

eTrice 0.2.0 Release Review



Planned Review Date: 10.10.2012

Communication Channel: eclipse.etrice

Thomas Jung, Henrik Rentz-Reichert, Thomas Schütz

Introduction

eTrice provides an implementation of the ROOM modeling language for event driven real-time systems together with editors, code generators for Java and C code (in the future also C++) and exemplary target middleware.

The model is defined in textual form (Xtext) with graphical editors (Graphiti) for the structural and behavioral (i.e. state machine) parts.

Features: Model Level Inheritance

- The model level inheritance of ROOM is implemented for DataClasses, ProtocolClasses and ActorClasses.
- All editors fully support inheritance.
- All code generators fully implement model level inheritance.

Features: ANSI-C Code Generator

- Additionally to the Java an ANSI-C code generator and a C runtime has been developed.
- A concept has been developed that allows to share the most complex parts of the generator for all target languages.
- Model level inheritance is fully supported by the C generator.
- Special emphasis has been laid to a small footprint of the generated code.

Non-Code Aspects

- Online documentation (Eclipse Help) is provided and available also in PDF form.
- Several Tutorials are available from basic to more advanced and comprehensive.
- Tutorial code is shipped with the eTrice bundles and is easily installable to the user's workspace.
- Tutorials using C as target language have been added.

APIs

- eTrice is a modeling tool, not a framework. Therefore it is not very API centric.
- A typical way to extend eTrice will be to create new generators for new languages together with a corresponding middleware. Therefore we extracted major functionality of the generator including model traversing algorithm into the target language independent generator model and helper classes.

Architectural Issues

- eTrice has been built on top of the popular EMF and Xtext and the new Graphiti framework.
- Emphasis has been laid on a simple and sound architecture. Continuous refactorings have been done to keep the architecture clear and concise.
- eTrice is still in its very beginning. The ways it can be adopted and extended will evolve together with a growing community of users.

Tool Usability

- From the very beginning simplicity of use has been a major goal of the eTrice project.
- ROOM as a modeling language allows the user to solve typical problems of the development of highly concurrent real-time systems on a higher level of abstraction.
- All editors, textual and graphical are simple to use and designed to support a fast development.

End-of-Life

- eTrice is still under heavy development. In some cases we couldn't avoid even changes in the basic ROOM language. But our community still is very small and thus we don't have to worry too much about compatibility issues.

Bugzilla

- All bugs scheduled for the 0.2.0 version of eTrice

Standards

- The modeling language ROOM is not standardized but was published in [1].
- [1] defines a graphical notation which was adopted.
- [1] also contains a textual notation. But this is incomplete and we decided to develop our own notation based on this.

[1] Bran Selic, Garth Gullekson, Paul T. Ward:
Real-Time Object Oriented Modeling, New York:
John Wiley, 1994 (ISBN 0-471-59917-4)

UI Usability

- The Eclipse UI Guidelines have been adhered to.
- No language packs have been provided since the user community is used to tools in English language only.
- eTrice delivers the very good UI usability of the underlying Graphiti and Xtext frameworks

Schedule

- The two milestones that have been planned for this release have been met on time.
- The originally planned release date had to be shifted due to limited resources. But since the main consumers of eTrice already follow the most recent development the exact release date is not so important.

Communities

- Since its start in October 2010 the project has won one additional committer and several contributors.
- Three Google Summer of Code and one bachelor student had projects related to eTrice.
- Talks have been given on a number of conferences.
- eTrice is already used as part of the tool chain in industry projects.
- Up to now eTrice users and developers mainly use direct communication. Therefore the newsgroup and the development mailing list are only rarely used.

IP Log

- The Eclipse IP policies and procedures have been followed
- The eTrice IP Log can be found at http://www.eclipse.org/projects/ip_log.php?projectid=modeling.mdt.etrice
- A frozen version can be found at <http://eclipse.org/etrice/pages/etrice-ip-log-0.2.0.htm>

IP Issues

- The EMO explicitly asks during the Release Review if any Member would like to assert that this release infringes their IP rights.
- If so, the EMO and the project will follow the Eclipse IP Policy in discussions with that Member.

Credits and Kudos

- Kudos to all who contributed to the project, be it in form of feedback, suggestions, questions or most valuable code contributions