OCL Visualization
A Reality Check

Edward Willink

Willink Transformations Ltd

Eclipse Foundation

MMT Component co-Lead
OCL Project Lead
QVTd Project Lead
QVTo Committer

OMG (Model Driven Solutions)

OCL 2.3, 2.4, 2.5 RTF Chair
QVT 1.2, 1.3, 1.4 RTF Chair

OCL 2019 @ MODELS 2019

16th September 2019

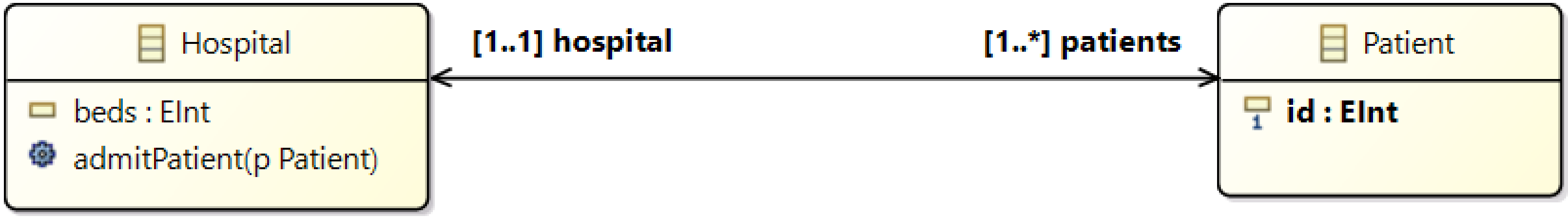
Overview

- Background
- Traditional AST visualization
- vOCL 2018 problems
- Fixed vOCL
- Eclipse QVTs visualization
- Constraint Diagram
- Visual OCL
- Summary / Conclusion

Background

- Text can be good
- Pictures can be good
- Text can be better
- Pictures can be better
- Text dominant for expressions (except SDL)
- UML graphical – why no graphical OCL ?
- OCL 2018 => vOCL paper worth discussion
  - reviewer's / audience criticisms not rebutted

Running Example



- Example constraint

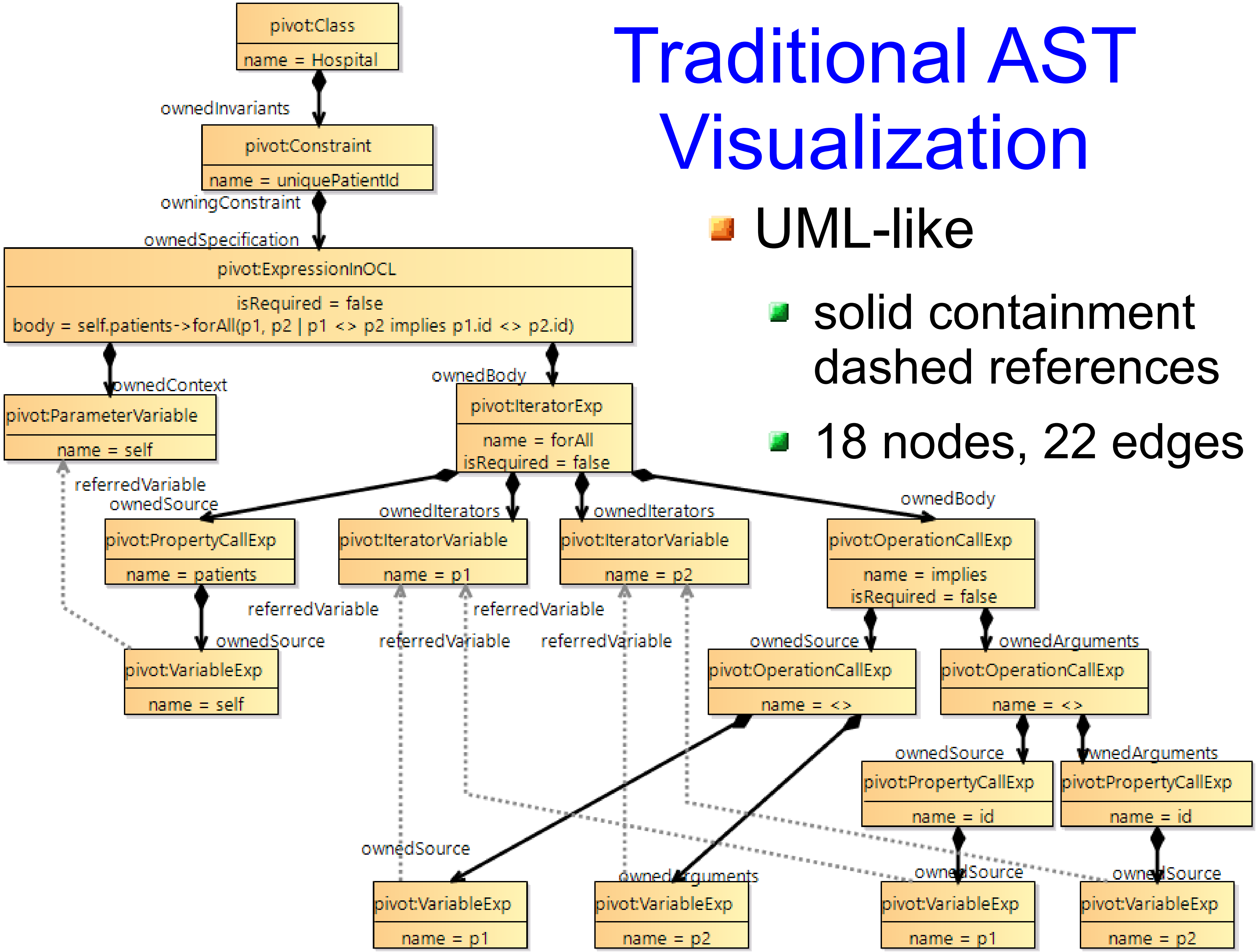
```
context Hospital
inv uniquePatientId:
   self.patients->forAll(p1, p2 |
      p1 <> p2 implies p1.id <> p2.id)
```

