

## A pi-calculus ADL for semi-automatic code generation

---

ANASTASIOS PAPAPOSTOLU, DIMITAR BIROV

*Department of Computing Systems,  
Faculty of Mathematics and Informatics  
Sofia University st Kliment Ohridski, Sofia, Bulgaria*  
papapostol@fmi.uni-sofia.bg, birovfmi.uni-sofia.bg

Software architectures are used to describe the structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time. The runtime behaviour can be presented by the component-and-connector architectural view where components represent the primary computational units and connectors represent the communication between components. In order for software architectures to be expressed (design, evaluation etc.) formal description languages are used, called Architecture Description Languages (ADL). Although ADLs could be proved to be a very powerful tool in the hands of architects, for a number of reasons (e.g. too formal language constructs), in practice they are not used by software developers in development process these days.

In this paper we present a new ADL, named jADL, which targets on automatically generating implementation code stubs providing a software tool not just for software architects but for developers as well, that ensures the consistency between the architecture and the actual implementation. jADL is a well-formed extension of a process calculus developed for mobile processes typed polyadic asynchronous pi-calculus. The pi-calculus is a process algebra for communicating processes with a dynamically changing topology which have been used for specification of concrete software architecture topology and behavior during run-time of a software system.

In this paper we briefly will present the principles and notation of jADL through simple examples. We will present pi-calculus extensions and we will establish strong type system useful for code generation.