



Model Driven Engineering

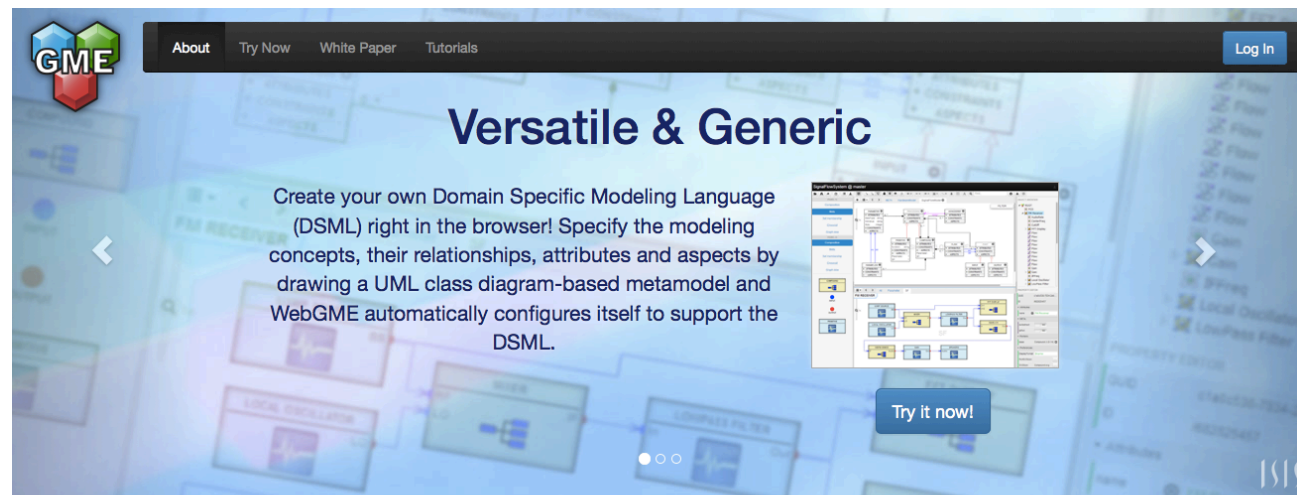


Modeling RPG game using **WebGME**

Raha Naseri

Introduction to WebGME

- From the developers of GME
- A tool to create DSML for complicated systems
- Hosted on Amazon cloud at <http://webGME.org>

The banner features the GME logo (three colored diamonds: blue, green, red) on the left. A dark navigation bar contains links for 'About', 'Try Now', 'White Paper', and 'Tutorials', with a 'Log In' button on the right. The main text reads 'Versatile & Generic' in a large, bold font. Below this, a paragraph describes creating a DSML in a browser. To the right is a screenshot of the WebGME interface showing a UML class diagram. A 'Try it now!' button is positioned at the bottom right of the banner area.

GME

About Try Now White Paper Tutorials Log In

Versatile & Generic

Create your own Domain Specific Modeling Language (DSML) right in the browser! Specify the modeling concepts, their relationships, attributes and aspects by drawing a UML class diagram-based metamodel and WebGME automatically configures itself to support the DSML.

Try it now!

Collaborate

Collaborate with your colleagues by simultaneously working on the same project. WebGME works just like Google Docs; it updates and shows all changes to each user concurrently. And you never lose work because the models are stored in a database in the cloud.

Crosscuts

It is always the cross cutting concerns that are hardest to model and manage. WebGME introduces Crosscuts, a novel way to visualize and modify associations among model elements that are potentially far away in the composition and/or inheritance hierarchy.

APIs

Both a native Javascript and a RESTful API are supported. The former makes it possible to write high-performance code generators, while the latter enables interfacing with WebGME in practically any programming language.

