Categorization-based Concept in Requirements Engineering Process

Tatiana Chuprina
Model-based System Engineering
Fortiss GmbH
Munich, Germany
email: chuprina@fortiss.org

***Abstract*—In this paper Categorization-based concept (CBC) is presented, that proposes a solution for handling unstructured textual system requirements. The CBC offers a method for requirements elaboration and for their systematic review. We claim, that Categorization-based concept provides a common instrument for all roles of system development process exposing a new Artifact. Thus, the research of CBC raises the scientific questions regarding quality of system requirements.**

Furthermore, this article considers an implementation of Categorization-based concept in AUTOFOCUS 3 (AF3) development tool [1] as a framework called "*aspects*". Here *aspects* represent requirements' categories, which apply to requirements. They constitute the templates inferred from requirements facets.

I. INTRODUCTION

According to information from a review of requirement engineering issues [2], challenges regarding requirements quality, communication gap between teams in development process, etc. bring about plenty impediments into system development and may result in great increase of resources from industry. Moreover, a research of industry needs [3] provides a statistics of failure projects also causing by the mentioned problems among another.

In this paper we provide a description of Categorization-based concept (CBC), that suggests a solution for handling the problems with concern to the quality of system requirements. CBC brings textual unstructured requirements to a consistent view categorizing them by their characteristics. Moreover, CBC supports a method for transforming requirements into less complex composition and for reviewing them in a more precise way. By that means,Categorization-based concept improves qualitative aspect of requirements.

This concept embraces general requirements categorization advocated in the scientific society and includes experience gathered by the industry in software-intensive systems development. On practice, incoming document for system developers can be system requirements in the form of text (use cases, specifications with detailed lines of description or bulleted list, etc.), which are usually poor structured and error-prone. The list of most popular requirements management tools in the industry community include DOORS [4], Requirement Quality Suite (RQS) [5], BTC. Alternatively, to bring the requirements into formalized view, system designers can apply patterns by using languages such as EARS [6] or UML [7] etc.

In contrary, the Categorization-based concept categorization-based provides a method with noticeable degree of freedom from any particular tools for requirements engineering and reviewing process. It allows to work with unstructured or unorganized textual requirements by adhering incrementally categories to them.

Categorization-based concept has been incarnated as a framework within AUTOFOCUS 3 and named *aspects* . Here *aspects* represent requirements categories with regards to their characteristics. In AF3 the *aspects* have been implemented as templates supporting main facets of the requirements. For instance, *Signal aspect* categories the requirement, which provides a signal definition. This *aspect* captures information about the signal type and range in the template. More detailed description of *aspects* as an instrument of CBC provides further in subsection II-A.

A. Motivation

Motivation for this research is to elaborate and investigate Categorization-based concept for structuring textual requirements, that helps in solving the problems regarding requirements quality. Additionally, to study an impact of category-based approach on requirements development process. This motivation leads to the following research questions:

Research question 1: *How to measure a quality of requirements?*

Research question 2: *How to develop a set of requirements, which corresponds to requirements of high level quality?*

Evaluation of a proposed method in requirements engineering field is a challenging task, which requires an effort from a researcher due to its subjective nature [3]. In this study raises the question of appropriate and adequate evaluation techniques and methods for the concept assessment, which allow to demonstrate the categorization-based concept feasibility on practice. The research will cover estimation of requirements quality with outcomes comparison before and after application of the categorization-based concept.

Based on these questions we have formulated these hypotheses:

Hypothesis 1: The Categorization-based concept for structuring textual requirements can improve their quality. The CBC appears as a general concept, which provides independence from a certain chain of tools. It grants a user flexibility in selecting categories with an appropriate level of granularity.

Hypothesis 2: The Categorization-based concept for structuring textual requirements serves to support requirements engineering process and procedure of review. It provides a common instrument for all roles of system development process. It maintains a traceability between requirements on all stages of system design. The concept works with requirements meta-data seamlessly integrating it through the whole development process and thereby, improves quality of requirements.