- Equivalent to

```
context Hospital
inv uniquePatientId:
   self.patients->isUnique(id)
```

Traditional AST Visualization

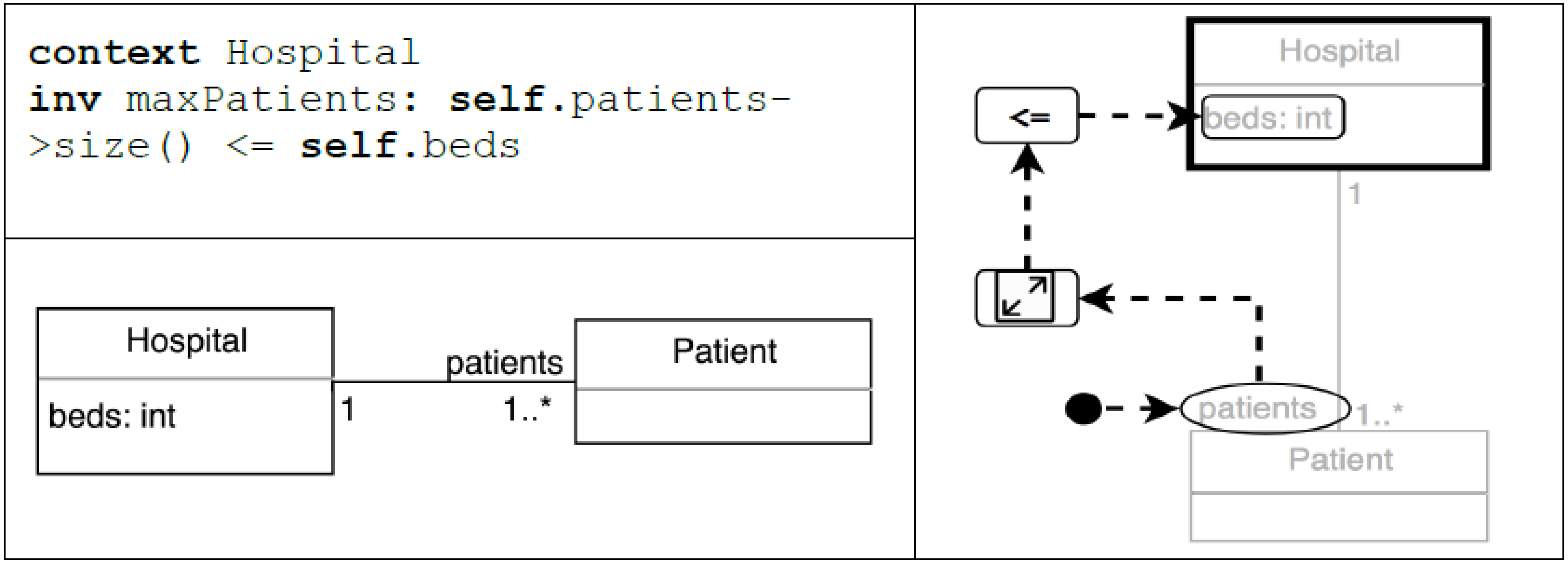
- UML-like
  - solid containment dashed references
  - 18 nodes, 22 edges



vOCL

- Traditional AST is instance-based
  - instances of OCL metamodel
    - requires knowledge of OCL internals
  - references by name to user metamodel
- vOCL is class-based
  - re-uses user metamodel to define variables
  - mnemonic icons for common OCL operations
  - vOCL constraint is a doodle on the class diagram
    - new doodling semantics

