Meta-models and Infrastructure for Smalltalk Omnipresent History

Verónica Uquillas-Gómez
Stéphane Ducasse
Theo D'Hondt

Andy Kellens, VUB

Argentina - Nov 12th 2010

Source Code History Analysis

Source Code History Analysis

- Linear history

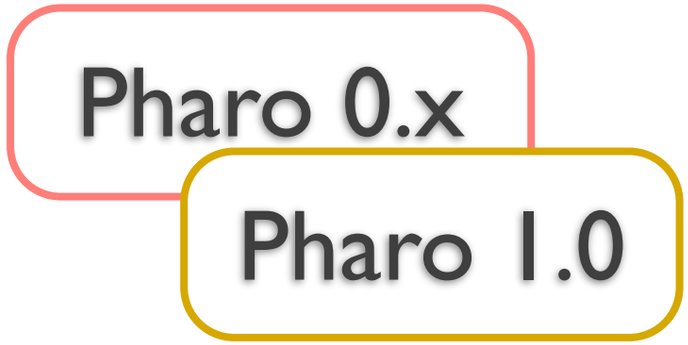
Source Code History Analysis

- Linear history

Pharo 0.x

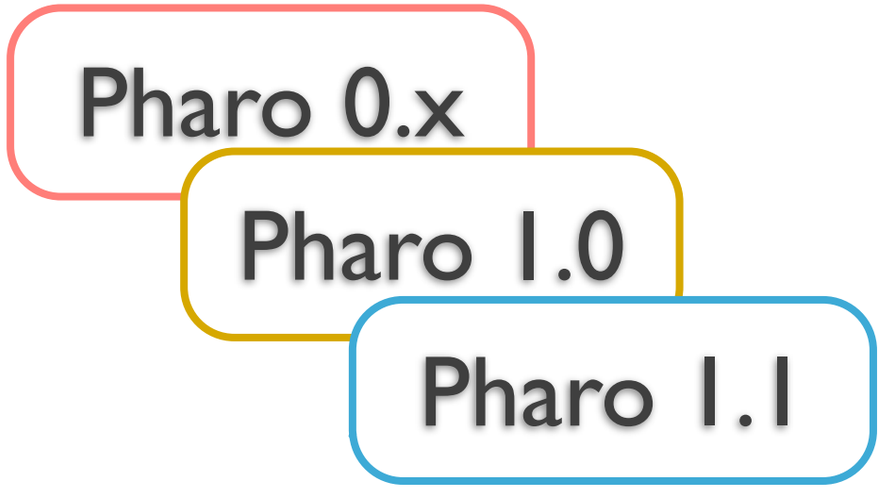
Source Code History Analysis

- Linear history



Source Code History Analysis

- Linear history



Source Code History Analysis

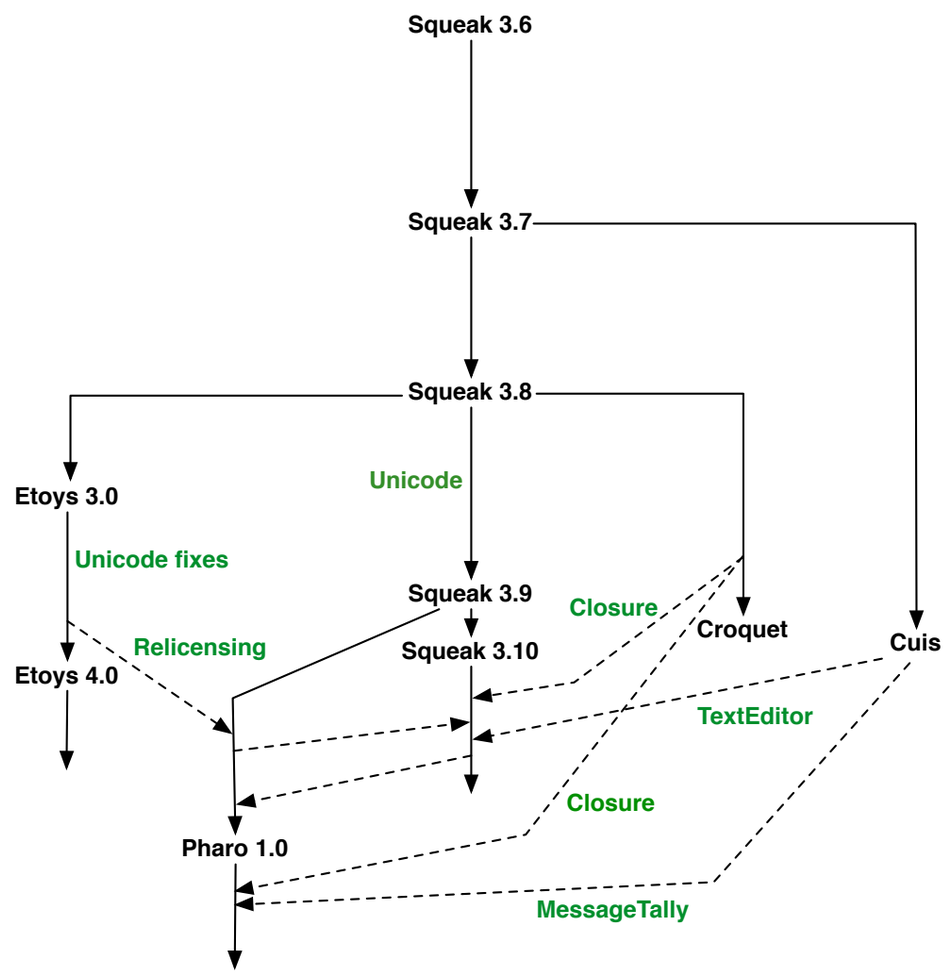
- Linear history

Source Code History Analysis

- Linear history
- Cross history

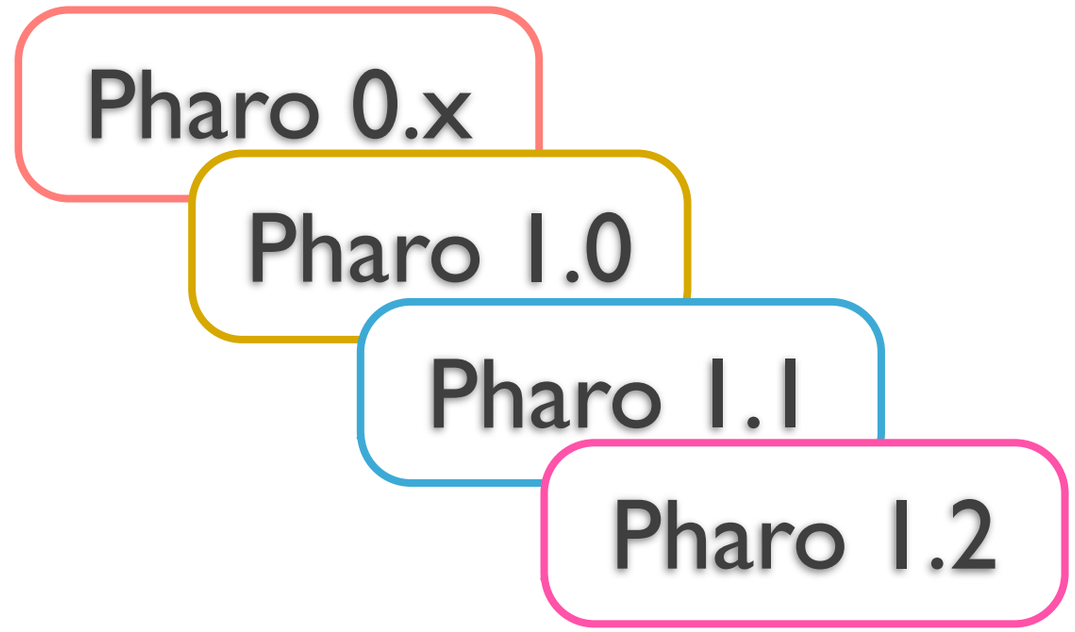
Source Code History Analysis

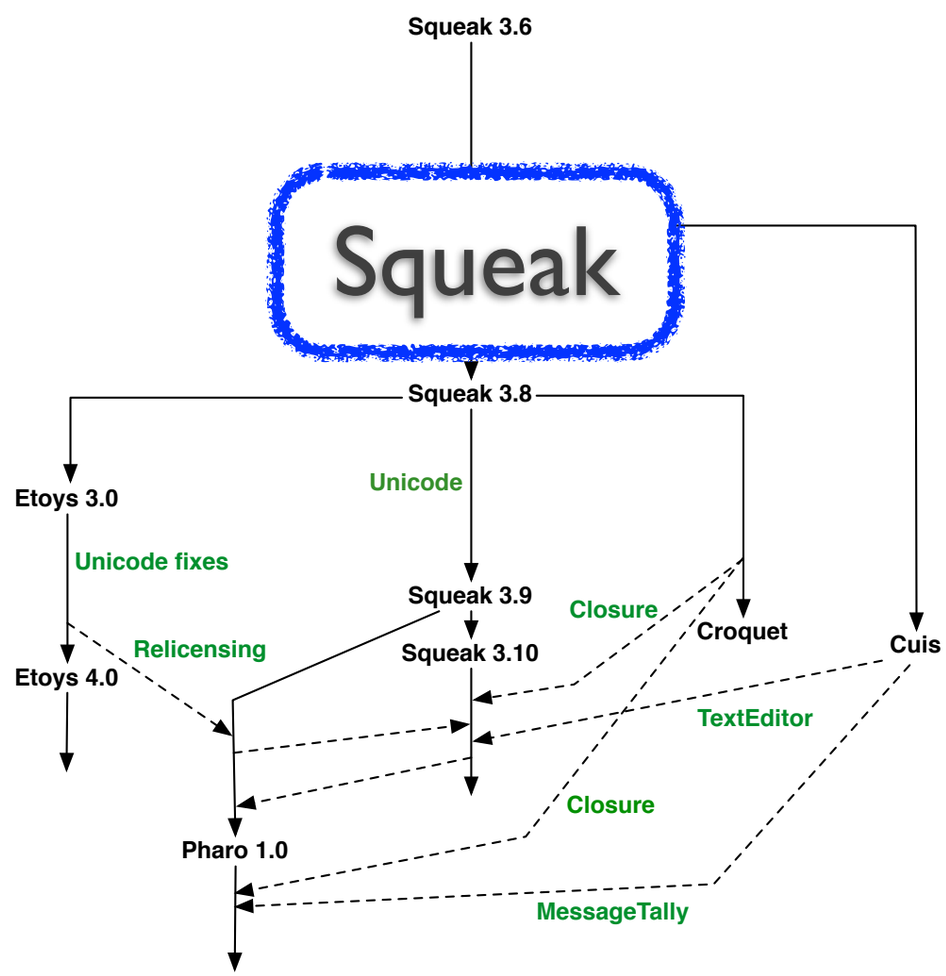
- Linear history
- Cross history



Source Code History Analysis

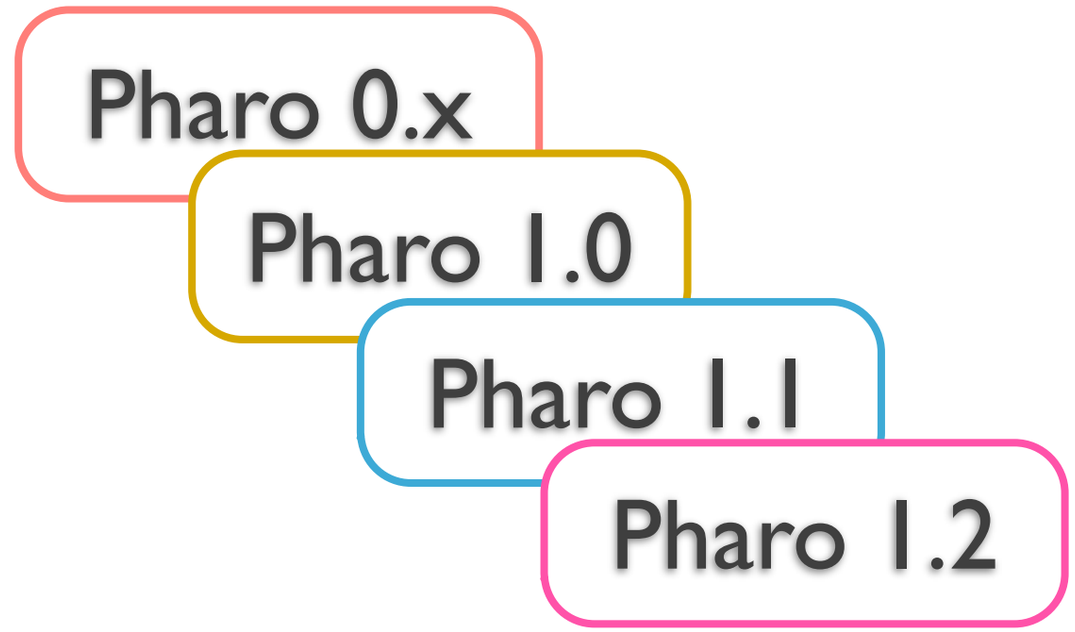
- Linear history
- Cross history

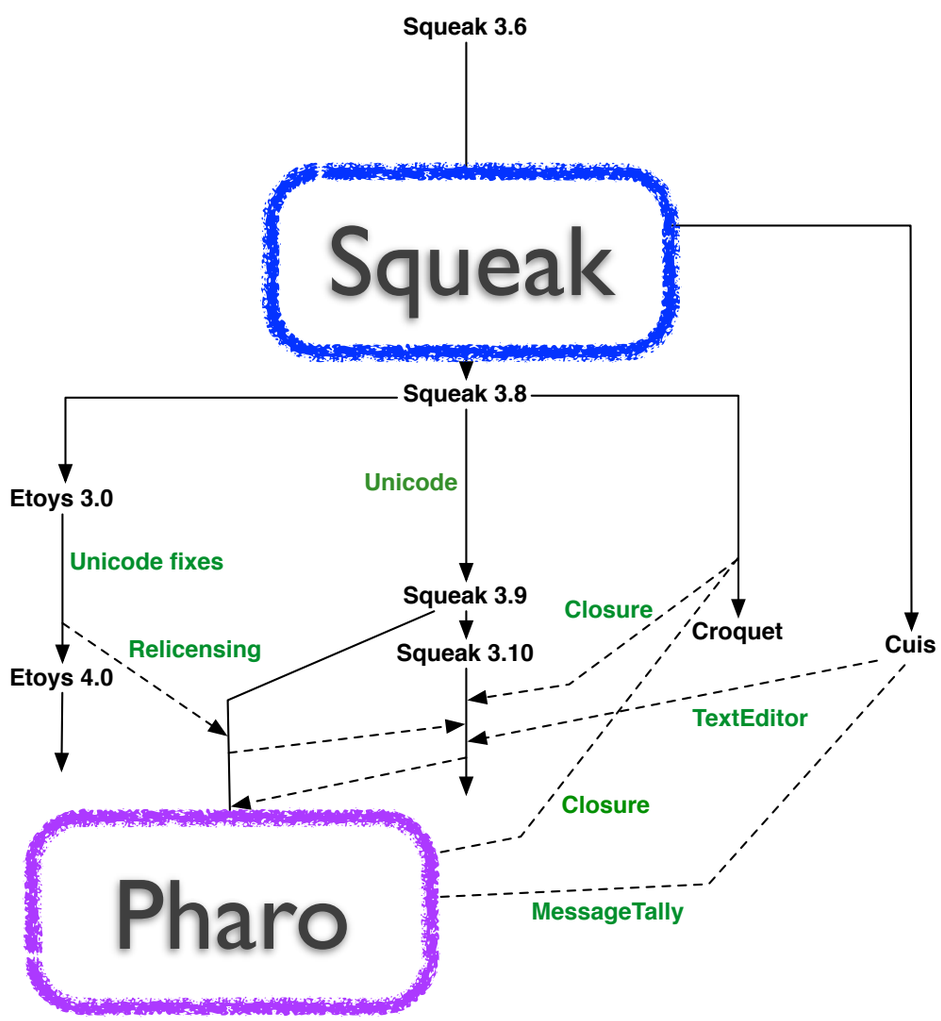




Source Code History Analysis

- Linear history
- Cross history





Querying the History of a System

Querying the History of a System

Co-change analysis	what are the entities that changed together with entity *Number* in version 3 ?

Querying the History of a System

Co-change analysis	what are the entities that changed together with entity *Number* in version 3 ?
Queries as in the past	what were the senders of the method *#asString* in Squeak 3.9?

Querying the History of a System

Co-change analysis	what are the entities that changed together with entity *Number* in version 3 ?
Queries as in the past	what were the senders of the method *#asString* in Squeak 3.9?
Queries as in the present	what are the messages sent by method #printOn: in version 2?

Querying the History of a System

Co-change analysis	what are the entities that changed together with entity *Number* in version 3 ?
Queries as in the past	what were the senders of the method *#asString* in Squeak 3.9?
Queries as in the present	what are the messages sent by method #printOn: in version 2?
Bug spot	was the method *#printString* regularly changed over the last 5 years?

Querying the History of a System

Co-change analysis	what are the entities that changed together with entity *Number* in version 3 ?
Queries as in the past	what were the senders of the method *#asString* in Squeak 3.9?
Queries as in the present	what are the messages sent by method #printOn: in version 2?
Bug spot	was the method *#printString* regularly changed over the last 5 years?
Global analysis	what is the whole history of method *#detect:ifNone:*?

Querying the History of a System

Co-change analysis	what are the entities that changed together with entity *Number* in version 3 ?
Queries as in the past	what were the senders of the method *#asString* in Squeak 3.9?
Queries as in the present	what are the messages sent by method #printOn: in version 2?
Bug spot	was the method *#printString* regularly changed over the last 5 years?
Global analysis	what is the whole history of method *#detect:ifNone:*?
Forks analysis	if the version of method *#isNil* changed in Squeak 3.9, should it be changed in Pharo?

Querying the History of a System

Co-change analysis	what are the entities that changed together with entity *Number* in version 3 ?
Queries as in the past	d *#asString* in
Queries as in the present	od #printOn: in
Bug spot	rly changed over
Global analysis	d *#detect:ifNone:*?
Forks analysis	ged in Squeak 3.9,
Comparison profiler	what are the differences or similarities of running versions 4 and 5?

We need ...

We need ...

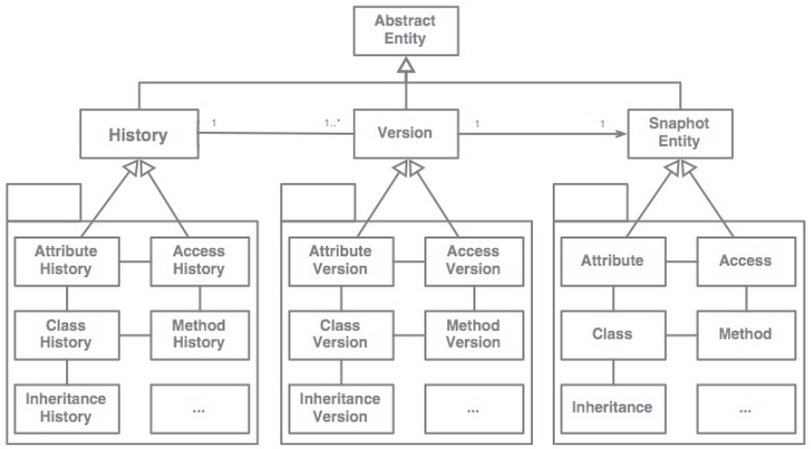
- History meta-model

We need ...

