xtUML: Current and Next State of a Modeling Dialect

Cortland Starrett
cortland.starrett@onefact.net

xtUML: Current and Next State of a Modeling Dialect

Shlaer-Mellor Object Oriented Analysis 1988-2016+

Cortland Starrett
One Fact Inc.
10412 US HWY 52 S
Clarks Hill, Indiana 47930 USA
cortland.starrett@onefact.net

***Abstract*— xtUML and its Papyrus-xtUML (BridgePoint) tooling have advanced further in the last two years than in the ten preceding years.**

Acceleration in usage and improvements in tooling have been fueled by the open source software (OSS) ecosystems of Apache, Eclipse and Papyrus. Development teams have transformed from licensed users into user-contributors.

Executable Translatable Unified Modeling Language (xtUML) is descended from Shlaer-Mellor Object Oriented Analysis (SM-OOA) and has survived the Method Wars, notation unification (UML) and tooling evolution. Sparse but focused deployment in the embedded control and distributed processing worlds has produced hundreds of years of accumulated application model intellectual property (IP).

This paper summarizes a brief history, current status and future of xtUML. It highlights recent advancements, forward direction, key players, related modeling dialects, architectural migration and implications to the metamodels and compliance with Papyrus Platform, UML2, fUML and Alf.

Keywords*— *xtUML, Shlaer-Mellor, Executable UML, UML-RT, MASL

openness, transparency and elimination of exclusive ownership fosters an environment of security. There is no single entity (licensing company) that must be trusted to serve the interests of all parties. Users can add features (or pay suppliers to add features) at will. In the xtUML community, features and fixes are being delivered faster than ever before. Related executable modeling languages and technologies are sharing in these improvements.

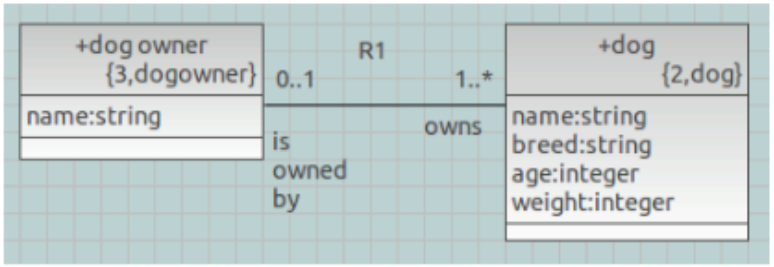
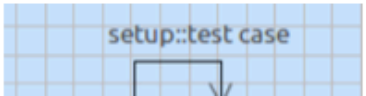


Figure 1: dog owner *owns* dog



II. Background

The Shlaer-Mellor xtUML method steps

Outline

- Introduction
- Background
- Brief History
- Key Players
- Current State
- Related Modeling Dialects
- Next State
- Conclusion

Introduction

PROJECT TECHNOLOGY INC.

