Information Systems 37 (2012) 737–752

Contents lists available at SciVerse ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Wiki Scaffolding: Aligning wikis with the corporate strategy

Oscar Díaz *, Gorka Puente **

ONEKIN Research Group, University of the Basque Country (UPV/EHU), Paseo Manuel Lardizabal 1, 20018 San Sebastián, Spain

ARTICLE INFO

Available online 17 May 2012

Keywords:
Wiki
Wiki management
DSL
Mind map
FreeMind

ABSTRACT

Wikis are main exponents of collaborative development by user communities. This community may be created around the wiki itself (e.g., community of contributors in *Wikipedia*) or already exist (e.g., company employees in corporate wikis). In the latter case, the wiki is not created in a vacuum but as part of the information ecosystem of the hosting organization. As any other Information System resource, wiki success highly depends on the interplay of technology, work practice and the organization. Thus, wiki contributions should be framed along the concerns already in use in the hosting organization in terms of glossaries, schedules, policies, organigrams and the like. The question is then, *how can corporate strategies permeate wiki construction while preserving wiki openness and accessibility*? We advocate for the use of "Wiki Scaffoldings", i.e., a wiki installation that is provided at the onset to mimic these corporate concerns: categories, users, templates, articles initialized with boilerplate text, are all introduced in the wiki before any contribution is made. To retain wikis' friendliness and engage layman participation, we propose *scaffoldings* to be described as mind maps. Mind maps are next "exported" as wiki installations. We show the feasibility of the approach introducing a *Wiki Scaffolding Language (WSL)*. WSL is realized as a plugin for *FreeMind*, a popular tool for mind mapping. Finally, we validate the expressiveness of *WSL* in four case studies. *WSL* is available for download.

1. Introduction

Companies are increasingly realizing the benefits of wikis [1]. Indeed, the Intranet 2.0 Global Survey reports that around 61% of the respondent companies (1,401 participants) were somehow using wikis [2]. As any other Information System, the interplay of technology, work practice, and organization is paramount to achieve successful wiki deployments. Therefore, we can expect differences in wikis depending on the hosting organization, let this be an open community (e.g., *Wikipedia*), a learning organization [3] or a company [4]. The peculiarities of each organization will certainly percolate the wiki. Documentation, organigrams, project milestones are all there by the time the wiki is created. This contrasts with open wikis (e.g., *Wikipedia*) where the community did not exist prior to the wiki. As a result, corporate wikis (i.e., wikis host by an existing organization) might be tuned at the onset to the already existing information ecosystem. This is the assumption behind "Wiki Scaffolding". Next paragraphs introduce the "what", the "why" and the "how" of this term.

What: "Wiki Scaffolding" stands for a wiki installation (a.k.a. a wiki project) that is available from the wiki's onset, before any contribution is made. Such installation mirrors the practices of the hosting organization. Some examples follow: (1) company schedulings might impact the pace at which wiki articles are provided (e.g., deadlines, project milestones); (2) products, services, customers or established terminology within an organization might become categories to classify wiki articles; (3)

* Corresponding author.
** Principal corresponding author. Tel.: +34 943 018714.
E-mail addresses: oscar.diaz@ehu.es (O. Díaz), gorka.puente@ehu.es (G. Puente).

0306-4379/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2012.05.002

employees eligible to contribute, and their access control permissions, might be based on the company's organigram. A "Wiki Scaffolding" captures this setting as a wiki installation where the basic wiki configuration might be extended (through plugins) based on the selected scaffolding features (e.g., a plugin for events and calendars).

Why: The fact that wikis facilitate knowledge creation does not imply that such knowledge comes out of the blue. Both, the paralysis of facing an empty article and the lack of a holistic view of the wiki content, might prevent grassroot initiatives from "getting off" the ground. At this respect, scaffolding brings three main benefits:

1. Scaffolding facilitates wikis to be better aligned with the organization strategy. Wikis are frequently a bottom-up phenomenon whereby the wiki is introduced by an individual employee or a small group within the organization without the support of management. This approach may be useful to uncover hidden knowledge or hidden ways-of-working in a dynamic and unplanned way. However, it might fail in having a strategic intent. A lack of strategy might result in no clear guidelines about what, how and who should contribute. If so, "Wiki Scaffolding" forces to think about these concerns right from the beginning.
2. Scaffolding promotes user engagement. In a corporate setting, a wiki article might require some permissions, be subject to a deadline, belong to some wiki categories, or follow a given template. All these aspects might not be directly related with the article's content as such, yet they frame the contribution. Setting this frame is cumbersome and delays users in putting their wheels in motion (e.g., start to edit the article). "Wiki Scaffolding" permits this frame to be available by the time contributors start their articles.
3. Scaffolding as a wiki map. The "rules of practice" that govern a site (i.e., roles, access rights, templates, etc.) should be easily accessible to newcomers. So far, this information is scattered around the wiki, and frequently hidden in administrative pages. At best, a *README* page can provide some textual description of these practices. "Wiki Scaffolding" can play the role of an initial "practice sitemap". Newcomers can consult the scaffolding to have an eye-bird view of the rules that govern the wiki's operation.

How: "Wiki Scaffolding" faces two main obstacles. First, it implies an upfront investment before any content is provided. Second, it requires knowledge about the wiki engine (e.g., *MediaWiki*) and third-party extensions, both outside the competences of the layman. This will make "Wiki Scaffolding" yet another burden for the organization's IT department since most users will lack the required skills. Akin to the wiki spirit, the scaffolding should be managed by the users on their own. Therefore, both cost-effectiveness and end-user affordability are main prerequisites for scaffolding to be adopted. This advocates for the use of Domain-Specific Languages (DSLs) [5]. Furthermore, collaboration and easy sharing can be promoted by using graphical DSLs (as opposed to textual DSLs). Mind maps are popular diagrams that capture ideas around a central topic [6]. We capitalize from this popularity, and introduce a DSL described as a mind map to both capture and enact "Wiki Scaffoldings".

In short, this paper addresses the following research question: *how can corporate strategies permeate wiki construction while preserving wiki openness and accessibility?* To this end, we introduce the notion of "Wiki Scaffolding", and advocate for the use of DSLs as the engineer means. Specifically, we introduce the *Wiki Scaffolding Language (WSL)* (pronounced "whistle"). *WSL* is built on top of *FreeMind* [7], a popular, open source tool to create mind maps. You create your scaffolding by drawing mind maps. Next, you can "export" your mindmap as a "Wiki Scaffolding": a new wiki is created along the lines of the directives of the scaffolding (see a video of WSL at work at http://vimeo.com/31548363). The source code, examples and installation instructions can be found at http://www.onekin.org/wsl. Alternatively, WSL source code is also available in the official *FreeMind* repository http://bit.ly/xsA040.

This paper is organized along the design and use of *WSL*: *WSL* analysis (Section 2), *WSL* design (Section 3), *WSL* usage (Section 4) and *WSL* at work (Section 5). Conclusions end the paper.

2. WSL analysis

It is important to note that a scaffolding is "piece of code", i.e., a wiki installation. "Pieces of code" that support scaffolding for different companies would be different, yet they share a family likeness. That is, they belong to the same domain: "Wiki Scaffolding". This section identifies the scope and main abstractions behind this domain. The aim is to capture the company's work practice and settings, as long as they impact wiki operations. A main outcome of this analysis is a *feature diagram* that describes the domain concepts and their interdependencies [8]. A feature is a prominent and distinctive user visible characteristic of a system. In classical conceptual modelling, concepts are described by listing their features (attributes), which differentiate instances of a concept. In software engineering, software features differentiate software systems. Hence, the first question is what makes a scaffolding different from other scaffolding. This entails to assess the extent to which the wiki community suffers the traditional approach, and determine which would be the corporate aspects that, if available at the wiki onset, would have made a change. This is the topic of the next subsection.

2.1. The need for Wiki Scaffolding

To the best of our knowledge, the notion of scaffolding for wikis is rather new. We firstly need to collect evidences that suggest the necessity of scaffolding, even if they do not term it that way. To this end, we conducted a literature review on wiki usage in organizations. Next, we provide those seven cases that more clearly seems to suggest the need for scaffolding in wikis. The aim is to

provide vivid examples outside our own experience, which sustain this work.

Using Wiki Technology to Support Student Engagement: Lessons from the Trenches [9]. This paper reports on a failed experiment to use wiki technology to support student engagement. Thirty-seven percent of the students cited difficulties with the use of the technology. Authors conclude that *had greater instructional scaffolding be provided, in the form of lab-based exercises and the creation of an accompanying instruction handout, then maybe some of those students that experienced technical difficulties, or self-confidence issues, would have posted to the class Wiki*. "Wiki Scaffolding" can help to readily provide (1) wiki templates that guide and advice student contributions, or (2) wiki categories along the terminology set at the classroom.

Designing Knowledge Management Systems for Teaching and Learning with Wiki Technology [10]. This case study reports on the use of wikis to support collaborative activities in a knowledge management class at a graduate-level information systems course. The authors indicate that *Wiki technology can be used as a collaborative learning technology, but a lot of design needs to be done before bringing it into the classroom*. The paper indicates that *the initial findings suggest that effective... use of a wiki… is contingent upon familiarity of both students and instructors with the technology, level of planning involved prior to system implementation and use in class*. This ending is particularly insightful for our purpose: the need of planning prior to system implementation is regarded as a success criterion. This is what scaffolding is for.