- History meta-model
  - Linear history

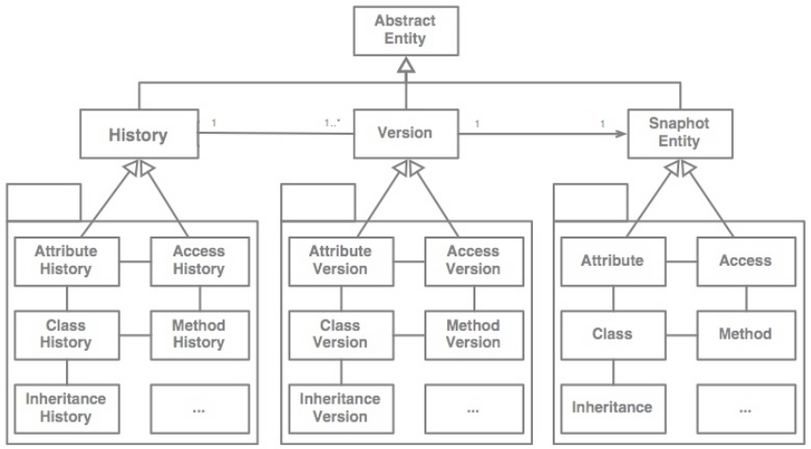
We need ...

- History meta-model
  - Linear history
  - Cross history



We need ...

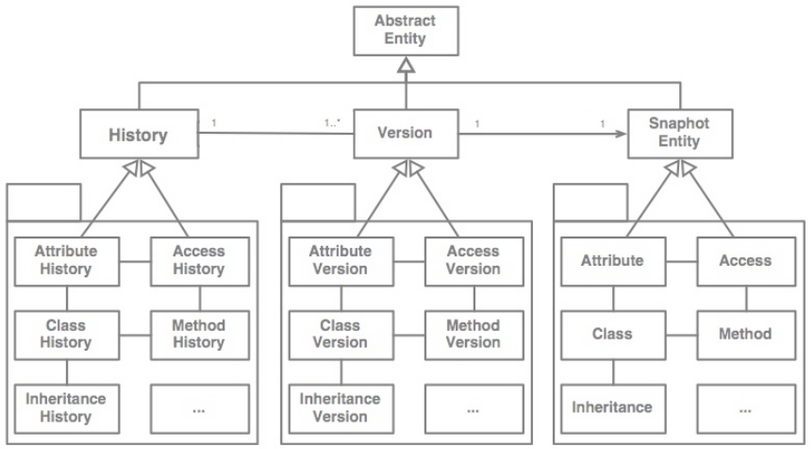
- History meta-model
  - Linear history
  - Cross history



- Infrastructure

We need ...

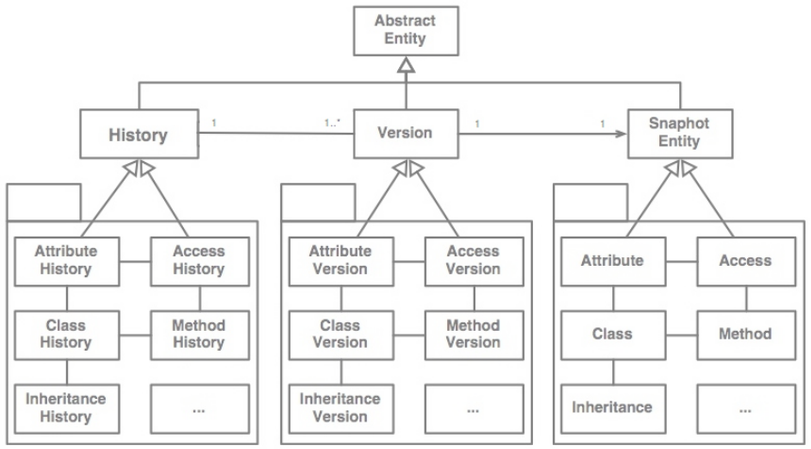
- History meta-model
  - Linear history
  - Cross history



- Infrastructure
  - Store lot of data

We need ...

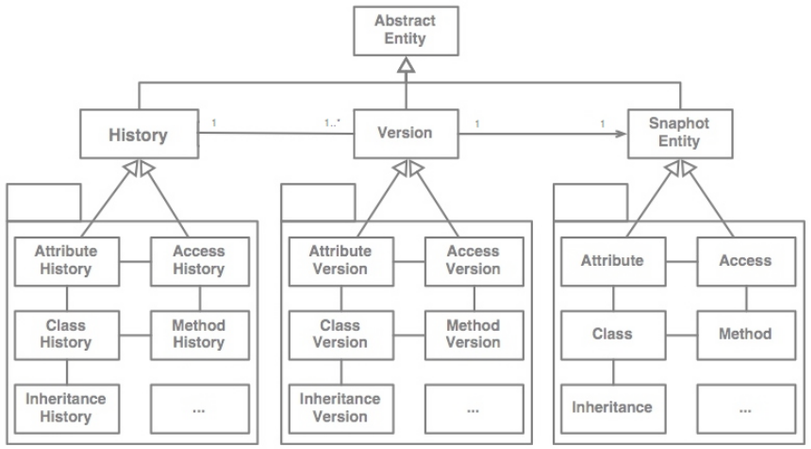
- History meta-model
  - Linear history
  - Cross history



- Infrastructure
  - Store lot of data
  - Efficient /Fast

We need ...