II. Categorization-based Concept

Usually, requirements provided by stakeholders have poor structure and ambiguous description of a designed system. This can possibly lead to misunderstandings from developers or reviewers. Furthermore, a poor readability of requirements might result in mistakes within system design or time increasing for their review. All these points indicate a low quality.

In order to improve quality of requirements and prevent mentioned problems, requirements should be structured and formalized to become more exact. Here we claim, that Categorization-based concept aims this purpose providing a method for arranging requirements in a consistent structure.

The idea behind CBC arises from common requirements categorization and goes ahead forming a structure of textual requirements. Requirements are categorized by their characteristics. Therefore, one or more categories can be attached to a single requirement, depending on the requirement's composition. The number of applied categories increases with rising complexity of the considered requirement. That implies, a requirement with a range of categories should be splitted into several simplified requirements with less number of attached categories, if it is possible with respect to context.

The user can define, which categories correspond to each requirement. Moreover, CBC provides a certain degree of freedom for the user to choose a granularity of every attached category. In other words, the user makes a decision on how detailed the requirements should be considered and categorized. After an analysis of the whole scope of requirements, every requirement should be categorized.

The output of this procedure is structured and more consistent requirements with attached set of corresponding categories. This set constitutes a new artifact in system development process, that represents a full structure of the requirements. Such structured requirements are more precise and unambiguous. Thanks to that the reviewing process becomes easier and more accurate; risk of mistakes in system design declines due to better legibility of requirements. We can conclude, that quality of requirements increases by applying Categorization-based concept .

Additionally, the idea of the CBC is general and allows a user to be independent from specific tools. It can be applied within already known for a user chain of tools and thereby, save costs and time for a project realization, instead of training employees for new extra tools.

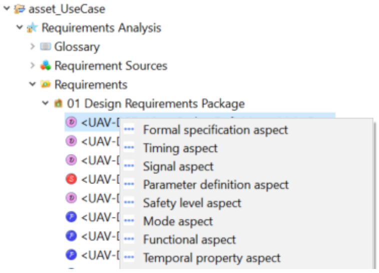


Fig. 1. List of *aspects* in AutoFOCUS 3

A. *Categorization-based concept in* AutoFOCUS 3

CBC has been implemented in AutoFOCUS 3 . In AF3 requirements categorization incarnates so called *aspects* (templates for requirements characteristics, which represent requirements' categories).

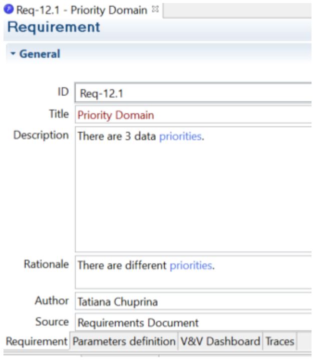
In AF3 eight *aspects* have been created, which are depicted in Figure 1.

- *Timing aspect* - the requirement refers to timing constraints
- *Signal aspect* - the requirement defines a signal, its type and its range
- *Parameter definition aspect* - the requirement defines a parametric value. This allows to re-use the requirement by changing the value of the parameter
- *Safety level aspect* - the requirement defines a safety level of the system
- *Mode aspect* - the requirement defines an operational state of system e.g. active/inactive mode.
- *Design choice aspect* - the requirement was derived from a design choice of the software developer
- *Functional aspect*- the requirement specifies a particular behavior by relating inputs and outputs
- *Temporal property aspect* - the requirement defines a property expressed by temporal patterns

However, the number of *aspects* can vary with respect to a context of project and needs of requirements engineer/ reviewer. *Aspects* can be applied to system requirements from initial stage of the development process, from requirements elicitation. *Aspects* form a structure of the analyzed requirements tagging them. Figure 2 presents a view of a textual requirement with assigned *Parameter definition aspect*.

As it was mentioned above, every aspect provides a template for requirement's main characteristics. The templates collect meta-data of requirements. For instance, Figure 3 shows a view of *Parameter definition aspect* template attached to a corresponding requirement.

