

Post-print de 4th International Conference on Metadata and Semantic Research,
MTSR 2010, Alcalá de Henarès (Spain)
Communication in Computer and Information Science n° 108, pp. 119-130,
Springer-Verlag, Berlin-Heidelberg

Using a semantic wiki for documentation management in very small projects

Vincent Ribaud and Philippe Saliou

LISyC, Université de Brest, UEB, France, CS 93837, 29238 Brest Cedex, France
{Vincent.Ribaud@univ-brest.fr, Philippe.Saliou@univ-brest.fr}

Abstract. The emerging ISO/IEC 29110 standard Lifecycle profiles for Very Small Entities is targeted at very small entity (VSE) having up to 25 people, to assist them unlock the potential benefits of using software engineering standards. VSEs may use semantic web technologies to improve documentation management infrastructure and processes. We proposed to use a semantic wiki for documentation management based on an identification scheme inspired from an IFLA proposition called Functional Requirements for Bibliographic Records. The document identification scheme allows documents to be managed by the internal resource management of the semantic wiki, hence benefiting from a straightforward but powerful version control. With few inputs of semantic annotations by VSE employees - through usable semantic forms and templates, the semantic wiki acts as a library catalog, and users can find, identify, select, obtain, and navigate resources.

Keywords: very small entities, Functional Requirements for Bibliographic Records, ISO/IEC 29110, semantic wikis.

1 Introduction

The term 'Very Small Entity' (VSE) was defined by the emerging ISO/IEC 29110 standard "Lifecycle profiles for Very Small Entities" [1] as being "an entity (enterprise, organization, department or project) having up to 25 people". VSEs can find it difficult to relate software engineering standards to their business needs and to justify the application of the standards to their business practices. A disciplined documentation management may be seen as the first step towards standardization, but at least will provide a significant help for the achievement of projects.

Documentation management is a piece of the puzzle of Knowledge Management (KM). In [2], Chan and Chao present a research survey conducted among 68 small and medium-sized enterprises (SME) which have implemented Knowledge Management (KM) initiatives. They conclude that effective KM is influenced by two types of KM capability, infrastructure and process, which have to be deployed. Publishing and content management systems (CMS) provide generally the

documentation management infrastructure, associated with a documentation workflow process. Instead of these feature-rich systems, we move to an “as simple as possible” system using a semantic wiki as a base technology and a straightforward identification scheme to support the acquisition, organization, maintenance, retrieval, and sharing of documentation.

Section 2 presents the positioning of our work. Section 3 drafts some challenges of documentation management; the core (§3.4) is an identification scheme inspired from Bibliographic Records standards. We conclude with perspectives.

2 Work positioning

2.1 Infrastructure

Publishing and content management systems (CMS) are generally used as the basis for a documentation management infrastructure. But several authors have criticized the rigidity of the editorial control required by a CMS [3] and the need to balance structure/constraint and flexibility [4]. Some are promoting the use of Wikis and RDF (Resource Description Framework) to resolve these issues [5].

As [6] pointed out, a first step towards building the Semantic Web is to have the infrastructure needed to handle and associate metadata with content. However, most authoring environments have a major drawback. In order to provide metadata about the content of a document or a Web page, the author must first create the content and second annotate the content in an additional annotation step. As a way out of this problem, [7] propose that an author needs the possibility to easily combine authoring of a Web page and the creation of relational metadata describing its content.

Our proposition uses a [semantic] wiki to handle Web pages and documents. In a wiki, it is exceptionally easy for anybody to edit Web pages following a small number of conventions. Semantic wikis let users add semantic information to the pages. We use Semantic MediaWiki (SMW, <http://semantic-mediawiki.org>), a free semantic extension of the free software MediaWiki (<http://www.mediawiki.org>). SMW let users edit metadata in a straightforward manner similar to the editing of page content; hence it fulfills the requirement stated above. The problem differs with documents, usually produced in a word processor. Once the document stored in a wiki, access to the document is performed through a wiki page with the same name. This page handles metadata managed by the wiki (e.g. versioning information) and let users upload the resource. In a semantic wiki, this page handles also metadata provided by users and acts as the resource description. Editing the resource description (document metadata) is unfortunately separated from editing the resource.

