Gmodel

Product Roadmap 2010 Q4

internal document, not for public release

1 November 2010

Gmodel

Gmodel

Introduction

The Gmodel product roadmap covers the expectations that are associated with the Gmodel Open Source Edition, the Gmodel Semantic Extensions Edition, and related products, such as the Requirements Formalizer and the Visual Syntax Designer. This document outlines expectations related to functionality, to scalability of Gmodel-based solutions, and to the boundary between the Open Source Edition and the Semantic Extensions Edition and further non-Open Source products.

Product Family Management

The Gmodel product roadmap is updated on a monthly basis to reflect the latest insights into the needs and priorities that are articulated by Gmodel users and by potential users of Gmodel. On a quarterly basis the level of detail of the envisaged **use cases** that are scheduled for inclusion in the next **quarterly release** is elaborated to the extent of concrete user-system-interaction steps.

On a monthly basis the next five weeks of the product roadmap are translated into an iteration plan consisting of **use case scenarios** that are scheduled for realisation as part of the next **monthly release**. Use case scenarios are assigned to **solution developers** either as a whole, or may be broken down further into **tasks**, which by definition must be assignable to an individual solution developer.

On a weekly basis the **work in progress** on the monthly release is published in a **weekly release** that is intended for informal user acceptance testing by **modellers** who have a concrete need for the functionality uner development.

Product Familiy Scope

Gmodel is the solution for **Enterprise Modelling** and **Data Management**

With the universal availability of computing power and the Internet as a ubiquitous information source, the sheer volume of data is overwhelming. Making sense of the flood of raw data, and translating it into relevant information requires powerful mechanisms for filtering, structuring, modelling, and aggregation. The number of tools to support the attempt of transforming data into valuable know-how is endless. Tools vary in their degree of formalisation, in basic concepts, and in representation formats. When immersed in the world of data and representations, it quickly becomes apparent that individual tools can be very useful, but that often there either is no – or only insufficient support for exchanging semantic information between tools. This is where Gmodel enterprise modelling and data management technology comes into the picture, as a semantic interoperability platform for any number of tools in any number of domains.

Gmodel

Mathematical Foundation

Gmodel was designed from the ground up without any consideration for popular software modelling paradigms or modelling languages. The Gmodel kernel provides a universally and recursively applicable module concept that completely shields the user from the technical details of traditional databases, file systems, and programming languages (= implementation technologies). The open-source kernel of Gmodel can be extended in a shell-like manner with an arbitrary set of user-defined domain-specific modelling languages. The implementation of the kernel is based on a small number of concepts that have their origin in model theory and denotational semantics, enabling the exploitation of proven mathematical theorems and transformations. Furthermore, the design of Gmodel guarantees that a large number of best practices for the construction of modular and scalable models are automatically enforced, not only by the kernel, but also by all user-defined extensions of the kernel:

- The domain to be modelled is partitioned into well-defined areas of knowledge, each of which relates to exactly one role.
- All work products (= artefacts) are based on information produced by a specific role as a result of a specific event.
- Each category of work products is expressed in a dedicated modelling language.
- Conversely, every modelling language defines exactly one category of work products.
- Variants of modelling language definitions and model instances are expressed as extensions of a common category.
- Work products are the granularity at which versioning and locking is applied.
- Language definitions (category models) are treated as work products as well.
- Circular or bi-directional dependencies between work products are not possible.
- Work products always contain a reference to the corresponding language definition (category model).
- Powerful work product modularisation and search mechanisms guarantee maintainability, even in geographically distributed organisations with millions of work products.

Advantages of Solutions Based on Gmodel

- Gmodel is the foundation of an enterprise-wide modelling philosophy with unique features:
- No limit to the levels of abstractions.

Gmodel

- Any model instance can be used in the role of a language definition (multi-level-instantiation).

- No limits to the number of user-defined notational styles and jargons.

- Interoperability with any database or modelling standard (as long as a formal specification of the standard is available).

- Referential integrity across all modelling languages and work products.

- Precise definition of the commonalities and variabilities in product lines.

- Unlimited scalability of work products.

- Explicit scope management across all work products that are managed by the Gmodel repository.

- Universal enterprise content-management functionality that also extends to traditional file-based work products.

- Highly portable kernel based on a minimal Java implementation.

- Implementation/generation of derived work products (for example program code) on any technology stack.

- Adaptable to any business domain and any IT organisation.

- All work products are managed in a repository implemented with relational database technology (or alternatively, cloud-database technology).

- Web-based visualisation with multiple interface styles.

- Out-of-the-box integration with ECLIPSE.