WebGME features

- Scalability and extensibility
- Platform independent
- Version control
- Multi paradigm with crosscuts

Version control

- go back to previous states of the project

Project repository history

| Graph | Actions | Commit | Message | User |
|-------|---------|---|---|------|
| | | #c1acc5bb4d85e7d5275b75f0f00e436d792b11ef master ✖ | startTransaction() addMember(/1619890906,MetaAspectSet_6f9d8a84-41e4-3bd8-d83b-a65ca76858a5) setMemberRegistry(/1619890906,Meta... | demo |
| | | #d0cf38beecde3eea4c64a79cccd3f538700b72dd | startTransaction() addMember(/1950669966,MetaAspectSet_6f9d8a84-41e4-3bd8-d83b-a65ca76858a5) setMemberRegistry(/1950669966,MetaAspectSet... | demo |
| | | #67e452f4ba91accbab4d4779dddfb8f42c048faa | startTransaction() addMember(/1305724177,MetaAspectSet_6f9d8a84-41e4-3bd8-d83b-a65ca76858a5) setMemberRegistry(/1305724177,MetaAspectSet... | demo |
| | | #8d9cb75014f03e944cb4b122e75bd989ec5ac101 | createChildren("/1*:/1950669966") | demo |
| | | #d9618f12aa8c1d15505e17c8d1d1eacaf4fc64a6 | createChildren("/1147273918*:/1518317810") | demo |
| | | #a802595cfb207f12bb7f65d1c73f054305bd9c13 | createChildren("/371045888*:/1305724177") | demo |
| | | #e7c2531af8c7a9d6f9d5213e55db75fb25c6dcd1 | createChildren("/855890922*:/1619890906") | demo |
| | | #02597e68de66372a1b73e9b2e01823fe2efaefcc | startTransaction() addMember(/371045888,MetaAspectSet_6f9d8a84-41e4-3bd8-d83b-a65ca76858a5) setMemberRegistry(/371045888,MetaAspectSet_6f... | demo |
| | | #f6d56af7b12f9bc3d4ed6500c99c47e9b7674417 | createChildren("/1*:/371045888") | demo |
| | | #aa2272ba0def7437845ca47edbac42f1e6b64d4f | startTransaction() createChild(/1./855890922) setRegistry(/855890922,...position,{object Object}) setAttribute(/855890922,name,FCO_instance) addMember(/... | demo |
| | | #700b02ab63732ead414266dd072bda071a9e73fe | startTransaction() addMember(/1147273918,MetaAspectSet_6f9d8a84-41e4-3bd8-d83b-a65ca76858a5) setMemberRegistry(/1147273918,MetaAspectSet... | demo |
| | | #680d64893d7f19d4fedff8ae96de360206b101ad | createChildren("/1*:/1147273918") | demo |
| | | #95b4aeddd4115721ad2b4e4f62f54aea30794404 | startTransaction() setMemberRegistry(/1,MetaAspectSet_6f9d8a84-41e4-3bd8-d83b-a65ca76858a5,position,{object Object}) completeTransaction() | demo |
| | | #bca0a55697ab313cf3a9ac47afc5d5a38a8f07e1 | startTransaction() createSet(/,MetaAspectSet_4309ffa9-d0b9-ac61-71f3-c22364085529) setRegistry(/,MetaSheets,{object Object},{object Object}) completeTra... | demo |
| | | #5e81474b5a0b4f0387d516509a7e02539603a7fc | startTransaction() setRegistry(/,validPlugins,) createChild(/undefined,/1) setMeta(/) setMeta(/1) setAttribute(/1,name,FCO) setAttribute(/name,ROOT) setReastr... | demo |
| | | #60c227279c276e92fdc2e66c391758cc9b7506ed | project creation commit | demo |

