A Catalogue of Functional Software Requirement Patterns for the Domain of Content Management Systems

C. Palomares, C. Quer, X. Franch
Universitat Politécnica de Catalunya (UPC)
Barcelona, Catalunya, Spain
{cpalomares, cquer, franch}@essi.upc.edu

S. Renault, C. Guerlain
Public Research Centre Henri Tudor
Luxembourg, Luxembourg
{samuel.renault, cindy.guerlain}@tudor.lu

ABSTRACT
Software requirement patterns have been proposed as an artifact for fostering requirements reuse. When we define these patterns for the functional part of a software system, we realize that most of patterns are specific of a software domain. This paper presents and analyzes a catalogue of functional software requirement patterns for the domain of content management, and gives an overview of how this catalogue has been constructed from the systematic analysis of 6 existing software specification documents with the support of expert assessment.

Categories and Subject Descriptors
D.2.1 [Requirements/Specifications]: Elicitation methods, Languages, Methodologies. D.2.7 [Distribution, Maintenance, and Enhancement]: Documentation.

General Terms
Management, Documentation.

Keywords
Software requirement patterns, Requirements reuse, Functional requirements.

1. INTRODUCTION
Requirements that appear in Software Requirement Specification (SRS) documents can be classified into three categories:

- Functional requirements (F-reqs) that establish the observable behavior that must exhibit the system (calculations, manipulations, listings, evolution aspects, etc.), as well as the data types specification [1].
- Non-Functional requirements (NF-reqs) that determine the criteria or global qualities of the software system and set restrictions (internal and external) on the software and the development process [2].
- Non-Technical requirements (NT-reqs) that do not refer directly to the intrinsic quality of software, but to the context of the system under analysis; they include economic, political and managerial issues [3].

SAC'13, March 18-22, 2013, Coimbra, Portugal.

In our recent research, we have had the chance to analyze a number of SRS documents regarding software procurement projects. We have observed that a big percentage of NF-reqs and NT-reqs therein are domain-independent (i.e., they appear basically in the same way in different documents, even if belonging to projects from different domains), whilst in the case of F-reqs it is quite the opposite. This observation aligns with our research in other related fields, e.g. about software quality models [4] and with the preliminary results of a systematic literature review focused on reuse techniques in requirements engineering that we are currently completing. In this review, the approaches we have found that address reuse of the requirements specification functional part, propose reuse artifacts that are mostly specific for a given software category or domain. This is the case of Liu et al.'s work [5], that focuses in the reuse of requirements for the domain of networked software; and Konrad and Cheng's [6] that proposes reuse of requirement for the domain of embedded systems.

In this paper we present and analyze a catalogue of Functional Software Requirement Patterns (F-SRP) constructed in the context of the PABRE framework [7] after overviewing the method to construct it. Specifically the F-SRP in the catalogue address the domain of Content Management Systems (CMS). An SRP [8] basically consists of: a template that generates one or more requirements; some information to identify its adequacy to a particular project; and how it may be tailored to that project. The main benefits of using SRPs may be summarized as: 1) more effective requirement elicitation (requirements are not built from scratch; a process guides the engineer by giving recommendations, suggesting information, etc.); 2) improved quality and consistency of requirements documents (by using a uniform style); 3) improved requirements management (e.g., clear traceability from requirements to goals). The presented F-SRP catalogue is an extension of our current SRP catalogue which includes 29 Non-Functional SRP [7] and 37 Non-Technical SRP [11].

The structure of the paper is the following. After describing the PABRE framework (Section II), we introduce the steps followed to construct the F-SRP catalogue and present its general structure (Section III), and, as example, we describe in detail one F-SRP in the catalogue (Section IV). Next, we provide the analysis of the catalogue (Section V) and finally we elaborate some conclusions and future work (Section VI).

2. THE PABRE FRAMEWORK
The PABRE framework was born from our collaboration between the Public Research Center Henri Tudor (TUDOR) at Luxembourg and the Software Engineering for Information Systems research group (GESSI).