Using Wiki to Support Constructivist Learning: A Case Study in University Education Settings [11]. Here, the aim is threefold: the assessment of learning, the monitoring of student participation, and the need for communication support in the learning process using wikis. For the purpose of our work, we notice the importance given to communication and how basic wiki mechanisms seem to fall short: *communication problems seemed to be a hinder to the writing of the wiki* while *groups which communicated more actively achieved better results, both in terms of quantity and quality*. The authors finally resorted to creating *an external discussion forum and encouraged students to use it to discuss and coordinate the development of the wiki*. This seems to suggest that communication design should be included as part of the wiki scaffolding. In addition, it is reported that at times it was difficult to know who was supposed to do what. Some anxiety about the end result was also a concern for many students. Both remarks hint to the need of an existing context where to frame the contributions.

Did You Put It on the Wiki? Information Sharing through Wikis in Interdisciplinary Design Collaboration [12]. This paper explores the use of wikis in software development projects. The author states that *the project wiki was created by the project manager a few weeks after the project started. At the beginning of the project, the project manager created a project definition page*, *which contained important information about the project such as goal*, *project team members*, *stakeholders*, *project description*, *success criteria*, *high-level schedule*, *deliverables*, *and communication plan. The document was reviewed and accepted by all team members*. This suggests the collaborative production of a blueprint for the wiki, but in this case, this blueprint has to be manually turned into a wiki installation.

A Wiki Instance in the Enterprise: Opportunities, Concerns and Reality [13]. This work reports on *ResearchWiki*, a wiki that supports yearly planning work by members of a globally distributed, research organization. The authors point out that users *preferred to use their project-specific repositories for recording progress in their projects rather than using the ResearchWiki. In many cases these repositories pre-dated the ResearchWiki and had evolved to support the operational needs of particular projects. This included access control as in many cases their project partners were from outside the research division and had not been given access to the ResearchWiki*. This highlights the role of the wiki as part of the information ecosystem, and the fact that companies tend to have stringent access control policies.

Enterprise Wikis—Types of Use, Benefits and Obstacles: A Multiple-Case Study [14]. The study highlights a main factor for wiki success: *a sufficient number of wiki-articles must exist right from start. Only then will employees perceive and accept the wiki as a useful knowledge base*. This suggests the role of scaffolding as a way to engage users. In addition, *first wiki properties and wiki structures had been eagerly discussed within internal group meetings, but no strict definitions arose*. This hints the notion of blueprint. Finally, *the 'built-in' simplicity of the wiki-software is rather a minimum requirement than a success factor*. Besides content editing, simplicity should also be sought in setting up an environment that helps in matters other than editing (e.g., category setting or permission restrictions).

Planning for a Successful Corporate Wiki [15]. Based on 30 case studies, this work aims to identify the key factors that affect the success of a corporate wiki. This analysis considers both technological and cultural aspects of wiki adoption. As for the purpose of this work, the following success factors are identified: *Bottom-up knowledge sharing culture* (scaffolding might help to involve different stakeholders at the very beginning on the search for a balance between free, bottom-up participation and the alignment with the corporate strategies), *content structure to avoid difficulties during navigation and information retrieval* (create a basic initial structure may help users to avoid the empty-wiki syndrome, and provide an early, global view of the wiki goals), *mechanisms to inform users of changes* (e.g., *RSS* feeds or email might complement wiki offerings) and *pre-populating with existing content*. The latter highlights the importance of providing direction settings [16]. When different people participate, it might not be clear who has to do what. Highlighting specific tasks at the onset in terms of articles to be written (e.g., related to hallmarks already scheduled) might help to spur people to start contributing.

2.2. Setting the features

Previous subsection provides empirical evidence for scaffolding. These insights are now made precise in terms

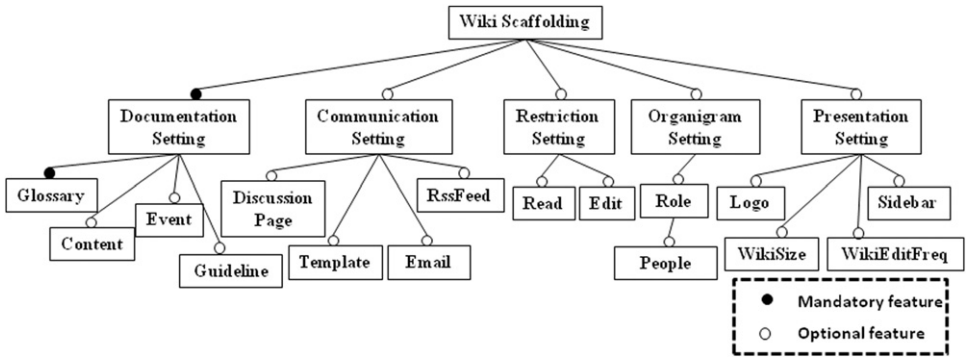

Fig. 1. Feature diagram: corporate concerns that might impact wiki operation.

of features. Fig. 1 depicts the feature diagram for *WSL*. The diagram states that a *WSL* expression captures the company settings in terms of existing documentation practices, communication means, restrictions, the existing organigram and finally, presentation concerns. The rationales for these features should be sought in the previous quotations as well as our own experience. Next paragraphs introduce each feature.

Documentation Setting: A common problem for open communities is that of fixing a common terminology and understanding. This is easier in the case of corporate wikis where glossaries, documentation guidelines or even, some content might already exist. This setting needs to be captured in wiki terms. A basic classification of wiki pages is that of *articles*, *categories* and *templates*. Articles stand for the *content* that is incrementally and collaboratively edited. Next, categories are commonly used as tags to easily locate, organize and navigate among articles. Corporate glossaries can help to identify initial wiki categories. Finally, templates provide content to be embedded in other pages. Through parametrization, they permit to reuse and ensure a formatted content along distinct pages. Corporate guidelines can then be re-interpreted as wiki templates that guide article editing.

Fig. 1 depicts *glossary*, *content* and *guideline* as three features of the company's documentation practices that can impact the wiki. Moreover, wikis frequently support ongoing projects where project milestones might need to be accounted for by the wiki. This does not apply to other settings where content is the result of free-willing participation and hence, contribution is not tight to set schedules. Wiki wise, this implies that *event* is a semantically meaningful piece of data, and so should it be markuped and rendered (e.g., through a calendar).

Communication Setting: Wikis are an effective mechanism to support knowledge formation. This implies the existence of coordination and conflict resolution strategies. When wikis are deployed in an existing organization, wikis become an additional means that should be integrated with existing communication channels. This poses a range of questions: Who is going to be notified of what? Does the existing organizational structure need to be mirrored in the wiki? How is currently achieved such communication? Can email/phone/chatting be effectively used for this purpose?

Wiki wise, communication can be internal or external. Internal communication is achieved within the wiki. At this respect, two mechanisms are considered: *discussion pages and templates*. Discussion pages (a.k.a. *talk* pages in *MediaWiki*) can be used for discussion and communicating with other users. In this way, discussions are kept aside from the content of the associated page. Templates have also been identified as effective means to deliver fixed messages (e.g., warnings, to-do reminders, etc.). On the other hand, external communication refers to the ability to notify wiki changes outside the wiki itself (e.g., through *RSS feeds* or *email*).

Restriction Setting: Unlike open wikis, corporate wikis normally limit access to its own staff. Permissions are counterintuitive in a wiki setting where openness is a hallmark. Indeed, *MediaWiki* natively supports a basic mechanism where the scope of permissions is the whole wiki: you can either edit the whole set of wiki pages or not. By default, wiki pages can be freely operated. However, permissions are more stringent in a company setting, and fine-grained scopes need to be introduced. Indeed, a study on the use of wikis in the enterprise reports that power relationships and competition between stakeholders created a need to read-only access [13]. For the time being, two permissions are considered: *read* and *edit*. Additional permissions could be added in future releases if feedback so advises.[1]

Organigram Setting: The notion of *role* captures the distinct activities *people* (i.e., employees) can play within the organization (e.g., developer, analyst, etc.), and which can also impact wiki edition and management.

Presentation Setting: Companies care for their image on the Web. Wikis resort to *skins*[2] for rendering. These skins are platform specific. However, we do not expect our target audience to know about skins. We should strive to capture presentation concerns in abstract terms, better said, through domain criteria that could later be used by the *DSL* engine to determine the most appropriate skin. Specifically, we consider the expected size and editing

[1] *MediaWiki* permissions include *read*, *edit*, *createpage*, *createtalk*, *upload*, *delete*, *protect* (i.e., allows locking a page to prevent edits and moves), etc.

[2] A skin is *a preset package containing graphical appearance details*, used to customise the look and feel of wiki pages.

frequency of the wiki **as domain criteria, denoted as the** *wikiSize* and the *wikiEditFreq*, respectively. Based on these criteria, heuristics make an educated guess about the wiki skin. In this way, the *DSL* engine frees stakeholders from being knowledgeable about presentation issues, offering good-enough outputs. Notice that the wiki administrator can later change this automatically-generated skin. Additionally, the *logo* and *sidebar* features are introduced for customizing both the headers and the index panes of the wiki.

3. WSL design

In a DSL context, a feature diagram serves to state the commonalities and variabilities of the domain at hand, so that commonalities are built-in into the DSL engine whereas variabilities are supported as parameters to be set by the DSL user [5]. These parameters to-be-set-by-the-user are so provided as a DSL expression. This expression follows a concrete syntax which in turn, is a realization of the DSL's abstract syntax. The latter takes the form of a metamodel where the features captured during the DSL analysis are enriched to be fully operative.