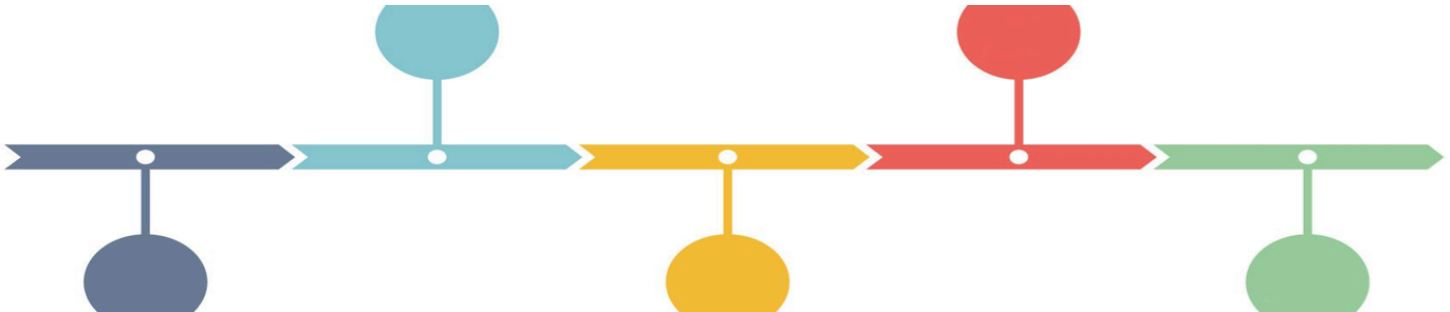
Shlaer-Mellor Method · BridgePoint

Background

- Shlaer-Mellor Method (xtUML)
  - subject matters, separation of concerns
  - data, control, processing
  - BridgePoint
- data modeling (Object Oriented Analysis (OOA))
- state machines
- action language
- interpretive execution
- model compilation

History

- 1988, 1991 Shlaer-Mellor Method published by Stephen Mellor and Sally Shlaer.
- 2002 Executable UML established as Shlaer-Mellor OOA using UML notation.
- 2004 Commercial Corporate Proprietary Licensed.
- 2013 BridgePoint xtUML Editor goes open source under Apache 2.0.
- 2014 all of BridgePoint (including Verifier and model compilers) goes open source under Apache and Creative Commons.
- 2015 Papyrus Industry Consortium and xtUML/BridgePoint contribution
- 2015 OSS of alternate generator engine (community building)
- 2016 Papyrus-xtUML (BridgePoint) Eclipse Foundation governance
- 2016 OSS contributions from industry, university and individuals

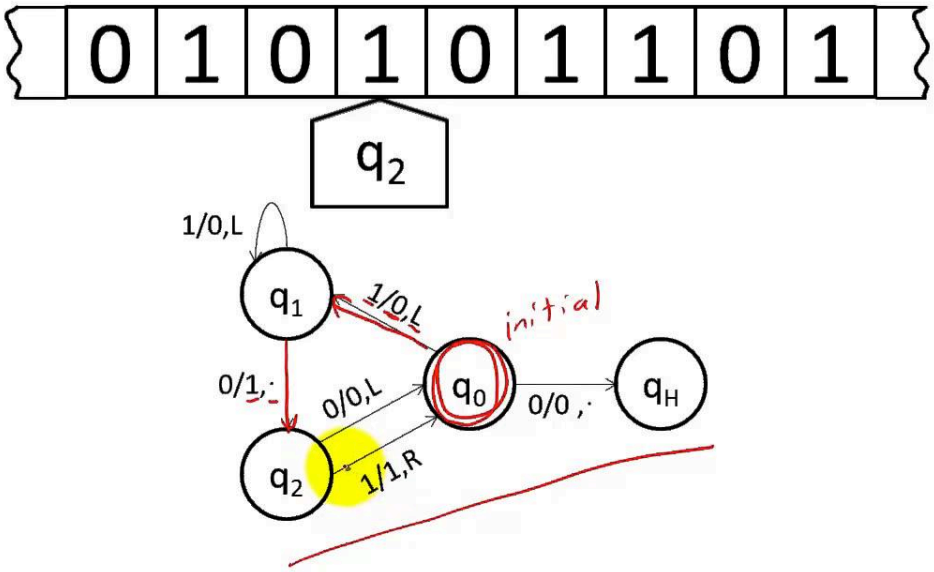


Key Players

- Saab
- UK Crown
- Agilent
- Ericsson
- Fuji-Xerox
- Academia

Current State

- body of IP
- self-hosting
- Papyrus (and Papyrus Industry Consortium)



Related Dialects

- MASL
- Alf
- UML-RT

Next State

- Papyrus Platform
- action language
- persistence
  - semantic model persistence manifesto
- hybrid textual/graphical
- type system (Darwen and Date)
- fUML and Alf
- roadmap: xtuml.org/xtumldayemd/