TUDOR works with a business network of freelance independent federated consultants, referred as CASSIS, and trains them to innovative methods produced by research projects for their use in industrial contexts. One of the processes whose improvement has been targeted in the CASSIS consultants work was the introduction of requirements reuse. After different, but unsuccessful, approaches to reuse requirements (reusing fragments of a former SRS document resulting from a similar project as a basis to build a new SRS; reusing a generic SRS document with "holes" to fill), TUDOR began the collaboration with GESSI and our PABRE framework [7] arose.

Currently the framework includes:

- An SRP catalogue with 29 Non-Functional SRP [7] and 37 Non-Technical SRP [9]. These two catalogues are generic and apply to every kind of software domain (although every domain may have some particular NF-reqs and NT-reqs).
- A metamodel that describes the structure of SRP and the catalogue [8].
- A method for the use of the catalogue in the requirements engineering stage [7].
- A system for the catalogue use, management and evolution [10].

It is not the aim of this paper to explain neither the PABRE SRP metamodel nor the method for using it in the requirements elicitation process.

3. THE CATALOGUE

The catalogue presented in this paper corresponds to the initial version of an F-SRP catalogue for the domain of CMS. The catalogue was constructed starting from 6 SRS documents composed of circa 210 F-reqs in average and that were generated in the corresponding CMS IT procurement projects.

The PABRE process applied to construct these F-SRP consists on the following steps (more details can be found in [7]):

1. Filtering. The requirements of the different SRS were classified according to their type (F, NF or NT). Requirements of type NF-reqs and NT-reqs were discarded due to the purposes of our work.
2. Alignment. The F-reqs of the different SRS were consolidated and aligned according to the functional aspect they refer. This step is needed in order to cluster the requirements for pattern identification purposes.
3. Analysis. For each of these functional aspects, a study of their adequacy as an SRP was performed. The main criterion of course was repetition, i.e. how many of the SRS documents included the aspect, because it seems to indicate high probability of reuse, but also expert assessment was required since an F-req appearing in a few, even just one, SRS may be considered adequate as SRP. During the analysis, a glossary of terms and metrics was built.
4. Formulation. The selected SRP candidates were converted into SRP. Not every candidate was necessarily converted into a different SRP, since some of them were considered close enough as to be integrated in the same pattern by an abstraction process. As a result, a final set of 45 F-SRP was built. The integration of SRP-candidates during this formulation causes the difference among the number of requirements in the books and the number of resulting SRP.
5. Catalogue construction. Finally, the SRP were classified according to some existing classification schema. ISO/IEC 9126 in our case, resulting in a two-level hierarchy containing 14 classifiers in the first level, and 24 in the second level. Also, relationships among SRP were recorded. It is worth to mention that this last step, in fact, started in parallel to Step 2, since when aligning requirements, their classification naturally emerges.

The result of the catalogue construction after the process above is shown in Table 1. The first two columns present the classification schema. As commented above, we are adopting as framework the ISO/IEC 9126-1 quality standard [11]. The standard includes six characteristics (higher-level quality factors), being *Functionality* one of them. Inside *Functionality*, the main subcharacteristic is *Suitability*, defined as the adequacy of a software product regarding its functionality. Therefore, this is the natural place to group the classifiers for CMS functionality. As starting point for the decomposition of *Suitability,* we used the ISO/IEC-9126-1-based quality model for the CMS domain that we proposed in [12]. Over the result, we implemented some changes in order to accommodate some of the F-SRP. First, it was necessary to add four new subcharacteristics as direct children of *Suitability*, regarding the management of media, agenda, storage and text input. Second, it was necessary to add a new subcharacteristic *Content Verifiabiliness* as child of the existing *Content* subcharacteristic. All this new subcharacteristics are marked with the word "new" in Table 1.