3.1. WSL abstract syntax

The abstract syntax describes the concepts of the language, the relationships among them, and the structuring rules that constrain the model elements and their combinations in order to respect the domain rules. This is expressed as the DSL metamodel (see Fig. 2). The constructs of the DSL are obtained from the feature diagram. A *Scaffolding* model includes four main model classes, namely:

- the *Content* class, which is a graph described along *Items* and *Links*. *Items* capture the different kinds of data existing in the organization that need to be also available at wiki inception. As identified in Section 2, this content includes glossary terms to be included as categories (*glossary itemType*), content to be readily available as a wiki article (*content itemType*), guides for content structure (*template itemType*), or events to capture scheduling milestones (*event itemType*). Next, *Links* relate these *Items* together. *Links* are also typed based on the type of the related *items*: *relatedWith* link (a general *item*-to-*item association)*; *belongsTo* link (to associate a *category* to an *item); templatedBy* link (to associate a *template* to an *item*); *scheduledFor* link (to associate an *event* to an *item*). Items also hold three boolean properties: *discussion* (to indicate whether this *item* is subject to discussion), *rssFeed* (to specify the availability of a feed subscription for this *item*) and *indexPaneEntry* (to capture that the *item* is to be indexed in the sidebar).
- the *Organigram* class, which captures a basic arrangement of *People* (i.e., employees) in terms of *Roles*.
- the *Restriction* class, which binds together three elements: a permission *subject* (i.e., an *Item*), a permission *grantee* (i.e., a *Role*) and a *denial* (i.e., *read* and *edit*).
- the *Presentation* class, which holds properties to guide the rendering of the wiki (i.e., *wikiSize*, *wikiEditFreq*). In addition, three common index schemas are preset: *toolboxPane* (index entries: *what links here*, *Upload file*, *printable version*, etc.), *navigationPane* (index entries: *recent changes*, *help*, *main page*, etc.) and *searchPane* (a search box to locate articles based on content). It is also possible for the designer to define an *ad hoc* index pane, i.e., its entries are defined through the *indexPaneEntry* attribute: true causes an entry in the pane for that *item*.

3.2. WSL concrete syntax

The concrete syntax comprises a mapping between the metamodel concepts (i.e., the abstract syntax) and their textual or visual representation. While the abstract syntax

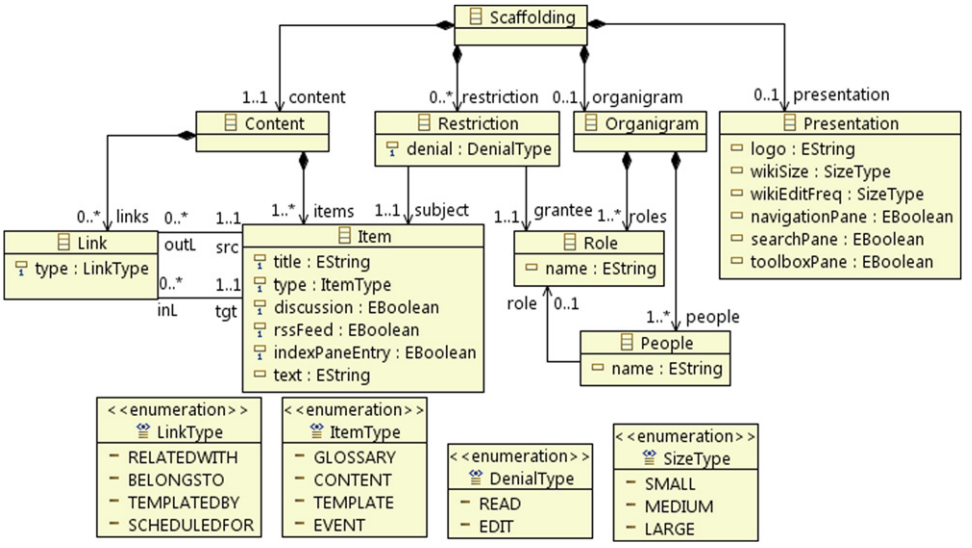

Fig. 2. *WSL* metamodel (abstract syntax).

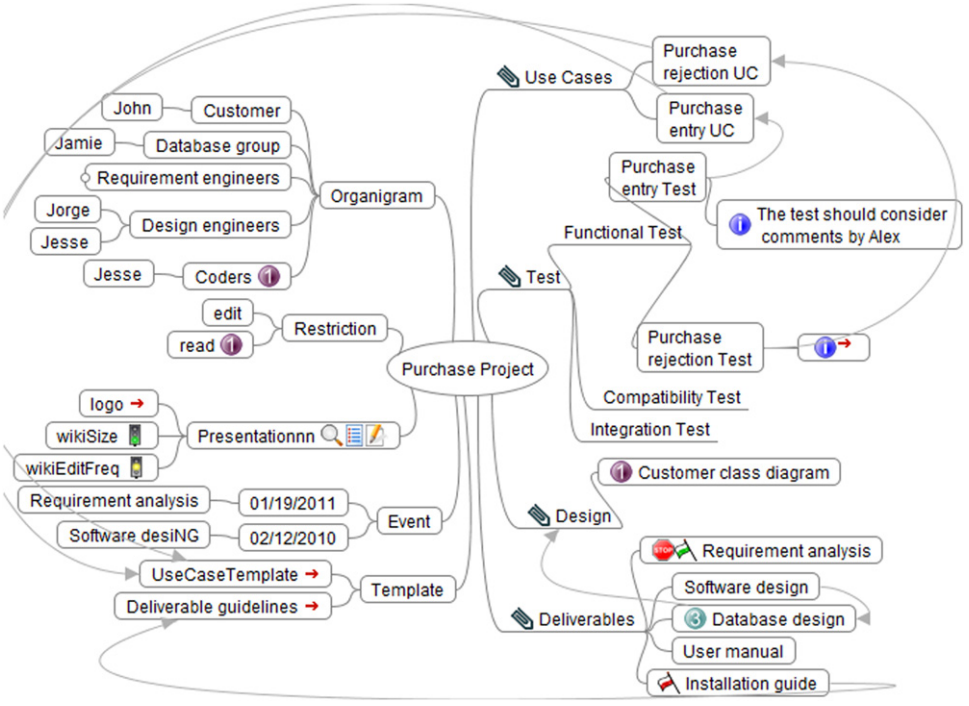

Fig. 3. A sample *Scaffolding* for wiki-based software project management (some errors are made on purpose for future debugging).

addresses expressiveness, the concrete syntax cares for usability as for the target audience. Our target audience is ordinary users. On these grounds, we select mind maps as the concrete syntax for *WSL*. That is, a scaffolding is to be captured as a mind map. The reasons are twofold. First, mind maps offer a way to display different concerns radiantly (see Fig. 3). The limited coupling between the different scaffolding features suits this radial distribution. Second, mind maps are catching on for decision taken within organizations. Indeed, mind maps are reckoned to be a valuable, visual approach for people to collaborate and share ideas [6]. Therefore, we expect organizations to be used to mind mapping, and hence, reducing the learning cost of *WSL*.

However, there exists a plethora of graphical representation and tools for mind mapping. Rather than developing our own visual representation, we decide to capitalize on an existing editor: *FreeMind*. We stick to *FreeMind* on the following grounds: popularity (over 6000 daily downloads), soundness (over 8 years in the market), interactiveness (e.g., easiness to play around with the map: nodes and their descendants can be easily moved and edited; branches can be collapsed, etc.), open source (access to the source code), extensibility (through plugins), export facilities (maps can be turned into applets, html code, flash code or image formats, which can next be embedded as part of the wiki content) and finally, scripting (*Groovy* scripts can be attached to nodes.[3]) Before delving into how *WSL* constructs are mapped into *FreeMind* elements, next subsection introduces an example.

[3] http://freemind.sourceforge.net/wiki/index.php/FreeMind_0.9.0:_The_New_Features accessed 25-Jan-12.

3.2.1. *WSL to support software projects*

Wikis have been proposed for software documentation and planning. The distribution of stakeholders, the need for collaboration and tracking, and the iterative manners that characterize software projects make wikis an attractive platform [17]. Fig. 3 provides an example for the "Purchase" project.

FreeMind depicts ideas and their relationships as nodes and edges that follow a radial distribution. In our example, the *Organigram* branch captures the existing roles (e.g., *Customer*, *Coders*, etc.) as well as the employees (e.g., *Jamie*, *Jesse*, etc.) assigned to these roles. The *Restriction* branch lists limitations in terms of wiki operations. The *Event* branch captures two milestones attached to *Requirement analysis* and *Software desiNG*. Next, the company already has some guidelines to capture use cases and document deliverables. Such practices should also be adhered to when in the wiki. The *Template* branch refers to two such guidelines through the *UseCaseTemplate* node and the *Deliverable guidelines* node. 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 (red light), medium (yellow light) and small (green light). As for the sidebar, this node includes a navigation pane (denoted by the *list* icon) and a search pane (denoted by the *magnifier* icon). The sidebar is finally completed with an index pane (denoted by the *look here* icon on categories *Use Cases*, *Test*, etc.). Regarding to restrictions, *priority* icon sets a restriction whereby *Coders* (i.e., the role) are restricted from *read* (i.e., the denial type) the article *Customer class diagram* (i.e., the item).

As for the corporate glossary, common terms already in use include *Use Cases*, *Functional Test*, *Compatibility Test*, etc.

