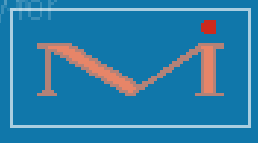
Stupid Modeling Tricks

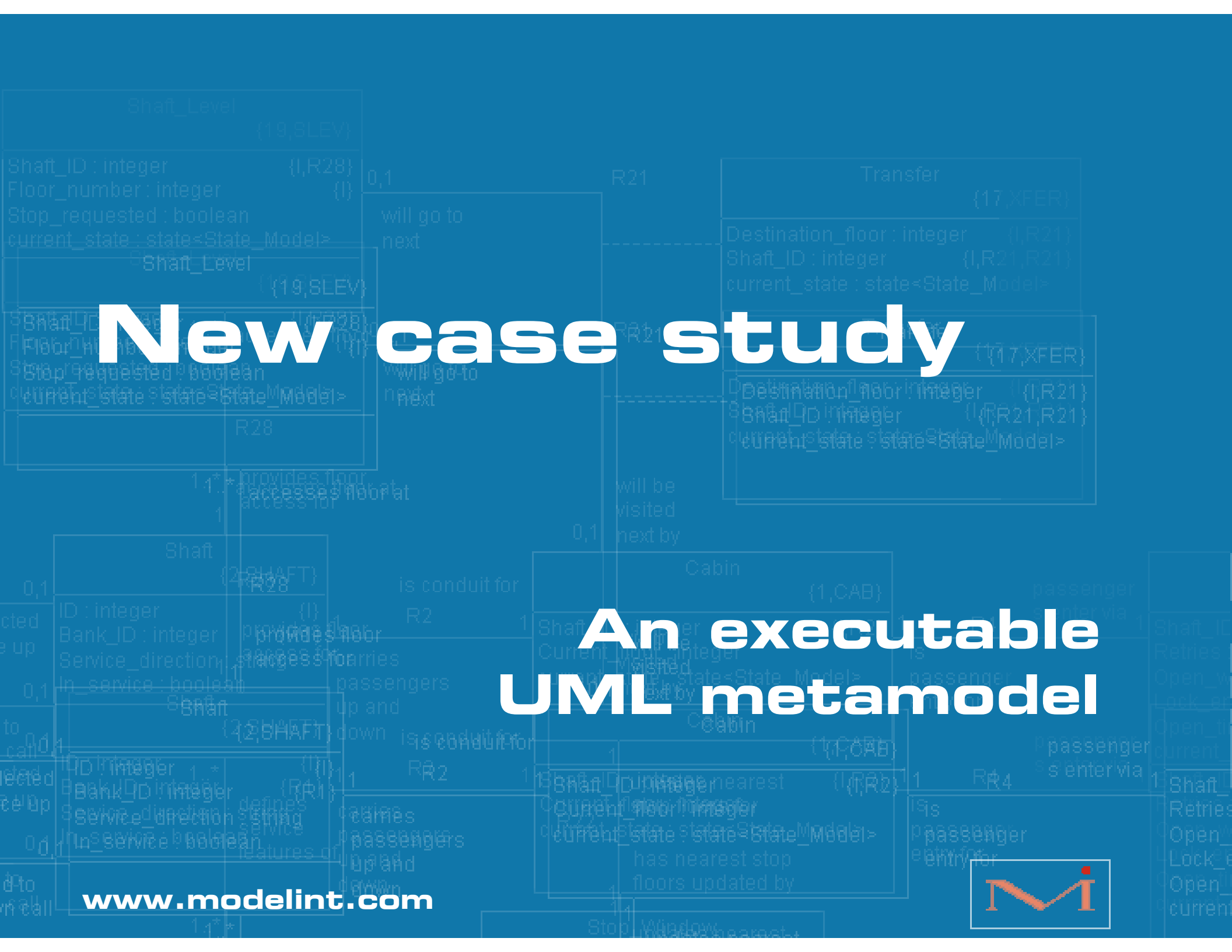
Leon Starr

Model Integration, LLC.

Version 0.1

www.modelint.com




New case study
An executable UML metamodel
www.modelint.com

Some existing metamodels

- OMG standard (UML, but not executable)
- Project Technology (executable, but Shlaer-Mellor)
- Others I don't know about… public?
- New! Model Integration – open source

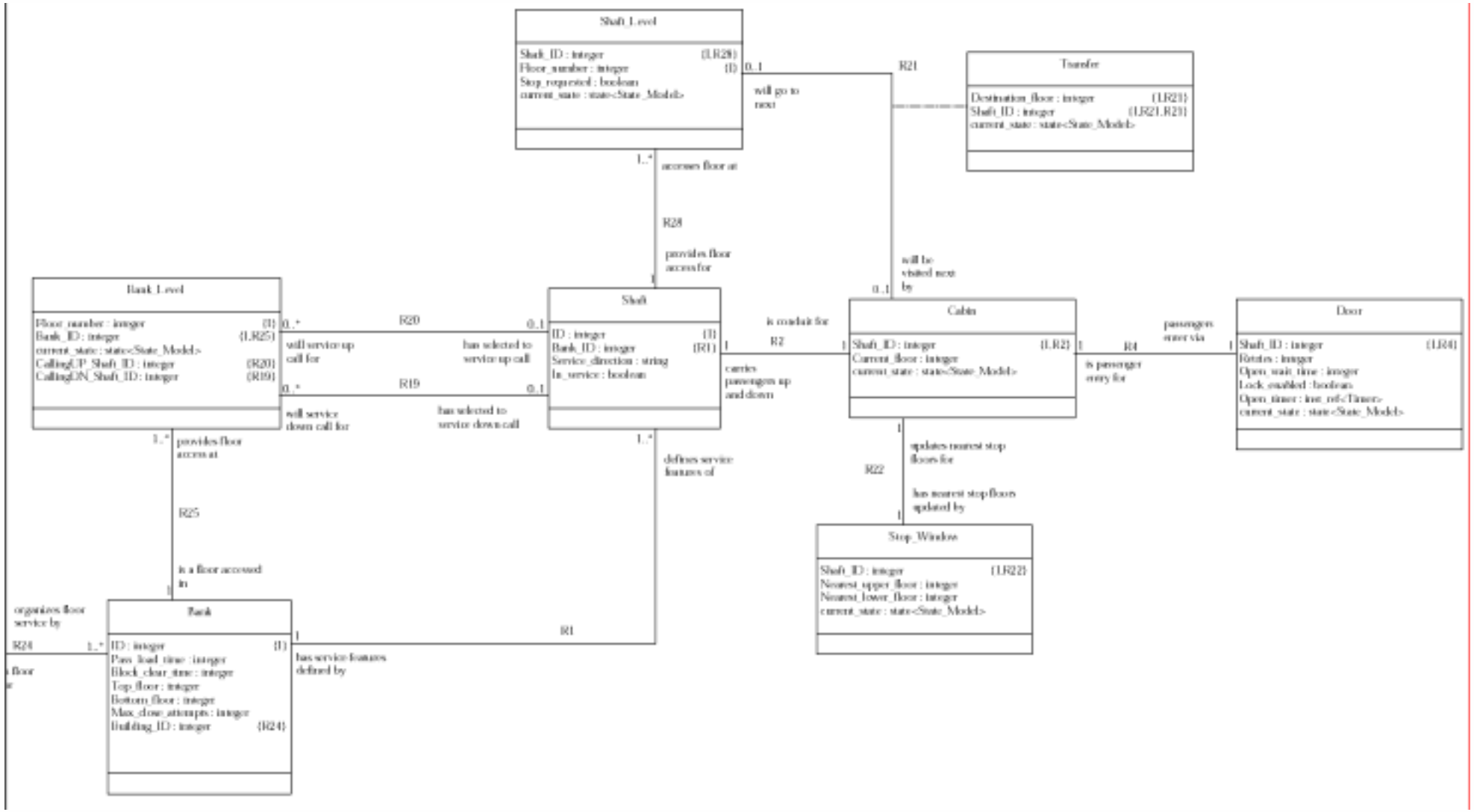
Executable UML differences

- Identifiers not essential
- Referential attributes not essential
- No many-associative objects
- Terminology differences
- Constrained generalization instead of supertypes
- Constrained loops are modeled