Concerning the patterns themselves, which appear in the third column of Table 1, we remark that the PABRE framework allows, if it makes sense, the classification of the same SRP in more than one subcharacteristic. This situation appeared three times in our F-SRP catalogue: in the *Assets to import/export* and in the *Import/Export feature*s SRP, since they can define constraints over both *Contents management* and *Mail files management* subcharacteristics, and in the *E-mail notifications* SRP, since it can define constraints over both *Agenda management* and *Mail files management* subcharacteristics. These duplicated SRP are highlighted with the term "dup" in Table 1.

It is worth to mention that, according to the construction process of the PABRE SRP catalogues applied to NF-reqs and NT-reqs as described in [7], two iterations for reviewing the catalogue are required in order to improve the quality of F-SRP and arrive to a more stable catalogue.

The first iteration has the goal of making the patterns' granularity uniform. The main task in this iteration corresponds to merge patterns that have similar goals (i.e., solve the same problem) or that restrict the same functionality. Conversely, SRP that try to give solution to more than one problem or that restrict different functionalities have to be identified and split into several SRP.

A second iteration is also necessary to align the contents of the SRP in two directions. First, ensuring the consistent use of the glossary of terms and metrics built during Step 3. Second, checking some predefined grammatical rules on the patterns text and taking corrective actions in case those deviations are found.

As a side effect of these two iterations, it is expected that some slight changes on the classification schema and the dependencies are implemented.

4. AN EXAMPLE OF F-SRP

We present here in detail the *Content Version Management* F-SRP contained in the catalogue for CMS. The general aspects of the pattern are summarized in Table 2 according to the template proposed for the Requirement Patterns workshop (http://www.utdallas.edu/~supakkul/repa12/submission.html) and the detailed solution is included in Table 3 according to the PABRE framework metamodel [8].

4.1 General Aspects

This SRP, as the rest of SRP in the catalogue, is suitable for IT procurement projects. More precisely, it is useful during the elicitation and specification of requirements (see *Known Uses* and *Context* in Table 2). Its application is interesting when the *Problem* arises, that is, when a customer wants to state the functionalities of a CMS related with version management, that is, when a customer requires version management in the CMS she wants to acquire. The *Solutions* identified may vary depending on whether it is necessary or not to distinguish among the version management needed for different content types (*Forces*).

Once the pattern is applied, the Related Patterns or Sibling patterns in the catalogue are natural candidates to be applied [7] (*Catalo-guing* step in Section 3). In our case, we find as related pattern the Content Management F-SRP. The idea behind this dependency is that it is not possible to state requirements about versioning of content, if the requirements about the contents are not established. The second related SRP is History Features, since keeping track of the events occurred is needed to provide versioning. On the other hand, the sibling SRP are the ones belonging to the same subcharacteristic, i.e. Content verifiabiliness (see Table 1).

4.2 Detailed Solution

The Content Version Management Goal (see Table 3), represents the solution of the Problem stated in the general description of the SRP. An SRP consists of several Forms, each one representing a different solution for achieving the goal. In the Content Version Management SRP, its goal can be attained by having version management over all contents stored in the system (General Version Management form), or by having version management over specific contents stored in the system (Specific Version Management form).

The *Forms* are organized into *Parts*, each of them being a template. Each *Form* is characterized by a *Fixed Part* which states the minimal requirement that always holds when applying that form, and some *Extended Parts* which may be applied or not. The *Fixed Part* always becomes a requirement when an SRP is applied with this *Form*. *Extended Parts* are only used if more precise information is required in the specification. Due to this nature, the *Fixed Part* is usually quite generic and hardly measurable. For instance, in the *Specific Version Management* form, the application of the *Fixed Part* will be included in the SRS, expressing a generic requirement for having version management over specific contents. Also it will be possible to apply one or more of the four *Extended Part*s expressing specific requirements on: which shall be the contents on which versioning will be implemented (*Versioned Contents*); which shall be the contents on which automatic versioning will be implemented (*Specific Automatic Versions*); which shall be the contents on which to allow the retrieval of old versions (*Specific Version Retrieval*), and manual saving of contents as a previous version (*Savings with Previous Versions*).