As it is shown in Figure 3, the aspect template includes information from tagged requirement, thereby binding the aspect and the requirement together. Data captured in aspects' templates depends on the granularity of the applied category (aspect), chosen by the CBC user. Furthermore, the structured requirements with assigned *aspects* appear as an input for development and reviewing processes. This means, that *as-*



Req-12.1 - Priority Domain

Requirement

General

ID: Req-12.1

Title: Priority Domain

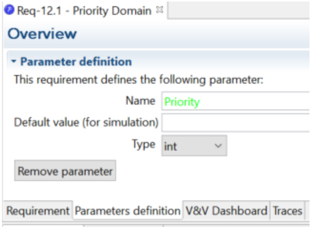
Description: There are 3 data priorities.

Rationale: There are different priorities.

Author: Tatiana Chuprina

Source: Requirements Document

Fig. 2. Requirement with Assigned *Parameter definition aspect* in AUTOFOCUS 3



Req-12.1 - Priority Domain

Overview

Parameter definition

This requirement defines the following parameter:

Name: Priority

Default value (for simulation):

Type: int

Fig. 3. *Parameter definition aspect* template in AUTOFOCUS 3

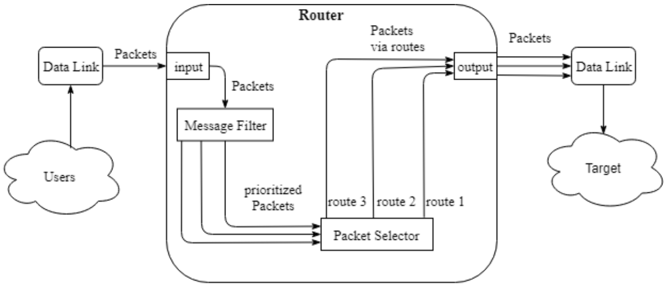


Fig. 4. Diagram of the Router in the Network

pects link artifacts of these processes (e.g system architectures, low-level requirements (LLR), review results, etc.) supporting traceability between them; contribute in mapping requirements into system design.

In addition, it is easier to check structured requirements.

Summing up all statements from above, *aspects* based on Categorization-based concept , facilitate in increasing quality of requirements.

Let us consider CBC with the ”Router” use case, which was inspired by an industrial project being conducted together with Airbus and Fortiss.

B. Use Case for applying Categorization-based concept

“Router” use case exposes an example of a simple “Router” component with configured quality of service (QoS) management, which is used in telecommunication networks. In this use case Stakeholders have provided the system requirements (Table I) as a text document with semi-structured information and a simple component diagram (Figure 4).

System requirements.

“In communication system Data Link (DL) is usually used by several users. The users can proceed with different types of traffic. For better use of Data Link resources, traffic from diverse users can differ by priorities (depends on QoS and user’s agreement). Depending on the priority, traffic should be forwarded along a certain route with a defined latency. For high priority packets the route shall be defined with the lowest latency. The lower priority of traffic, the higher latency the provided route has. Configuration input for Selector should be specified by QoS agreement of user.

Message Filter defines priority of packets and send them directly to Selector.

Selector should:

- receive packets from Message Filter;
- define a proper route, according to packet priority;

Packets shall be sent from Selector to Data Link using 3 routes.

All QoS policy is defined in Selector component.

There are 3 data priorities:

1) High priority information - Selector should route such packets into 1st route only!
2) Middle level priority - packets can be sent by Selector into 1st route , if the route is free; otherwise, traffic is transmitted into 2nd route;
3) Low priority - data can be sent in 3 routers, depends on which route is free at the moment.”

TABLE I
SYSTEM REQUIREMENTS PROVIDED BY STAKEHOLDERS

1) *Requirements Structuring:*

Initial phase of development process is elicitation and analysis of requirements. We claim, that during analysis process *aspects* should be applied to the requirements in order to categorize them and form their structure. Considering further the “Router” use case, the following *aspects* can be assigned to the given requirements:

“Req-1. In communication system Data Link is usually used by several users.”