The main difference

Less clutter

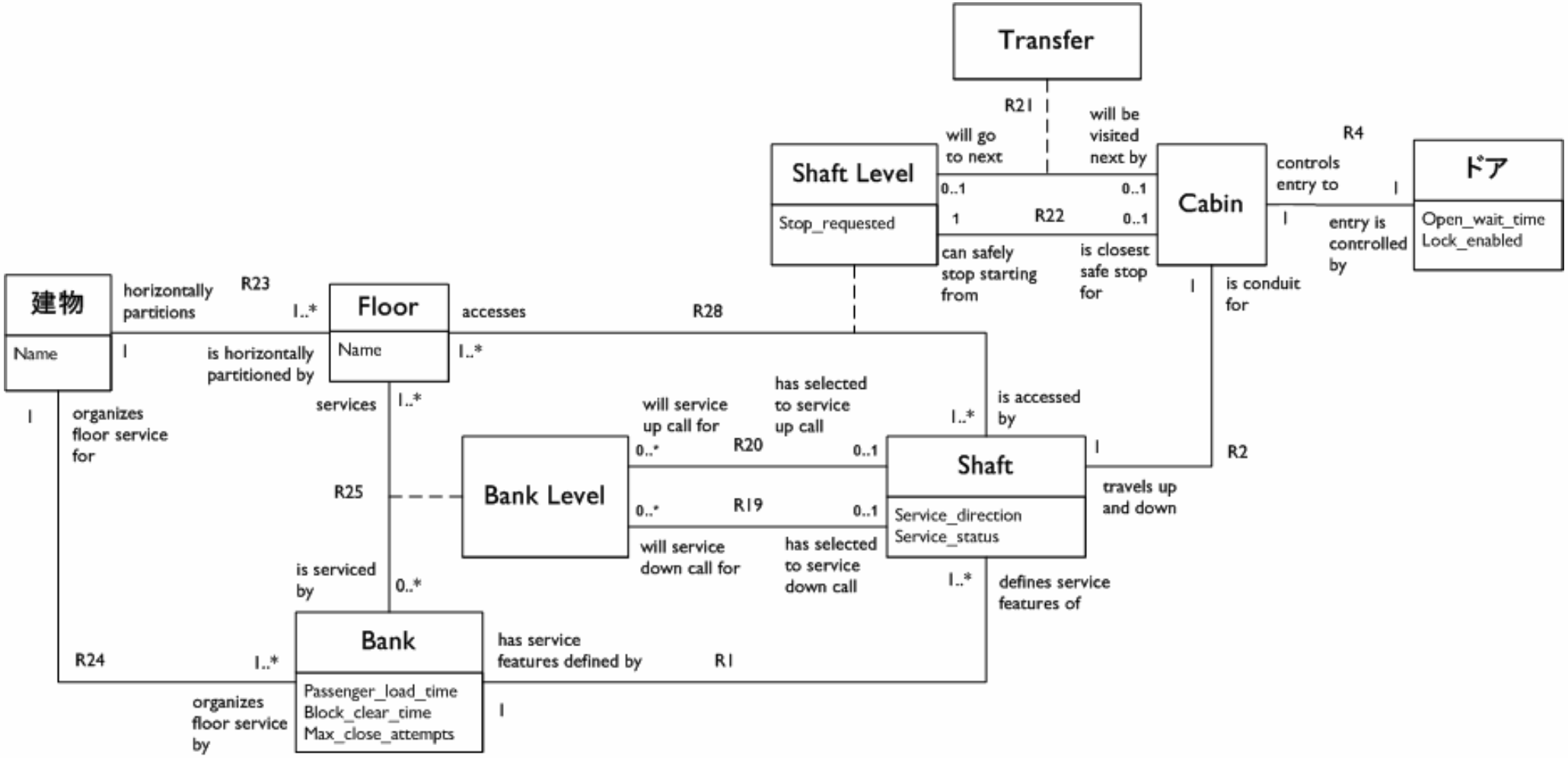
Elevator with excess attributes



 Rev 0.1

www.modelint.com

Executable UML Elevator



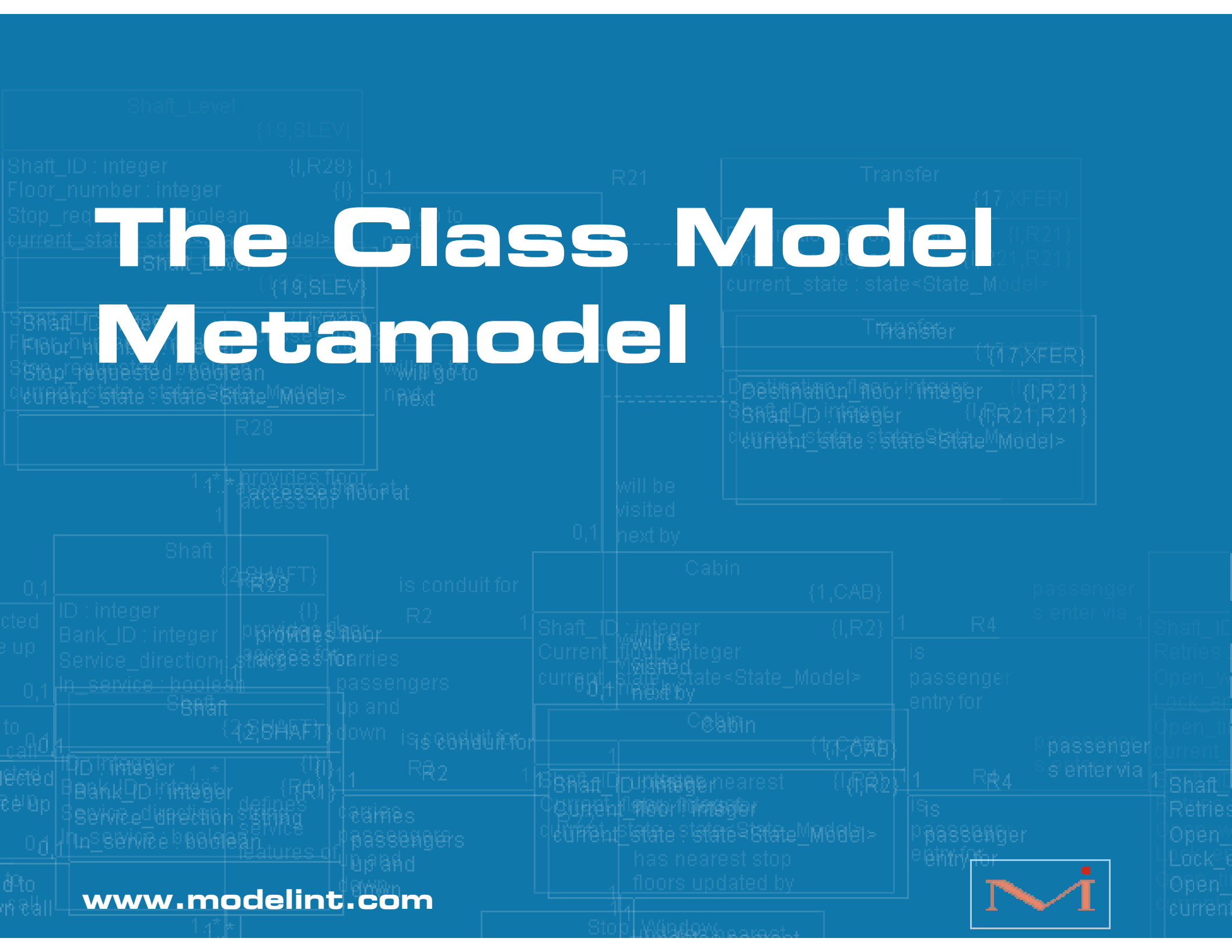
The MI Metamodel

- Restrictive – no lint
- Open source – you can…
  - Download it
  - Suggest changes
  - Change it yourself and submit an update
  - Use it to build your own tools
  - Just keep the author credits intact

Topics for today

- Modeling tricks (Class model metamodel)
- Modeling tricks (State model metamodel)
- Why are we doing this?
- Utilities based on this metamodel
- Robots and stuff

The Class Model Metamodel

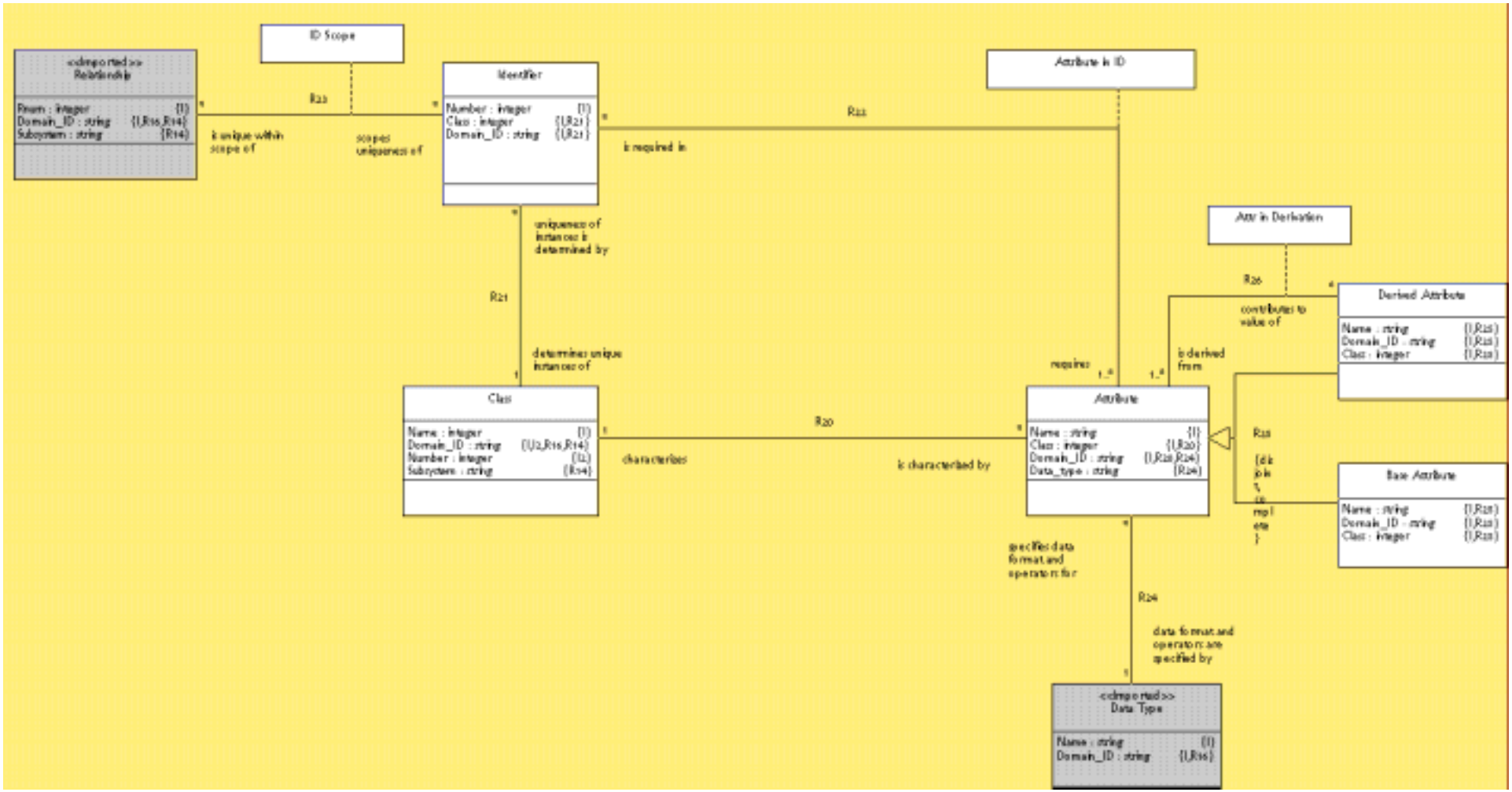


www.modelint.com

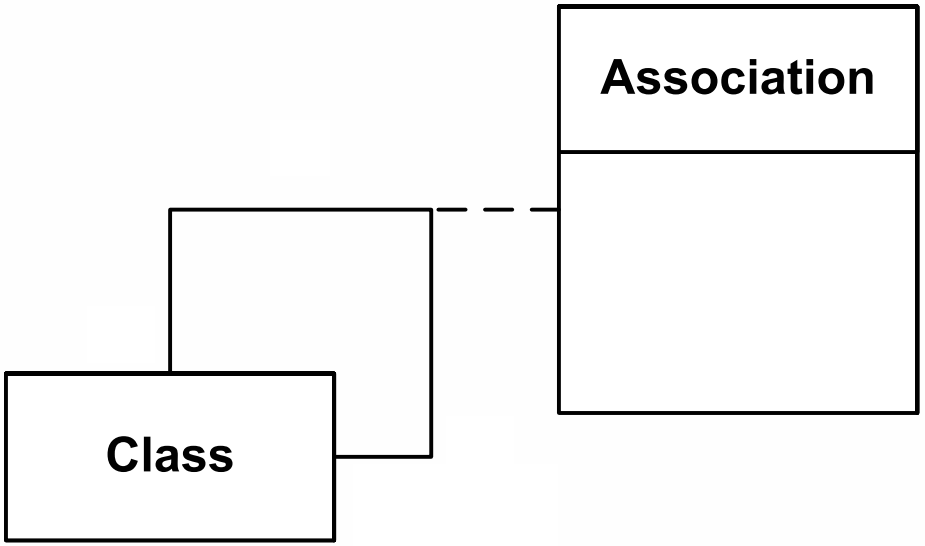
Elements to model

- Class and Attributes
- Associations (reflexive, non-reflexive)
- Association Class
- Generalization (multiple, overlapping)
- Domains and Subsystems
- Constrained Loops

Classes and Attributes



First stab - Associations



Which side is which?