Table 1. The Catalogue of F-SRP and their Classification under the ISO-IEC 9126-1 Suitability Subcharacteristic

FUNCTIONALITY characteristic, SUITABILITY subcharacteristic		
1. Users	*Users management*	----------
	Users actions	----------
2. Roles	*Roles management*	----------
3. Groups	*Groups management*	----------
4. Content	*Contents management*	• Content management • Content customization • Content annotation • Content preview • Content lifetime • Broadcast modes • Broadcast features • Automatic broadcast • Manual broadcast • Assets to import/export (dup) • Import/export features (dup)
	Contents security	----------
	Contents veriafibiliness (new)	• Content version management • History features • Annotation history • Broadcasting history • Access history • Workflow history
5. Folders	*Folders management*	• URL features
	Folders security	----------
6. Alias	*Alias management*	• External content references • Hyperlinks management
	Alias security	----------
7. Query	*Searches*	• Indexing • Content search
	Search API	----------
	Searches security	----------
8. Lifecycle	*Lifecycle management*	• Workflows management • Publication workflow • Pre-publication actions • Post-publication actions • Content validation
	Lifecycle security	----------
8. Mail	*Mail files management*	• E-mail notifications (dup) • Assets to import/export (dup) • Import/export features (dup)
	Mail security	----------
10. Web contents	*Web contents management*	• Website navigation features • Website features • Websites management • Website use
	Web contents security	----------
11. Media (new)	*Media management* (new)	• Marketing operations • Newsletters • Electronic payment
12. Agenda (new)	*Agenda management* (new)	• Contacts management • Incoming faxes management • E-mail notifications (dup) • To-do list
13. Storage (new)	*Storage management* (new)	• Storage features • Storage compression • Automatic storage • Storage rights
14. Text input (new)	*Text input management* (new)	• Editor • Spell checker • Formularies features

Table 2. Content Version Management pattern summary

CONTENT VERSION MANAGEMENT PATTERN SUMMARY	
Name	Content Version Management
Authors	The authors of this paper
Context	**RE Activity** Elicitation, Specification
	Pattern Type Product
	Business Domains Content Management System
	Organizational Environment Factors Customer organization that needs to acquire a software system for managing some types of contents. Customer organization that is interested in managing versions of the data in the different contents they manage.
	Stakeholders Customer, Supplier, Customer Data Administrator, Customer Maintenance Team
Problem	Allowing the customer having a content version management, in case is necessary to retrieve a previous version of a content.
Forces	It is necessary to distinguish among different types of versioning depending on the type of content or not.
Solution	The detailed solution is described in Table 3
Application	The whole process is described in [7] • Browse the pattern, • Check if the goal is relevant for the context, • Choose the most appropriate form, • Extract the fixed part, • Check and extract the most relevant extended parts taking into account the constraints • Choose the parameters values taking into account the constraints
Known Uses	• IT Procurement Projects
Cataloguing	**Classification** Functionality: Suitability: Content: Content Verifiabiliness
	Related Patterns • Content Management • History Features

Usually, fixed and extended parts must conform to some *Part Constraint* represented by means of a regular expression that may involve some predefined operators (e.g., for declaring multiplicities or dependencies among parts, as *Excludes* and *Requires*). In our SRP example, both the first and the second form have *Fixed* and *Extended Parts Constraints*. Specifically these constraints state that the extended parts must be used at most once (0..1) in one project if that form is chosen. On the other hand, the second form has *Consistency Rules* that establish some dependencies among the parts. Specifically, the extended part *Versioned Contents* must be used previously to the extended parts *Specific Automatic Versions* and *Specific Version Retrieval*.

