

Towards Architectural Design Method for Rich Web-based Applications

Nalaka R. Dissanayake, Simon Courtenage, and Alexander Bolotov

Software designing plays an important role in software engineering, providing a blueprint for the solution to be implemented. Software design activities define the architecture, components, interfaces, and other characteristics of a software system. The software design process is divided into 1) preliminary design, which produces high-level architectural design, and 2) detailed design, which focuses on low-level details.

Software architecture provides an abstract overall picture of the system and assists in realizing the system. Architecture is the foundation of any software system, and the support gained from a carefully designed sound architecture is significant, throughout all the phases of a Software Engineering project. The increased realization is helpful in reducing complexity since the meaning of software complexity encloses the difficulties in understanding.

Through an intensive literature survey, we have identified methods and tools, used to design software architectures. Architectural designing may use either informal box-and-line drawing, or formal methods and tools. Formal software architecture description languages are available like Acme, AADL, C2, SBC-ADL, Darwin, Wright, Rapide, which follow the architectural description standards like ISO/IEC/IEEE 42010:2011. There are domain-specific architecture description languages such as EAST-ADL and EADL for embedded and real-time systems, DiaSpec for control-loop applications, Koala for product line architectures, and Π -ADL for dynamic systems; and also analysis-specific architecture description languages like Fractal for behavioural analysis, and TADL for trustworthiness analysis. We have also identified software architectural design languages: C4 model, ArchiMate, Arch42, SAP's TAM, SysML. Some of these are based on the aforesaid architecture description languages, and some are aligning with UML. These languages provide visual designing notations and guidelines for software architecture.

There are significant differences between these available software architectural design languages, and none of them is specifically dedicated to designing the architecture of Rich Web-based Applications. We present our research on identifying the weaknesses of these formal methods which will help us in introducing a dedicated architectural design language for the Rich Web-based Applications.