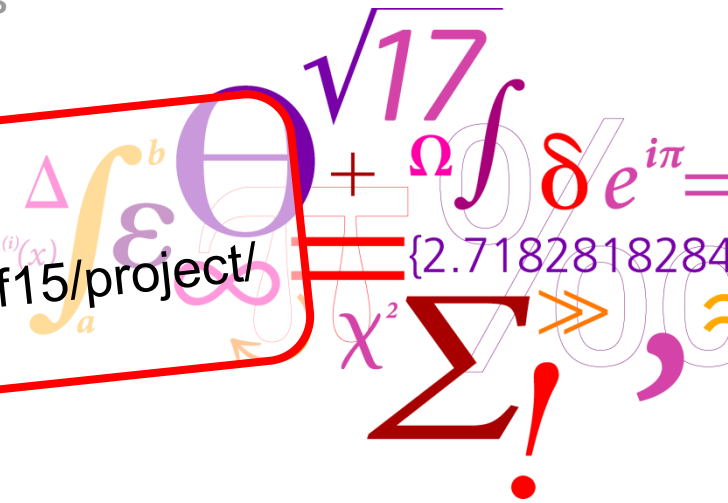
Advanced Topics in Software Engineering (02265)

– The project(s) –

Ekkart Kindler

DTU Informatics
Department of Informatics and Mathematical Modeling

For more details on projects see:
http://www2.compute.dtu.dk/courses/02265/f15/project/



Projects

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

- Part of this course is a project (ca. 2 ECTS points)
- The evaluation of this course is based on the result of this project
- The project is done in groups of 2-4 students
- There are different projects you can chose from
  (and you can suggest own ones, which need to get special approval, though)
- The different projects must use some of the concepts and technologies presented in this course

- The following presentation, gives an overview of the projects (much more details will follow in the project and tutorial slots over the next weeks).

Overview of proposals

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

1. An editor for a network of hierarchical module definitions

   Context: Often, software models are defined in a hierarchical way, defining modules with some ports, which can be used to define modules.

   Task: A graphical editor for views navigating in such a hierarchy (and to interactively ”inline” the substructure of modules in order to see the overall structure of the system (to the depth the user wants to).

   Mechanisms for defininig such modules individually (simple – not necessarily grahical).

   Scope: Focus on technology study for such an editor (→ Oticon masters project could follow).

Overview of proposals

2. A component model for defining formal models

Context: Formal models (here Petri nets) can be used for formally verifying that a system fulfills some requirements. Modelling a large system as a Petri net is tedious. One solution for this problem is to define typical components of some application domain as a Petri net, and then to build the system in this domain by combining these components.

Task: Implement a tool for define some basic Petri net components and an editor for combining these components. From the definition of these components, and model that connects such components, the tool should automatically generate the Petri net of the then do some simple form of analyis or verification of the system (overall Petri net).

Scope: Tool can reuse the ePNK for defining components; focus on editor for combining the defined components, and for generating the overall Petri net and do some simple analysis.

Overview of proposals

3. A GUI modelling language for ECNO Applications

Context: The Event Coordination Notation (ECNO) is a notation for modelling a system including the structure as well as the behaviour of the system. From such ECNO models, the software can be generated fully automatically. The automatically generated GUI is not very nice and very flexible.

Task: Desing a language to define the GUI for an ECNO application, and a GUI engine that implements the GUI as defined (interprted or generated). The language must as a minimum support the features of an existing example (Workers example from ECNO report), which was manually implemented.

Scope: The workers example must be completely covered; and flexible enough for modelling GUIs for other applications. GUI implementation does not need to look fancy (proof of concept) is enough.

Overview of proposals

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

4. Flexible data collection with smart phones

Context: Scientific projects in many areas need to collect personal data for their empirical studies (health, wellbeing, dietary needs), which they would collect via apps for smartphones. Implementing such apps over and over again is a waste of time (in particular, when support of different platforms is needed).

Task: Design simple language for defining surveys in a general and flexible way. Then, an app should be implemented, which equipped with such a definition of a survey conducts the survey. The definition of the survey should allow defining the questions asked and the type of the data to be entered; it should be possible that the questions asked depend on the answer to earlier questions. Also it should be possible to define the frequency of how often certain parts of the survey

Scope: No web backend for now; data stored locally. Focus is clear design of such a language and a proof of concept implementation (supporting all language features); Android only.

Overview of proposals

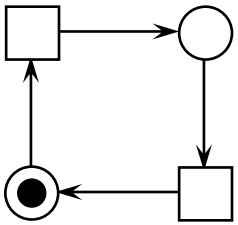
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

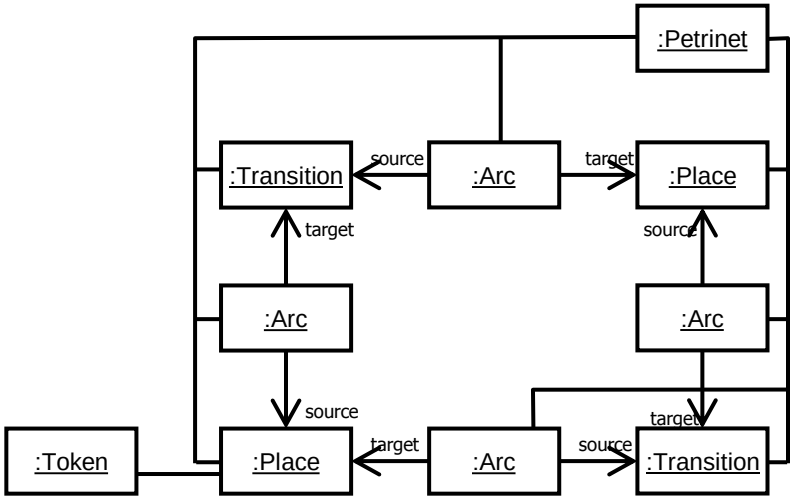
In order to understand the presentation of these projects, this presentation gives an overview of these projects (more details to follow over the next weeks)

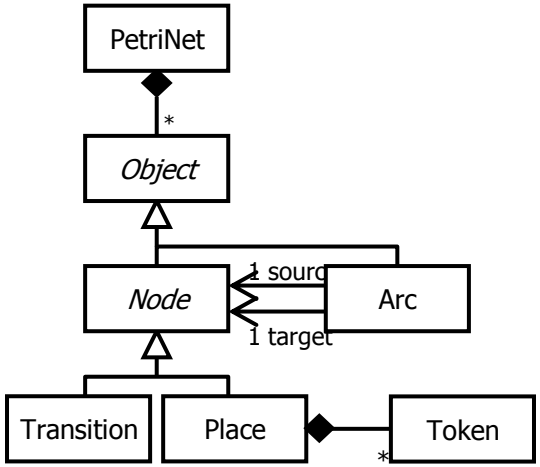
- **ePNK**
- **ECNO**

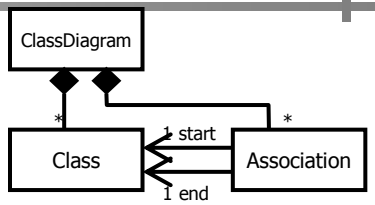
Meta-levels (reminder)

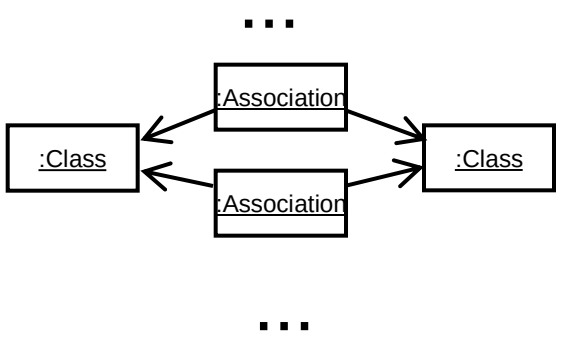
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler





