Generic Instantiation and Tool Support

Thai Son Hoang

Institute of Information Security, ETH Zurich, Switzerland

Rodin Workshop 2013, Turku, Finland
11th June 2013

(joint work between ETH Zurich and Hitachi Ltd.)

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Tackling the complexity of systems modelling

Abrial and Hallerstede (2007)

> *... modeling a large and complex computer system results in a large and complex model. ... proofs will be more and more difficult to perform as models become inevitably larger and larger.*
>
> *... We present three techniques, refinement, decomposition, and instantiation, that we consider indispensable for modeling large and complex systems.*

Refinement, Decomposition, and Instantiation

Current Status

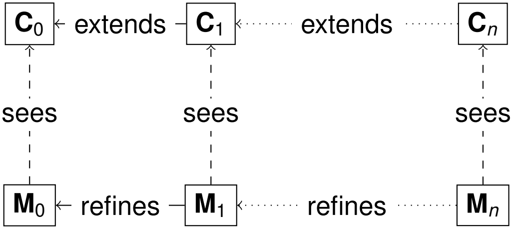
- Refinement: An integral part of Event-B
- Decomposition: Shared-event / shared-variable decomposition

Refinement, Decomposition, and Instantiation

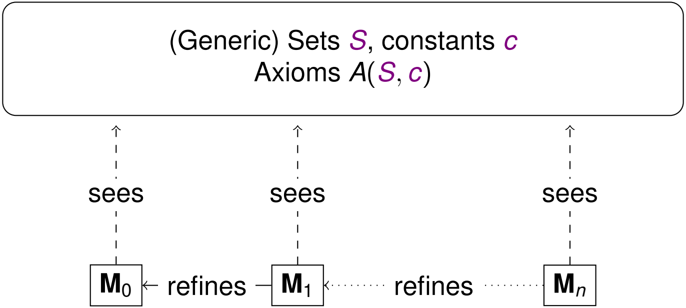
Current Status

- Refinement: An integral part of Event-B
- Decomposition: Shared-event / shared-variable decomposition
- Generic instantiation:
  - R. Silva and M. Butler: Supporting Reuse of Event-B Developments through Generic Instantiation. (ICFEM 2009)
  - Ulyana Tikhonova at. al. (Rodin Workshop 2013) (this morning)

Generic Instantiation



Generic Instantiation



- The development is parameterised by S and c

Generic Instantiation

(Generic) Sets S, constants c
Axioms $A(S, c)$

↑

Instantiation: $S = E(T)$, $c = F(T, d)$

(Specific) Sets T, constants d
Axioms $B(T, d)$

- The development is parameterised by S and c
- Reuse model: Instantiating to $S = E(T)$ and $c = F(T, d)$

$$B(T, d) \Rightarrow A(E(T), F(T, d))$$

Instantiating Sets and Constants. An Example

Generic context

sets : *MESSAGE*

constants : *maxsize*

axioms :

finite(*MESSAGE*)

$maxsize \in \mathbb{N}1$

Instantiating Sets and Constants. An Example

Generic context

sets : *MESSAGE*

constants : *maxsize*

axioms :
$\text{finite}(MESSAGE)$
$maxsize \in \mathbb{N}1$

Specific context

sets : *ID*, *INFO*

axioms :
$\text{finite}(ID)$
$\text{finite}(INFO)$

Instantiating Sets and Constants. An Example

Generic context

sets : *MESSAGE*

constants : *maxsize*

axioms :

$\mathrm{finite}(MESSAGE)$

$maxsize \in \mathbb{N}1$

Specific context

sets : *ID*, *INFO*

axioms :

$\mathrm{finite}(ID)$

$\mathrm{finite}(INFO)$

- Instantiation: $MESSAGE = ID \times INFO$, $maxsize = 3$
- To be proved:

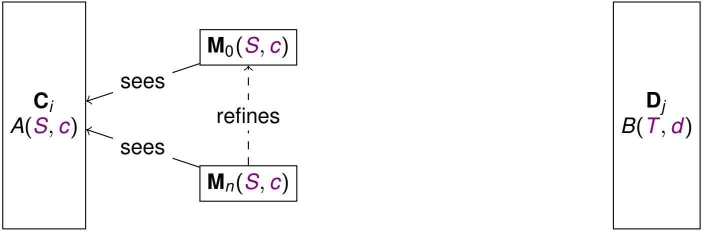
$$\mathrm{finite}(ID) \land \mathrm{finite}(INFO) \;\Rightarrow\; \mathrm{finite}(ID \times INFO) \land 3 \in \mathbb{N}1$$