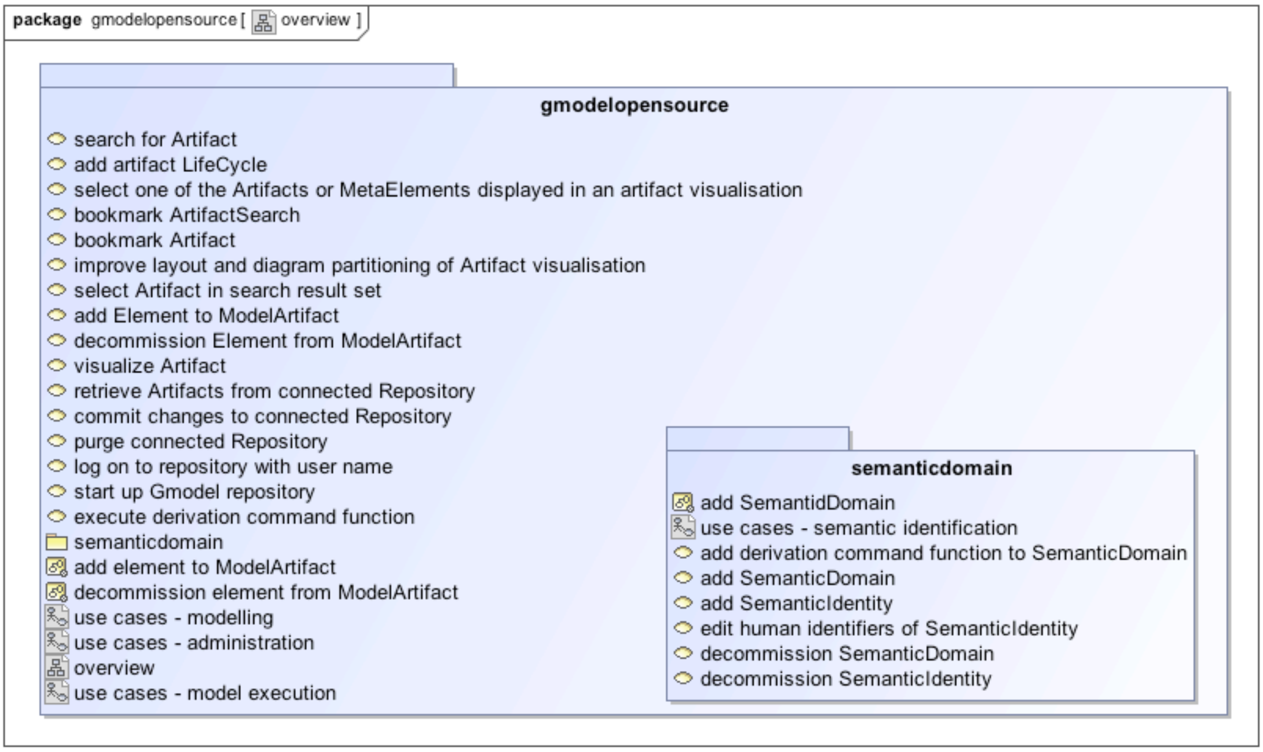
- Open-source kernel with flexible licences for extensions.

Since any data technology and any modelling language can be emulated with Gmodel, it is a universal interoperability platform for databases, data-warehouse solutions, and modelling tools

Gmodel

Scope of the Gmodel Open Source Edition

The following diagram illustrates the scope of the open source edition.



package gmodelopensource [overview]