Conclusion

- method sound
- body of IP driving
- community growing
- tooling on the rise

Questions and Discussion

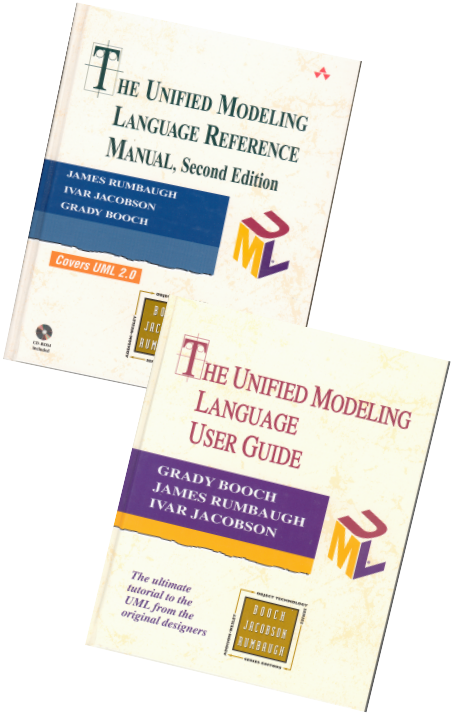
Papyrus-xtUML (BridgePoint)

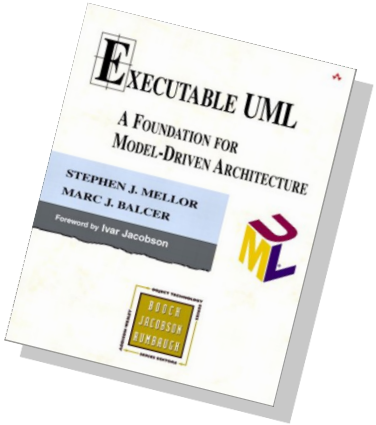
- Overview
- Demo
- OSS License
- Process for Contributing
- Quality
- Road Map
- Welcome Contributions

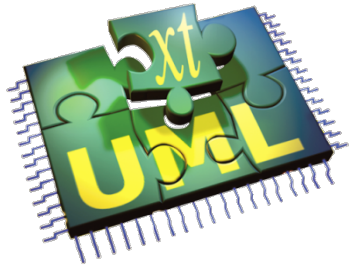
xtUML – Executable Translatable UML

Unified Modeling Language

- Industry standard *notation*
- *Family* of languages







"Executable UML"

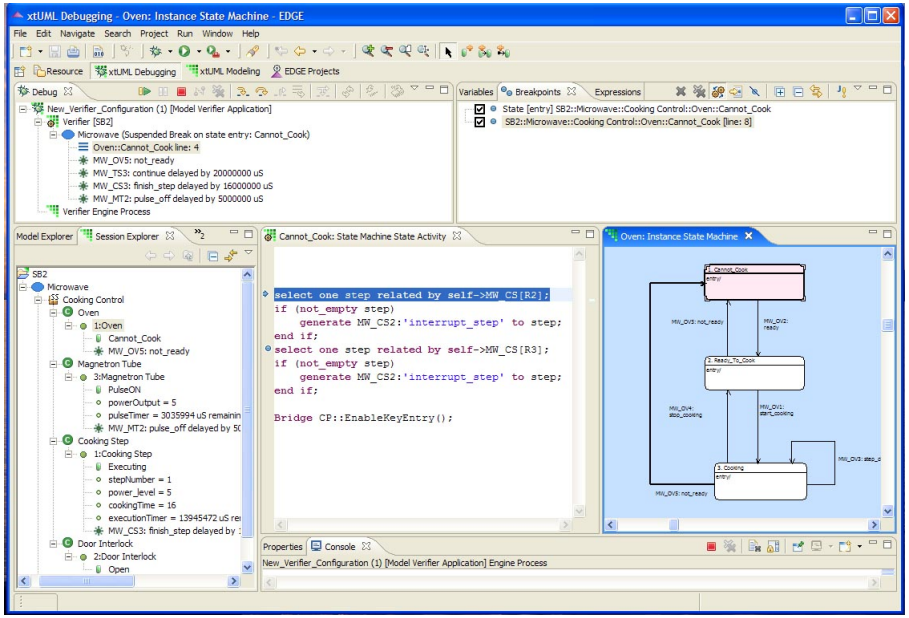
- Defines a method, including:
  - Semantics of diagrams
  - Relationship between diagrams
  - Execution rules
  - Order of construction
  - Path to implementation

