


www.omilab.org

Execute 

OMiLAB Training Module 2

Fundamental Conceptual Modelling Languages
using Bee-Up
Theoretical Foundations

Agenda

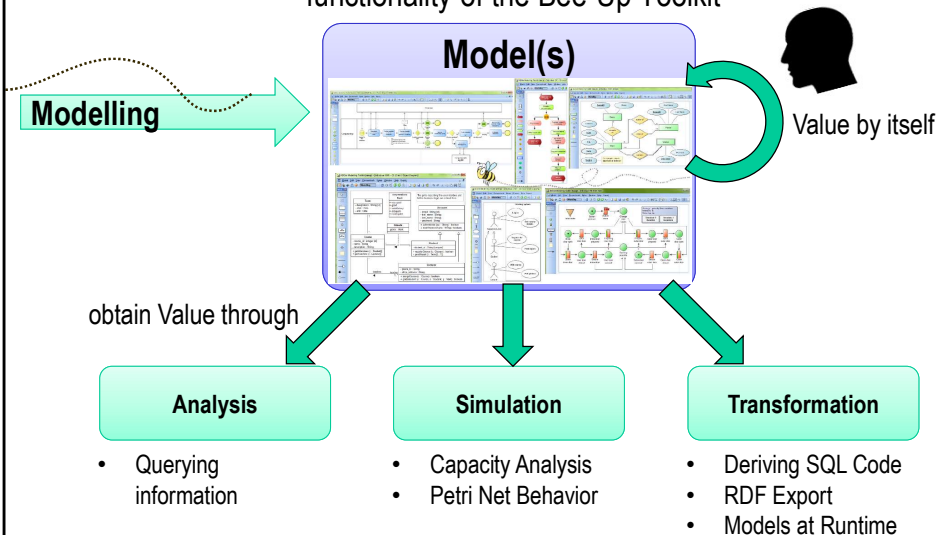
1. Learning Goals
2. Motivation for Models
3. What is Bee-Up
4. Why Bee-Up
5. Modelling Languages in Bee-Up
6. Modelling in Bee-Up
7. Lessons Learned

Learning Goals

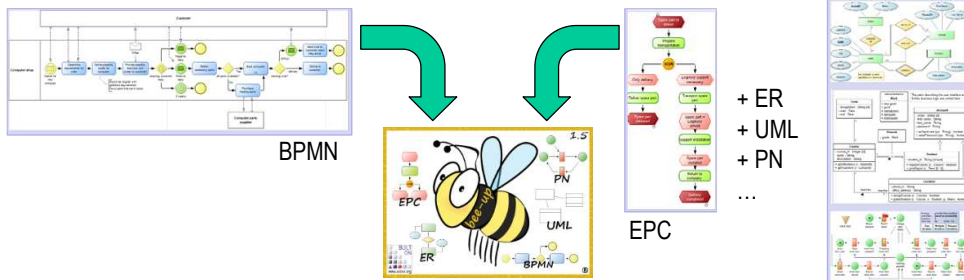
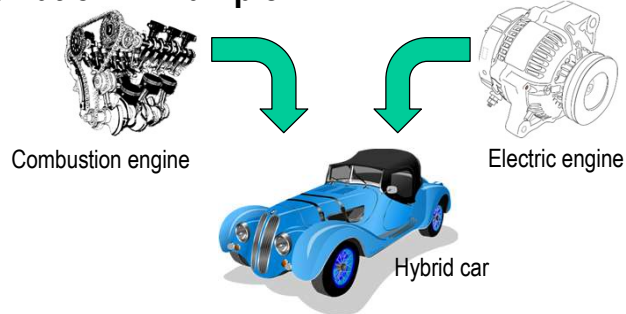
- The use of conceptual models to capture a system's characteristics, while abstracting away irrelevant details, provides benefits for their creation, analysis and adaptation. Independent of the domain the models are applied in, there are recurring patterns which have led to the emergence of commonly used modelling languages.
- In this lecture we will look at the Bee-Up tool, which hybridizes several commonly used modelling languages in one prototypical implementation. These languages include: Business Process Model and Notation (BPMN), Event-driven Process Chains (EPC), Entity-Relationship models (ER), Unified Modeling Language (UML) and Petri Nets. Model processing capabilities, like process simulation or model transformation (e.g. to SQL), provide additional showcases on how these models can be utilized.

Motivation for Models

Gain an understanding of creating models and obtaining value through functionality of the Bee-Up Toolkit



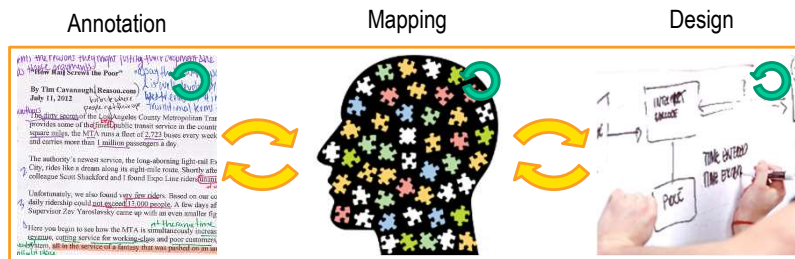
Hybridization – Example



Modeling

Annotation – Mapping – Design

- Understanding input and filtering out relevant information, e.g. through text annotation
- Mapping of relevant domain knowledge to concepts of the selected modeling language
- Design and create / complete model



Why Bee-Up exists?

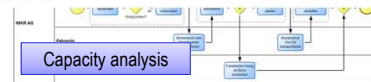
- We wanted to:
 - Enable a user to model in several commonly used languages in the same tool.
 - Exploit recurring semantics by allowing to execute certain mechanisms on different notations that comply to specific patterns.
 - Provide language-specific mechanisms and language-independent mechanisms by exploiting the different layers of abstraction. [source: Fundamental Conceptual Modeling Languages in OMILAB]
- A simpler motivation:
 - We have a course ...
 - ... where the students have to learn several modelling languages ...
 - ... and we'd like them to understand that having models provides advantages.
- and of course:
 - Proof of Concept
 - Cool experiments



YOU ARE HERE: U: FIND > COURSE

051023 VU Modellierung (2017S)

- ER-Modell
- UML - u.a. Anwendungsfall-, Klassen-, Aktivitäts-, Zustand
- TEIL 2:
 - Ziel und Zweck, sowie Arten von Geschäftsprozessen
 - Erhebung von Geschäftsprozessen
- EPK
- BPMN
- Petri Netze



- Doppelte Produktion → doppelt so viele Bienenstöcke
- Prozess wird 20 mal pro Tag ausgeführt.

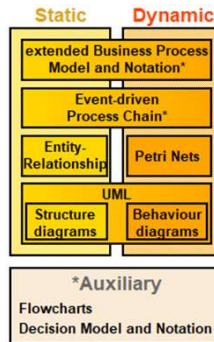
| Platz | Initial | Maximal | Minimal | Maximal | Minimal | Maximal | Minimal |
|-------|---------|---------|---------|---------|---------|---------|---------|
| P1 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P2 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P3 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P4 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P5 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P6 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P7 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P8 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P9 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P10 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P11 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P12 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P13 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P14 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P15 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P16 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P17 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P18 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P19 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |
| P20 | 10 | 10 | 0 | 10 | 0 | 10 | 0 |

Capacity exceeded



What is Bee-Up? – The Modelling Languages

- **B**PMN – Business Process Model and Notation
- **E**PC – Event-driven Process Chains
- **E**R – Entity Relationship
- **U**ML – Unified Modeling Language
- **P**etri Nets
- + Auxiliary
 - DMN
 - Flowcharts
 - ...



