Requirements Engineering Education at Academia: A Model-based Approach

Erik Kamsties und Fabian Kneer
Fachhochschule Dortmund
{erik.kamsties, fabian.kneer}@fh-dortmund.de

Motivation. Teaching of a requirements engineering (RE) course at academia is a challenge from our perspective, because students need a feeling of success which is hard to implement in a pure RE course. When a course involves also coding, a feeling of success often results from a running piece of software that meets the course assignments.

Research question. We look in this paper at the question of providing early feedback to students in RE education. We try to find a solution for a heterogeneous group of participants from different directions of Computer Science (business, medical, and technical domains) in a Master curriculum.

Solution. We propose the $\mathrm{DO}_{\mathrm{RE}}F$ ("do requirements first") educational framework[1], which offers a language for requirements engineering combined with generators. Figure 3 shows an example of three requirements with a short name and different attributes. Our language is an internal DSL (domain-specific language) based on Python, thus it is executable. It is supported using a Python workbench such as PyCharm. A collaboration tool such as GitLab may be used for distributed authoring of requirements.

The language bases on a common view on RE (shown in Figure 1), which includes a *product* and a *process* perspective, as well as the *domain of discourse* (called *world*). Documents describe relevant parts of the world and a workflow orchestrates the production of documents by employing a particular way to analyze the world and to design systems.

The abstract syntax of our language is shown in Figure 2. The language offers just basic constructs for RE. The language is

- *comprehensive* - it addresses products and processes
- *method-agnostic* - it does not infer a particular RE approach
- *extensible* - using modules, the language can be tailored to a particular RE approach (e.g., istar).

The abstract syntax shows that the world is actually a *system*, which composed of (sub-)systems. The outermost system is called *context*. A *system* is related to a project and a *project* results in a set of documents (product perspective). A *document* contains *textual elements* like requirements and *models*.

