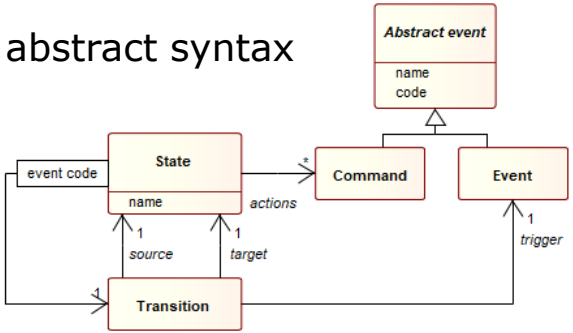
What Goes Wrong with Language Definitions *and* How to Improve the Situation

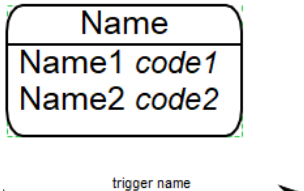
Juha-Pekka Tolvanen
jpt@metacase.com

Language definition

abstract syntax



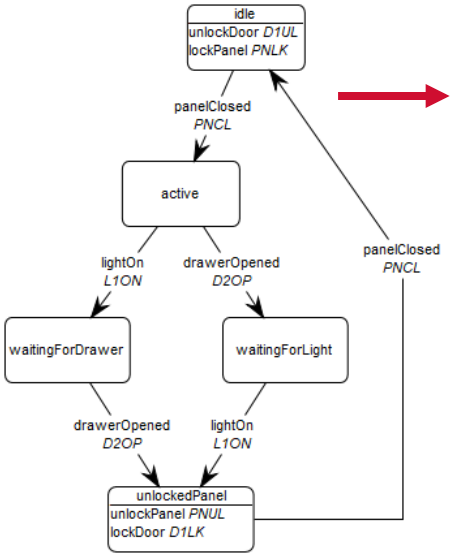
concrete syntax



semantics

```
foreach .State; {
   do :Command {
      id;l 'State.addAction(' :Name 'Cmd);'
   }
   do ~Source>Transition {
      id;l 'State.addTransition('
      do :Trigger {:Name} ', '
      do ~Target.() {id}
      'State);' newline
   }
```

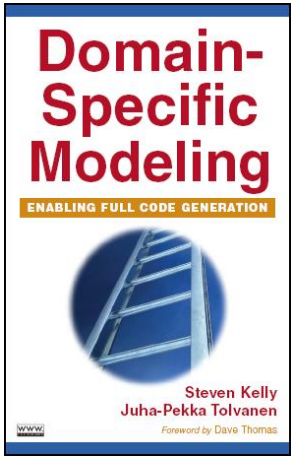
Modeling




```
Event doorClosed = new Event("doorClose
Event drawerOpened = new Event("drawe
Event lightOn = new Event("lightOn", "L1
Event panelClosed = new Event("panelClo

Command unlockDoorCmd = new Comma
Command lockPanelCmd = new Commar
Command unlockPanelCmd = new Comm
Command lockDoorCmd = new Command

State activeState = new State("active");
State idleState = new State("idle");
State unlockedPanelState = new State("u
State waitingForDrawerState = new State
State waitingForLightState = new State("

activeState.addTransition(drawerOpened,
activeState.addTransition(lightOn, waiting

idleState.addAction(unlockDoorCmd);
idleState.addAction(lockPanelCmd);
idleState.addTransition(doorClosed, active

unlockedPanelState.addAction(unlockPa
unlockedPanelState.addAction(lockDoorC
unlockedPanelState.addTransition(panelC
```

About me: Juha-Pekka Tolvanen

- Works for MetaCase
  - Provider of modeling and code generation tool MetaEdit+
- Acts as a consultant for creating DSLs
  - 100+ DSL solutions
- Co-author of a book on Domain-Specific Modeling, IEEE-Wiley
- PhD in computer science, adjunct professor
- Enjoys sailing and skiing



What goes wrong?