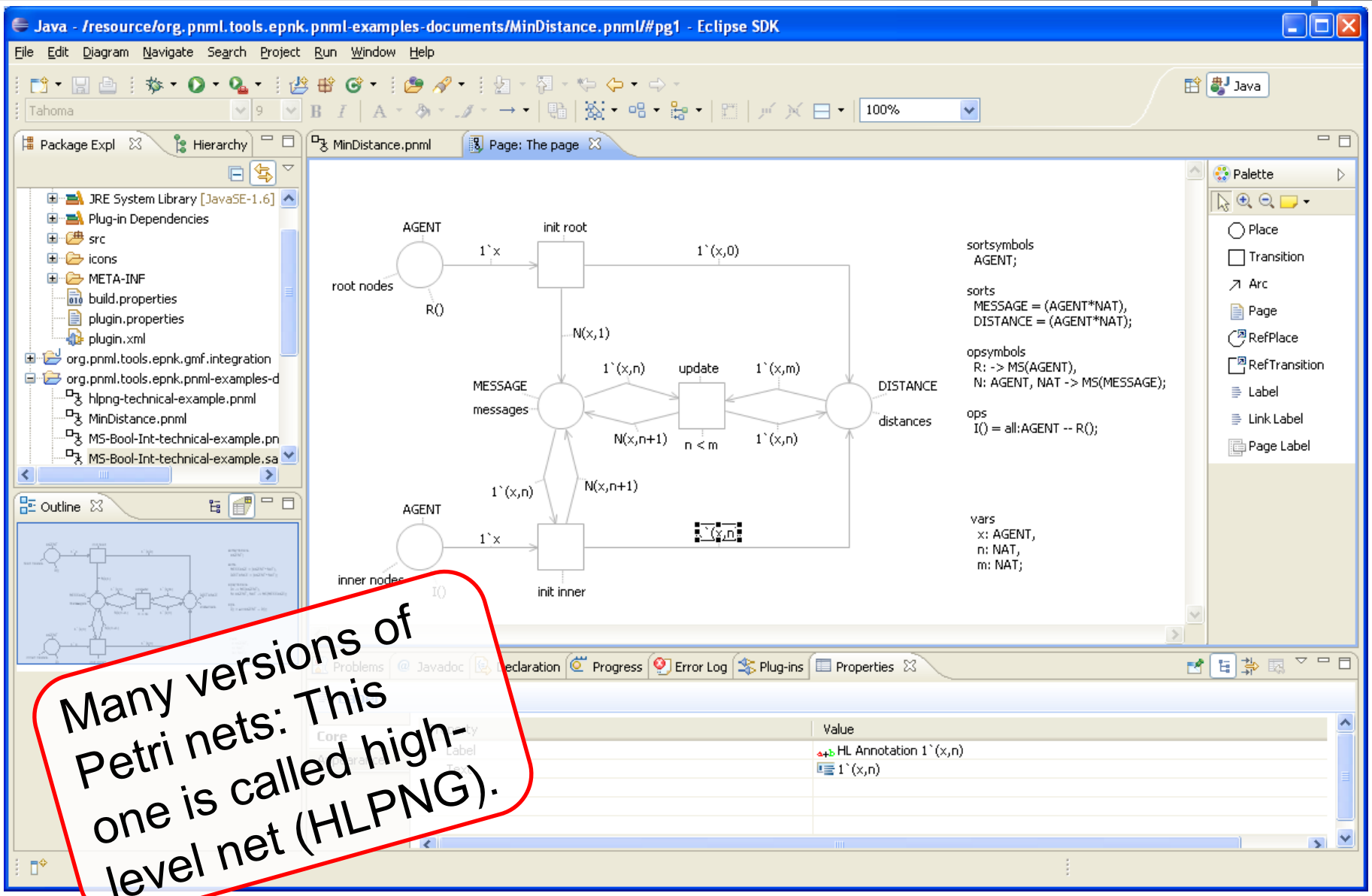






ePNK

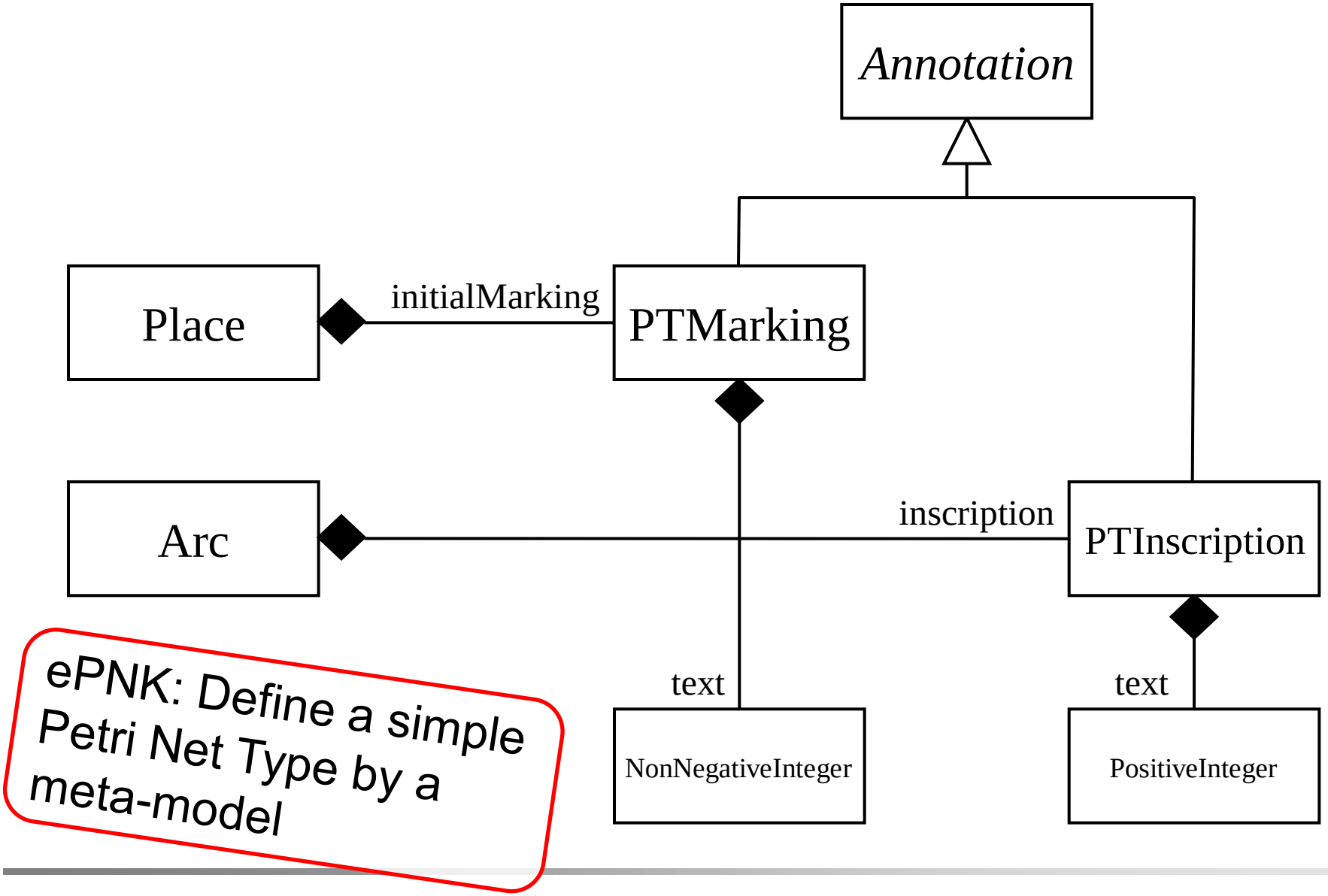
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



Many versions of Petri nets: This one is called high-level net (HLPNG).

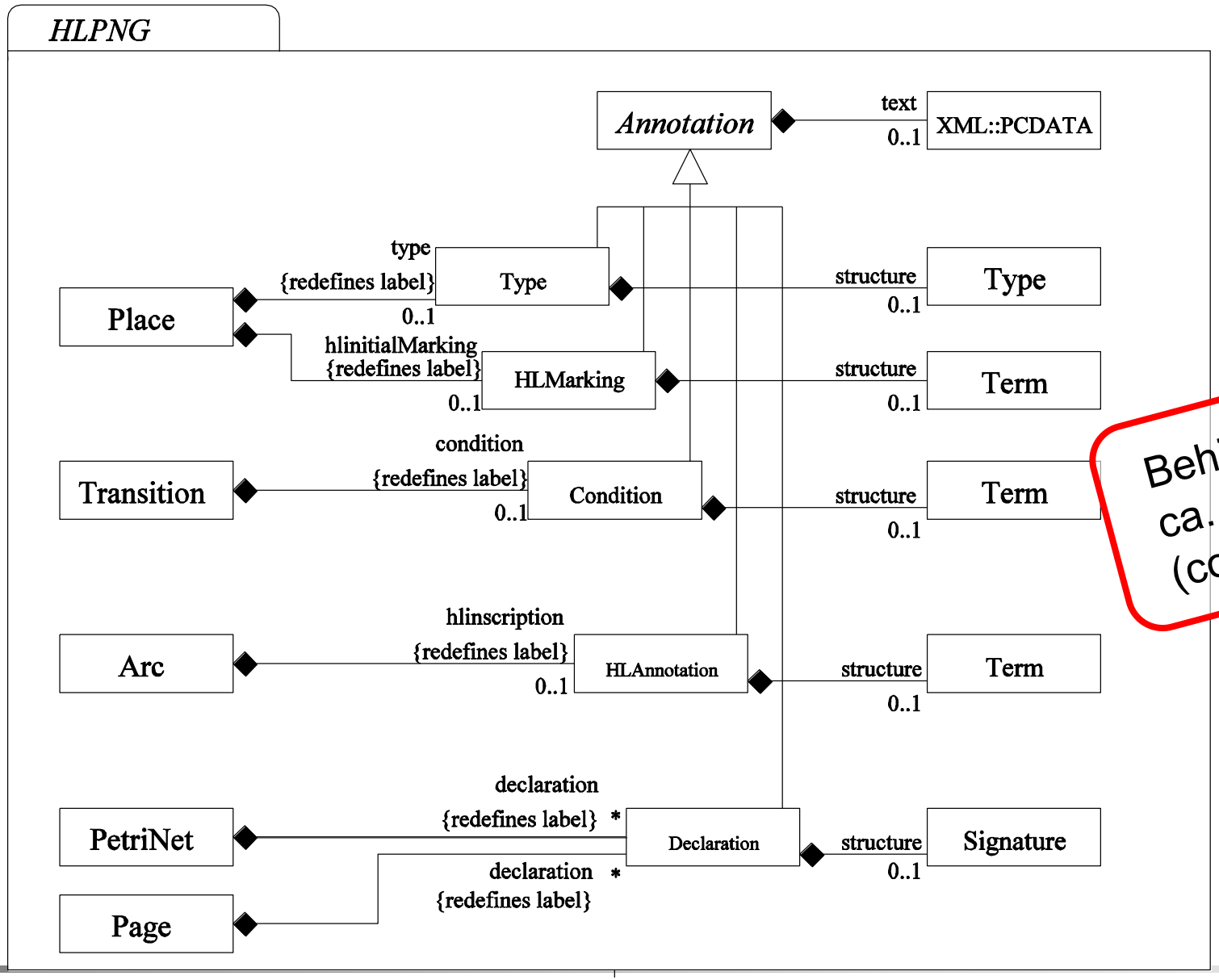
Type Definition: PT-Net

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



Type Definition: HLPNG (outline)