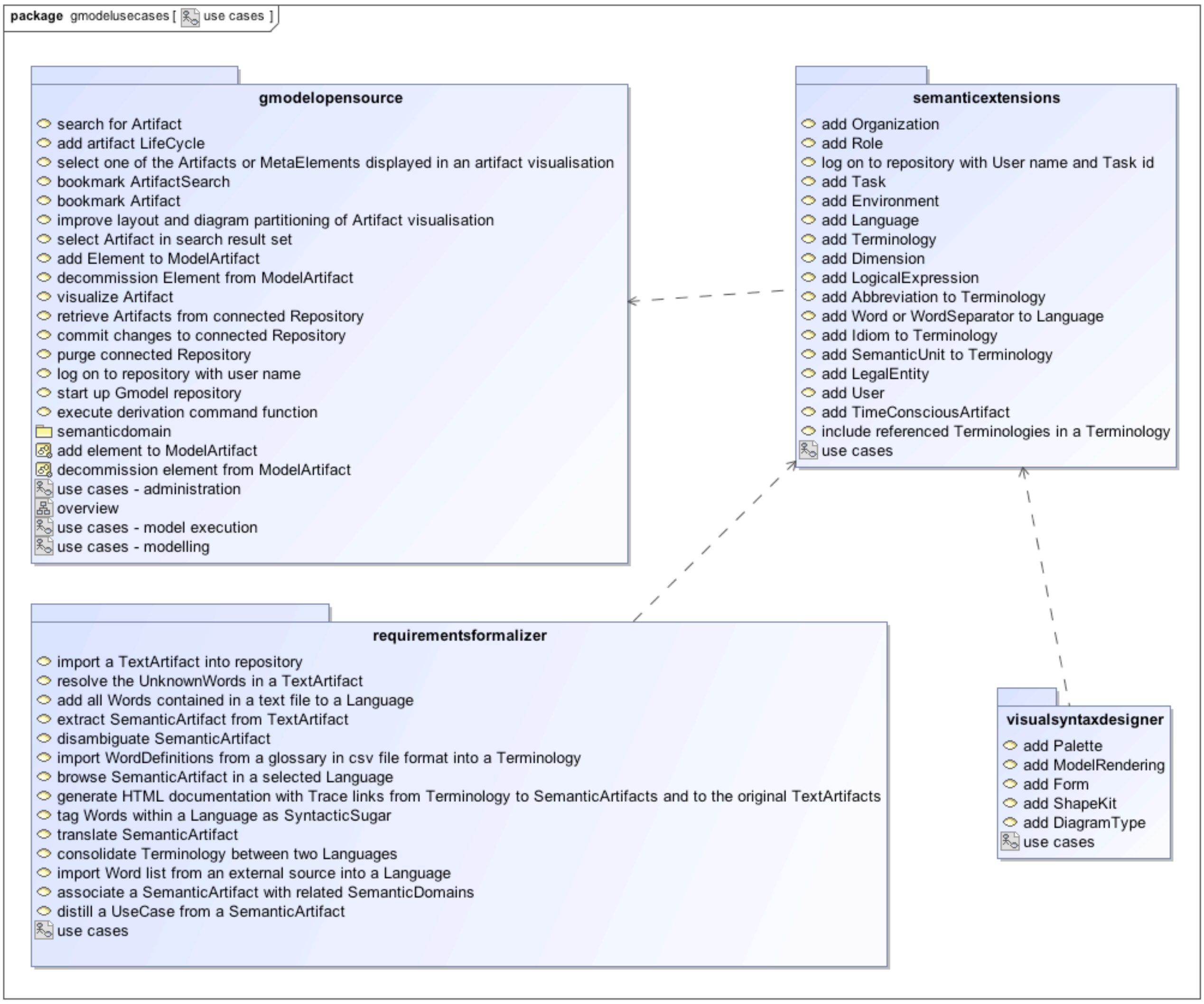
gmodelopensource

- search for Artifact
- add artifact LifeCycle
- select one of the Artifacts or MetaElements displayed in an artifact visualisation
- bookmark ArtifactSearch
- bookmark Artifact
- improve layout and diagram partitioning of Artifact visualisation
- select Artifact in search result set
- add Element to ModelArtifact
- decommission Element from ModelArtifact
- visualize Artifact
- retrieve Artifacts from connected Repository
- commit changes to connected Repository
- purge connected Repository
- log on to repository with user name
- start up Gmodel repository
- execute derivation command function
- semanticdomain
- add element to ModelArtifact
- decommission element from ModelArtifact
- use cases - modelling
- use cases - administration
- overview
- use cases - model execution

semanticdomain

- add SemantidDomain
- use cases - semantic identification
- add derivation command function to SemanticDomain
- add SemanticDomain
- add SemanticIdentity
- edit human identifiers of SemanticIdentity
- decommission SemanticDomain
- decommission SemanticIdentity

Wider Gmodel Product Family Context



package gmodelusecases [use cases]

gmodelopensource

- search for Artifact
- add artifact LifeCycle
- select one of the Artifacts or MetaElements displayed in an artifact visualisation
- bookmark ArtifactSearch
- bookmark Artifact
- improve layout and diagram partitioning of Artifact visualisation
- select Artifact in search result set
- add Element to ModelArtifact
- decommission Element from ModelArtifact
- visualize Artifact
- retrieve Artifacts from connected Repository
- commit changes to connected Repository
- purge connected Repository
- log on to repository with user name
- start up Gmodel repository
- execute derivation command function
- semanticdomain
- add element to ModelArtifact
- decommission element from ModelArtifact
- use cases - administration
- overview
- use cases - model execution
- use cases - modelling

semanticextensions

- add Organization
- add Role
- log on to repository with User name and Task id
- add Task
- add Environment
- add Language
- add Terminology
- add Dimension
- add LogicalExpression
- add Abbreviation to Terminology
- add Word or WordSeparator to Language
- add Idiom to Terminology
- add SemanticUnit to Terminology
- add LegalEntity
- add User
- add TimeConsciousArtifact
- include referenced Terminologies in a Terminology
- use cases