Defect Removal through Execution (interpretive execution of partial models)

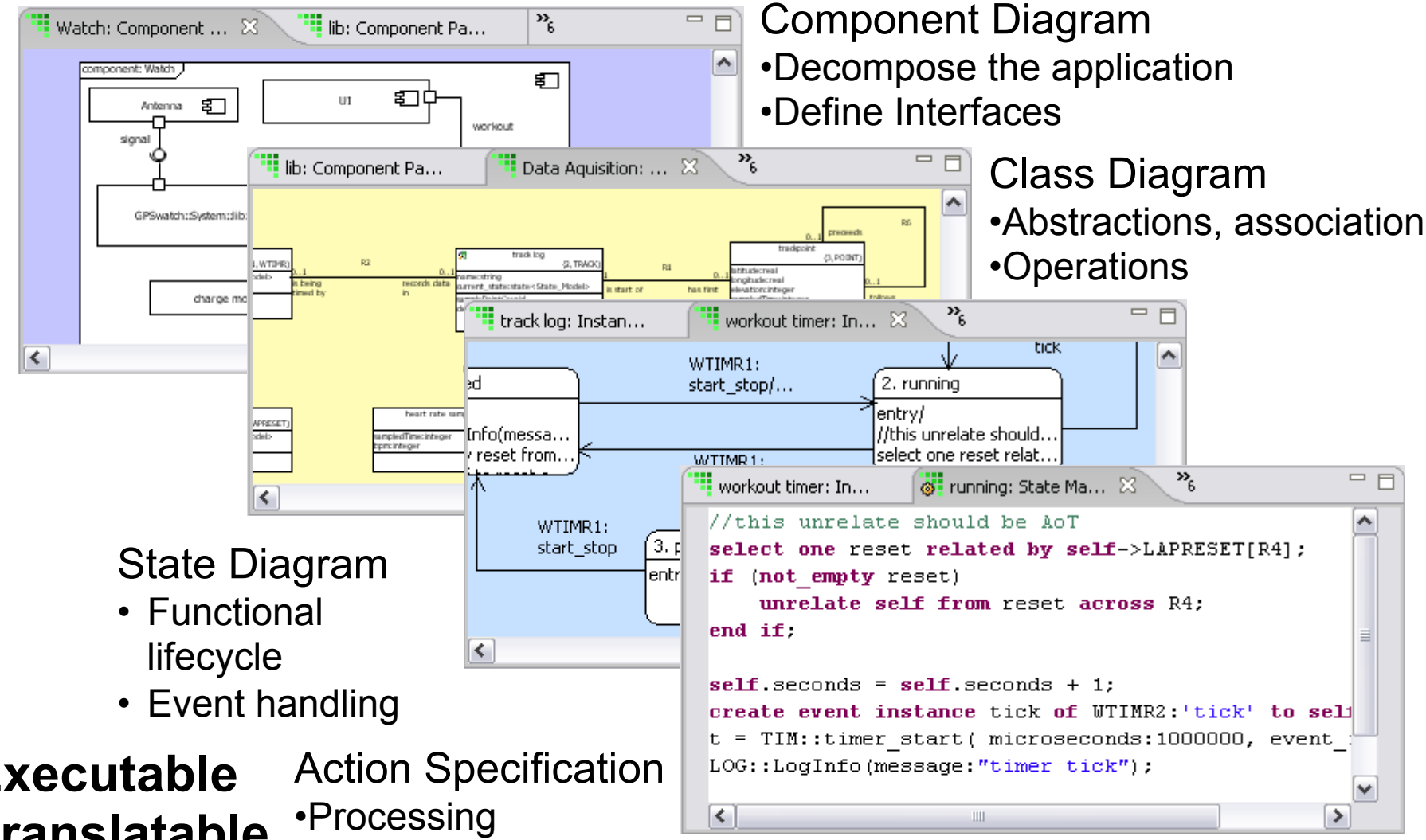
We find *many* defects through inspection, but…

…we find *the rest* by testing the system.