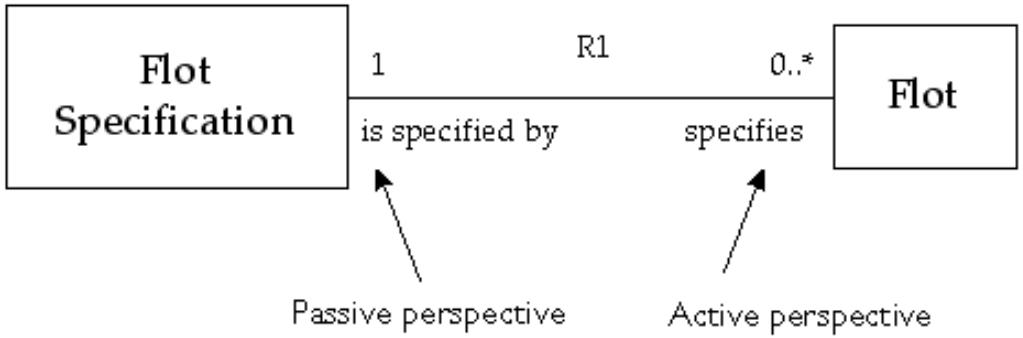
Associations

Association

ASSC

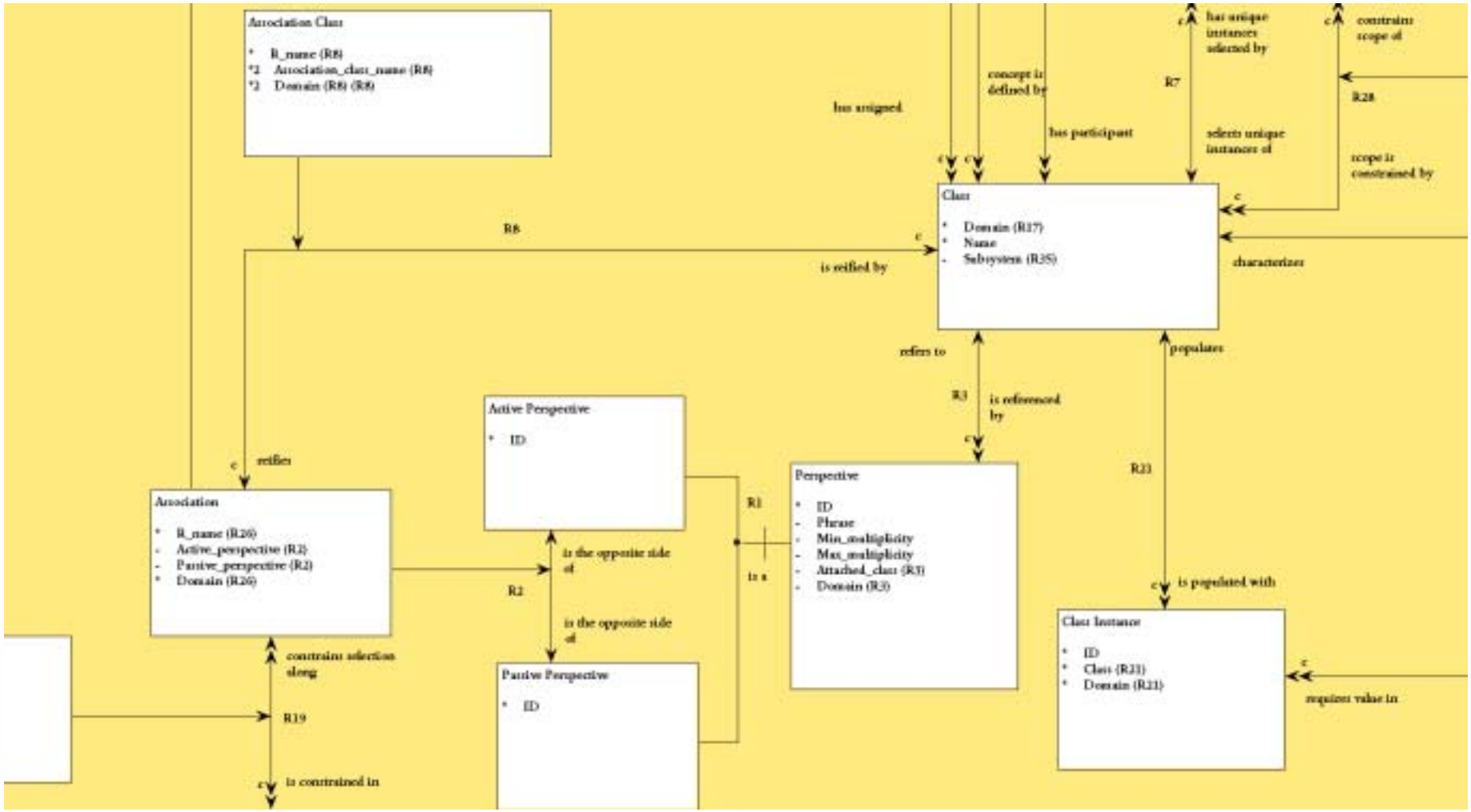
A association as defined in Executable UML.

Association = Active Perspective + Passive Perspective



An association is created by establishing an active perspective on one class and a passive perspective on another or the same class.

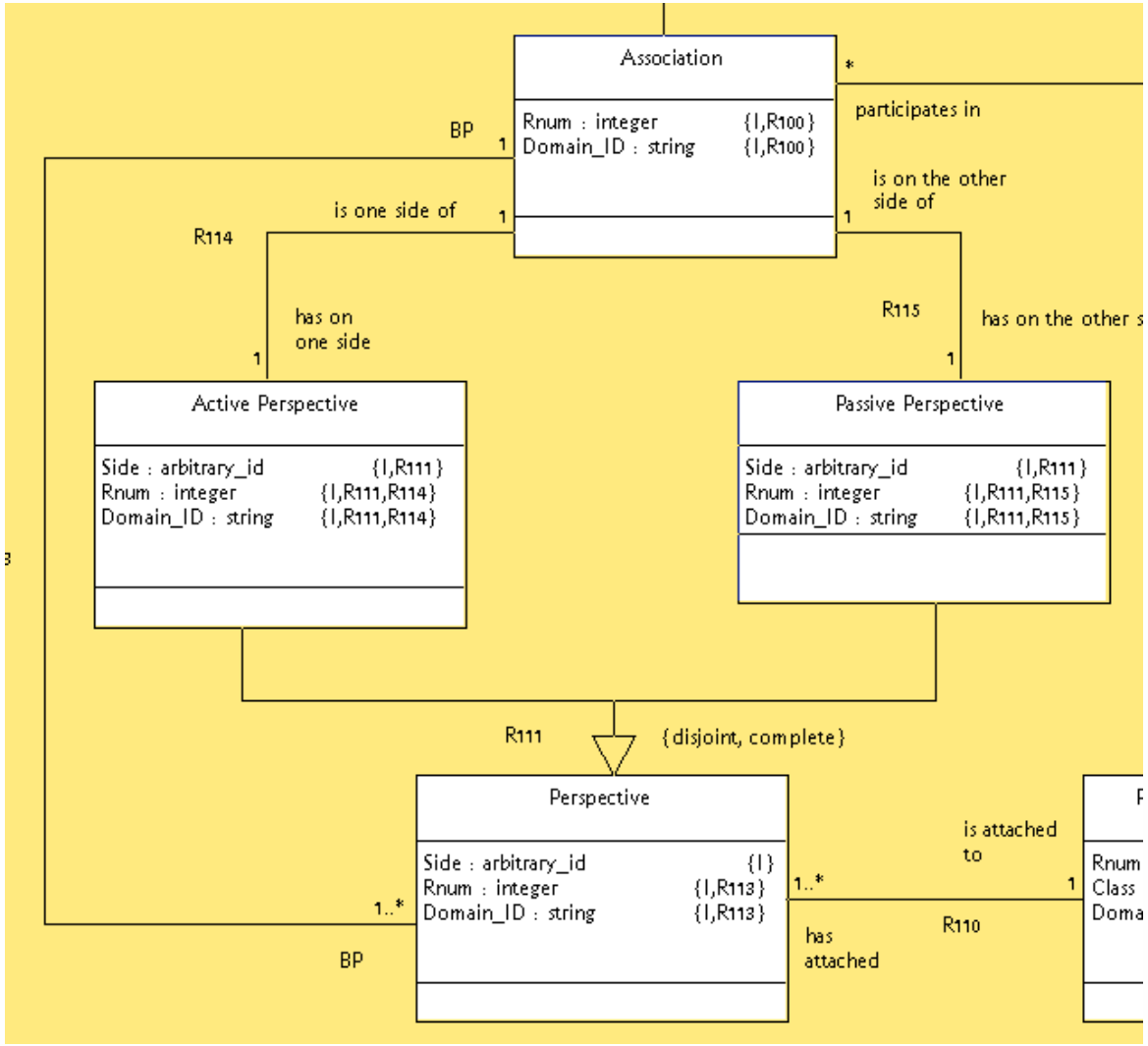
2nd stab at associations



 Rev 0.1

www.modelint.com

Assocations



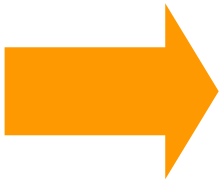
Association class - description

Relationship descriptions

R8 - participates in

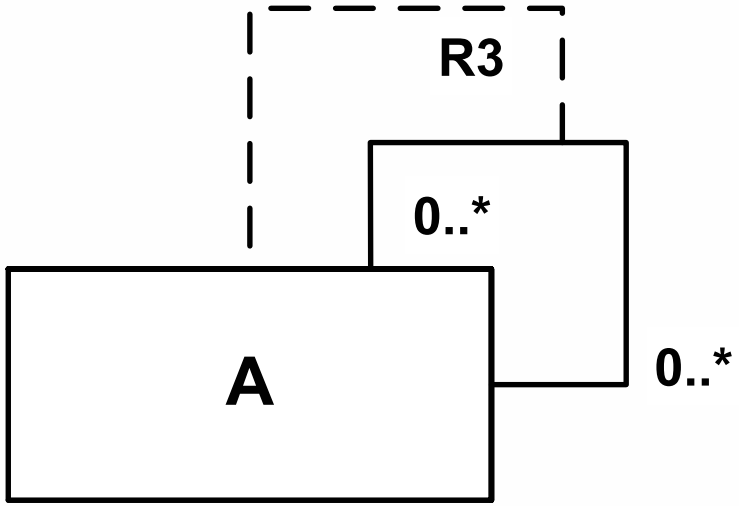
Class PARTICIPATES IN 0..* Association
Association HAS PARTICIPANT 1..* Class

A Class need not participate in any relationships, but that would make for an uninteresting Class Model. Typically a Class will participate in at least one Relationship. Keep in mind, that that one Relationship might be a generalization. In fact, it is common for a class model to have several classes in a generalization that do not participate in any Associations.