2.2 Processes

Rech et al. [8] identified several challenges related to knowledge transfer and management for small and medium-sized enterprises in the software sector: recording,

reusing, locating and sharing information. The same observations are applying to documentation management. However, before to address these challenges, we have to consider a supplementary feature: documents may be referred by a name, vague (e.g. user's manual) or precise (e.g. French translation of software requirements for the 2.6 version of given software product). Referring a document (or a set of documents) by name requires identifying documents and relationships among documents.

Identifying documents and their relationships. We are proposing to use a documentation identification scheme, inspired from the FRBR, Functional Requirements for Bibliographic Records [9].

Let us describe the FRBR proposition in a nutshell. FRBR gathers information about *Work*, a distinct intellectual or artistic creation. We recognize the *work* through individual realizations of the *work*, but the *work* itself (e.g. the Hebrew Bible) is a set of concepts regarded as commonly shared by a number of individual sets of signs (e.g. the Hebrew Bible in Biblical Hebrew or its Latin form, *Biblia Hebraica*) called *Expressions* [9]. An *expression* is the specific intellectual or artistic form that a work takes each time it is "realized." An *expression* excludes aspects of physical form, such as typeface and page layout, that are not integral to the intellectual or artistic realization of the work as such [9].

Work and *Expression* are abstract entities; when a *work* is realized, the resulting *expression* of the *work* may be physically embodied on or in a medium such as paper, tape, canvas, etc. That physical embodiment constitutes a *Manifestation*. In some cases there may be only a single physical exemplar produced of that manifestation of the work (e.g. an 11th century manuscript of the Hebrew Bible with Aramaic Targum). In other cases there are multiple copies produced in order to facilitate public dissemination or distribution (e.g. the *Biblia Hebraica Quinta* published by the Deutsche Bibelgesellschaft). Whether the scope of production is broad or limited, the set of copies produced in each case constitutes a *manifestation* [9]. A specific copy - a single exemplar of a manifestation - constitutes an *Item*. In terms of intellectual content and physical form, an *item* exemplifying a *manifestation* is normally the same as the *manifestation* itself.

The FRBR proposition allows us to establish distinctions and precise relationships between the various intellectual creations - artifacts - handled during a software project. Various terms are used by creators and publishers of intellectual and artistic entities to signal relationships between those entities. Terms such as "edition" and "version" are frequently encountered on publications and other materials, as are statements such as "based on ..." or "translated from ..." [9]. The FRBR analyzed specifically relationships that operate between one *work* and another, between a *work* and an *expression*, between one *expression* and another, between a *manifestation* and an *item*, etc.

We will detail in section 3.4, a documentation management scheme that may be suitable for some very small projects. Other projects may have different requirements and interpretations. For instance, we considered that different translations of the same document are different *manifestations* of the same *expression*. Hence, different translations are supposed to share exactly the same set of concepts. That may be not true for VSEs localizing a software product and its associated documentation for a worldwide market. Each translation is itself an *expression* and defining different

expressions gives us a means of reflecting the distinctions in intellectual or artistic content that may exist between one translation and another of the same *work* (the software product and its documentation).

Reusing. Several authors pointed out the help of ontologies to support reuses [8], [10]. By construction, using the ISO/IEC 29110 standard in a VSE introduces its underlying ontology: process, activity, task, role, products, etc. In the case of documentation, the ISO/IEC standard - Basic Profile ([11], Clause 4.5) defines an alphabetical list of the input, output and internal process products, its descriptions, possible states and the source of the product. For instance, the main process related to software development proceeds with:

- Input products: Project Plan;
- Output products: Requirements Specification, Software Design, Traceability Record, Software Components, Software, Test Cases and Test Procedures, Test Report, Product Operation Guide, Software User Documentation, Maintenance Documentation, Change Request;
- Internal products: Validation Results, Verification Results.

We use the Basic Profile list of 22 work products as a basis to categorize documentation (with the possibility for the VSE to add documents types related to its business and organization).

Recording. Rech et al. [8] observed that information and knowledge about the software projects and products exist during the runtime of a project but get lost soon after its end. They propose that adequate documentation should either be supported automatically or semi-automatically by using a single point of consistent knowledge to simplify storage and retrieval.

A VSE needs a simple model to locate, store, and retrieve work products. Our proposal is to replace the hierarchically physical organization (as it may be found in a file system or a CMS) with a logical organization based on the identification scheme presented in section 3.4. Documents are naturally categorized into the class associated with their type. A generic template for documents as well as a template for each document type provide users with a way of specifying resource descriptions without learning any new syntax and ensure that properties and classes are used consistently.

