A Recommender System for Didactical Approaches in Software Engineering Education

Michael Koch, Dieter Landes

Faculty of Electrical Engineering and Informatics
Coburg University of Applied Sciences and Arts
Friedrich-Streib-Straße 2
96450 Coburg, Germany
{michael.koch, dieter.landes}@hs-coburg.de

Abstract: As a basis to improve software engineering education, comprehensive information is needed on learning and teaching arrangements and experiences that have been made with these arrangements, covering a wide range of aspects. In addition, this information needs to be analyzed in a flexible manner in order to identify suitable arrangements that might be adapted for future use in a specific software engineering course.

This paper details a software tool called CORE (Competence Repository) that we are currently developing in order to support the improvement of software engineering education. In particular, we provide some details on the data model and the analysis and information retrieval facilities that CORE offers to its user.

1 Introduction

Software engineering has become increasingly important over the last years, not only for computer scientists. The research project EVELIN (Experimental improVEment of Learning software engINeering) aims at identifying competencies which students should have, and didactic approaches that are most appropriate to foster given competencies in the context of software engineering.

The improvement of software engineering education is a complex iterative process since courses and their environment are changing over time and need to be adjusted. To identify didactical approaches which are appropriate to foster students' competencies in a given context, new teaching methods need to be tried and tested in experiments, and results need to be evaluated. These experiments are always performed under certain conditions. Capturing these variables is essential for measuring the quality of an experiment as well as the quality of an entire course.

To support instructors enhancing their courses by sharing their knowledge and experiences, supply recommendations for making didactical decissions, and provide an overview about the quality and evolution of their courses, we are currently developing a

software tool called CORE (Competency Repository). In this article we give a short overview about the intentions of CORE, the current state of development, and future work.

2 Intentions of CORE

CORE is intended to be a tool for documenting, tracing, and improving software engineering courses. It is designed as a shared knowledge base of experiences made by instructors, e.g. through evaluations. Based on these data CORE will be able to assist instructors of future courses in didactical decisions, e.g. by recommending appropriate teaching methods depending on the current setting, or in general recommending experience reports about specific learning and teaching arrangements in a similar settings, as well as useful literature and resources.

Since the recommendation of suitable experience reports in a given setting is a core feature of CORE, it is important to know about the intentions of a given course or at least of a complex of themes. Therefore it is necessary to find a way to describe these intentions or in general the educational goals, their inter-relationships and dependencies, as well as resources and tasks used to achieve the educational goals. We want to make these relationships editable and visible to the user – if desired. This might be also a useful approach to make decision paths made by the recommender system visible to the user, since transparent recommendations are more accepted than non-transparent ones [SR02]. Therefore we analyzed several existing goal modeling notations like i* and proposed a own notation for high-level educational goals representing competencies in [KL14b] and [KL14c]. In accordance to the structured description of educational goals, we also plan to visually describe teaching methods, as well as their evolution, dependencies, and suitability to meet a specific goal. One promising approach could be means-end maps (ME-Maps) as defined in [WSY14]. In combination with ratings and experience reports, and instructors' and students' preferences, these models will be used as a basis for recommendations for adequate teaching methods.

As discussed in [Ko14], CORE should also be the basis for in-depth analyses giving a feedback about the quality and evolution of a software engineering course. We want to generate an integrated quality report based on the combination of multiple heterogenous data sources. This is also intended to trace the evolution of a software engineering course over a longer period of time.

In particular, we are searching for ways to see the development process of competencies of multiple stakeholders in education, namely students but also instructors. Here it is important to mention that we explicitly do not want to observe individual students, but the entire group of students for example by calculating the median. This is necessary to reduce outliers in the data but also guarantees the students' privacy. In accordance to the evolution of students' competencies, we also want to observe the evolution of skills of an instructor over a longer period. Here we focus on properties that are independent of a given course, as well as properties that depend on a given course, mainly based on students' feedback but also based on self-reflection reports.

A related aspect that CORE should support is the ability to document and trace the influences of evaluation results, observations, didactical reflections, and analyzes made by instructors and educational scientists. This also includes the documentation of the structural evolution of a course, including iterative changes in the course profile (e.g. changes in the intended learning outcomes an used teaching method) as well as changes in the setting (e.g. number of students, available resources, etc.).