Executable models enable early and frequent testing.



xtUML Modeling Flow



Component Diagram
- Decompose the application
- Define Interfaces

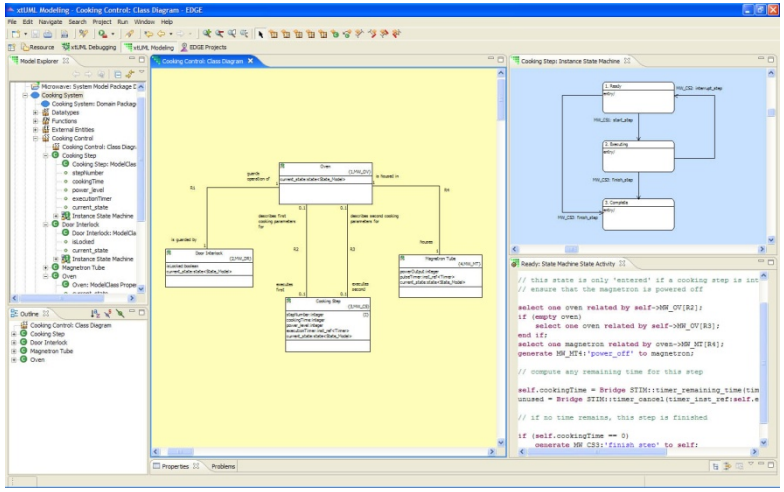
Class Diagram
- Abstractions, associations
- Operations

