

A DSL for Corporate Wiki Initialization: Demo Tool

Oscar Díaz and Gorka Puente

Onekin Research Group, University of the Basque Country, Spain,
{oscar.diaz, gorka.puente}@ehu.es

1 Introduction

Our premise is that, unlike other environments, corporations have all, personal organigrams, documentation practices and task schedules that frame both wiki users and wiki editing. The term “**wiki scaffolding**” is introduced to denote a wiki installation where some categories, templates, permissions, etc are initialized at the outset to mimic the corporate background. This includes structural concerns, communication means, permission needs, etc. So far, this background is patiently replicated by wiki users. Wiki scaffolding implies not only being knowledgeable about the wiki engine but also installing third-party extensions. This can certainly discourage users. Lawyer, architects, medical doctors are all profiting from wikis. Hence, *our aim is for wiki scaffolding to be made accessible to non technical people (**who**) that collaboratively agree (**how**) on a blueprint for the wiki (**what**)*. Thus, we propose the use of a *Domain Specific Language* (DSL). Additionally, collaboration and easy sharing can be promoted by using a *graphical DSL*. Specifically, the collaborative mandate suggests capitalizing on existing tools for supporting brainstorming. A common way of recording and expressing brainstorming sessions are *mind maps*. Now, this central topic is “wiki scaffolding”, and mindmaps constructs are reinterpreted to denote scaffolding concerns. This demo tool paper presents the *Wiki Scaffolding Language (WSL)* (pronounced “whistle”). *WSL* is built on top of *FreeMind*, a popular, open source tool to create mindmaps. Hence, *WSL* expressions are mindmaps. Our bet is that users might already been exposed to mindmaps and even to *FreeMind*, hence reducing the learning curve for *WSL*. These maps (i.e., *WSL* expressions) are then compiled into a set of *MediaWiki* directives whose execution generates the wiki scaffold.

2 WSL Example

Consider the use of wikis to support software projects. Fig. 1 provides an example for the *Purchase Project* as a *WSL* mindmap. In our example, the *Organigram* branch captures the existing roles as well as the employees assigned to these roles. The *Restriction* branch lists limitations in terms of wiki operations.

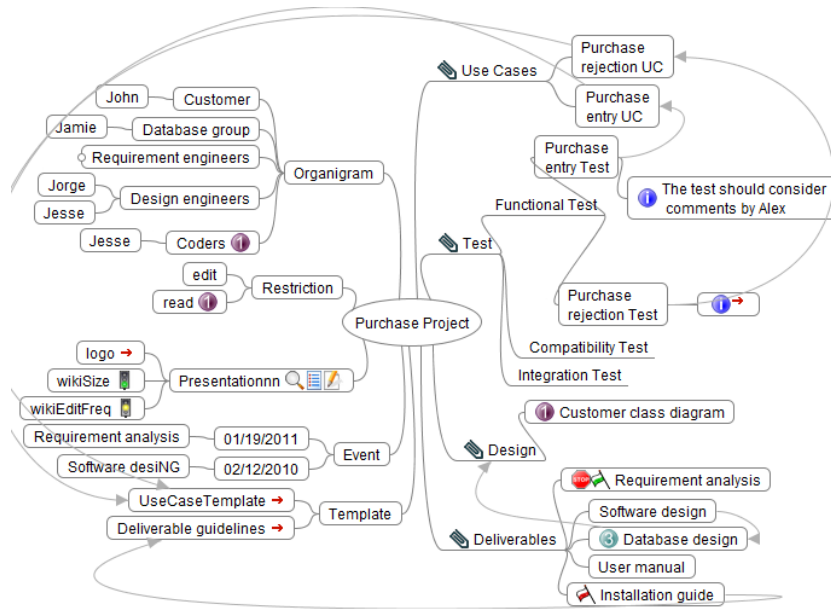


Fig. 1. Purchase Project Scaffolding.

