TECHNISCHE UNIVERSITÄT DRESDEN
LANCASTER UNIVERSITY
Telefónica
THE UNIVERSITY of York

Fakultät Informatik, Institut für Software- und Multimediatechnologie, Lehrstuhl Softwaretechnologie

Abstracting Complex Languages through Transformation and Composition

Jendrik Johannes, Steffen Zschaler,
Miguel A. Fernández, Antonio Castillo,
Dimitrios S. Kolovos and Richard F. Paige

08/10/2009, MoDELS '09, Denver, Colorado

Abstracting Complex Languages

Stakeholders:

Domain Expert 1:
Telecommunications
Device Configuration

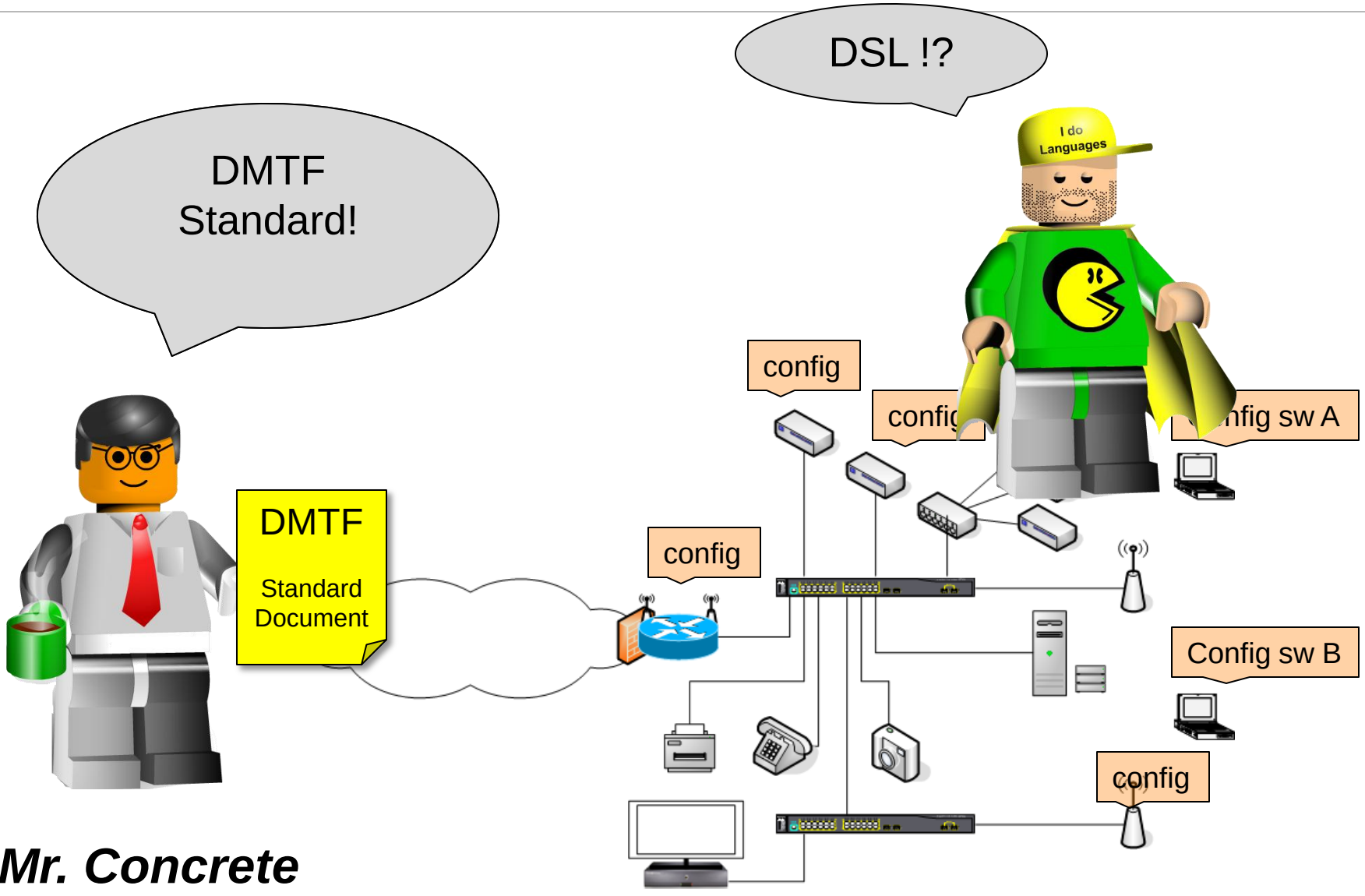
Mr. Concrete

Domain Expert 2:
Telecommunications
Network Configuration

Mrs. Abstract

Language
Engineer

A Complex Problem



Building the ***Concrete*** *DSL*

Defining Syntax

DMTF

Standard Document

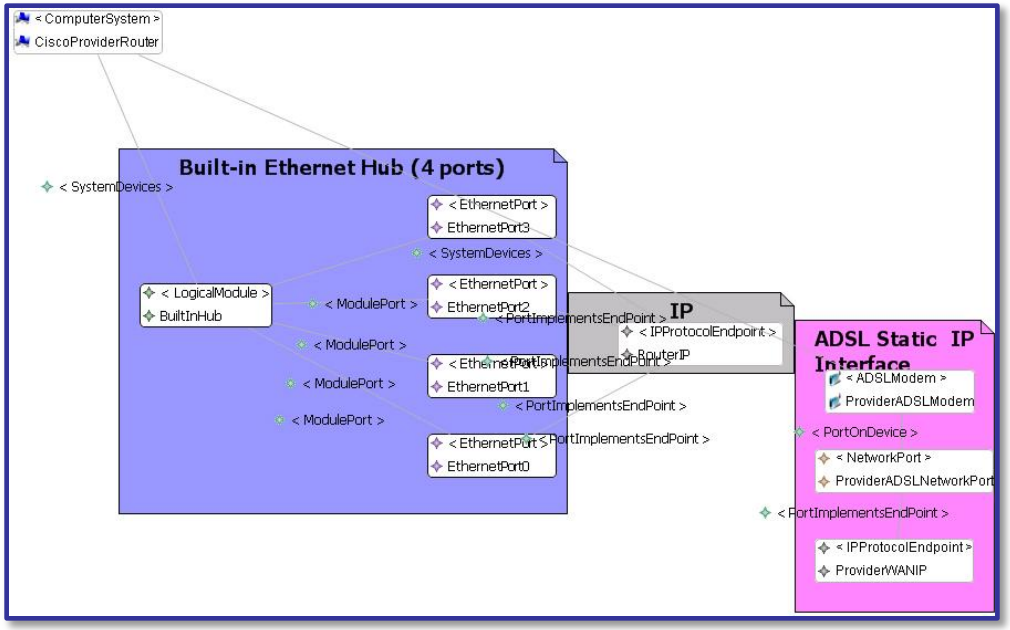
Domain Meta Model

epsilon

EuGENia

minimal input

emftext



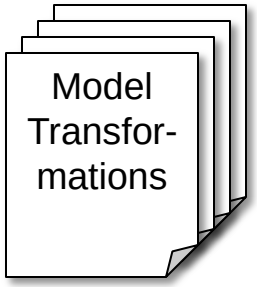
In the Language Engineering Lab

Building the ***Concrete*** *DSL*

much more input

ETL, QVT,
MOF2Text, etc.