State Diagram
- Functional lifecycle
- Event handling

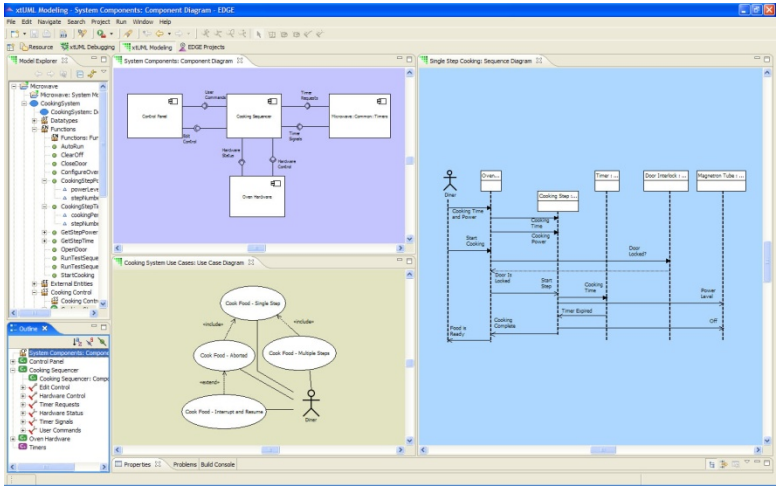
Executable Translatable

Action Specification
- Processing

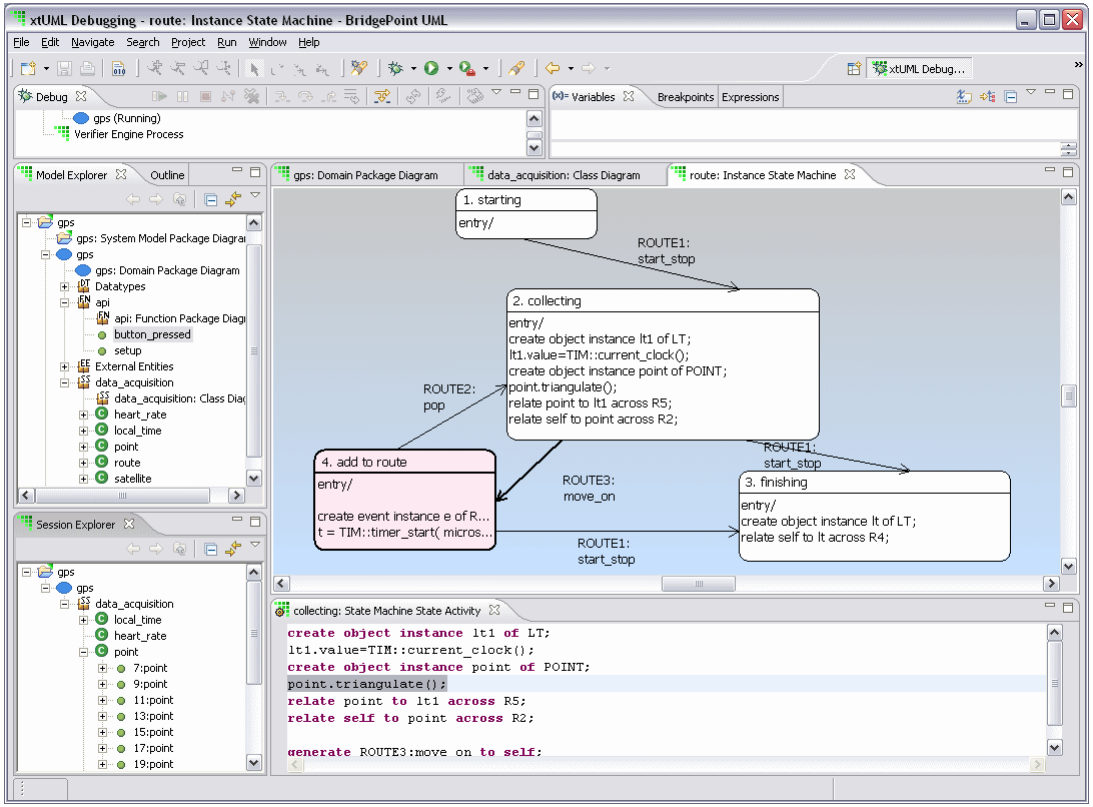
xtUML Editor



- Intelligent model entry, navigation
- Smart action-language editors
- Flexible configuration management



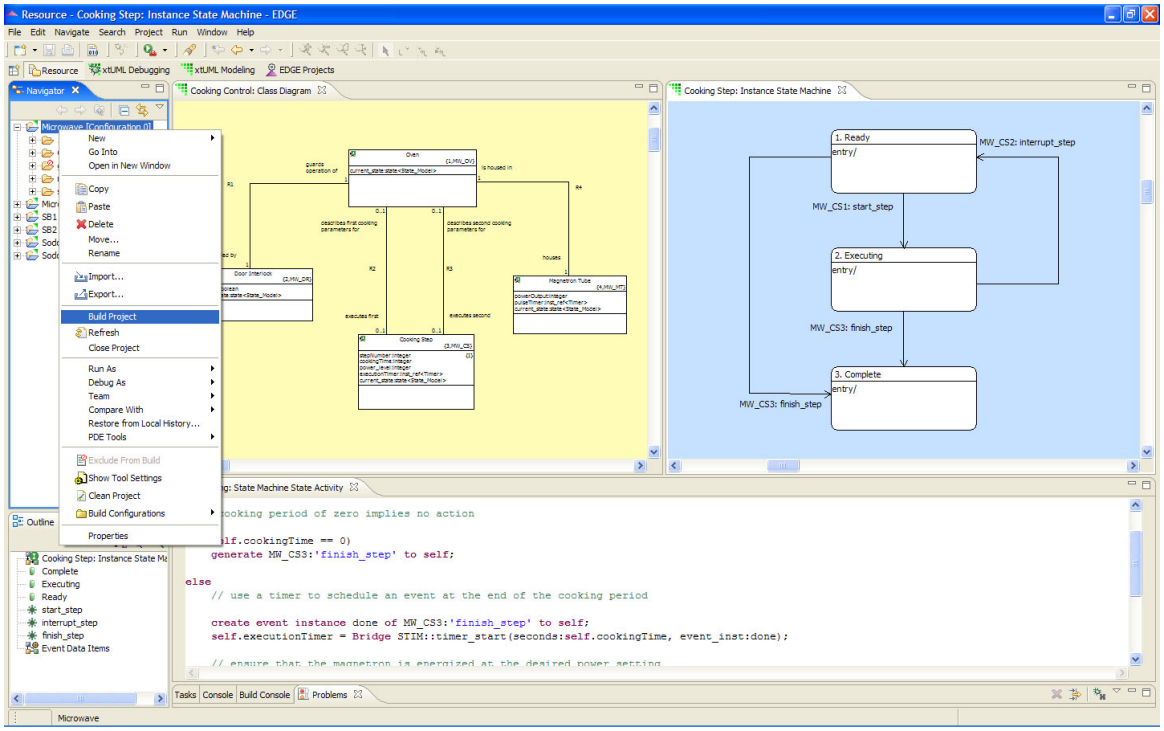
xtUML Verifier (Interpretive Execution)