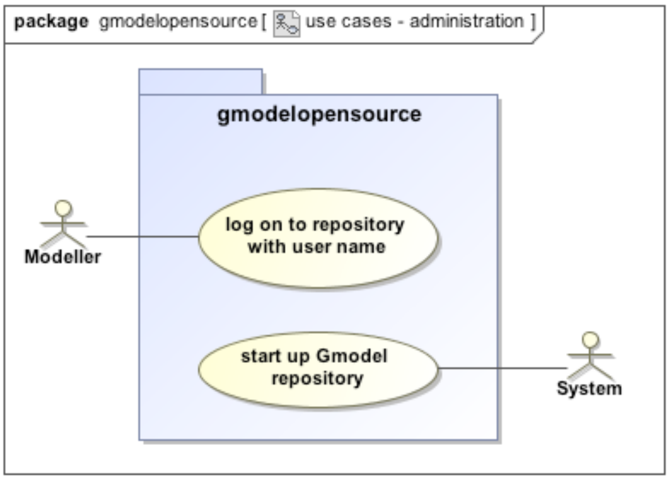
requirementsformalizer

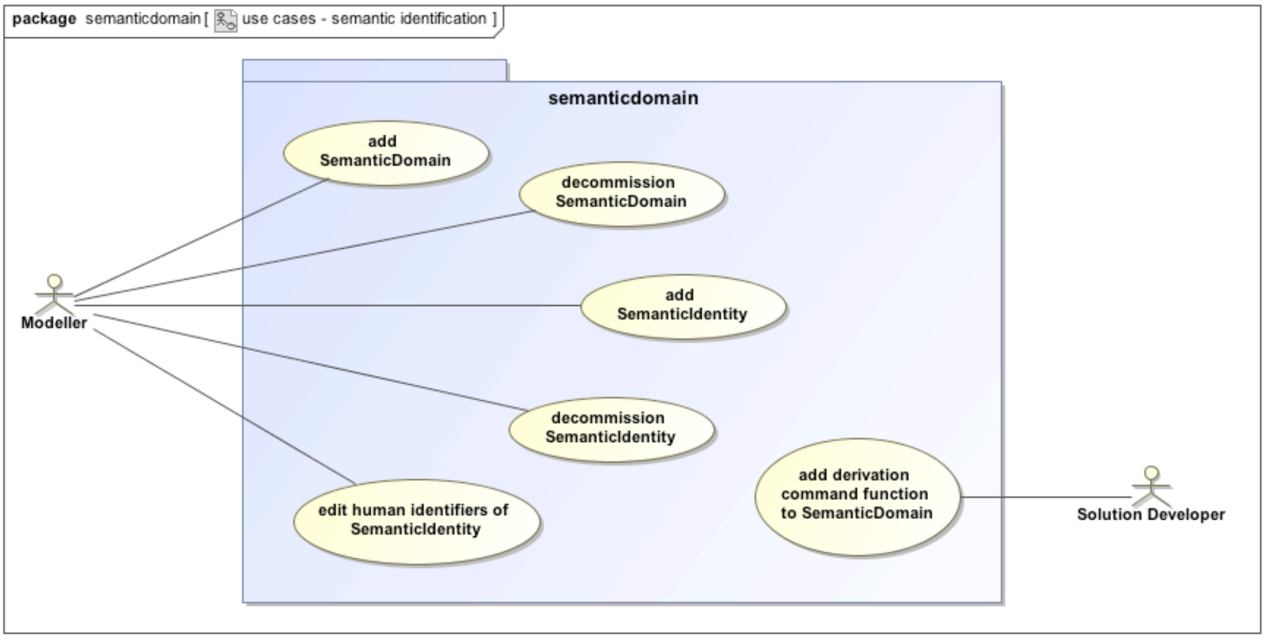
- import a TextArtifact into repository
- resolve the UnknownWords in a TextArtifact
- add all Words contained in a text file to a Language
- extract SemanticArtifact from TextArtifact
- disambiguate SemanticArtifact
- import WordDefinitions from a glossary in csv file format into a Terminology
- browse SemanticArtifact in a selected Language
- generate HTML documentation with Trace links from Terminology to SemanticArtifacts and to the original TextArtifacts
- tag Words within a Language as SyntacticSugar
- translate SemanticArtifact
- consolidate Terminology between two Languages
- import Word list from an external source into a Language
- associate a SemanticArtifact with related SemanticDomains
- distill a UseCase from a SemanticArtifact
- use cases

visualsyntaxdesigner

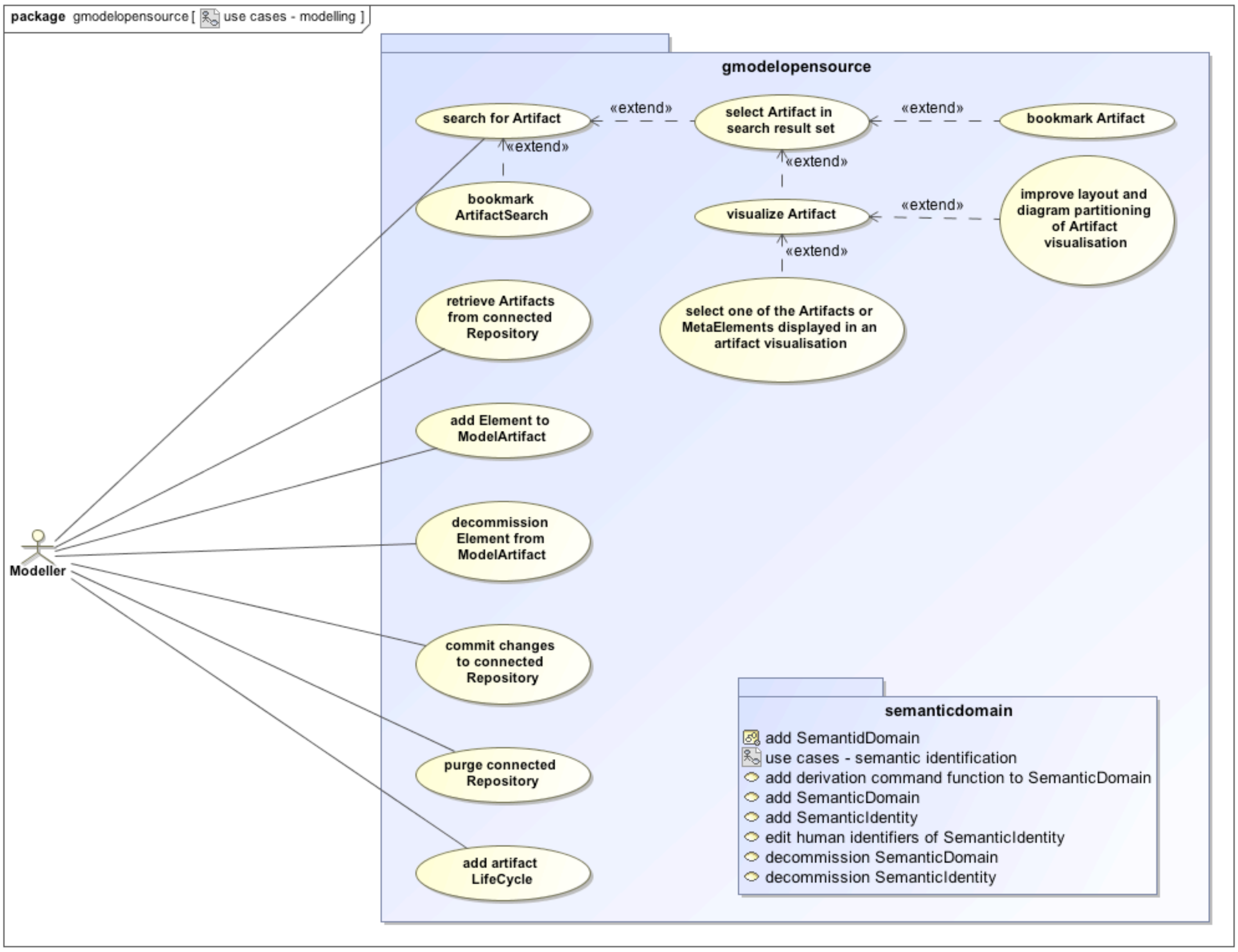
- add Palette
- add ModelRendering
- add Form
- add ShapeKit
- add DiagramType
- use cases

