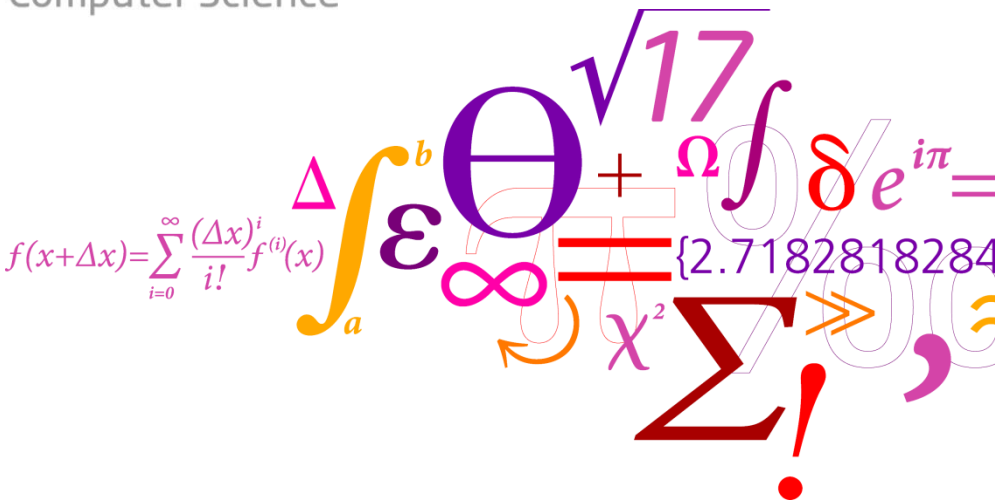
Model-based Software Engineering
(02341, spring 2017)

Ekkart Kindler

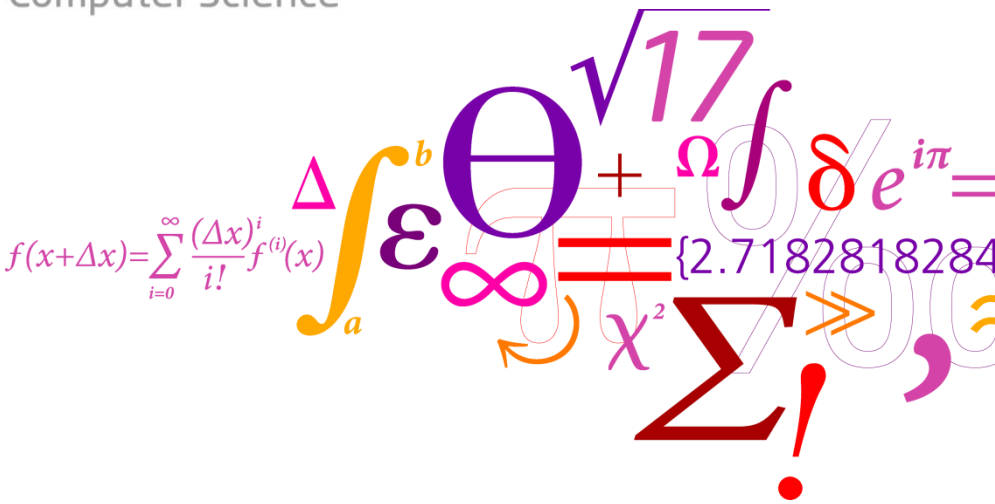
DTU Compute
Department of Applied Mathematics and Computer Science



The Project

A YAWL editor and simulator

DTU Compute
Department of Applied Mathematics and Computer Science



Course Structure

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

The course consists of three main parts:

- Lectures:
  Understand, ideas, principles, concepts, and technologies

- Tutorials:
  Acquire experience with technologies and tools in a small and controlled setting

- Project:
  Demonstrate that you can use ideas, concepts, technologies, and tools in (slightly) larger context.

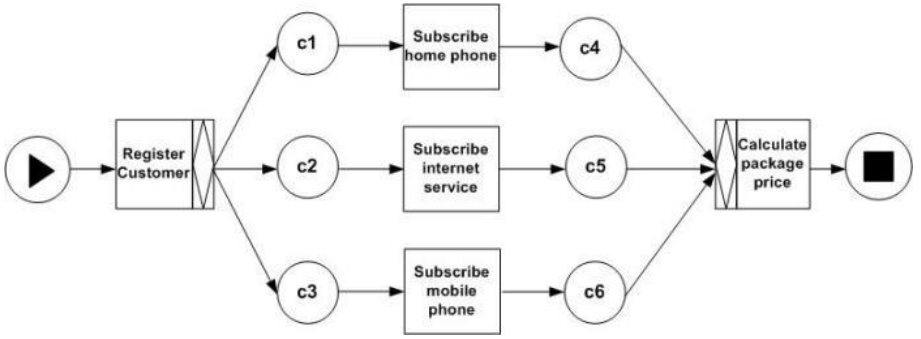
Lectures and tutorials are necessary prerequisites for properly doing the project!