From a syntactic point of view, both fixed and extended parts are similar. They are composed by the text to be used as a requirement and optionally some *Parameters* to be instantiated when applying the pattern. Parameters have a *Metric* that states the valid values that may take the parameter in applying the pattern. For instance, three of the extended parts of the second form of the *Content Version Management* SRP allow stating prerequisites on content types for which it is desired to define versioning. The parameter is in all the cases *ContentTypes*, whose metric declares that it may have as values the different types of contents managed by a CMS. Sometimes it is necessary to define constraints that establish relationships among the values of different parameters. For instance, the two last consistency rules state that the values used for the parameter of the extended parts *Specific Automatic Versions* and *Specific Version Retrieval* must be a subset of the values of the parameter present in the extended part *Versioned Contents.*

5. ANALYSIS

In this section, we analyze the F-SRP catalogue for CMS from different points of view: 1) SRS coverage: how many of the F-reqs that appear in the 6 SRS documents can be obtained as application of the F-SRP; 2) Classification schema coverage: how many of the classifiers identified in the classification schema have F-SRP bound; 3) Generality: how many of the F-SRP can be applied to other software domains.

Regarding the coverage of the 6 SRS documents used as departing point of the catalogue, we tried to produce them again starting from the constructed catalogue. The result is that we could reproduce 87% of the original F-reqs (i.e., about 183 requirements per SRS document) as a result of application of the F-SRP, which is a very high reuse rate. Due to the use of the F-SRP for doing this reconstruction, the text of the requirements improved, due to changes introduced during the formulation of the F-SRP (to improve their uniformity, quality and consistency). Despite these changes, the underlying restrictions of the requirements remained the same. As an example, in one of the SRS documents we found the requirement *"Before the physical storage of electronic files in the storage structure, the solution must propose to rename these files from the information contained in the indexing criteria"*, stating the need of proposing a better naming for an electronic file before storing it. During our reconstruction, we used to reproduce this requirement the Extended Part Renaming of the Form Manual Storage of the *Storage Features* SRP, achieving the following requirement *"The system shall propose to rename electronic files using* ***information contained in the indexing criteria*** *before physically storing them"* (value of parameters in bold). Although both requirements express the same constraint, the second one states it in a more direct way, easier to understand, and following the uniformization rules that SRP provide (i.e. the use of shall as modal verb in all the requirements).

The 13% of requirements left were too specific to the system itself to become a pattern, and that was the reason why we jointly with requirement experts decided to not include them as SRP. Some of these requirements dealt about the information present in the documents generated by the system, such as *"Users must identify, from a printed document, if it is stored or not in the content management system"* or about interface details, as *"It will be possible to the system to simulate a tablet or laptop screen visualization"*.

Regarding the coverage of the used classification schema (see Table 1), the first level of subcharacteristics already present in the initial classification schema (see [11]) were the ones from row 1 to row 10. We have formulated F-SRP for 7 of those subcharacteristics. The 3 subcharacteristics without patterns are:

Table 3. Functional SRP: Content Version Management

<table>
<tr><th colspan="4">CONTENT VERSION MANAGEMENT</th></tr>
<tr><td colspan="4">Goal: Stating the functionalities of version management</td></tr>
<tr><td colspan="4">Description: This pattern expresses the need of having a system that manages versions of contents.</td></tr>
<tr><td colspan="4">Keywords: Version, Management, Automatic versioning, Version retrieval</td></tr>
<tr><td rowspan="5">Requirement Form

General Version Management</td><td>Description</td><td colspan="2">This form establishes the need of having a version management over all contents stored in the system.</td></tr>
<tr><td>Constraints</td><td colspan="2">Fixed part (1)
Extended parts
Automatic Versions (0..1)
Version Retrieval (0..1)
Savings with Previous Versions (0..1)</td></tr>
<tr><td>Fixed Part</td><td>Form Text</td><td>The system shall manage version of the stored contents.</td></tr>
<tr><td>Extended Part
Automatic Versions</td><td>Form text</td><td>The system shall propose automatically the creation of new versions depending on the changes done in the content.</td></tr>
<tr><td>Extended Part
Version Retrieval</td><td>Form text</td><td>The system shall allow the retrieval of a previous version of a stored content.</td></tr>
<tr><td rowspan="13">Requirement Form