- Interpretive execution
- Interactive debugging
- Animation of models
- Connect to legacy

- Execute models before generating or writing any code
  - Run models immediately and incrementally
  - Remove defects early

xtUML Model Compilers



- Design patterns
  - Models of target
  - Templates
- Translation rules
  - Query
  - Transform (M2M)
  - Populate templates
- Marking
  - Like compiler flags
  - Determine which patterns are applied to each model element

- Build
  - Customize tool chain and build parameters
  - Generate build script for CM and batch
  - Existing model compilers for C, C++, SystemC

Details

- OSS Licensing:
  - Eclipse (EPL) + Apache 2.0
- Contribution Process:
  - Join the xtUML Community chat on Skype
  - **Run the Developer Getting Started (xtuml.org/participate/developers)**
  - Select an issue (or open one); assign it to yourself
  - Do the work, including test and documentation
  - Submit a pull request, following Eclipse governance
- Quality Assurance:
  - Written development process
  - Reviews of intermediate work-products (e.g., design notes)
  - Run unit test suite
  - Final review and merge by community committer
- Future Development:
  - Roadmap in issue tracker (and here: xtuml.org/xtumldayemd/)
  - Driven by customer contracts and community contributions
- Contributions of particular interest:
  - Papyrus Platform Migration
  - Marking editor

https://xtuml.org

xtUML

eXecutable Translatable UML with BridgePoint

Log In Register

Search

Participate | Learn | Download | News & Events | Forums | Community Chat! | Marketplace | About

- Users
- Developers
- Projects

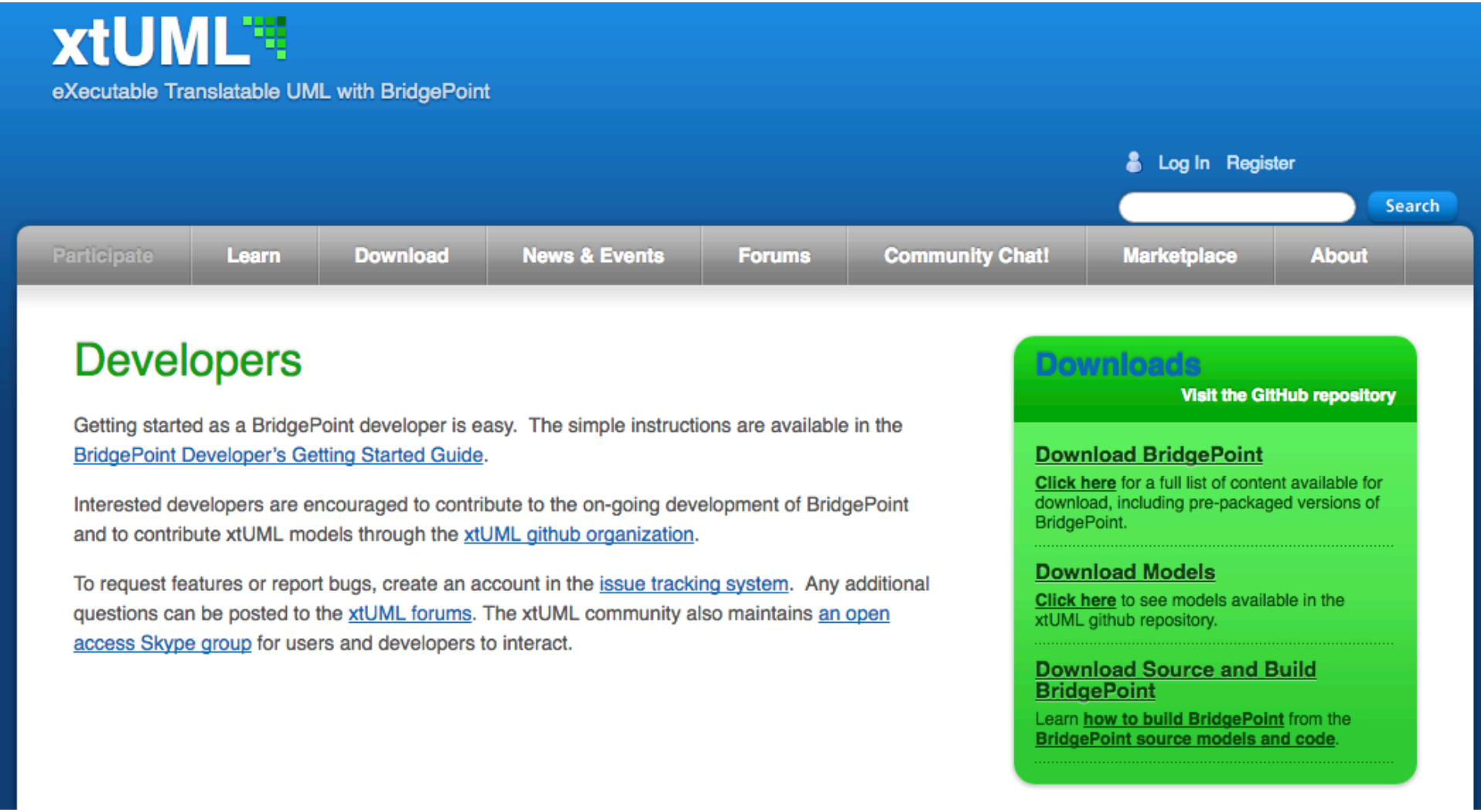
xtUML.org provides the system design community with access to comprehensive UML modeling, execution and translation capabilities through **BridgePoint**, and a forum to advance the methodology of executable and translatable model-driven development.