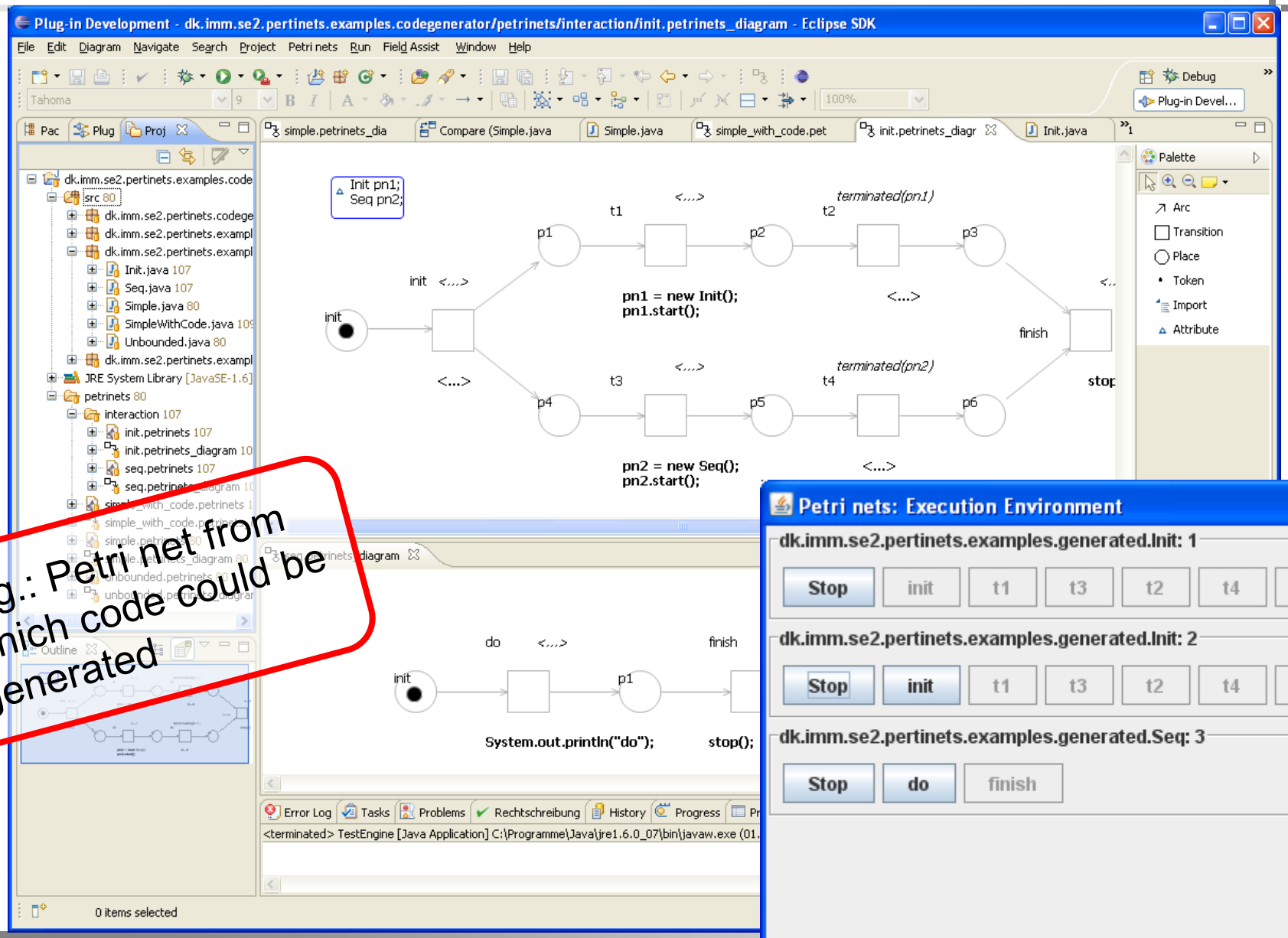
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



Behind this part ca. 80 classes (constructs).

Extended Petri nets

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



E.g.: Petri net from which code could be generated

Overview of proposals

2. A component model for defining formal models

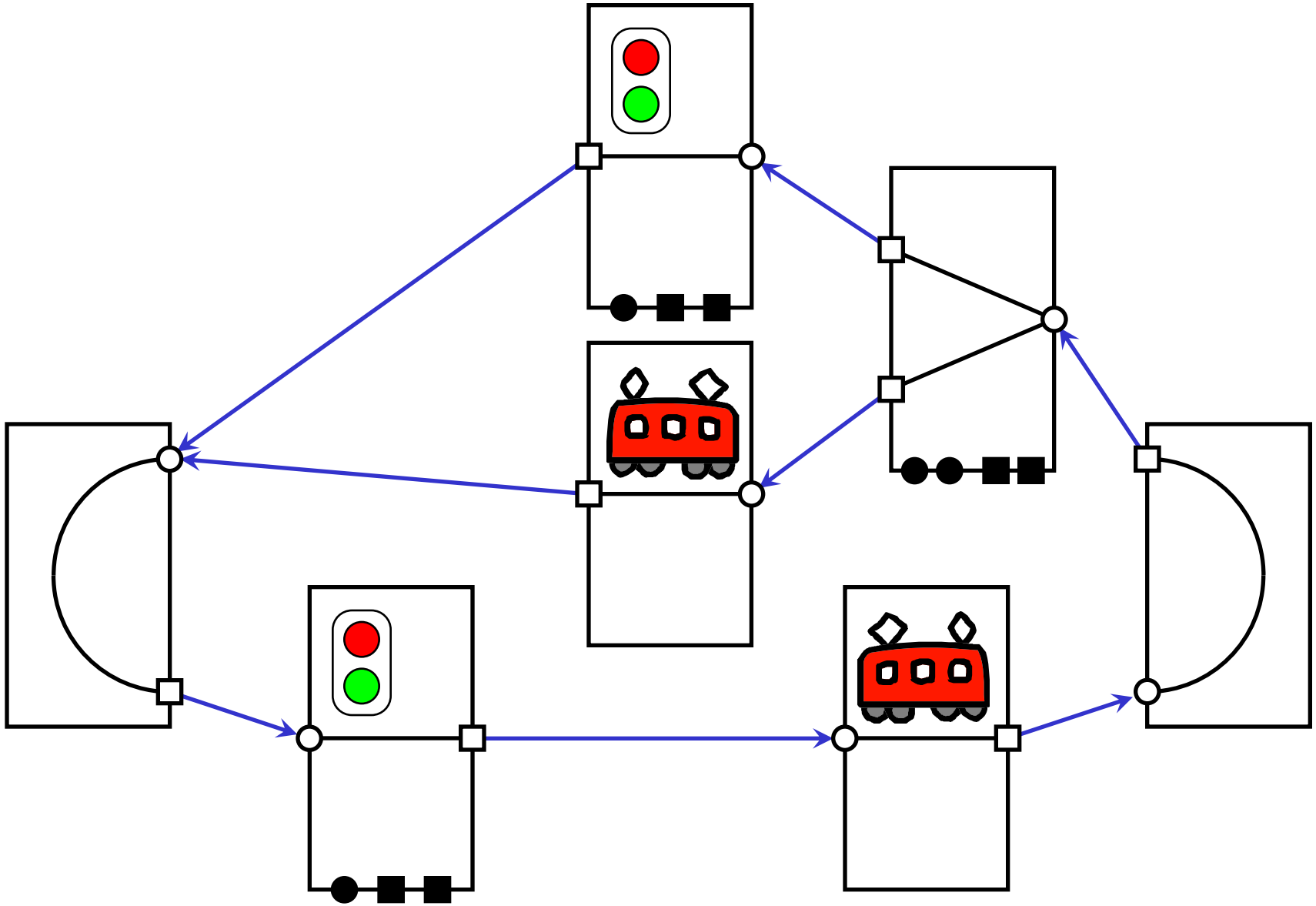
   Context: Formal models (here Petri nets) can be used for formally verifying that a system fulfills some requirements. Modelling a large system as a Petri net is tedious. One solution for this problem is to define typical components of some application domain as a Petri net, and then to build the system in this domain by combining these components.

   Task: Implement a tool for define some basic Petri net components and an editor for combining these components. From the definition of these components, and model that connects such components, the tool should automatically generate the Petri net of the then do some simple form of analyis or verification of the system (overall Petri net).

   Scope: Tool can reuse the ePNK for defining components; focus on editor for combining the defined components, and for generating the overall Petri net and do some simple analysis.