Use Case Model - Open Source Edition

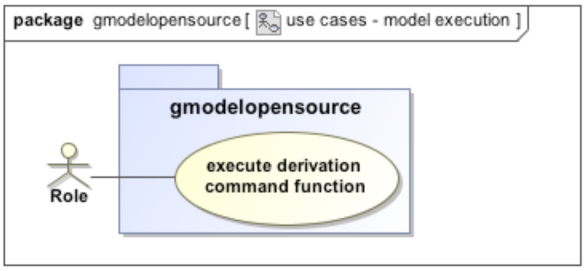

package gmodelopensource [use cases - administration]
gmodelopensource
Modeller
log on to repository with user name
start up Gmodel repository
System


package semanticdomain [use cases - semantic identification]
semanticdomain
add SemanticDomain
decommission SemanticDomain
Modeller
add SemanticIdentity
decommission SemanticIdentity
edit human identifiers of SemanticIdentity
add derivation command function to SemanticDomain
Solution Developer

Gmodel

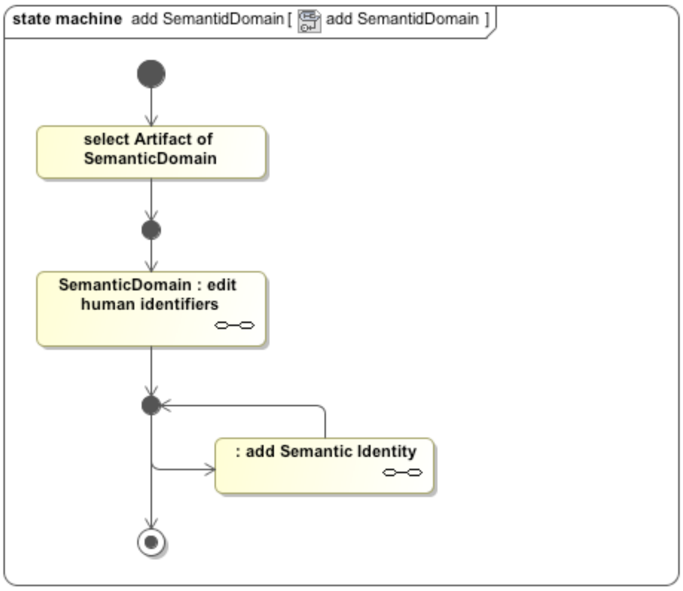


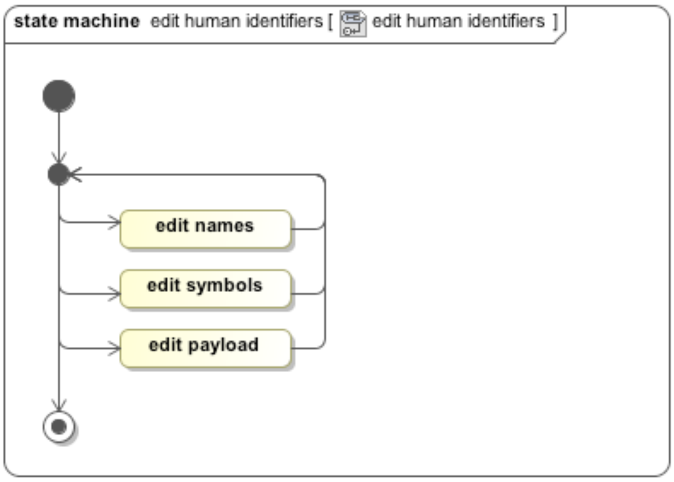
package gmodelopensource [use cases - modelling]
gmodelopensource
search for Artifact
«extend»
select Artifact in search result set
«extend»
bookmark Artifact
«extend»
bookmark ArtifactSearch
«extend»
visualize Artifact
«extend»
improve layout and diagram partitioning of Artifact visualisation
«extend»
select one of the Artifacts or MetaElements displayed in an artifact visualisation
retrieve Artifacts from connected Repository
add Element to ModelArtifact
decommission Element from ModelArtifact
commit changes to connected Repository
purge connected Repository
add artifact LifeCycle
Modeller
semanticdomain
add SemantidDomain
use cases - semantic identification
add derivation command function to SemanticDomain
add SemanticDomain
add SemanticIdentity
edit human identifiers of SemanticIdentity
decommission SemanticDomain
decommission SemanticIdentity

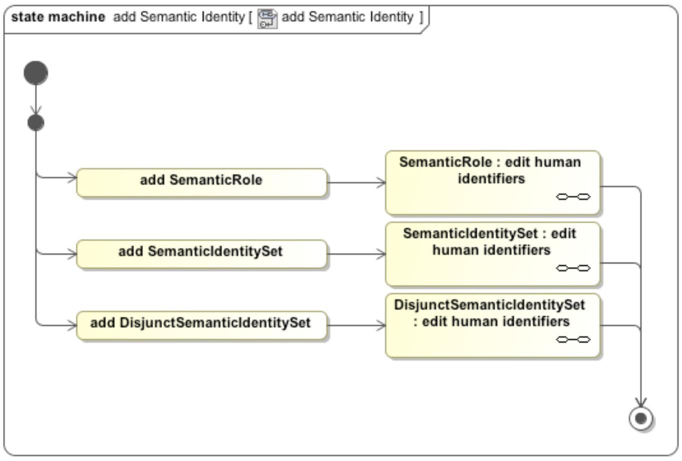


package gmodelopensource [use cases - model execution]
gmodelopensource
execute derivation command function
Role