1. Language definitions fragmented, inconsistent, partial
2. Metamodel express something other than a language
3. Exchange does not work completely
4. Certificates sold, no actual learning
5. Language definitions are not tested
6. Human aspects are not recognized
7. Tools leave room for improvements
8. …

Due to NDA, public examples to follow…

Language definition…

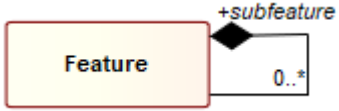
- …fragmented into different formats, hard to define
  - UML standard(s) contain hundreds of errors between metamodel and related OCL constraints [1, 2]
  - Situation not improved over the years/versions
- …internally inconsistent
  - EAST-ADL 2.1 error model hierarchy expects that each level has behavior – but it's only mandatory for leaves
- …partial
  - ArchiMate 3.0 followed with error corrections like 3.0.1

1. Bauerdick et al., Detecting OCL Traps in the UML 2.0 Superstructure: Experience Report. Procs of Unified Modeling Language - Modeling Languages and Applications, Springer, 2004
2. Wilke, C., Demuth, B., UML is still inconsistent! How to improve OCL Constraints in the UML 2.3 Superstructure, Procs of OCL and Textual Modelling workshop, 2011

Metamodels express something other than a language

Examples:

- AUTOSAR/EAST-ADL/etc: modeler is expected to follow a naming policy for the sake of XML exchange
  - Start with alphabet, certain length, no special characters
- Feature model owns subfeatures
  - Deleting a feature removes all subfeatures



- Language forces a mandatory value for an element because it is marked mandatory in XSD schema
- Usability issues, like when selecting among alternatives:
  - what is default value?
  - in which order they are provided, etc.

Exchange does not work completely

- XMI
  - VP of OMG said it would work by the end of 2005…
- ArchiMate exchange format 2023:
  - Allows sharing illegal models
  - ArchiMate supports language extensions, yet the exchange format does not cover them

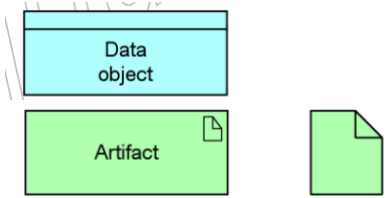
- **Note:** Exchange between tools is important, but not really an issue if tooling is not closed
  - Today metamodel-based tools (also known as Language Workbenches) by nature are never closed: you can always export your data to any format!

Certificates sold but their coverage…

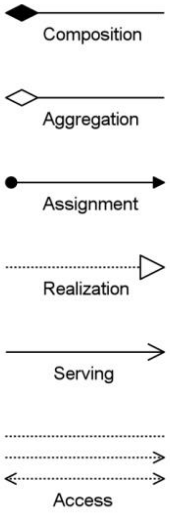
- Certificates for modelers do not cover (fully) the language
- Tool certificates do not check details
  - E.g. ArchiMate certification by Open Group expects that relationship types are available in the tool but **not** that they are applied legally

Language definitions are not tested
e.g. ArchiMate spec states

- Legend:



A.3 Relationships



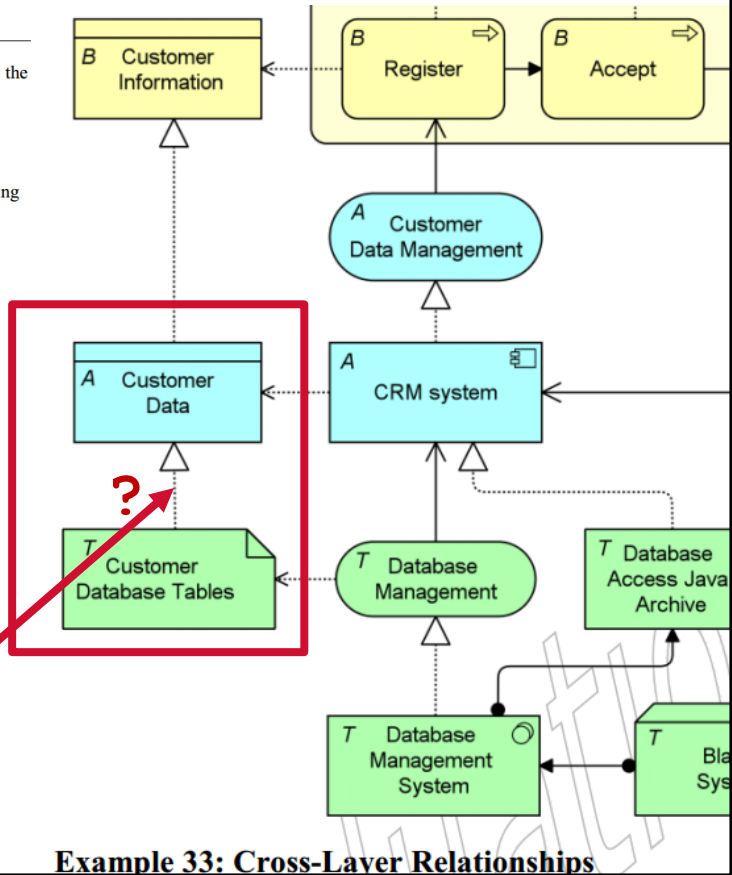
B Relationship Tables

This appendix details the normative requirements for relationships between elements of the ArchiMate modeling language.