[1] *http://www.doref.org*

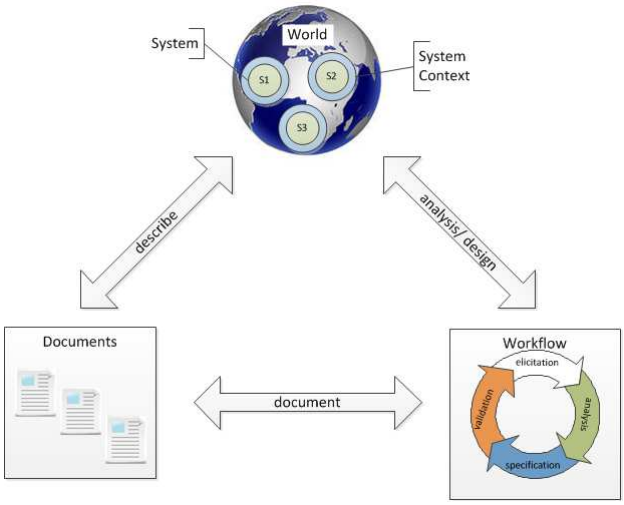

Abbildung 1: View on Requirements Engineering

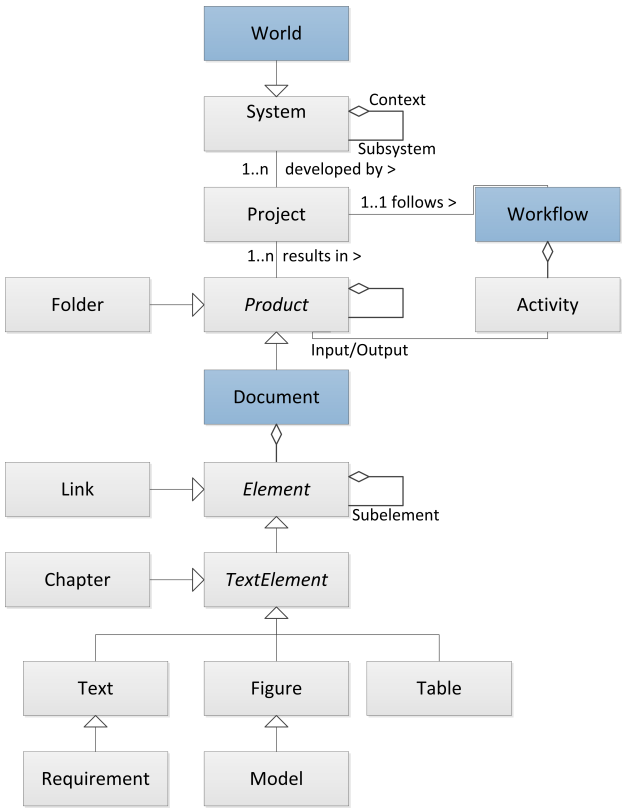

Abbildung 2: Abstract Syntax

```
Requirement("The robot shall clean the appartment at night.",
            "Clean at night",
            {'Priority': 1,
            'Effort': 20,
            'Optional': 1})
Requirement("The suction power must not exceed ${suction_power}.",
            "Power",
            {'Priority': 1})
Requirement("The operation of the vacuum cleaner should be as silent as possible.",
            "Silence",
            {'Priority': 2})
```

Abbildung 3: Example Requirements

A project follows a particular *workflow*, made-up of *activities* (process perspective). Activities are informally described using a wiki markup language (RST). Additionally, they can be implemented using Python if they allow for full formalization (like consistency checks). Scripts can be implemented this way, which operate on the requirements.

Two generators are currently available. One generates documents (HTML, PDF) from the *project* → *folder* → *document* hierarchy. The second generates a HTML process guide from the activity descriptions.

RE-specific Extension We implemented an i* module as an example of a method-specific extension. The i* module supports all elements of i* [5] in order to generate Strategic Dependency and Strategic Rationale models in a requirements document. In addition, it provides a specific process guide to i* and implements a few activities. Most important of all, an *interactive* evaluation of goal satisfaction is available based on the approach by Grau et al. [2]. The interactivity comes from the Python console, which allows to query an i* model. The steps of the goal evaluation are shown to the user as a sequence of diagrams.

A few RE modules are already available (including use cases, i*, some document templates). Further modules can be developed easily by students. That is, the students can acquire also method engineering skills and develop their own project-specific RE process.

Related Work. Education and training of requirements engineering is a topic of many research surveys. A survey of Ouhbi et al. [2] shows that present tools and trainings focus on a better requirements elicitation. The main challenges are described in [1] and [3] including (1) the lack of awareness for the motivation and problems in the area of RE and (2) how to create a realistic situation for requirements elicitation and validation without a real customer.

ReqT is a language-based approach for training which was introduced by Regnell [4]. It comes with a DSL, differences are in the philosophy behind the abstract syntax, e.g. $\mathrm{DO_{RE}}F$ covers also processes.

[2] *http://istar.rwth-aachen.de/tiki-view_articles.php*

Summary. Requirements engineering education is facing a number of challenges, some of them are addressed by the $\mathrm{DO_{RE}}F$ framework. The core idea of the framework is to let students focus at the content and the semantics of requirements documents rather than layout. The executability of the language provides early feedback to students, especially when simulation of RE models is possible as in the case of i* models.

Traceability is facilitated, especially, if the implementation is done in Python as well.

The first trial with students starts in the Master course "Requirements Engineering" in the winter term 2014/2015 at the University of Applied Sciences and Arts Dortmund. First results were presented at the RE-Fachgruppen meeting 2014 in Dortmund.

Literatur

[1] David Callele and Dwight J. Makaroff. Teaching requirements engineering to an unsuspecting audience. In Doug Baldwin, Paul T. Tymann, Susan M. Haller, and Ingrid Russell, editors, *SIGCSE*, pages 433–437. ACM, 2006.

[2] Sofia Ouhbi, Ali Idri, JosLuis Fernndez-Alemn, and Ambrosio Toval. Requirements engineering education: a systematic mapping study. *Requirements Engineering*, pages 1–20, 2013.

[3] Gil Regev, Donald C. Gause, and Alain Wegmann. Experiential learning approach for requirements engineering education. *Requir. Eng.*, 14(4):269–287, 2009.

[4] Björn Regnell. reqt.org - towards a semi-formal, open and scalable requirements modeling tool. In *Requirements Engineering: Foundation for Software Quality - 19th International Working Conference, REFSQ 2013, Essen, Germany, April 8-11, 2013. Proceedings*, pages 112–118, 2013.

[5] Eric Siu-Kwong Yu. *Modelling Strategic Relationships for Process Reengineering.* PhD thesis, Toronto, Ont., Canada, Canada, 1996. UMI Order No. GAXNN-02887 (Canadian dissertation).