Evaluation will be based on project: submitted software, written report, and the final presentation

YAWL

See lecture 4 for references and more details!

YAWL (Yet another Workflow Language)
a graphical notation for modelling and enacting business processes



[from: http://www.yawlfoundation.org/pages/research/orjoin.html]

Task

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Implement a graphical editor for a subset* of YAWL and a simulator on top of that graphical editor visualizing the behaviour of a YAWL process
- This editor and simulator must be implemented based on the ePNK**

*) see scope for the specific subset of YAWL that your tool must supported

**) the ePNK will be discussed in tutorials 5-8 (see idea on next slides)

ePNK

”The ePNK is a platform for Petri net tools based on the PNML transfer format. Its main idea is to provide generic Petri net types, which can be easily plugged into it, and to provide a simple generic GMF editor, which can be used for graphically editing nets of any plugged in type.” [ePNK Homepage]

ePNK

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

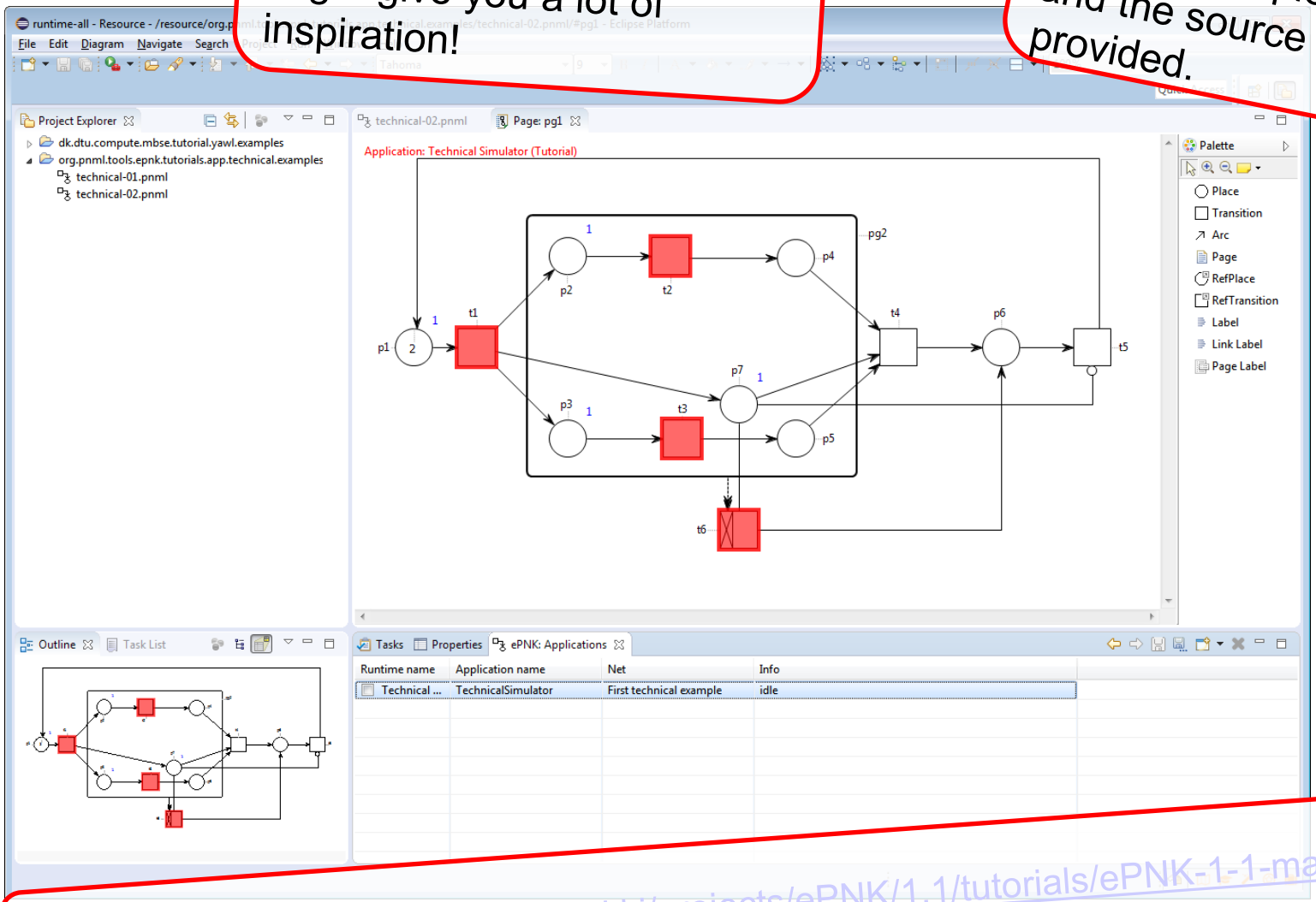
- A new Petri net type is defined by an EMF model, which, basically, can be plugged in to the ePNK (t5)
- ePNK provides a simple graphical editor, which can be customized (by programming) to feature specific graphical representations of the new net type (t6)
- Additional consistency conditions on the Petri net type can be plugged in too, as OCL or Java constraints (t7)
- Applications like simulators with graphical feedback can also be plugged in to the ePNK for some net types (t8)

