

Draft

Eclipse Project Proposal eTrice

The eTrice project is a proposed open source project under the [Eclipse Modeling Project Container Project](#).

This proposal is in the Project Proposal Phase (as defined in the Eclipse Development Process) and is written to declare its intent and scope. We solicit additional participation and input from the Eclipse community. Please send all feedback to the [eTrice Forum](#).

Background

Domain Specific Languages (DSLs), whether they use a textual or graphical notation, are widely recognized as a convenient way to formalize abstractions of a given domain. They allow the engineer working on a problem in this domain to express his solution on exactly the level of abstraction which is best suited for the domain and best adapted to the way he states the problem and thinks about a solution.

For event driven real-time embedded systems engineers traditionally tend to use general purpose (and sometimes low level) programming languages like assembler, C and (usually a restricted subset of) C++. This way they have maximum control over the memory and performance impact of every single detail of their system. But the exponentially growing complexity of current software systems in the domains of telecommunication, automotive and industrial automation calls for advanced engineering methods. The challenge is to master the new complexity of highly distributed, concurrent and large software systems while maintaining quality and reliability.

In recent years UML has been adopted for modeling in many areas including embedded systems. But while UML1 lacked concepts for abstraction, e.g. the notion of ports and components, which are agreed to be essential for distributed systems, UML2 suffers from its mere complexity which is owed to its general and domain independent applicability. Additionally there is a gap between the UML model and the semantics of the code produced from it which is not addressed by the standard. State machines and ports leave many degrees of freedom. So vendors of UML2 modeling tools independently filled this gap while narrowing down the abundance of modeling elements by stereotypes.

In the early 1990s a DSL called [\[ROOM\]](#) (Real-Time Object Oriented Modeling) was proposed by Bran Selic et al. This language was designed for the, at this time, most complex real time systems developed for the upcoming mobile telephony. In a way, ROOM can be thought of a specific implementation, including the semantics of the model, of a subset of UML2 for the very specific needs of those kinds of systems - even though ROOM was proposed before the advent of UML. In its relation to UML2 ROOM is in a similar position as EMF compared to MOF where the design was guided by the principles of simplicity and performance. ROOM was tailored to suit the needs of a very special class of problems and uses concepts that address them in a highly formalized way. Maybe it is worth to stress that ROOM is not limited at all to the domain of telecommunication. It is general enough to be applied to any distributed event driven real time system.

The proposed eTrice project aims at an implementation of the ROOM language together with code generators and tooling for model editing. For the embedded target a portable runtime library will be supplied as part of the project which implements a platform abstraction, basic services like messaging, thread and process creation and a layer with the invariant parts of the modeling elements

Draft

Scope

The objectives of the eTrice project are to:

- provide an implementation of the real time modeling standard [\[ROOM\]](#)
- build exemplary but ready to use editors for ROOM models (textual and graphical)
- create code generators and portable target runtime libraries for Java and C++, later also for ANSI-C
- provide built-in support for modeling level debugging of the running target software: state machine animation, data inspection and manipulation and message injection
- provide built-in possibilities for sequence diagram creation from the running software
- support distributed heterogenous systems out of the box

Description

The eTrice project will strive to deliver state of the art tooling and target middleware for the model driven development of industry quality event driven real-time software. Emphasis will be laid on the usability of tooling, produced code and middleware in real industry projects.

The design will be guided by the following principles

- *Simplicity*: eTrice will be kept as simple as possible to fulfill its purpose
- *Conceptual Integrity*:
 - eTrice will offer only *one* solution for a given class of problems
 - eTrice will offer coherent architecture concepts
- *Extensibility*: eTrice can be extended or changed in various ways by
 - extending the ROOM metamodel
 - extending or adding Editors
 - extending or adding new codegenerators e.g. for other languages or other middleware
 - extending or replacing the middleware

The eTrice project will create

- a ROOM metamodel
- textual and graphical model editors for classes, structure, behavior, deployment and instance configuration
- code generators
- target middlewares as environment for the generated code

The ROOM Metamodel

The ROOM metamodel will be based on EMF. Since there is no industry standard the project adopts the language as it is described in [\[ROOM\]](#). The model will be completed by elements to describe deployment and configuration data. Also the model will be refined where appropriate as the project evolves. E.g. a concept of libraries will be introduced.

Model persistence will be in a textual, human readable form at a medium granularity. This provides an easy way for a team to lock some parts and merge others if necessary.

Draft

Model Editors

Initial editors will be implemented using Xtext. The concrete syntax will be similar to the one proposed in [ROOM] p. 493ff. Later graphical editors for structure (component hierarchy) and behavior (state machines) will be added using GMF or Graphiti.

Code Generators

Code generators will be based on Xpand and Xtend. Initially code generation in Java and C++ will be supported. Later also an ANSI C version will follow.

Target Middleware

The middleware will be supplied for exemplary OS/hardware combinations like Windows/PC and Linux/PC. A middleware will be provided for each supported OS/hardware combination and target language. Emphasis will be laid on easy portability.

Relationship with other Eclipse Projects

- eTrice will use EMF for the ROOM metamodel
- eTrice will use Xtext for textual editors
- eTrice will use Xpand and Xtend for code generation
- eTrice will use GMF and/or Graphiti for graphical editors

Literature

[ROOM]

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

Committers

The following individuals are proposed as initial committers to the project:

Thomas Schuetz, project lead, protos (www.protos.de)

Thomas has many years of experience in the development of real-time software for industrial automation, automotive and telecom applications. Together with Henrik and others he developed a ROOM tool which he used over a long period for modeling and code generation. He also has a sound knowledge in designing the middleware for event driven real-time software.

Henrik Rentz-Reichert, committer, protos (www.protos.de)

Henrik has many years of experience in the development of modeling tools. Together with Thomas S. and others he created a ROOM tool with code generators for C and C++ and a runtime library for the generated code. He has developed Eclipse RCP tools and successfully used technologies from the Eclipse Modeling Project like EMF, GMF, M2M/ATL and TMF/Xtext.

Draft

Mentors

The following Architecture Council members will mentor this project:

- N.N.
- N.N.

Interested Parties

The following individuals, organisations, companies and projects have expressed interest in this project:

- Thomas Jung, Tieto (www.tieto.com)
- Klaus Birken, Harman Automotive (www.harman.com)
- Bjoern Eschrich, PARIttec (www.paritec.de)
- Georg Huba, Infineon (www.infineon.com)

Project Scheduling

Changes to this Document

Date	Change
16-June-2010	Document created