Empirical Software Engineering Group

Centro de Informática U·F·P·E

Universidade Federal de Pernambuco

GENERATION OF REQUIREMENTS SPECIFICATIONS FROM SOFTWARE ENGINEERING MODELS

This briefing reports evidence on methods and techniques dealing with the generation, of requirements from models based on scientific evidence from a systematic review.

MAIN FINDINGS

Following are the techniques used to generate requirements from software engineering models:

- Literate modelling: In this approach a new type of document is defined, Business Context Document (BCD), in which (1) the diagrams are embedded as figures in a document written in natural language and the visual model is paraphrased; (2) brief explanations of the relevant UML syntax are included in footnotes; (3) real, yet simple examples of the notation are presented to illustrate specific points; and (4) some important questions such as the rationale are emphasized. Literate models are thus UML models that are embedded in a natural language text which explains them.
- Deriving requirements from business modelling: (1) discover the applicability of Michael Jackson's problem frame propositions to complex, industrial projects; (2) link business process models by means of the RAD (RoleActivity Diagram) notation on these problem frames; and finally (3) derive requirements from the process models. The research is validated through an industrial ebusiness system and the case study is conducted using actionresearch.
- Deriving requirements from UML models: The GeNLangUML (Generating Natural Language from UML) generates English specifications that "paraphrase" UML 1.5 class diagrams by making use of a linguistic ontology called WordNet. (1) on the one hand they wish to provide users with two different views of the system specification at any time: UML and natural language; (2) on the other hand they see the motivation for their tool in a reverse engineering process during the maintenance stage, to enable backwards transformation allowing the stakeholders to "visualize" the changes in the system's implementation in natural language.
- Use cases, scenarios and user stories: An approach proposes a process that uses the existing use case model as a starting point and derives new scenarios, taking into account situations which have not yet been considered. This derivation of new scenarios leads, in turn, to new requirements. Another approach is concerned with generating the hierarchy of requirements while other proposes a pattern with which to derive the requirements' text. In contrast, another research proposes the generation of formal language requirements from use case models. This work has been tested in several complex models from Siemens operating companies.
- Deriving requirements from user interface modelling: A research propose extended usage models that consist of three submodels: scenario model, action model, and user interface model: (1) the scenario model describes the semantics of the usage model in terms of goals that can be achieved when using the software, tasks that have to be accomplished in order to achieve the goals, and solutions, which are scenarios on how to use the software to solve a particular task; (2) the action model is a state machine that defines all possible sequences of user inputs (actions) and covers the information of a conventional usage model; and (3) the user interface model describes the user interface of the software and the interface elements that are themselves subject to user inputs. In this context, an HTML document can be automatically generated which structures the information already introduced in the extended usage models and can serve as user documentation.

A proposed taxonomy with two dimensions of generation methods is composed by "combination mode" and "scope".

With regard to the combination mode, there are two approaches, the generative and the integrative:

- Generative: These approaches propose algorithms, rules or patterns to generate textual requirements starting from models. These proposals are presented with or without automatic support and the text generated can be written in natural or formal language:

  Natural language: These approaches generate candidate natural language requirements which must be validated. In this group there are the generation of requirements from i* models; the generation of English specifications that paraphrase UML class diagrams; the derivation of textual requirements from use cases, scenarios and user stories; and the generation a hierarchy of requirements from use case diagrams.

  Formal language: These approaches generate requirements written in a formal notation. Formal methods bring benefits such as the mechanical analysis of a system to check for deadlock and livelock freedom, but the adoption of formal methods in industry is challenged by the cost and complexity involved in the formal specification of the system. Hence some approaches in literature have investigated the derivation of formal specifications from requirements models. In this group we should highlight the work on goals models, which addresses the automatic generation of operational requirements described by means of pre and postconditions and triggers, and the generation of requirements described by means of the tabular technique. In addition,- there is the generation of operational requirements in CSP process algebra from use case specifications.

- Integrative: These approaches do not provide algorithms, rules or patterns to generate requirements from models, but rather openended guides to relate models and textual requirements. The resulting requirements can be written in natural language or in a formal notation.

With regard to the scope of the approach, some proposals deal with the generation of single requirements while others deal with the generation of requirements documents.

- Requirement: These approaches deal with the generation of requirements or sets of requirements, but do not address the requirements documents in which the requirements should be placed. One example is the approach that deals with deriving requirements from i* SD models.
- Documental: These approaches concentrate on the manual, automatic or semi--automatic generation of requirements documents. For example, the tool Objectiver which semiautomatically generate requirements documents structured from a goal specification. The literate modelling trend as a whole can also be included in the documental approach.

Keywords:

Specification generation from software engineering model

Textual requirements generation from software engineering model

Requirements document generation from software engineering model

Systematic literature review

Who is this briefing for?

Software engineers practitioners who want to make decisions about software requirements generation from models based on scientific evidence.

Where the findings come from?

All findings of this briefing were extracted from the systematic review conducted by Nicolás et al.

What is systematic reviews?

cin.ufpe.br/eseg/systematic-reviews

What is included in this briefing?

The main findings of the original systematic review.

Evidence characteristics through a brief description about the original systematic review and the studies it analized.

What is not included in this briefing?

Additional information not presented in the original systematic review.

Detailed descriptions about the studies analised in the original systematic review.

For additional information about this briefing:

cin.ufpe.br/eseg/briefings

ORIGINAL SYSTEMATIC REVIEW REFERENCE

Joaquín Nicolás, Ambrosio Toval, On the generation of requirements specifications from software engineering models: A systematic literature review, Information and Software Technology, Volume 51, Issue 9, September 2009, Pages 12911307, ISSN 09505849, http://dx.doi.org/10.1016/j.infsof.2009.04.001.