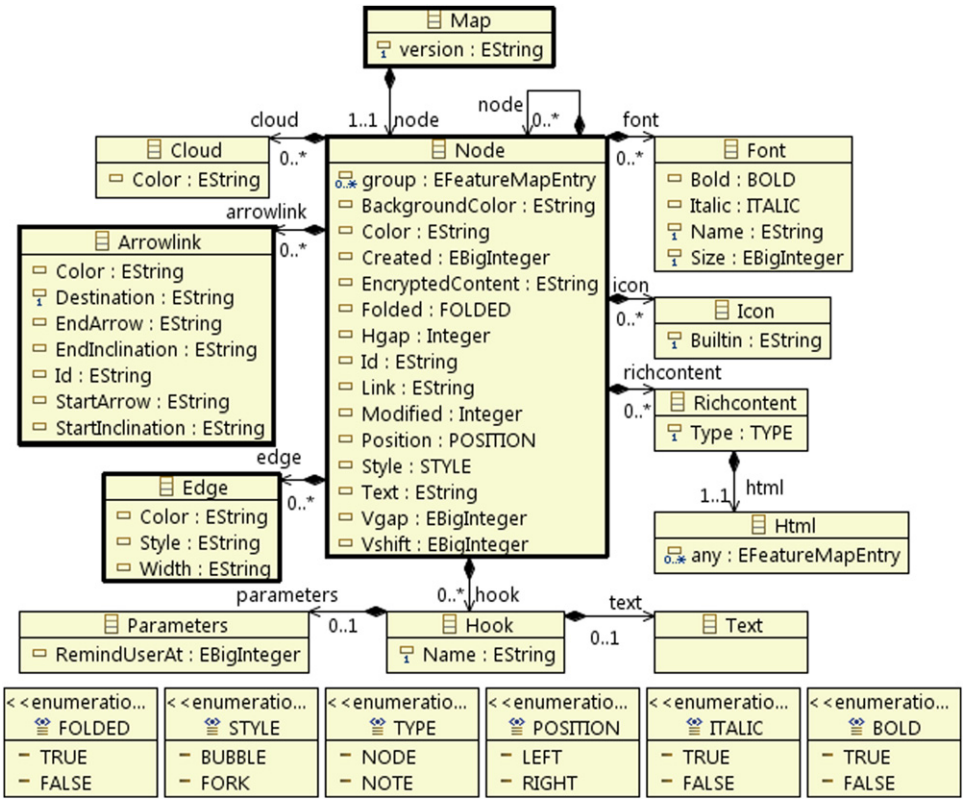

Fig. 4. *FreeMind*'s metamodel for mind mapping.

These terms find their way as wiki categories. Hierarchical relationships among categories are captured by describing a category as a child of the parent category (e.g., *Test Functional Test*). Wiki articles are denoted as *bubble* nodes (e.g., *Requirements analysis* stands for an article which is categorized as *Deliverables*).

It can look odd to introduce articles at wiki inception since wiki's *raison d'etre* is precisely collaborative editing. Indeed, we do not expect too many articles to be introduced at scaffolding time. However, the need to come up with some articles might be known from the very beginning. The scaffolding permits so by introducing a node whose title becomes the title of the wiki article. For instance, the node *Software design* yields a wiki article with the namesake title. Even more, some relationships might be known at the outset. For instance, trace requirements made advisable to keep a hyperlink between the *Purchase entry test* and the *Purchase entry UC*. This is depicted as an arrow link between the node counterparts.

Based on preliminary user feedback, we also consider article content to be known at scaffolding time. This is realized as a child of the given article (together with the *info* icon). Fig. 3 illustrates the two options. The content of *Purchase entry test* is explicitly provided as the text of its child node. By contrast, the content of *Purchase rejection test* is already 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 our advantage to link *Purchase rejection test* to the external document holding its content. Likewise, corporate guidelines can find their way as wiki templates. So far, *WSL* only supports *Word* documents (exported as *XML*). At deployment time (i.e., when the *WSL* expression is enacted), these external documents are turned into either, article content or wiki templates. The rest of this section provides a detailed account of *WSL* expressivity.

3.2.2. WSL concrete syntax

WSL is a visual language for "Wiki Scaffolding" on top of *FreeMind*. This implies: (1) the setting of a mapping between the *WSL* metamodel (see Fig. 2) and the *FreeMind* metamodel (see Fig. 4), and (2), a set of constraints that restricts *FreeMind* maps to be compliant WSL expressions.

WSL-to-FreeMind mapping: First, we introduce the *FreeMind* metamodel. *FreeMind* uses a *XML Schema* to denote what is a *valid* map. Fig. 4 depicts the *FreeMind* metamodel obtained from this *XML Schema*. A *Map* is a compound of *Nodes* (there is a root node and its descendants). Nodes have a *Text* that represents its title, and might hold a *link* to an external document (local or remote) as well as a set of properties mainly referring to rendering concerns. For instance, the *Style* property can be *fork* or *bubble* and determines the look of the node as a tagged line or a bubble, respectively. Next, nodes are basically arranged in a tree-like way. Tree structures are constructed using *Edges*. An edge is a graphical connector that relates a node with its immediate descendants. In addition, *Arrowlinks* are also connectors but in this case, the connection is

Table 1
Wiki Scaffolding-to-*FreeMind* mapping and *FreeMind*-to-*MediaWiki* mapping.

WSL	FreeMind	MediaWiki
Scaffolding	*root* node	main page[a]
Organigram	*Organigram* bubble node	n.a.
Role	child of *Organigram* node	wiki group
People	grandson of *Organigram* node	wiki user and user page
Presentation	*Presentation* bubble node	wiki skin[b]
logo	*logo* node	wiki logo
wikiSize	*wikiSize* node	wiki skin
wikiEditFreq	*wikiEditFreq* node with *traffic light* icons	
navigationPane	*list* icon	navigation in sidebar
searchPane	*magnifier* icon	search in sidebar
toolboxPane	*refine* icon	toolbox in sidebar
indexPane entry	*look here* icon at *Item*	element in the navigation bar
Restriction	*Restriction* bubble node and *priority* icons	blacklisted pages for groups[c]
denial	child of *Restriction* node	wiki permission.
Item		
title	node text	page title
category Item	fork node	categorypage
article Item	bubble node	article page
template Item	child of *Template* node	template page
event Item	child of *Event* node	calendar extension[d]
discussion	*stop-sign* icon	talk page for that page
RSSfeed	*flag* icons	RSS generator for that page[e]
text	child with *info* icon or linked files	page content
Link		
relatedWith Link	arrowLink connector	inter-page hyperlink *[[page]]* *[[:Category:parentCat]]*
belongsTo Link	edge connector	page-category hyperlink *[[Category:parentCat]]*
templatedBy Link	arrowLink connector	template-page hyperlink {{*template*}}
scheduledFor Link	edge connector	event-to-page link in the calendar widget

[a] CategoryTree: www.mediawiki.org/wiki/Extension:CategoryTree accessed 9-Feb-12.
[b] MediaWiki skins include monobook (default), vector (e.g., used by Wikipedia), etc. WSL completes the offer with cavendish, rilpoint, guMax, guMaxDD and guMaxv.
[c] Blacklist: www.mediawiki.org/wiki/Extension:Blacklist accessed 9-Feb-12.
[d] Barrylb: www.mediawiki.org/wiki/Extension:Calendar_(Barrylb) accessed 9-Feb-12.
[e] WikiArticleFeeds: www.mediawiki.org/wiki/Extension:WikiArticleFeeds accessed 9-Feb-12.

between two arbitrary nodes (this enables mind maps to support graph-like structures). Finally, *Icons*[4] and *Fonts* can be associated with nodes in an attempt to reflect the underlying semantics of the node (e.g., a user identifies in red important nodes). Of course, this semantics resides in the users' head.

Once the elements of a *FreeMind* map are introduced, we proceed to indicate how *WSL* metamodel elements are going to be depicted using these *FreeMind* elements. *FreeMind* extensibility would have allowed us to introduce our own symbols and icons. However, we strive to stick with *FreeMind* notation (including icons) to minimize the gap to what *FreeMind* users are accustomed to. Table 1 (first two columns) indicates this mapping:

- *Scaffolding* class: The *root node* is the *FreeMind* counterpart of this class.
- *Organigram* class: A bubble node with title *Organigram* denotes the origin of the organigram hierarchy. Nodes having *Organigram* as parent denote *roles*. Likewise, nodes having *Organigram* as grandparent are interpreted as *people* (i.e., employees).
- *Presentation* class: A bubble node with title *Presentation* denotes this class. Boolean properties are captured as icons on *Presentation*. Value-based properties are

[4] *FreeMind* provides a fixed set of icons. In the last version, users can introduce their own icons, though this is not recommended for interoperability reasons.

represented as children nodes: *logo* (captured as a link to an image file), *wikiSize* and *wikiEditFreq*. The latter are decorated with traffic-light icons to account for their values.