vOCL example from 2018 paper

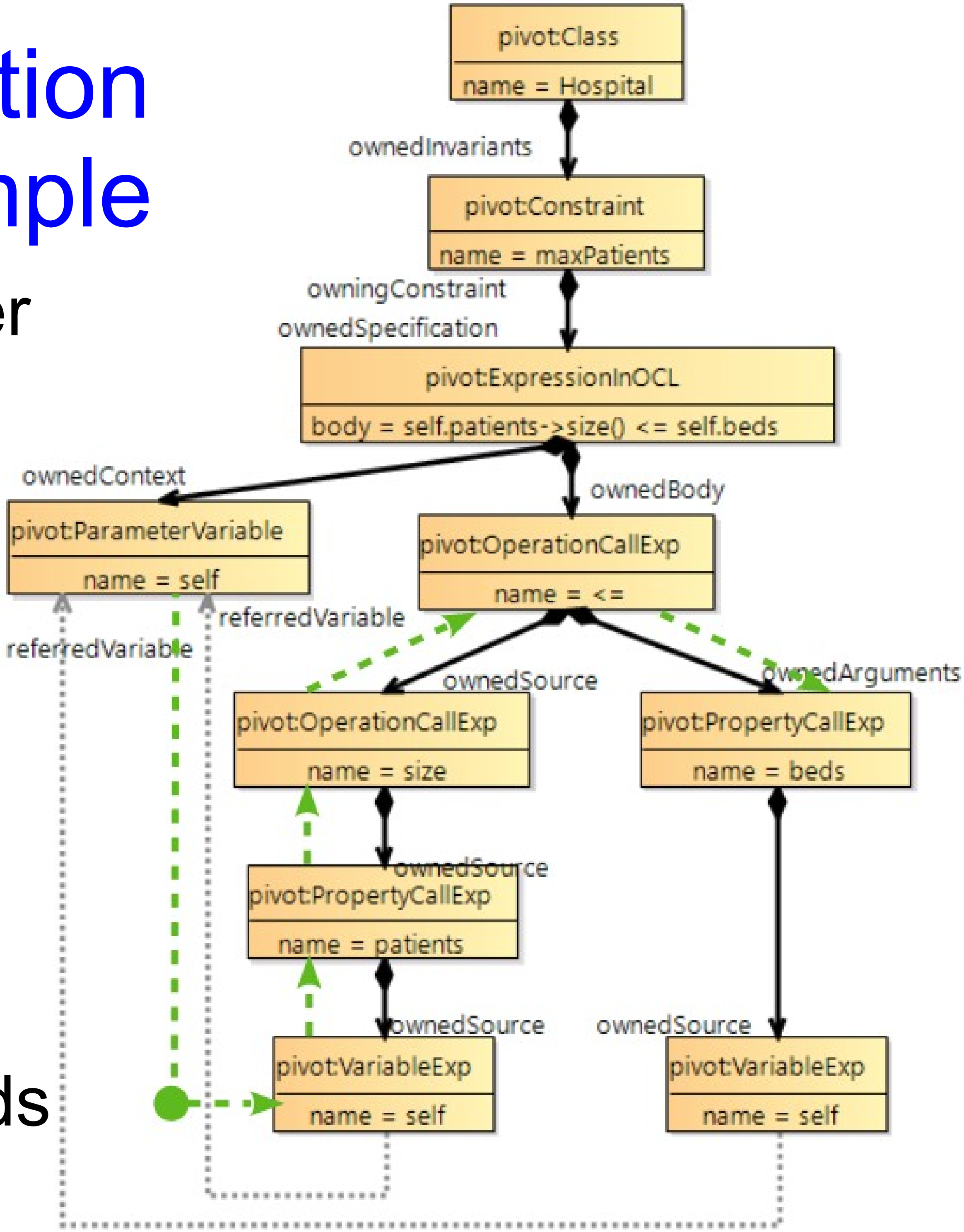


```
context Hospital
inv maxPatients: self.patients-
>size() <= self.beds
```