Actual Model Types:

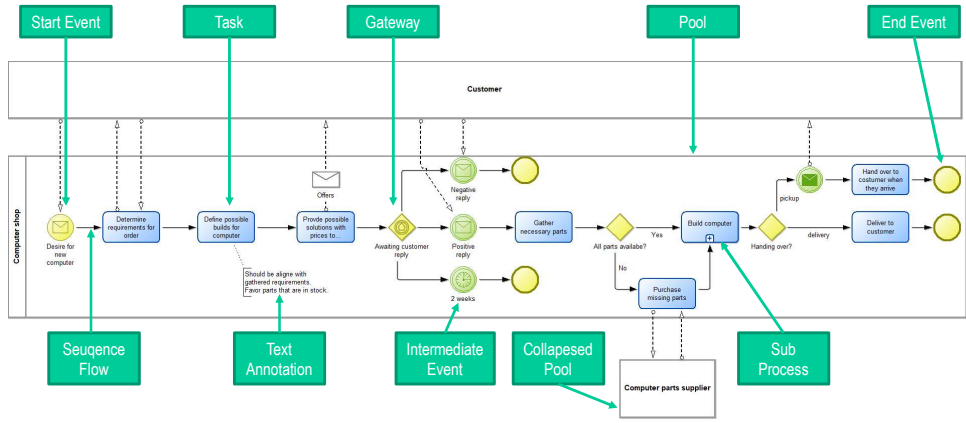
- Business Process Diagram (BPMN 2.0)
- EPC Model
- ER Model
- Activity Diagram
- Class / Object Diagram
- Classifier Pool
- Communication Diagram
- Component Diagram
- Composite Structure Diagram
- Deployment Diagram
- Interaction Overview Diagram
- Package Diagram
- Sequence Diagram
- State Machine Diagram
- Timing Diagram
- Use Case Diagram
- Petri Net
- Company Map
- Document Model
- Flowchart
- Decision Requirements Diagram
- Working Environment Model

MODELING LANGUAGES IN BEE-UP

Business Process Model and Notation (BPMN)

- Standardized language for the description of business processes
- Open standard for the modeling of business processes
- Supports both business and technical views of processes
- Developed by the Business Process Management Initiative (BPMI)
 - Today standard of the Object Management Group (OMG)
 - Since 2011 in the version BPMN 2.0
- Very complex business processes can be mapped

BPMN



BPMN – Further Information

OMG
OBJECT MANAGEMENT GROUP®

ABOUT US RESOURCE HUB OMG SPECIFICATIONS PROGRAMS MEMBERSHIP MEMBERS AREA

BUSINESS PROCESS MODEL & NOTATION™ (BPMN™) [BPMN](#) [HOME](#)

GRAPHICAL NOTATIONS FOR BUSINESS PROCESSES

The Business Process Model and Notation (BPMN) specification provides a graphical notation for specifying business processes in a Business Process Diagram. Its goal is to support Business Process Modeling by providing a standard notation that is comprehensible to business users yet represents complex process semantics for technical users.

Business Process Modeling Notation has become the de-facto standard for business processes diagrams. It is intended to be used directly by the stakeholders who design, manage and realize business processes, but at the same time be precise enough to allow BPMN diagrams to be translated into software process components. BPMN has an easy-to-use flowchart-like notation that's independent of any particular implementation environment.

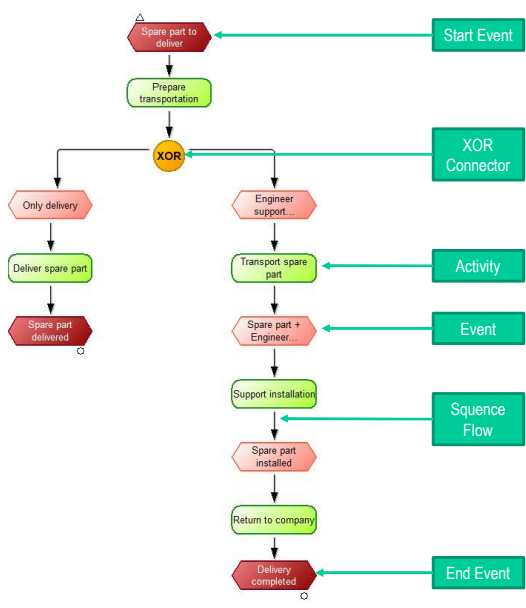
The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the

<https://www.omg.org/bpmn/index.htm> & <http://www.bpmn.org/>

Event-driven Process Chain (EPC)

- EPCs are used to model business processes
- The sequence of events and functions creates a coherent chain called the "event-driven process chain" (EPC)
 - Events trigger functions and functions generate events
- Part of an approach for modeling companies, together with data, organizational structure etc.

EPC



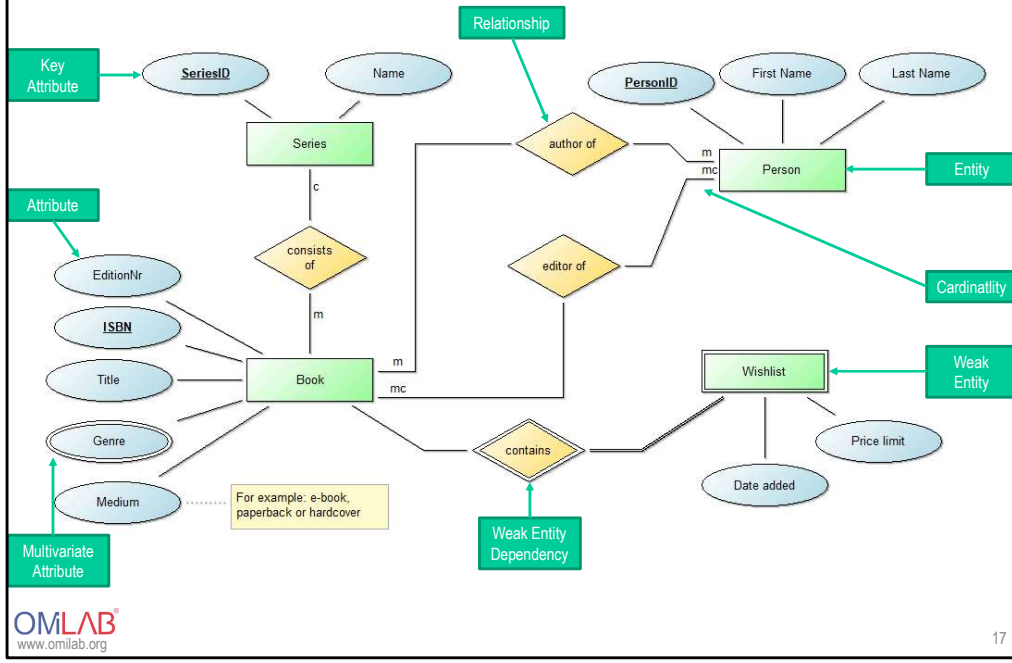
EPC – Further Information

- https://link.springer.com/chapter/10.1007/978-3-540-89224-3_2
- https://www.visual-paradigm.com/support/documents/vpuserguide/285/2144/56976_creating_epcd.html
- <https://dl.gi.de/handle/20.500.12116/830;jsessionid=386E78CC655ED5A361392B8F4BA6B67E>
- https://en.wikipedia.org/wiki/Event-driven_process_chain

Entity Relationship Model (ER)

- Entity Relationship Model, abbreviated ER Model or ERM
- 1976 presented by Peter Chen in his publication The Entity-Relationship Model
- Description tools for generalization and aggregation were introduced by Smith and Smith in 1977
- Some further developments e.g. at the end of the 80s by Wong and Katz
- In 1985, IDEF1X was published by the US Air Force as a standard for ER-based modelling.
- 1994 IDEF1X was adopted as a US government standard
- System independent data model
- In the conceptual phase of application development, serves the communication between users and developers, allows to create an abstract model of a static section of the real world (what, but not how)
- Serves as a basis for database design in the implementation phase; widely accepted tool for conceptual database design
- Elements of the ER model can be found in other languages, also in UML

ER



ER – Further Information

The screenshot shows the ACM Digital Library interface. At the top, there is a navigation bar with links for Journals, Magazines, Proceedings, Books, SIGs, Conferences, and People. A search bar is also present. The main content area displays the article title "The entity-relationship model—toward a unified view of data" by Peter Pin-Shan Chen. The article is marked as "FREE ACCESS". Below the title, there are social media sharing icons for Twitter, LinkedIn, Facebook, and Email. The author's name and a link to "Authors Info & Affiliations" are provided. The publication information is "ACM Transactions on Database Systems • March 1976 • https://doi.org/10.1145/320434.320440". The article has 4,394 citations and 25,731 views. There are buttons for "eReader" and "PDF". The abstract text is visible, starting with "A data model, called the entity-relationship model, is proposed. This model incorporates some of the important semantic information about the real world. A special diagrammatic technique is introduced as a tool for database design. An example of database design and description using the model and the diagrammatic technique is given. Some implications for data integrity, information retrieval, and data manipulation are discussed. The entity-relationship model can be used as a basis for unification of different views of data: the network model, the relational model, and the entity set model. Semantic ambiguities in these models are analyzed. Possible ways to derive their views of data from the entity-relationship model are presented." On the left side, there is a sidebar with navigation links for "ACM Transactions on Database Systems", "Volume #, Issue #", "Previous", "Next", "Abstract", "References", "Index Terms", and "Comments". A "Feedback" button is located at the bottom left of the article content area. On the right side, there is a "PDF" button with a "Help" link below it.