A Class may participate in more than one Association. In fact, a Class may be an association class in one Association and a participating class in a one or more other associations. Note that if a class is an association class, it can only be an association class in ONE Association.

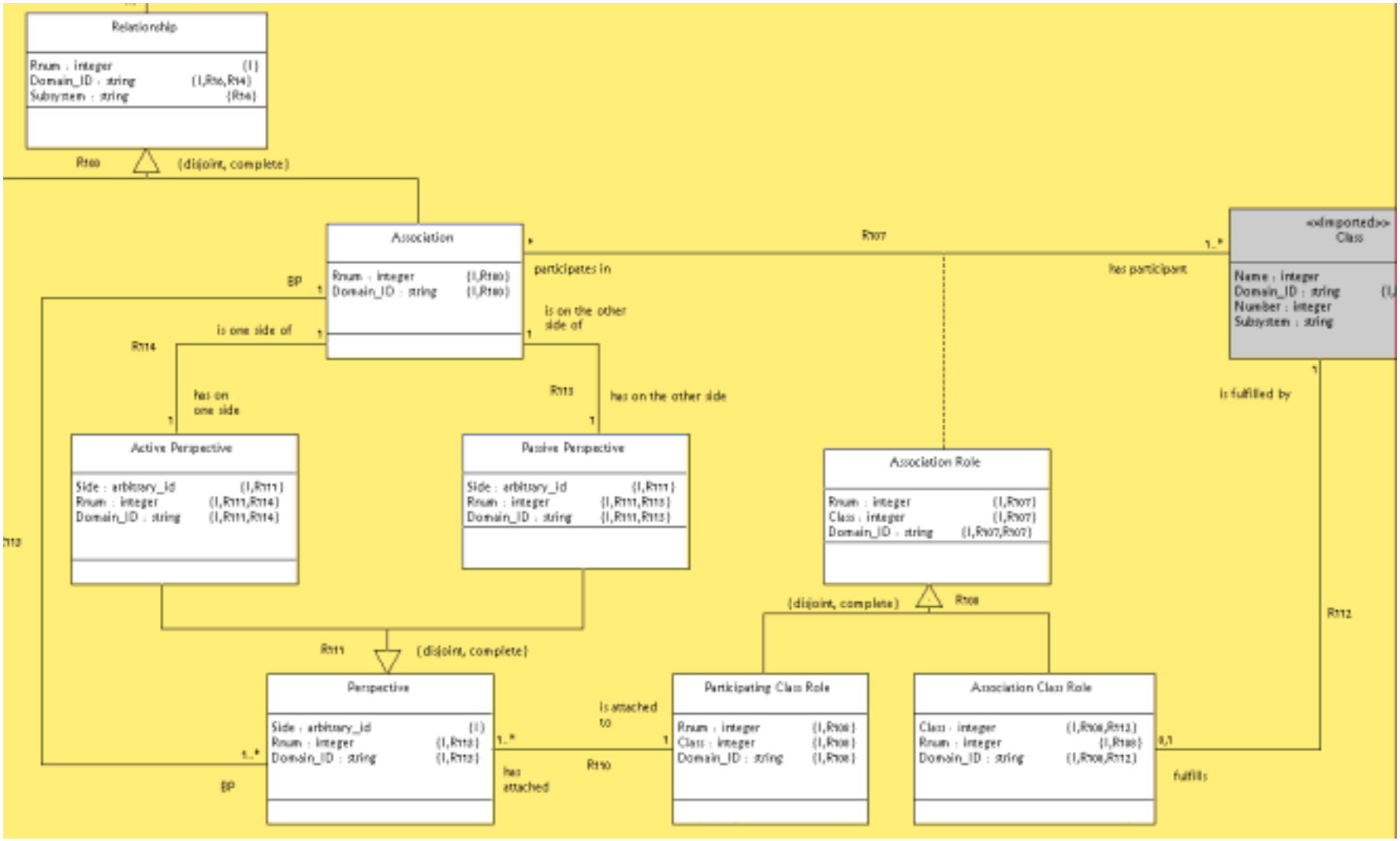
The illegal case



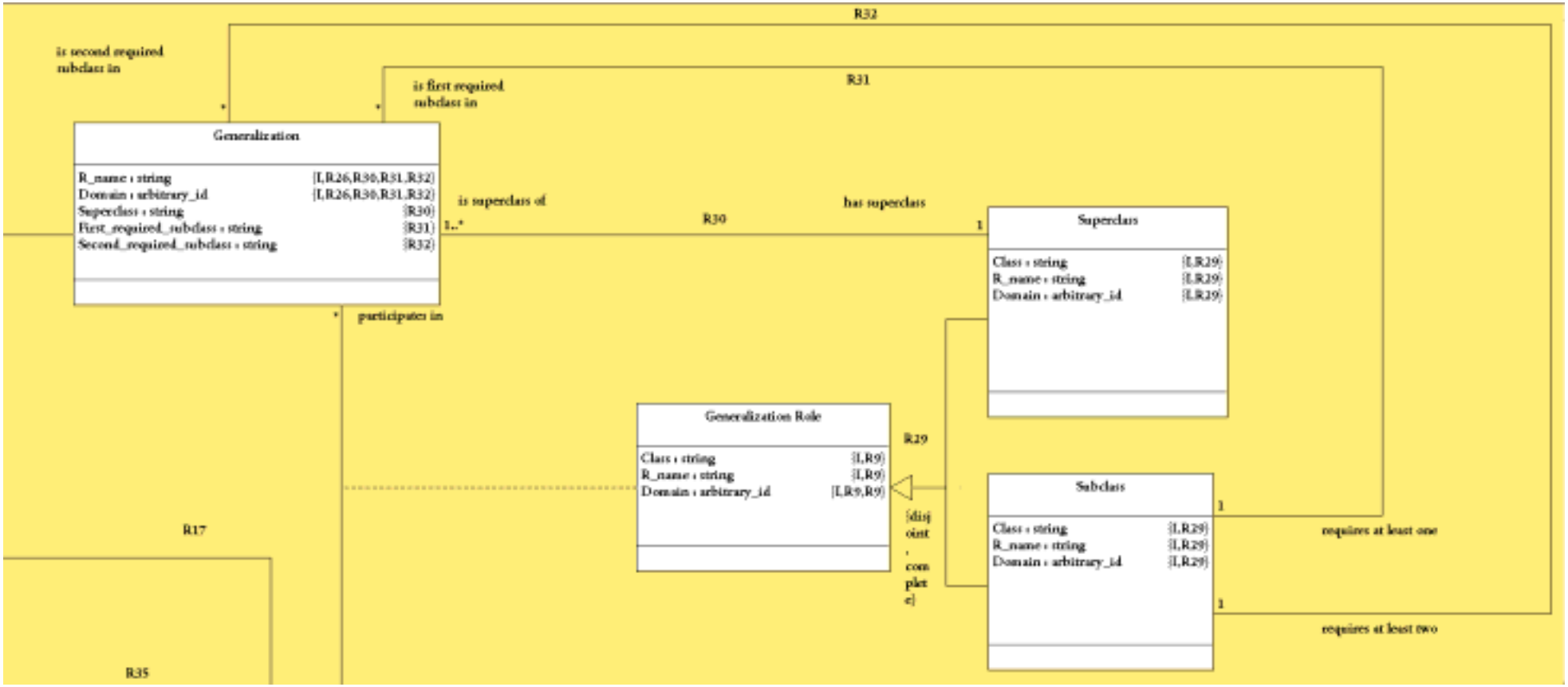
State the rules

- In a single association, a class may be EITHER a Participating Class OR an Association Class, but not both.
- A Class may be an Association Class in one relationship a Participating Class in another relationship, a superclass in yet another relationship, and so on.
- And remember, all Classes, regardless of role, have attributes, identifiers, etc. (A class is a class is a class!)