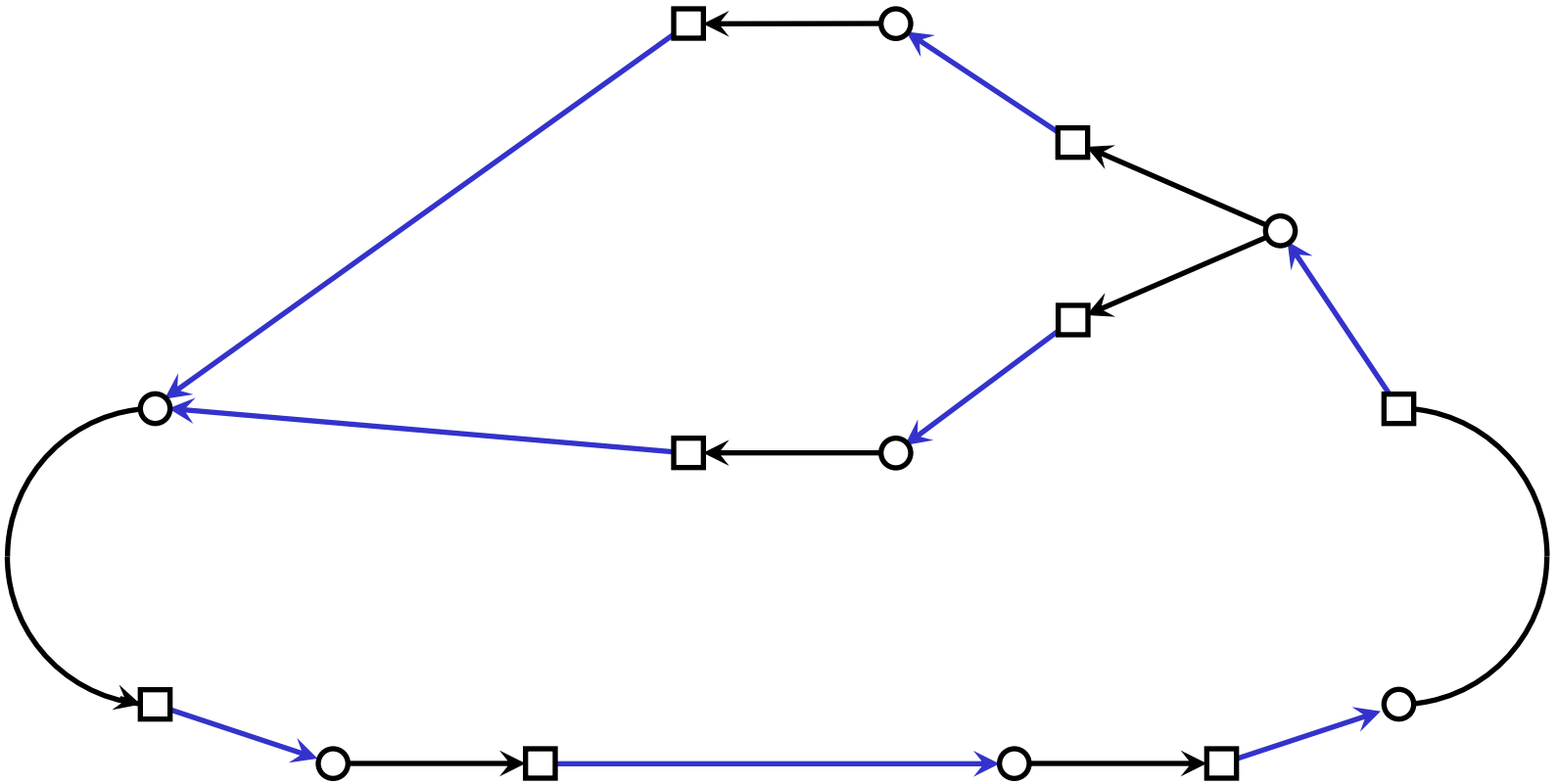
Model built from components

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



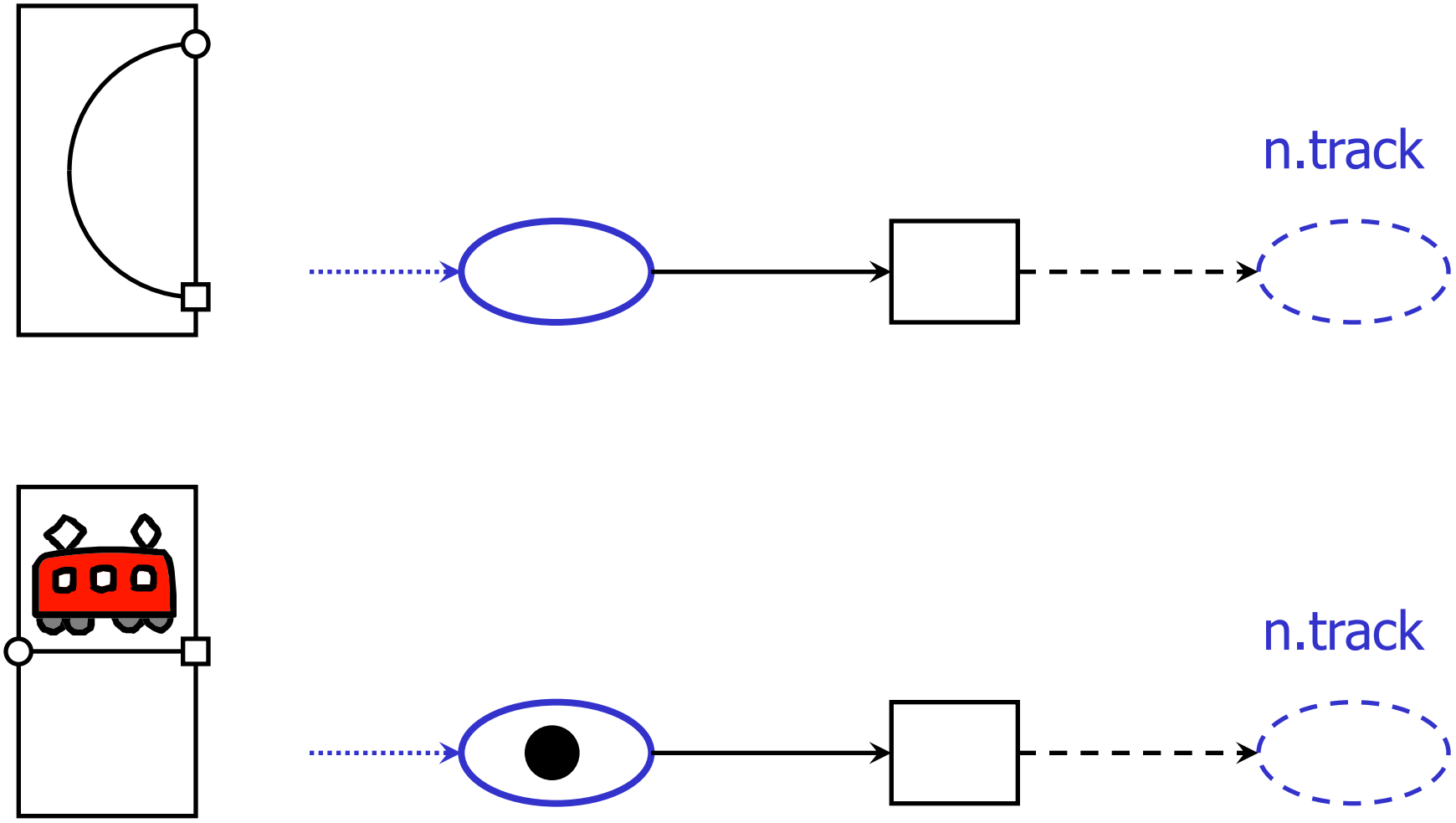
The Petri net behind

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



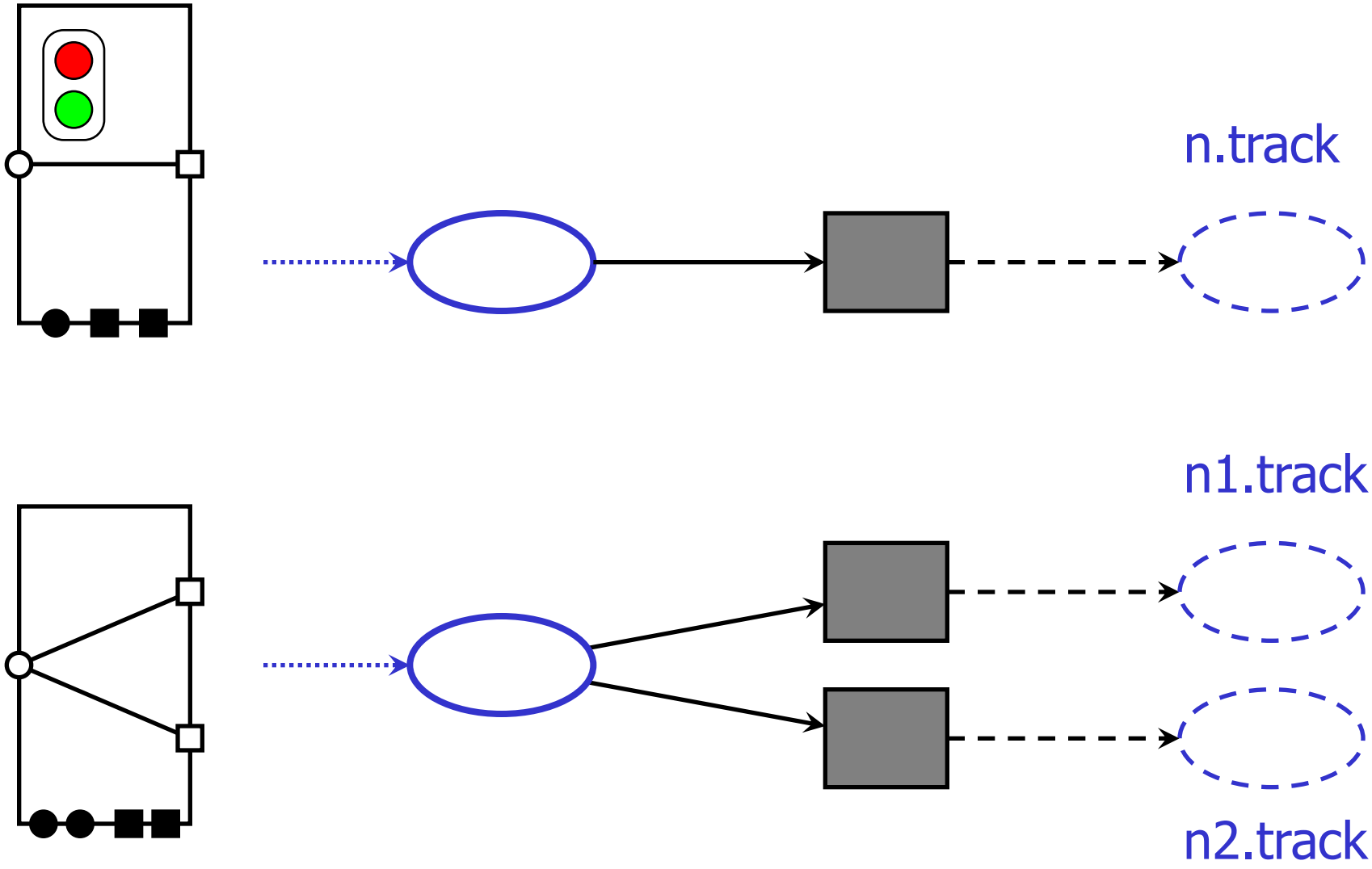
Component definitions

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



Component definitions

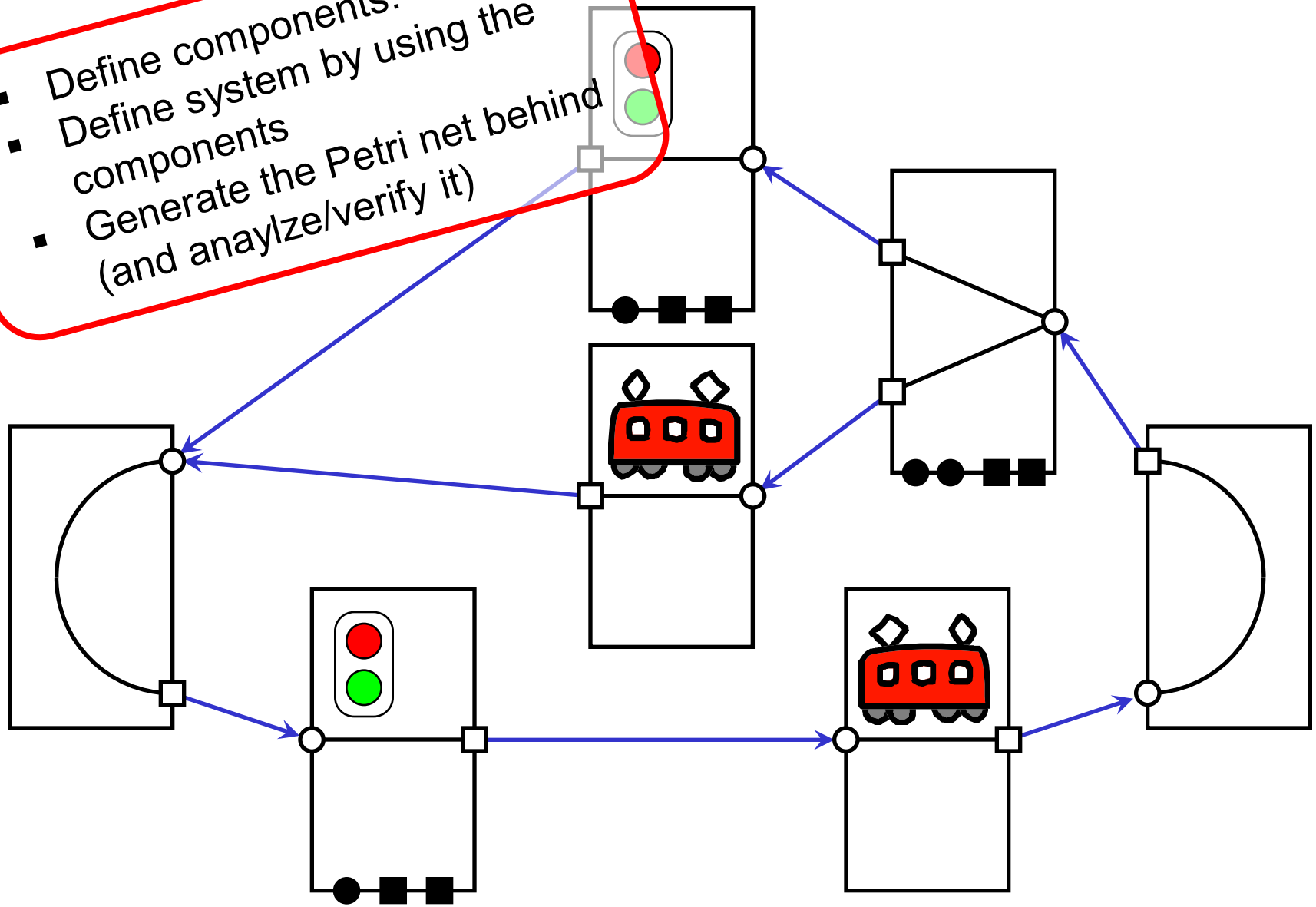
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



Model built from components