- *Restriction* class: A bubble node with title *Restriction* denotes this class. A restriction is a triple: subject (i.e., an *Item* node), grantee (a *Role* node), and the denial type (i.e., *read* or *edit*). We resort to *priority* icons to denote those elements that conform to a restriction unit. That is, mind map nodes decorated with the same *priority* icon belong to the same *restriction*. Due to icon availability, *permissions* are limited to ten (*priority* icon 0. .9).
- *Content* class: There is not a *FreeMind* counterpart for the *Content* class as such. All nodes in the mind map except for *Organigram*, *Presentation*, *Restriction*, *Event* and *Template* nodes (and descendants) stand for *Content Items*. The node title behaves as an identifier; so that two nodes placed differently but with the same title, stand for the same *Item*. This allows the *Content* graph to be flattened as a *FreeMind* tree.
- *Item* class: *Items* are typed as *category*, *article*, *template* and *event*. *Category Items* are denoted as fork nodes (i.e., nodes with the fork style). *Article Items* are captured as bubble nodes. Next, *Template Items* are children of the *Template* node. These nodes can either hold the page text content (i.e., text attribute) themselves as a child with the *info* icon or point to external documents from where the content is obtained at compile time (only txt and word as xml exported files in the current version). Finally, *Event Items* are children of the *Event* node. As for the boolean properties, *discussion, rssFeed and indexPaneEntry*, the affected *Items* (regardless of their type) are decorated with the *stop sign* icon, a *flag* icon and *look here* icon, respectively.
- *Link* class: *Links* are classified as *relatedWith*, *belongsTo*, *templatedBy* and *scheduledFor*. *FreeMind* offers two kinds of connectors: *Edges,* which are the default arcs connecting a node with its child, and *ArrowLinks*, which are arcs connecting two nodes anywhere in the map. *Edges* are interpreted as *belongsTo* links when they connect an *Item* to a *category Item* (e.g., Fig. 3, arc from *Database design* to *Deliverables*) and as *scheduledFor* when they connect an *Item* to an *event Item* (e.g., Fig. 3, edge from *Requirement analysis* to *01/19/2011*). As for *ArrowLinks*, they sustain (1) *RelatedWith* links when they relate an *Item* to another *Item* (e.g., Fig. 3, arc from *Software design* to *Database design*) and (3) *TemplatedBy* links when the ingoing node stands for a *template Item* (e.g., Fig. 3, arc from *Purchase entry UC* to *UseCaseTemplate*).

Constraints: A *WSL* expression is a compliant *FreeMind* mind map. However, the opposite does not hold. Some mind maps might not deliver a compliant Wiki Scaffolding, where compliance is determined by WSL's abstract syntax (Fig. 2). Therefore, *WSL* maps are a subset of the possible maps that can be drawn in *FreeMind*. Specifically, *FreeMind* mind maps are internally represented as *XML* files along an *XML* schema. On top of it, *WSL* imposes an additional set of constraints that ensures that maps

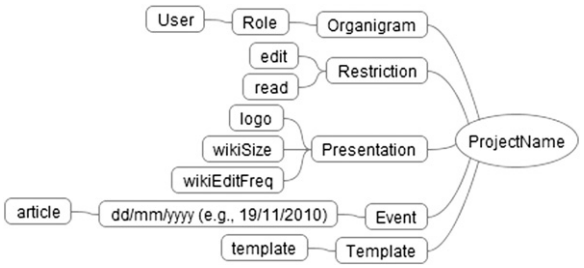

Fig. 5. *WSL* skeleton: a basic *WSL* template to get going.

account for compliant scaffoldings (i.e., conform to the *WSL* abstract syntax). For instance, an *organigram* node should at least have one *role* node. If this is not the case, despite being a perfectly valid mind map, the system raises an error, and the map cannot be "exported" as a wiki installation (more examples in Section 4.2).

4. WSL usage

This section describes the common lifecycle of a *WSL* expression: edition, verification and enactment, and provides some hints about the installation of the WSL engine.

4.1. Edition

WSL expressions are edited as a map in *FreeMind*. To give users a head start, the canvas can be initialized with a "skeleton" that draws the main elements of a scaffolding map (see Fig. 5). From then on, users are free to handle the scaffolding as any other map. Notice, however, that not all maps are scaffolding maps. This moves us to the next subsection.

4.2. Verification

WSL maps are a subset of *FreeMind* maps, i.e., *WSL* metamodel imposes additional constraints on top of the *FreeMind* metamodel. Such constraints can be verified on user request or at enactment time. Fig. 6 provides a snapshot of the *Tools* menu now extended to address *WSL* maps*: WSL Configuration…* permits to configure parameters for the *MediaWiki* installation; *WSL Deployment* causes the generation of the wiki instance from the *WSL* specification; *WSL Skeleton* provides a *FreeMind* map with the basic *WSL* nodes (e.g., *Organigram*, *Restriction*, etc.) so that misspells are prevented; and finally, *WSL Map Checking* triggers *WSL* map verification.

Fig. 6 depicts the verification outcome for our sample problem (see Fig. 3). Messages can be either warnings or errors. For our sample, two warnings are noted. One informs about the lack of the *Presentation* node which, in this example, is due to a misspelling (e.g., *Presentationnn*). The other warning notifies about a common mistake in wiki construction: setting a *relatedWith* relationship between an article and a category. This is an odd situation that could be mistaken with the *belongsTo* relationship, and so is it indicated. As for errors, they prevent the wiki from being generated. For our sample

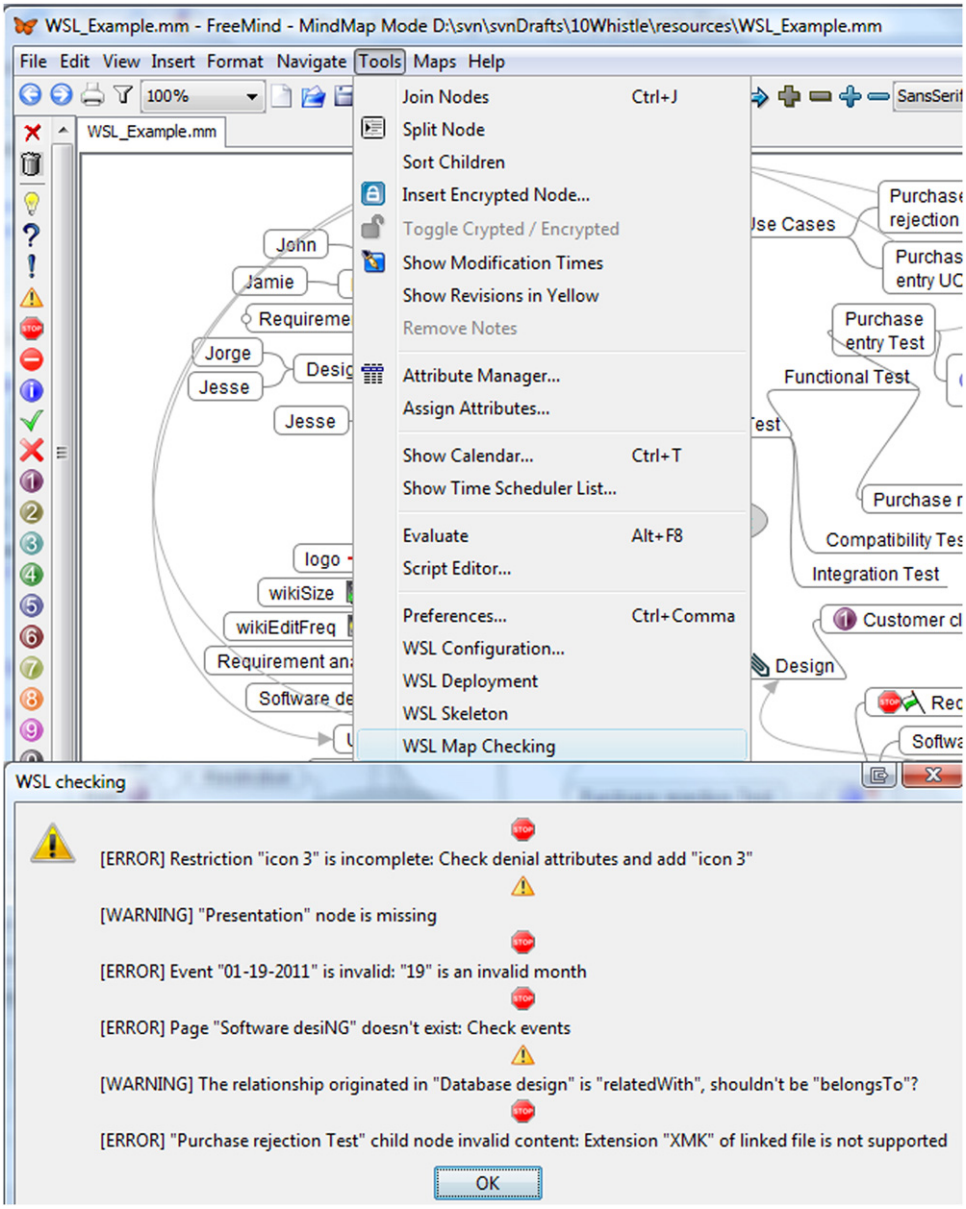

Fig. 6. Verifying *WSL* expressions: an example for the map in Fig. 3.

case, these errors include: a misspelling of an event date (e.g., "01/19/2011"); referring to a non-existent node (e.g., *Software desiNG*); partial definition of a restriction (i.e., either the denial, the employee or the article is missing) (e.g., restriction 3); unsupported document extension (e.g., extension *XMK* is not supported).

4.3. Enactment

By selecting the *WSL deployment* option of the *Tool* menu (see Fig. 6), the current map is turned into a wiki installation in *MediaWiki*. This means that around 1000 LOC (mainly *SQL* statements) are automatically generated for the current example.

Figs. 7 and 9 provide three screenshots of the generated pages: the former with the main page (illustrating the use of the *CategoryTree* and *Calendar* extensions), and the latter with the *Purchase rejection Test* article (which is obtained from a *Word XML* document) and the *Purchase Rejection UC* (which follows the *UseCaseTemplate* also externally obtained). For the purpose of this paper, it is enough to show the mapping between the *FreeMind* and the*MediaWiki* constructs. The last two columns in Table 1 indicate such mappings.