Locating. Locating - retrieving - resources difficulty is related to the crucial problem of interaction between resources providers and users. Ramadour et Cauvet [10] believe that this interaction can be supported and even automated by increasing the expressiveness of the language used for encoding component properties and formulating queries, enhancing therefore the quality of the retrieval.

SMW includes an easy-to-use query language which enables users to write simple or complex queries. The syntax of this query language is similar to the syntax of resource descriptions (typed by the document creator itself or used in templates). SMW provides users with category browsing and with a kind of hierarchical faceted navigation through semantic properties.

Sharing. Uren et al [12] state that a document centric process must handle three classes of data: ontologies, documents and annotations. Documentation sharing is accomplished in an easy manner through the use of the wiki resource management system - that is including version control. Sharing ontologies in a semantic wiki is also simple, because any update to the ontology is immediately available to user - the question is remaining how to update, automatically or semi-automatically, others

wikis using the same ontology. The difficult point is related to annotation sharing: after several unsuccessful attempts, we abandoned to provide a scope control on the annotations, and each published annotation is a public one.

3 Engineering Activities and Documentation Management

3.1 Software Engineering Standards

A concise definition of the objects of software engineering may be found in [13] “a project uses resources in performing processes to produce products for a customer.” It gives the model of Figure 1, centered on the software engineering project as the focal point for applying software engineering standards. This suggests a categorization of standards in four major areas: customer, process, product, and resource.

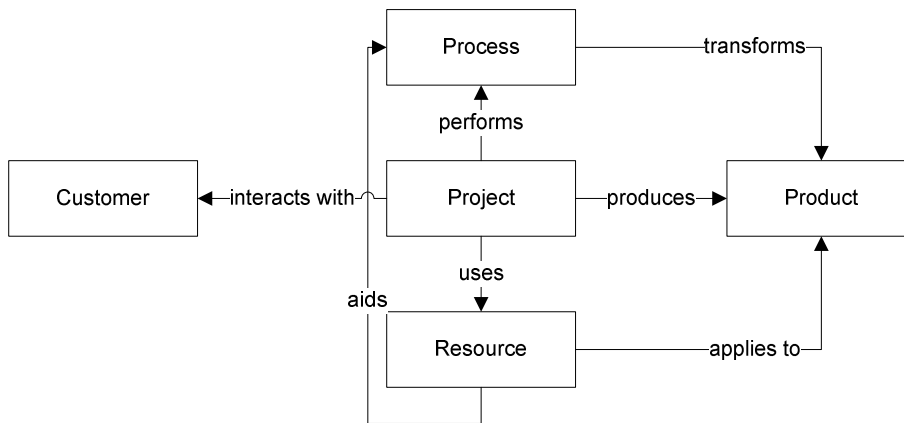


Fig. 1. The objects of software engineering, suggesting a categorization of standards in the subject areas of customer, process, product, and resource [13].

For VSEs, each category contains a number of standards that put them out of reach. There is a need for an umbrella standard within each category. The IEEE/IEC 12207, Software Life Cycle Processes [14], provides this umbrella for all of the customer and process standards. The on-going ISO/IEC 29110 standard [1] should provide such an umbrella for the customer, process, and products area.

3.2 ISO/IEC 29110

The ISO/IEC emerging standard “Software Engineering - Lifecycle Profiles for Very Small Enterprises (VSE)” - Basic Profile [11] contains 2 processes: Project Management (PM) and Software Implementation (SI). PM is subdivided in 4

activities (Project Planning, Project Plan Execution, Project Assessment and Control, Project Closure) and SI is subdivided in 6 activities (Software Implementation Initiation, Software Requirements Analysis, Software Architecture and Detailed Design, Software Construction, Software Integration and Tests Product Delivery).

The Basic Profile ([11], Clauses 4.2.8 and 4.3.8) proposes task decomposition of the PM and SI processes for each activity, together with inputs and outputs of each task. So, we can establish the workflow for each of the 22 work products (cf. §2.2). For instance, Figure 2 presents the workflow of Work Product 11, *Requirements Specification*.

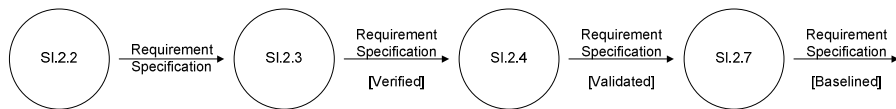


Fig. 2. WP11 Requirements Specification workflow.