[*Mode aspect*] implies, that DL is in active state, when the users send traffic.

“Req-2. The users can proceed with different types of traffic.”

[*Signal aspect*] describes, that the packets have different types.

“Req-3. For better use of Data Link resources, traffic from diverse users can differ by priorities (depends on QoS in user’s agreement).”

[*Signal aspect*] describes the traffic.

“Req-4. Depending on the priority, traffic should be forwarded along a certain route with a defined latency.”

- *Parameter definition aspect* indicates, that priorities, routes and latency can be kept as re-useable parameters;
- *Signal aspect* describes input for the routes;
- *Functional aspect* describes the behavior of “Packets Selector”;
- *Temporal property aspect* shows, that the statement includes a condition;

The rest of the requirements should be analyzed and structured similarly.

After requirements have been categorized, the requirements engineer can proceed with the following inspection. First of all, he/she should check, if all requirements have *aspects* . Next, re-consider the requirements with more than 1 aspect in their categorization; it can mean, that such requirements are not precise enough. For example, the requirement “Req-4”:

“Req-4. Depending on the priority, traffic should be forwarded along a certain route with a defined latency.“

- *Parameter definition aspect*,
- *Signal aspect*,
- *Functional aspect*,
- *Temporal property aspect.*

It can be splitted into several requirements according to *aspects* categories:

For [*Functional aspect*]:
“Req-4.1. Selector should forward traffic along a certain route.”

For [*Parameter definition aspect*]:
“Req-4.2. Every route operates with defined latency.”
“Req-4.3. There are several (three) routes.”
“Req-4.4. There are several priorities for traffic.”

For [*Signal aspect*]:
“Req-4.1. Traffic is prioritized.”

For [*Temporal property aspect*]:
“Req-4.2. Traffic priority defines a route for traffic.”

The ideal case is when every requirement has only one aspect, but in real project requirements categories usually overlap each other. Therefore, the number of *aspects* attached to a requirement should tend to be minimized.

After applying this method also to the other requirements, we have obtained the following structure of requirement, presented in Table II.

Now we have more consistent high-level requirements (HLR) in comparison with the requirements initially provided by the stakeholder. That means, the requirements are better structured, more precise and less complex. These characteristics rise quality of the requirements. Therefore, it is easier to work with such requirements during development phase or review process.

These structured HLR now can be an input for system design process. Developers can also analyze the requirements and provide additional requirements, which were inferred from HLR. For this purpose, there exists a *Design Choice aspect*. This aspect indicates, that changes in requirements have been done by developers team. Example of such requirement is provided:

”Req-13. Packets follow the Ethernet standard.”

the requirement now has two *aspects* [*Signal aspect* and ***Design Choice aspect***].

2) Requirements Review: Aspects support also requirements review process. Every *aspect* provides a check-list specified for respective category of requirement. This feature serves for more rigorous review of the requirements instead of a general check. More over, different *aspects* imply different activities of requirements checking. As an example from the use case, to inspect Req-10.2:

“Req-10.2. Selector shall use 3 routes for packets sending to Data Link.”

[*Functional aspect*], here reviewer checks correctness of algorithms implemented within this requirement. However, the same check is useless for Req-4.3:

“Req-4.3. There are several (three) routes.”
,which is defined by [*Parameter definition aspect*].

Working with HLR structured by *aspects* , the reviewer can trigger the following revision:

- whether *aspects* have been applied to each requirement,
- whether the *aspects* match appropriately with the considered requirements (the reviewer can propose additional *aspects* or make changes in the ones already attached),
- whether all *aspects* check-lists give successful outcomes.

III. Evaluation and Validation

Firstly, Categorization-based concept has been validated by industry use case within ASSET project during avionics component development.

