ReDSeeDS: Requirements Driven Software Development System

Michał Śmiałek, Albert Ambroziewicz, Jacek Bojarski, Wiktor Nowakowski, and Tomasz Straszak

Warsaw University of Technology, Warsaw, Poland
{smialek,ambrozia,bojarsj1,nowakoww,straszat}@iem.pw.edu.pl

Abstract. This paper presents a tool suite which enables requirements-driven software reuse. The suite is built around requirements models that are transformed into design and code thus forming so-called 'software cases'. These requirements models can be compared for software case retrieval and reuse.

1 Motivation and related work

Effective reuse of design and code artifacts is associated with mechanisms for asset indexing. Normally, additional indexing information has to be added to the reuse repository in a manual process. Some approaches allow for generating the indexes based on the information contained in the meta-model which defines the structure of the assets (see [1]). In other approaches, semantic information is attached to the assets to enable case-based retrieval (see [2]). In many product line approaches, a family of systems is indexed through requirements where the problem domain plays crucial role (see [3] for an interesting insight).

The fundamental problem with current requirements-based indexing approaches is that requirements lack precision. This prevents from automatic indexing and retrieval based on such indexes. In this work we follow the use-case-based path started by Jacobson et al. in [4]. We make the use cases precise as introduced in [5] at the same time offering tools that enable formulating requirements with asset indexing potential. This can be achieved by treating requirements as models and applying modern model transformation techniques. This way we address the problems with wider adoption of software reuse (see [6]) consisting mainly in the existence of investment barriers (effort necessary to prepare assets for reuse and then to retrieve them). The presented tool suite enables software development scenarios where requirements specifications (problem statements) automatically form indexes to design and code artifacts (problem solutions) at a very detailed level thus forming so-called 'software cases' ready for comparison and reuse.

2 ReDSeeDS tool mechanisms

Requirements based reuse has been accomplished through building of a tool called ReDSeeDS Engine in the context of the ReDSeeDS (www.redseeds.eu) Project. The tool allows for the following reuse scenario.

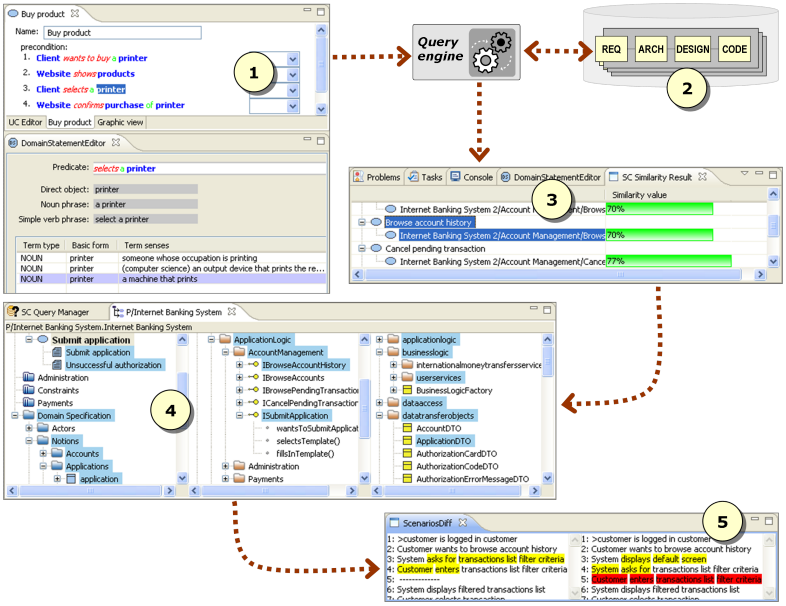

Fig. 1. ReDSeeDS Tool supporting the reuse scenario

The requirements, specified through an appropriate editor (1 in Figure 1) serve as queries to the software case repository (2). The tool responds with a set of similarity values (both for whole software cases and for individual requirements, see 3)). After choosing one of the similar software cases we can select its elements for reuse. This selection is done by following generated mapping links in a special multi-tree browser (4). Within this browser, we can visualize mapping links of the retrieved software case. These links identify a slice of a software case suitable for reuse. The slice (4) shows the root requirements (here: the 'Submit application' use case) together with the associated domain vocabulary and design elements. The design elements depend on the actual transformation (including the architectural style) used when creating the software case. Appropriate components with interfaces, operations and code-level (eg. DTO) classes can be taken for reuse. The last activity is to merge the resulting slice of a former software case with the current project by keeping those parts that completely fit, and adapting the other parts. This is again supported by the detailed results of the similarity measure that highlight the differences between two scenarios (5). Below we briefly present some of the important research contributions that allow for this scenario.

Requirements Specification Language. The language allows for building semantically rich use case models with detailed scenarios and is based on a detailed metamodel. The scenarios use a simple subject-verb-object(s) notation presented in Figure 2. The terms used within scenarios are linked to a domain vocabulary specific for a given software case (left parts of Q_1 and R_{21}).