3.3 Functional Requirements for Bibliographic Records

The Functional Requirements for Bibliographic Records (FRBR) is a conceptual model of the bibliographic universe, describing the entities in that universe, their attributes, and relationships among the entities [15]. The entities have been divided into three groups. The first group comprises the products of intellectual or artistic endeavor that are named or described in bibliographic records: *work*, *expression*, *manifestation*, and *item* [9]. The second group comprises those entities responsible for the intellectual or artistic content, the physical production and dissemination, or the custodianship of such products: *person* and *corporate body* [9]. The third group comprises an additional set of entities that serve as the subjects of intellectual or artistic endeavor: *concept*, *object*, *event*, and *place* [9].

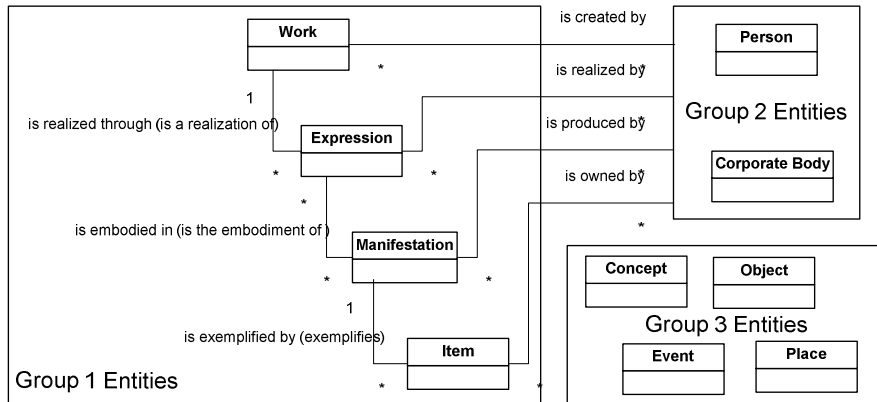


Fig. 3. FRBR entities and relationships.

Relationships depicted in the Figure 3 (a synthesis of Figure 3.1, 3.2, 3.3 [9]) indicate that a work may be realized through one or more than one expression. An expression, on the other hand, is the realization of one and only one work (there is a one-to-many relation linking work to expression). An expression may be embodied in one or more than one manifestation; likewise a manifestation may embody one or more than one expression (a many-to-many relation linking expression to manifestation). A manifestation, in turn, may be exemplified by one or more than one item; but an item may exemplify one and only one manifestation (a one-to-many relation linking manifestation to item).

Let us have an example. Starting with the work of the Software Requirements Specification of a project called XYZ, it was created by XYZ project manager and expressed in several ways including a current version, previous releases, a summary, etc. Once the expressions are recorded in some physical form - may be using a tool -, we have different manifestations such as the original text - as a part of the tool data -, and two electronic editions - one in PDF format and the other in HTML. We may have also several translations of any expressions, e.g. English and Spanish translations of the French original text, and a Spanish translation of a previous release. Those manifestations are related to the expression they are based on. At the item level - that's where we would see the specific copies held in a various places. An item would have attributes like its call number and the location where it is stored and any item specific notes, for example, that it is a customer-signed copy of the printed text of the French edition - would have a box linked to the paper manifestation.

As represented in Figure 3, there are relationships for the Group 2 entities with the Group 1 entities: a work is created by a person or corporate body; an expression is realized by a person or corporate body; a manifestation is produced by a person or corporate body; an item is owned by a person or corporate body.

3.4 Documentation identification scheme

An annotated system is a system which “knows about” its own content in order that automated tools can process annotations to improve use of the system. For example, semantic annotations can describe documents’ authors and their relationships, as well as including traditional metadata, such as the document subject and date of publication. Document (and more generally resource) identification is one of the main issues of an electronic management system; and especially if it is Web-based, Uniform Resource Identifiers (URIs) may be used as identifiers.

Identifying documents is not enough, because we have to keep the relationships between the different kinds of documents, for instance that three different documents are different translations of the current Software Requirements Specification. A Semantic Web-based management system is providing a way to define semantic links between resources e.g. that XYZ project manager created the Software Requirements Specification or that this resource is a translation of another one. Such assumptions are called semantic annotations and may be expressed using the Resource Description Framework (RDF) language. A semantic annotation in RDF is a triple <subject, property, object>. The subject is a URI; the object value can be a literal or a URI; the property is the meaning that is given to the link.