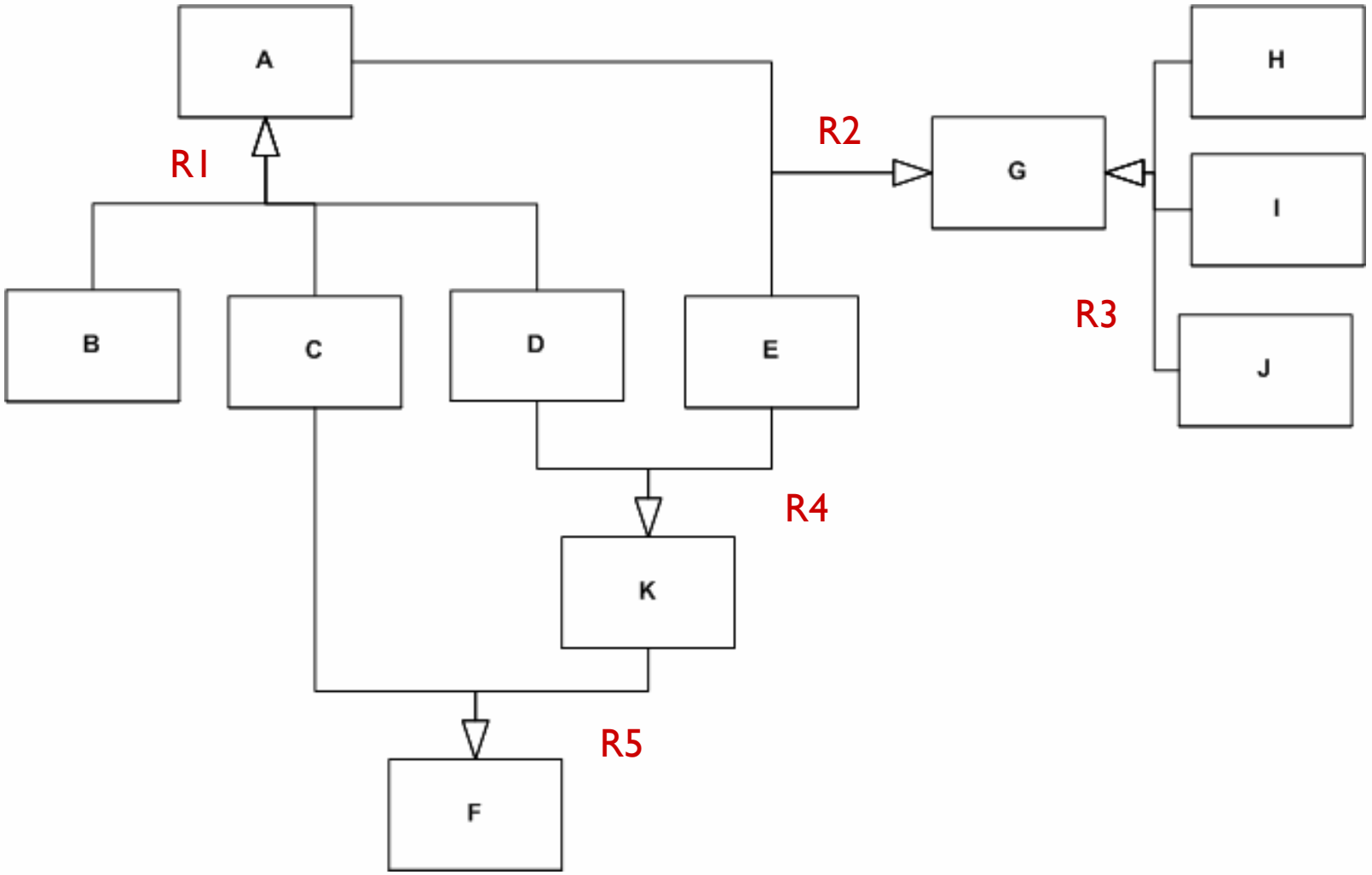
The assocation class



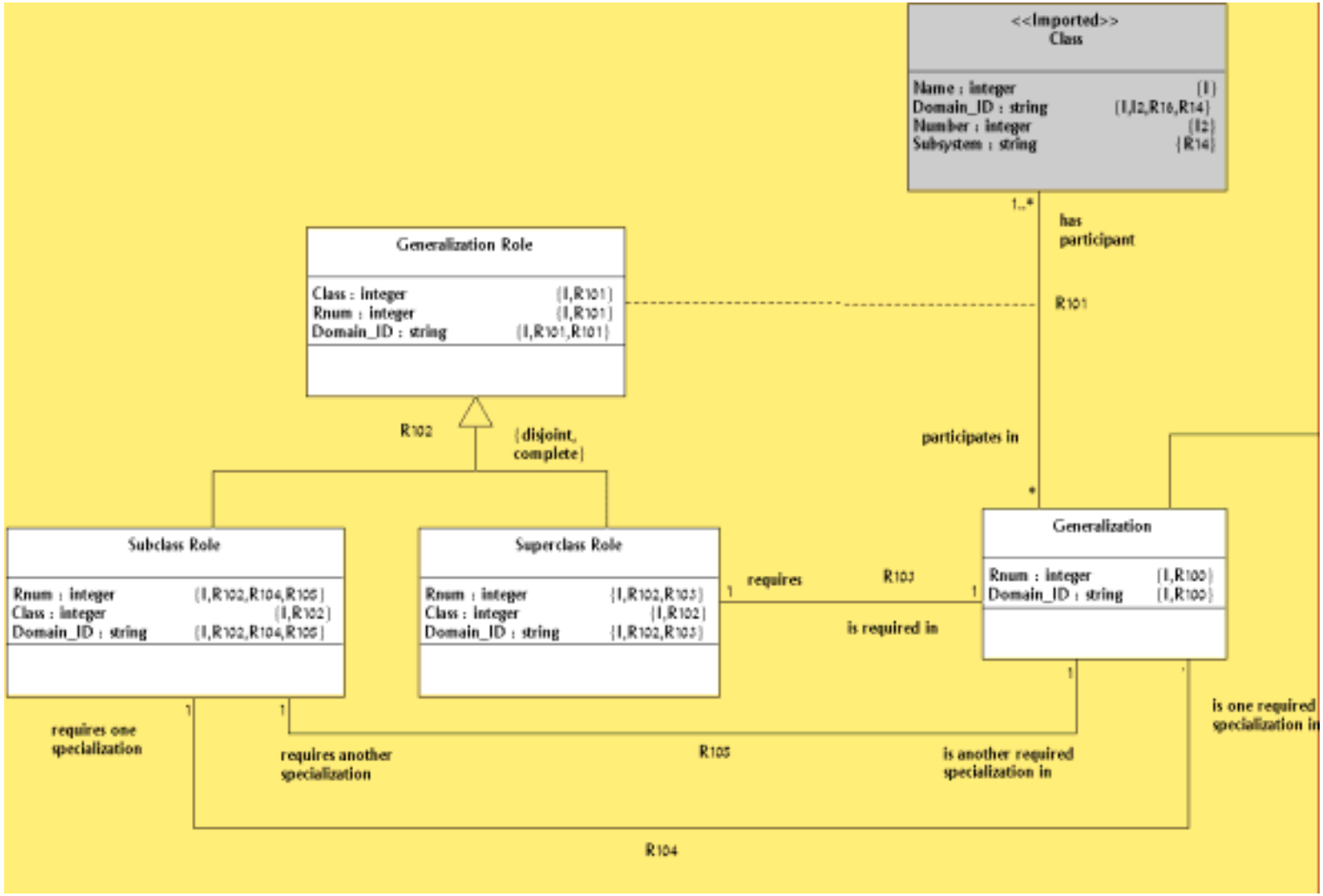
Generalization 1st stab



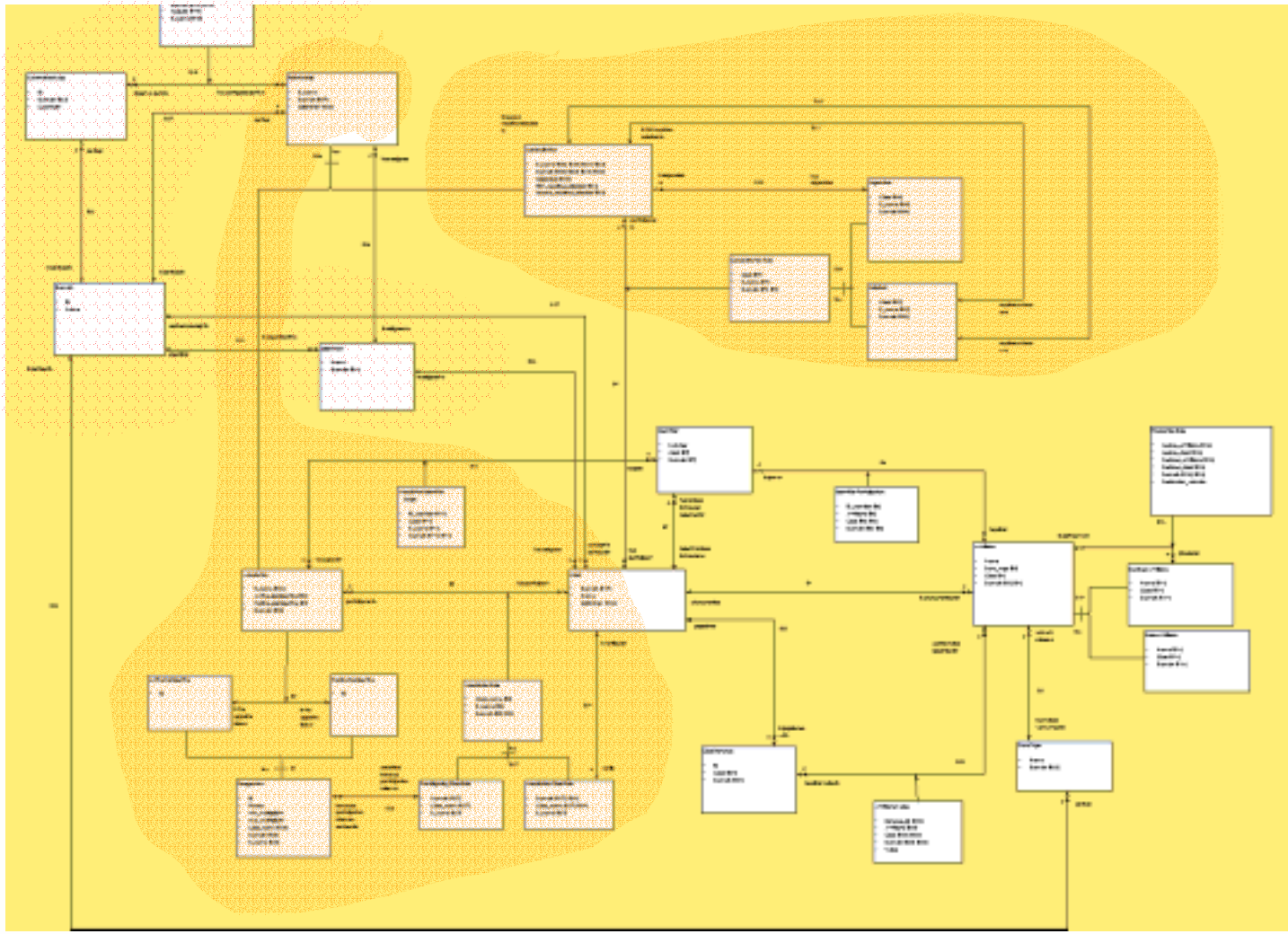
A complicated case



Generalization

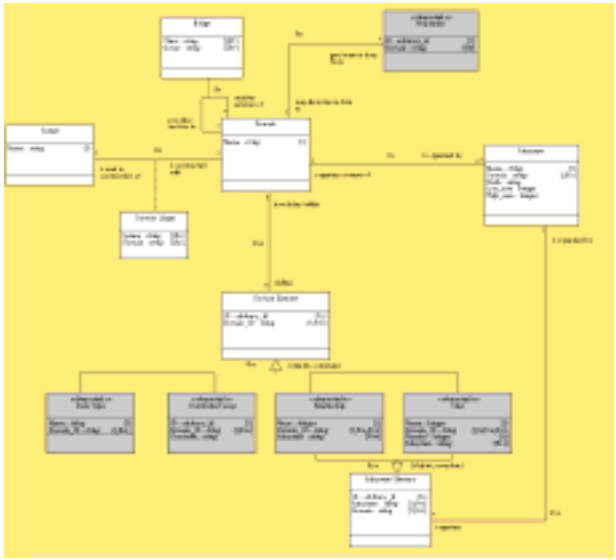


A big mess

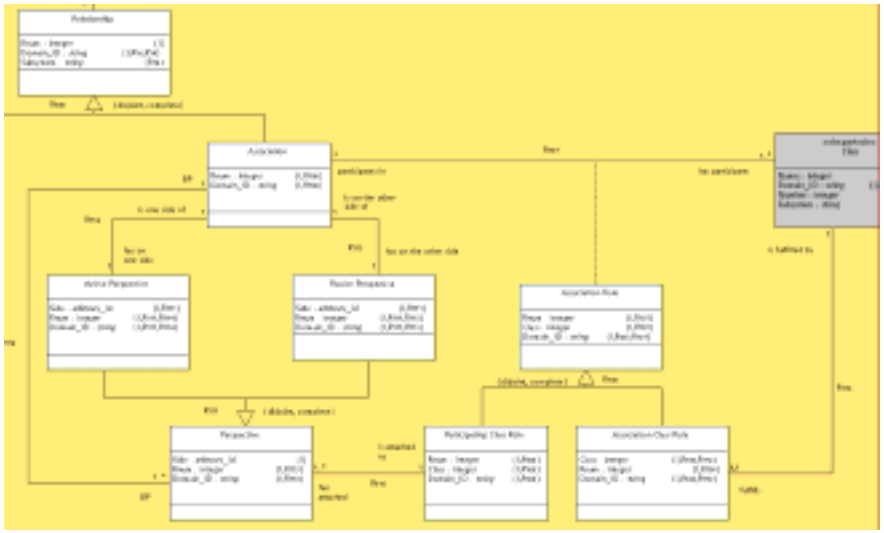