Next step for evaluation and validation of Categorization-based concept can be series of empirical studies conducting with students, at Fortiss research institute to find out, how a user can handle unstructured system requirements applying Categorization-based concept and without CBC . The outcomes will be compared in order to check, whether quality of the considered requirements increases with CBC application. For this purpose criteria of comparison can be defined as:

- time-consumption during system development based on the structured and unstructured requirements
- comprehension of the structured and unstructured requirements (e.g. time for read and understanding requirements)
- number and types of errors have been corrected within reviewing process

However, these criteria can vary depending on further results of study.

IV. Related Work

The scientific base of our research includes study work about requirements categorization [8]. There the author considers general requirements categorizations commonly applied in industry and recognized by the scientific community [9], [10], [11], [12] and investigates requirements categorization based on a system model and its impact on requirements quality, relying on state-of-the-Art research results such as [13], [14] and others.

"Req-1. In communication system Data Link is usually used by several users."	[*Mode aspect*]
"Req-2. The users can proceed with different types of traffic."	[*Signal aspect*]
"Req-3. For better use of Data Link resources, traffic from diverse users can differ by priorities (depends on QoS in user's agreement)."	[*Signal aspect*]
"Req-4.1 Selector should forward traffic along a certain route."	[*Functional aspect*]
"Req-4.2 Every route operates with defined latency."	[*Parameter definition aspect*]
"Req-4.3 There are several (three) routes."	[*Parameter definition aspect*]
"Req-4.4 There are several priorities for traffic."	[*Parameter definition aspect*]
"Req-4.5 Traffic is prioritized."	[*Signal aspect*]
"Req-4.6 Traffic priority defines a route for traffic."	[*Temporal property aspect*]
"Req-5. For high priority packets the route shall be defined with the lowest latency."	[*Temporal property aspect*]
"Req-6. The lower priority of traffic, the higher latency the provided route has."	[*Temporal property aspect*]
"Req-7. Configuration input for Selector should be specified by QoS agreement of user."	[*Signal aspect*]
"Req-8.1 Message Filter defines priority of packets."	[*Functional aspect*]
"Req-8.2 Message Filter sends packets directly to Selector."	[*Functional aspect*]
"Req-9.1 Selector should receive packets from Message Filter."	[*Functional aspect*]
"Req-9.2 Selector should define a proper route, according to packet priority."	[*Functional aspect and Temporal property aspect*]
"Req-10.1 Selector shall send packets to Data Link."	[*Functional aspect and Signal aspect*]
"Req-10.2 Selector shall use 3 routes for packets sending to Data Link."	[*Functional aspect*]
"Req-10.3 There are 3 routes for sending packets from Selector to Data Link."	[*Parameter definition aspect*]
"Req-11 All QoS policy is defined in Selector component."	[*Functional aspect*]
"Req-12.1 There are 3 data priorities."	[*Parameter definition aspect*]
"Req-12.2. High priority information - Selector should route such packets into 1st route only!"	[*Temporal property aspect*]
"Req-12.3. Middle level priority - packets can be sent by Selector into 1st route , if the route is free; otherwise, traffic is transmitted into 2nd route."	[*Temporal property aspect*]
"Req-12.4. Low priority - data can be sent in 3 routers, depends on which route is free at the moment."	[*Temporal property aspect*]

TABLE II
STRUCTURED REQUIREMENTS WITH ATTACHED ASPECTS

The next paper, which made an impact to start this research was [15], where the authors raise doubts about applying of goal-oriented requirements engineering (GORE) in practice due to missing sufficient documentation for industry and lack of comprehension between the scientific society and industry. A consequence paper [3] describes current problems in industry with concern to requirements engineering field and possible reasons for existing gap between researchers and industry.

From the industry perspective, Categorization-based concept study embraces practical knowledge assembled by industry partners of Fortiss with conformity to industry standards and regulations, such as DO-178C [16], DO-331 [17], ISO29148:2011 [18] etc. The standards comprise guidelines and recommendations for system development process.

Our research contributes to the requirements categorization topic, investigating an influence of the general Categorization-based concept on requirements quality.