- History meta-model
  - Linear history
  - Cross history



- Infrastructure
  - Store lot of data
  - Efficient / Fast
  - Accessible

Multiple Meta-Models

Smalltalk model

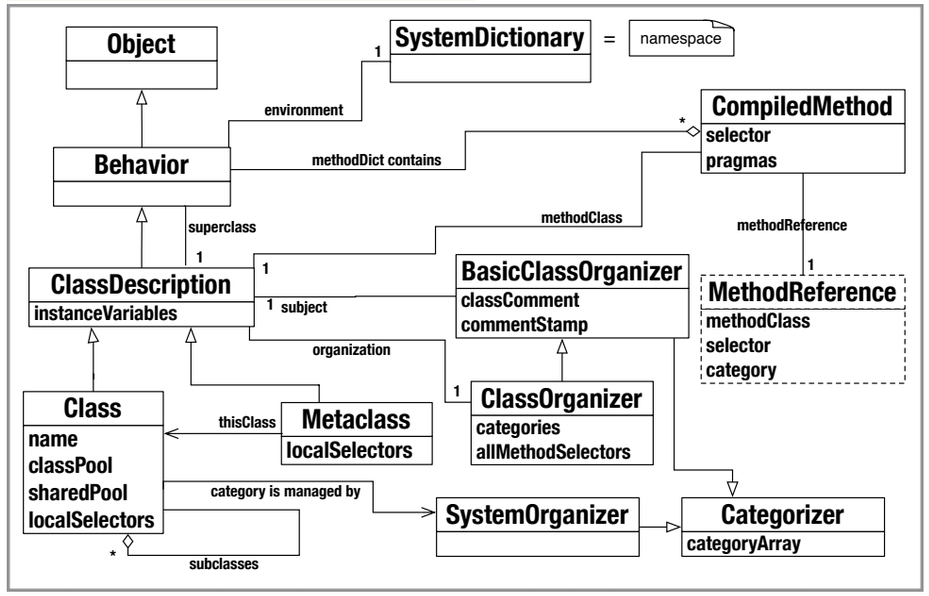
RB source code model

MC source code model

Tools source code model

Multiple Meta-Models

Smalltalk model



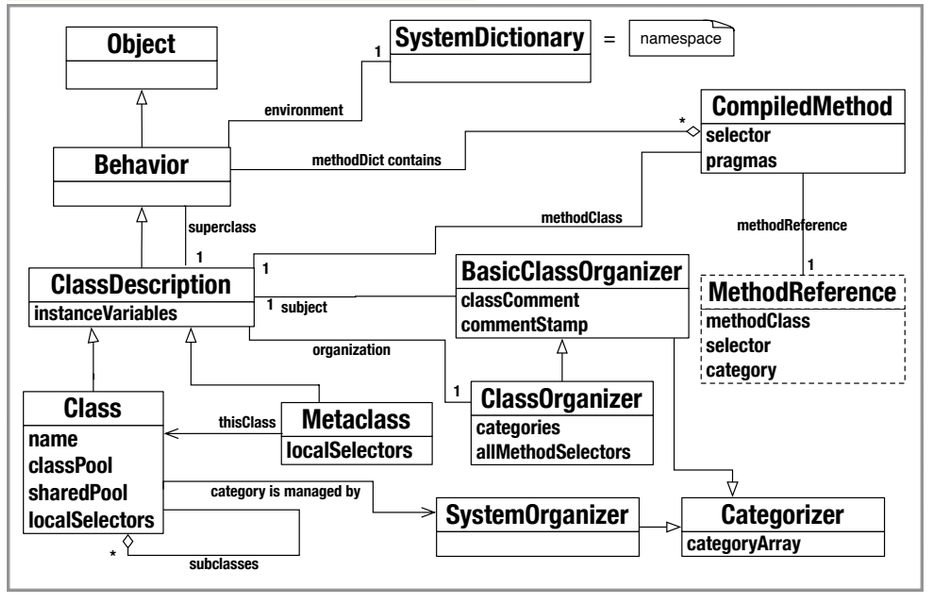
RB source code model

MC source code model

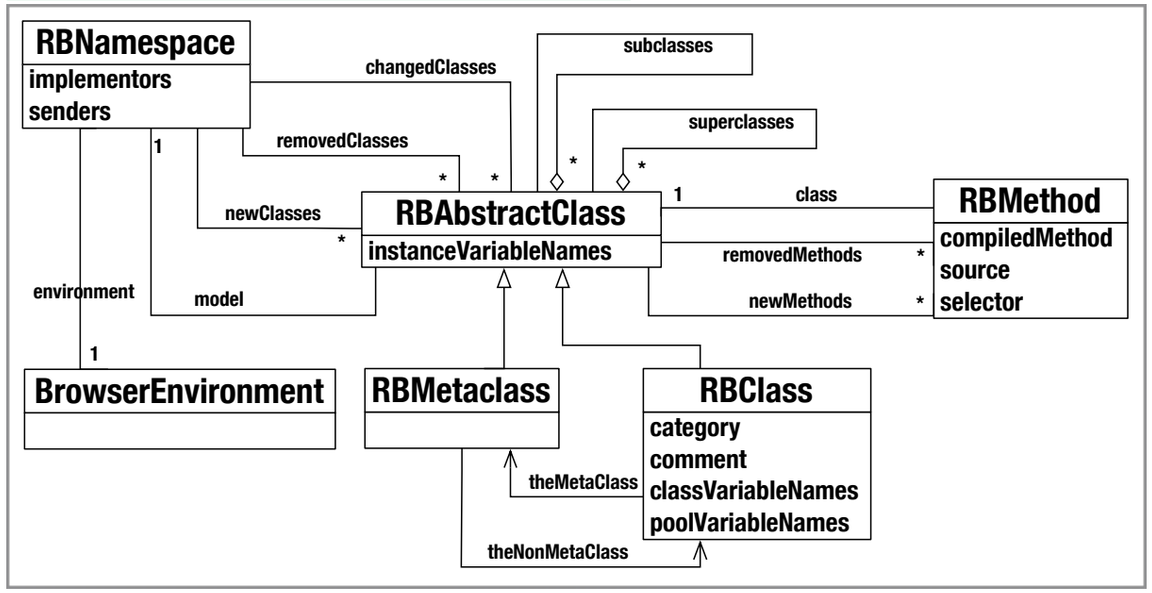
Tools source code model

Multiple Meta-Models

Smalltalk model



RB source code model

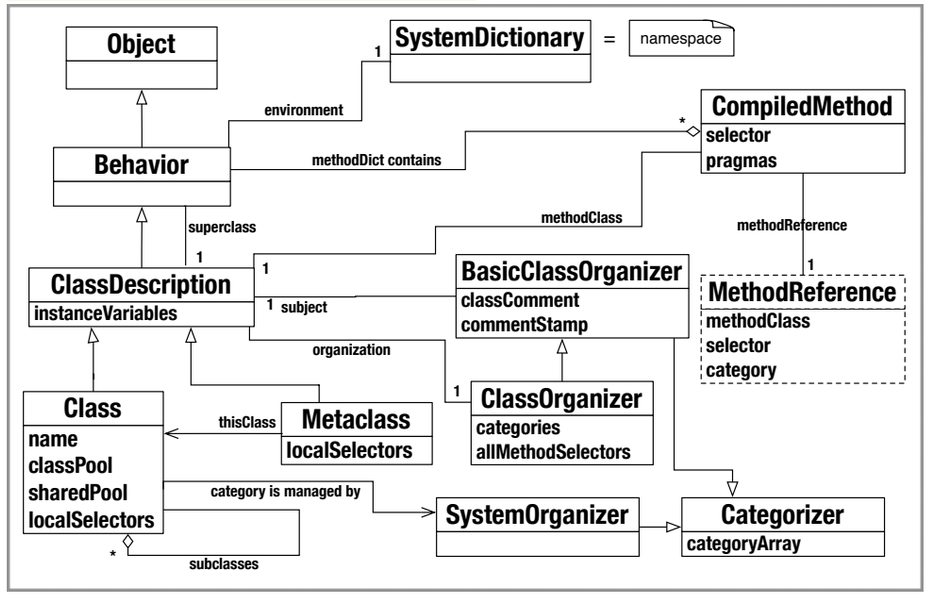


MC source code model

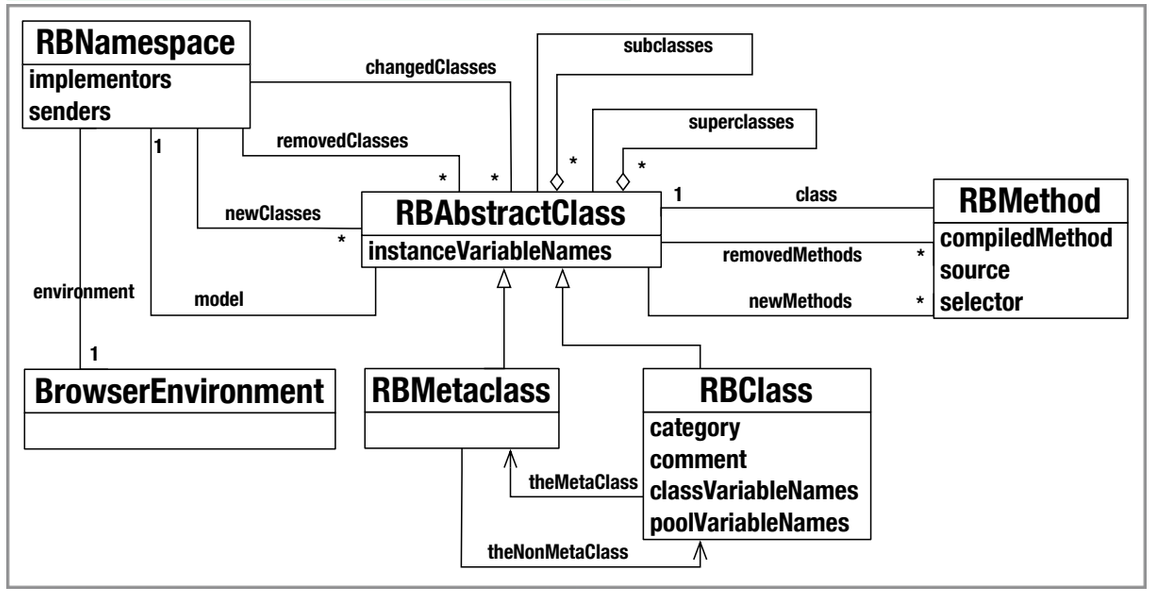
Tools source code model

Multiple Meta-Models

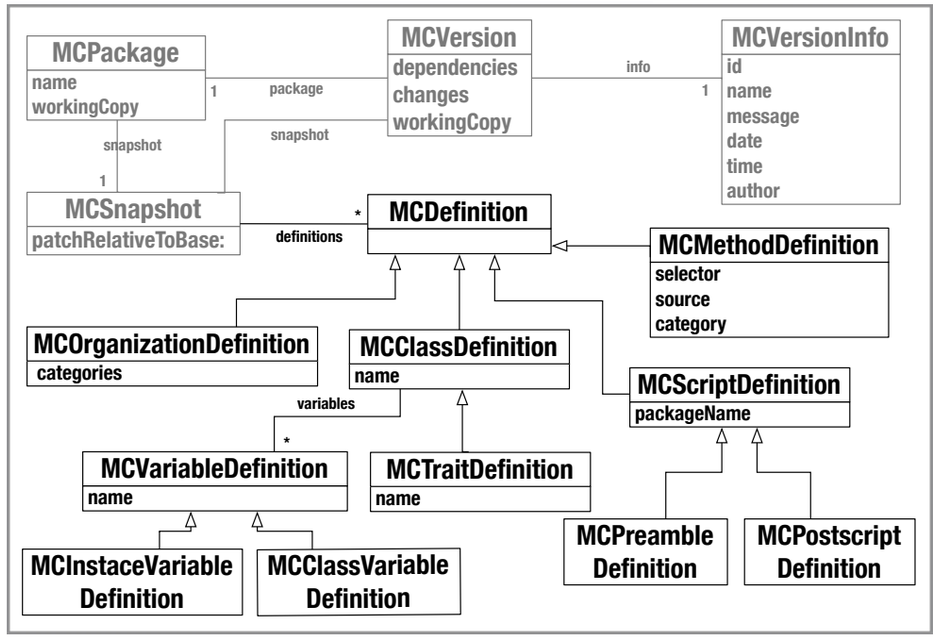
Smalltalk model



RB source code model



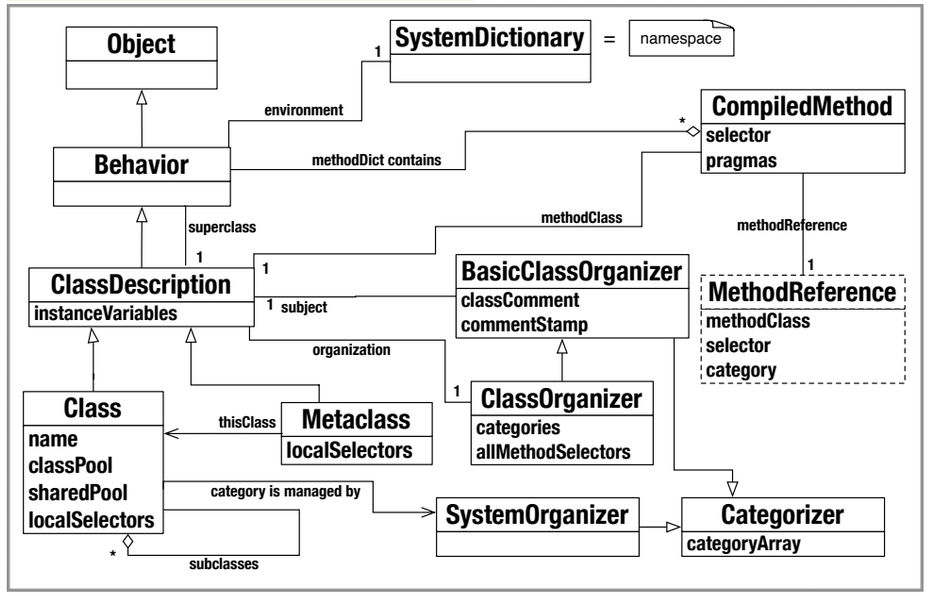
MC source code model



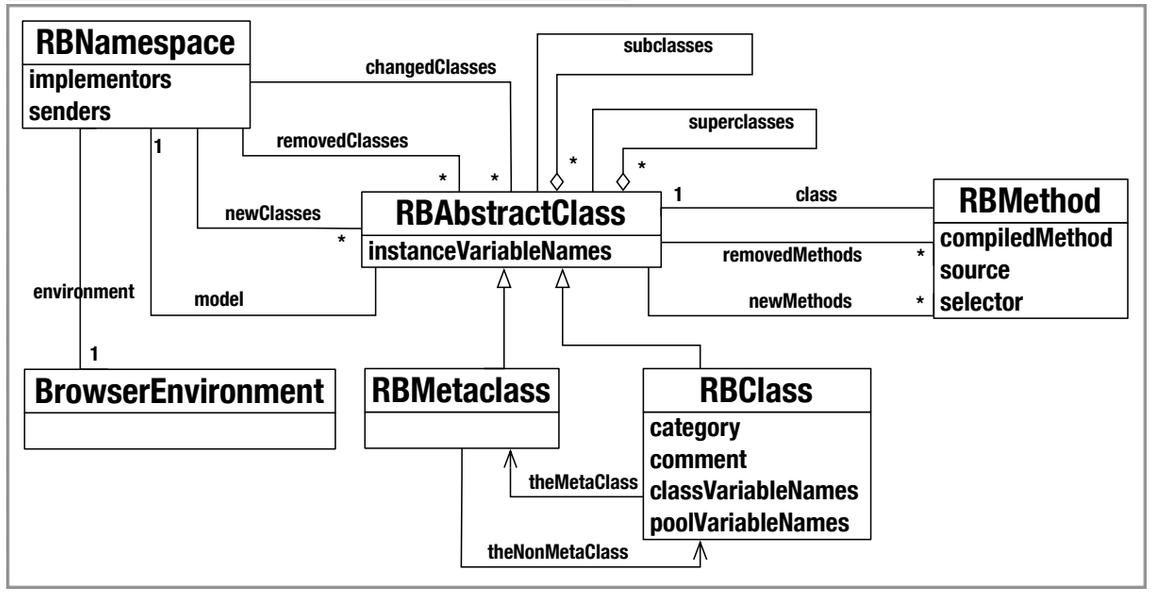
Tools source code model

Multiple Meta-Models

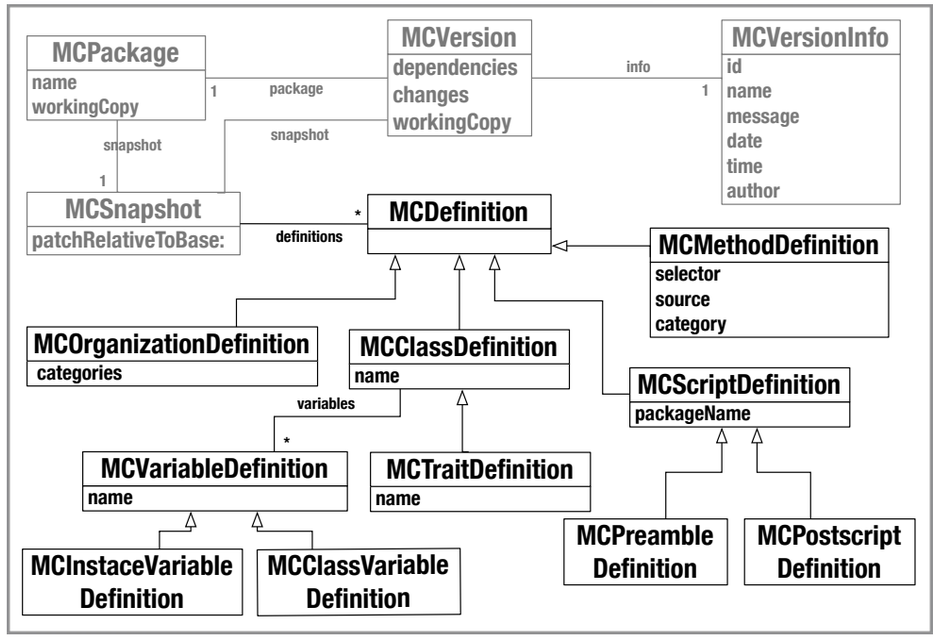
Smalltalk model



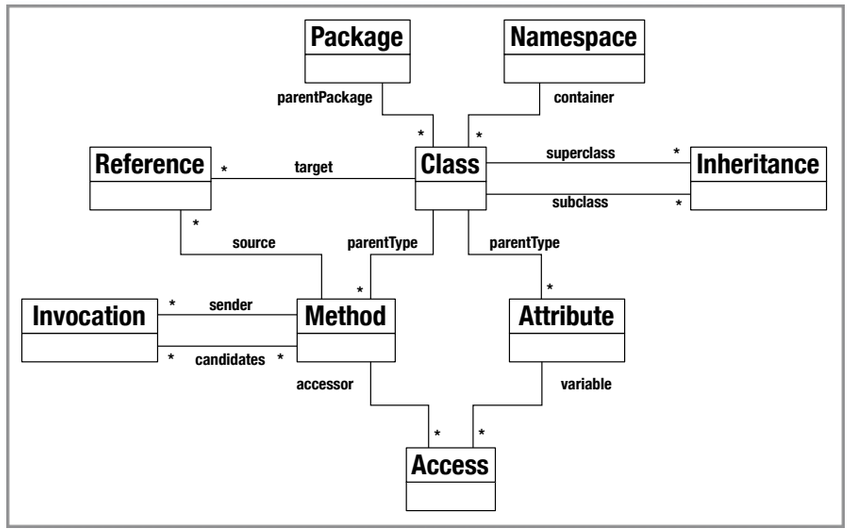
RB source code model



MC source code model

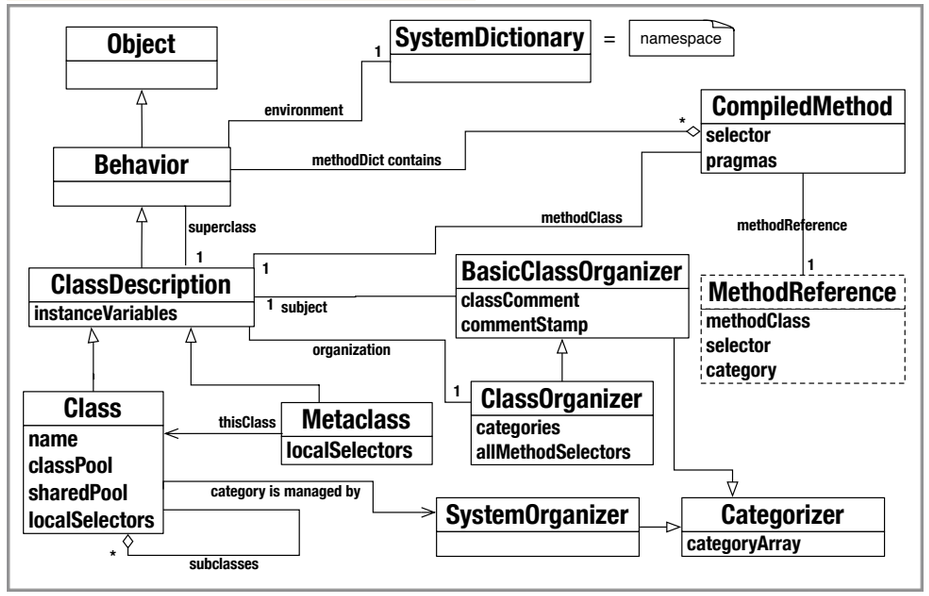