Class metamodel subsystems

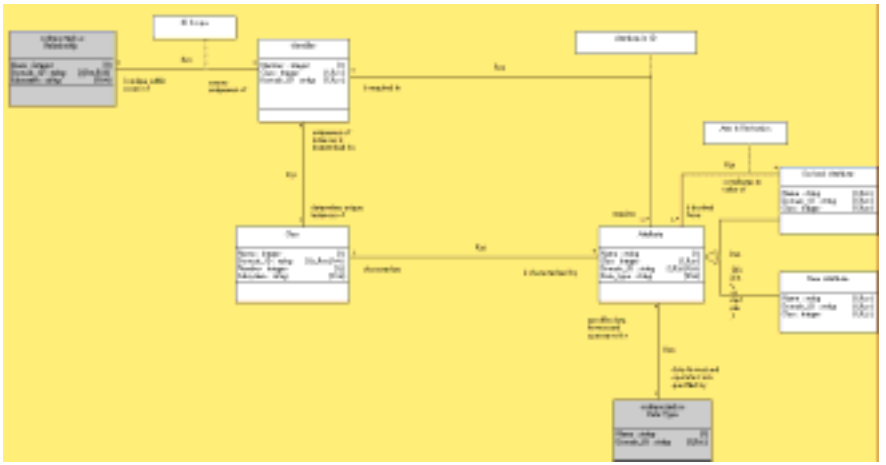
Domain



Relationship



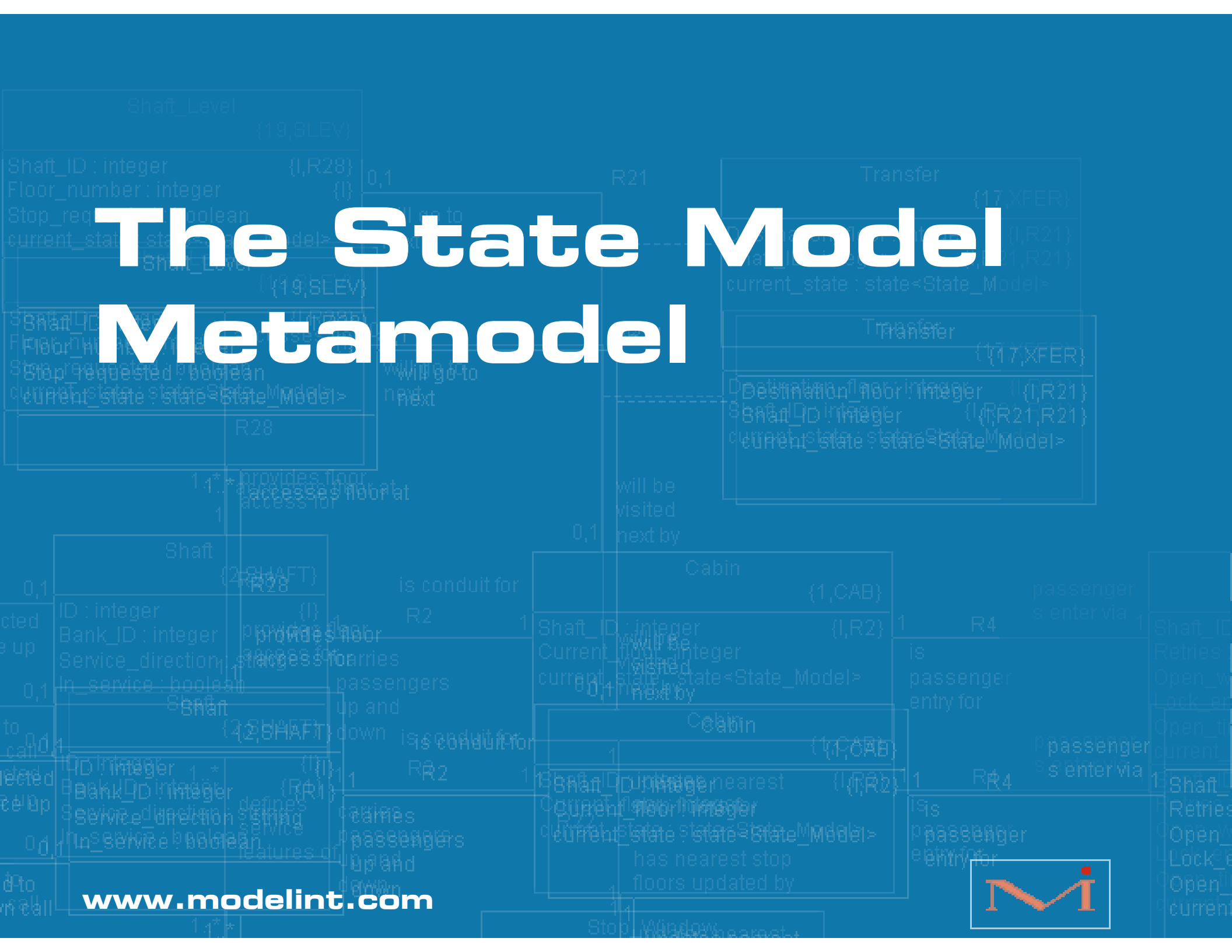
Class and Attribute



The other subsystems

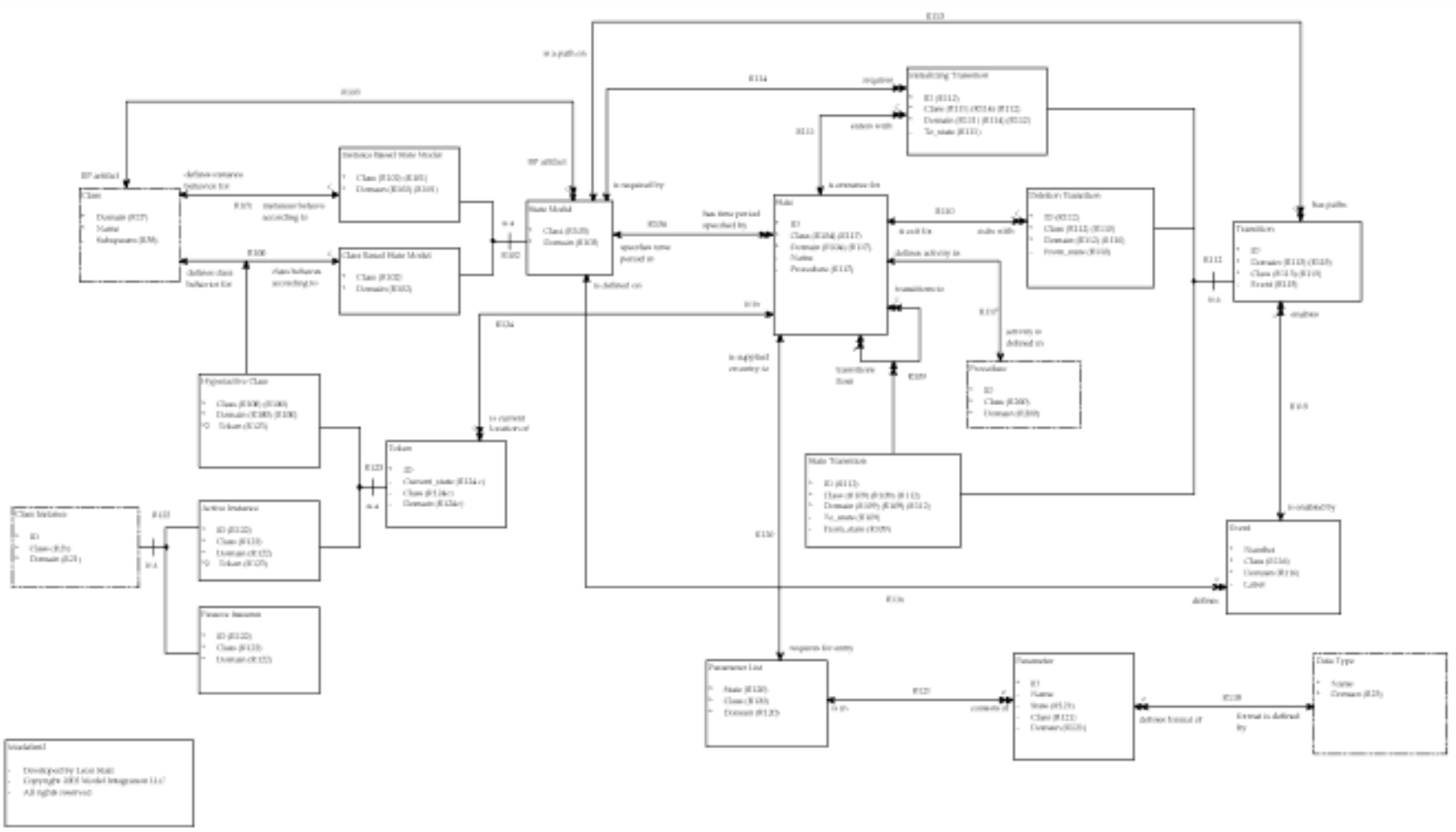
- Data type
- ✓ Population
- ✓ State model
- Action language