Time is vital for a VSE, and employees would not take the time to record semantic annotations linking resources if it can be done straightforwardly. An identification scheme may provide the VSE with documentation identification together with the implementation of Figure 3 relationships. Our main concern is to relate the different resources (abstract or physical) together.

Work identification. As noted in §2.2, the VSE has an enhanced list of document types (based initially on the 22 types provided with the ISO/IEC 29110). A trigram may be assigned to each type. Each project is identified with an acronym. In order to generate unique identifier, we can use the scheme year-number within the year; e.g. 2009-3 identifies the third document created by the VSE during the year 2009. Combining all these features together provide us with the work identification scheme. Each work is identified with a unique string based on the pattern:

Project - Document Type - Year - Number within a Year

For instance, the XYZ Software Requirements Specification is identified with XYZ-SRS-2009-3.

Expression identification. Work identifier serves a root for all expressions that realize the work (the main case is the different versions of this work). Versions - expressions - are abstract such as works are. You may reference a version that does not (and will never) have a physical existence. It is only the embodiment (the recording) of an expression (a version in our case) in or on some carrier that move from the abstract “work/expression” to a physical entity [15]. In the documentation managed by a VSE, there are two main types of expression: version and translation, that combine together (you may have Spanish and French translations of a version *n*, as well several versions of an on-going English translation of an important document).

Deciding whether versions or translations are expressions of a work or manifestations of an expression depends on VSEs business processes. Our proposition is to consider versions as expressions, and translations as manifestations. Hence, a version may be embodied in several languages, including the original language of the first manifestation. According to Figure 3, a work is realized through expression(s). Implicitly the expressions of the same work have a sibling relationship to each other [15]. Expressions are linked to the work they “realize” or express. The expression identification scheme relies on the work identification scheme. Each expression is identified with a unique string based on the pattern:

Project - Document Type - Year - Number within a Year - Version Number

For instance, the version A of the XYZ Software Requirements Specification is identified with XYZ-SRS-2009-3-A. It may be immediately deduced that this expression - version A - realizes the work XYZ-SRS-2009-3.

Manifestation identification. A manifestation is the physical embodiment of an expression of a work. In order to record something you have to put it on or in some container or carrier. As represented in Figure 3, a manifestation may be the physical embodiment of several expressions; typically when several documents are recorded on the same media records. As we are concerned with documentation management and not with configuration management, we would mostly not use this many-many relationship between expressions and manifestations. Usually, an expression (such as a version of a work) is embodied in several manifestations (such as different file formats or translations) and a manifestation, on the other hand, is the embodiment of one and only one expression. Implicitly the manifestations of the same expression have a sibling relationship to each other - that may be an equivalent content [15]. Expression identifier serves a root for all manifestations that embody the expression (main cases are the different translations of this expression, as well the different file formats that can be used for this embodiment). The manifestation identification scheme relies on the expression identification scheme. Each manifestation is identified with a unique string based on the pattern:

Project - Document Type - Year - Number within a Year - Version Number –
Language . Extension

For instance, the pdf file containing the English version A of the XYZ Software Requirements Specification is identified with XYZ-SRS-2009-3-A-eng.pdf. It may be immediately deduced that this manifestation - the English translation in a pdf file - embodies the expression XYZ-SRS-2009-3-A.

Short codes for language names may use a standard like ISO 639-3 (e.g. eng for English, deu for German). File name extensions (e.g. pdf for the Adobe file format) denote a particular way that information is encoded for storage in a computer file.

Item identification. An item is then a single exemplar of a manifestation – an individual copy. All copies that are linked to the same manifestation have a sibling relationship to each other [15]. Item management is related to configuration management, and is out of scope of documentation management (and of this paper).

3.5 File and meta-data management

Manifestations (files) can be uploaded into Semantic MediaWiki using the "Upload file" feature. Once a file is uploaded, other pages can include or link to the file. Uploaded files are given the "File:" prefix by the system, and this allows the files to be used in articles instantly. Every editable page on Wikipedia has an associated page history (sometimes called revision history or edit history). The page history contains a list of the page's previous revisions, including the date and time (in UTC) of each edit, the username or IP address of the user who made it, and their edit summary.