Tools source code model

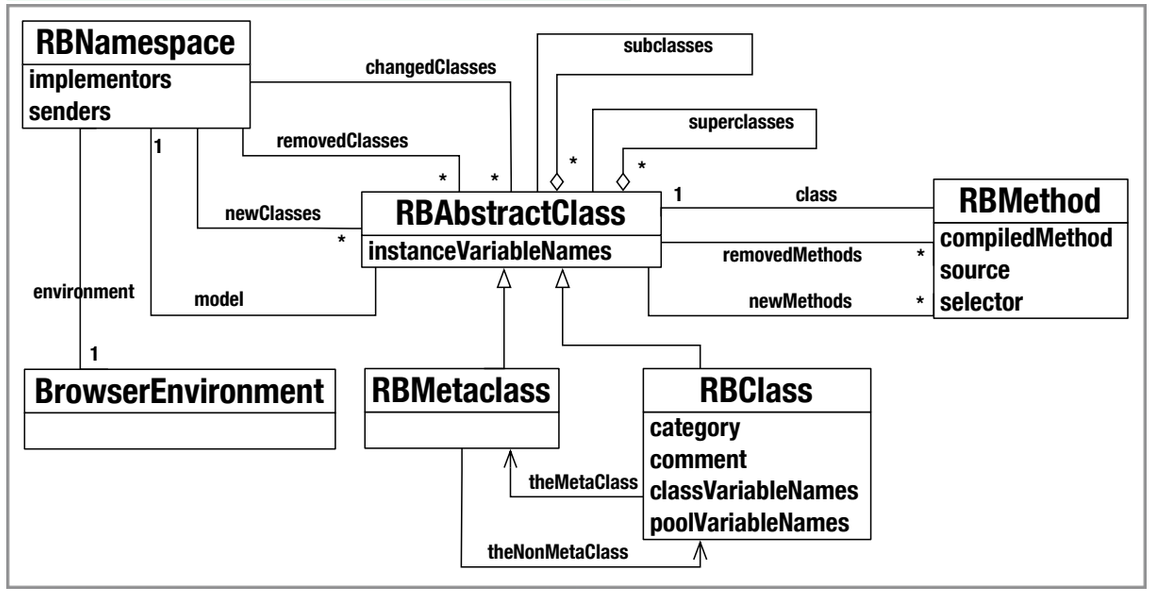


Multiple Meta-Models

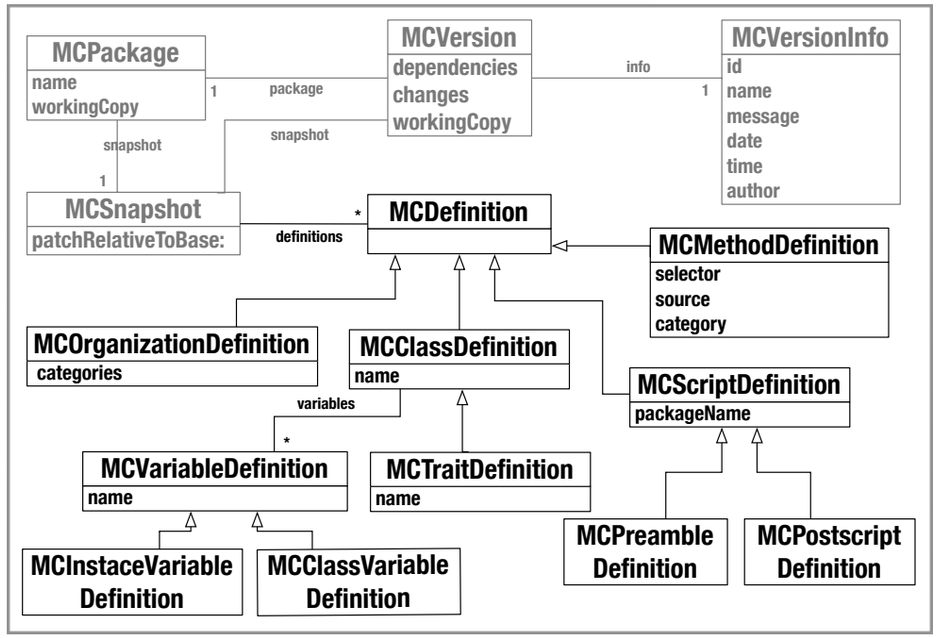
Smalltalk model



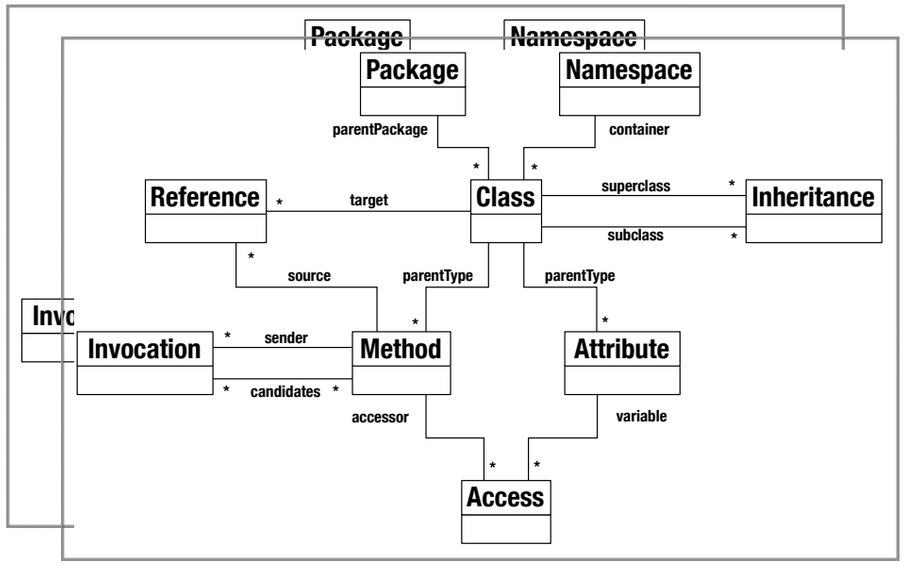
RB source code model



MC source code model

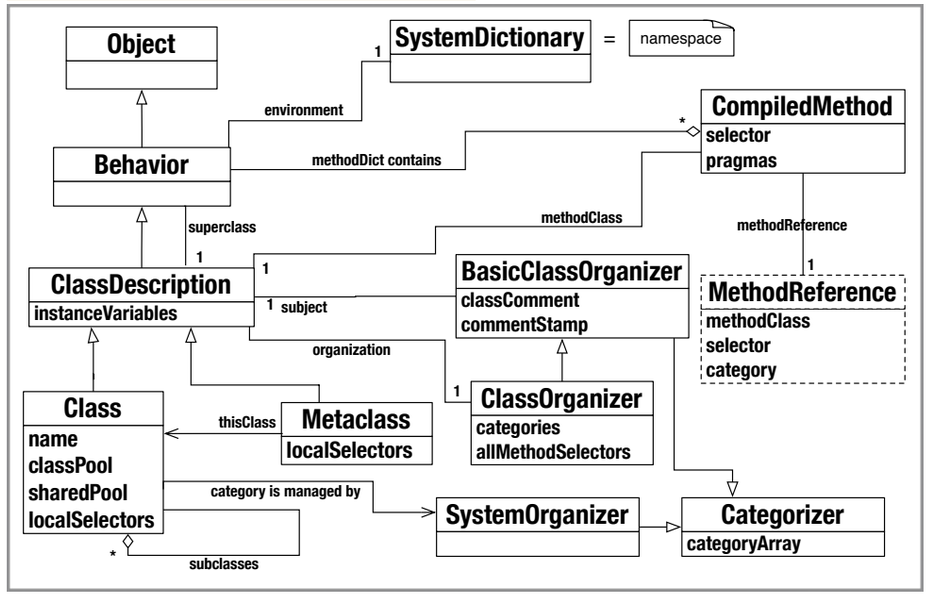


Tools source code model

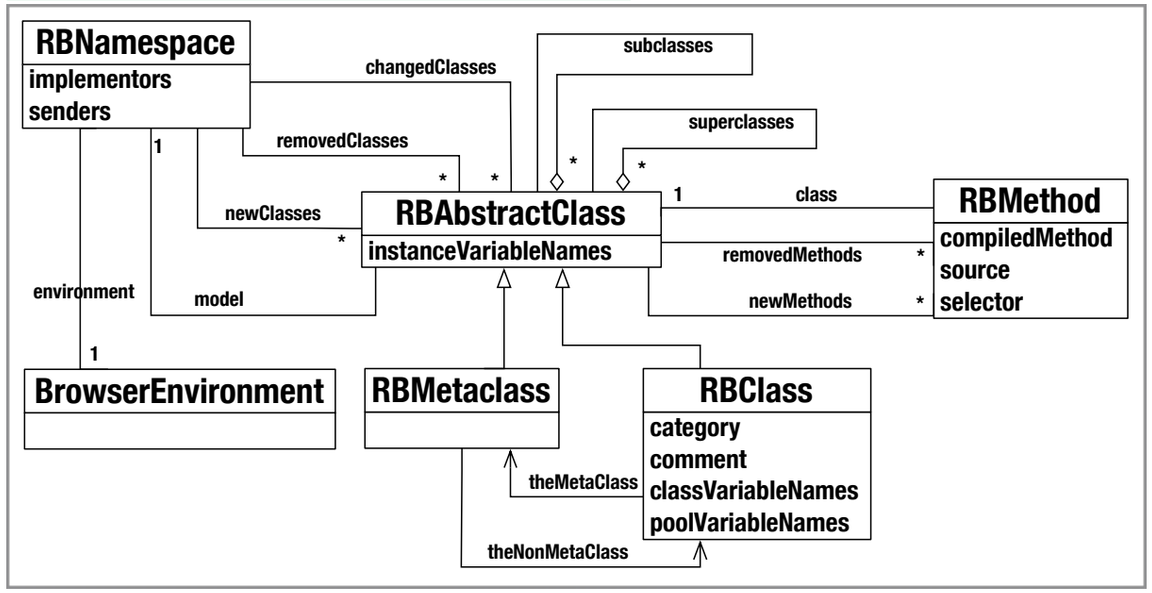


Multiple Meta-Models

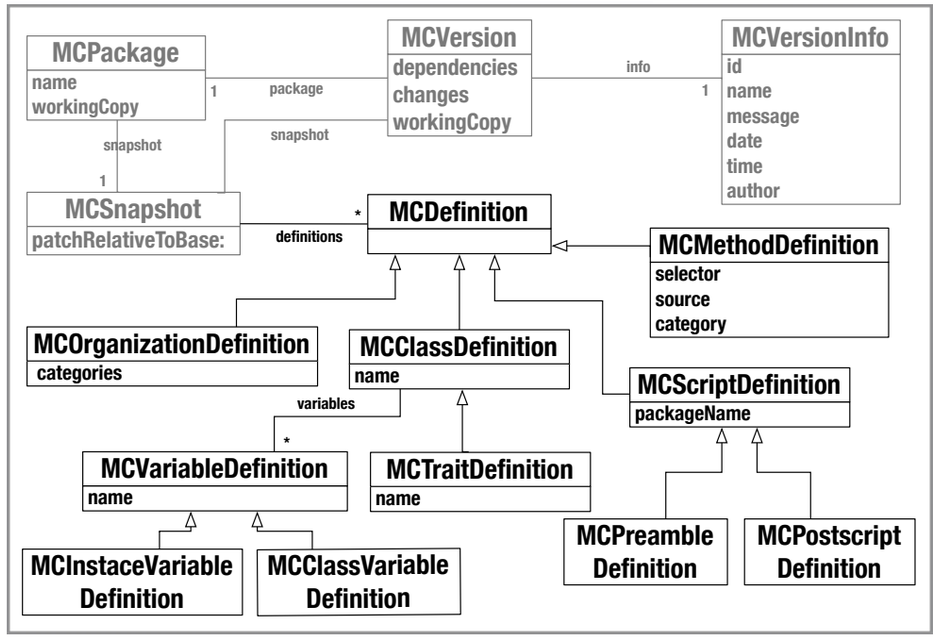
Smalltalk model



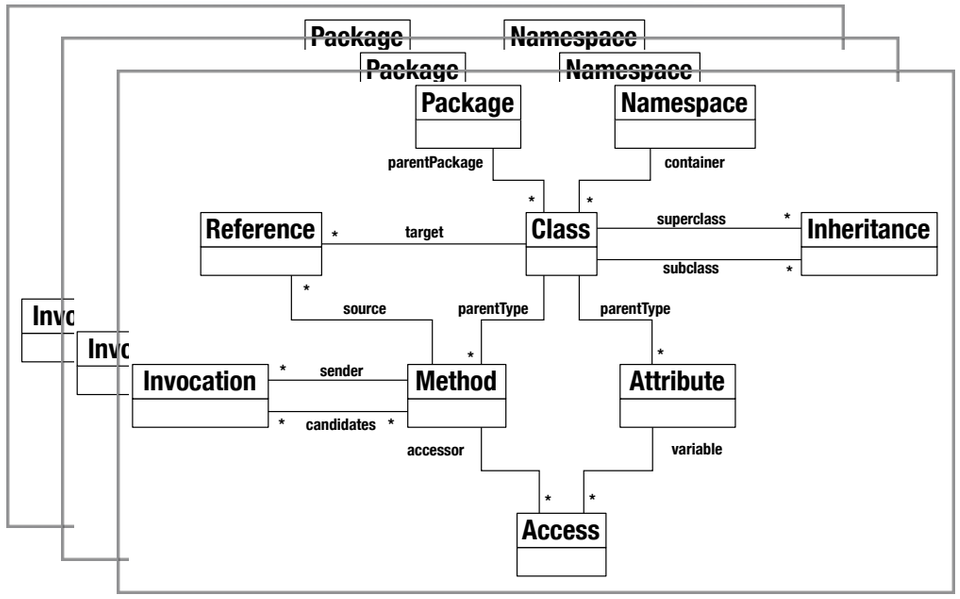
RB source code model



MC source code model

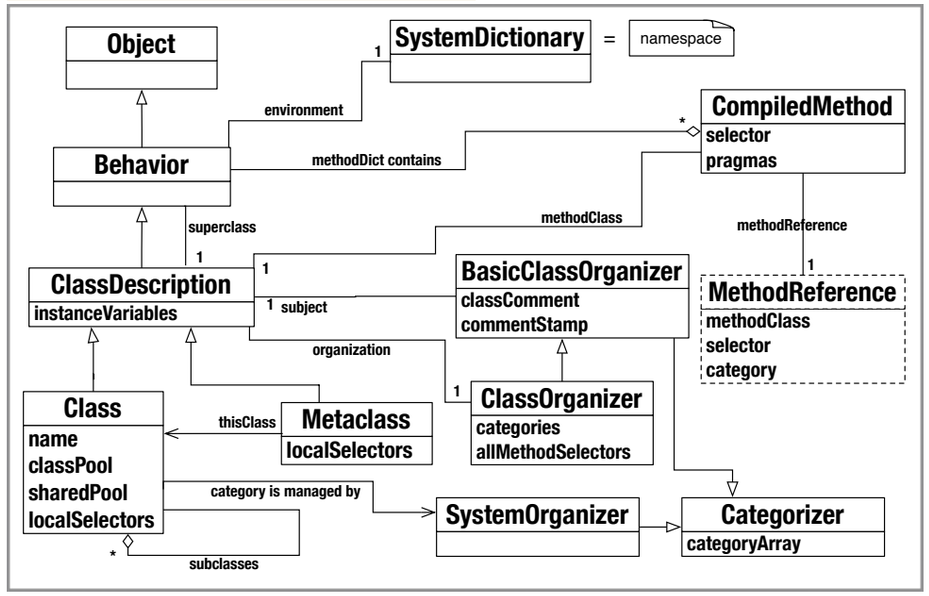


Tools source code model

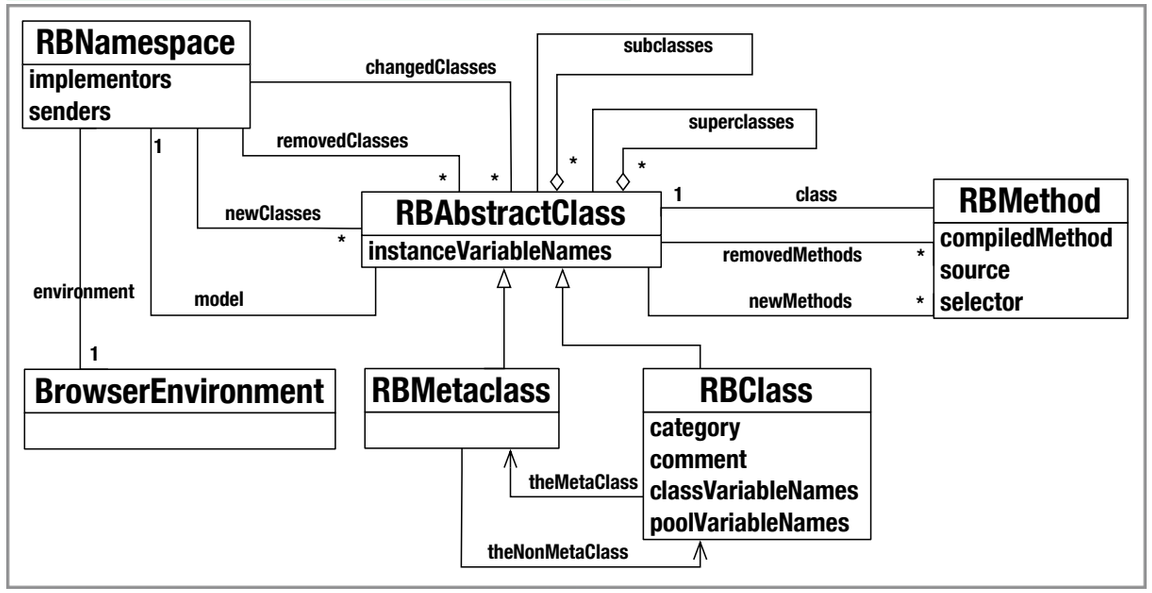


Multiple Meta-Models

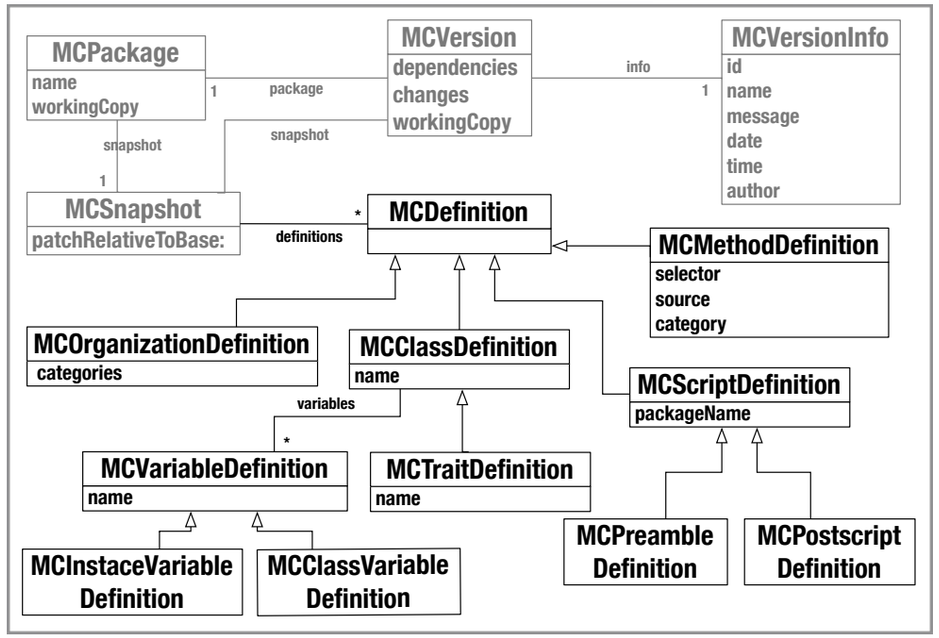
Smalltalk model



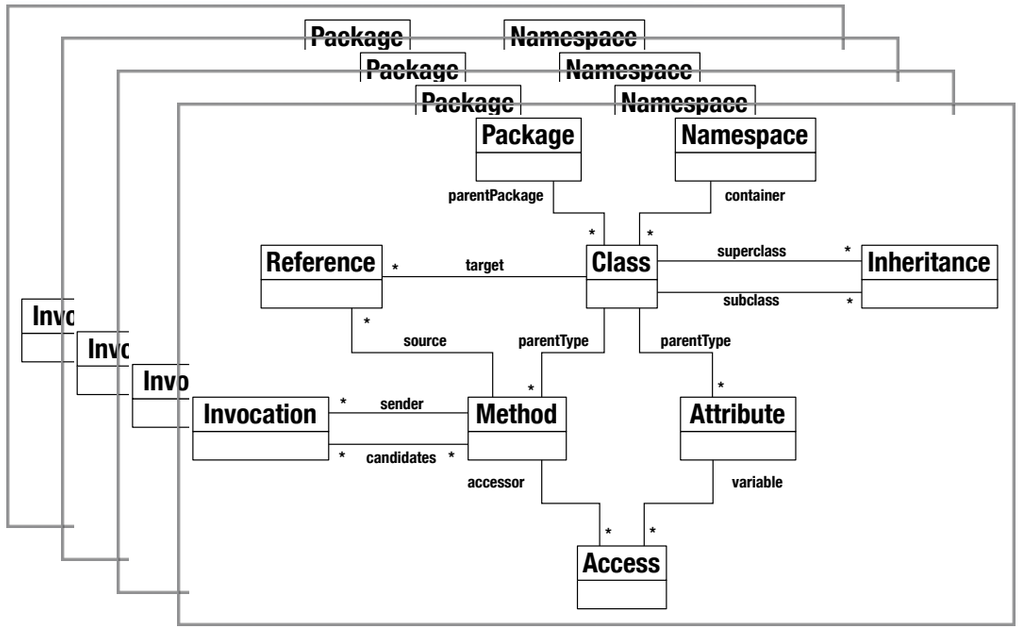
RB source code model



MC source code model

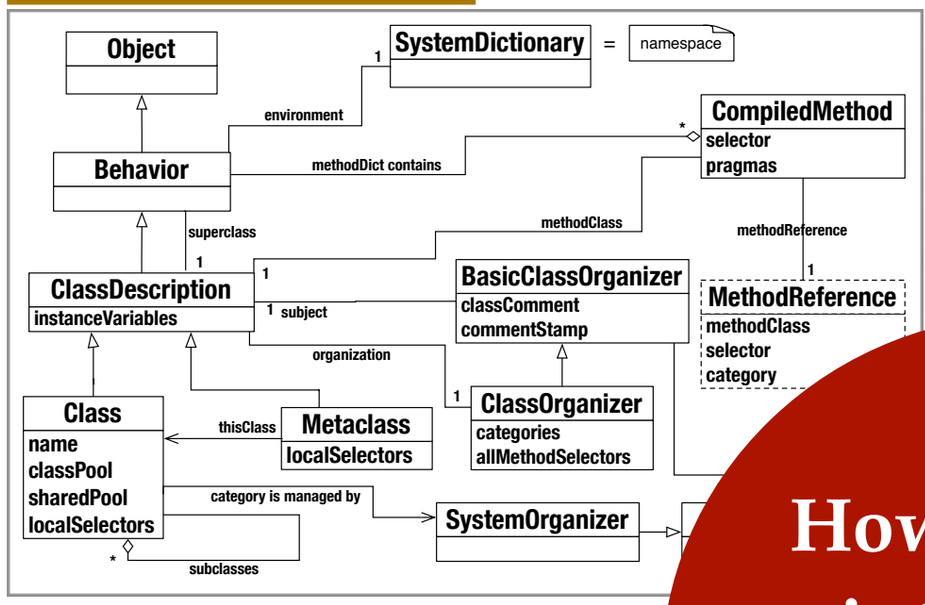


Tools source code model

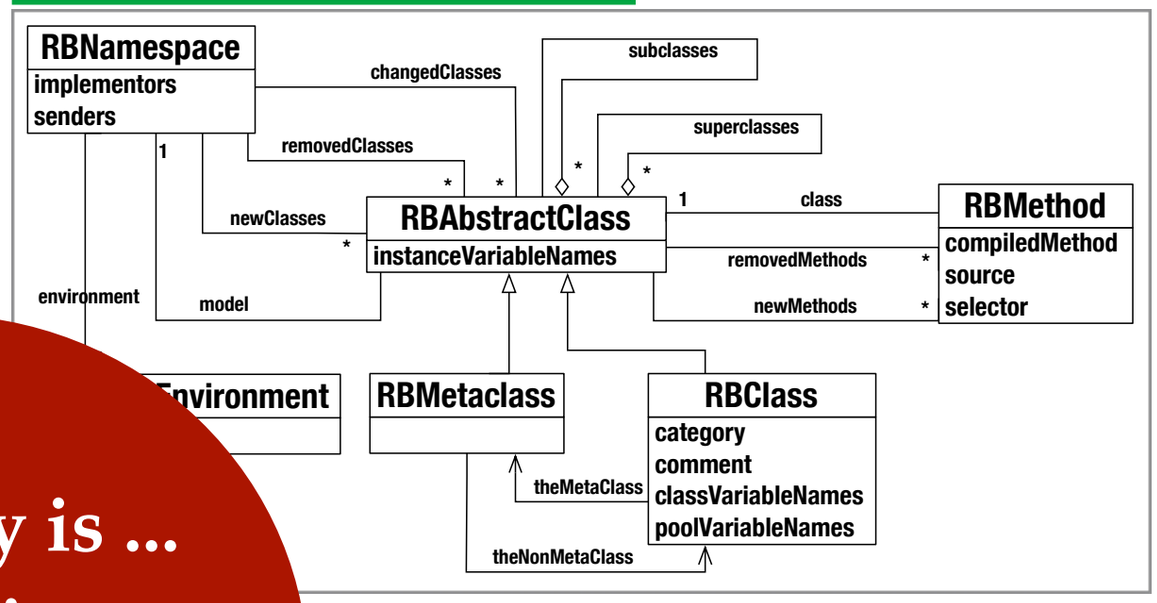


Multiple Meta-Models

Smalltalk model

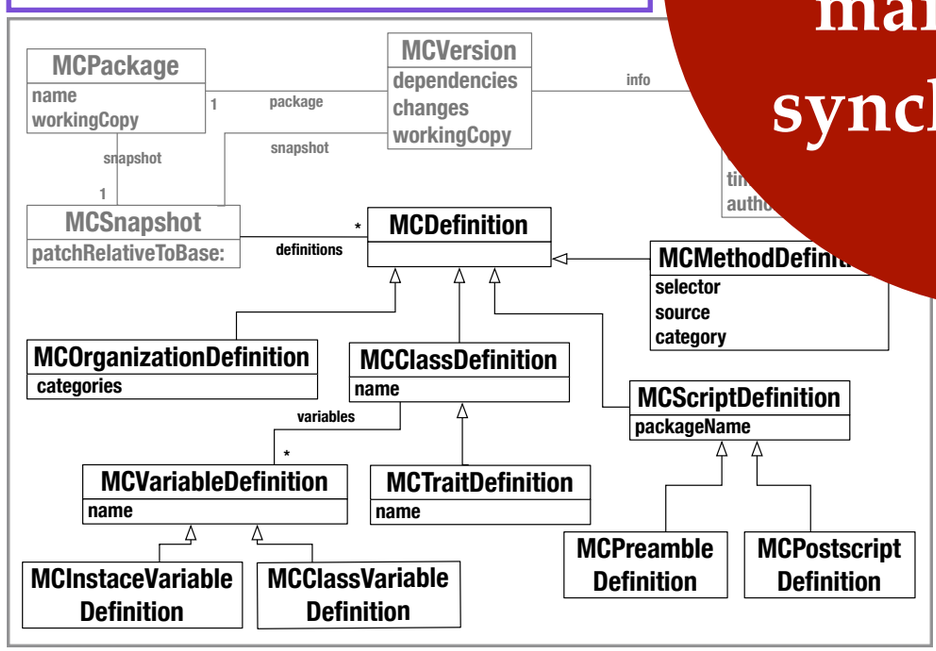


RB source code model

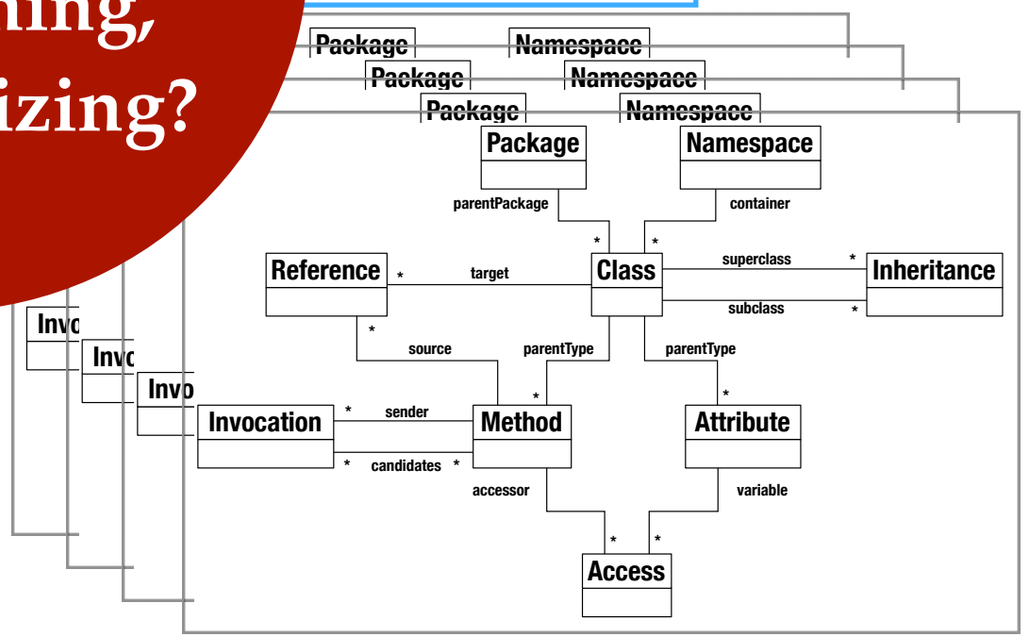


How easy is ...
interacting,
maintaining,
synchronizing?