4.4. Installation and architecture

The *WSL* engine is implemented as a *FreeMind* plugin. For installation, proceed as follows: (1) download and

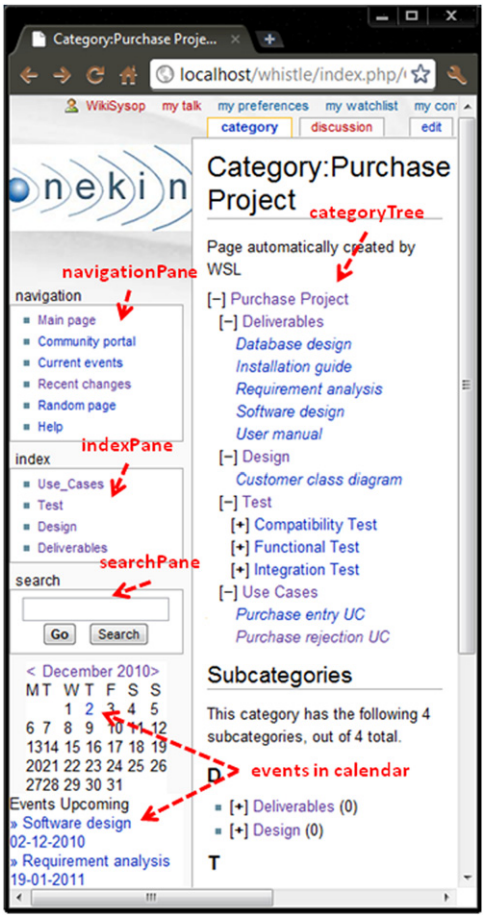

Fig. 7. Wiki main page as generated by *WSL* for the *Purchase Project* example.

install *FreeMind*,[5] (2) download and install the *WSL* engine,[6] and (3) download a *WSL* sample.[6] *WSL* has been tested against *MediaWiki* 1.16.1 and *FreeMind* 0.9.0.

Fig. 8 outlines the architecture. Wikis are supported by wiki engines (e.g., *MediaWiki*). Although files are also an option (e.g., *DokuWiki*), wiki engines tend to store the wiki content in a database. A "Wiki Scaffolding" is the initialization of a wiki project. Therefore, a "Wiki Scaffolding" initializes a database. In other words, the enactment of a*WSL* expression causes a set of tuples to populate the *MediaWiki* database. This process goes as follows (see Fig. 8): (1) the mind map (i.e., a.*mm* file) is turned into a *WSL* model (i.e., a. *xmi* file along the *WSL* metamodel)[7]; (2) this *WSL* model is next turned into a *SQL* script[8]; and (3) this *SQL* script is run against the *MediaWiki* DBMS (i.e., *MySQL*).[9] In addition, realizing the scaffolding might require specific *MediaWiki* extensions. Specifically, *Blacklist*, which restricts access to specified pages in a black list, *Calendar*, for event rendering in a calendar, *CategoryTree*, which gives a view of the wiki structure as a tree, *EmailPage*, to send wiki pages by email, and *WikiArticleFeeds*, for turning wiki changes into *RSS* and *Atom* feeds.

As any other plugin, this architecture raises evolution concerns. The *WSL* engine might be affected by (1) changes in *MediaWiki* (or its extensions), (2) changes in the underlying database schema (this impacts the *MOFScript* transformation), and (3) changes in the *FreeMind* metamodel (this impacts the *XSLT* transformation). This is certainly true. But, how real is this threat? Both *FreeMind* and *MediaWiki* are stable platforms backed by thousands of installations. In addition, wikis can be upgraded once deployed. Remember that *WSL* is used just to initialize the wiki. Once the scaffolding is deployed, users can upgrade the wiki to the newest version if required. Notice that the wiki can next evolve in ways that contradict the scaffolding (e.g., new users or templates can show up), but this does not erode the benefits that scaffolding brings at the onset. All in all, wiki refactoring is certainly an issue (see [18] for the use of conceptual maps for wiki refactoring).

5. WSL at work

This section tests out WSL through different case studies. Each case is inspired in a real scenario where the use of scaffolding can be at an advantage. For each case study, we identify scaffolding matters, depict the *WSL* maps, and finally, generate the *MediaWiki* installations.[10] Besides illustrating *WSL*, each example highlights a scaffolding advantage (in bold). Table 2 outlines how different scaffolding matters are realized in these scenarios.

5.1. Scaffolding to promote user engagement

Prompt user engagement has been identified as a main success factor for wikis [9]. "Wiki Scaffolding" gives users an initial setting where some artefacts (e.g., categories, templates, articles) are available from the start. Fig. 10 depicts a *WSL* map along the experiences reported in [9]. Cole mentions six areas that are know from the start. They could be represented as either articles (e.g., *Paradigm shift*) or categories (e.g., *Development techniques*). Regarding the comment of a student *there aren't any useful guidelines or tips that could be used*, content about wiki usage (e.g., a *wikis for dummies* internal report, or URLs to appropriate places) might be included as page text just by linking that file to the *WSL* node. Furthermore, FAQ collected in the classroom might be made readily

[5] http://freemind.sourceforge.net/wiki/index.php/Download accessed 25-Jan-12.

[6] http://www.onekin.org/wsl accessed 25-Jan-12.

[7] Since *FreeMind* maps are XML files, this transformation is realized as a *XSLT* transformation, which is natively supported by *FreeMind*.

[8] This model-to-text transformation is accomplished through *MOFScript* http://eclipse.org/gmt/mofscript/ accessed 25-Jan-12.

[9] All the connection parameters (database name, db user login, db password, db host name) are obtained from the *LocalSettings.php* as part of the *MediaWiki* installation, and provided by the user in the *WSL Configuration…* option.

[10] Available for inspection at www.onekin.org/wsl accessed 25-Jan-12.

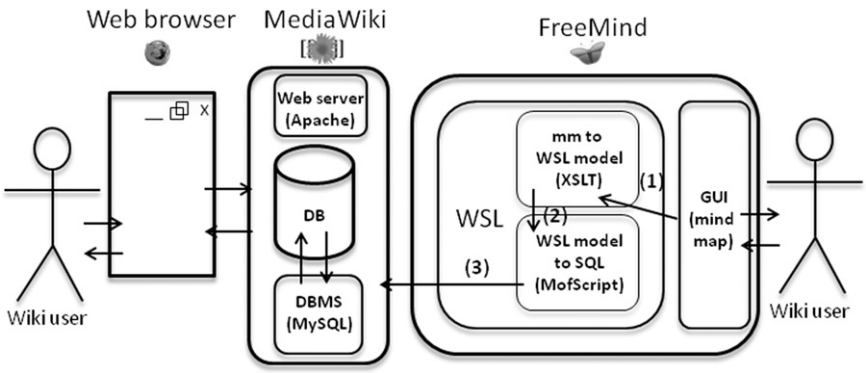

Fig. 8. *WSL* architecture: *FreeMind* as another interface on top of the *MediaWiki* database.

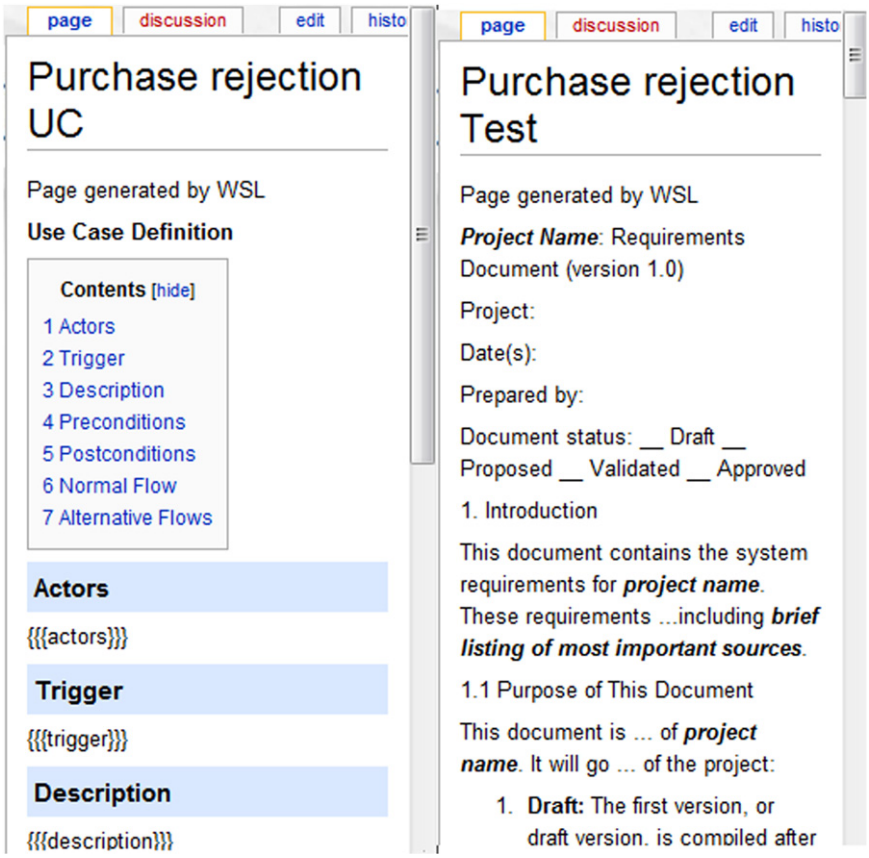

Fig. 9. Template and article pages as generated by *WSL* for the *Purchase Project* example.

available at the onset. In addition, communication mechanisms (e.g., email, *RSS* feed and discussion pages) can be added to promote all, student collaboration (e.g., do you know an answer to a common doubt?), encourage participation (e.g., do you agree with the present year assessment method?) and incite the work group (e.g., could we improve our individual grade by working together?).