Each uploaded file has a file description page. The purpose of these pages is to provide information about the file, such as who uploaded the file, any modifications that may have been made, an extended description about the file's subject or context, where the file is used, and license or copyright information. All these information are metadata, and the description file is a metadata record.

Technically, there are two main solutions to manage metadata records, either to build an independent system or to add an extension to the resource management system itself. Our proposition belongs to the second type, while we are using a semantic wiki to manage metadata and the internal resource management system of the wiki to manage resources (documents). The file management system of a wiki is very straightforward but powerful, as long as we are able to identify documents. The identification scheme provides identification together with relationships management.

3.6 Information resource and non-information resource

One of the first steps towards the Semantic Web has been to use URI to identify "anything" and not only resources that can be located and accessed on the Web (some authors call them "*non-information resources*" [16]). One problem is related to know what the URI is identifying: the resource itself or the description (metadata record) of the resource. In a (semantic) wiki, access to an information resource is provided through its description page (mentioned above) identified with the same name as the resource itself. We extend these principles to "non-information resource". A page named with the name of a "non-information resource" is a description of the "non-information resource", not the "non-information resource" itself.

As well we use this URI internally to the wiki, it works fine. But if we wish to generate and export correct triples providing assumptions to the real "non-information resource", we have to dispose of the right URI of this "non-information resource".

FRBR sorts entities in three groups. Group 1 entities represent the different aspects of user interests in the products of intellectual or artistic endeavor: work, expression, manifestation, and item [9]; all are information resources. Group 2 entities include person (an individual) and corporate body (an organization or group of individuals and/or organizations) [9]; all are "non-information resource". Group 3 entities include concept (an abstract notion or idea), object (a material thing), event (an action or occurrence), and place (a location) [9]; all examples of "non-information resource".

3.7 Information resource catalogue

In the section 4 of [17], IFLA define that an Online Public Access Catalogue should be an effective and efficient instrument that enables a user:

- 4.1. to find bibliographic resources in a collection as the result of a search using attributes or relationships of the resources
- 4.2. to identify a bibliographic resource or agent
- 4.3. to select a bibliographic resource that is appropriate to the user's needs
- 4.4. to acquire or obtain access to an item described; or to access, acquire, or obtain authority data or bibliographic data
- 4.5. to navigate within a catalogue and beyond

The straightforward documentation system proposed in this paper has to provide, from its users point a view, a resource catalogue, with the tasks in the list above. So what are these user tasks? Briefly, they are find, identify, select, obtain, and navigate.

'Find' involves meeting a user's search criteria through an attribute or a relationship of an entity. Semantic MediaWiki includes an easy-to-use query language which enables users to access the wiki's knowledge. The syntax of this query language is similar to the syntax of annotations in Semantic MediaWiki.

'Identify' enables a user to confirm they have found what they looked for, distinguishing among similar resources. The identification scheme, based on the FRBR Group 1 entities [9] and presented in section 3.4 is used to confirm that the described entity corresponds to the entity sought or to distinguish between two or more entities with similar characteristics.

'Select' involves meeting a user's requirements with respect to content, physical format, etc. or to reject an entity that doesn't meet the user's needs. 'Select' is strongly related to search capabilities. The first kind of search exploits properties used as annotations. Search criteria can be combined through Boolean operators. Searches can use taxonomies, based on the categories of the wiki.

'Obtain' enables a user to acquire an entity through electronic remote access. The internal file management system let an immediate access to the current version of the resource, and its history as well. The history page let users to see all past changes to the page in question, to view a specific version, to compare two specific versions, etc.

FRBR recognizes the importance of being able to 'navigate'. Semantic MediaWiki provides a simple browsing interface that displays all semantic properties of a page, as well as all semantic links that point to that page. By clicking on these links, the user can browse to another article. Faceted classification provides a way to design hierarchies which are simpler and more lightweight. The extension Semantic Drill Down (http://www.mediawiki.org/wiki/Extension:Semantic_Drilldown) provides users with a hierarchical faceted navigation of categories through semantic properties.

The tasks listed above require that everybody records resources and metadata record. The latter information should be accurate in order to support documentation management. In a small project, this verification task will be devoted to the Project Manager. As mentioned in section 2.2, the workflow of each product is defined in the ISO/IEC 29110 standard. Our proposal is to review and update the metadata

associated with a work product when performing the last activity that outputs the final version of this work product. Conforming to the example presented in Figure 2, WP11 Requirements Specification metadata will be reviewed and updated during the ‘SI.2 SW Requirements Analysis’, while, for instance, WP17 Software User Documentation metadata will be updated during the ‘SI.5 SW Integration and Tests’.