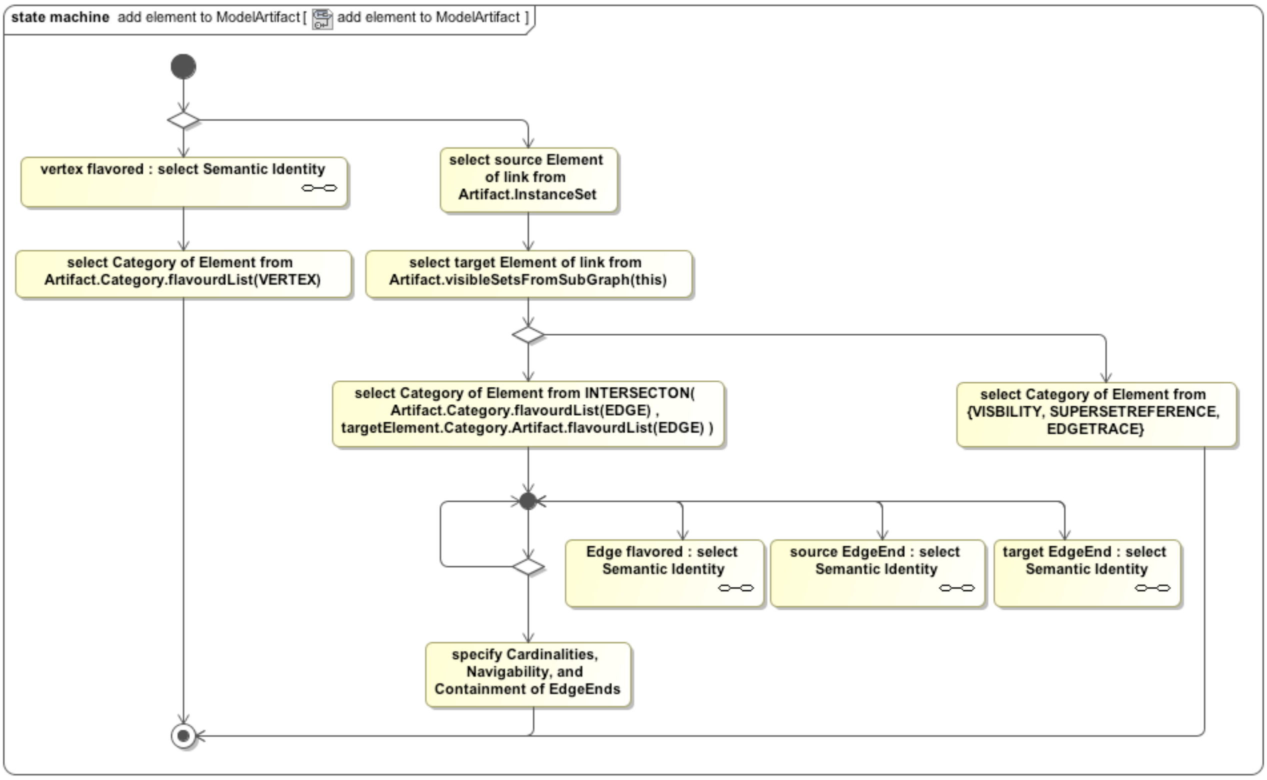
Use Case Descriptions


state machine add SemantidDomain [add SemantidDomain]
select Artifact of SemanticDomain
SemanticDomain : edit human identifiers
: add Semantic Identity