Summarizing, it can be stated that CORE should be an intelligent knowledge-base for experiences made in software engineering education that actively supports instructors by providing quality analyses and recommendations for didactical decisions.

3 Integration of Available Heterogenous Data Sources

Currently, the two prime features of CORE are generating context-sensitive recommendations for didactical decissions and offering in-depth analyses about the quality and evolution of software engineering courses. As a basis, we identified available heterogenous data sources fitting to our purposes in [Ko14] and categorized them into "Evaluation Data", "Domain Knowledge", and information about the "Context".

Our evaluation data consists of questionnaieres, transcribed interviews, and experience reports from multiple stakeholders, namely students, instructors, as well as customers in the context of, e.g., capstone projects. In addition, common statistical information like drop-out rates and exam statistics are available in an anonymized form.

Domain knowledge is static in the context of a given course and covers, among other things, a pool of available teaching methods, SWEBOK [BF14], and several existing software engineering education ontologies.

The context information covers dynamic characteristics of concrete learning and teaching arrangements, including, e.g., personal profiles of instructors, group profiles of participating students, intended learning outcomes (goal profile), lecture notes, available resources, used teaching methods, as well as environmental parameters.

4 Current State

In [KL14a] we introduced the first operational version of CORE, realized as an independent Java Enterprise Application with a web-based user interface. This version already stores a significant amount of educational aspects in a database. It allows users to enter, manage, and search data in a form-based manner. Competencies and their inter-dependencies can be visualized as an directed acyclic graph as described in [KL14c]. This version of CORE also offers generic EJB- and REST-interfaces, allowing external applications to retrieel and manage entities.

However, this version of CORE can still be seen as a passive data repository, without capability to offer context-sensitive recommendations. The data model of this version

focuses on storing structural information, but lacks personal profiles als well as evaluation data for courses that are important for making recommendations and quality analyses in this context.

5 Summary and Future Work

This article gives an overview about the intentions, the database structure, and the current development state of CORE, a recommender system for didactical approaches in software engineering education.

Since the knowledge base of CORE will cover a wide range of educational aspects, it might be also used for other applications. An additional use case for CORE might be an assistant for planing and designing curricula. Another application could be a tool for adaptive learning.

Acknowledgement

The research project EVELIN is funded by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung) under grant no. 01PL12022A.

For additional information on EVELIN see http://www.evelinprojekt.de.

References

[BF14] Bourque, P., Fairley, R. E. (Eds.): Guide to the Software Engineering Body of Knowledge (SWEBOK): Version 3.0, IEEE Computer Society Press, New York, 2014

[Ko14] Koch, M., Ring, M., Otto, F., Landes, D.: Combining Statistical and Semantic Data Sources for the Improvement of Software Engineering Courses. In (Stamper, J. et al. Eds.): Proc. 7th International Conference on Educational Data Mining, London, 2014; pp. 341-342.

[KL14a] Koch, M., Landes, D.: Design and Implementation of a Competency Repository. In (Rocha, A. et al. Eds.): New Perspectives in Information Systems and Technologies Volume 1, Springer, Heidelberg, 2014; pp. 249-255

[KL14b] Koch, M., Landes, D.: Modeling Software Engineering Education with i*. In: Proc. 7th International i* Workshop (istar'14), CEUR Workshop Proceedings Volume 1157, Thessaloniki, Greece, 2014

[KL14c] Koch, M, Landes, D.: Notations for Modeling Educational Goal Profiles. In: Proc. 1st European Conference of Software Engineering Education (ECSEE 2014), to appear

[SR02] Swearingen, K., Rashmi, S.: The Role of Transparency in Recommender Systems. In (Terveen, L. G. et al. Eds.): Extended abstracts of the 2002 Conference on Human Factors in Computing Systems (CHI'02), ACM press, New York, 2002; pp. 830-831

[WSY14] Wang, J., Sturm, A., Yu, E.: Know-How Mapping: From i* to ME-maps. In: Proc. 7th International i* Workshop (istar'14), CEUR Workshop Proceedings Volume 1157, Thessaloniki, Greece, 2014