The letters in the tables have the following meaning:

(a)ccess (c)omposition (f)low a(g)gregation ass(i)gnment
(i(n)fluence ass(o)ciation (r)ealization (s)pecialization (t)riggering ser(v)ing

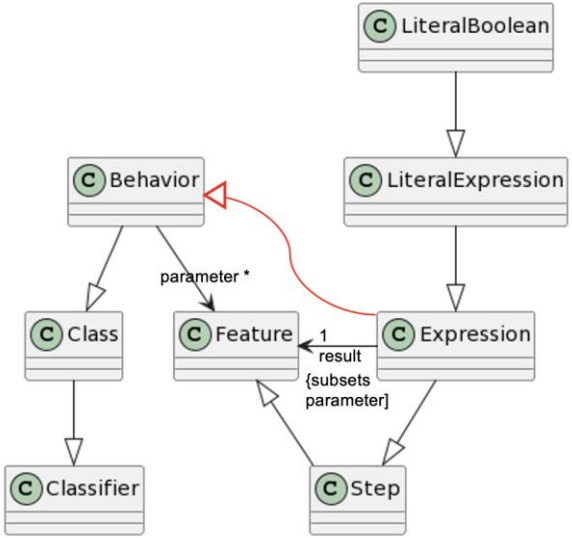
↓ From / → To	Application Collaboration	Application Component	Data Object	Application Event
Application Collaboration	cfgorstv	cfgorstv	ao	fiortv
Application Component	cfgorstv	cfgorstv	ao	fiortv
Data Object	o	o	cgos	o
Application Event	fot	fot	ao	cfgost
Application Function	fotv	fotv	ao	fotv
Application Interaction	fotv	fotv	ao	fotv
Application Interface	fotv	fotv	ao	fotv
Application Process	fotv	fotv	ao	fotv
Application Service	fotv	fotv	ao	fotv
Artifact	fortv	fortv	ao	fortv
Technology Collaboration	fortv	fortv	ao	fortv
Communication Network	fortv	fortv	ao	fortv



Example 33: Cross-Layer Relationships

Still repeated today*

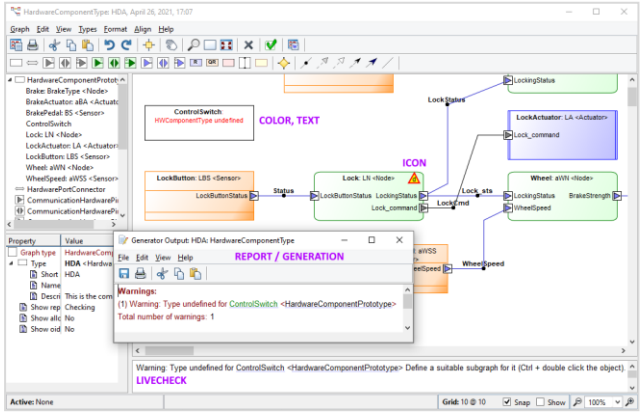
- SysML v2 2023



* https://www.opencaesar.io/blog/2023/08/27/When-Literal-Boolean-Is-Unsatisfiable.html

Human aspects are not recognized (some work, but much to do)

- Cognitive dimensions?
- Physics of notation guidelines by Moody?
- Modularity, 7+2 rules etc.
- Models show entered data only, like pen & paper style
  - Errors and warnings?
  - Guidance?
  - Simulation?
  - Debugging?
  - Views/skins for different usage/stakeholders?