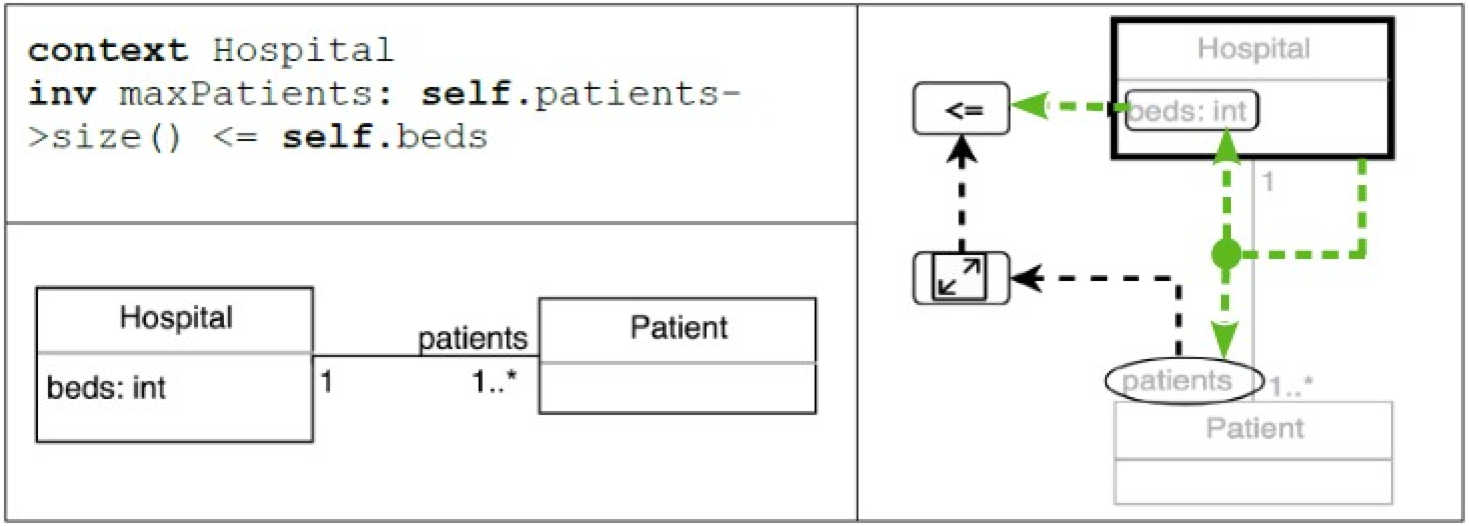
- blob is the **`self`** instance of the not-grayed class
- ellipses use Class-typed reference properties
- rounded rectangles use DataType properties
- additional icons for OCL operations
- arrows denote 'reading order'
  - vague semantics

AST visualization of vOCL example

- green reading order markup
- blob - self - ok
- ....patients - ok
- ....size() - ok
- ....<= - lhs ok
- <= ... - rhs inverted pragmatic self.beds backwards



Fixed vOCL example

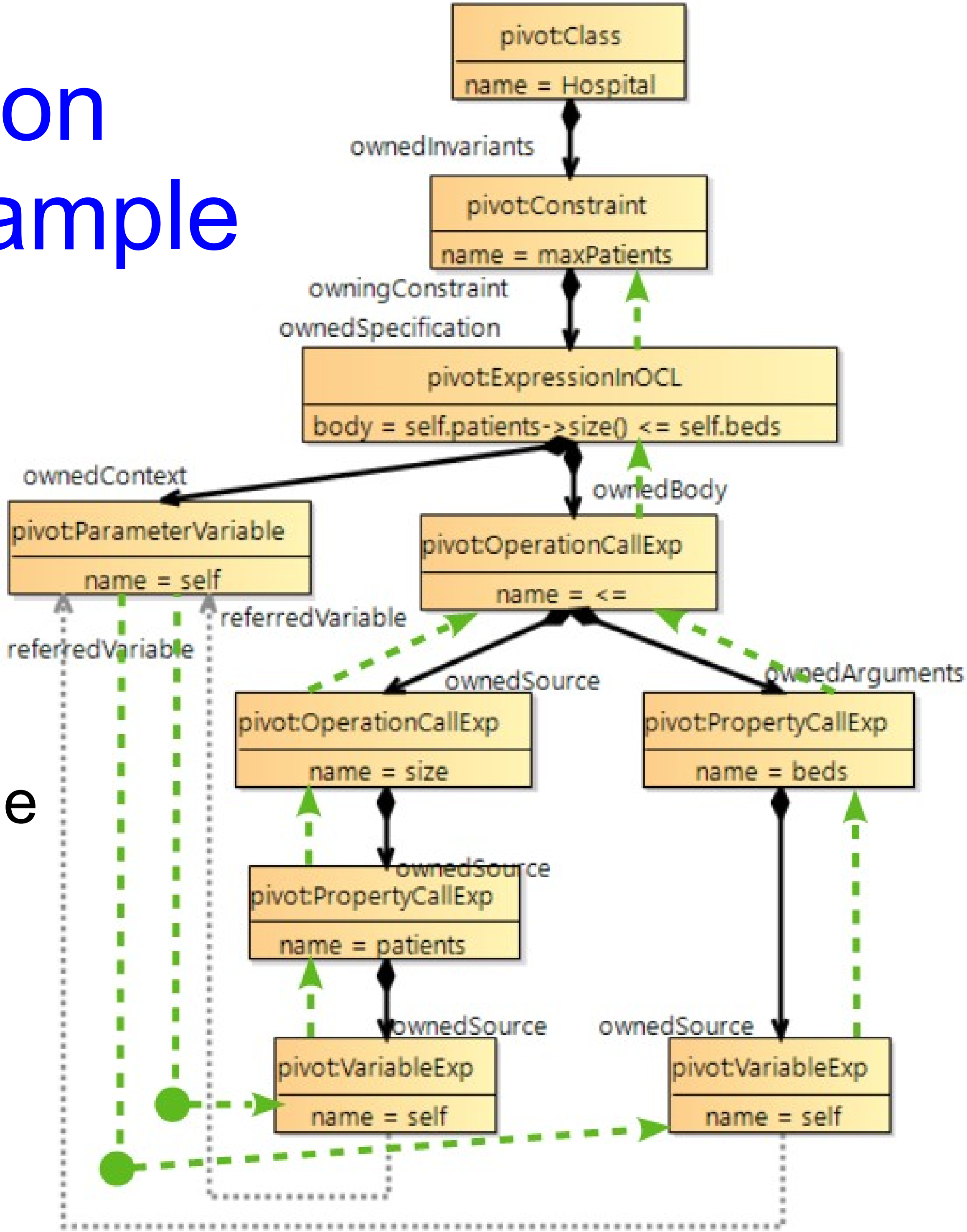


```
context Hospital
inv maxPatients: self.patients-
>size() <= self.beds
```

