Meta-models and Infrastructure for Smalltalk Omnipresent

History

Verónica Uquillas-Gómez
Stéphane Ducasse
Theo D’Hondt

Andy Kellens, VUB
Source Code History Analysis

Monday 29 November 2010
Source Code History Analysis

- Linear history
Source Code History Analysis

- Linear history

Pharo 0.x
Source Code History Analysis

- Linear history

Pharo 0.x
Pharo 1.0
Source Code History Analysis

- Linear history
Source Code History Analysis

- Linear history

- Pharo 0.x
- Pharo 1.0
- Pharo 1.1
- Pharo 1.2

Monday 29 November 2010
Source Code History Analysis

- Linear history
- Cross history
Source Code History Analysis

- Linear history
- Cross history
Source Code History Analysis

- Linear history
- Cross history

[Diagram showing the history of Squeak with branches pointing to different versions such as Squeak 3.6, Squeak 3.8, Etoys 3.0, Etoys 4.0, Pharo 0.x, Pharo 1.0, Pharo 1.1, Pharo 1.2, and more.]
Source Code History Analysis

- Linear history
- Cross history

---

Monday 29 November 2010
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? |

---

**Monday 29 November 2010**
Querying the History of a System

<table>
<thead>
<tr>
<th>Analysis Type</th>
<th>Query Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-change analysis</td>
<td>what are the entities that changed together with entity Number in version 3?</td>
</tr>
<tr>
<td>Queries as in the past</td>
<td>what were the senders of the method #asString in Squeak 3.9?</td>
</tr>
</tbody>
</table>
## Querying the History of a System

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-change analysis</td>
<td>what are the entities that changed together with entity <em>Number</em> in version 3?</td>
</tr>
<tr>
<td>Queries as in the past</td>
<td>what were the senders of the method <em>#asString</em> in Squeak 3.9?</td>
</tr>
<tr>
<td>Queries as in the present</td>
<td>what are the messages sent by method <em>#printOn:</em> in version 2?</td>
</tr>
</tbody>
</table>

Monday 29 November 2010
## Querying the History of a System

<table>
<thead>
<tr>
<th>Co-change analysis</th>
<th>what are the entities that changed together with entity <em>Number</em> in version 3?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries as in the past</td>
<td>what were the senders of the method <em>asString</em> in Squeak 3.9?</td>
</tr>
<tr>
<td>Queries as in the present</td>
<td>what are the messages sent by method <em>printOn:</em> in version 2?</td>
</tr>
<tr>
<td>Bug spot</td>
<td>was the method <em>printString</em> regularly changed over the last 5 years?</td>
</tr>
</tbody>
</table>
### Querying the History of a System

<table>
<thead>
<tr>
<th>Co-change analysis</th>
<th>what are the entities that changed together with entity <code>Number</code> in version 3?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries as in the past</td>
<td>what were the senders of the method <code>asString</code> in Squeak 3.9?</td>
</tr>
<tr>
<td>Queries as in the present</td>
<td>what are the messages sent by method <code>printOn:</code> in version 2?</td>
</tr>
<tr>
<td>Bug spot</td>
<td>was the method <code>printString</code> regularly changed over the last 5 years?</td>
</tr>
<tr>
<td>Global analysis</td>
<td>what is the whole history of method <code>detect:ifNone:</code>?</td>
</tr>
</tbody>
</table>
# Querying the History of a System

<table>
<thead>
<tr>
<th>Analysis Type</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-change analysis</td>
<td>what are the entities that changed together with entity <em>Number</em> in version 3?</td>
</tr>
<tr>
<td>Queries as in the past</td>
<td>what were the senders of the method <em>#asString</em> in Squeak 3.9?</td>
</tr>
<tr>
<td>Queries as in the present</td>
<td>what are the messages sent by method <em>#printOn:</em> in version 2?</td>
</tr>
<tr>
<td>Bug spot</td>
<td>was the method <em>#printString</em> regularly changed over the last 5 years?</td>
</tr>
<tr>
<td>Global analysis</td>
<td>what is the whole history of method <em>#detect:ifNone:</em>?</td>
</tr>
<tr>
<td>Forks analysis</td>
<td>if the version of method <em>#isNil</em> changed in Squeak 3.9, should it be changed in Pharo?</td>
</tr>
</tbody>
</table>
## Querying the History of a System