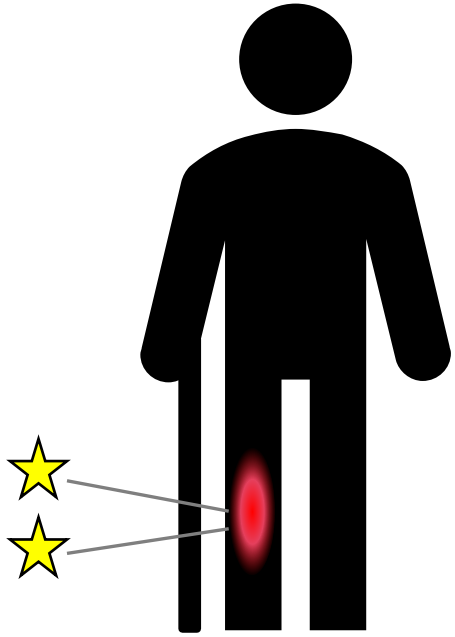


Tools leave room for improvements

- Tools are limited, not easily customizable
  - Tool dictates, not its user
- Modeling functionality is added to a non-modeling tool
  - IDE tool with modeling
  - Usability based on text editing
- Collaboration support missing or limited
- Diff and merge based on text
- Diversity is not supported
  - One prefers local, other virtual and third browser-based

Result: All suffer

- Users suffer when trying to learn and apply languages
- Tool developers suffer when trying to implement modeling support, or tools that use the models
- Language developers suffer when defining languages

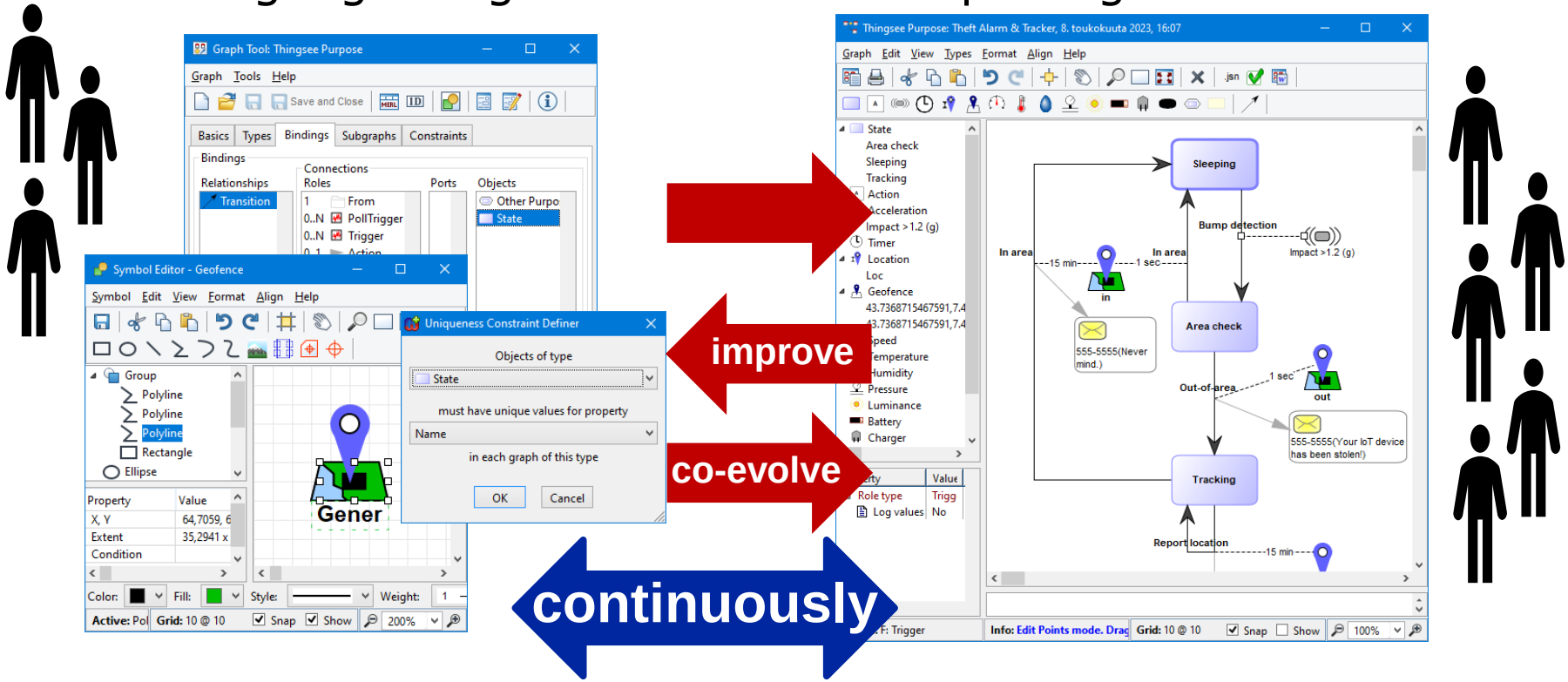


What we can do, how to improve?