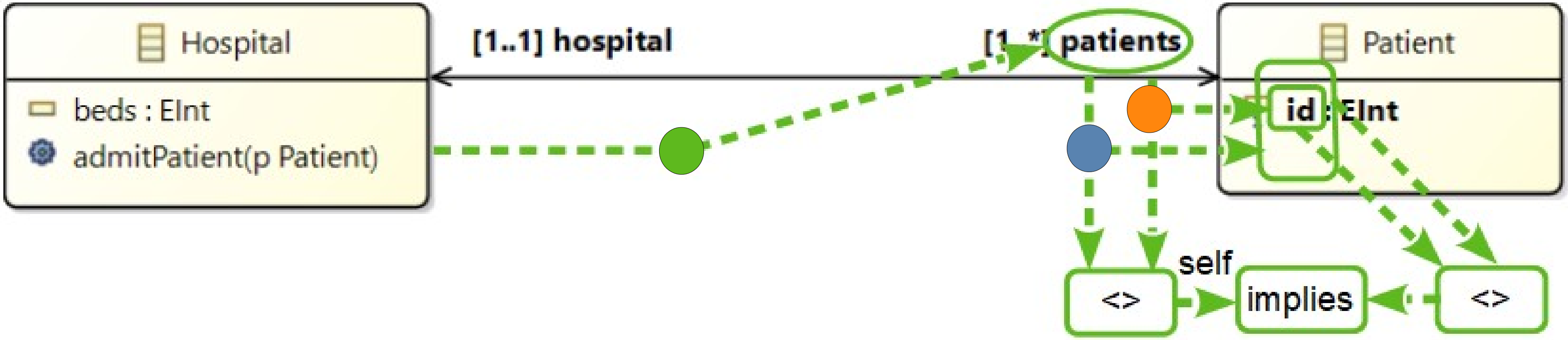
- Multiple paths for multiple instances/evaluations
- Blob now starts two paths
  - black path as in original paper to read the LHS self.patients.size() <=
  - new green path to read the RHS self.beds >=
  - both paths read towards the binary <= operation

AST visualization of fixed vOCL example

- two blobs
  two bottom-up paths
- 'reading order' is now AST evaluation
  - blob defines 'self' value
  - edge passes value
  - node computes value



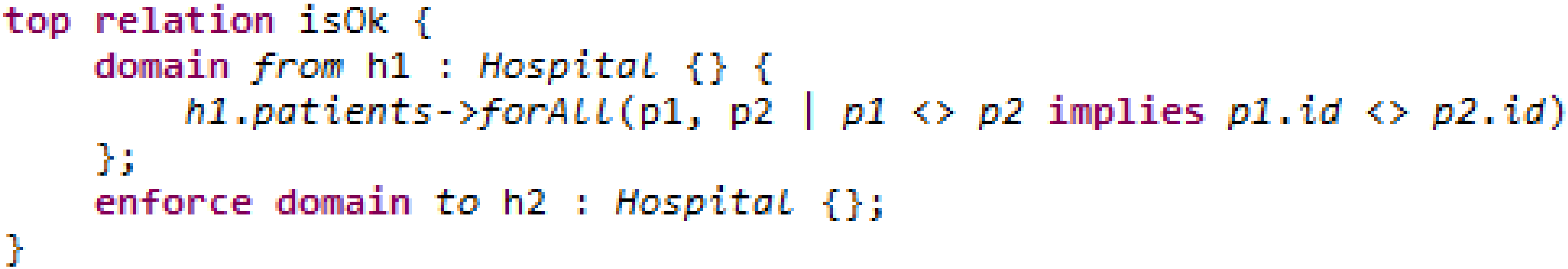
Running example using Fixed vOCL



- Blobs identify three variables starting five paths
  - self.patients
  - self.patients.id <> implies
  - self.patients.id <> implies
  - self.patients <> implies
  - self.patients <> implies
- Nested rounded rectangles for multiple uses
- 9 nodes + 12 edges (AST: 18 nodes, 22 edges)