<https://dl.acm.org/doi/10.1145/320434.320440> or
http://bit.csc.lsu.edu/~chen/pdf/Chen_Pioneers.pdf

Unified Modeling Language (UML)

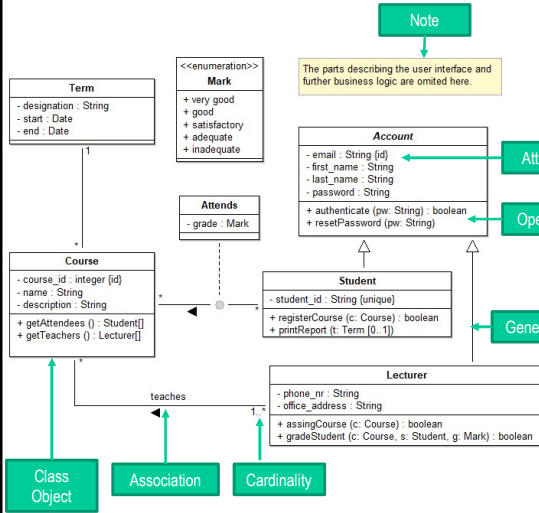
- Object-oriented concepts were first used in the 1960s in the programming language SIMULA.
- Object-oriented programming languages first became widespread through Smalltalk, Ada, C++ and Java (see graphic in section "2.3.4 Object-oriented models").
- Object-oriented analysis and design methods from the mid-80s on.
- A "method war" is followed in 1996 by the Object Management (OMG) call for the creation of a modeling standard.
- 1998 the Unified Modeling Language (UML) 1.1 is accepted by the OMG as a modeling standard.
- Since then, UML 1.1 has been extended and UML 2.x established as the central, widely accepted modeling language.

UML

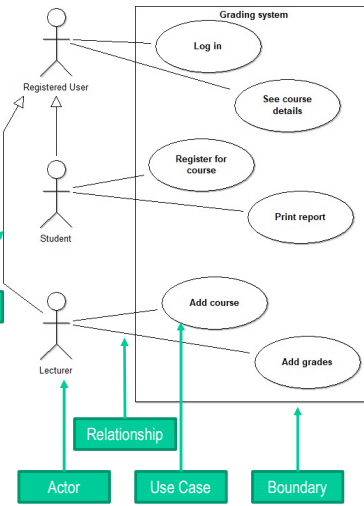
| Structure diagrams (static structures: class structures up to structuring of entire systems and architectures) | Behaviour diagrams (sequences between static parts) |
|---|---|
| <ul style="list-style-type: none">• Class Diagram• Object Diagram• Component Diagram• Profile Diagram• Compositional Structure Diagram• Distribution Diagram• Package Diagram | <ul style="list-style-type: none">• Activity Diagram• Use Case Diagram• Status Diagram• Sequence Diagram• Interaction Overview Diagram• Communication Diagram• Time Diagram |

UML

UML Class Diagram



UML Use Case Diagram



UML – Further Information

The screenshot shows the official UML specification page on the Object Management Group (OMG) website. The page title is "ABOUT THE UNIFIED MODELING LANGUAGE SPECIFICATION VERSION 2.5.1". The main heading is "UML® Unified Modeling Language". Below this, a brief description states: "A specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems." A metadata table provides details: Title: Unified Modeling Language; Acronym: UML®; Version: 2.5.1; Document Status: Formal; Publication Date: December 2017; Categories: Software Engineering; IPR Mode: RF-Limited. A "Specification" icon is also present. The page includes a "TABLE OF CONTENTS" section with links to "About the Specification", "Specification Documents" (Normative, Normative Machine-Readable, Informative), "History", "Formal Versions", and "Links". At the bottom, it provides two URLs: <https://www.omg.org/spec/UML/About-UML/> and <https://www.uml.org/>. The footer features the OMILAB logo and the website address www.omilab.org.

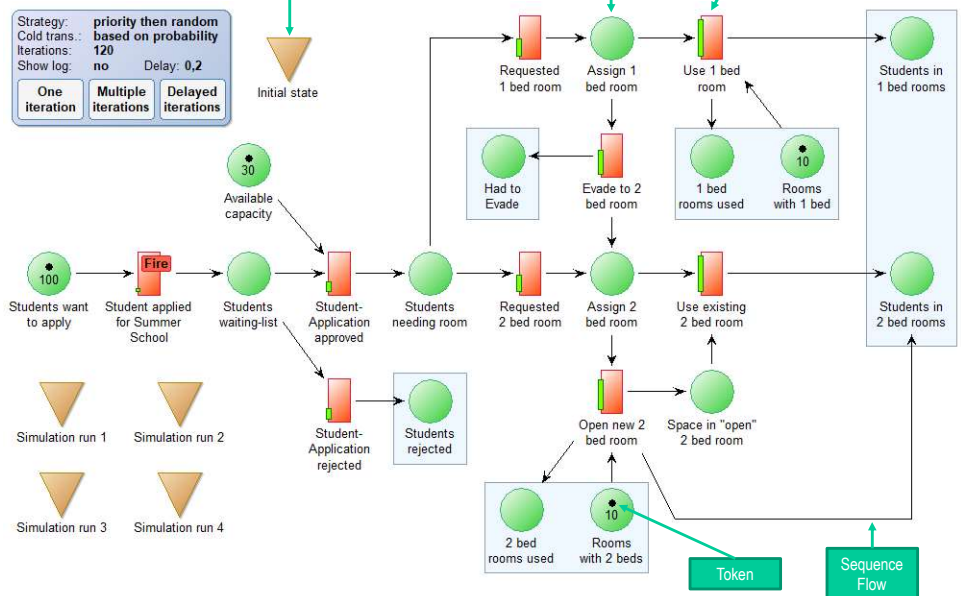
OMILAB®
www.omilab.org

22

Petri-Net

- Developed by computer scientist Carl Adam Petri in the 1960s
- In the 60's and 70's, the focus was on theoretical foundations, since the 80's also for application and tool support
- Similarities to state-transition diagrams
- Strong mathematical / formal foundation
- Good comprehensibility through simple graphic visualization
- Formal syntax and semantics
- States are explicitly represented
- Numerous applications, analysis methods, scientific literature, tools available
- Application in many domains
 - Design, implementation and control of software systems
 - Design of computer-aided collaboration
 - Verification of concurrent processes
 - Modeling of distributed algorithms and agent networks
 - Specification of workflows

Petri-Net



Petri-Net – Further Information

Welcome on

Pnml.org

resources

Pnml Tools

grammar

latest grammar 2009
2008-08-29

validation Petri-Net
2010-04-25

standard

Part 1 IS
Mar 2008

Amd 1 IS
May 2010

Part 2 IS
Feb 2011

last update Sept 21, 2015

News

- International Workshop on ISG/EC 1999-9, June 3, 2015, in UPMC (L3H), Properties, modularity and stochastic Petri nets.
- The purpose of this workshop is to define the conceptual (formal definitions) and technical (metamodels, and exchange format) frameworks for properties, modularity and stochastic Petri nets, that are currently the most important requested additions to the standard by the Petri net community. This international workshop gathers experts from France (UPMC, Inria, Paris 13 and CNRS), and Denmark (Technical University of Denmark).
- A corrigendum on part 1 (Pnml) was edited in 2012 to fix some conceptual and syntactic bugs in the Pnml definitions. It is now published by the ISO.
- The Model Checking Contest @ Petri nets is the first international Petri nets competition making a large use of Petri net models in Pnml. Check it out!