<table>
<thead>
<tr>
<th>Co-change analysis</th>
<th>what are the entities that changed together with entity <em>Number</em> in version 3?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries as in the past</td>
<td>what were the senders of the method <em>#asString</em> in Squeak 3.9?</td>
</tr>
<tr>
<td>Queries as in the present</td>
<td>what are the messages sent by method <em>#printOn:</em> in version 2?</td>
</tr>
<tr>
<td>Bug spot</td>
<td>was the method <em>#printString</em> regularly changed over the last 5 years?</td>
</tr>
<tr>
<td>Global analysis</td>
<td>what is the whole history of method <em>#detect:ifNone:</em>?</td>
</tr>
<tr>
<td>Forks analysis</td>
<td>if the version of method <em>#isNil</em> changed in Squeak 3.9, should it be changed in Pharo?</td>
</tr>
<tr>
<td>Comparison profiler</td>
<td>what are the differences or similarities of running versions 4 and 5?</td>
</tr>
</tbody>
</table>
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
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

MC source code model

RB source code model

Tools source code model
Multiple Meta-Models

Smalltalk model

MC source code model

RB 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

How easy is ... interacting, maintaining, synchronizing?
Meta-Models Plague

- Multiple models
- API
Meta-Models Plague

- Multiple models ≠ API
Meta-Models Plague

✦ Multiple models $\equiv$ API

✦ conversion or transformation
Meta-Models Plague

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

- Multiple models \(\equiv\) API
  - conversion or transformation
  - duplication of data, tests
  - more maintenance tasks
Meta-Models Plague

- Multiple models
  - API conversion or transformation
  - Duplication of data, tests
  - More maintenance tasks

Mass duplication of efforts/tasks
Versioning Systems vs. Source Code Meta-Model
Versioning Systems vs. Source Code Meta-Model

SVN  | GIT  | MC

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

- SVN
- GIT
- MC

Versioning System

Source Code Meta-Model

- algorithm
- changes
- versions

Monday 29 November 2010
Versioning Systems vs. Source Code Meta-Model

SVN  GIT  MC

Versioning System

Source Code Meta-Model

* algorithm
* changes
* versions
Versioning Systems vs. Source Code Meta-Model

**Versioning System**

**Source Code Meta-Model**

- Algorithm
- Changes
- Versions
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
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 Phar
- 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 Phar
- Support history analysis
Ring Overview
Ring Overview
Ring Overview
Ring Overview
Ring Overview

- Change Model
- Structural Model
- Runtime Model
- RB Source Model
- Versioning Model

Connections:
- Structural Model to Change Model: structural API
- Structural Model to Runtime Model: structural API
- Runtime Model to other models: runtime API

Monday 29 November 2010
Ring Overview

Merging Model → Structural Model → Change Model

Structural Model → RB Source Model

Structural Model → Versioning Model

Structural Model → Runtime Model

structural API

runtime API

Monday 29 November 2010
Ring Overview

- Change Model
- Structural Model
- Tools
- Runtime Model
- Merging Model
- RB Source Model
- Versioning Model

structural API

runtime API
The Onion Structure of the Ring
The Onion Structure of the Ring

- Definition
- ClassDefinition
- MethodDefinition
- Comment Definition
- Properties

Core Source Model
The Onion Structure of the Ring
The Onion Structure of the Ring

Basic Tools

Properties
Definition
MethodDefinition
ClassDefinition
Comment Definition
Core Source Model

Runtime Model

Monday 29 November 2010
The Onion Structure of the Ring

Basic Tools

Core Source Model

PackageDefinition

ClassDefinition

MethodDefinition

Definition

Properties

Runtime Model

VariableDefinition

Comment Definition

RB Source Model

Monday 29 November 2010
The Onion Structure of the Ring
The Onion Structure of the Ring

- RB Source Model
- Basic Tools
- Core Source Model
- Versioning Source Model
- Runtime Model

- Definition
- Properties
- MethodDefinition
- Comment Definition
- ClassDefinition
- PackageDefinition
- VariableDefinition
- PoolDictionaryDefinition
- ScriptDefinition
- Version

Monday 29 November 2010
The Onion Structure of the Ring

**RB Source Model**

- Version
- ScriptDefinition
- PackageDefinition
- VariableDefinition
- PoolDictionaryDefinition

**Core Source Model**

- Definition
- ClassDefinition
- MethodDefinition
- Comment Definition
- Properties

**Versioning Source Model**

**Basic Tools**

**Runtime Model**

Merging Model

Monday 29 November 2010
The Onion Structure of the Ring
The Onion Structure of the Ring

RB Source Model

Basic Tools

Core Source Model

Versioning Source Model

Full off line analysis Source Model

Runtime Model

Merging Model

Change Model

Monday 29 November 2010
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

Monday 29 November 2010
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

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