Eclipse QVTs OCL visualization

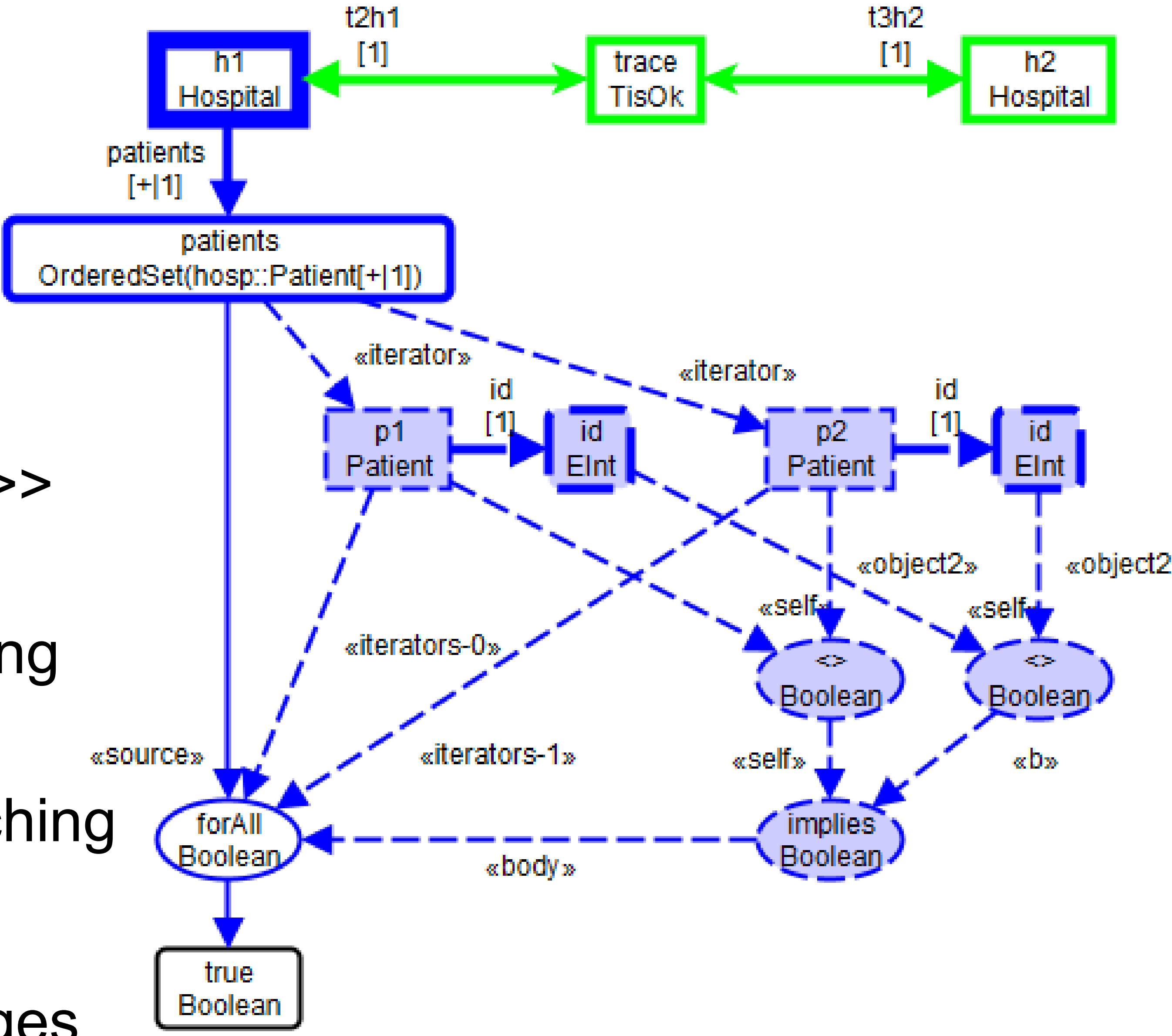
- AST visualization
  - Object Diagram: instances of OCL metamodel
- vOCL
  - Class Diagram: doodle on user metamodel
- Eclipse QVTs patterns
  - Object Diagram: instances of user metamodel
  - auto-generated debug aid for matching schedule
  - example embedded in QVTr



```
top relation isOk {
    domain from h1 : Hospital {} {
        h1.patients->forAll(p1, p2 | p1 <> p2 implies p1.id <> p2.id)
    };
    enforce domain to h2 : Hospital {};
}
```

Running Example using Eclipse QVTs

- Rectangles
  => class instances
- Rounded rectangles
  => DataType values
- Ellipses
  => OCL operations
  edges have <<roles>>
- Solid edges
  => unit value matching
- Dashed edges
  => [0..*] values matching
- 11 nodes and 15 edges