Context

Pnml.org is the reference site for the implementation of Petri Net Markup Language (Pnml), defined by the standard ISO/IEC 15959-1:2010 Part 1, Part 1 is published here.

This site provides you with information about the Pnml Standard, to foster its quick and large adoption. The IEC/ISO specification of Pnml, and links to supporting tools can be found on this site.

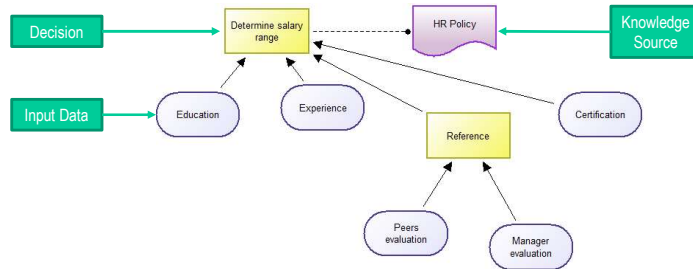
The Petri net community has long sought the means to exchange Petri net models unambiguously, based on an agreed-on interchange format. However, they quickly realized that it would be even better if that format was compliant with an agreed-on formal definition of Petri nets. Clearly defined, it must be the concrete syntax of the abstract syntax specified from that formal definition.

<http://www.pnml.org/>

Decision Model and Notation (DMN)

- DMN is a standard published by the Object Management Group (OMG)
- In August 2013, OMG published the first version
- DMN is a specialized notation to model decisions
- It is independent form BPMN

DMN



Based on <https://why-what-how.com/lms/mod/page/view.php?id=578>

DMN – Further Information

The screenshot shows the top portion of the DMN website. At the top is a teal header with the text 'HOME', 'CONTACT', and 'LEGAL' on the left, and social media icons for Facebook, LinkedIn, and Twitter on the right. Below this is a white navigation bar with the 'OMG' logo and the text 'OBJECT MANAGEMENT GROUP' on the left, and a menu with 'ABOUT US', 'RESOURCE HUB', 'OMG SPECIFICATIONS', 'PROGRAMS', 'MEMBERSHIP', and 'MEMBERS AREA' on the right. A dark grey banner below the navigation bar contains the text 'DECISION MODEL AND NOTATION™ (DMN™)' on the left and 'POPULAR STANDARDS • HOME' on the right. The main content area has a light grey background with a large, faint image of a person. The text reads: 'PRECISE SPECIFICATION OF BUSINESS DECISIONS AND BUSINESS RULES'. Below this is a small graphic of the 'DMN' logo with the text 'Decision Model & Notation™'. To the right of the logo is a paragraph: 'DMN is a modeling language and notation for the precise specification of business decisions and business rules. DMN is easily readable by the different types of people involved in decision management. These include: business people who specify the rules and monitor their application; business analysts. DMN is designed to work alongside BPMN and CMMN, providing a mechanism to model the decision-making associated with processes and cases. While BPMN, CMMN and DMN can be used independently, they were carefully designed to be complementary. Indeed, many organizations require a combination of process models for their prescriptive workflows, case models for their reactive activities, and decision models for their more complex, multi-criteria business rules. Those organizations will benefit from using the three standards in combination, selecting which one is most appropriate to each type of activity modeling. This is why BPMN, CMMN and DMN really constitute the "triple crown" of process improvement standards. Using DMN to model your organizational decision-making will:'. Below this paragraph is a bulleted list of six points.

DMN is a modeling language and notation for the precise specification of business decisions and business rules. DMN is easily readable by the different types of people involved in decision management. These include: business people who specify the rules and monitor their application; business analysts.

DMN is designed to work alongside BPMN and CMMN, providing a mechanism to model the decision-making associated with processes and cases. While BPMN, CMMN and DMN can be used independently, they were carefully designed to be complementary. Indeed, many organizations require a combination of process models for their prescriptive workflows, case models for their reactive activities, and decision models for their more complex, multi-criteria business rules. Those organizations will benefit from using the three standards in combination, selecting which one is most appropriate to each type of activity modeling. This is why BPMN, CMMN and DMN really constitute the "triple crown" of process improvement standards.

Using DMN to model your organizational decision-making will:

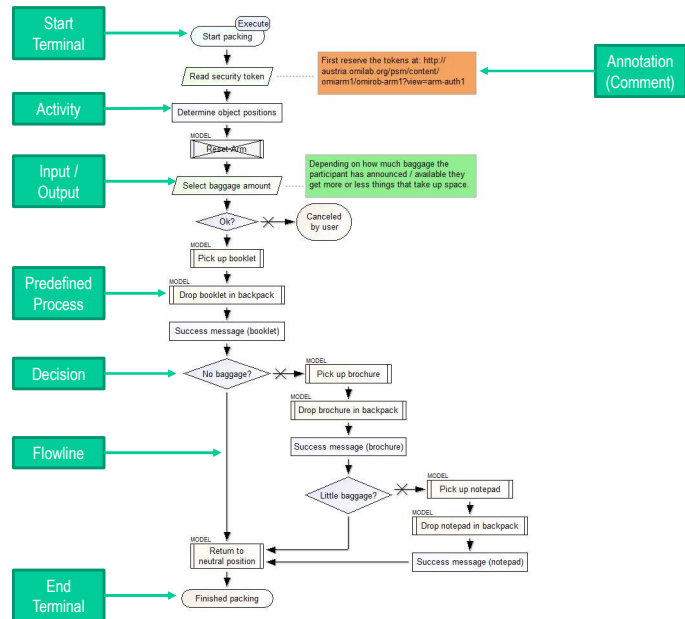
- help all stakeholders understand a complex domain of decision-making using easily readable diagrams
- provide a natural basis for discussion and agreement on the scope and nature of business decision-making
- reduce the effort and risk of decision automation projects through graphical decomposition of requirements
- allow business rules to be defined simply and reliably in unambiguous decision tables
- simplify development of decisioning systems using specifications that may be automatically validated and executed
- provide a structured context for the development and management of predictive analytic models
- enable the development of a library of reusable decision-making components.

<https://www.omg.org/dmn/>

Flowchart

- Represents a workflow or a process
- Can be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task
- First introduced by Frank and Lillian Gilbreth as the “flow process chart”
- In the 1930s, business people were trained in the use of some of the tools of industrial engineering
- Used and further developed by American Society of Mechanical Engineers (ASME), Procter and Gamble, IBM
- Popular tool for describing computer algorithms until the 1970s
- Nowadays used for describing computer algorithms
- UML activity diagrams and Drakon-charts can be considered to be extensions of the flowchart
- Flowchart Types
 - Document flowcharts, showing controls over a document-flow through a system
 - Data flowcharts, showing controls over a data-flow in a system
 - System flowcharts, showing controls at a physical or resource level
 - Program flowchart, showing the controls in a program within a system

Flowchart



Flowchart – Further Information

- <https://www.visual-paradigm.com/tutorials/flowchart-tutorial/>
- <https://www.smartdraw.com/flowchart/>