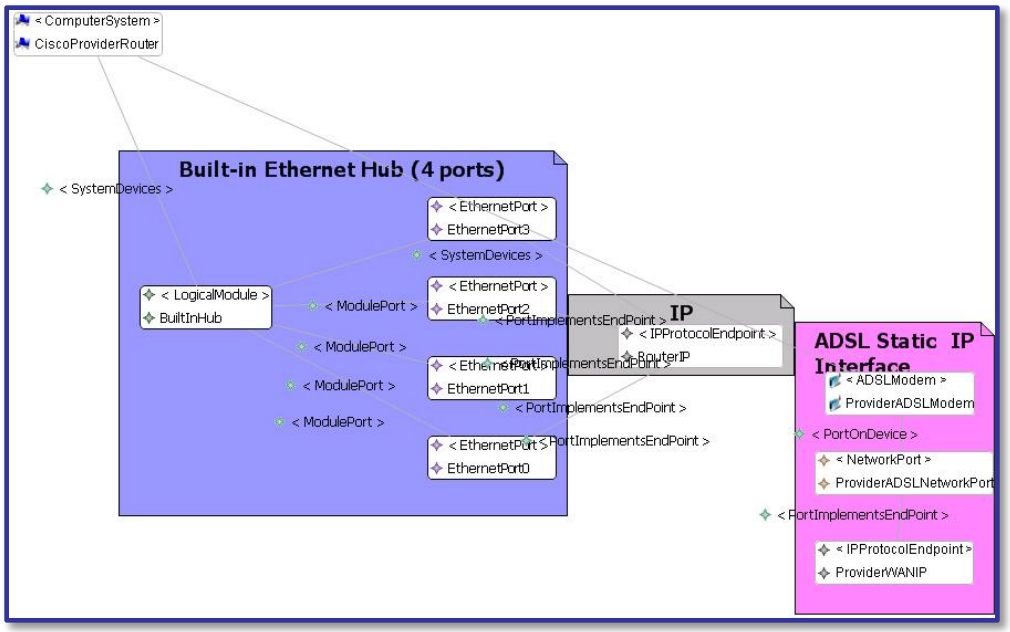
Defining Semantics

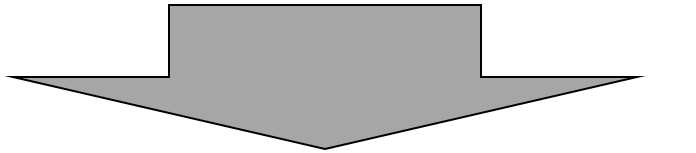


Some Time Later…

~~complex~~
simple

Mr. Concrete





Config Scripts

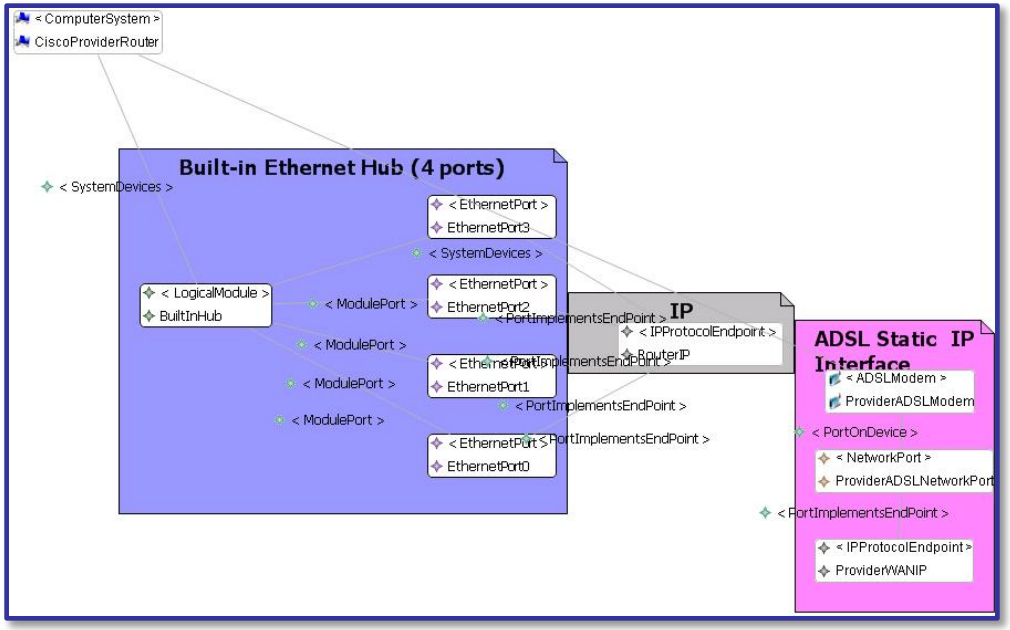
Soft-ware

Docs

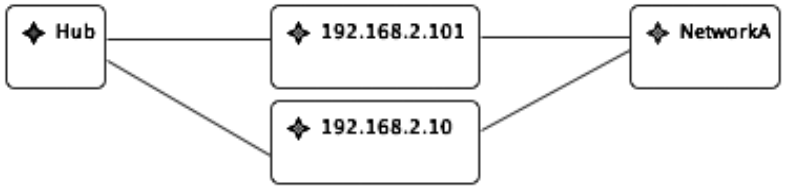
Suddenly…

~~DMTF Standard Document~~

~~simple~~ **complex!**



Abstract Domain Concepts



Mrs. Abstract

In the Language Engineering Lab

*Building the **Abstract** DSL*

Abstract Domain Concepts

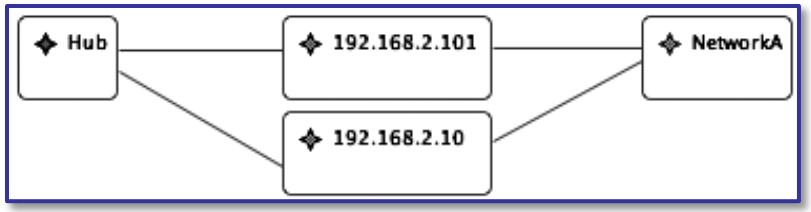
minimal input

epsilon EuGENia

emftext

Domain Meta Model 2

Defining Syntax



In the Language Engineering Lab

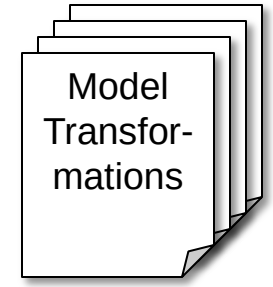
Building the ***Abstract*** *DSL*

?