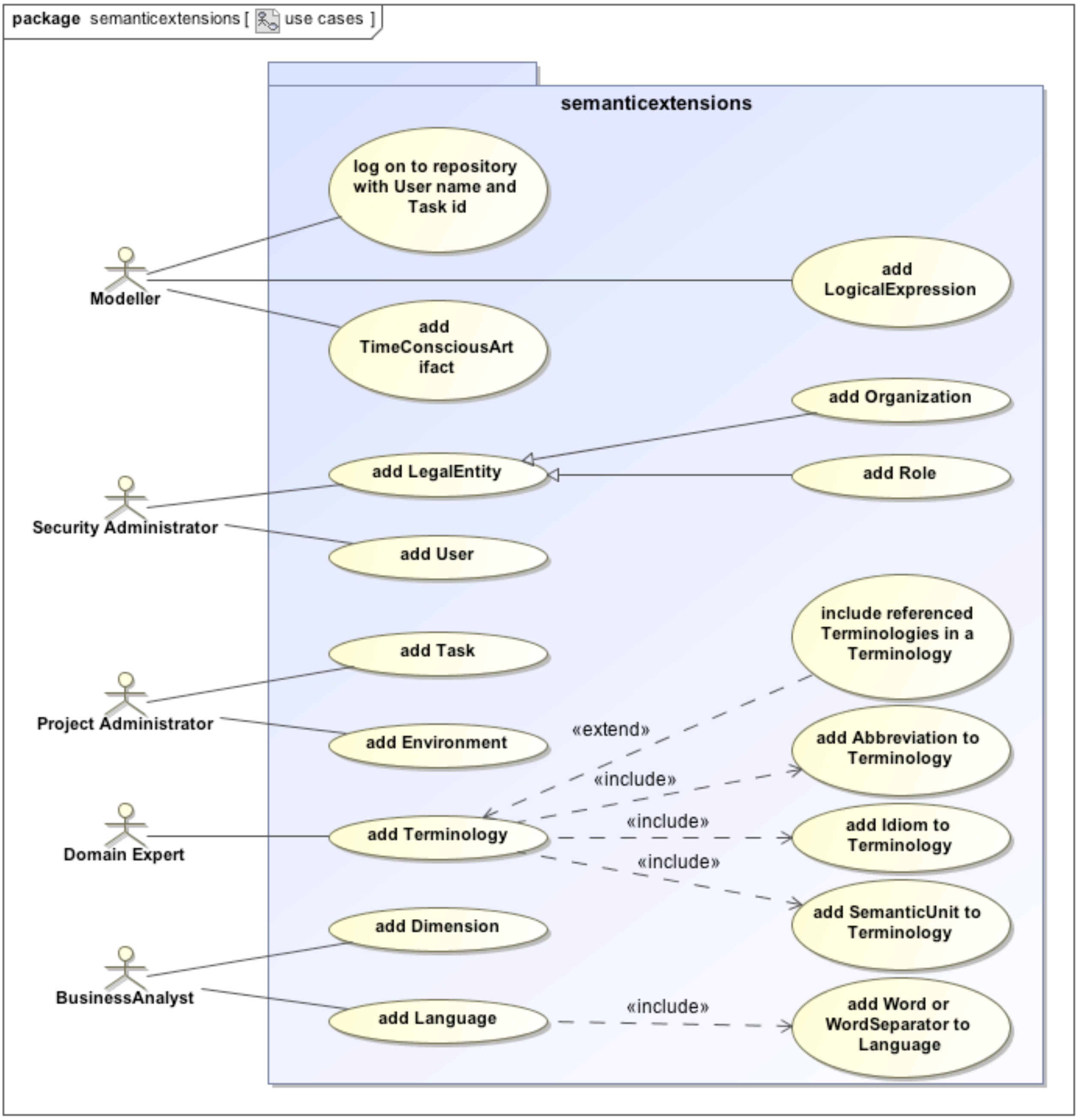

state machine edit human identifiers [edit human identifiers]
edit names
edit symbols
edit payload


state machine add Semantic Identity [add Semantic Identity]
add SemanticRole
add SemanticIdentitySet
add DisjunctSemanticIdentitySet
SemanticRole : edit human identifiers
SemanticIdentitySet : edit human identifiers
DisjunctSemanticIdentitySet : edit human identifiers