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

- Define components.
- Define system by using the components
- Generate the Petri net behind (and anaylze/verify it)



ECNO

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

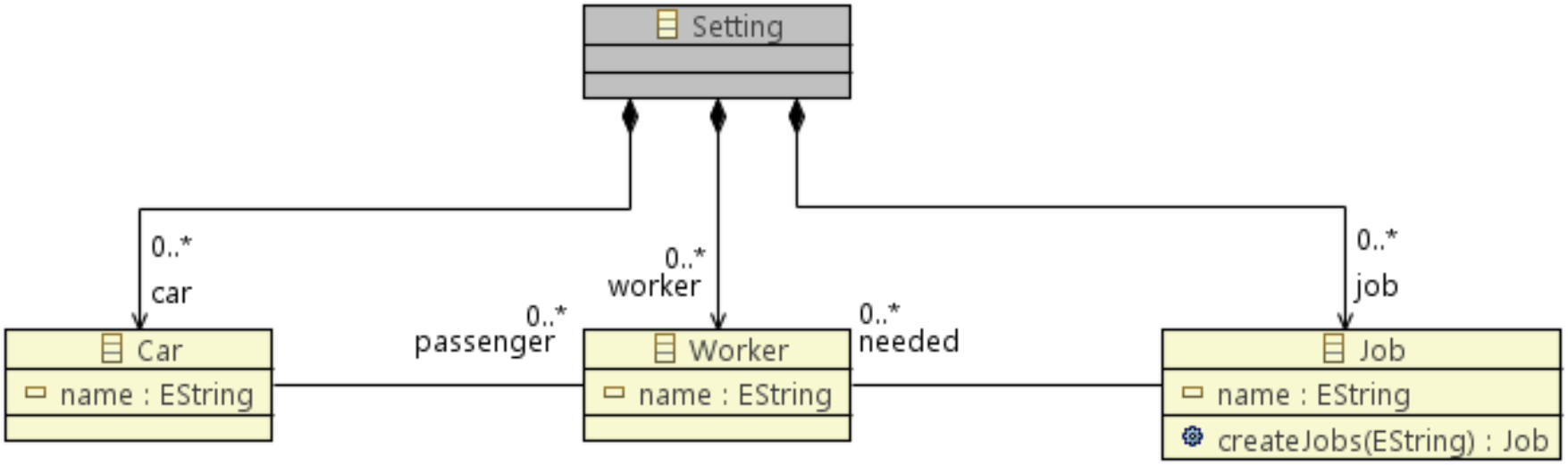
Event Coordination NOtation:

- Modelling the local behaviour of system components (objects)
- Describe how this local behaviour is coordinated with each other → interactions
- On top of OO models or OO implementations