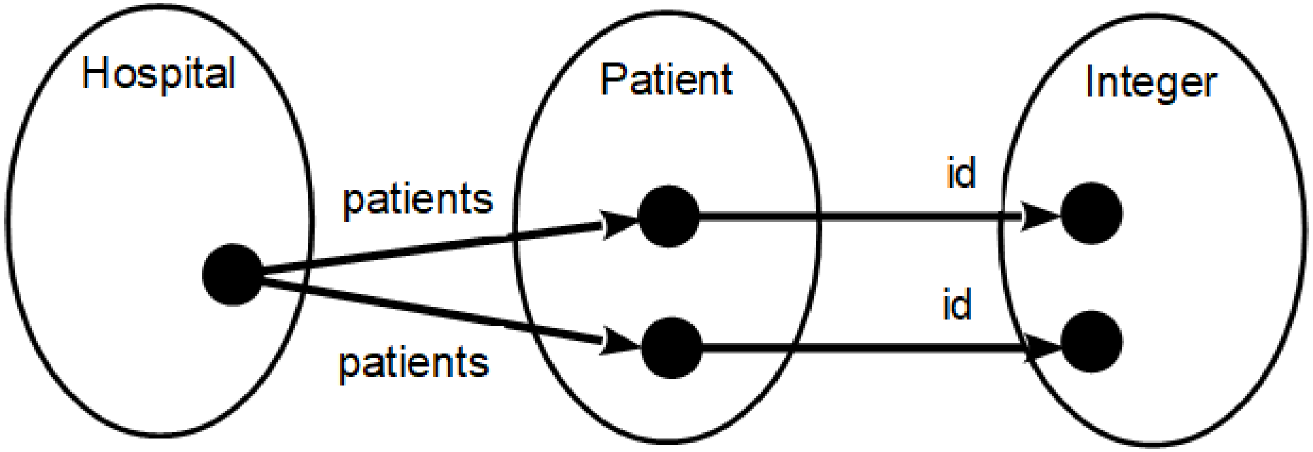


Constraint Diagrams

- Stuart Kent, OOPSLA 1997

  Constraint Diagrams: Visualizing Invariants in Object-Oriented Models.
- Visual Set membership
  - blobs for members
  - shapes for types
  - blobs within shapes for membership
  - edges for relationships
  - ... Venn Diagrams
- Not OCL
  - related concepts ... better perspective

Running example - Constraint Diagram



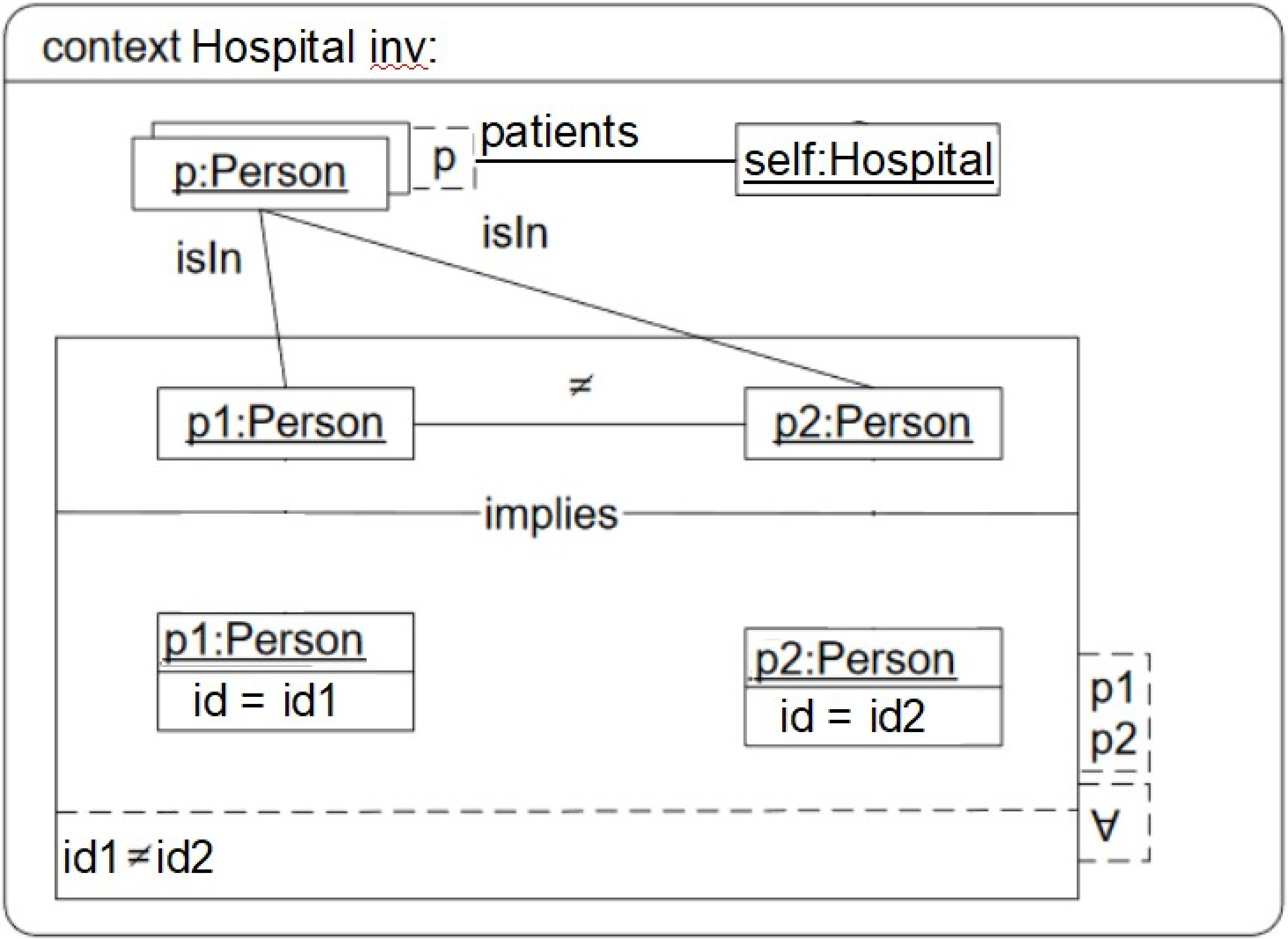
- Ellipses - user metamodel types
- Blobs - distinct instances typed by background
- Edges - metamodel relationships
- Powerful set concepts align well to OCL
  - some OCL concepts hard to represent
- 8 nodes, 4 edges