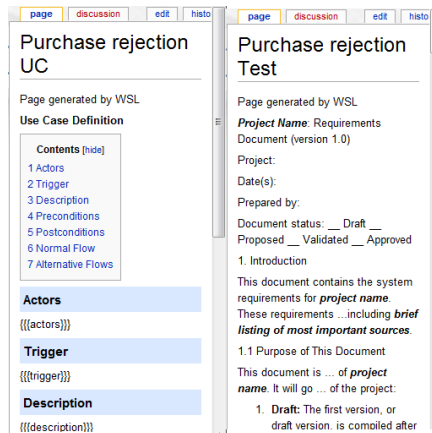







Fig. 2. Generated template and article.

The *Event* branch captures two milestones attached to pages “*Requirement analysis*” and “*Software desiNG*” at the onset. Next, the company already has guidelines to capture use cases and document deliverables. The *Template* branch refers to two such guidelines through the “*UseCaseTemplate*” and the “*DeliverableGuidelines*” nodes. The *Presentation* branch will impact on the rendering of the wiki based on the expected “*wikiSize*” and “*wikiEditingFreq*”. A “*traffic light*” icon  is used to indicate the three possible values of these properties: large, medium and small. As for the sidebar, this node includes a navigation pane (“*list*” icon ) and a search pane (“*magnifier*” icon ). It is completed with an index pane (“*look here*” icon ) on categories “*Use Cases*”, “*Test*”, etc). Regarding to restrictions, “*priority*”  icon sets a restriction whereby “*Coders*” are restricted from *reading* the article “*Customer class diagram*”. Common terms already in use include “*Use Cases*”, “*Functional Test*”, etc.

These terms find their way as wiki categories. Hierarchical relationships are captured by describing a node as a child of the parent category (e.g., “*Test*” ← “*Functional Test*”). Wiki articles are denoted as *bubble* nodes (e.g., “*Requirements analysis*” is an article categorised as “*Deliverable*”). The title of a node is its identifier, so that two *FreeMind* nodes placed differently but with the same title, stand for the same notion. The node “*Software design*” yields a wiki article with the namesake title. Some relationships are known at the outset e.g., trace requirements made advisable to keep a hyperlink between the “*Purchase entry test*” and the “*Purchase entry UC*” (i.e., an arrow between the node counterparts). Article content is realized as a child of the given article (with the “*info*” icon ⓘ). Fig. 1 illustrates the two options. The content of “*Purchase entry test*” is provided as the text of its child node. By contrast, the content of “*Purchase rejection test*” is available at the company as a *Word* document. *FreeMind* permits to introduce hyperlinks as node content (denoted through a small red arrow). This facility is used to link “*Purchase rejection test*” to the external document. Likewise, corporate guidelines can find their way as wiki templates. *WSL* only supports *Word* documents (exported as *XML*). These external documents are turned into either, article content or wiki templates. Fig. 3 shows the main page as generated by *WSL*.

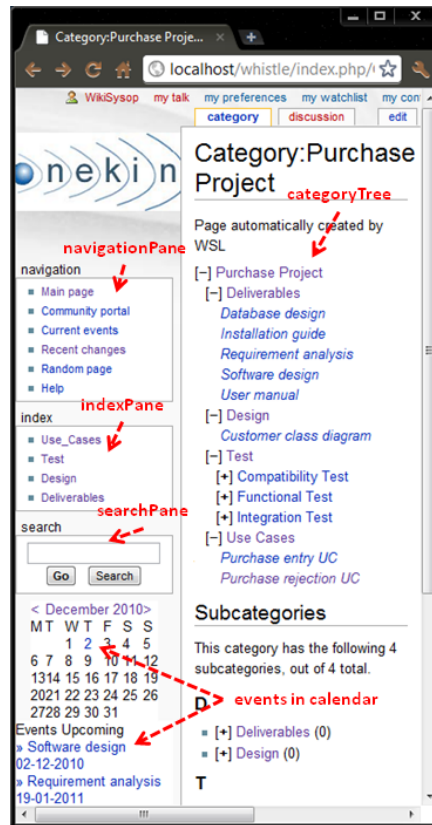


Fig. 3. Generated “*Purchase Project*”.