The State Model Metamodel

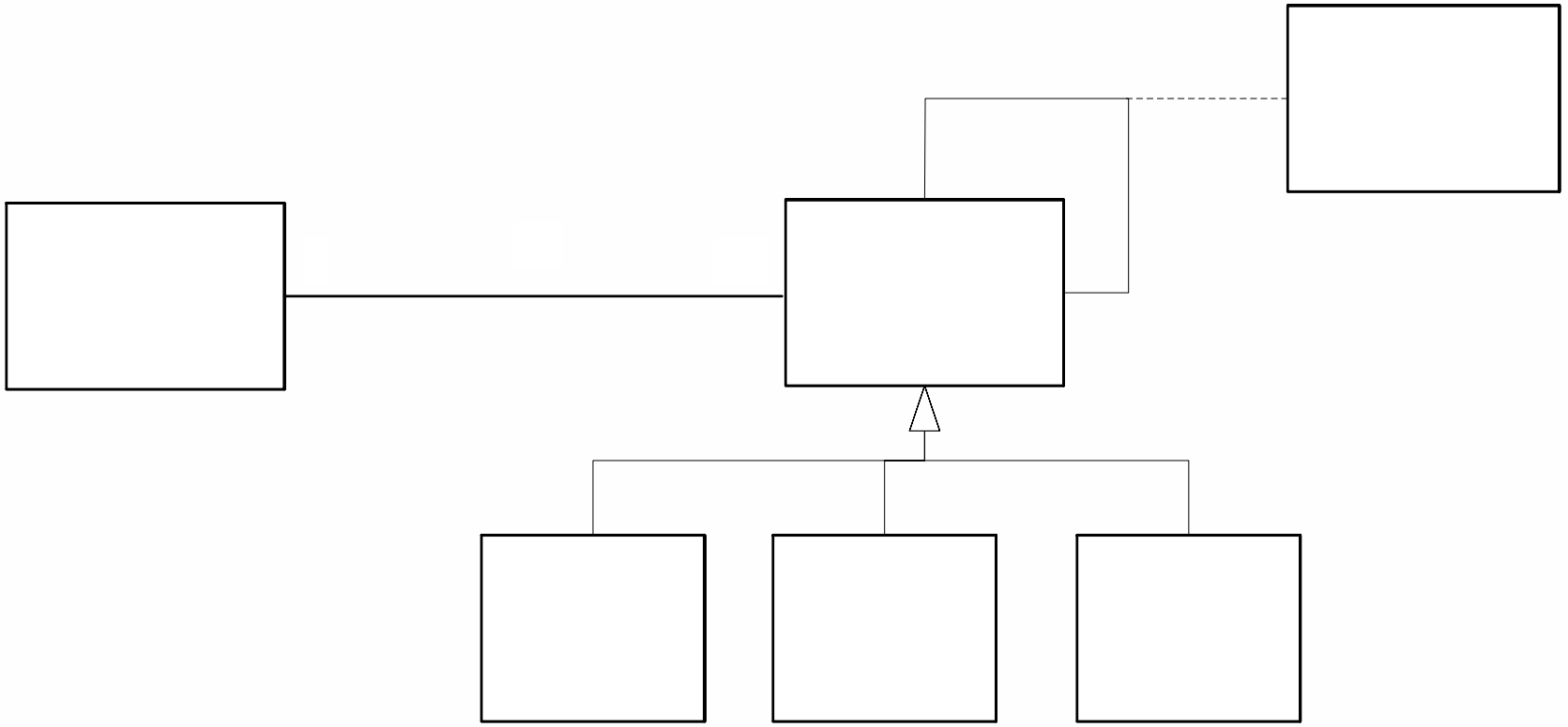


www.modelint.com

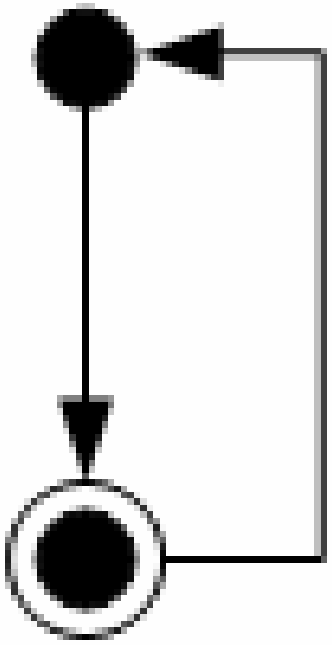
State model subsystem



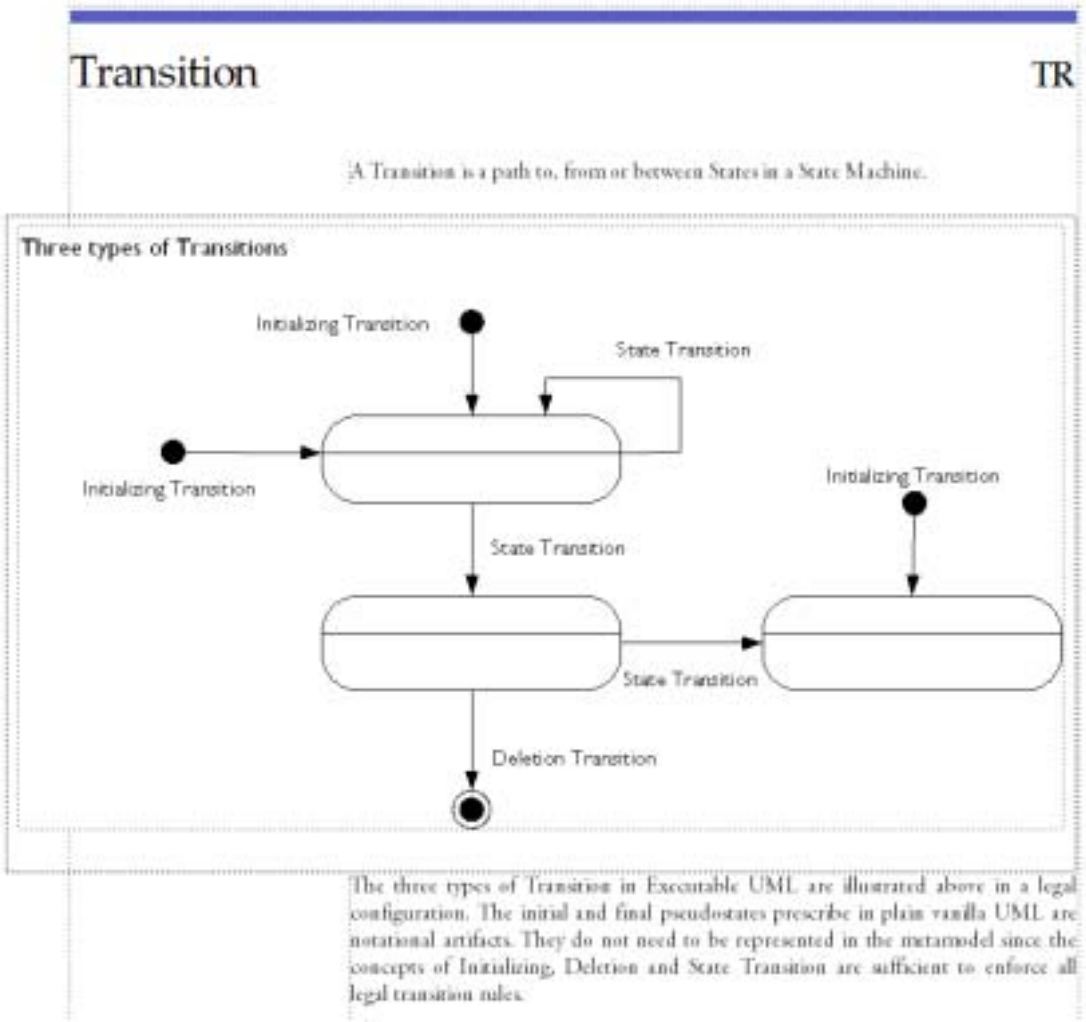
First stab – what's wrong here?



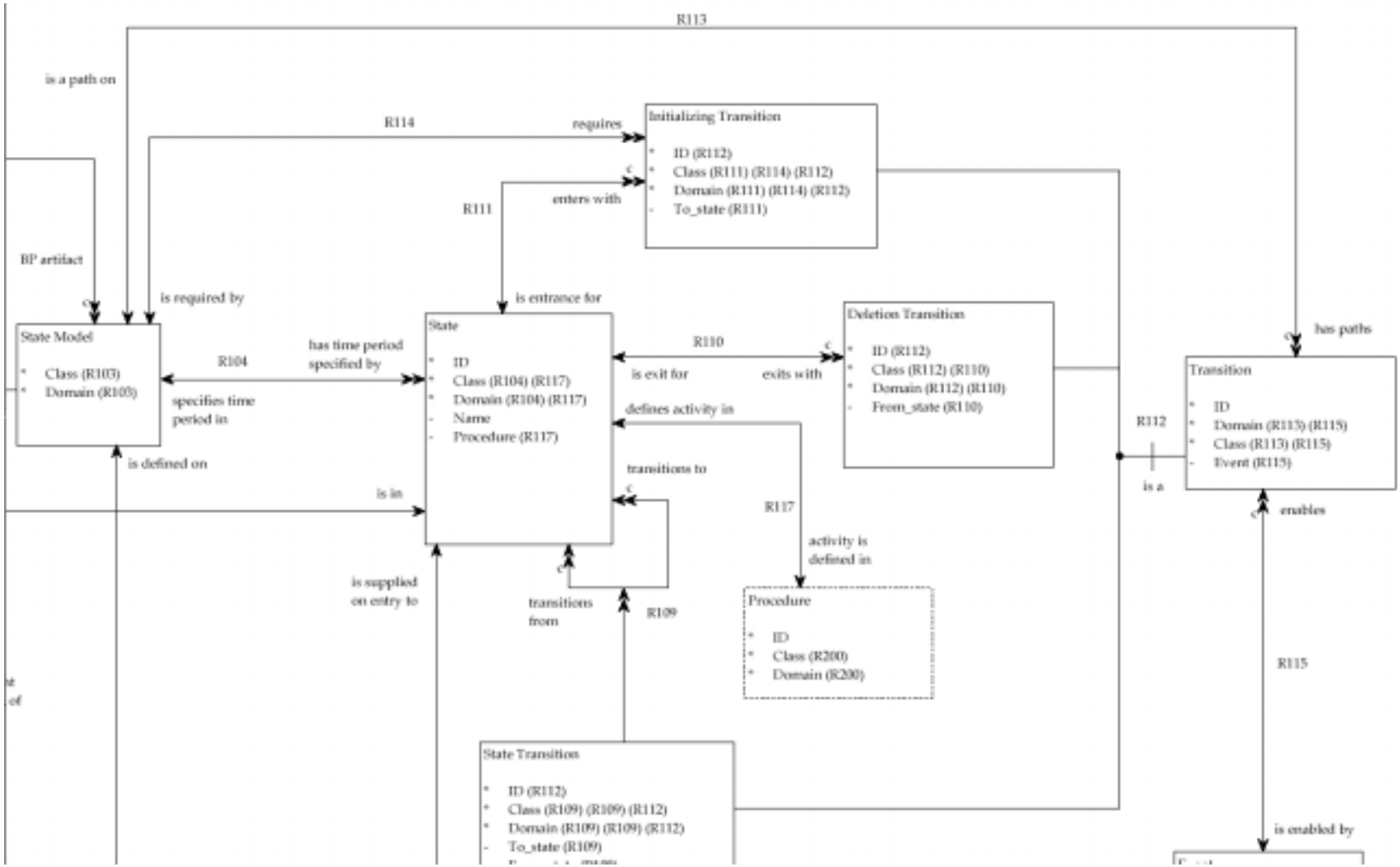
At least one illegal case



Transition types



UML State transitions



A subtle illegal case discovered

R230 - EXITS WITH

State Machine EXISTS WITH 0..* Deletion Transition
Deletion Transition IS EXIT ON 1 State Machine

A final pseudostate is optional in a UML or Executable UML State Machine. A Class Based State Machine, by definition, always exists. An instance that represents equipment or a specification may persist permanently as well. In each of these cases there may be a final quiescent State where there is no Transition exit but nothing is deleted either. So this does not require a Deletion Transition.