Visual OCL

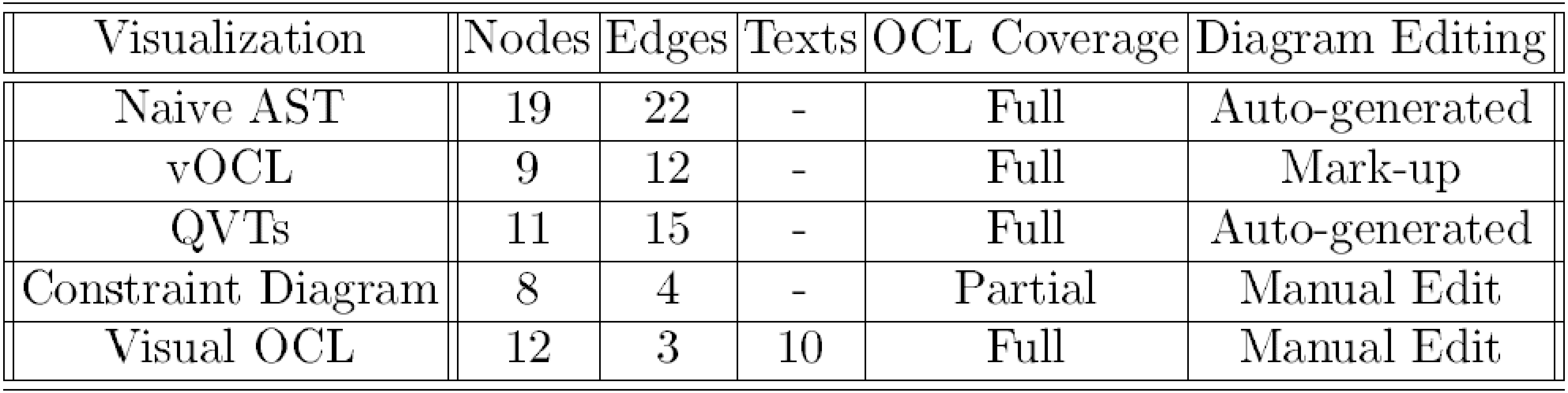
- Christiane Kiesner, Gabriele Taentzer, Jessica Winkelmann TU Berlin Technical Report 2002/23
  Visual OCL: A Visualization of the Object Constraint Language
- A new 'UML' diagram
  - re-uses statechart idioms
  - hierachy of decorated compartments

Running Example - Visual OCL

- ? Draw as:
  12 nodes / compartments
  3 edges
  10 text tokens



Summary / Comparison



Visualization	Nodes	Edges	Texts	OCL Coverage	Diagram Editing
Naive AST	19	22	-	Full	Auto-generated
vOCL	9	12	-	Full	Mark-up
QVTs	11	15	-	Full	Auto-generated
Constraint Diagram	8	4	-	Partial	Manual Edit
Visual OCL	12	3	10	Full	Manual Edit

- Naive AST - verbose, uses OCL metamodel
- vOCL - user-friendly, may become cluttered
- QVTs - patterns, variables avoid clutter
- Constraint Diagram - can be very good
  - beware: running example hits a 'sweet spot'
- Visual OCL - adequate

Summary / Reality

- Text can be better, Pictures can be better
- Text dominant for expressions
- Complex expressions need the 'best' exposition
  - patterns

- Text tooling unavoidable - ASCII compatibility
- Visual tooling specialized - UMLDI is a 'future'
- AST / QVTs visualization - a helpful debug aid
- Other Visual OCL editors unlikely to happen

Future Work ?

- Eclipse QVTs - adequate, complete
  - node => metamodel instance / operation
  - edge => metamodel relationship / operation role
- Constraint Diagram - compact, partial
  - node => metamodel instance
  - edge => metamodel relationship
  - background => set membership / typing
- ?? Merge the two ??
- ?? Questions ??