Semantic templates are a straightforward tool that Semantic MediaWiki offers to record annotations (metadata). Users specify annotations without learning any new syntax; annotations are used consistently, i.e. users do not have to look for the right properties or categories when editing a page; templates provide data structure, by defining which values belong in which pages. Semantic forms allows for the creation of forms built on semantic template that often provide a nice display and input.

4 Conclusion

We proposed to use a semantic wiki for documentation management in Very Small Enterprises (VSEs). The FRBR proposition [9] includes a description of the conceptual model (the entities, relationships, and attributes), a four-level classification for all types of resources, and user tasks associated with the bibliographic resources described in catalogs. We used the four-level classification to define a document identification scheme that allows documents to be managed by the internal resource management of the semantic wiki, hence benefiting from a straightforward but powerful version control. With few inputs of semantic annotations by VSE employees - through usable semantic forms and templates, the semantic wiki acts as a library catalog, and users can find, identify, select, obtain, and navigate resources. The next step is to implement this proposition in a pilot project. Two VSEs may be interested and will provide us with a feedback on our proposition.

References

1. International Organization for Standardization (ISO): ISO/IEC DTR 29110-1 Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) -- Part 1: Overview. ISO, Geneva (2010)
2. Chan, I., Chao, C.: Knowledge management in small and medium-sized enterprises. *Communications of the ACM*, vol. 51 (4), pp. 83--88 (2008)
3. García Alonso, J. M., Berrocal Olmeda, J. J., Murillo Rodríguez, J. M.: Documentation Center - Simplifying the Documentation of Software Projects. In: Wiki4SE Workshop - 4th International Symposium on Wikis. <http://www.wikis4se.org/doku.php/> (2008)
4. Maxwell, J. W.: Using Wiki as a Multi-Mode Publishing Platform. In: 25th annual ACM international conference on Design of communication, pp. 196--200. ACM, New York (2001)
5. Rauschmayer, A.: Next-Generation Wikis: What Users Expect; How RDF Helps. In: Third Semantic Wiki Workshop. ESWC, Redaktion Sun SITE, Aachen, poster (2009)
6. Kahan, J. and al.: Annotea: an open RDF infrastructure for shared Web annotations. In: 10th Int. Conference on World Wide Web (WWW'01), pp. 623--632. ACM, New York (2001)

7. Handschuh, S. and Staab, S.: Authoring and annotation of web pages in CREAM. In: 11th Int. Conference on World Wide Web (WWW '02), pp. 462--473. ACM, New York (2002)
8. Rech, J., Bogner, C., Haas, V.: Using Wikis to Tackle Reuse in Software Projects. IEEE Software 24 (6), pp. 99-104 (2007)
9. IFLA : FUNCTIONAL REQUIREMENTS FOR BIBLIOGRAPHIC RECORDS, <http://www.ifla.org/VII/s13/frbr/> (2009)
10. Ramadour P., Cauvet C.: An Ontology-based Support for Asset Design and Reuse. In: ENC '08 Mexican International Conference on Computer Science, Mexico, pp. 20--32. IEEE, Los Alamitos, CA (2008)
11. International Organization for Standardization (ISO): ISO/IEC DTR 29110-5-1-2 Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) -- Part 5: Management and Engineering Guide-Basic VSE Profile. ISO, Geneva (2010)
12. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web, vol. 4 (1) pp. 14-28 (2006)
13. Moore, J.W.: An integrated collection of software engineering standards. IEEE Software, vol. 16 (6), pp. 51-57 (1999)
14. International Organization for Standardization (ISO): ISO/IEC 12207:2008 Information technology -- Software life cycle processes. ISO, Geneva (2008)
15. Tillett, B.: The FRBR Model. FRBR Seminar Australian Committee on Cataloguing, Sydney (2004)
16. Booth, D.: URIs and the Myth of Identity. In Workshop on Identity, Reference, and the Web, WWW2006, Edinburgh, Scotland (2006)
17. IFLA: STATEMENT OF INTERNATIONAL CATALOGUING PRINCIPLES http://www.ifla.org/files/cataloguing/icp/icp_2009-en.pdf (2009)