much more input

ETL, QVT, MOF2Text, etc.

Defining Semantics



Our Assets

- Languages in the same domain
  - One is an abstraction of the other

- Experts of the domain
  - Can use tooling for the concrete language

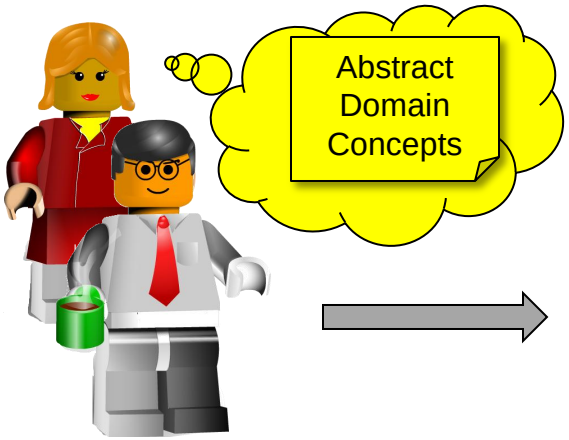
- Specific class of model transformations
  - Reoccurring mapping patterns can be reused

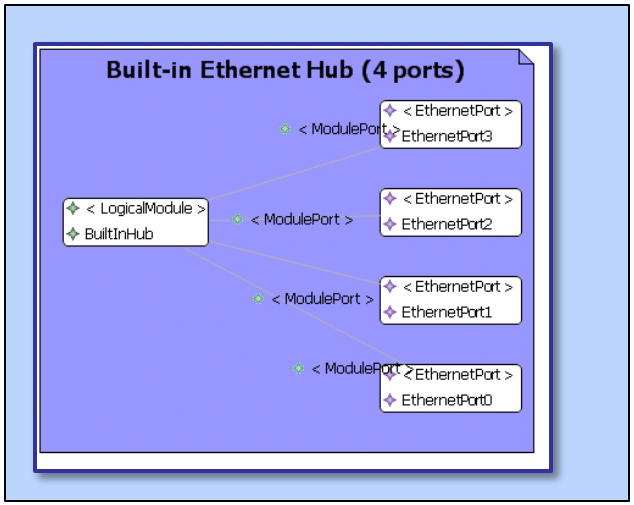
Abstract Language Building

```
Abstract_Concept   --->  Set_Of_Concrete_Concepts
```

✔ ✔ ✔

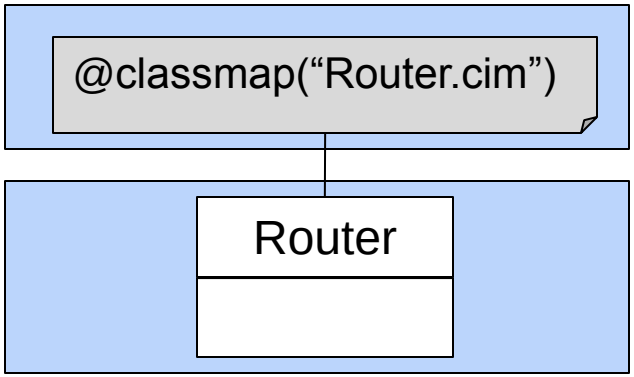
- Model each *abstract concept* in the *concrete DSL* as ***model fragment***



