Model-Driven and Domain Specific Language Paradigms in Information System (Re)Engineering Projects

Sonja Ristić

University of Novi Sad, Serbia
sdristic@uns.ac.rs

Abstract— An information system (IS) serves the different concerns of numerous stakeholders. Its design and implementation involve understanding of the social and organizational context of the system and making decisions according to the limitations of environment and technology. The open, dynamic and almost unbounded nature of contemporary environment of the ISs poses many new challenges. The need for reengineering of information systems has dramatically increased as legacy ISs are migrated to new platforms. Reengineering of an IS has the objective of extracting the contents, data structures, and flow of data and processes contained within existing legacy IS in order to reconstitute them into a new form for subsequent implementation. Inability of traditional information system (re)engineering approaches, methodologies and tools to cope with ever increasing complexity of contemporary information systems leads towards paradigm shift. Here we discuss an approach to IS (re)engineering that is based on Model-driven (MD) and Domain Specific Language paradigms. MD paradigm assumes orientation on models at all stages of system development and addresses complexity through abstraction. It promotes the idea of abstracting implementation details by focusing on: models as first class entities and automated generation of models or code from other models. MD software development approaches usually rely on Domain-Specific Languages (DSLs) that are specific to the certain domain of application. By providing developers and domain experts with a DSL, it is possible to reduce the accidental complexity. MD and DSL paradigms applied in information system (re)engineering projects increase: the development speed through automation and single point of system definition, the software quality, and component reuse. Their application reduces conceptual gap between problem domains and software implementation, improves manageability of complexity through abstraction, enables greater domain expert inclusion in the development process, and supports better communication between different stakeholders. The role of MD and DSL paradigms in information system (re)engineering is illustrated on the example of IIS*Studio development environment aimed at MD information system (re)engineering.