MODELLING IN BEE-UP

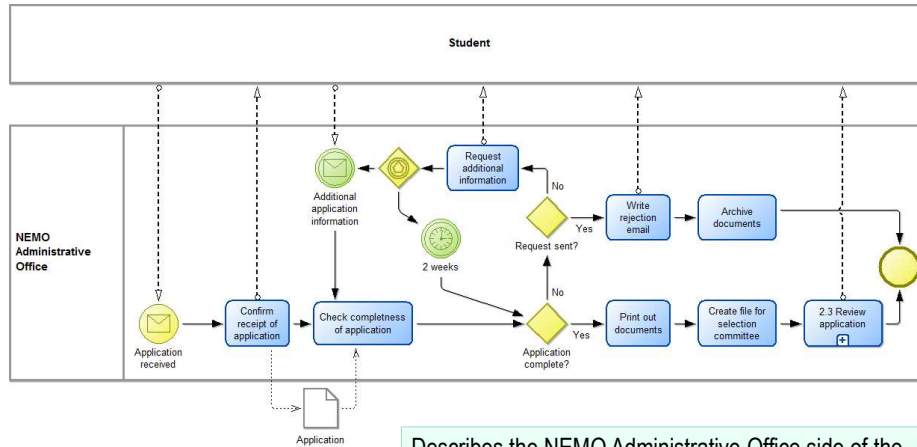
Bee-Up Overview

The screenshot shows the Bee-Up software interface. At the top right, a green box highlights the 'Modelling' menu, which includes 'Analysis', 'Import/Export', 'Modelling', and 'Simulation'. A green arrow points from this menu to the 'My saved models' area on the left. The main workspace displays a grid of model thumbnails. A 'Create new model' dialog box is open in the center, showing a tree view of model types categorized into 'Static' and 'Dynamic'. A green box labeled 'New model' points to the 'Create new model' dialog. Another green box labeled 'Recently opened models' points to the bottom section of the model grid. The OMILAB logo and website URL are visible in the bottom left corner.

Find out who handles application data

QUERYING INFORMATION

Querying Information – Used Model



Describes the NEMO Administrative Office side of the application submission

Querying Information – Execute the Query

Objects linked to "Application"

Show "Responsible"

results - ({<"Data Object (BPMN)"><"Data Association (BPMN)"> OR ({<"Data Object (BPMN)">->"Data Association (BPMN)">})
Responsible
 1. Application Process (NEMO)
 Check completeness of application [Iulia Vaidian \(Performer\) - NEMO Organization \(Working Environn\)](#)
 Confirm receipt of application [Iulia Vaidian \(Performer\) - NEMO Organization \(Working Environn\)](#)

OMILAB
www.omilab.org

36

Query:

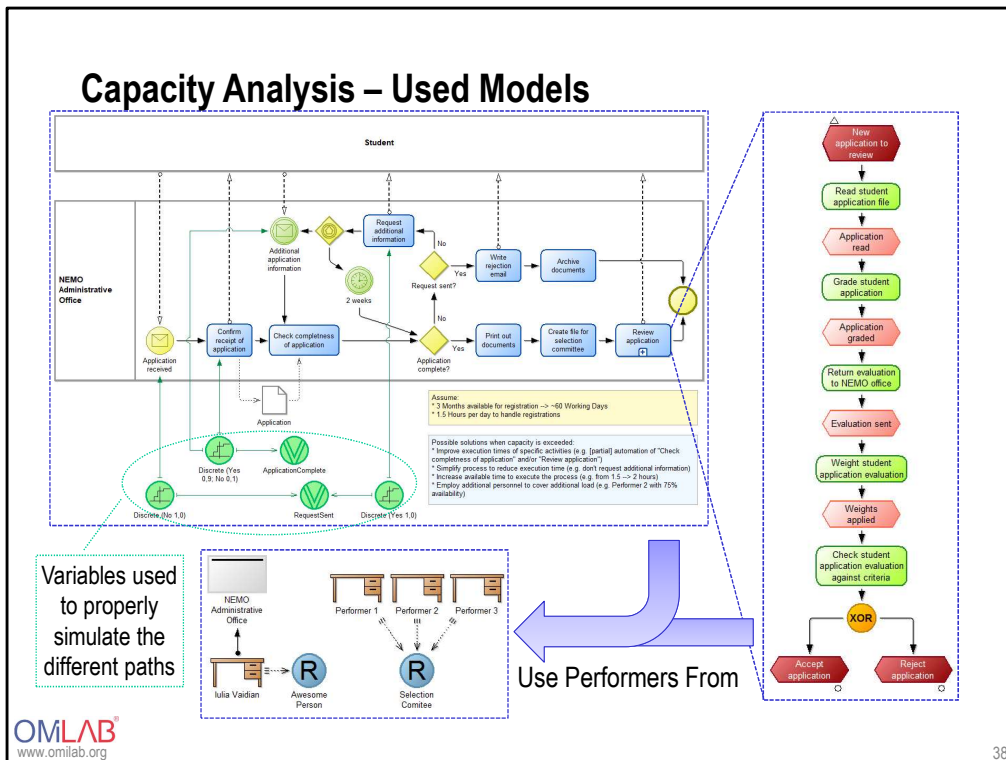
```
{\"Application\": \"Data Object (BPMN)\"} <- \"Data Association (BPMN)\") OR
({\"Application\": \"Data Object (BPMN)\"} -> \"Data Association (BPMN)\")
```

Alternative Query:

```
((<"Data Object (BPMN)"><"Data Association (BPMN)">) OR
((<"Data Object (BPMN)">->"Data Association (BPMN)">))
```

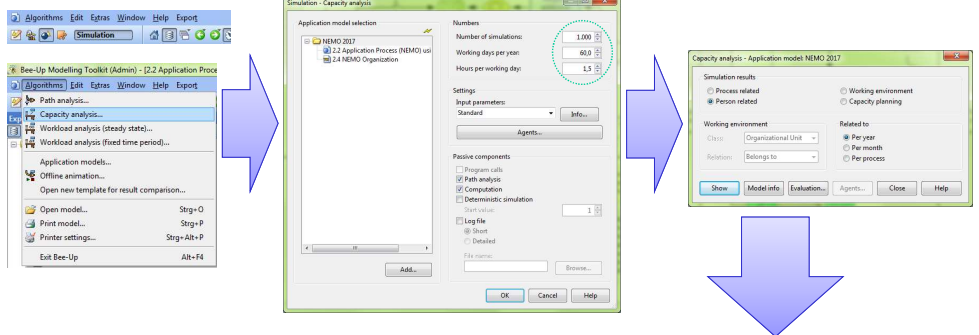
Determine load on employees for a process

CAPACITY ANALYSIS



The number of applications per year is specified in the Start Event. Times have been entered into the BPMN/EPC elements, but not costs (since we don't do Cost analysis)! The Subsequent relations from the two Gateways "Application complete?" and "Request sent?" have conditions based on the Variables. The Subsequent relations from the one Event-Based Gateway use probabilities (80% → information provided; 20% → 2 weeks pass).

Capacity Analysis – Perform the Analysis



Simulation results - Capacity analysis (Person related/Per year) - Application model: NEMO 2017