Learn more »

Executable, translatable UML (xtUML) is an extension to UML based upon the Shlaer-Mellor Method of Model-Driven Architecture (MDA), which supports a powerful approach to Model-Driven Development (MDD). xtUML.org provides the system design community with access to xtUML editing, execution and translation capabilities, along with a forum to advance the use of this methodology. Find the **BridgePoint** tool along with models, training materials and community discussion here.

Downloads

Visit the GitHub repository


xtUML
eXecutable Translatable UML with BridgePoint
Log In
Register
Search
Participate
Learn
Download
News & Events
Forums
Community Chat!
Marketplace
About
Developers
Getting started as a BridgePoint developer is easy. The simple instructions are available in the BridgePoint Developer's Getting Started Guide.
Interested developers are encouraged to contribute to the on-going development of BridgePoint and to contribute xtUML models through the xtUML github organization.
To request features or report bugs, create an account in the issue tracking system. Any additional questions can be posted to the xtUML forums. The xtUML community also maintains an open access Skype group for users and developers to interact.
Downloads
Visit the GitHub repository
Download BridgePoint
Click here for a full list of content available for download, including pre-packaged versions of BridgePoint.
Download Models
Click here to see models available in the xtUML github repository.
Download Source and Build BridgePoint
Learn how to build BridgePoint from the BridgePoint source models and code.

keithbrown job #8637 - removing the committed version of pt_antlr's antlr.jar. 654804f on Aug 22

4 contributors 

100 lines (75 sloc) | 6.48 KB | Raw | Blame | History

Developer Getting Started Guide

This document provides the information required to create a fully functional BridgePoint development environment.

Preparation

1) This is a link to git documentation that describes working with forks. BridgePoint development requires developers to have a working knowledge of git and git forks. Throughout this document we will refer to the repositories using the formula: `https://github.com/"username"/"repository".git` where "username" is your personal Github user name. (For example: `https://github.com/keithbrown/bridgepoint.git`)

2) This document may be used in Linux or Windows, but its examples use Linux. In Windows, cygwin is used to ease setup. Throughout this document we use "~" in the example path names. If you are building on Windows replace "~"