Example: Class diagram

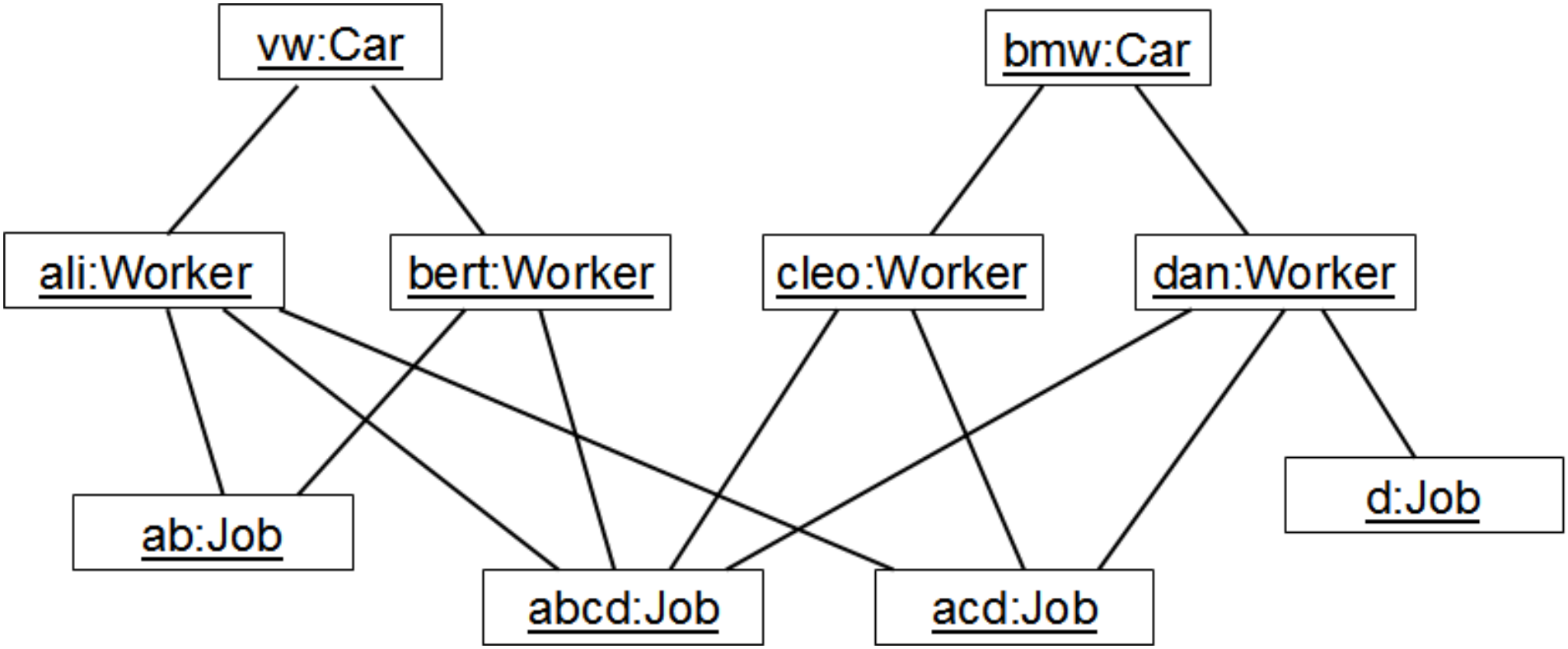
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

See ECNO tech report



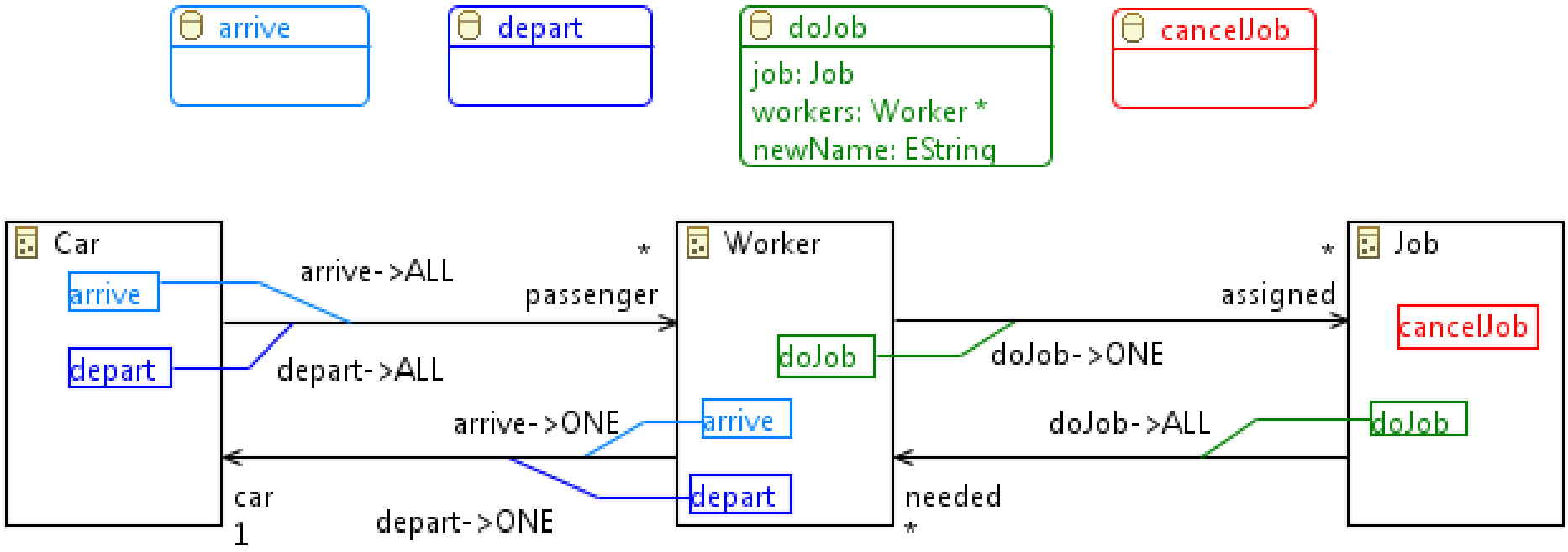
Example: Initial Config

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



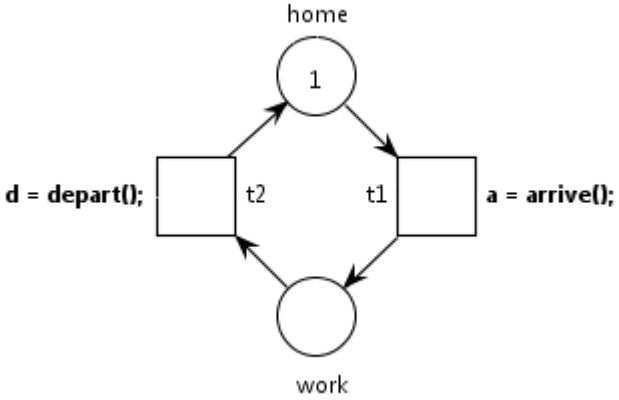
Example: Coordination

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

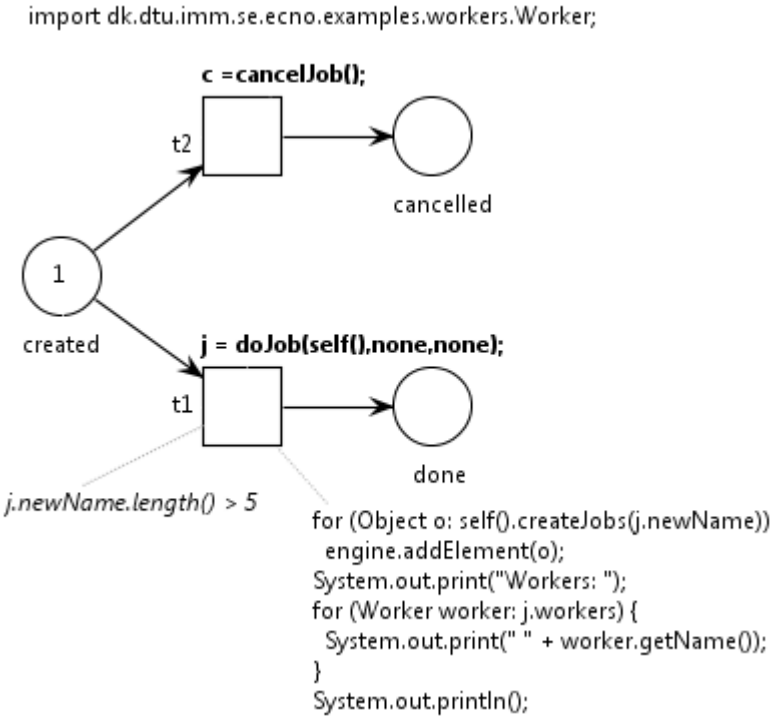


Elements: Live Cycle

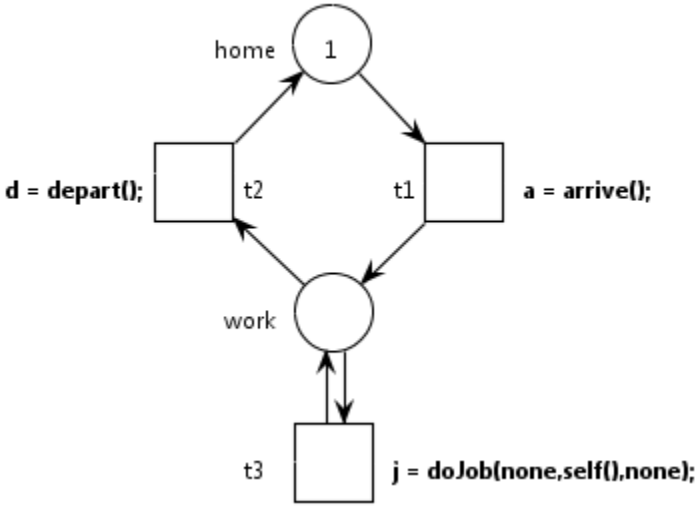
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



