Refinement Propagation
Towards Automated Construction of Visual Specifications

Irina Rychkova and Alain Wegmann,
School of Computer and Communication Science,
Lausanne, Switzerland
Outline

1. Context
   - Enterprise Architecture
   - Visual Modeling
   - Problem: Design decisions vs. Adjustments

2. Knowledge Base
   - Refinement

3. Research Result
   - Refinement of Visual Specifications
   - Refinement Propagation
   - Example
SEAM in Enterprise Architecture

- SEAM is a method for enterprise design.
- SEAM addresses business issues as well as IT issues.
- Idea: the enterprise, its environment and its construction are represented as a hierarchy of systems.
- One diagram type
Modeling Process

1. Creation and transformation of visual specifications is driven by modeler's design decisions.

2. After a design decision has been made, the modeler needs to adjust the specification to maintain its correctness.
Practical Problem

- The number of adjustments might make the design process tedious for large specifications

- We are interested in techniques that will
  - reduce the modeler's obligation
  - control specification correctness by automated adjustment.
1 Context
- Enterprise Architecture
- Visual Modeling
- Problem: Design decisions vs. Adjustments

2 Knowledge Base
- Refinement

3 Research Result
- Refinement of Visual Specifications
- Refinement Propagation
- Example
Theories

- **Stepwise refinement**
  - A well-known paradigm that states that a program specification can be built step-by-step from the abstract level down to executable code.
  - Each transformation of the spec is considered as a refinement \[\text{[Wirth, N. (1971)], [Dijkstra, E. W. (1971)]}\]

- **Refinement calculus**
  - Having defined the logic of “accepted” and “forbidden” transformations or “refinement laws”, refinement step can be automatically calculated \[\text{[Back, R.-J. and von Wright, J. (1998)]}\]
1. Context
   - Enterprise Architecture
   - Visual Modeling
   - Problem: Design decisions vs. Adjustments

2. Knowledge Base
   - Refinement

3. Research Result
   - Refinement of Visual Specifications
   - Refinement Propagation
   - Example
Refinement of Visual Specifications

- Visual specification development can be recognized as a sequence of refinement steps starting from an abstract level.

- A correct refinement step = a design decision + all necessary adjustments to keep the refined (NEW) model consistent and non-contradictory with the initial (OLD) model.
Definition of Refinement for SEAM Visual Specifications

**DEF:** Specification $S$ (old) is correctly refined by specification $S'$ (new) $S \subseteq S'$, if $S'$ satisfies any requirement satisfied by $S$. 
Context
- Enterprise Architecture
- Visual Modeling
- Problem: Design decisions vs. Adjustments

Knowledge Base
- Refinement

Research Result
- Refinement of Visual Specifications
- Refinement Propagation
- Example
Refinement Propagation

- Properties, events, actions, and activities are related within specification.

- A basis refinement of an arbitrary element $X$ of the specification may cause a conflict between certain elements, such that

$$X \subseteq X' \land S(X) \nsubseteq S(X')$$
To resolve this conflict, refinement $X_1'$ of some other element $X_1$ of the spec is usually required.

Within a final number of steps, refinement propagation either results in correct refinement of the specification:

$$S: \ (X \subseteq X') \Rightarrow (X_1 \subseteq X_1') \Rightarrow (X_n \subseteq X_n') \Rightarrow S(X,X_1,..,X_n) \subseteq S(X',X_1',..,X'_n)$$

or indicates such a refinement impossible.
Algorithm

1. Initiating refinement
2. Conflict detection: what must be adjusted?
3. Adjustment
4. Repeat 2-3 until No adjustment can be done…
The Rules of Refinement Propagation

- Resolving conflicts between:
  - Properties
  - Properties and Events
  - Events and Actions
  - Events and the System State Space
  - Pre-, Post-conditions, and Updates
  - an Activity and its Component Actions

- Incorrectness
1. **Context**
   - Enterprise Architecture
   - Visual Modeling
   - Problem: Design decisions vs. Adjustments

2. **Knowledge Base**
   - Refinement

3. **Research Result**
   - Refinement of Visual Specifications
   - Refinement Propagation
   - Example
Step 1

Initiating modification:
Name ↔ {FName, LName}

Conflict:
The property Name is lost!

Adjustment:
To express “old” Name using a “new” data structure..
Rule 1

- No Data can be lost!
- Property P, eliminated from the “old” model must be reproducible in the “new” (refined) model

$S_R \subseteq S' = S \subseteq S'; R$

Where R is a refinement relation, that connects ‘old’ and ‘new’ state spaces
Rule 1

Client_old = {n:Name};
Client_new = {fn:FName, ln:LName};

R: (Σ_{Client'} → {true, false} ) →
(Σ_{Client} → {true, false }) ⇔
(Σ_{FName} × Σ_{LName}) ↦ Σ_{Name}

1. n := substr(fn + ln, 30)
2. n := ln
3. ...
Step 2

Conflict:
Name is used by input event

Adjustment:
E (n:Name) ⇔ E1(fn:FName; ln: LName),

Considering R: n = ln
E1("John", "Smith") ⇔ E("Smith")
(Rule 2)
Step 3

Conflict:
Event parameter is used by the action

Adjustment:
Pre (n:Name) ⇔
Pre1(fn:FName; ln: LName), ...
(Rules 4-5)
Example 2

Design decision:
Example 2

```
Example 2

SpecA2

m|c[1..c[m]

Client

FName
{String[15]}

? Pre/U/Post ?

LName
{String[20]}

? Pre/U/Post ?

? Pre/U/Post ?

AddClient

Get FName

Get L Name

Save Data

E2.1 [new_fn: FName]

E2.2 [new Ln: LName]

Design decision:

E1

{Pre1, U1, Post1}

Irina Rychkova, Alain Wegmann
ICEIS 2007
```
Summary: Towards Automated Construction of visual specifications

Design decision analysis:
- Adjustments can be automatically calculated and applied to the rest of the model
- Additional information needed (the decision is ambiguous)
- Adjustments not needed (the decision is consistent)
- Adjustments are impossible! (the decision has to be revised)