A Generic Instantiation Tool

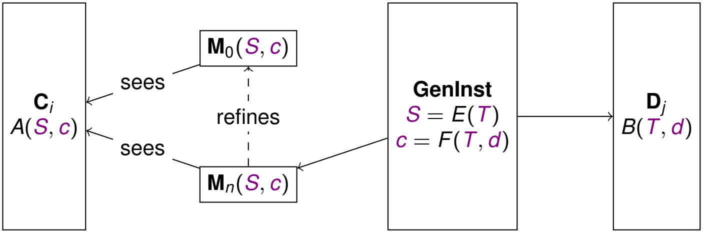
HITACHI
Inspire the Next

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

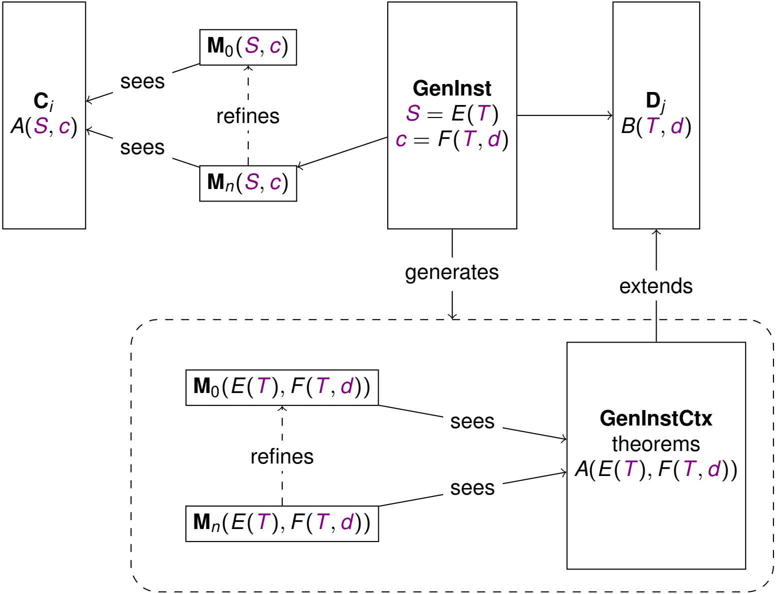
A Generic Instantiation Tool



A Generic Instantiation Tool



A Generic Instantiation Tool



Demo

The Usefulness of Generic Instantiation

The similarity between refinement and generic instantiation

The Usefulness of Refinement

The ability to perform abstraction with state variables.

The Usefulness of Generic Instantiation

The similarity between refinement and generic instantiation

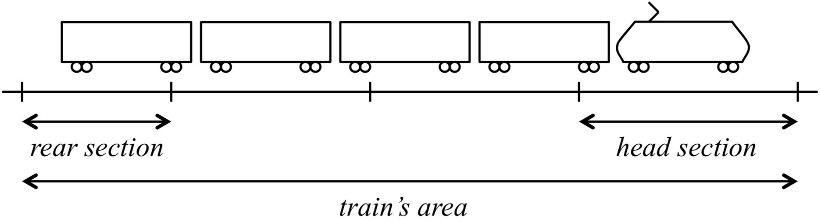
The Usefulness of Refinement

The ability to perform abstraction with state variables.

The Usefulness of Generic Instantiation

The ability to perform abstraction with sets and constants

Train Control Example (1)



Typical modelling style using variables

$$
\begin{aligned}
&\mathit{trains} \subseteq \mathit{TRAIN\_ID}\\
&\mathit{head} \in \mathit{trains} \rightarrow \mathit{SECTION}\\
&\mathit{rear} \in \mathit{trains} \rightarrow \mathit{SECTION}\\
&\mathit{area} \in \mathit{trains} \rightarrow \mathbb{P}(\mathit{SECTION})\\
&\mathit{connection} \in \mathit{trains} \rightarrow (\mathit{SECTION} \mathrel{+\mkern-15mu\rightarrow} \mathit{SECTION})
\end{aligned}
$$

Train Control Example (2)

- Trains can be represented by some abstract data type (e.g., sequence of sections *SEQUENCE*).
- Operations with sequences:
  - Extend head: $\mathit{extend} \in \mathit{SEQUENCE} \times \mathit{SECTION} \rightarrow\!\!\!\!\!+ \mathit{SEQUENCE}$
  - Remove rear: $\mathit{front} \in \mathit{SEQUENCE} \rightarrow\!\!\!\!\!+ \mathit{SEQUENCE}$
  - Disjointness: $s_1 \mapsto s_2 \in \mathit{disjoint}$
  - Sub-sequence: $s_1 \mapsto s_2 \in \mathit{subset}$
- Properties of sequences

$$\forall s_1, s_2, s_3 \cdot$$
$$s_1 \mapsto s_2 \in \mathit{subset} \land s_2 \mapsto s_3 \in \mathit{disjoint} \;\Rightarrow\; s_1 \mapsto s_3 \in \mathit{disjoint}$$

$$\forall s_1, s_2 \cdot s_1 \mapsto s_2 \in \mathit{disjoint} \Rightarrow s_2 \mapsto s_1 \in \mathit{disjoint}$$

Summary

- Plug-in on sourceforge:
  `http://sourceforge.net/projects/gen-inst/`
- Instantiation enables abstraction with models' parameters.
- Context instantiation as a part of the development?