Close



WebGME User Interface

- Main toolbar
- Mode bar
- Object browser
- Part browser
- Property editor

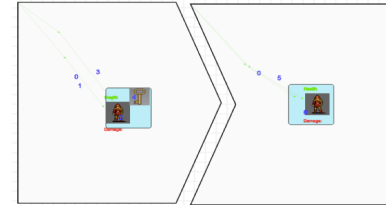
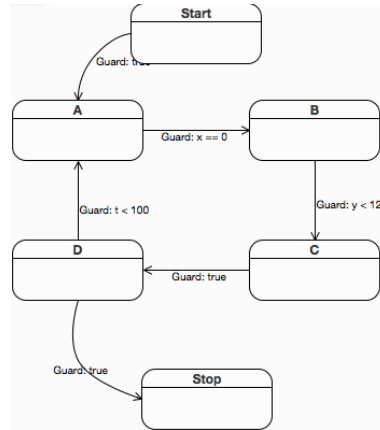
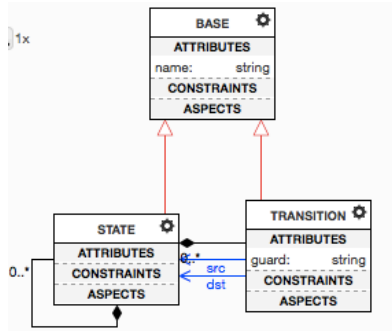
The screenshot displays the WebGME user interface for editing a state machine model. The browser address bar shows the URL: `editor.webgme.org/?user=demo&create=true&project=demo_87_66_141_44`. The page title is "HFSM @ master [READ ONLY]".

The interface is divided into several panels:

- Left Panel (PANEL 1):** Contains a "Composition" section with options for "Meta", "Set membership", "Crosscut", and "Graph view". Below it is "PANEL 2" with options for "Composition", "Meta", "Set membership", "Crosscut", and "Graph view". At the bottom of this panel are visual representations of the "BASE" and "State" objects.
- Center Panel:** Displays a UML-like state machine diagram. It features a "BASE" class with "ATTRIBUTES" (name: string), "CONSTRAINTS", and "ASPECTS". Below it are "STATE" and "TRANSITION" classes, each with "ATTRIBUTES", "CONSTRAINTS", and "ASPECTS". A transition arrow connects the "STATE" to the "TRANSITION" with labels "src" and "dst".
- Right Panel (OBJECT BROWSER):** Shows a tree view of the model structure. The root node is "ROOT", which contains a "BASE" object, a "State" object, and a "Sample" object. A "GUARD: TRUE" object is also visible, associated with a "Connection".
- Bottom Right Panel (PROPERTY EDITOR):** Displays the properties of the selected "ROOT" object. The GUID is `f6e3174c-c191-9fea-...`. The "name" property is set to "ROOT". Other expandable sections include "META", "isAbstract", "isPort", "validPlugins", "Pointers", "base", "Preferences", and "DisplayFormat".

At the bottom of the interface, there is a status bar with the text "© 2014 Vanderbilt University version: 0.6.5" and several status indicators: "master", "IN SYNC", "CONNECTED", "LOG: WARNING", and "ON".

RPG game modelling procedure



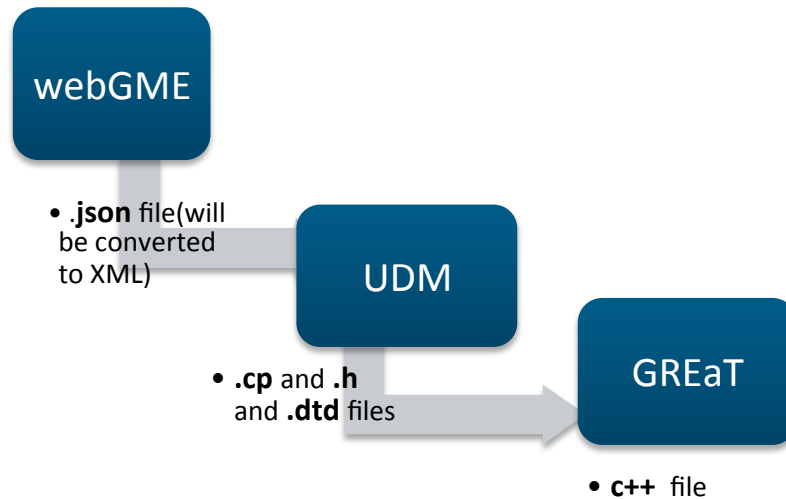
Metamodel
UML

Model
Concrete syntax

Rule-based
Transformation

Future plan

- WebGME : UML and DSML
- GREaT tool : rule_based transformation



Questions?