Commonly recognized in scientific community, the goal-based requirements categorization doesn't seem to prevalent in the industry. The study of Categorization-based concept will expose the idea, which goes ahead of general requirements categorization, providing a common instrument within system development process. Moreover, this concept will bring a method, that guides requirements engineering and review activities. Our research will grant the approach for improving quality of requirements by their structuring using Categorization-based concept .

V. Current Status

At the moment, the research goes further with the idea of Categorization-based concept . It implies to consider an impact of CBC on a whole development process including testing phase. Also the study is supposed to cover requirements reuse and a role of the Categorization-based concept in supporting that process.

Currently, question of relevant and sufficient methods for proper evaluation of the Categorization-based concept still arises. The investigation of possible techniques is in process.

References

[1] V. Aravantinos, S. Voss, S. Teufl, F. Hölzl, and B. Schätz, "AutoFOCUS 3: Tooling concepts for seamless, model-based development of embedded systems," in *Proc. 8th Int. MODELS Workshop Model-based Archit. Cyber-physical Embed. Syst. (ACES-MB '15)*, 2015, pp. 19–26. [Online]. Available: http://ceur-ws.org/Vol-1508/paper4.pdf

[2] T. Shah and S. V Patel, "A review of requirement engineering issues and challenges in various software development methods," *International Journal of Computer Applications*, vol. 99, pp. 36–45, 08 2014.

[3] D. M. Fernández, "Supporting requirements-engineering research that industry needs: The napire initiative," *IEEE Software*, vol. 35, no. 1, pp. 112–116, 2018. [Online]. Available: https://doi.org/10.1109/MS.2017.4541045

[4] "Ibm rational doors." [Online]. Available: http://www-03.ibm.com/software/products/en/ratidoor

[5] "Requirements quality suite." [Online]. Available: www.reusecompany.com/requirements-quality-suite

[6] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy approach to requirements syntax (ears)," in *Proceedings of the 2009 17th IEEE International Requirements Engineering Conference, RE*, ser. RE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 317–322. [Online]. Available: http://dx.doi.org/10.1109/RE.2009.9

[7] Obeo, "UML Designer.Getting started," http://www.umldesigner.org/tutorials/tuto-getting-started.htm, 2018.

[8] J. Eckhardt, "Categorizations of product-related requirements in practice: Observations and improvements," Ph.D. dissertation, Technical University Munich, Germany, 2017.

[9] K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[10] S. Robertson and J. Robertson, *Mastering the Requirements Process: Getting Requirements Right*, 3rd ed. Addison-Wesley Professional, 2012.

[11] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*. John Wiley & Sons, Inc., 1998.

[12] A. Van Lamsweerde, “Goal-oriented requirements engineering: A guided tour,” in *Proceedings of the Fifth IEEE International Symposium on Requirements Engineering*, ser. RE ’01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 249–. [Online]. Available: http://dl.acm.org/citation.cfm?id=882477.883624

[13] M. Broy, “Rethinking nonfunctional software requirements: A novel approach categorizing system and software requirements.” Hinchey, M., editor, 2016, software Technology: 10 Years of Innovation in IEEE Computer. John Wiley & Sons/IEEE Press.

[14] P. Mager, “Towards a profound understanding of non-functional requirements,” Master’s thesis, Technical University Munich,, 2015.

[15] A. Mavin, P. Wilkinson, S. Teufl, H. Femmer, J. Eckhardt, and J. Mund, “Does goal-oriented requirements engineering achieve its goal?” in *RE*. IEEE Computer Society, 2017, pp. 174–183.

[16] “Do-178c, software considerations in airborne systems and equipment certification,” Special C. of RTCA, 2011.

[17] “Rtca do- 333: Formal methods supplement to do-178c and do- 278a,” 2011, standard.

[18] “Iso/iec/ieee 29148-2011 (2011). international organization for standardization (iso) and international electrotechnical commission (iec) and ieee. iso/iec/ieee 29148: Systems and software engineering – life cycle processes – requirements engineering.”