See http://www2.compute.dtu.dk/~ekki/projects/ePNK/1.1/tutorials/ePNK-1-1-manual-draft.pdf

ePNK

Looking into the source code and models of these projects might give you a lot of inspiration!

Screenshot of a simulator for a version of Petri nets, for which a complete tutorial and the source code is provided.



See http://www2.compute.dtu.dk/~ekki/projects/ePNK/1.1/tutorials/ePNK-1-1-manual-draft.pdf

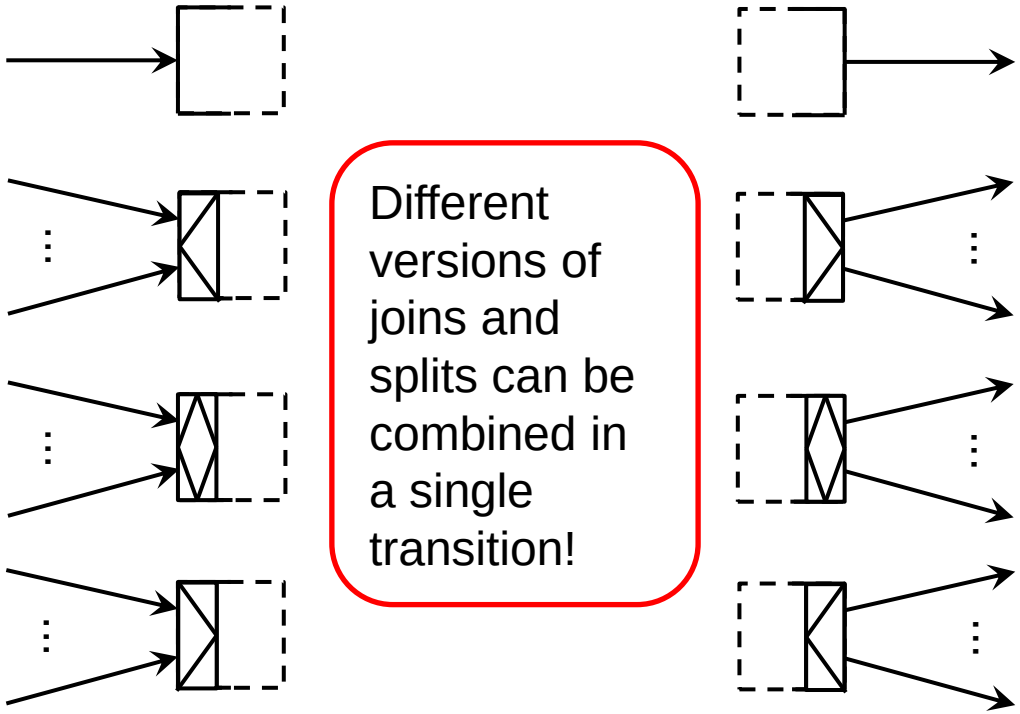
Scope

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

The tool must support the following YAWL features

- Start and end conditions (exactly on of each kind)

- Transition input/output: single, AND, XOR, OR

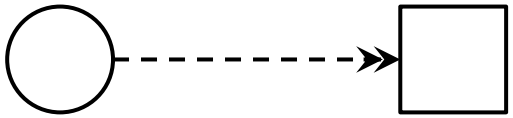


Scope (cntd.)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

The tool must support the following YAWL features (cntd.)

- Reset arcs



All tokens removed, when transition fires (does not require a token on this place)

- Support the page concept of ePNK (”flattening” of nets discussed in lectures/tutorial)

Data and organisation concepts do **not** need to be supported

Scope (cntd.)