MC source code model



ce code model



Meta-Models Plague

Meta-Models Plague

- Multiple models API

Meta-Models Plague

- Multiple models ≠ API

Meta-Models Plague

- Multiple models ≠ API
  - conversion or transformation

Meta-Models Plague

- Multiple models ≠ API
  - conversion or transformation
  - duplication of data, tests

Meta-Models Plague

- Multiple models ≠ API
  - conversion or transformation
  - duplication of data, tests
  - more maintenance tasks

Meta-Models Plague

- Multiple models ≠ A
  - conversion or t ation
  - duplic ata, tests
  - aintenance tasks

Mass duplication of efforts/tasks

Versioning Systems vs. Source Code Meta-Model

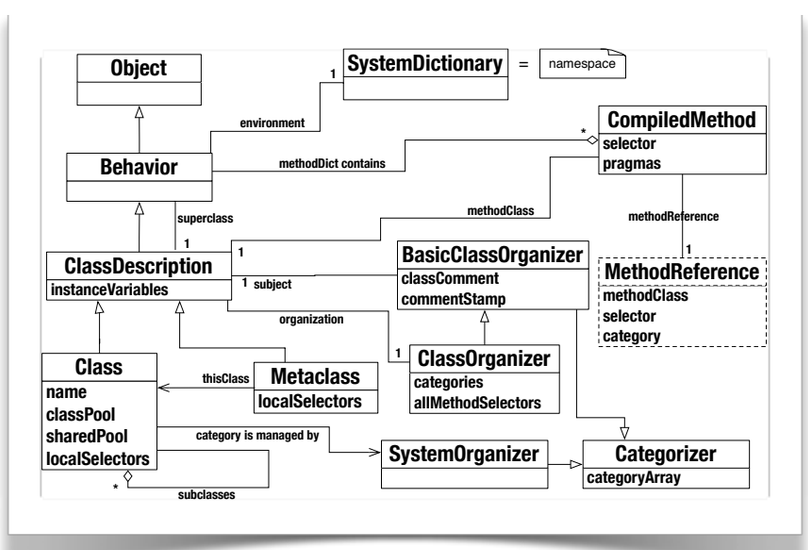
Versioning Systems vs. Source Code Meta-Model

Versioning System

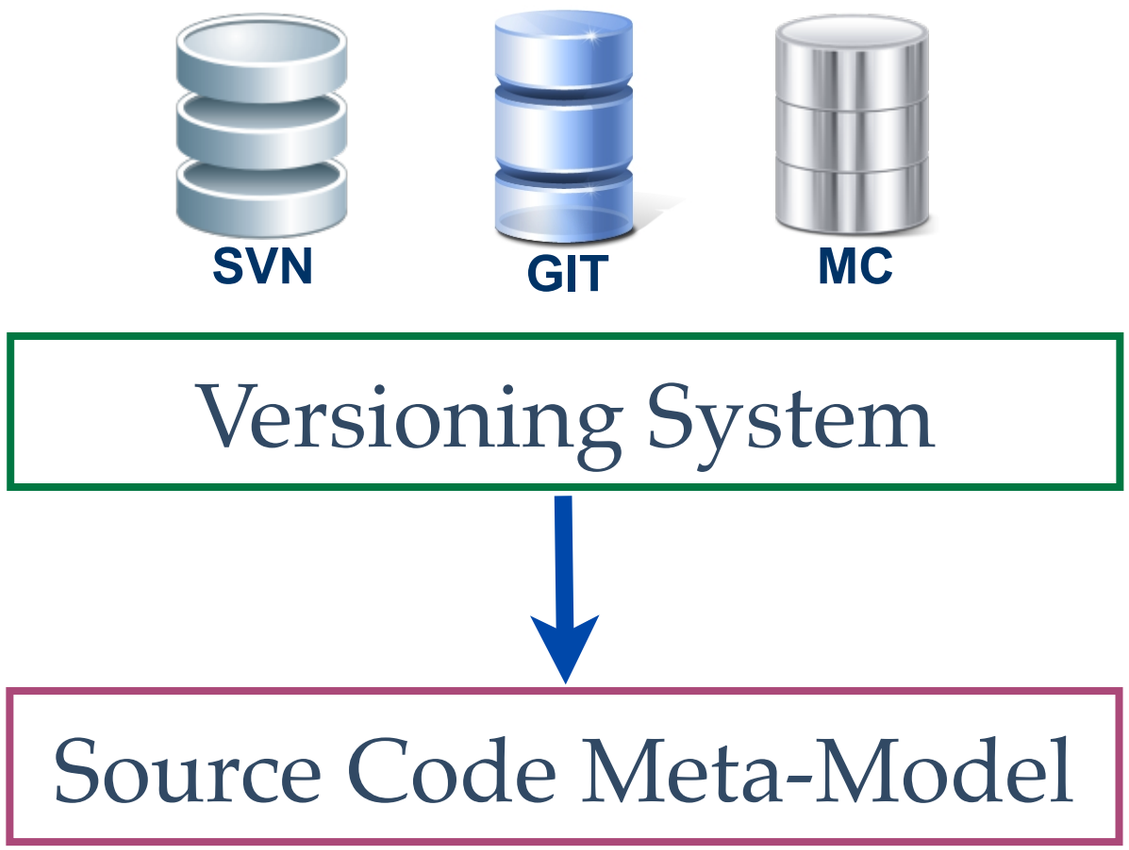
Versioning Systems vs. Source Code Meta-Model

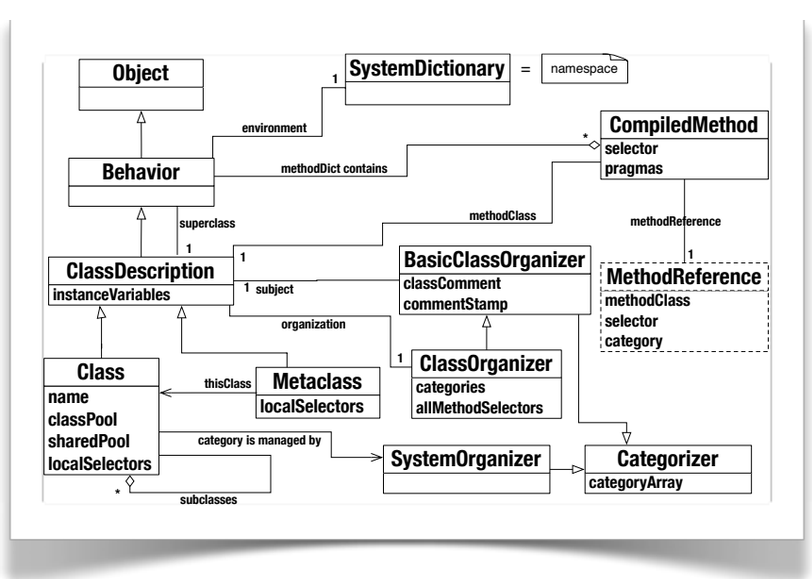
Versioning System

Source Code Meta-Model



Versioning Systems vs. Source Code Meta-Model



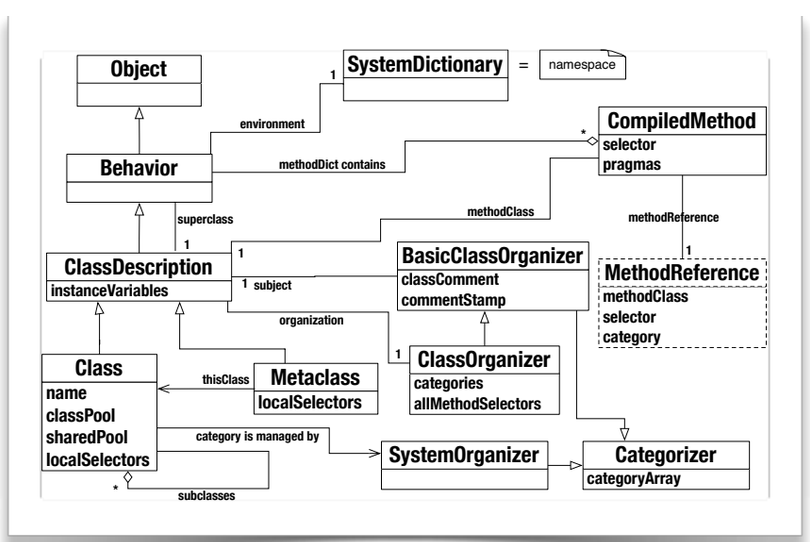


Versioning Systems vs. Source Code Meta-Model

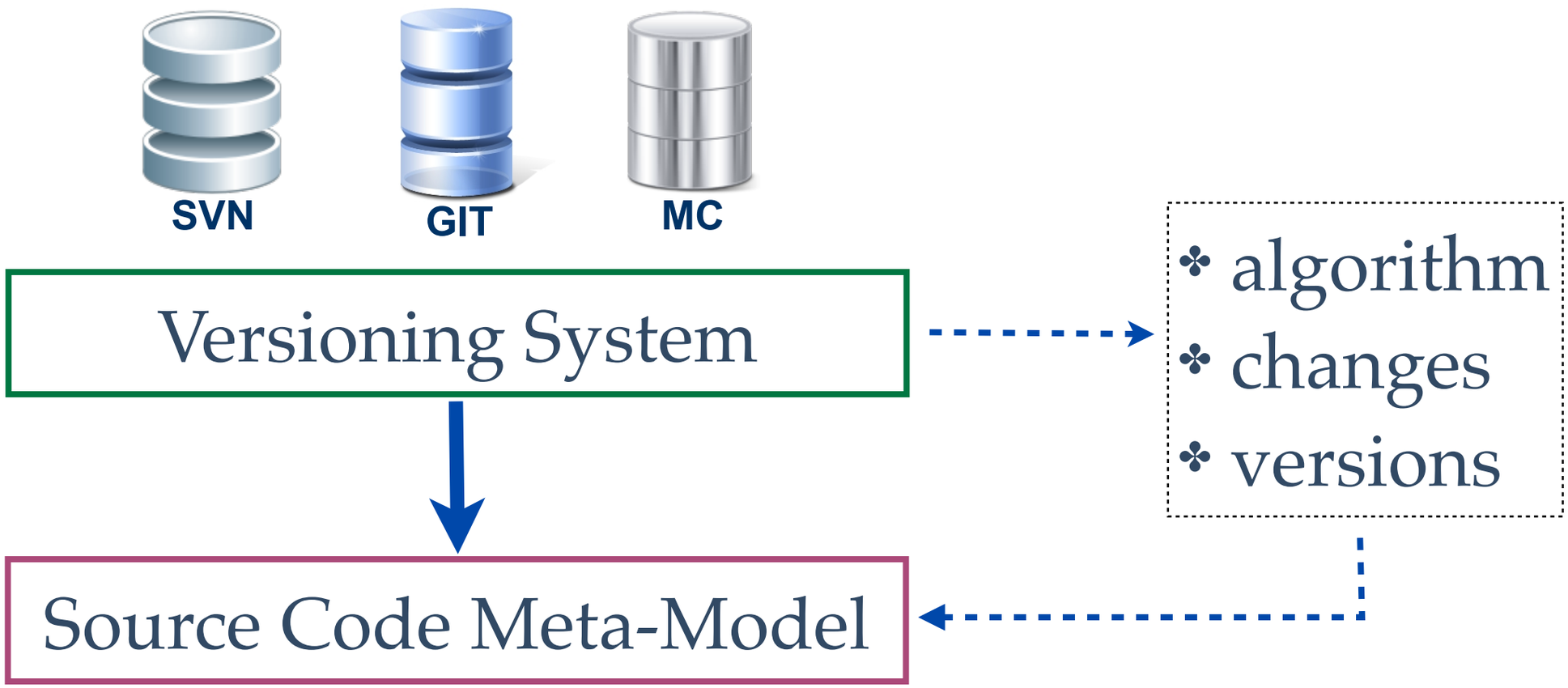
Versioning System

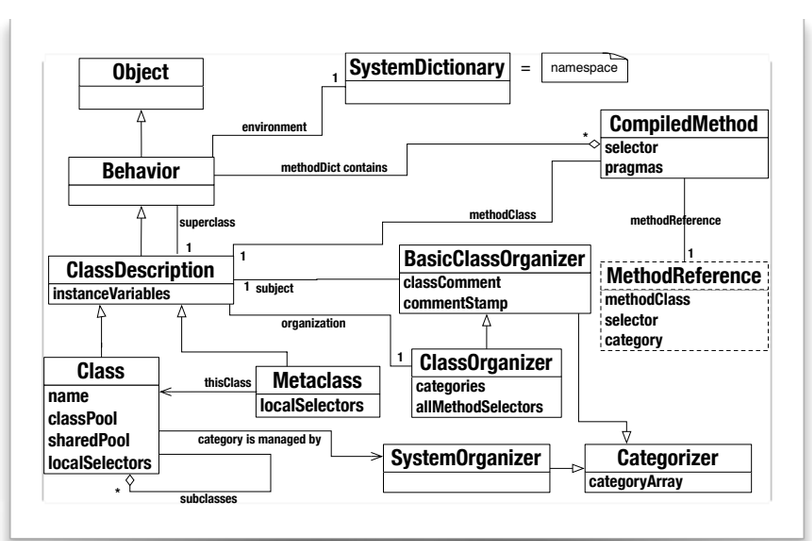
- algorithm
- changes
- versions

Source Code Meta-Model

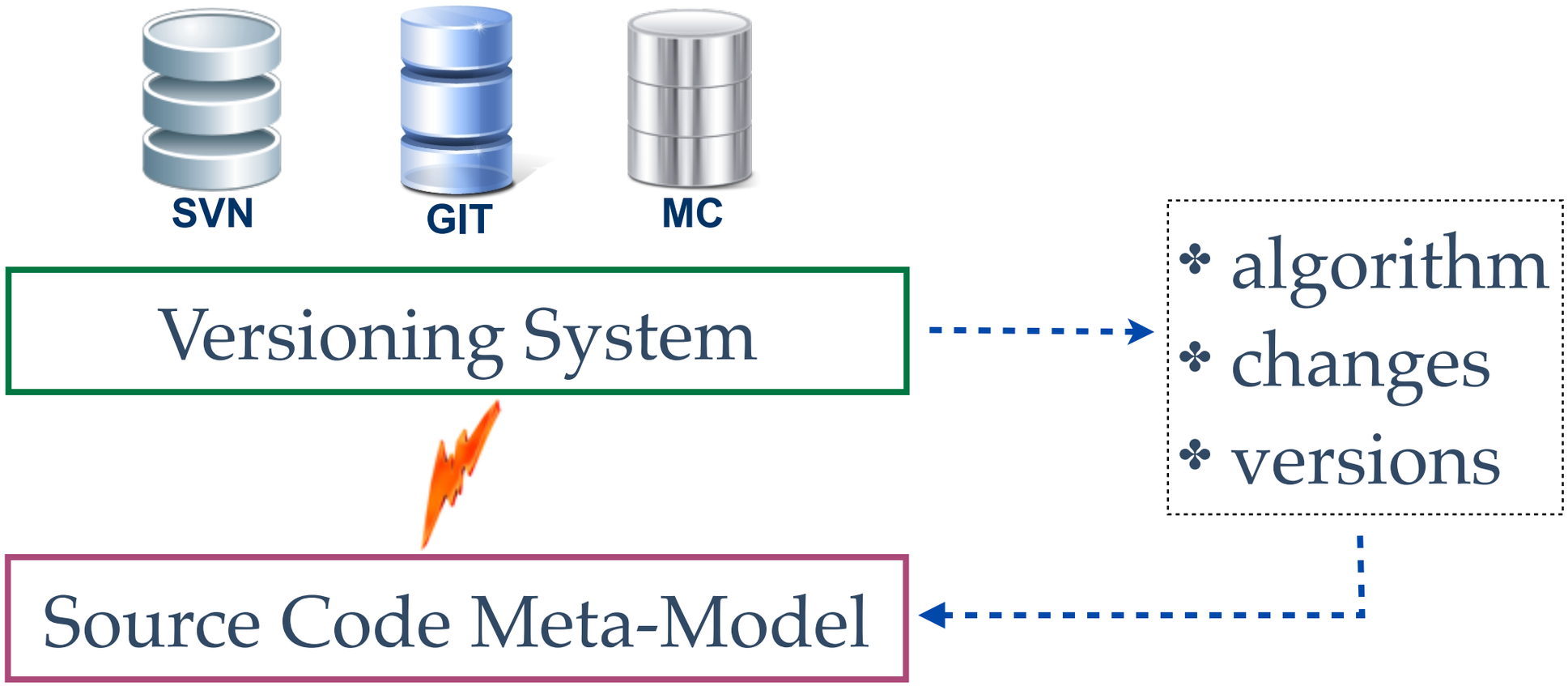


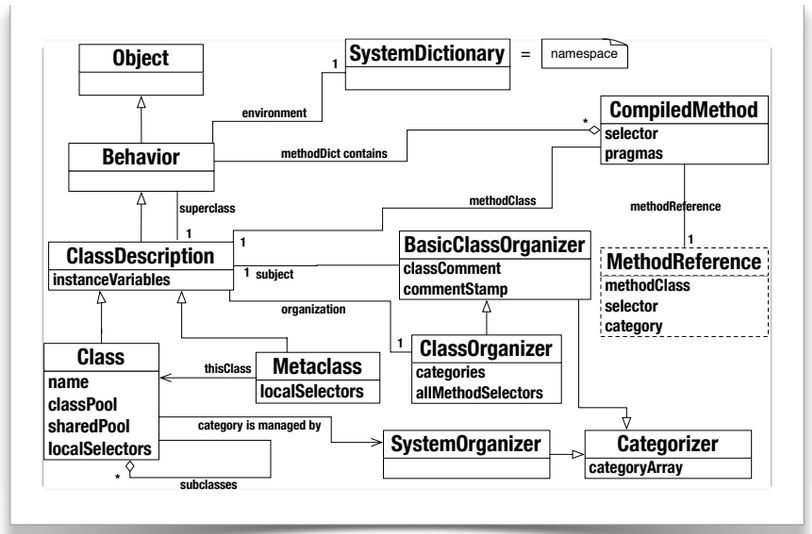
Versioning Systems vs. Source Code Meta-Model





