KRATES
inseneribüroo • engineering bureau

Next steps from the developers viewpoint

Tõnu Näks
IB Krates, Estonia
23/09/2009

www.krates.ee

Challenges

- DO-reglemented development v.s. open community
  - Management of the requirement process
  - Consensus v.s. different user groups
  - Qualification v.s. flexibility
- Scalability and scope
  - Dedicated scope v.s. “critical mass” of functions
  - Size and intent of documentation
  - The scope and level of verification activities
- Financing and business models
  - Consortium-based development v.s. traditional vendor-user relation
  - Return on investment for users and developers!
- Organising the development
  - Critical mass of developers!
  - Responsibility distribution

Development goals

- Long term
  - Performing exhaustive verification
  - Qualifying the toolset for industrial usage
- Short term
  - Increasing the robustness of the tool
  - Performing verification and improving the documentation
  - Extending the functionality
  - Growing the user base
  - Identifying requirements on specialised tool-chains

Ongoing tasks (robustness, verification, features)

- Maintenance for Airbus France and EADS Astrium
- Ada language backend with AdaCore
- Support for community and parallel projects (e.g. EDONA)
- Dissemination and marketing

Increasing the user base.

- Primary usage
  - Qualified code generator for Simulink/Scicos/??? models
  - Long term maintenance guarantee supported by open-source nature of the tool
  - Platform for making customised tool-chains
- Secondary usage (not targeted, but possible)
  - (Low-cost?) replacement for RTW in code generation
  - (Low-cost?) replacement for RTW in simulation support

External contributions

- Active contribution – dedicated development / maintenance
- Passive contribution – flat fee maintenance (http://geneauto.krates.ee)

Extending the functionality

- SysML importer
  - Goal: provide an alternative input (ideally free from vendor restrictions)
  - Tasks: define suitable subset of SysML, develop the importer
- Simulink exporter
  - Goal: make the importer independent from Simulink/Stateflow file syntax
  - Tasks: develop a Matlab script that converts model to GASystemModel language
- Support for subset of Matlab language
  - Goal: support functions defined in Matlab language
  - Tasks: define approach for handling function interface (block inputs and parameters, Matlab workspace), define the supported subset, develop the transformation

Extending the functionality

- Verification tool(s) to check the conversion from system model to code model
  - Goal: Facilities for verification of the model either to complement model editor verification (e.g. Simulink) or to check transformation
- WCET-based optimisation
  - Goal: Improve the performance of generated code
- Support for target adaptation
  - Goal: Allow target-specific optimisations either based on RTW parameters or separate configuration mechanism

Supporting infrastructure

- Automating the testing environment
  - Work ongoing at IB Krates (basic testing framework in use)
  - The primary goal is the validation of requirements
  - The long-term goal is to support requirements-based testing
- Development of qualification documentation
  - Work ongoing in scope of EADS demonstrators project (small scale updates)
  - Experimentation on Wiki-based environments together with AdaCore (the AdaPrinter project)
- Change management and community support
  - Currently change management tasks spread in different environment GForge, gPM, Mantis, gPMPlus …
  - The goal is to provide a single front-end for bug/feature request reporting, costing and change management

The testing environment (*status quo*)

- A framework for composing and executing test scripts
- Test scripts composed of steps, each performing one elementary operation using an appropriate agent
- Existing agents:
  - Code generation (C + Ada)
  - Code compilation and linking (C + Ada)
  - Code execution
  - Comparison of generated code with reference code
  - Comparison of execution result with simulation result
- Matlab script for generating test vectors from Simulink model
- Test scripts compose a testsuite that can be executed fully automatically
- Isolated test server (controlled through a web interface) to guarantee independence of development environment

The testing environment (plans)

- Integration with requirement management environment to form a “qualifying machine” (e.g. using FitNesse)
- Improved reporting
- Service to execute tests with contributor patches
  - Allows contributor to upload planned change to test server and get feedback if it passes all tests in the main trunk

Qualification documentation (points to consider)

- Significant amount of documentation is still to be developed or updated to qualify the toolset
- Document-based approach is difficult to handle – need for requirement management environment (preferably open-source)
- There is some evidence that requirements will conflict for different types of users
  - Strict limitations v.s. large user base (e.g. type inferencing)
  - Possibility to exclude unnecessary functionality
  - Preliminary experiments on-going with maintaining parallel sets of requirements and allowing to compose tool configurations corresponding to each requirement set on source-code level

Follow-up projects

- OPEES
  - Participation decided, preliminary feedback from local PA positive
  - Requested to show a “tangible result” (i.e. new features, not just qualitative updates and new business model)
  - Financing and return on investment?
- OpenDO (http://www.open-do.org/)
  - Participation through developing components for the “qualifying machine” and experiments with AdaPrinter requirements
- Gene-Auto2 ?
- ESA ?