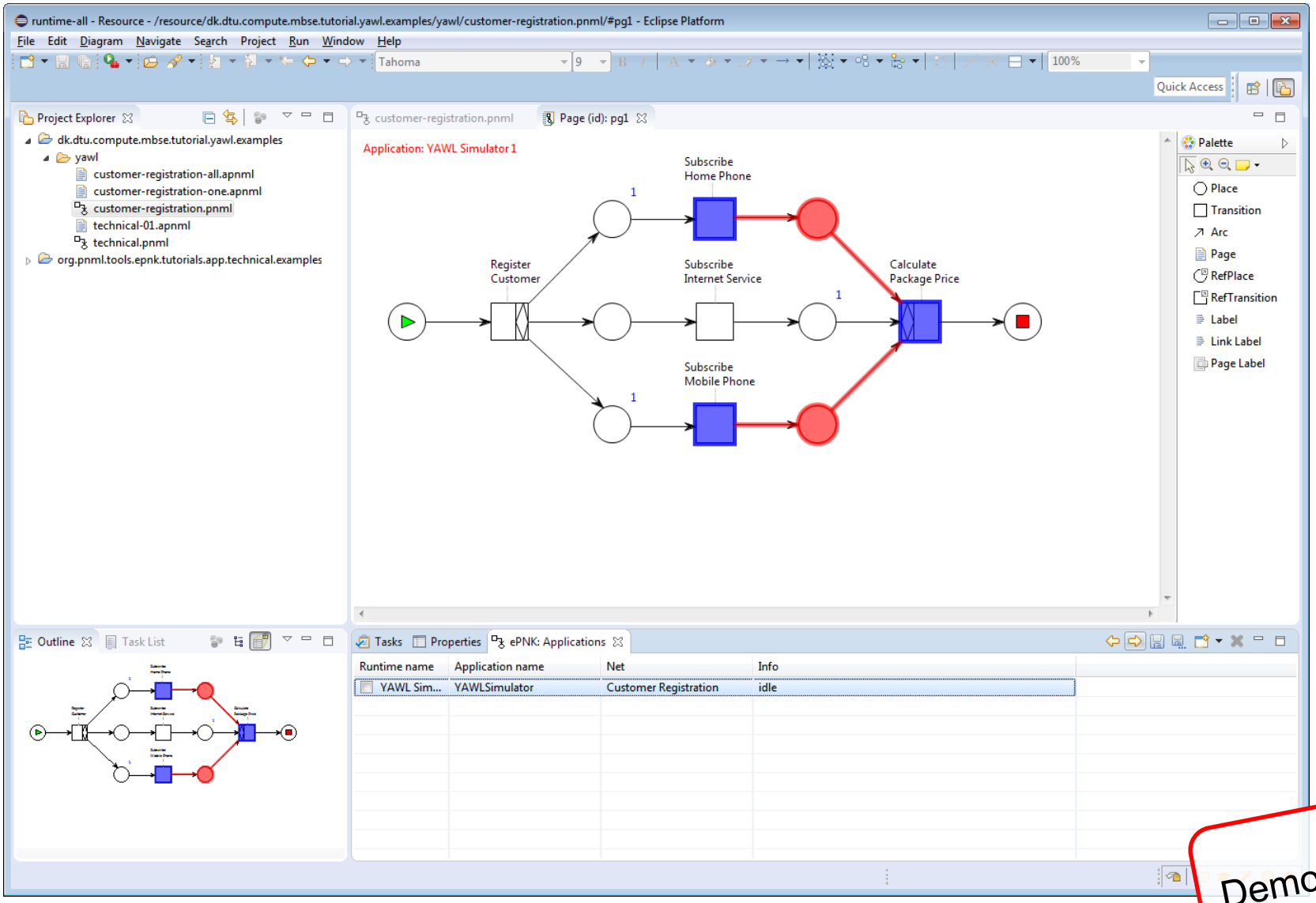
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

The simulator must

- Provide graphical feedback on the current state of the process (marking)
- Visually indicate the enabled transitions/actions, and allow the user to select a transition to fire
- For XOR-joins and -splits, allow the user to select from which place a token should be consumed and to which place the token should be produced
- For OR-splits allow the user to chose to which places a token should be produced
- For OR-joins indicate (give a warning) that on some unmarked input places a token might still arrive (and graphically indicate from where)

Result: Example

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



Demo!

Submission

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- The software as source code including models
  (exported Eclipse plugin projects)
- At least two YAWL examples (modelling reasonable **business processes**)
- A report documenting your software (underlying models and design), including (but not limited to):
  - Intro and overview of your project and ePNK extension
  - Domain models (EMF models) with detailed discussion
  - Discussion of how your extension works together with the ePNK (software models, interfaces, interactions)
  - A brief handbook explaining the use of all features of your software (for an end user) using your examples (standard features of the ePNK as documented in the ePNK handbook do not need to be explained in detail)

Project

All submissions as groups via CampusNet (detailed instructions later)

Submissions / presentations:

- **February 10**: **Group members** of each group (4 to 5)
- **March 10**: **Project definition**
- **April 18** (optional) : preliminary **submission of project** (including software and written report)

- **May 15**: **Final submission of project** (including software and report)
- **May 24/29**: **Presentation of final submission**
  - **Presentation of result by groups**
  - **Questions and answers**

If these dates are not suitable for you, speak up right away!