, and to coordinate distributed team members in applying the method. Moreover, this project will serve to explore how Scaki affects tailoring of the work processes it models.

References

- [1] G. A. Bolcer and R. N. Taylor. Endeavors: a process system integration infrastructure. In *Proceedings of the 4th International Conference on the Software Process*, pages 76–89. IEEE Computer Society Press, 1996.
- [2] I. Ines Grützner, J. Münch, A. Fernandez, and B. Garzaldeen. Guided support for collaborative modelling, enactment and simulation of software development processes. In *Proceedings of ProSim'03*, 2003.
- [3] B. Leuf and W. Cunningham. *The Wiki Way*. Number 020171499X. Addison-Wesley Professional, 2001.
- [4] Swiki swiki. <http://minnow.cc.gatech.edu/swiki>, March 2004.
- [5] L. S. Vygotsky. *Thought and Language*. MIT Press, Cambridge, MA, 1986.
- [6] W. Wang and J. M. Haake. Tailoring groupware: The cooperative hypermedia approach. *Computer Supported Cooperative Work: the Journal of Collaborative Computing*, Volume 9(1):123–146, 2000.