CRESCO: Construction of Evidence Repositories for Managing Standards Compliance

Rajwinder Kaur Panesar-Walawege[1,2], Torbjørn Skyberg Knutsen[1,2], Mehrdad Sabetzadeh[1], and Lionel Briand[1,2]

[1]Simula Research Laboratory, Lysaker, Norway
[2]University of Oslo, Oslo, Norway
{rpanesar,mehrdad,briand}@simula.no, torbjsk@ifi.uio.no

Abstract. We describe CRESCO, a tool for **C**onstruction of Evidence **RE**positories for Managing **S**tandards **CO**mpliance. CRESCO draws on Model Driven Engineering (MDE) technologies to generate a database repository schema from the evidence requirements of a given standard, expressed as a UML class diagram. CRESCO in addition generates a web-based user interface for building and manipulating evidence repositories based on the schema. CRESCO is targeted primarily at addressing the tool infrastructure needs for supporting the collection and management of safety evidence data. A systematic treatment of evidence information is a key prerequisite for demonstration of compliance to safety standards, such as IEC 61508, during the safety certification process.

Keywords: Safety Certification, Conceptual Modeling, MDE, IEC 61508.

1 Introduction

Safety critical systems are typically subject to safety certification based on recognized safety standards as a way to ensure that these systems do not pose undue risks to people, property, or the environment. A key prerequisite for demonstrating compliance to safety standards is collecting structured evidence in support of safety claims. Standards are often written in natural language and are open to subjective interpretation. This makes it important to develop a precise and explicit interpretation of the evidence requirements of a given standard. In previous work [4, 3], we have proposed conceptual modeling for formalizing the evidence requirements of safety standards. This approach on the one hand helps develop a shared understanding of the standards and on the other hand, provides a basis for the automation of various evidence collection and management tasks.

In this paper, we describe CRESCO, a flexible tool infrastructure for creating repositories to store, query, and manipulate standards compliance evidence. Additionally, CRESCO generates a web-based user interface for interacting with these repositories. Our work was prompted by an observed need during our collaboration with companies requiring IEC 61508 compliance. In particular, we observed that little infrastructure support has been developed to date to support management of safety evidence based on a specific standard. This issue has also been noted in the literature as an important gap in the safety certification process [2, 5]. While CRESCO is general and can be used in conjunction with different standards, we ground our discussion in this paper on IEC 61508, which is a key standard for safety certification of programmable electronic systems.

In the rest of this paper, we will describe the key components of CRESCO. For the actual demonstration, we will follow, and expand where necessary, our presentation in this paper. Specifically, the demo will begin with motivational material – similar to this paper's introduction and augmented with snippets of the conceptual model in [4]. We then go on to describe the overall architecture of the tool, as shown in Figure 1. In the next step, we will use a combination of pre-recorded and live demonstrations to illustrate the main functions of the tool, discussed in Sections 2. Finally, as we outline in Section 3, our demonstration will provide information about the tool's implementation based on our existing documentation [1], and give details about availability.

2 Tool Overview

Users can interact with CRESCO in two roles: the administrator and general user. The administrator is responsible for importing the conceptual model into CRESCO, running the transformations and setting up and starting the web server. Once the server is started, the general users – typically experts from the supplier company or certification body – can add, view and manipulate the data in the database. In this section we provide an overview of the main components of CRESCO as shown in Figure 1(a).

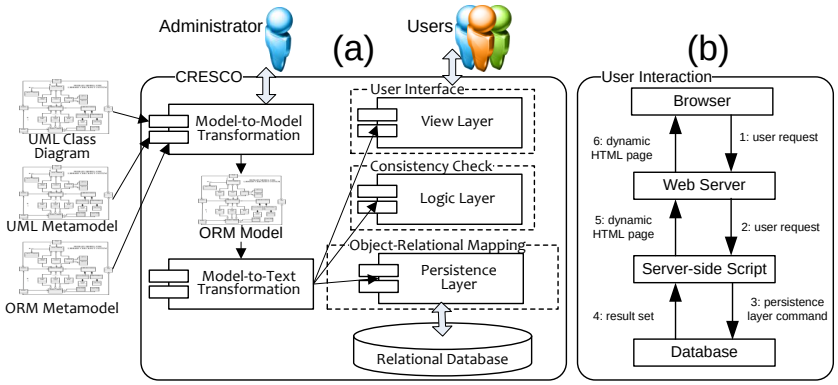

Fig. 1. Components of CRESCO and How the User Interacts With It

2.1 Generation of database schema and UI

The generation of the database and the user interface code involves two steps: a model-to-model (M2M) transformation and a model-to-text (M2T) transformation. The M2M transformation takes as input a conceptual model of a standard in the form of *a UML class diagram* [6]. This model can be created in any UML tool and then imported into CRESCO. An in-depth description of the conceptual model is beyond the scope of this demonstration paper – further details are available in [4]. The M2M transformation makes use of the UML meta-model [6] and a meta-model for an object-relational mapping (ORM) that we have created – see [1]. This ORM meta-model enables the storage (in a relational database) of objects that have been created based on a UML class diagram. The ORM meta-model includes a database schema (with tables, columns, foreign keys, etc)

and object-oriented concepts, mainly generalization. The M2M transformation iterates over the conceptual model and transforms it into a model that corresponds to the ORM meta-model.