Based on previous teaching experiences, articles which are expected to raise a debate, can be created with either a companion discussion page (i.e., *stop-sign* icon) or a *RSS* feed (i.e., *flag* icon). This is the case of the *ISD methodologies* and *future directions* articles. Event though, the articles are empty, the scaffolding already provides the infrastructure to initiate the discussion. In addition, some articles might need to follow some guidelines. An obvious example is that

Table 2
Setting matters that could impact wiki operation.

Scaffolding	Academic wiki	Gaming community wiki	Veterinary education wiki	Software project wiki
Glossary	Keywords of the taught subject	Game jargon characters, worlds, weapons	Clinical taxonomies pathology, drugs, viruses	Software development keywords: use cases, tests
Content	Syllabuses, exams, FAQ	Story line, tricks	Patents, terms of service, sponsors	Technical manuals, README
Events	Exam dates, assignment deadlines	Game releases	—	Planned meetings, project milestones
Guidelines	Exam patterns, assignment guidelines	Weapon explanation, character features	Animal features, treatment steps, student's page	Use case, deliverable template
RssFeed	Doubts	Patch release, announcements, online news	—	Requirement updates
Discussions	FAQ, tough themes	New features, support	Drugs in trial, innovative sponsors	Requirements, test results
Email	Teamwork	Bug communication	teamwork	Meeting with customer
Restrictions	Certain students edit assignments, only lecturers edit exams	Developers' pages	Discussions only by qualified people	Requirements set only by stakeholders and analysts
Roles	Student, lecturer	Developer, player, tester, resellers	Student, nurse, veterinarian	Stakeholder, analyst, designer, coder
Presentation	University logo	Game "look and feel"	—	Organizational image

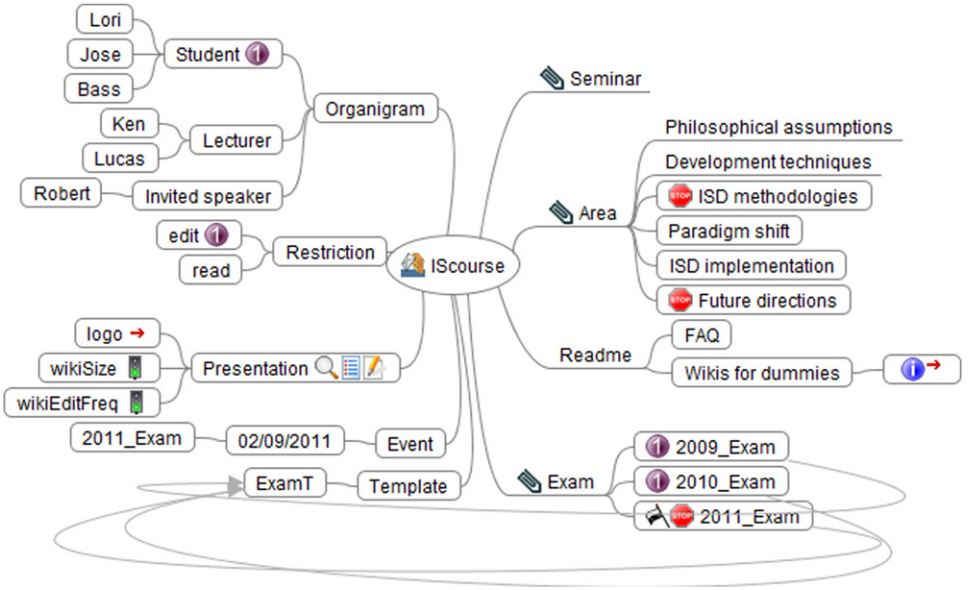

Fig. 10. *WSL* scaffolding for a wiki to support student engagement. Output available at www.onekin.org/wsl/IScourse. Username: Lori, Password: 12345.

of exams. Templates can be used to guide template-aware articles. The example shows an arrow link from exam articles to the *ExamT* template. Access rights are defined that prevents contributors, belonging to the *Student* role, from editing the *2010_Exam* and *2009_Exam* articles. Exam articles can be qualified by an event. Finally, the expected size and

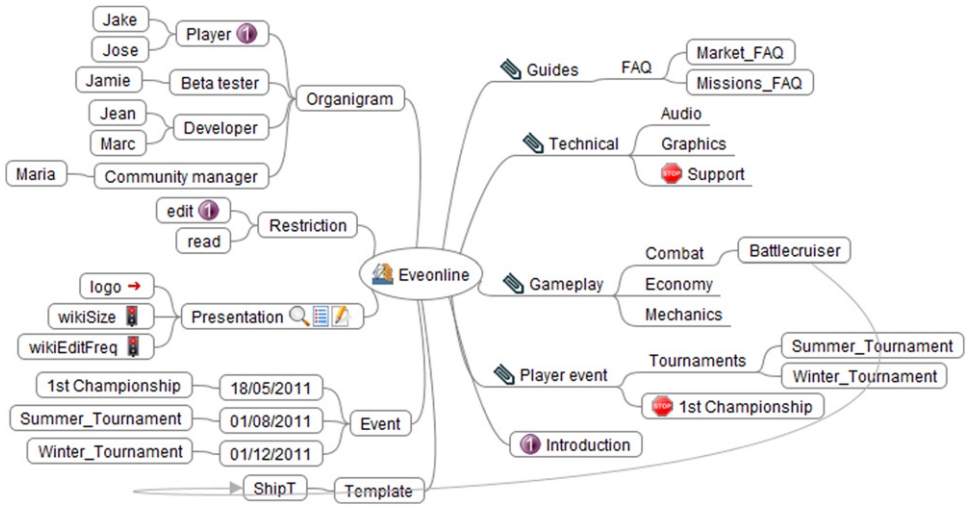

Fig. 11. *WSL* scaffolding for a video-gaming wiki. Output available at www.onekin.org/wsl/Eveonline Username: *Jake*, Password: 12345.

editing frequency are both low as denoted by the green *traffic lights* icons.

5.2. Scaffolding to mirror existing organizational practices

Organizational wikis frequently need to mirror (and follow) existing organizational practices. Introducing wikis in organizations is not easy [19]. Stuff might lack the motivation to learn yet another new technology. After all, other collaboration tools may already exist in the organization including email, distribution lists, intranets, etc. "Wiki Scaffolding" forces to ponder on those practices and resources which might need to be migrated to/integrated into the wiki.

This situation is illustrated by wikis supporting software projects [17]. From a scaffolding perspective, characteristics of relevance include (see Fig. 3): distinct stakeholders work together to organize, track and publish project documentation; wikis act as a version control system to keep track of changes; wikis are useful as *discussion* means (e.g., *Requirements analysis*); they also provide *rssFeeds* to advise changes (e.g., *Installation guide*), *email* capabilities for notifications, project milestones as *events* (e.g., a meeting for the *Software design*), scheduling capabilities, etc. This collaborative management of the project documentation does not occur in a vacuum, but normally adheres to some "work of practice" existing in the company. This includes a role organigram (e.g., *Requirement Engineer*, *Design Engineer*, etc.) where contributions and permissions might depend on the user role (e.g., *Coders* are not allowed to *edit* the *Customer class diagram*, as denoted by the *priority* icon), glossaries (e.g., terms such as *Use Case*, *Functional Test*, *Compatibility Test*, and so on, might be used to categorize wiki content) or company guidelines for artefact production (e.g., a common example is that of use cases).

5.3. Scaffolding as a way to engage management

In order to provide value to the organization, wikis have to solve a clearly specified problem and be aligned with the organizational strategy [20]. Unfortunately, organizational wikis are in many cases a grass root phenomenon whereby the wiki is introduced by an individual employee or a small group within the organization without the support of management. This bottom up approach frequently fails in having a strategic intent. More to the point, a lack of strategy might result in no clear guidelines about what to contribute, how to contribute and who should make the contribution. An example is reported in [21] where a wiki failure was due to an ambiguity in the wiki's aim: some users saw the wiki as a project documentation repository whereas others used it for glossary entries. This led to confusion and dissensions on the wiki's intent.

From this perspective, "Wiki Scaffolding" forces to have a blueprint before releasing the wiki for contribution. Thinking about how the wiki will fit into the existing information ecosystem helps to devise the aim of the wiki in advance. In addition, management support would be facilitated if scaffolding is captured through intuitive means that ease self-edition, sharing or discussion. This favours the use of mind maps.

This situation is illustrated by a video-game community (e.g., www.eveonline.com) (see Fig. 11). The wiki intent is to offer a share space for both consumers and providers of video games to communicate new insights about potential enhancements and new game releases. Contributors are players (a.k.a. gamers) who discuss, share and edit *content*, *guidelines*, documentation, background and resources (i.e., *glossary*) about their favourite video games. Besides players, developers and testers (i.e., *roles*) also participate to gain insights from the players about how to improve their products.[11] There exist some *restrictions* to both avoid misunderstandings and keep organizational policies untouched (e.g., *Players* are not allowed to *edit* the *Introduction* node as denoted by the *priority* icon). Direct communication (through *discussions*, *email* notifications or *RSS* subscriptions) permits

[11] http://wiki.eveonline.com accessed 25-Jan-12.

