Fifth International Conference
Modelling and Development of Intelligent Systems
June 23 - 25, 2017
"Lucian Blaga" University
Sibiu - Romania

Plenary Lecture

Decision Model for a Cycle Computer Developers Environment

Detlef Streitferdt

Department of Computer Science and Automation
Technische Universität Ilmenau
Ilmenau, GERMANY
E-mail: detlef.streitferdt@tu-ilmenau.de

Current software development efforts are required to address very short development cycles for complex systems with high demands on the quality of the resulting product. At the same time the environments for software developers are manifold and are getting more and more complex as well, what requires relevant additional efforts to setup and maintain developers environments. Development methods and processes as well as the required and corresponding tools (to the development steps) are part of such developers environments.

Within a university project our research group developed a Cycle Computer. The developers environment for this project is based upon a large number of decisions for an optimized development process and tool set. This contribution introduces and explains the details of the Cycle Computer projects developers environment.

The ontology decision model is based on the PhD work of Franz Füßl with five ab- straction levels to capture and maintain constraints, interconnect them and use the model for automated decisions using deduction and ontology learning. This models extends trope- based (existentially dependent elements) ontologies by the inclusion of arbitrary metrics (e.g., based on measurements) and social factors.

At the most abstract 4th level the model hosts data sources representing very simple issues which are captured with corresponding multiple-choice or single-choice questions. Data sources may also use sensor values or measurements. The 3rd level includes the fea-tures of a project (e. g. budget, operating system or personal motivation). The features are connected to at least one data source element. Each feature is measured on a nominal, ordinal or metric scale which also corresponds to the connected data source element type. Features are connected to the cells on the 2nd level. Cells generate knowledge based on the connected features. The 1st level hosts items (e. g., requirements engineering, software architecture pattern) which use the information stored in the cells to model abstract compo- nents for the solution. Finally, solutions are at level zero and represent developer packets to be used in a given development effort. The selection of the solution packets is based on their feasibility to fulfill the items of the first level. The knowledge model is a directed graph. Arbitrary associations can be realized in this graph. Currently five associations have been defined (is path, has path, can path, part-of path, and used-for path) and fully realized in a software tool.

The Cycle Computer is a project for students up to PhD level. Embedded components, Android Apps and Windows (C++, C#) components are integrated into it. For such a large project many different tools and development processes need to be interconnected seam- lessly. With our ontology-based decision model the student ideas and preferences for the tool and process landscape can be captured, modeled and used to reason about specific components of developers environments.

At the data source level questions regarding the team roles, the experience level with technologies like Android, the MSP430 microcontroller or Bluetooth are asked, together with questions about the motivation or interdisciplinary knowledge are asked. Thus, a team specific view can be built with such questions. The results are used on the following feature level. Here, results are generated based on the given answers. The cells of the next level have been used to represent the possible answers for a feature with "isn't-it" relations and the feature itself uses the "has" relation to the corresponding questions. On the item level concepts like the "development method" as a whole or "requirements engineering" as part of the method are modeled. To model the knowledge and interconnect it to the answers of the students, items are connected via further cells to the cell – feature – data source path. As an example the feature "Scrum" is a "development method" and it needs the "feasibility" feature what cannot be fullfilled by "undisciplined" teams.

The deduction process is supported by algorithms built into the model. Currently, the most complex query, the "find"-algorithm is used to find tools compatible with the team situation. The situation was given by answering the questions. Now we can ask questions like "Find elements to prioritize requirements". The resulting tool is the best / optimized fit according to the above set of answers. To cover the continuously changing body of knowledge the ontology model is able to learn by weights at the edges of the model. The weights can then be adapted / adjusted by e.g. inductive reasoning.

Based on the Cycle Computer developers environment the improved acceptance of the individually "selected" developers environment can be shown and explained. The selection of a developers environment can be traced back to ontology based decisions, the knowledge model. The future goal is the further automation of the selection process for complete developers environments.

Brief Biography of the Speaker: Detlef Streitferdt is currently senior researcher at the Ilmenau University of Technology heading the research group Software Architectures and Product Lines since 2010. The research fields are the efficient development of software architectures and product lines, their analysis and their assessment as well as software development processes and model-driven development. Before returning to the University he was Principal Scientist at the ABB AG Corporate Research Center in Ladenburg, Germany. He was working in the field of software development for embedded systems. Detlef studied Computer Science at the University of Stuttgart and spent a year of his studies at the University of Waterloo in Canada. He received his doctoral degree from the Technical University of Ilmenau in the field of requirements engineering for product line software development in 2004.