Versioning Systems vs. Source Code Meta-Model





Requirements for Source Code and History Modeling

Requirements for Source Code and History Modeling

- No duplication of meta-models

Requirements for Source Code and History Modeling

- No duplication of meta-models
- Model update as cheap as possible

Requirements for Source Code and History Modeling

- No duplication of meta-models

- Model update as cheap as possible
- Tool reusability relying on common APIs

Unifying and Foundational Model Infrastructure

Unifying and Foundational Model Infrastructure

Unifying and Foundational Model Infrastructure

- Provide a common API (structural & runtime level)

Unifying and Foundational Model Infrastructure

- Provide a common API (structural & runtime level)
- Allow tools to interact and integrate directly with the host environment

Unifying and Foundational Model Infrastructure

- Provide a common API (structural & runtime level)
- Allow tools to interact and integrate directly with the host environment
- Become the foundational model in Pharo

Unifying and Foundational Model Infrastructure

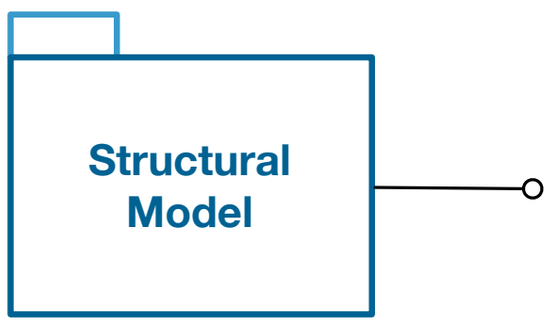
- Provide a common API (structural & runtime level)
- Allow tools to interact and integrate directly with the host environment
- Become the foundational model in Pharo
- Support history analysis

Unifying and Foundational Model Infrastructure

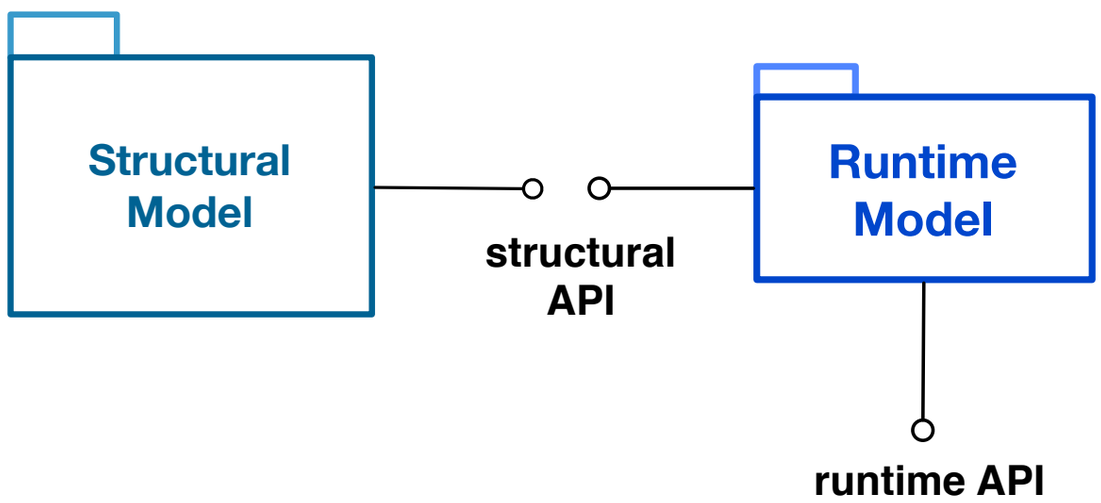
- Provide a common API (structural & runtime level)
- Allow tools to interact and integrate directly with the host environment
- Become the foundational model in Pharo
- Support history analysis