Car

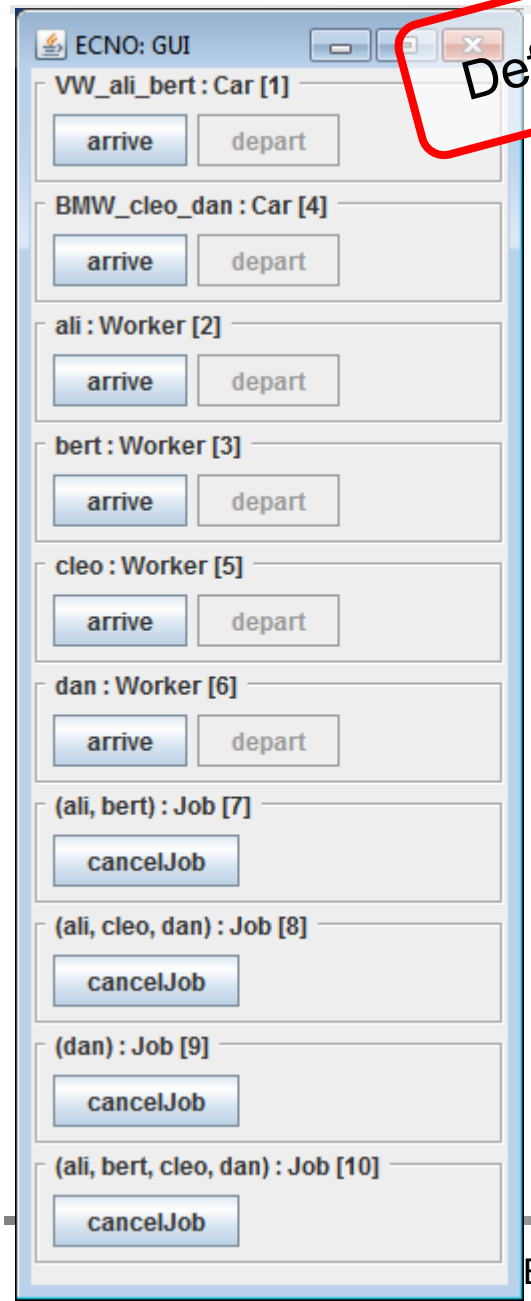


Job

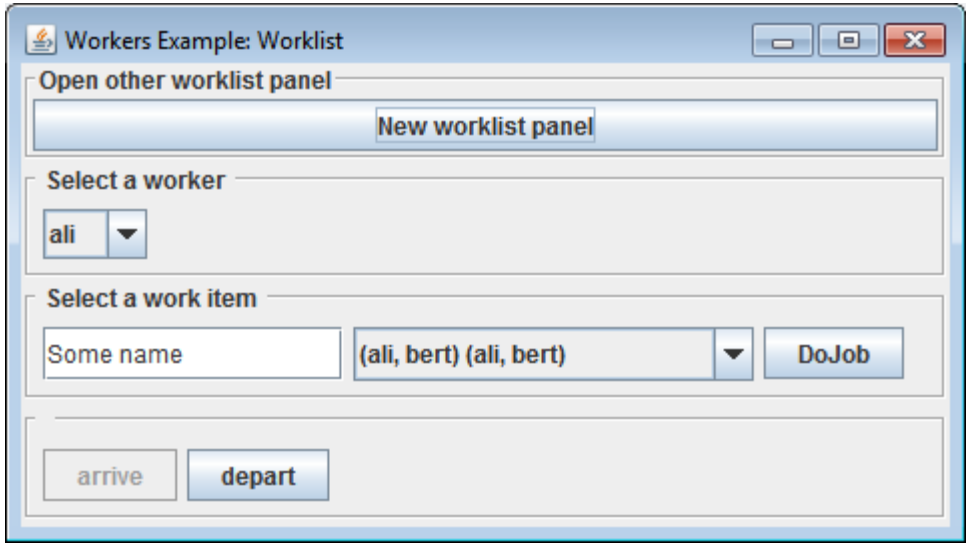


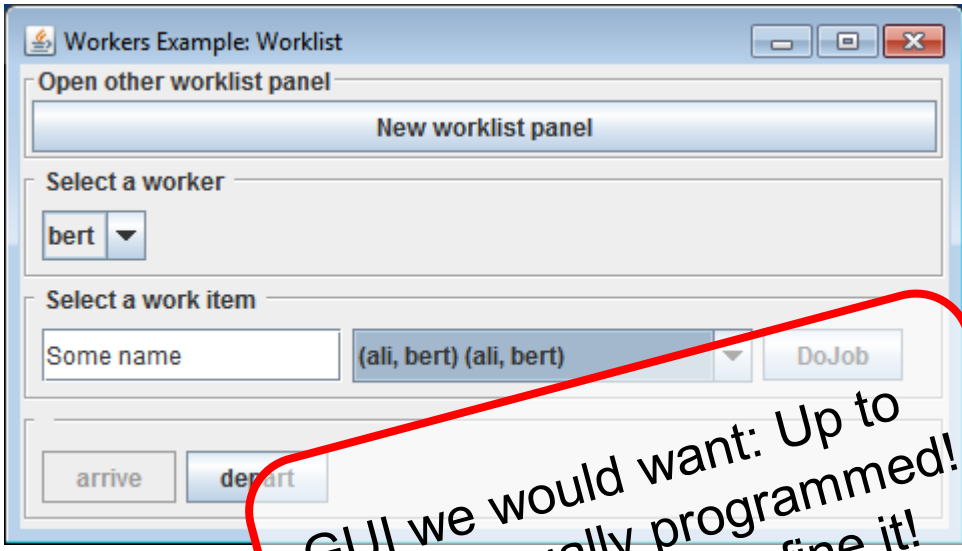
Worker

Example: GUI



Default GUI





GUI we would want: Up to now manually programmed! We would like to define it!

ECNO related project

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

3. A GUI modelling language for ECNO Applications

Context: The Event Coordination Notation (ECNO) is a notation for modelling a system including the structure as well as the behaviour of the system. From such ECNO models, the software can be generated fully automatically. The automatically generated GUI is not very nice and very flexible.

Task: Desing a language to define the GUI for an ECNO application, and a GUI engine that implements the GUI as defined (interprted or generated). The language must as a minimum support the features of an existing example (Workers example from ECNO report), which was manually implemented.

Scope: The workers example must be completely covered; and flexible enough for modelling GUIs for other applications. GUI implementation does not need to look fancy (proof of concept) is enough.

Overview of proposals

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

1. An editor for a network of hierarchical module definitions

   Context: Often, software models are defined in a hierarchical way, defining modules with some ports, which can be used to define modules.

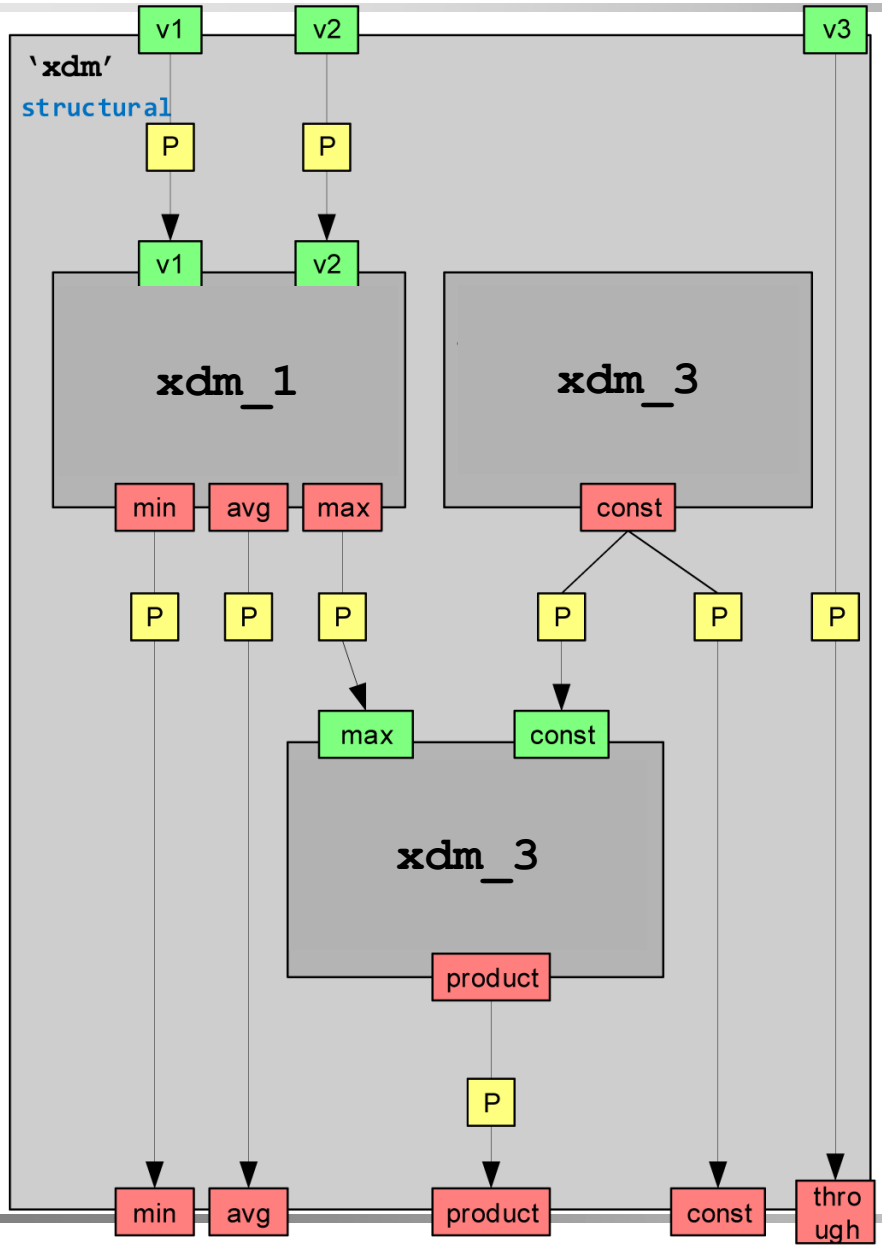
   Task: A graphical editor for views navigating in such a hierarchy (and to interactively ”inline” the substructure of modules in order to see the overall structure of the system (to the depth the user wants to).