Specific Version Management</td><td>Description</td><td colspan="2">This form establishes the need of having a version management over specific contents stored in the system.</td></tr>
<tr><td>Constraints</td><td colspan="2">Fixed part (1)
Extended parts
Versioned Contents (0..1)
Specific Automatic Versions (0..1)
Specific Version Retrieval (0..1)
Savings with Previous Versions (0..1)
Consistency rules
Versioned Contents must be applied before Specific Automatic Version or Specific Version Retrieval.
contentTypes (Specific Automatic Versions) subset of contentType(Versioned Contents)
contentTypes (Specific Version Retrieval) subset of contentType(Versioned Contents)</td></tr>
<tr><td>Fixed Part</td><td>Form Text</td><td>The system shall manage versions over specific stored contents.</td></tr>
<tr><td rowspan="3">Extended Part

Versioned Contents</td><td>Form text</td><td>The system shall do versioning over contentTypes contents.</td></tr>
<tr><td>Param</td><td>Metric</td></tr>
<tr><td>contentTypes is a non-empty set of content types</td><td>ContentTypes = Set (ContentType)
ContentType = Domain (publications, reports, base documents, etc.)</td></tr>
<tr><td rowspan="3">Extended Part

Specific Automatic Versions</td><td>Form text</td><td>The system shall propose automatically the creation of new versions over contentTypes depending on the changes done in the content.</td></tr>
<tr><td>Param</td><td>Metric</td></tr>
<tr><td>contentTypes as above</td><td>ContentTypes as above</td></tr>
<tr><td rowspan="3">Extended Part

Specific Version Retrieval</td><td>Form text</td><td>The system shall allow the retrieval of a previous version of the stored contentTypes contents.</td></tr>
<tr><td>Param</td><td>Metric</td></tr>
<tr><td>contentTypes as above</td><td>contentTypes as above</td></tr>
<tr><td>Extended Part

Savings with Previous Versions</td><td>Form text</td><td>The system shall not allow saving a new version of content as it was a previous one.</td></tr>
</table>

Users, *Roles* and *Groups*. The reason is that in the SRS documents used for the construction of SRP, the requirements associated to these subcharacteristics were included as NF-reqs, assuming that they were related with security. In the case of the empty second level subcharacteristics (*Folders Security*, *Mail Security*, etc.) no specific requirements appeared in the SRS documents related with them, but if they were, probably they would have been classified as NF-reqs in the SRS. Under this view, taking into account that the F-reqs in the SRS documents do not include security requirements, we may consider that the coverage of 70% of the classification schema is, in fact, a real coverage of 100%.

Concerning the five subcharacteristics added to the initial proposal, our analysis recognizes two situations. On the one hand, subcharacteristics that correspond to functionality and are not identified in [12]. These subcharacteristics are *Contents Verifiabiliness*, *Media Management* and *Agenda Management.* On the other hand, subcharacteristics which provide some functionality of an CMS that operationalizes some non-functional aspect, therefore they could be classified in both parts of the catalogue. They are *Storage*, that can be also classified as *Efficiency/Resources Utilization* pattern under the ISO 9126-1 [11] and *Text Input* as *Usability/Operability* patterns.

Finally, regarding to the percentage of SRP that are specific of the CMS software domain, the analysis of the catalogue shows that just the SRP in the subcharacteristics *Storage* and *Text input* could be shared with any software domain, since they state functionalities that can be interesting in any software. There are other subcharacteristics that are common to categories where CMS belongs. For instance, SRP in the *Lifecycle* subcharacteristic can be shared with all domains where a workflow process is required. This happens also in other four subcharacteristics which are *Mail*, *Web Contents*, *Media* and *Agenda*. Thus, we can say that just 15% of the SRP are general for any software domain, and the 31% can be shared with other domains of some common software category.