Ring Overview

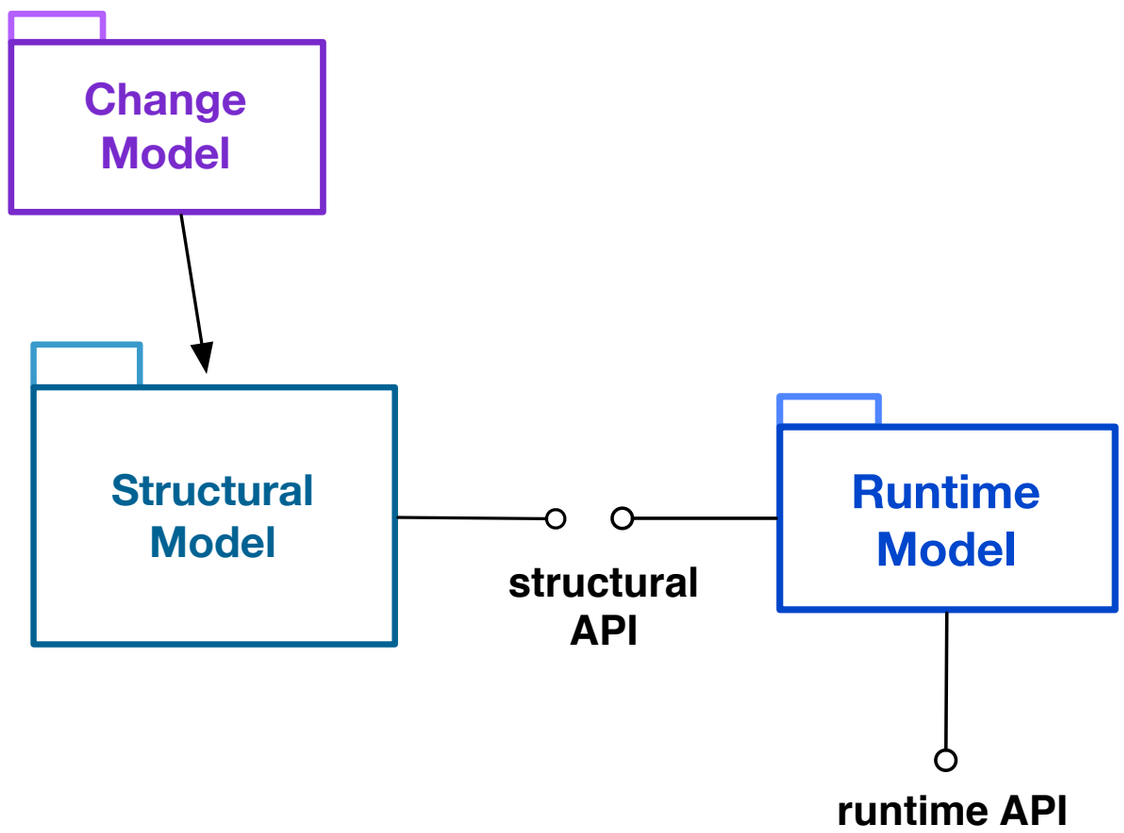
Ring Overview



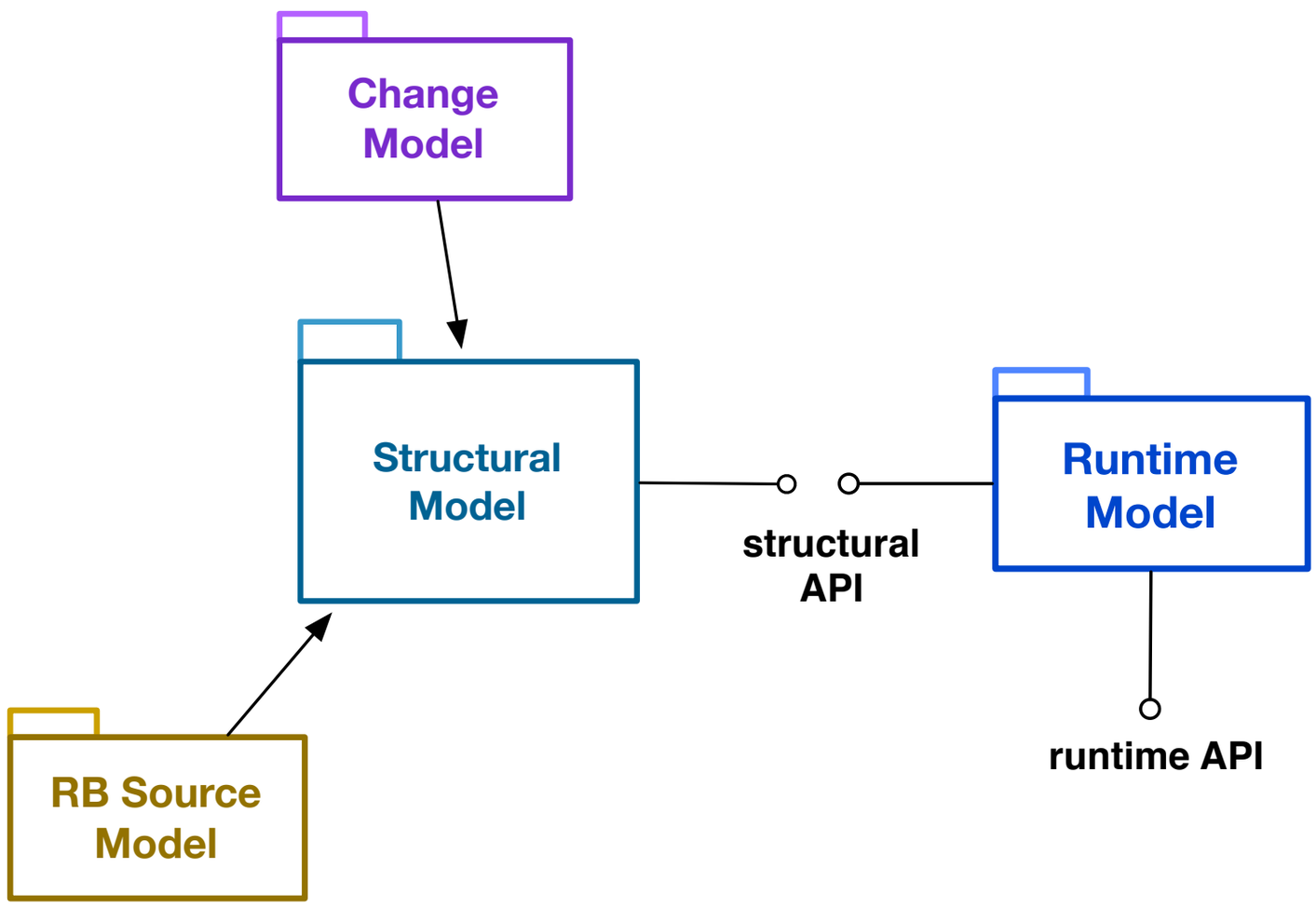
Ring Overview



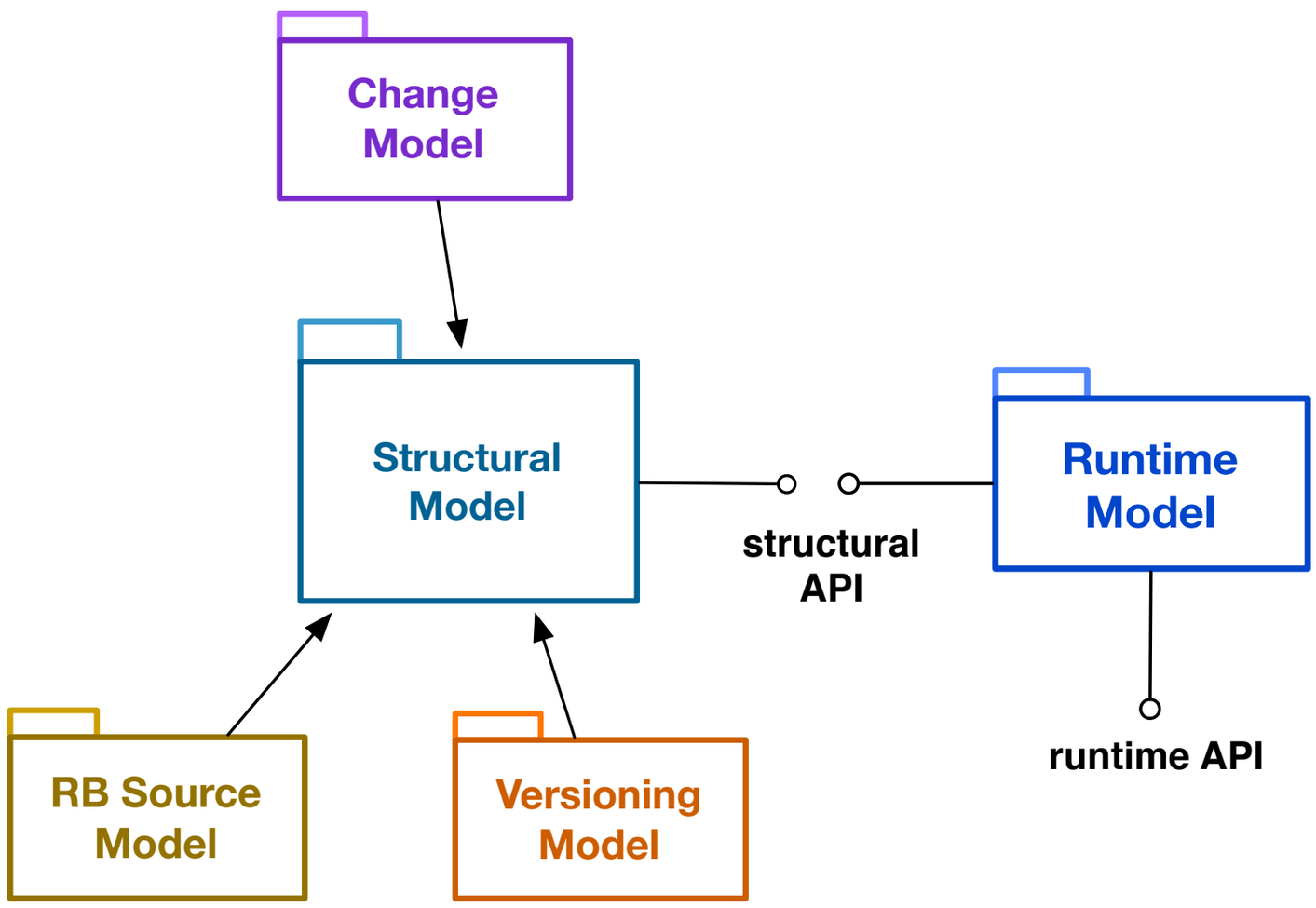
Ring Overview



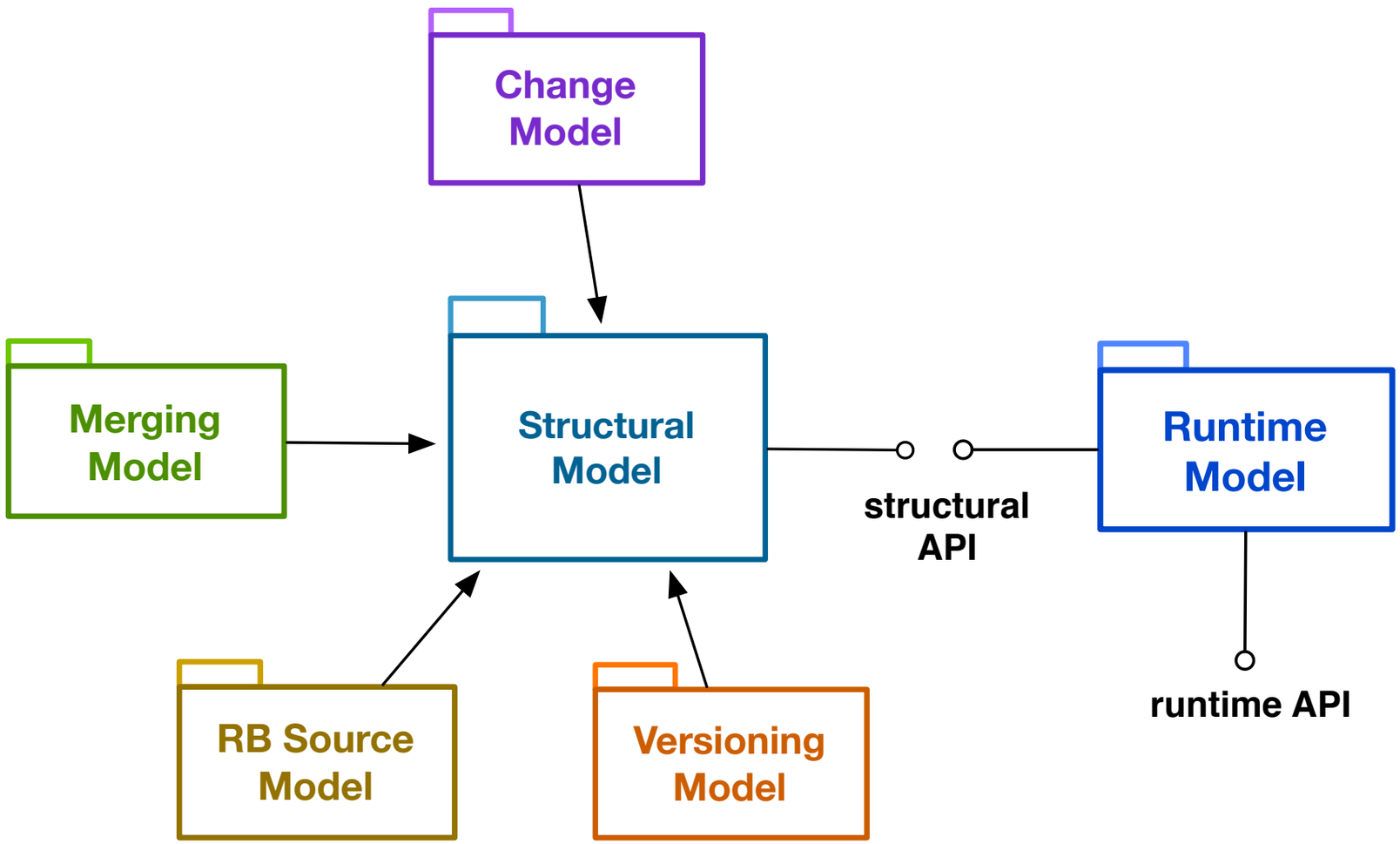
Ring Overview



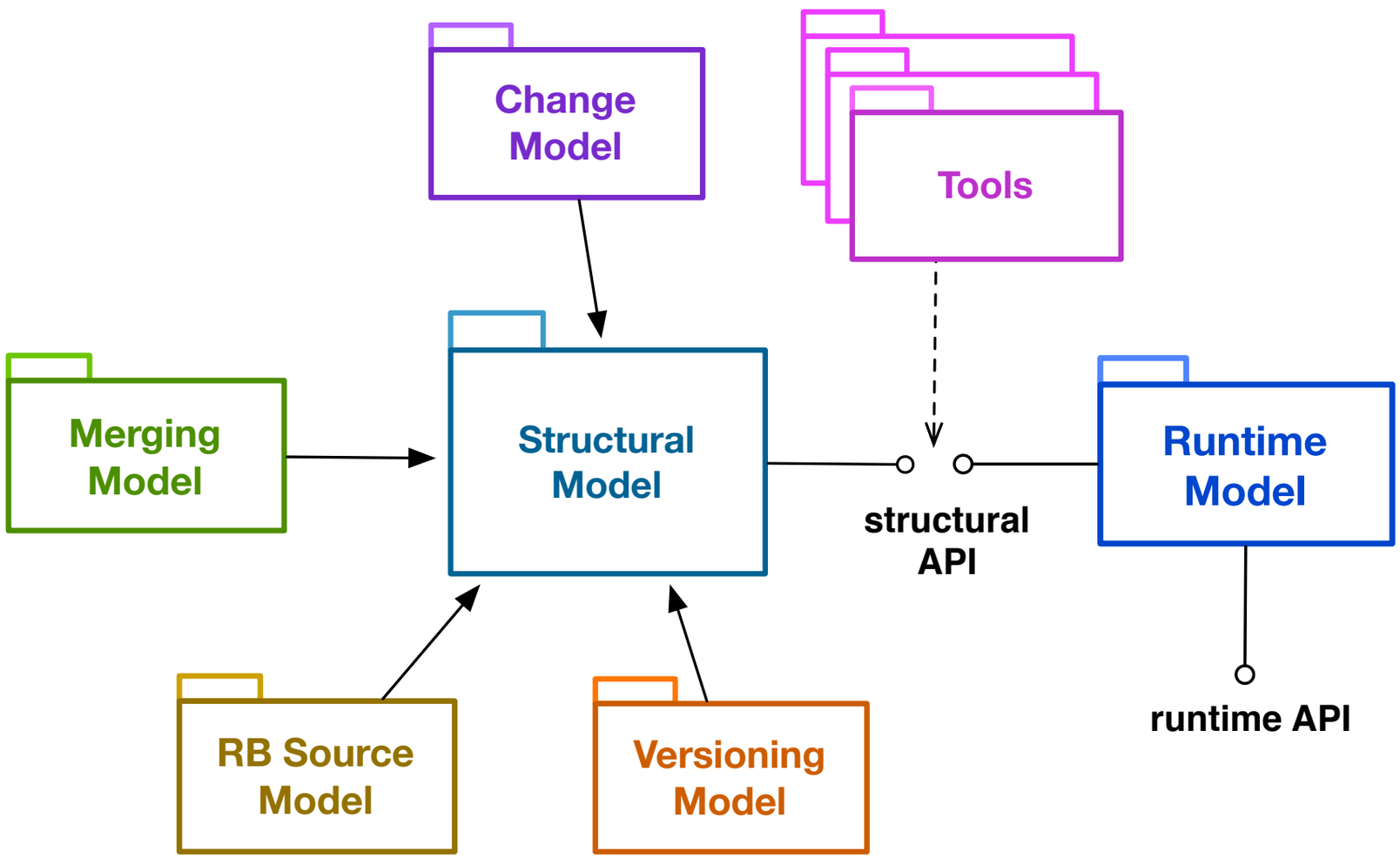
Ring Overview



Ring Overview

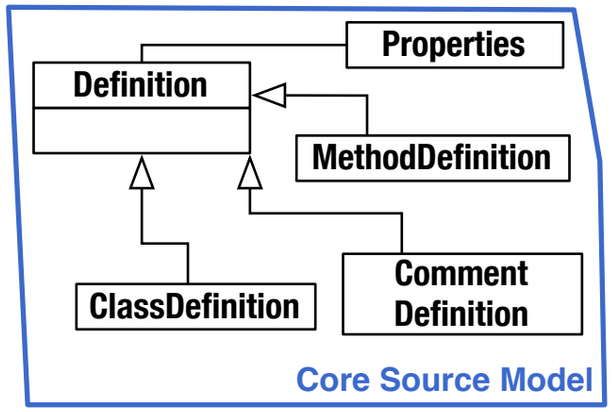


Ring Overview

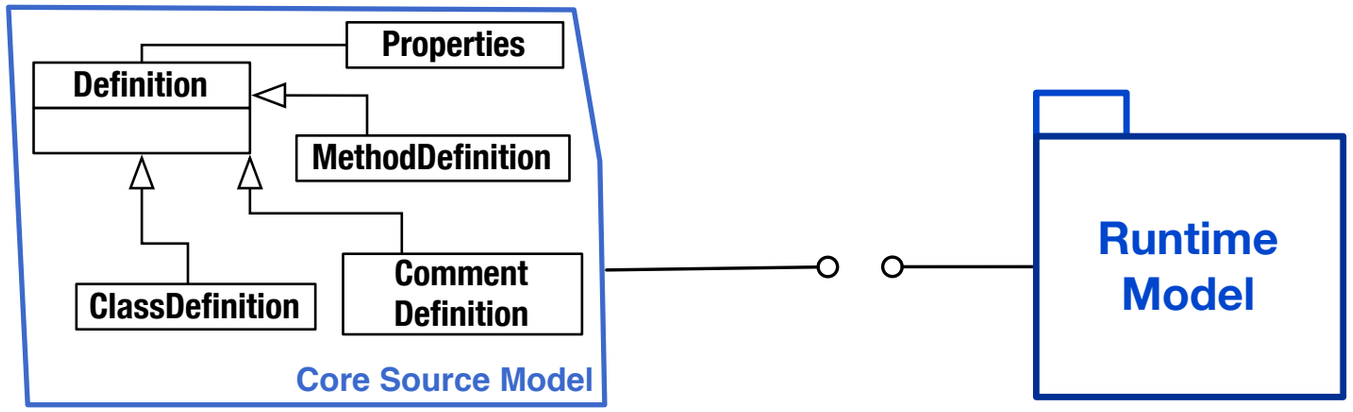


The Onion Structure of the Ring

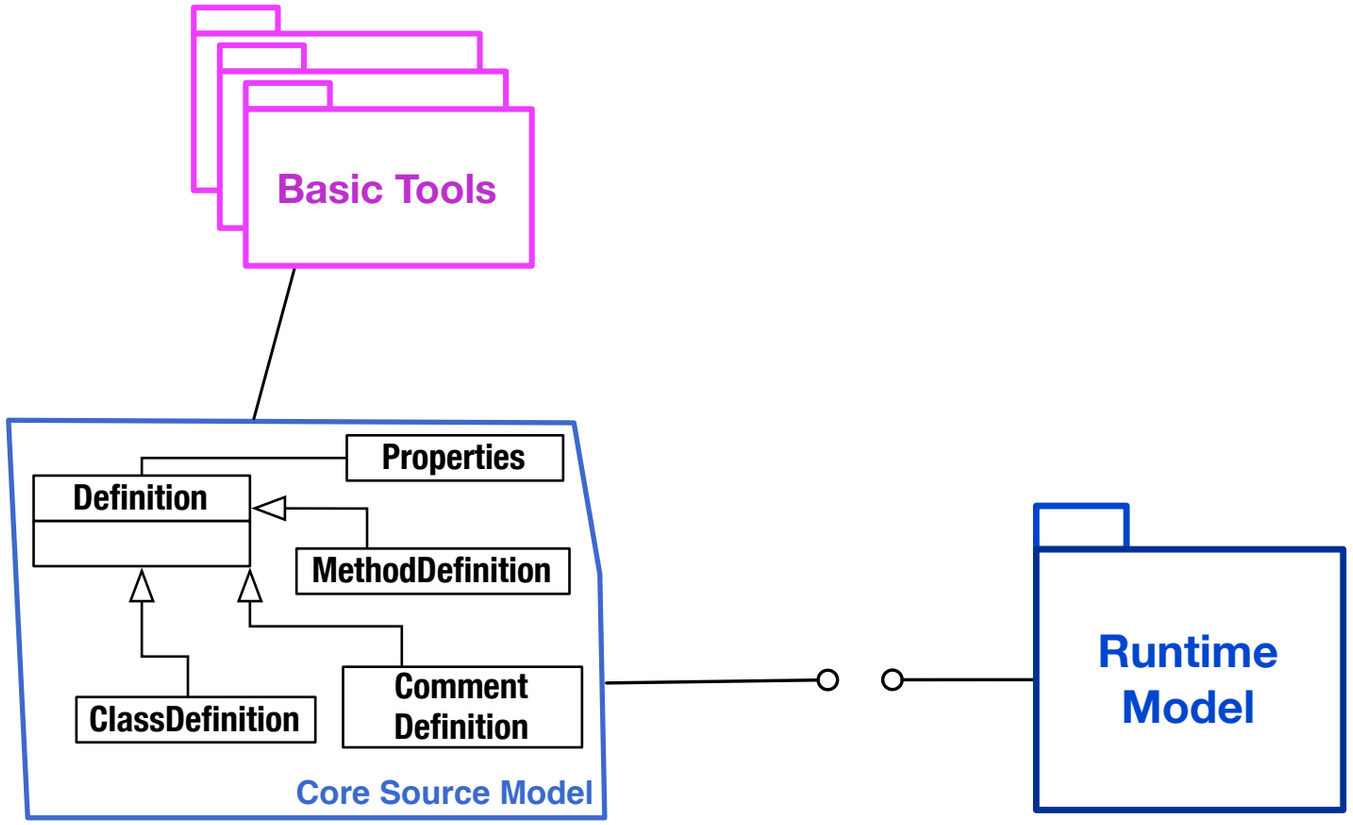
The Onion Structure of the Ring



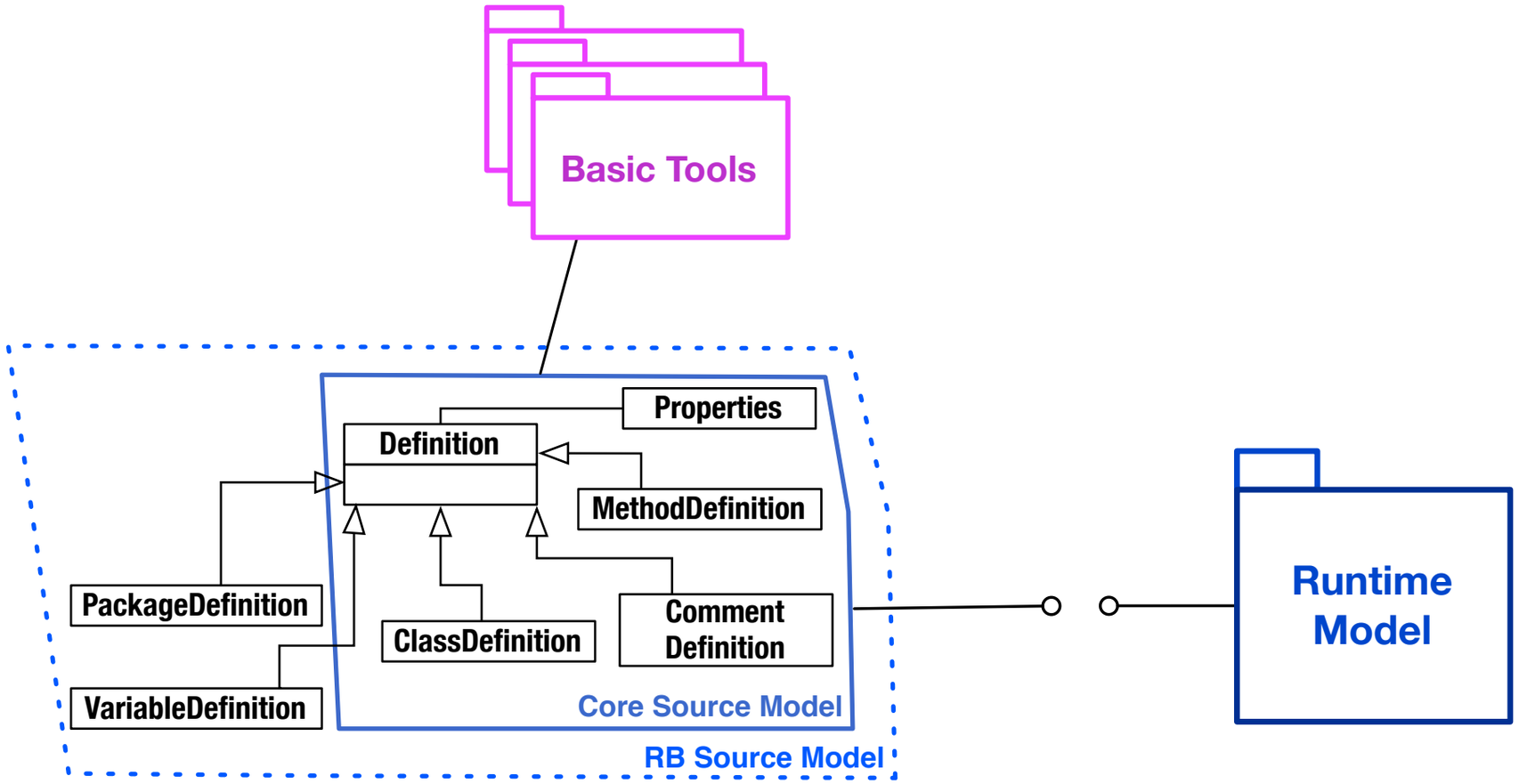
The Onion Structure of the Ring



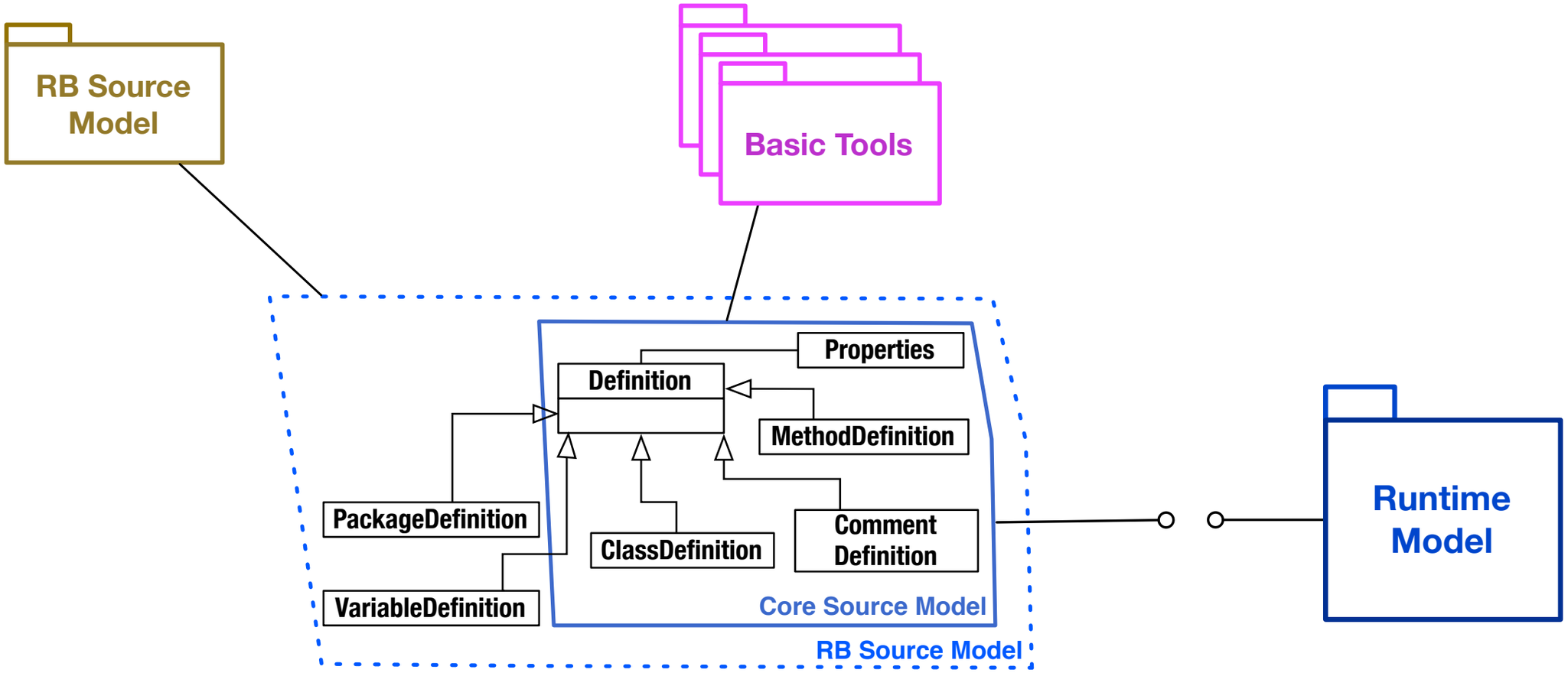
The Onion Structure of the Ring



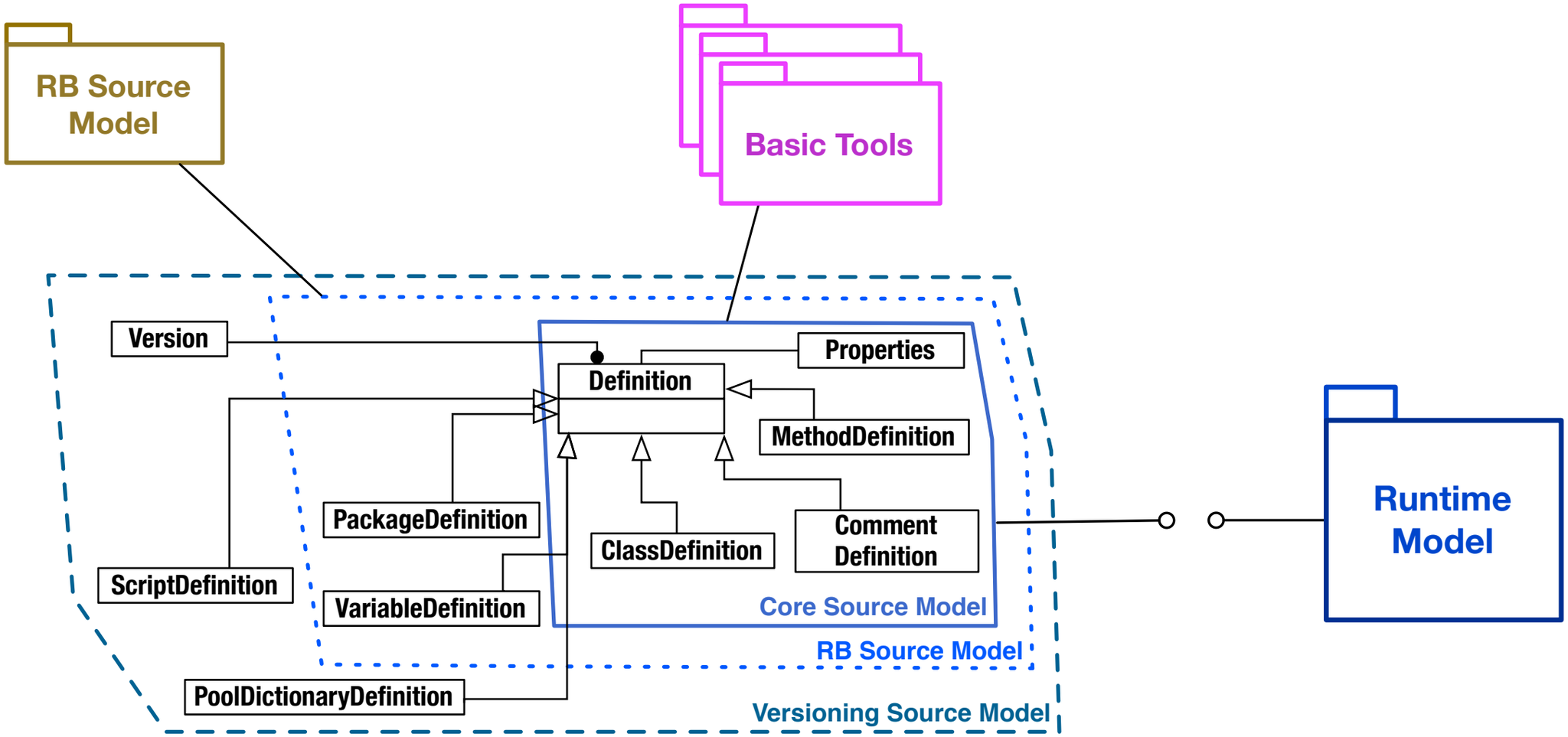
The Onion Structure of the Ring



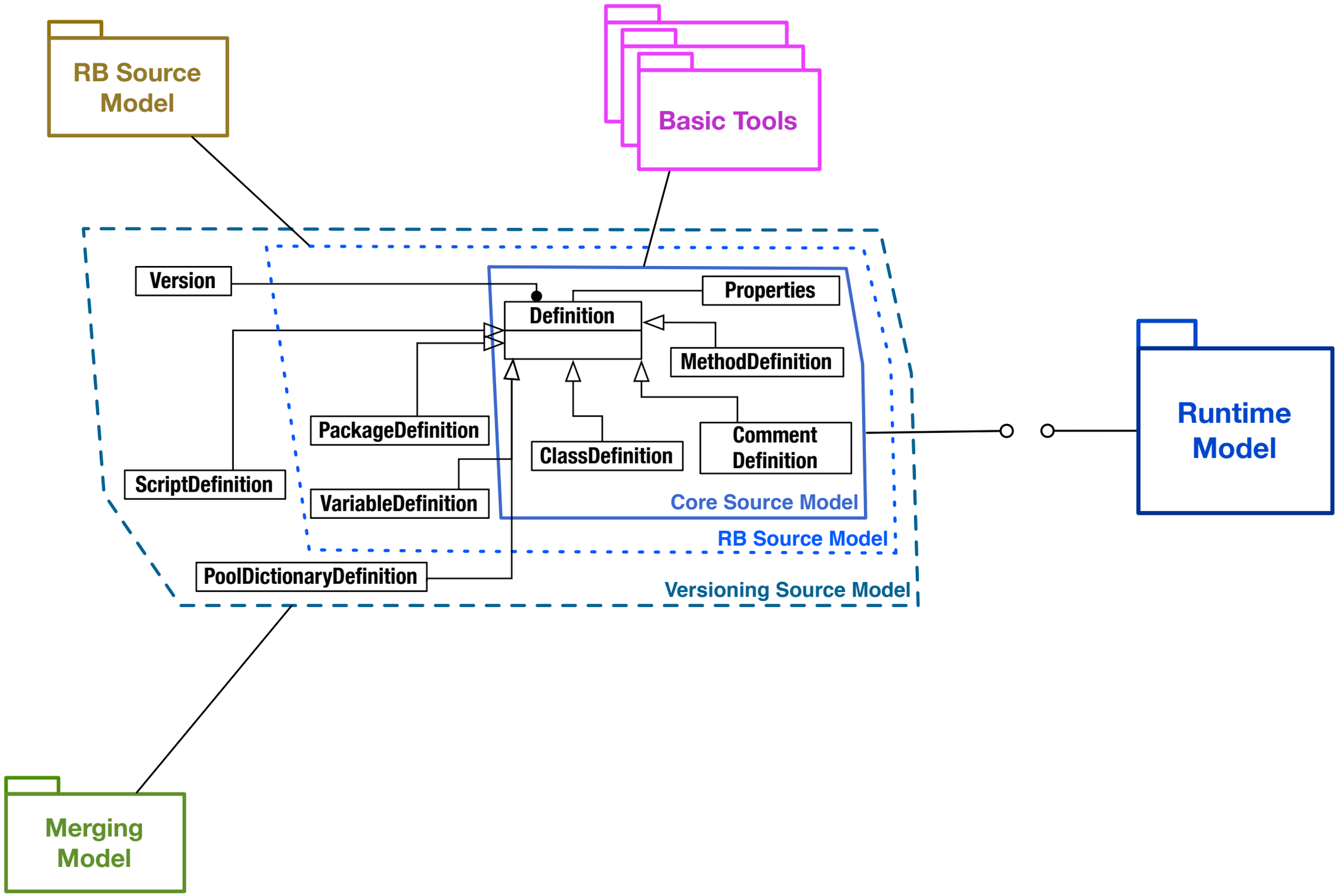
The Onion Structure of the Ring



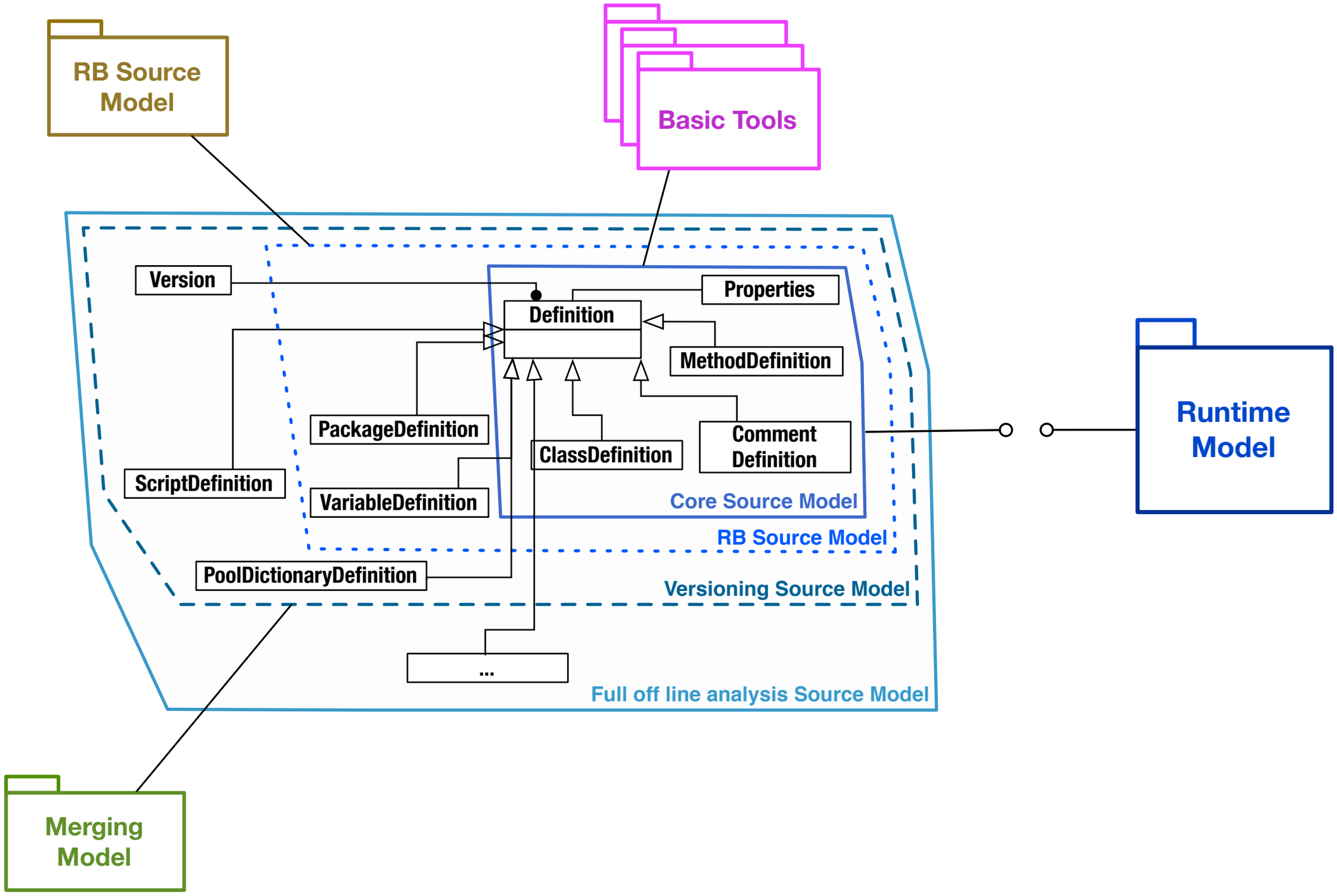
The Onion Structure of the Ring



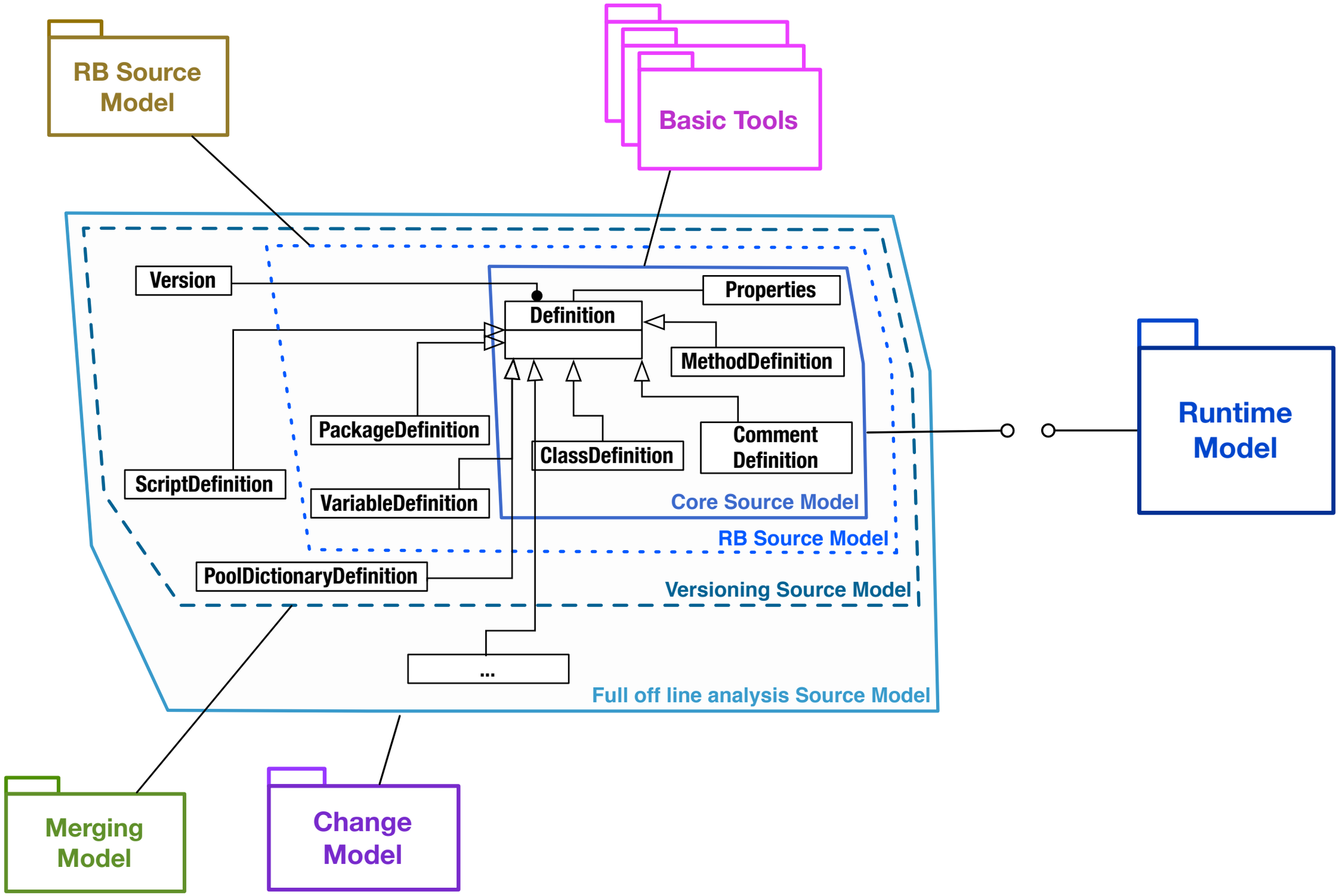
The Onion Structure of the Ring



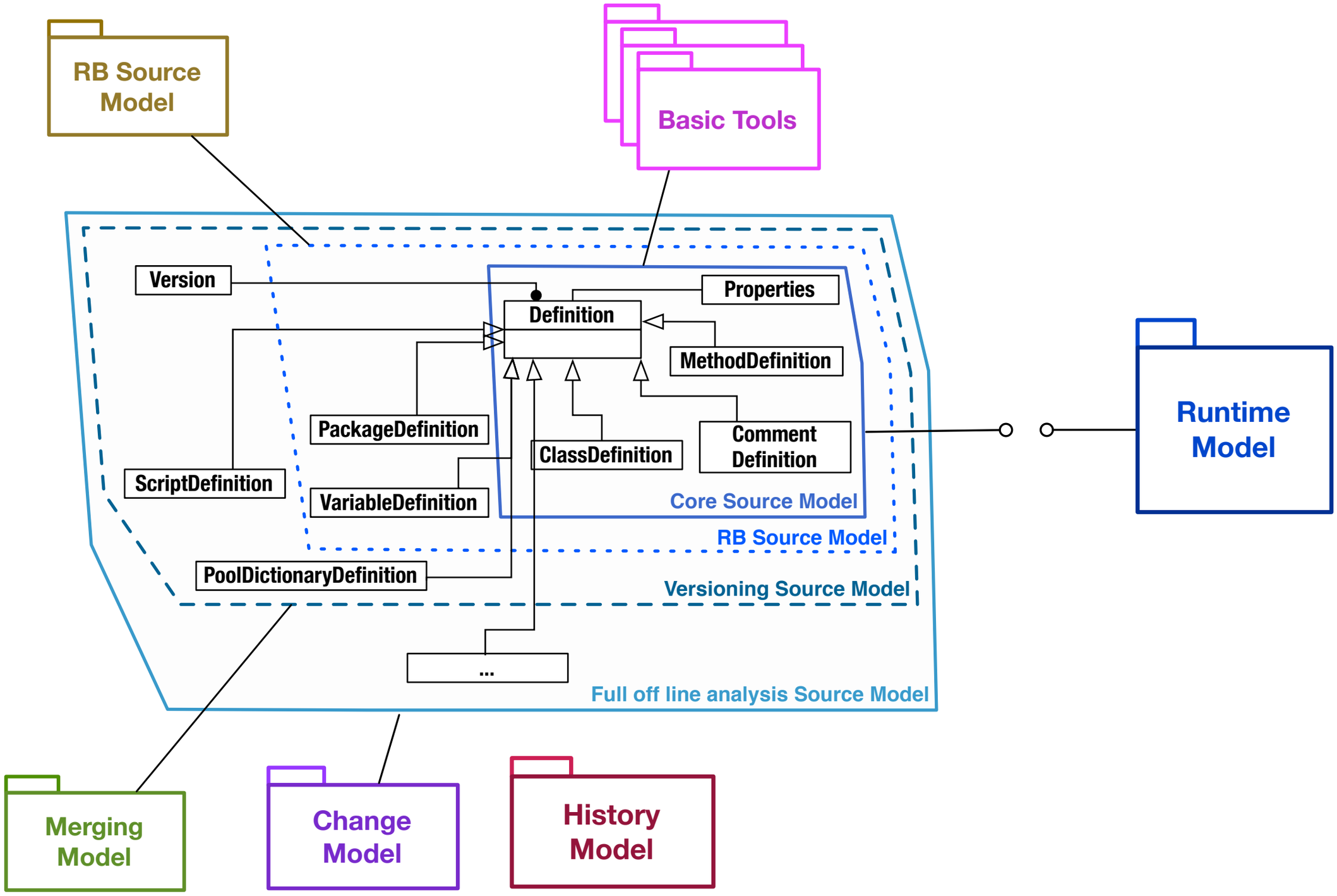
The Onion Structure of the Ring



The Onion Structure of the Ring



The Onion Structure of the Ring



Open Questions

Open Questions

- Expensive queries

Open Questions

- Expensive queries
- Version ids

Open Questions

- Expensive queries
- Version ids
- Annotations

Open Questions

- Expensive queries
- Version ids
- Annotations
- Meta-model vs. Database schema

Open Questions

- Expensive queries
- Version ids
- Annotations
- Meta-model vs. Database schema
- Core code model API

Open Questions