The user interface is generated from the ORM model created during the M2M transformation. The M2T transformation iterates over the elements of the ORM model and generates the database implementation as well as all the code for accessing and updating the database via a web interface. The generated code is a combination of server-side Java code and HTML (see Section 3). Figure 1(b) shows how the user interaction is processed via the generated code. Figure 2 shows the user interface generated. The left hand pane lists all the tables that have been generated and the right hand pane is used to manipulate the rows in a selected table. The 'New' button shown is used to add a new row into the selected table. Figure 2 shows the table for the concept of `Agent`, who is an entity that carries out an activity required during system development. An `Activity` is a unit of behavior with specific input and output `Artifacts`. Each activity utilizes certain `Techniques` to arrive at its desired output and requires certain kind of `Competence` by the agents performing it. The agent itself can be either an individual or an organization and is identified by the type of role it plays. In CRESCO, one can: (1) create instances of concepts such as `Agent`, `Activity`, `Artifact`, (2) fill out their attributes, (3) and establish the links between the concept instances. For illustration, we show in Figure 2, the addition of an agent record into the `Agent` table.

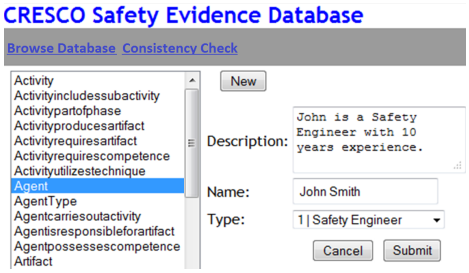

Fig. 2. CRESCO User Interface

2.2 Consistency Checking

The consistency check is a means of verifying that the state of the database is in accordance with the multiplicity constraints defined in the conceptual model. The consistency check is derived from the multiplicities of UML associations. We have chosen not to preserve the multiplicities in the database schema, where all associations are represented as many-to-many. This flexibility is required so that we can tolerate inconsistencies during the construction of the database. Trying to maintain consistency at all time would be intrusive, as this would enforce an unnecessary order on how the evidence items have to be entered. While our choice allows more freedom for the user when adding entries in the database, it also calls for the implementation of a consistency checker, to verify that the data in the database is in accordance with the constraints defined in the UML class diagram. For example, an `Activity` must have at least one `Agent` who is responsible for carrying out this activity (defined in the `Agentcarriesoutactivity` table shown in Figure 2). Such constraints are checked by CRESCO's consistency checker and any violations are highlighted to the user for further investigation.

3 Implementation and Availability

CRESCO is implemented in Eclipse for Java Enterprise Edition. We use two plugins, one for Kermeta that is used for the M2M transformations and the other is MOFScript for the M2T transformations. The M2M and M2T code are approx. 800 and 1500 lines, respectively. The total number of lines of code generated depends on the size of the input conceptual model. For the IEC 61508 model, the resulting code was in excess of 20,000 lines. Hence, significant manual effort can be saved by applying CRESCO. We use Apache Derby as the underlying database which is accessed by the Java code via Hibernate. The user interface for populating the database is via the web and we use Apache Tomcat as the web server and JavaServer Pages, Apache Struts and JavaScript to present and manipulate the objects residing in the database as well as to provide the navigation of the user interface. For our demonstration, we will present the import process of the conceptual model, the execution of the the two transformations and the user-interaction with the web-based user-interface. Due to space restrictions, we do not describe the technologies underlying CRESCO. See [1] for details and references. CRESCO is freely available at `http://home.simula.no/~rpanesar/cresco/`.

4 Conclusion and Future Work

We presented CRESCO a tool for the generation and manipulation of evidence repositories for demonstrating standards compliance during certification. CRESCO provides a centralized repository for keeping diverse data which, in the current state of practice, is often not collected systematically and needs to be extracted and amalgamated during certification. Our goal was to show feasibility via a coherent combination of existing open-source technologies. While our current tool provides a flexible infrastructure for managing compliance evidence, further work is required to turn it into a tool that can be deployed in a production environment. In particular, we are considering adding more sophisticated query facilities such that complex queries can be posed as well as professional reporting facilities in order to extract data from the database to create reports that can be directly given to the certification body.

References

1. T. Knutsen. Construction of information repositories for managing standards compliance evidence, 2011. Master Thesis, University of Oslo. `http://vefur.simula.no/~rpanesar/cresco/knutsen.pdf`.
2. R. Lewis. Safety case development as an information modelling problem. In *Safety-Critical Systems: Problems, Process and Practice*, pages 183–193. Springer, 2009.
3. R. K. Panesar-Walawege, M. Sabetzadeh, and L. Briand. Using UML profiles for sector-specific tailoring of safety evidence information. In *ER'11*, LNCS, 2011.
4. R.K. Panesar-Walawege, M. Sabetzadeh, L. Briand, and T. Coq. Characterizing the chain of evidence for software safety cases: A conceptual model based on the iec 61508 standard. In *ICST'10*, pages 335–344, 2010.
5. F. Redmill. Installing IEC 61508 and supporting its users – nine necessities. In *5th Australian Workshop on Safety Critical Systems and Software*, 2000.
6. UML 2.0 Superstructure Specification, August 2005.