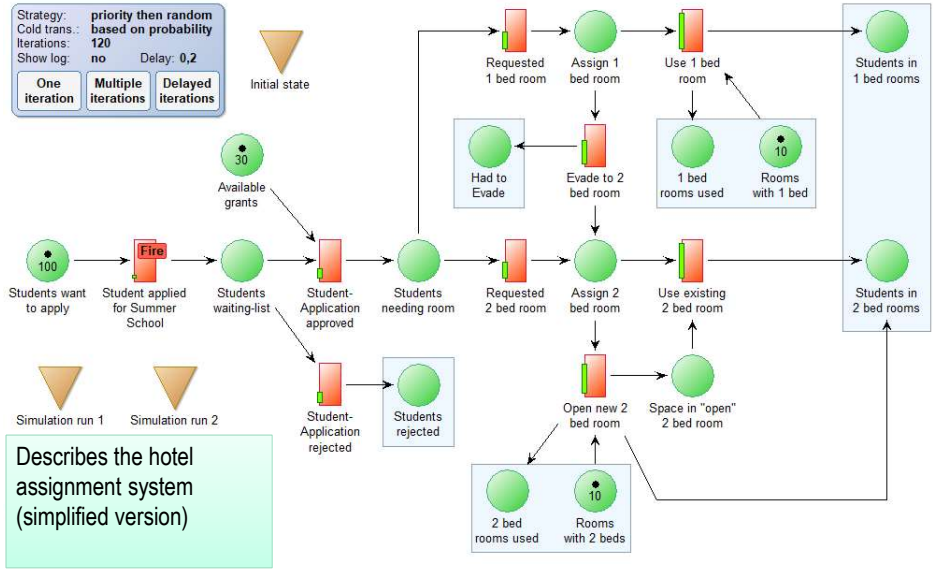
| | Performer | Business process | Activity | Number | Execution time (sum) | Capacity |
|--------|---------------------------------------|-------------------------|---|------------|----------------------|----------|
| 1. | Iulia Vaidian (2.4 NEMO Organization) | 2.2 Application Process | | | 00:044:01:03:36 | 0,745111 |
| 1.1. | | | Print out documents (2.2 Application Proce | 192,400000 | 00:044:01:03:36 | |
| 1.1.1. | | | Create file for selection committee (2.2 Appl | 192,400000 | 00:008:00:24:48 | |
| 1.1.2. | | | Request additional information (2.2 Applicat | 17,800000 | 00:008:00:49:36 | |
| 1.1.3. | | | Confirm receipt of application (2.2 Applicat | 200,000000 | 00:002:00:35:36 | |
| 1.1.4. | | | Write rejection email (2.2 Application Proce | 7,600000 | 00:002:00:20:00 | |
| 1.1.5. | | | Archive documents (2.2 Application Proce | 7,600000 | 00:000:00:15:12 | |
| 1.1.6. | | | Check completeness of application (2.2 Appl | 212,800000 | 00:011:01:14:00 | |
| 1.1.7. | | | Weight student application evaluation (2.3) | 192,400000 | 00:010:01:02:00 | |
| 1.1.8. | | | Check student application evaluation again | 192,400000 | 00:006:00:37:12 | |
| 1.1.9. | | | | | | |
| 2. | Performer 1 (2.4 NEMO Organization) | 2.2 Application Process | | | 00:010:01:04:48 | 0,178667 |
| 2.1. | | | Read student application file (2.3 Review.sp | 56,600000 | 00:010:01:04:48 | |
| 2.1.1. | | | | | 00:006:00:26:00 | |
| 2.1.2. | | | | | 00:003:01:03:00 | |

~75%
C

Analyze behavior of the modelled system

PETRI NET SIMULATION

Petri Net Simulation – Used Model



Petri Net Simulation – Simulation Results

| Simulation Run | Results |
|----------------|---------|
| 1 | |
| 2 | |
| 3 | |
| 4 | |

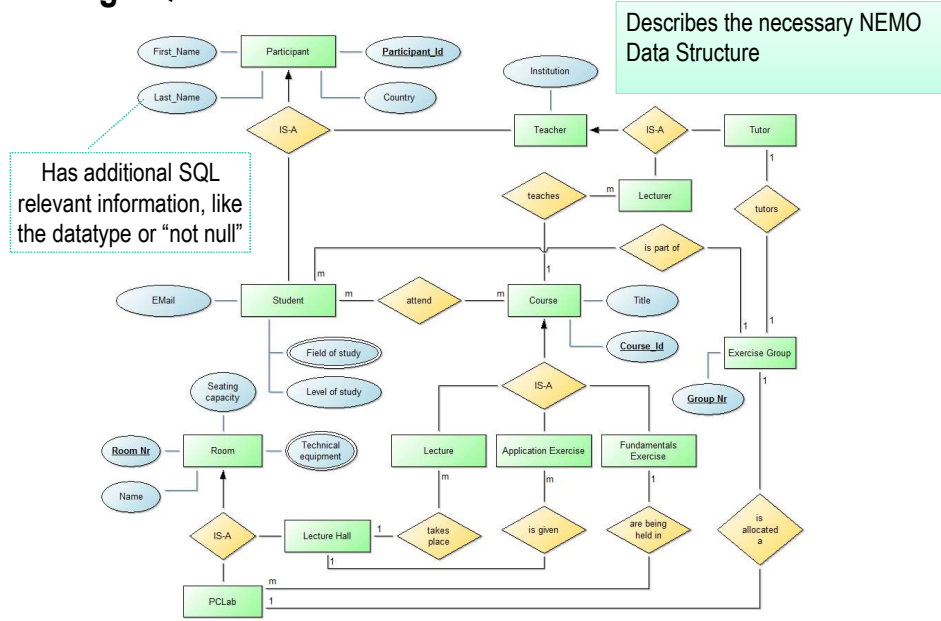
The system does not cover “evading” from 2 to a 1 bedroom

Are constant through all simulation runs

Automatically generate SQL code from a model describing the data structure

DERIVING SQL

Deriving SQL – Used Model



Deriving SQL – Code Snippet

```
CREATE TABLE attend (
  Participant_Id INT NOT NULL,
  Course_Id INT NOT NULL,
  CONSTRAINT pk_attend PRIMARY KEY
  (Participant_Id, Course_Id)
);
```

```
CREATE TABLE Course (
  Course_Id INT NOT NULL,
  Title VARCHAR(32),
  CONSTRAINT pk_Course PRIMARY KEY
  (Course_Id)
);
```

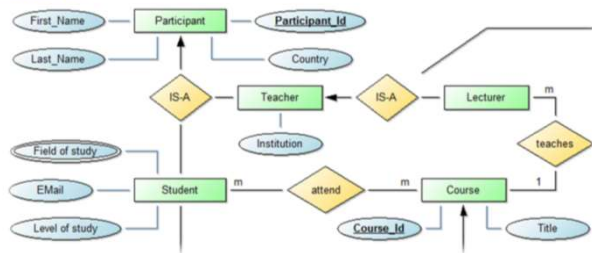
```
CREATE TABLE Student (
  Participant_Id INT NOT NULL,
  First_Name VARCHAR(32),
  Last_Name VARCHAR(32),
  Country VARCHAR(32),
  Level_of_study VARCHAR(32),
  Field_of_study VARCHAR(200),
  EMail VARCHAR(60),
  Group_Nr INT,
  CONSTRAINT pk_Student PRIMARY KEY
  (Participant_Id)
);
```

```
CREATE TABLE Lecturer (
  Participant_Id INT NOT NULL,
  Institution VARCHAR(32),
  First_Name VARCHAR(32),
  Last_Name VARCHAR(32),
  Country VARCHAR(32),
  Course_Id INT,
  CONSTRAINT pk_Lecturer PRIMARY KEY
  (Participant_Id)
);
```

```
ALTER TABLE attend ADD CONSTRAINT fk_attend_Student FOREIGN KEY
(Participant_Id) REFERENCES Student(Participant_Id);
```

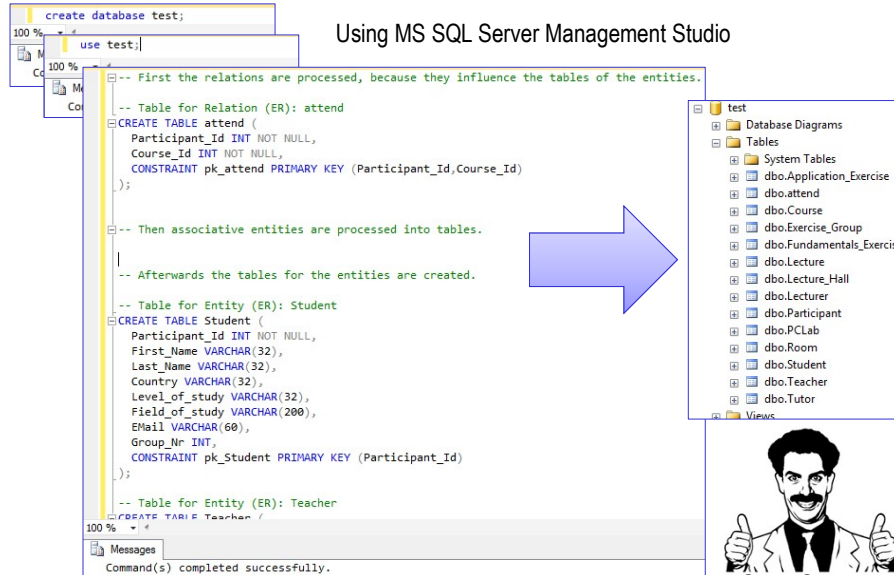
```
ALTER TABLE attend ADD CONSTRAINT fk_attend_Course FOREIGN KEY
(Course_Id) REFERENCES Course(Course_Id);
```

```
ALTER TABLE Lecturer ADD CONSTRAINT fk_Lecturer_Course FOREIGN KEY
(Course_Id) REFERENCES Course(Course_Id);
```



