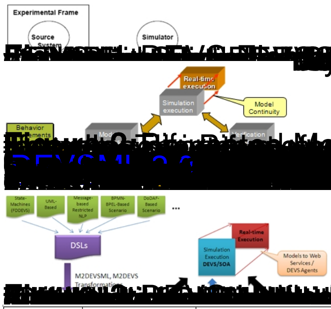
Book now out : Netcentric System of Systems Engineering with DEVS Unified Process, Francis & Taylor Group, CRC Press

Comprehensive book review in SCS Newsletter , by Dr. Bernard P. Zeigler

The DEVS Unified Process is based on an Open Systems concept. An open system is a dynamical system that can exchange energy, material and information with the outside world through its reconfigurable interfaces. An open system also possesses the capability to form complex hierarchical structures enabling them to compete and cooperate at the same time. In fact, the mechanism to reorganize in a hierarchical structure is one of the basic requirements to manage complexity. The open systems are also characterized by emerging behavior and evolving structure. These two facets are function of an open system's permeability to outside influence, inherited guidelines, ability to self-govern, and the degrees of synergistic efforts as it interacts with other systems and with its environment. In order to have an executable System of System, the framework must provide capabilities to model an open system. In addition, a process also needs to be defined that allows the development of an executable Open system. Much of an Open system development hinges on the variable structure capability within a component based system. The ability to add or remove hierarchical components, change connections between components and lastly, modify the behavior of a component as it evolves per its surroundings, is the desired characteristics of an open systems modeling framework. While the first two capabilities are structural in nature and have been documented in DEVS literature, the third one is behavioral modification at runtime. This capability is the most difficult to achieve. Using the latest advances in finite deterministic DEVS described as DEVS Domain specific Language in DEVSML 2.0 stack , runtime behavior modification in DEVS could be achieved. The DEVS open systems approach underlying the DEVS Unified Process gives it strong formal foundation to develop M&S complex systems software capable of designing emergent behaviors.

Service Oriented Architectures (SOA) present challenges to current model-based software engineering methodologies such as Rational Unified Process (RUP). In this effort, originally developed by Dr. Mittal in his doctoral research, a process called DEVS Unified Process (DUNIP) has been proposed. It uses the Discrete Event Systems Specification (DEVS) formalism as a basis for automated generation of models from various requirement specifications and their eventual realization as SOA collaborative services.



Phase	Agile Methodology	DEVS Unified Process
Model	Identify the domain and business use-case requirements and specify in DSLs such as UML, etc.	DUNIP begins by taking requirements in various DSLs and through M2DEVS, M2M and M2DEVSML transformations, transform them into FDDEVS specifications. It also constructs various Experimental Frames from requirement specifications for later Verification and Validation operations.
Implementation	Transform your models into executable code with running unit-tests	From PIMs, the DUNIP engine generates code in platform specific models (PSMs) such as Java, C++, C# etc. With strong DEVS theory underlying each atomic model, they can be mathematically verified. Unit-testing for each transition or an event is inherent in DEVS.
Test	Identify defects, ensure quality and verify requirements	The development of Test-suite is done in parallel with that of the DEVS PIM. The Test models verify the atomic model's operation at various levels of system specifications, such I/O pair, I/O function [illegible] this stage
Deployment	[illegible] end users	[illegible] deployment capabilities per model-[illegible] principles to SOA infrastructure, and zero transition times, the model is the actual software is readily moved to the production servers
Configuration Management	Managed access to project artifacts	DUNIP is very well positioned to reuse and contribute to model repository. PIMs are strong contenders for such tracking and version management. PSMs can very well be source versioned using tools like Subversion.
Project Management	Manage people, project iterations and budget	These qualities are universal and due to the [illegible] of DEVS [illegible] plan can [illegible] partitioned [illegible] and milestones
Environment	Ensure the proper process guidance and tools are available for the team	DEVS has been in existence for over 30 years and there is a large community support in basic theory and toolsets

Dissertation

- Overview
- DEVS Unified Process for Integrated Development and Testing of Service Oriented Architectures
- Table of Contents
- Prototype Demonstration movie (.avi 200MB)