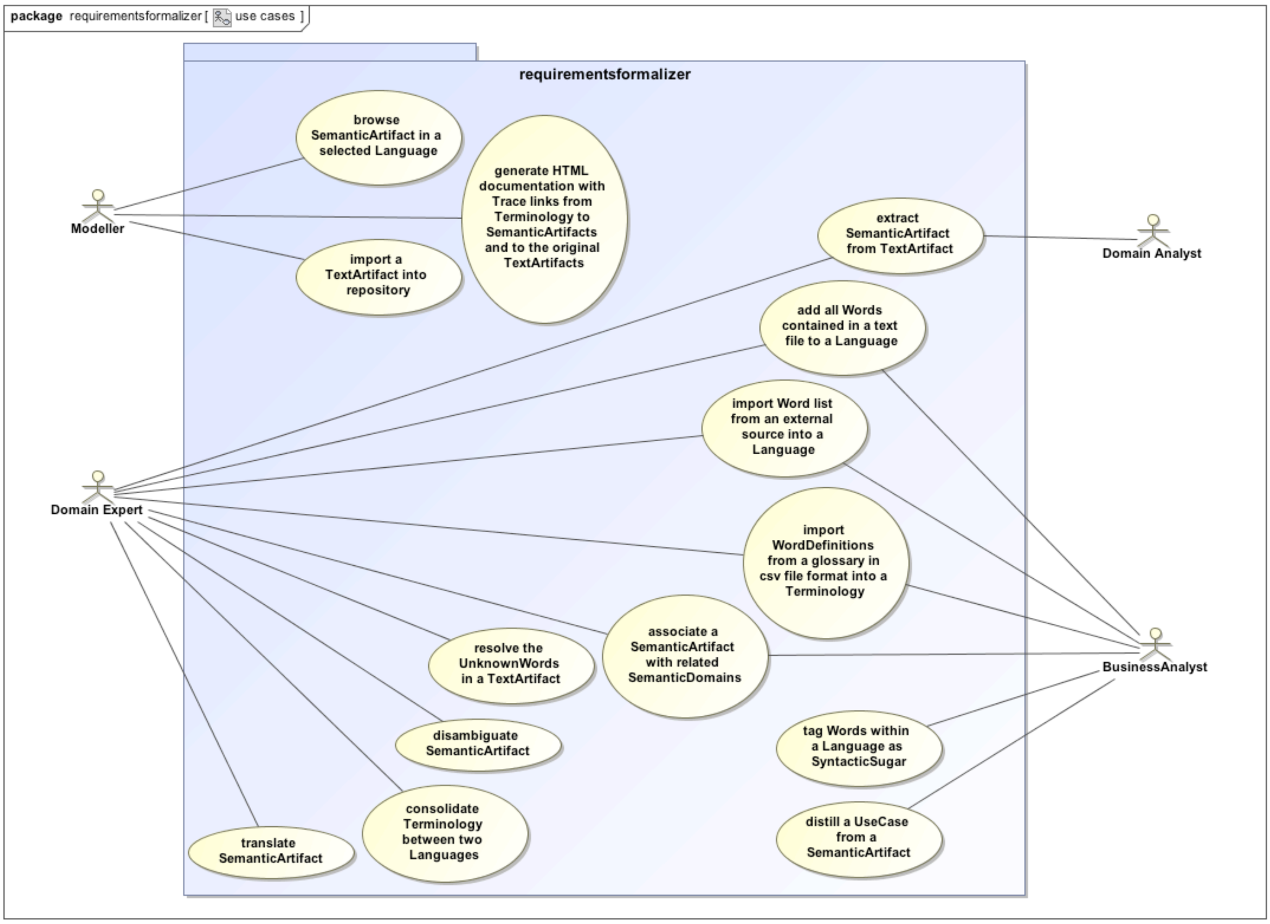
Gmodel


state machine add element to ModelArtifact [add element to ModelArtifact]
vertex flavored : select Semantic Identity
select source Element of link from Artifact.InstanceSet
select Category of Element from Artifact.Category.flavourdList(VERTEX)
select target Element of link from Artifact.visibleSetsFromSubGraph(this)
select Category of Element from INTERSECTON(Artifact.Category.flavourdList(EDGE) , targetElement.Category.Artifact.flavourdList(EDGE))
select Category of Element from {VISBILITY, SUPERSETREFERENCE, EDGETRACE}
Edge flavored : select Semantic Identity
source EdgeEnd : select Semantic Identity
target EdgeEnd : select Semantic Identity
specify Cardinalities, Navigability, and Containment of EdgeEnds

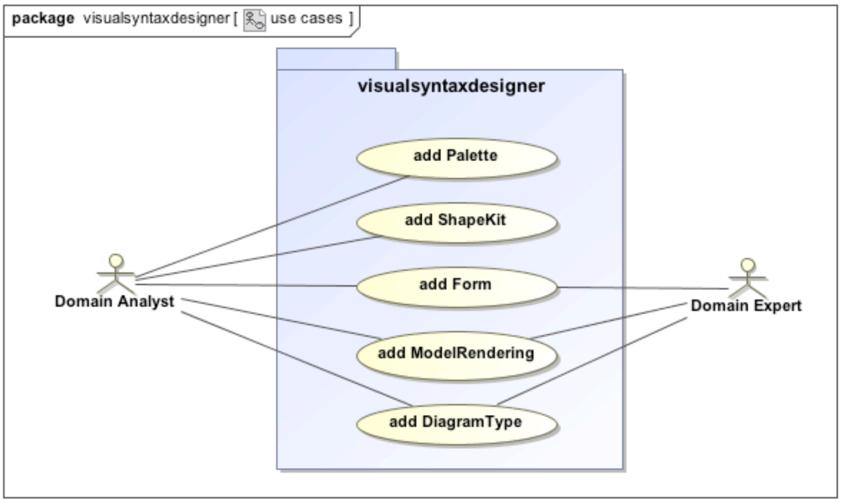
Use Case Model - Semantics Extensions Edition



Use Case Model - Requirements Formalizer


package requirementsformalizer [use cases]
requirementsformalizer
Modeller
Domain Expert
Domain Analyst
BusinessAnalyst
browse SemanticArtifact in a selected Language
generate HTML documentation with Trace links from Terminology to SemanticArtifacts and to the original TextArtifacts
import a TextArtifact into repository
extract SemanticArtifact from TextArtifact
add all Words contained in a text file to a Language
import Word list from an external source into a Language
import WordDefinitions from a glossary in csv file format into a Terminology
associate a SemanticArtifact with related SemanticDomains
resolve the UnknownWords in a TextArtifact
disambiguate SemanticArtifact
consolidate Terminology between two Languages
translate SemanticArtifact
tag Words within a Language as SyntacticSugar
distill a UseCase from a SemanticArtifact

Use Case Model - Visual Syntax Designer



Gmodel

Release Schedule

Version 0.x.2, monthly release, delivery 1/12/2010

Version 0.x.3, monthly release, delivery 1/01/2011

Version 1.0.0, quarterly release, delivery 1/02/2011

Version 1.0.1, monthly release, delivery 1/03/2011

Version 1.1.0, quarterly release, delivery 1/04/2011

...

Gmodel

Release Content

Gmodel Open Source Edition

Release	User	Feature
OS 1	DE	<to be agreed>
SX 1	DE	<to be agreed> • ... • ...

Gmodel Semantic Extensions Edition

A. Multi-language and multi-jargon
[Q4 2010]

B. Time conscious artefacts (versioning, branching etc.)
[Q4 2010]

C. Web based generic model editor (incremental delivery of more and more advanced features)

[Q4 2010]

D. Role based access control
[Q4 2010]

E. Optional hosting of repository content
[Q4 2010]

F. Task management
[Q1 2011]

G. Support for environments (dev, test, int, uat, prod, etc.)
[Q1 2011]

H. Change impact analysis functionality
[Q2 2011]

I. Derivation management console
[Q3 2011]

J. Model refactoring functionality (incremental delivery)
[Q3 2011]

Gmodel

Requirements Formalizer

See separate Requirements Formalizer document

Visual Syntax Designer

to be done