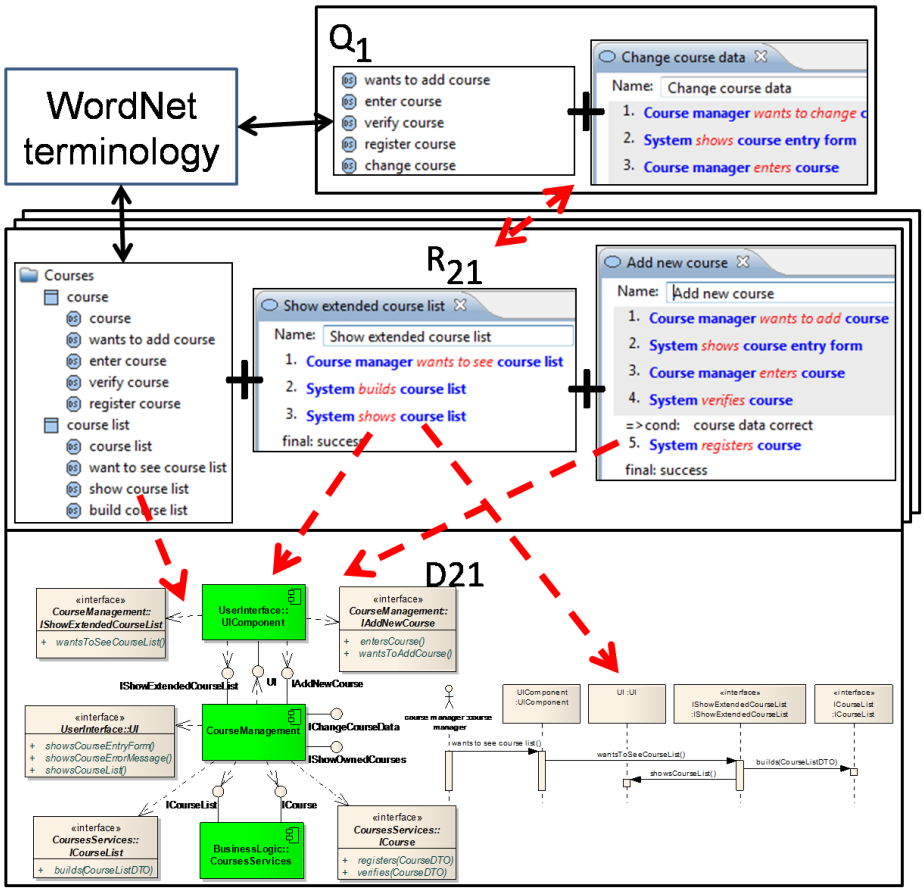

Fig. 2. Overview of the technological framework for the ReDSeeDS Engine

The vocabularies for different software cases are linked to a global terminology (WordNet based, see [7], [8]).

Indexing design models with requirements. The indexing is done with mapping links that are generated with an automatic MDA-style transformation. The transformation takes scenario steps and domain elements to create components, interfaces and operations at the design level. Moreover, interaction diagrams are generated which allow for partial code generation for the application logic. This is briefly illustrated in Figure 2 (see arrows between R_{21} and D_{21}).

Similarity comparison. The similarity measure is based on semantic and structural characteristics of use case scenarios. Queries to the software case repository are formulated through partial RSL models (see Q_1 in Fig. 2). For every query, a combination of measures is calculated. This combination includes mechanisms from information retrieval, graph comparison, description logic and WordNet (see [9] for more details).

3 Conclusion and validation summary

The ReDSeeDS Engine implements research contributions in several fields presented briefly in the previous section. It is a combination which forms a comprehensive system which facilitates creation and reuse of requirements-based 'software cases'. It is important that the system does not necessitate variability analysis and a family of systems is created implicitly through creation of evolving software cases. The validity of approach has been determined through a series of experiments led by Fraunhofer IESE with several development teams (see Acknowledgements). By using the UTAUT [10] methodology it has been acknowledged that the system is appreciated by the developers and has significant capabilities to improve performance of software development. The results of the validation study are available as a ReDSeeDS project report.

Acknowledgments. This work is partially funded by the EU: Requirements-Driven Software Development System (ReDSeeDS) (contract no. IST-2006-33596 under 6FP). The project is coordinated by Infovide, Poland with technical lead of Warsaw University of Technology and with University of Koblenz-Landau, Vienna University of Technology, Fraunhofer IESE, University of Latvia, HITeC e.V. c/o University of Hamburg, Heriot-Watt University, PRO DV, Cybersoft and Algoritmu Sistemos.

References

1. Lucredio, D., Fortes, R.P., Whittle, J.: MOOGLE: a model search engine. Lecture Notes in Computer Science **5301** (2008) 296–310
2. Gomes, P.: Software design retrieval using bayesian networks and WordNet. Lecture Notes in Computer Science **3155** (2004) 184–197
3. Moon, M., Yeom, K., Chae, H.S.: An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line. IEEE Transactions on Software Engineering **31**(7) (2005) 551–569
4. Jacobson, I., Griss, M., Jonsson, P.: Software reuse: architecture process and organization for business success. ACM Press (1997)
5. Biddle, R., Noble, J., Tempero, E.: Supporting reusable use cases. Lecture Notes in Computer Science **2319** (2002) 210–226
6. Sherif, K., Vinze, A.: Barriers to adoption of software reuse. A qualitative study. Information and Management **41** (2003) 159–175
7. Śmiałek, M., Ambroziewicz, A., Bojarski, J., et al.: Introducing a unified requirements specification language. In: Proc. CEE-SET'2007, Software Engineering in Progress, Nakom (2007) 172–183
8. Wolter, K., Smialek, M., Bildhauer, D., Kaindl, H.: Reusing terminology for requirements specifications from WordNet. In: Proc. RE'08 (2008) 325–326
9. Wolter, K., Krebs, T., Hotz, L.: A combined similarity measure for determining similarity of model-based and descriptive requirements. In: Proc. Artificial Intelligence Techniques in Soft. Eng. Workshop at ECAI 2008. (2008) 11–15
10. Venkatesh, V., Smith, R.H., Morris, M.G., et al.: User acceptance of information technology: Toward a unified view. MIS Quarterly **27** (2003) 425–478