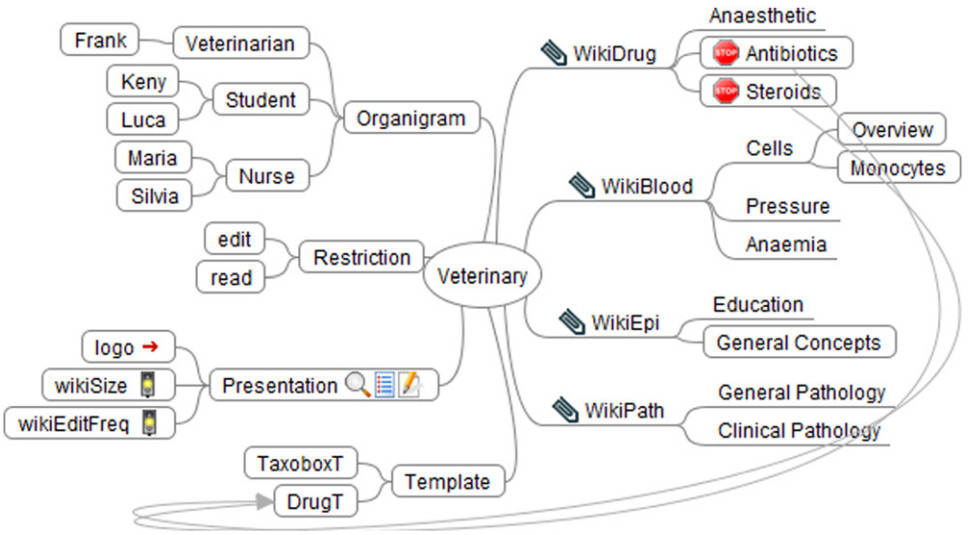

Fig. 12. *WSL* scaffolding for a wiki to support veterinary education. Output available at www.onekin.org/wsl/Veterinary Username: *Keny*, Password: 12345.

developers to know first-hand the players' opinion about new features, bugs, ideas, etc. Common *guidelines* about how to explain game items are represented as templates (e.g., a *Battlecruiser* is a kind of *ship*, so the namesake template is used).

5.4. Scaffolding as a wiki map

The "rules of practice" which govern a site (i.e., roles, access rights, templates, etc.) should be easily accessible to newcomers. So far, this information is scattered around the wiki, and frequently, hidden in administrative pages. At best, a *README* page can provide some textual description of these practices. From this perspective, a "Wiki Scaffolding" can play the role of an initial "practice sitemap". Traditional site maps provide a kind of interactive table of contents, in which each listed item links directly to its counterpart sections on the website. Some wiki engines (e.g., *MediaWiki*) readily provided such map for categories. One step in the same direction would be the use of "scaffolding maps": an HTML representation of a "Wiki Scaffolding" that permits to readily access the wiki's practices. Notice, however, that this will require to keep the scaffolding in sync with the wiki (i.e., new roles, terms, rights, etc.), and to conceive "Wiki Scaffolding" as a supporting infrastructure for collaborative content production whose usefulness goes far beyond wiki initialization. This could be useful for communities where different roles intertwine, and the implications of belonging to a certain role (e.g., in terms of contribution obligations or access rights) should be clearly stated.

This scenario is illustrated by the *WikiVet*[12] [22]. Contributors include veterinarians, veterinary students and nurses (i.e., *roles*), where anonymous users might not be allowed to edit and, sometimes, even read, pages (see Fig. 12). This restriction increases the trustworthy of the peer reviewed material since all the editors are knowledgeable about veterinary. Main categories (i.e., *WikiDrug*, *WikiBlood*, *WikiEpi* and *WikiPath*) pertain to the main index pane (i.e., *look here* icon). *WikiVet* aims to create a veterinary curriculum, e.g., viruses, drugs (i.e., *glossary*), patents, sponsors (i.e., *content*). Some content has a common structure (i.e., *guidelines*) e.g., both *Antibiotics* and *Steroids* follow the *DrugT* template.

6. Conclusions

We introduce the notion of "Wiki Scaffolding" as a means for corporate strategies to permeate wiki construction. We realize "Wiki Scaffolding" as mind map drawing to preserve wikis' openness. The result is *WSL*, a graphical DSL on top of *FreeMind*. By taping into *FreeMind* as the conduit for the *WSL* concrete syntax, we expect non-technical communities to benefit from scaffolding. Potential benefits include facilitating the alignment of the wiki with organizational practices, promoting management engagement, enhancing the visibility of the wiki's practices, or promoting employee participation through direction setting.

WSL constructs are based on a literature survey about the use of wikis in companies. However, the use of corporate wikis is at its inception. It is likely that social conventions and incentives will emerge and evolve to guide contributors, resolve disputes and help manage wikis. As these issues find support in wiki engines, *WSL* constructs will need to be extended. In addition, we have so far focused on the feasibility of the approach and its interest in different scenarios. Additional evidences are needed to claim scaffolding succeed on better aligning wikis to corporate strategies as well as engaging users through direction setting. We plan to deploy *WSL* in organizations that have already been exposed to wikis

[12] http://en.wikivet.net/Veterinary_Education_Online accessed 25-Jan-12.

to collect evidences about the advantages brought by the scaffolding. In so doing, we hope to introduce scaffolding as another step in the wiki ideal of removing "accidental complexity" from technology, and letting ordinary users directly manage and construct their own knowledge.

Acknowledgements

This work is co-supported by the Spanish Ministry of Education, and the European Social Fund under Contract TIN2011-23839 (*Scriptongue*). Puente has a doctoral grant from the Spanish Ministry of Science & Education.

References

[1] D. Carlin, Corporate Wikis Go Viral, 2011. Online; 〈www.businessweek.com/technology/content/mar2007/tc20070312_476504.htm〉 (accessed 25-Nov-11).
[2] P.D. Media, Intranet 2.0 Global Survey, 2011. Online; 〈www.prescientdigital.com/downloads/social-intranet-study-2011-summary-report〉 (accessed 23-Jan-12).
[3] S. Toker, J.L. Moseley, A.T. Chow, Is there a wiki in your future? applications for education, instructional design, and general use, Educational Technology Magazine (2008) 6.
[4] H. Lee, C. Bonk, The use of wikis for collaboration in corporations: perceptions and implications for future research, in: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education.
[5] M. Mernik, J. Heering, A.M. Sloane, When and how to develop domain-specific languages, ACM Computing Surveys 37 (2005) 316–344.
[6] T. Buzan, C. Griffiths, Mind Maps for Business, BBC active, 2010.
[7] Freemind, 2011. Online; 〈http://freemind.sourceforge.net〉 (accessed 25-Nov-11).
[8] K.C. Kang, S.G. Cohen, J.A. Hess, W.E. Novak, A.S. Peterson, Feature-oriented Domain Analysis (FODA) Feasibility Study, Technical Report, Carnegie-Mellon University Software Engineering Institute, 1990.
[9] M. Cole, Using wiki technology to support student engagement: lessons from the trenches, Computers & Education 52 (2009) 141–146.
[10] M. Raman, T. Ryan, L. Olfman, Designing knowledge management systems for teaching and learning with wiki technology, Journal of Information Systems Education 16 (2005) 311–321.
[11] F. Tétard, E. Patokorpi, K. Packalén, Using wikis to support constructivist learning: a case study in university education settings, in: Hawaii International Conference on Systems Science (HICSS), pp. 1–10.
[12] A.J. Phuwanartnurak, Did you put it on the wiki?: information sharing through wikis in interdisciplinary design collaboration, in: SIGDOC.
[13] C. Danis, D. Singer, A wiki instance in the enterprise: opportunities, concerns and reality, in: Computer Supported Cooperative Work (CSCW).
[14] A. Stocker, K. Tochtermann, Enterprise wikis types of use, benefits and obstacles: a multiple case study, in: A. Fred, J.L.G. Dietz, K. Liu, J. Filipe (Eds.), Knowledge Discovery, Knowledge Engineering and Knowledge Management, Communications in Computer and Information Science, vol. 128, Springer, Berlin, Heidelberg, 2011, pp. 297–309.
[15] I. Lykourentzou, Y. Djaghloul, K. Papadaki, F. Dagka, T. Latour, Planning for a successful corporate wiki, in: Digital Enterprise and Information Systems (DEIS), pp. 425–439.
[16] H. Zhu, R. Kraut, A. Kittur, Organizing without formal organization: group identification, goal setting and social modeling in directing online production, in: Computer Supported Cooperative Work (CSCW).
[17] P. Louridas, Using wikis in software development, IEEE Software 23 (2006) 88–91.
[18] G. Puente, O. Díaz, Wiki refactoring as mind map reshaping, in: Advanced Information Systems Engineering (CAiSE).
[19] J. Grudin, E.S. Poole, Wikis at work: success factors and challenges for sustainability of enterprise wikis, in: International Symposium on Wikis.
[20] A. Stocker, K. Tochtermann, Exploring the value of enterprise wikis—a multiple-case study, in: International Conference on Knowledge Management and Information Sharing (KMIS), pp. 5–12.
[21] L.J. Holtzblatt, L.E. Damianos, D. Weiss, Factors impeding wiki use in the enterprise: a case study, in: CHI Extended Abstracts, pp. 4661–4676.
[22] G. Brown, M. Quentin-Baxter, Z. Belshaw, Wikivet: building a community of practice to support a self-sustaining wiki for veterinary education, International Journal of Web Based Communities (IJWBC) 6 (2010) 183–196.