6. CONCLUSIONS

In this paper we presented a catalogue of F-SRP for the domain of CMS, which is constructed by following our PABRE framework. This catalogue is composed of 45 SRP. We have exposed the different SRP organized through a previous existent functionality classification for the CMS software domain [11], and we presented one SRP example. Finally we have done an analysis of the catalogue. This analysis leads to the following conclusions:

- Starting from a set of SRS documents for a certain software domain, it is possible to arrive to an F-SRP catalogue that covers most of the initial F-reqs, and that provides patterns for obtaining requirements for the different functionalities of the domain.
- Using an F-SRP catalogue with stated constraints for the application of the patterns, aligned with an stated glossary and that follows some stated grammatical rules, the quality of SRS documents obtained during elicitation processes can be improved.
- Using an F-SRP catalogue, the SRS documents obtained will follow a specific classification of the requirements that will be previously agreed for the experts that construct the catalogue. In the cases an SRP is classified in more than one place of the schema, the analyst that will do the elicitation will be able to decide where to classify the specific requirements in the SRS.
- The argument that most of the F-reqs are specific of the domain of the system under analysis continues being valid.

As future work, first of all we want to validate the constructed F-SRP with regard to SRS different from those used to construct it, trying to produce them again starting from the catalogue. We also are interested in studying and finding approaches for facilitating the patterns usability and understandability from a requirement analyst point of view. Finally, we want to construct F-SRP for different domains from the one presented in this paper and integrate F-, NF- and NT-SRP catalogues into a single catalogue and validate it in real elicitation processes.

7. ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish project TIN-2010-19130-C02-01.

8. REFERENCES

[1] I. Sommerville, "Software Engineering 8", 8th edition, International Computer Science Series, 2006.

[2] L. Chung, J.C. Sampaio do Prado Leite, "On Non-Functional Requirements in Software Engineering", Conceptual Modeling: Foundations and Applications, Springer, 2009.

[3] J. P. Carvallo, X. Franch, C. Quer: Managing Non-Technical Requirements in COTS Components Selection. RE 2006.

[4] J.P. Carvallo, X. Franch, C. Quer, "Determining Criteria for Selecting Software Components: Lessons Learned". IEEE Software, vol.24(3), 2007.

[5] W. Liu, K. He, K. Zhang, J. Wang, "Combining Domain-Driven Approach with Requirement Assets for Networked Software Requirements Elicitation", IEEE International Conference on Semantic Computing (ICSC), 2008.

[6] S. Konrad, B. H. C. Cheng, "Requirements Patterns for Embedded Systems", IEEE Joint International Conference on Requirements Engineering (RE), 2002.

[7] S. Renault, O. Mendez-Bonilla, X. Franch, C. Quer, "A Pattern-based Method for building Requirements Documents in Call-for-tender Processes". International Journal of Computer Science & Applications (IJCSA), vol.6(5) , 2009.

[8] X. Franch, C. Palomares, C. Quer, S. Renault, F. De Lazzer, "A Metamodel for Software Requirement Patterns", 16th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2010.

[9] C. Palomares, C. Quer, X. Franch, C. Guerlain and S. Renault, "A Catalogue of Non-Technical Requirement Patterns", Second International Workshop on Requirements Patterns (REPA), 2012.

[10] C. Palomares, C.Quer, X.Franch, "PABRE-Man: Management of a Requirement Patterns Catalogue", Tool demo at 19th IEEE International Requirements Engineering Conference (RE), 2011.

[11] ISO/IEC Standard 9126-1, "Software Engineering – Product Quality –Part 1: Quality Model", 2001.

[12] X. Franch, C. Quer, J.A. Cantón, R. Salietti, "Experience Report on the Construction of Quality Models for Some Content Management Software Domains", 7th International Conference on Composition-Based Software Systems (ICCBSS), 2008.