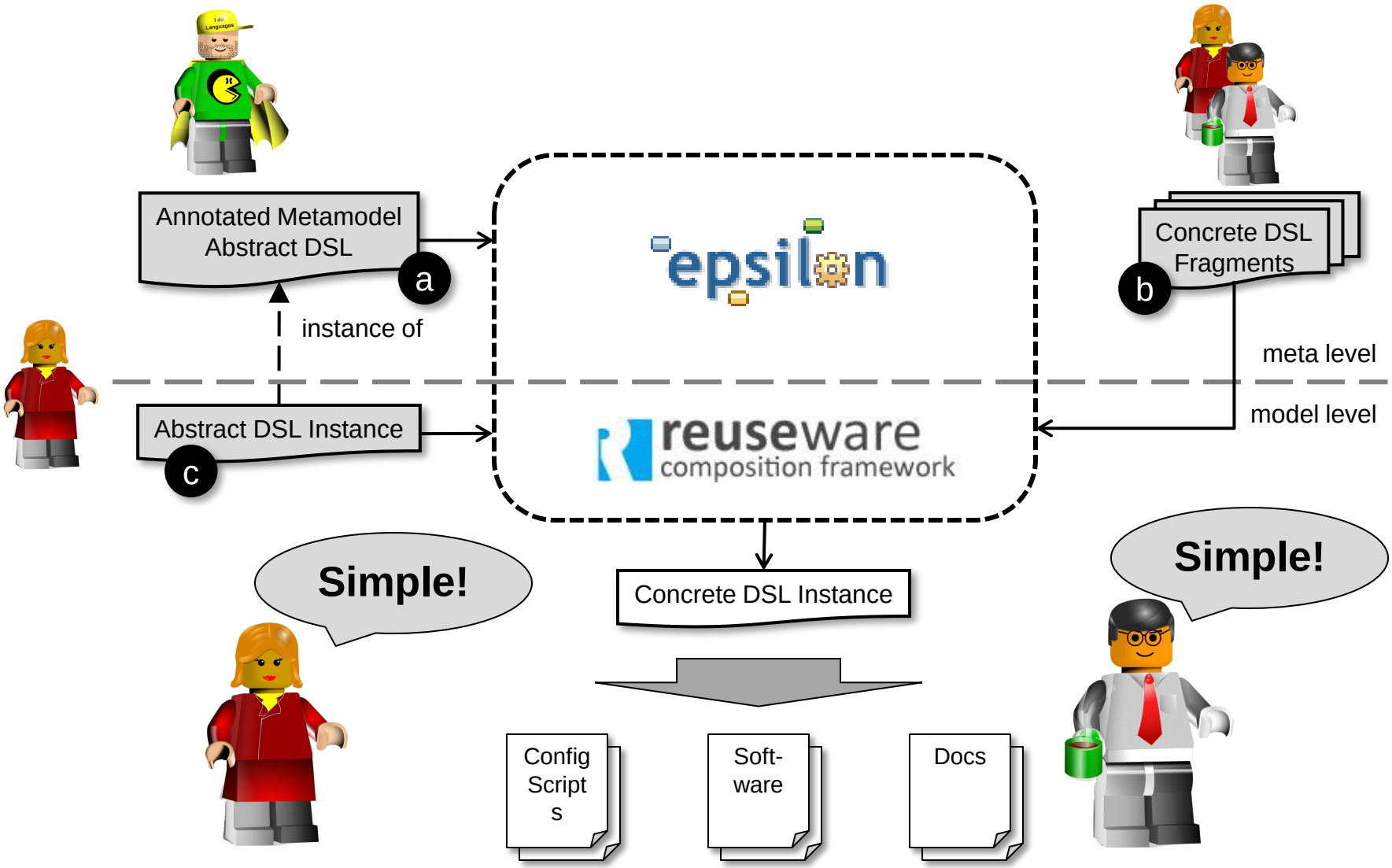


- Annotate *abstract DSL metamodel* with mapping using *mapping patterns*

minimal input



Implementation



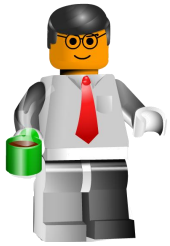
Conclusion

- Case study based on Common Information Model (CIM) – DMTF Standard
- Simplified construction of abstract languages
  - Using Mapping Patterns
    1. Element Mapping
    2. Element Mapping with Variability
    3. Attribute Mapping
    4. Link Mapping
  - Separation of concerns in model transformations
    - *Mapping* separated from *definition of concrete model fragment*
  - Use of concrete language tooling to define concrete model fragments
    - Better integration of domain experts
- Prototype based on Epsilon and Reuseware
  - http://www.eclipse.org/gmt/epsilon/
  - http://www.reuseware.org/
  - See also: http://reuseware.org/index.php/Abstract_CIM_DSLs

Thank You

http://www.eclipse.org/gmt/epsilon/

http://www.reuseware.org/



The stakeholders in this presentation were
modeled with the Reasonably Clever Mini-Mizer
(www.reasonablyclever.com/mm2/)