A Domain-Oriented Software Development Environment for Cardiology

Kátthia M. Oliveira¹,², MSc       Ana Regina Rocha¹, DSc       Guilherme H. Travassos¹, DSc

¹COPPE - Computer Science Department - Federal University of Rio de Janeiro - Brazil
²UCCV/FBC -Cardiology and Cardiovascular Surgery Unit/Fundação Bahiana de Cardiologia – Federal University of Bahia - Brazil
e-mail: {kathia, darocha, ght}@cos.ufrj.br

Software teams have encountered difficulty to develop software when they are not familiar with the problem domain. We realized this fact while developing software for cardiology during the last five years in UCCV/FBC. The lack of cardiological knowledge by the software team was a crucial problem during the software development. Moreover, domain experts (cardiologists) view the process of knowledge and requirement elicitation as boring and stressful in particular because they need to explain basic concepts of cardiology and patient health care to the computer science personnel for each new software development project. This problem is made worse by the high turnover of the software team, common in projects.

To address this problem we have defined Domain-Oriented Software Development Environment (DOSDE)¹ and are currently building it for the cardiology domain. DOSDE supports the development of software systems in a specific domain considering the embedded knowledge of this domain to guide the software developers across the several phases of the software process.

This new class of software development environment requires two essential features: (a) having domain-knowledge and, (b) using this knowledge during the software development. To support these features we have first defined Domain Theory as the domain model that contains the knowledge to assist the software developers. A Domain Theory uses a domain ontology² and contains an identification of tasks mapped with the domain knowledge defined. These tasks represent activities that take place in the specific domain (e.g. diagnosis). While defining the domain ontology, the Domain Theory can be divided in sub-theories since parts of the domain knowledge are in different context and different level of abstraction. The following sub-theories were defined for cardiology: (i) heart anatomy (concepts about the heart structure and the physiology), (ii) findings (concepts that are used in the physician’s investigation process), (iii) therapy (general kinds of therapies and their features), diagnosis (concepts and characteristics that identify syndrome and etiology diagnoses); and pathologies (representing different situations of heart components but whose classifications and features are important for the purpose of the software development). The tasks identified for cardiology are: diagnosis, therapeutic planning, simulation and monitoring. The mapping provides an idea of the concepts more closely related with the task and to which the software developers should give special attention. For example, for the diagnosis task, it is important to consider findings, pathology and diagnosis sub-theories.

The use of the Domain Theory during the software development (feature (b)), is defined by a specific sub-activity named domain investigation. This activity was introduced in a standard process³ defined for UCCV/FBC. This standard process will be used to create any software processes to be automated by a DOSDE for cardiology at UCCV/FBC.

To construct DOSDE we are using TABA⁴, a meta-environment developed at COPPE that allows the creation (i.e. instantiation) of diverse SDE. For this purpose, a number of new requirements were included to TABA, such as: assist in the definition of a Domain Theory; allow the description of tasks; and allow tools to be built using this theory.

We are working on the construction of those extensions and some of the tools necessary to make the DOSDE for Cardiology a reality. We know that much work remains to be done but we do believe that using domain knowledge in software development environment will become an essential factor for software design activity.

References