Deriving SQL – Running the Code

Using MS SQL Server Management Studio



The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'test' database structure with folders for Database Diagrams, Tables, System Tables, and Views. The 'Tables' folder is expanded, listing various tables such as dbo.Application_Exercise, dbo.attend, dbo.Course, dbo.Exercise_Group, dbo.Fundamentals_Exercise, dbo.Lecture, dbo.Lecture_Hall, dbo.Lecturer, dbo.Participant, dbo.PCLab, dbo.Room, dbo.Student, dbo.Teacher, and dbo.Tutor. The right pane shows the execution of SQL code in a query window. The code includes comments and SQL statements for creating tables: 'attend' (a relation table with primary key on Participant_Id and Course_Id), 'Student' (an entity table with primary key on Participant_Id), and 'Teacher' (an entity table). The Messages pane at the bottom indicates 'Command(s) completed successfully.' A blue arrow points from the code window to the table list. Below the table list is a cartoon character giving a thumbs up with the text 'Great Success'.

```
create database test;
use test;

-- First the relations are processed, because they influence the tables of the entities.
-- Table for Relation (ER): attend
CREATE TABLE attend (
  Participant_Id INT NOT NULL,
  Course_Id INT NOT NULL,
  CONSTRAINT pk_attend PRIMARY KEY (Participant_Id,Course_Id)
);

-- Then associative entities are processed into tables.
-- Afterwards the tables for the entities are created.
-- Table for Entity (ER): Student
CREATE TABLE Student (
  Participant_Id INT NOT NULL,
  First_Name VARCHAR(32),
  Last_Name VARCHAR(32),
  Country VARCHAR(32),
  Level_of_study VARCHAR(32),
  Field_of_study VARCHAR(200),
  EMail VARCHAR(60),
  Group_Nr INT,
  CONSTRAINT pk_Student PRIMARY KEY (Participant_Id)
);

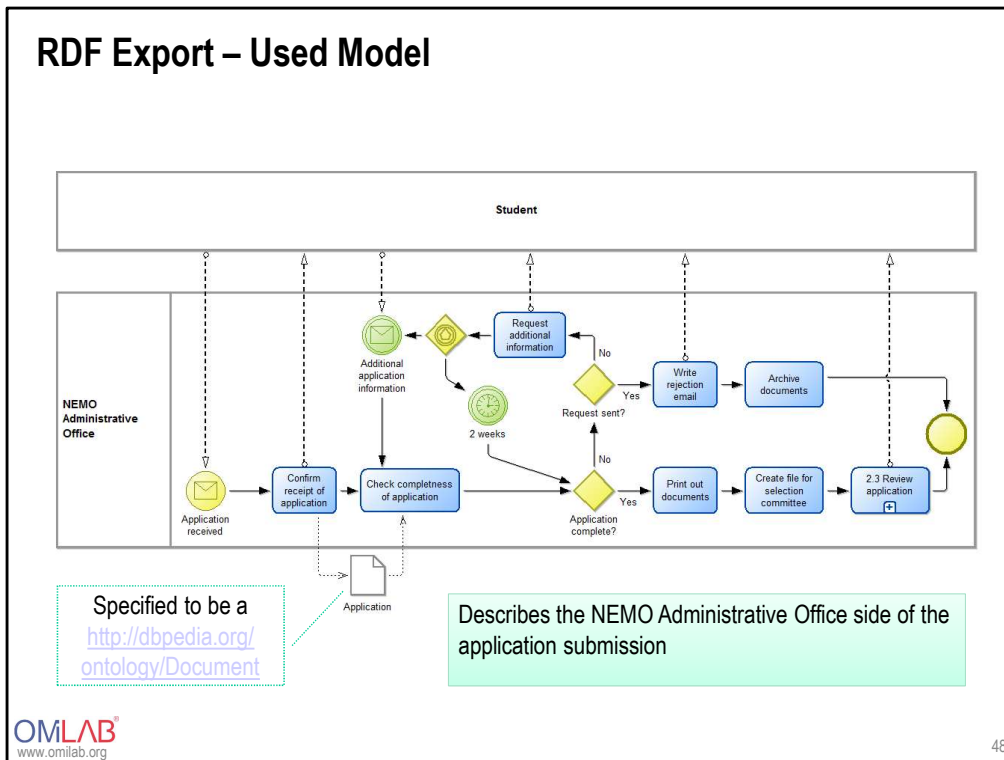
-- Table for Entity (ER): Teacher
CREATE TABLE Teacher /
```

Code to list all the tables in MS SQL:
SELECT * FROM INFORMATION_SCHEMA.TABLES;

Expose models as RDF to allow the use of semantic web technologies

RDF EXPORT

RDF Export – Used Model



From Dbpedia About page: “DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link the different data sets on the Web to Wikipedia data. We hope that this work will make it easier for the huge amount of information in Wikipedia to be used in some new interesting ways. Furthermore, it might inspire new mechanisms for navigating, linking, and improving the encyclopedia itself.”

RDF Export – TriG Code Snippet

```
<http://www.omilab.org/example#Business_Process_Diagram_BPMN_2_0-2_1_Application_Process_NEMO> {
  <http://nemo.omilab.org/ontology/registration#Application>
    a
      <http://dbpedia.org/ontology/Document> , cv:o_Modelling_object , mm:o_Data_Object_BPMN ;
    rdfs:label "Application";
    mm:a_Auditing "No";
    mm:a_Collection "No";
    mm:a_Monitoring "No";
    mm:a_URl <http://nemo.omilab.org/ontology/registration#Application>;
    cv:a_Name "Application" .

  <http://www.omilab.org/example#Data_Association_BPMN-39679-Confirm_receipt_of_application-Application>
    a mm:r_Data_Association_BPMN , cv:r_Modelling_relation_a ;
    cv:from <http://www.omilab.org/example#Task_BPMN-39573-Confirm_receipt_of_application>;
    cv:to <http://nemo.omilab.org/ontology/registration#Application> .

  <http://www.omilab.org/example#Task_BPMN-39573-Confirm_receipt_of_application>
    a mm:o_Task_BPMN , cv:o_Modelling_object ;
    rdfs:label "Confirm receipt of application";
    mm:a_Execution_time "00:00:00:01:00";
    mm:a_For_compensation "No";
    mm:a_Global_task "No";
    mm:a_Loop_type "Not specified";
    mm:a_Monitoring "No";
    mm:a_Resting_time "00:00:00:00:00";
    mm:a_Task_type "Not specified";
    mm:a_Transport_time "00:00:00:00:00";
    mm:a_Waiting_time "00:00:00:00:00";
    mm:r_Is_inside <http://www.omilab.org/example#Pool_BPMN-39608-NEMO_Administrative_Office>;
    mm:r_Responsible <http://www.omilab.org/example#Performer-37003-Lulia_Vaidian>;
    cv:a_Name "Confirm receipt of application" .

  <http://www.omilab.org/example#Performer-37003-Lulia_Vaidian>
    cv:described_in <http://www.omilab.org/example#Working_Environment_Model-2_4_NEMO_Organization> .
}
```

IMPORTANT: Export BOTH the Process and the Working Environment Model!