3 Verification of WSL Maps

Fig. 4 provides a snapshot of the “*Tools*” menu now extended to address *WSL* maps: “*WSL configuration*”; “*WSL deployment*” causes the generation of the wiki instance from the *WSL* specification; “*WSL Skeleton*” provides a *FreeMind* map with the basic *WSL* nodes; and finally, “*WSL Map Checking*” triggers *WSL* map verification. Fig. 4 depicts the verification outcome for our sample. A warning informs about the lack of the optional *Presentation* node that is due to a misspelling (“*Presentationnn*”). As for errors, they prevent the wiki from being generated. For our sample, these errors include: a misspelling of an event date

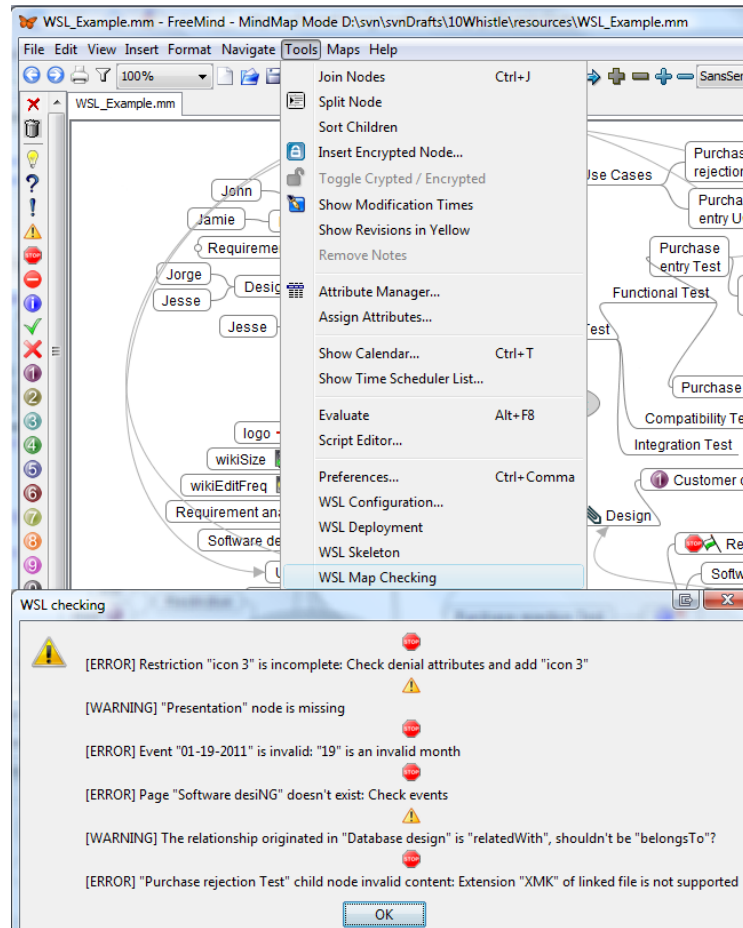


Fig. 4. Verifying the WSL map at fig. 1.

(e.g., “01/19/2011”); referring to a non-existent node (e.g., “Software desiNG”); partial definition of a restriction, etc.

4 Enactment of WSL Maps

Figs. 3 and 2 provides three screenshots of the generated pages: the main page (illustrating the *CategoryTree* and *Calendar* extensions), the “Purchase rejection Test” article (which is obtained from a *Word* document) and the “Purchase Rejection UC” (follows the “UseCaseTemplate” also externally obtained). Some features require additional extensions. The WSL engine builds upon *MediaWiki* and the extensions have been tested against it. Such composition is provided as a unit by WSL. Check www.onekin.org/wsl (examples and full paper).