NOTE: You can't have a Deletion Transition on a Class Based State Machine!!!!

Great, what can I do with it?

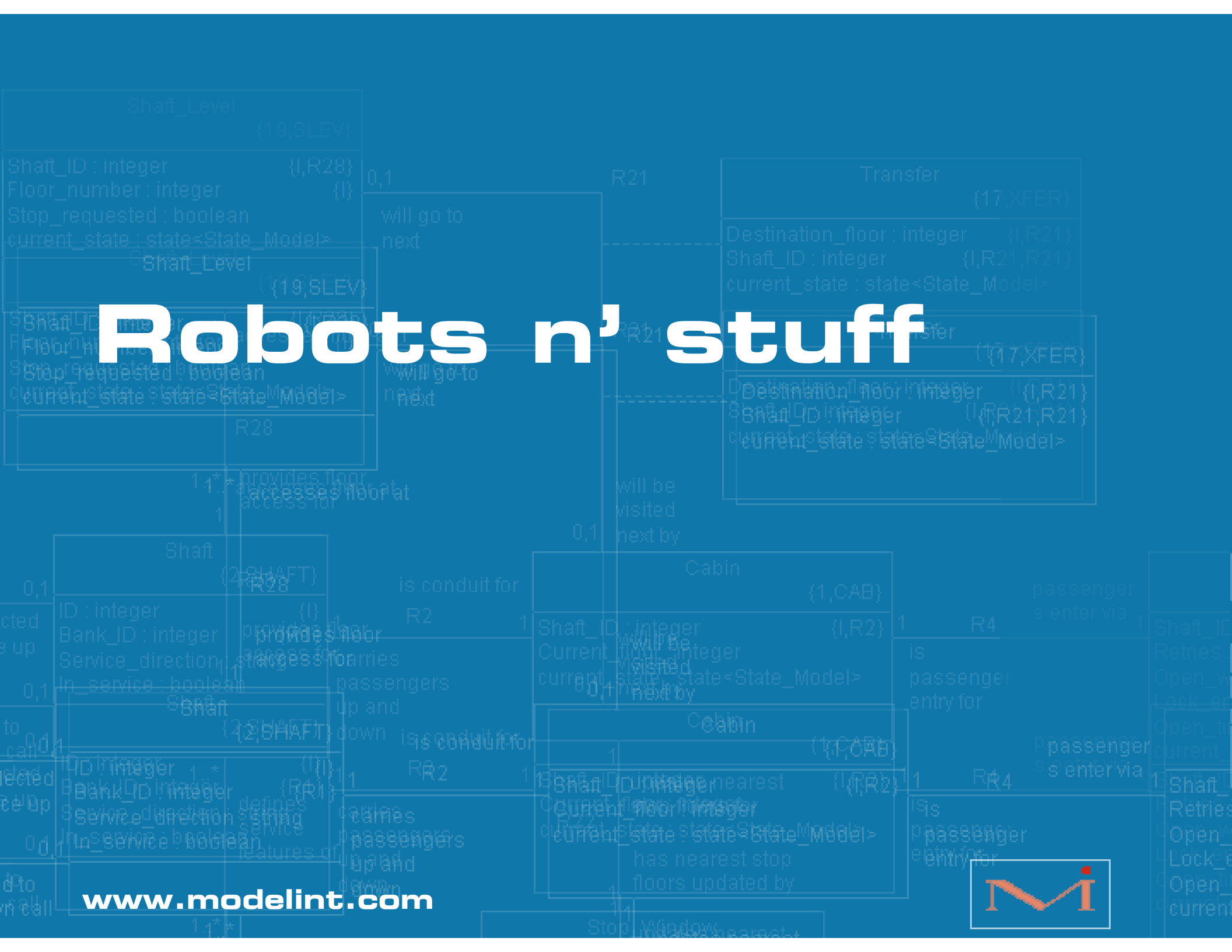
Some ideas

www.modelint.com

Create an XML schema

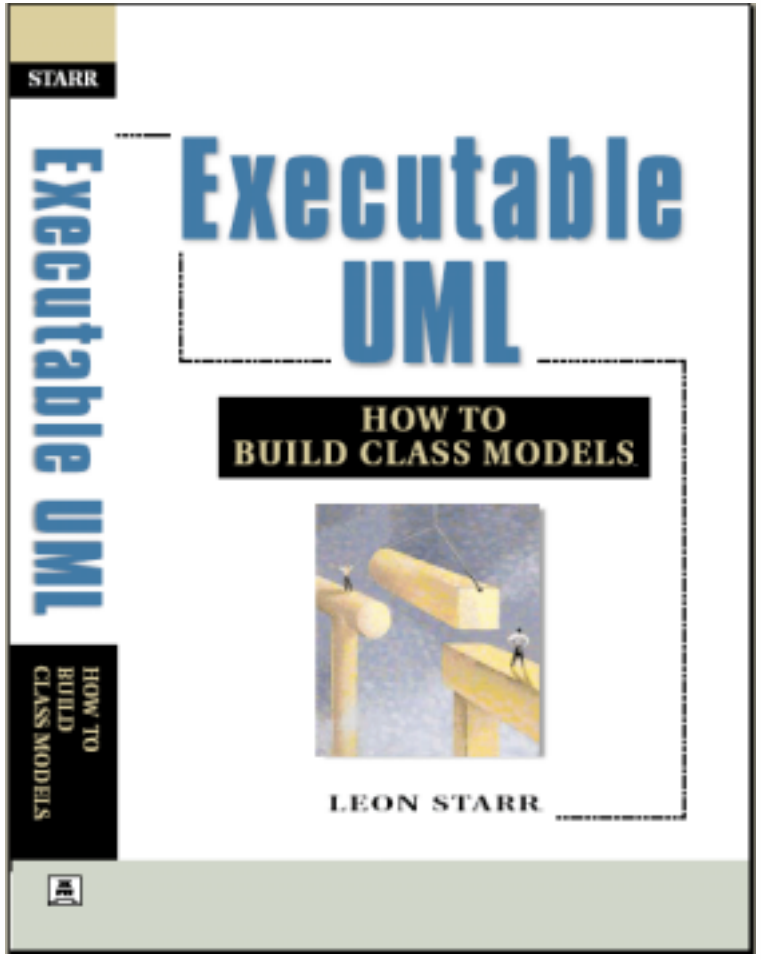
- XML is a convenient external representation.
- Parser's are easily available.
- meta-model.xsd is an XML Schema Definition for the meta-model.
- XML Schemas provide a way of defining an XML syntax in XML notation.
- Newer XML parsers support XML Schemas (and XML Namespaces and other newer XML standards).
- This schema validates against the Xerces-C++ parser (version 1.7.0) available from the Apache Software Foundation. (http://xml.apache.org/xerces-c/)

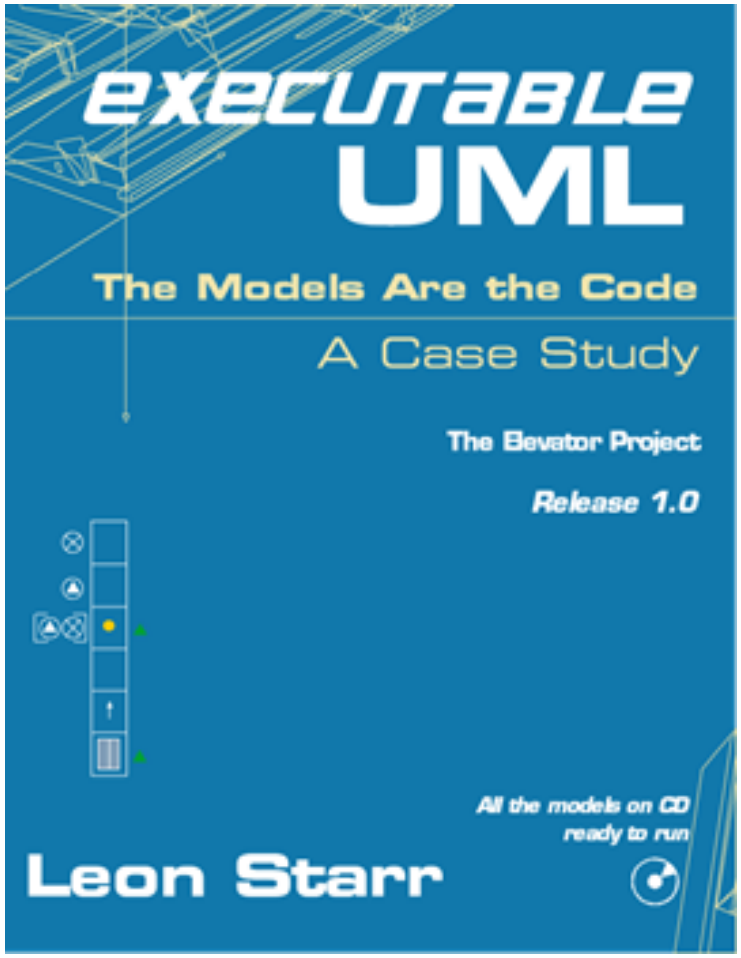
Robots n' stuff



www.modelint.com

Leon's xtUML Books





 Rev 0.1

www.modelint.com

Model Integration Website

- Newsletter – sign up!
- Discussion groups
  - Executable UML Metamodel
  - Elevator Case Study
- Downloads
- Coffee mugs and ant stuff