   Mechanisms for defininig such modules individually (simple – not necessarily grahical).

   Scope: Focus on technology study for such an editor (→ Oticon masters project could follow).

Modul Definition: xdm

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

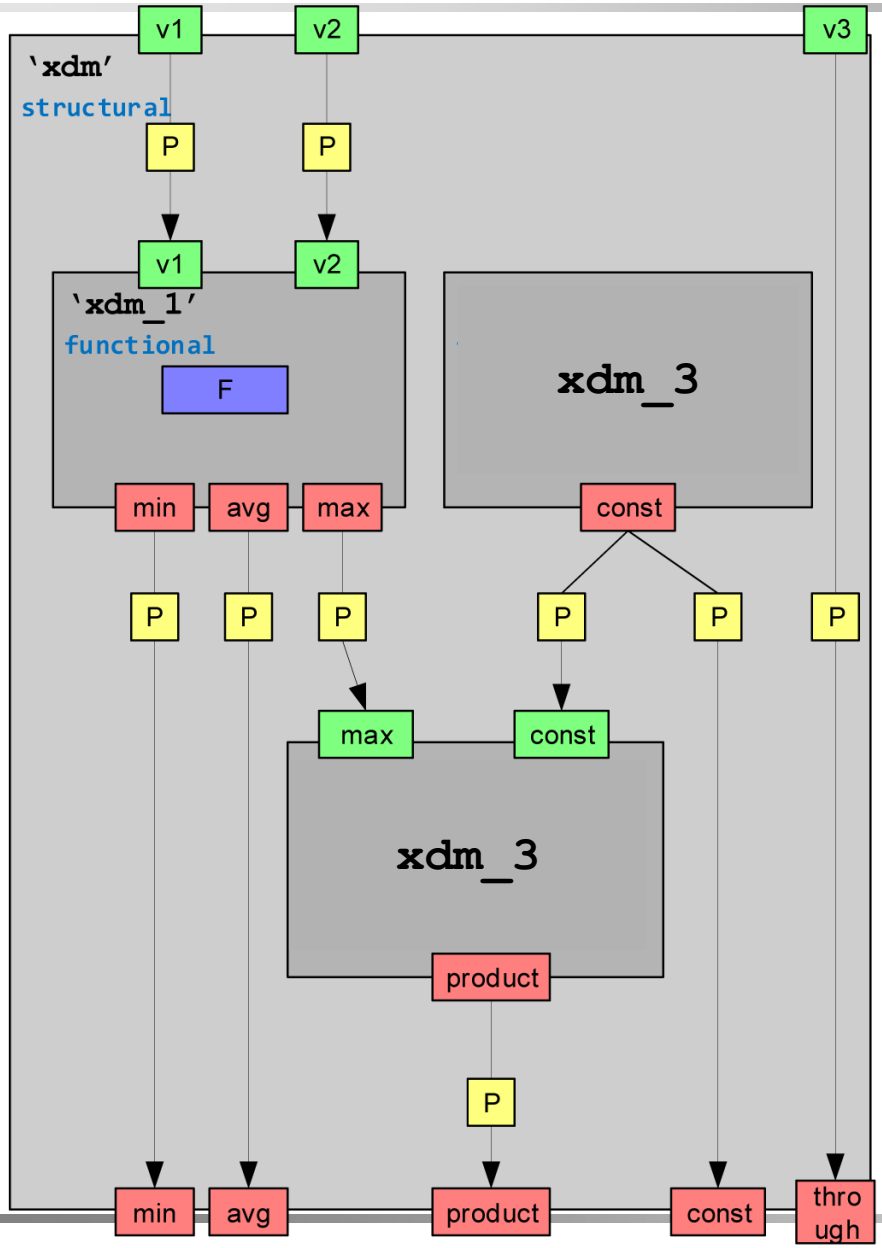


Example from Oticon!

Automatic layout!

Modul Definition: xdm

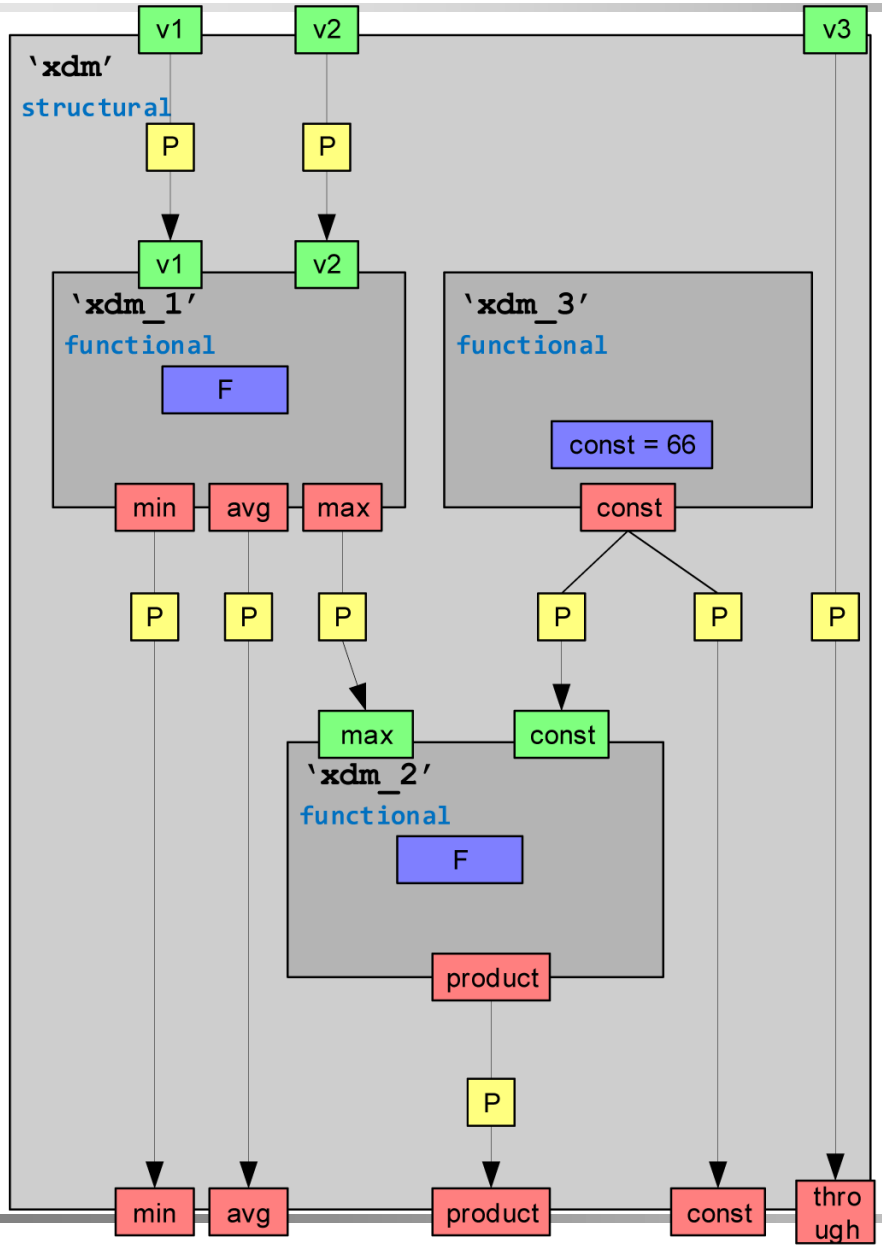
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



Zooming into definitions
(interactively by user)

Modul Definition: xdm

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



Zooming into definitions (interactively by user)

Overview of proposals

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

4. Flexible data collection with smart phones

Context: Scientific projects in many areas need to collect personal data for their empirical studies (health, wellbeing, dietary needs), which they would collect via apps for smartphones. Implementing such apps over and over again is a waste of time (in particular, when support of different platforms is needed).

Task: Design simple language for defining surveys in a general and flexible way. Then, an app should be implemented, which equipped with such a definition of a survey conducts the survey. The definition of the survey should allow defining the questions asked and the type of the data to be entered; it should be possible that the questions asked depend on the answer to earlier questions. Also it should be possible to define the frequency of how often certain parts of the survey

Scope: No web backend for now; data stored locally. Focus is clear design of such a language and a proof of concept implementation (supporting all language features); Android only.

Rough schedule for projects:

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

Week 7 (Feb. 11): Forming of groups
(via Campus Net)

Week 8 (Feb. 18): Every group decides on their project
(via Campus Net)

Week 9 (Feb. 25): Every group writes 1-2 pages of a project description
(via Campus Net)

Rough schedule for projects:

Week 11 (March 11).: Every group submits a project definition (ca. 5 pages)
(via Campus Net)

[Week 17 (April 22): Prototype (optional) ???
(via Campus Net)]

Week 19 (May 6): Every group presents their project (Maybe we need some additional slot for that)

Week 23, June 2 (last day of examination period): Submission (via Campus Net) of project result: Code/PlugIns and example; documentation of the design, implementation, and the use of the product/example (report)