- Expensive queries
- Version ids
- Annotations
- Meta-model vs. Database schema
- Core code model API
- Meta-models extensibility

Open Questions

- Expensive queries
- Version ids
- Annotations
- Meta-model vs. Database schema
- Core code model API
- Meta-models extensibility
- Unifying models

Summary

Summary

- Goal:
  - perform linear and cross history analysis

Summary

- Goal:
  - perform linear and cross history analysis

- Problem:
  - unsuitable infrastructure for querying the history
  - several meta-models overlapping & non-polymorphic APIs

Summary

- Goal:
  - perform linear and cross history analysis

- Problem:
  - unsuitable infrastructure for querying the history
  - several meta-models overlapping & non-polymorphic APIs

- Proposal:
  - unifying and foundational model infrastructure, Ring

Summary

- Goal:
  - perform linear and cross history analysis

- Problem:
  - unsuitable infrastructure for querying the history
  - several meta-models overlapping & non-polymorphic APIs

- Proposal:
  - unifying and foundational model infrastructure, Ring

- Current work:
  - implementation of the Ring

Meta-models and Infrastructure for Smalltalk Omnipresent History

Verónica Uquillas-Gómez, vuquilla@vub.ac.be

Argentina - Nov 12th 2010

Meta-models and Infrastructure for Smalltalk Omnipresent History

Verónica Uquillas-Gómez, vuquilla@vub.ac.be

Argentina - Nov 12th 2010

Meta-models and Infrastructure for Smalltalk Omnipresent History

Verónica Uquillas-Gómez, vuquilla@vub.ac.be

Argentina - Nov 12th 2010