1. Treat language definition as a whole
   - do not separate grammar, rules, symbols, transformations
2. Use proper metalanguages
   - ≠ MOF (made originally to define CORBA data structures)
   - = Language that is built to define languages (not just the basic minimum but manuals, guidelines, transformations)
3. Get users involved: collaborative language definition
   - Create, test and improve together

Experts define languages & generators

Team models with domain concepts & generate code

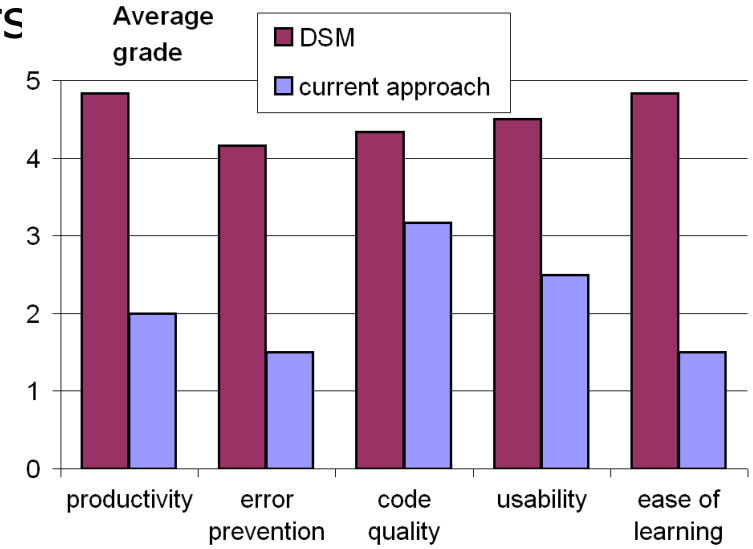


What can we do, how to improve?

1. Treat language definition as a whole
   - do not separate grammar, rules, symbols, transformations
2. Use proper metalanguages
   - ≠ MOF (made to define CORBA data structures)
   - = Language that is built to define languages (not just the basic minimum but manuals, guidelines, transformations)
3. Get users involved: collaborative language definition
   - Create, test and improve together
4. Apply proper tools: many language workbenches exist
   - If difficult or time-consuming, stop using that tool
   - e.g. tools that require coding do not believe in modeling and abstractions, skip them

Better results, satisfied users

- E.g study from language users
  - Usability
  - Easy of learning
  - Code quality
  - Error prevention
  - Productivity



- Grading from 6 developers:
  - results (scale 1-5, 5 best):

What do companies say?

Panasonic

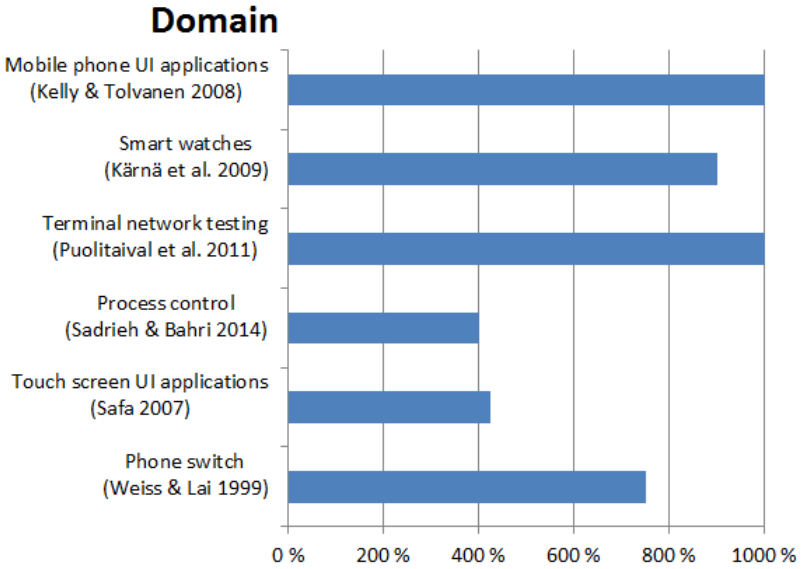
"**5-fold** productivity increase when compared to standard development methods"

AIRBUS

"The quality is clearly better, simply because the modeling language **rules out errors**"

POLAR

"**750%** increase in productivity, and **greatly** improved quality in the code and development"



Thank you

Questions?

Comments?

Counter-arguments?

Experiences?

Contact: jpt@metacase.com