RDF Export – TriG Code Snippet

```

<http://www.omilab.org/example#Business_Process_Diagram_BPMN_2_0-2_1_Application_Process_NEMO> {
  <http://nemo.omilab.org/ontology/registration#Application>
    a
      <http://dbpedia.org/ontology/Document> , cv:o_Modelling_object , mm:o_Data_Object_BPMN ;
    rdfs:label
      "Application" ;
    mm:a_Auditing
      "No" ;
    mm:a_Collection
      "No" ;
    mm:a_Monitoring
      "No" ;
    mm:a_URl
      <http://nemo.omilab.org/ontology/registration#Application> ;
    cv:a_Name
      "Application" .

<http://www.omilab.org/example#Data_Association_BPMN-39679-Confirm_r
  a
    mm:r_Data_Association_BPMN , cv:r_Modelling_relation_a ;
  cv:from
    <http://www.omilab.org/example#Task_BPMN-39573-Confirm
  cv:to
    <http://nemo.omilab.org/ontology/registration#Application> .

<http://www.omilab.org/example#Task_BPMN-39573-Confirm
  a
    mm:o_Task_BPMN , cv:o_Modelling_object ;
  rdfs:label
    "Confirm receipt of application" ;
  mm:a_Execution_time
    "00:00:00:01:00" ;
  mm:a_For_compensation
    "No" ;
  mm:a_Global_task
    "No" ;
  mm:a_Loop_type
    "Not specified" ;
  mm:a_Monitoring
    "No" ,
  mm:a_Resting_time
    "00:00:00:00:00:00" ;
  mm:a_Task_type
    "Not specified" ;
  mm:a_Transport_time
    "00:00:00:00:00:00" ;
  mm:a_Waiting_time
    "00:00:00:00:00:00" ;
  mm:r_Is_inside
    <http://www.omilab.org/example#Pool_BPMN-39608-NE
  mm:r_Responsible
    <http://www.omilab.org/example#Performer-3700
  cv:a_Name
    "Confirm receipt of application" .

<http://www.omilab.org/example#Performer-37003-Lulia_Vaidian>
  cv:described_in
    <http://www.omilab.org/example#Working_Environment_Model-2_4_NEMO_Organization> .
}

```

rd4j / workbench

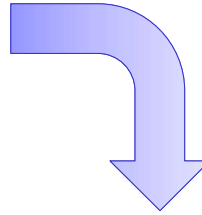
The screenshot shows the 'rd4j / workbench' interface. On the left, there is a sidebar with a menu containing 'Repositories', 'New repository', 'Delete repository', 'Export', 'Import', 'Save', 'Export', 'Import', 'Modify', 'SPARQL Update', 'Add', 'Remove', 'Clear', 'System', and 'Information'. The main area is titled 'Add RDF' and contains the following fields and options:

- Base URI:
 - use base URI as context identifier
- Context:
- Data format: (autodetect)
 - Location of the RDF data you wish to upload
 - Select the file containing the RDF data you wish to upload
- RDF Data URL:
- RDF Data File: RdfExportExample.trig
 - Enter the RDF data you wish to upload
- RDF Content:
-

RDF Export – Performers Responsible for a Document

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cv: <http://www.comvantage.eu/mm#>
PREFIX mm: <http://austria.omilab.org/psm/content/bee-up/1_5#>
PREFIX : <http://www.omilab.org/example#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT DISTINCT (?doc AS ?Document) (?resp AS ?Responsible) (?respname AS ?ResponsibleName)
WHERE {
  # Everything that is a document
  ?doc a <http://dbpedia.org/ontology/Document> .
  # Data associations ...
  ?rel a mm:r_Data_Association_BPMN .
  # ... which connect to the document and an activity/task
  {
    ?rel cv:from ?doc .
    ?rel cv:to ?act .
  } UNION {
    ?rel cv:to ?doc .
    ?rel cv:from ?act .
  }
  # The responsible for that activity/task
  ?act mm:r_Responsible ?resp .
  ?resp rdfs:label ?respname .
}
```



| Document | Responsible | ResponsibleName |
|---|---|-----------------|
| http://nemo.omilab.org/ontology/resistration#Application | http://www.omilab.org/example#Performer-37003-Julia_Vaidian | "Julia Vaidian" |

Packing the backpack using a robotic arm

MODELS AT RUNTIME

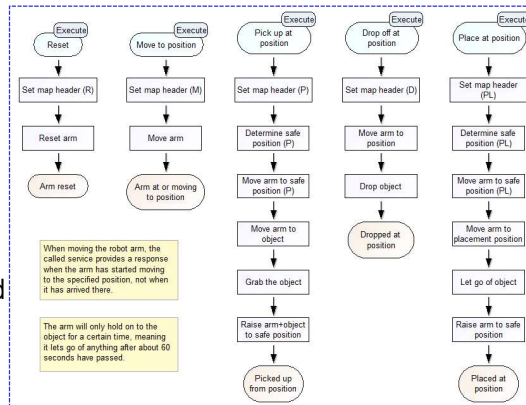
Models for Runtime – Used Models



Described Through

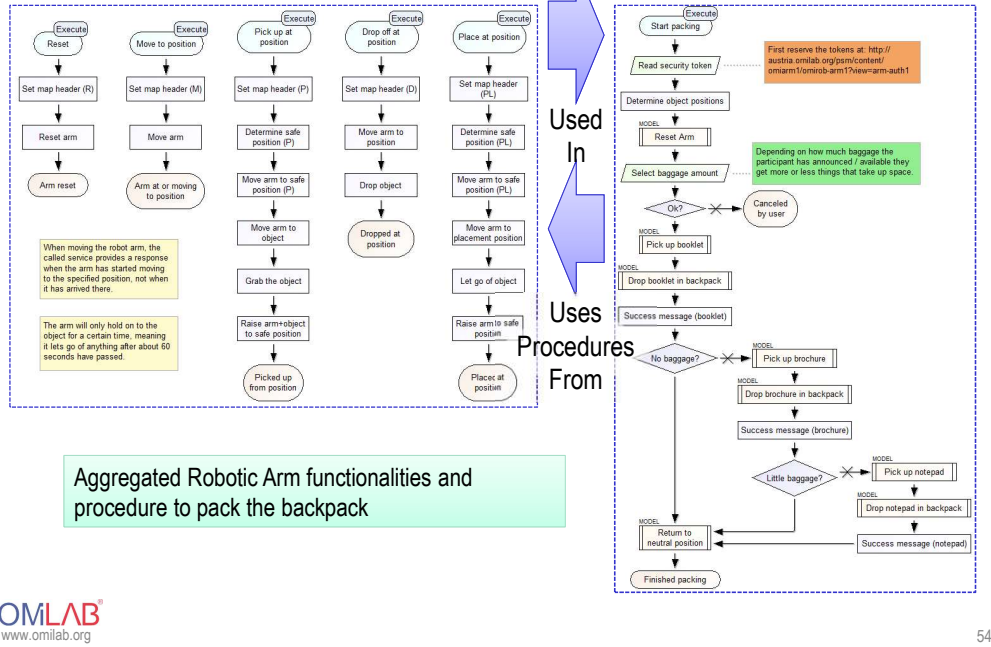
Available capabilities:

- Rotate
- Grab on
- Grab off

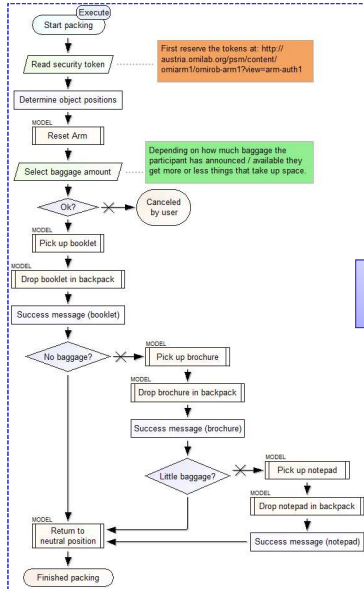


A Model aggregating basic capabilities to useful functionalities

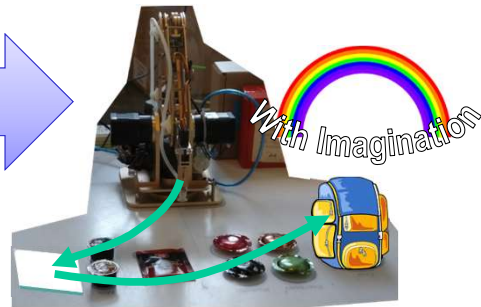
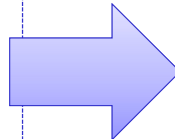
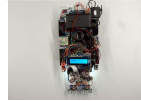
Models at Runtime – Used Models



Models at Runtime – Execution



OMILAB® - Rob



Try yourself with a similar scenario:

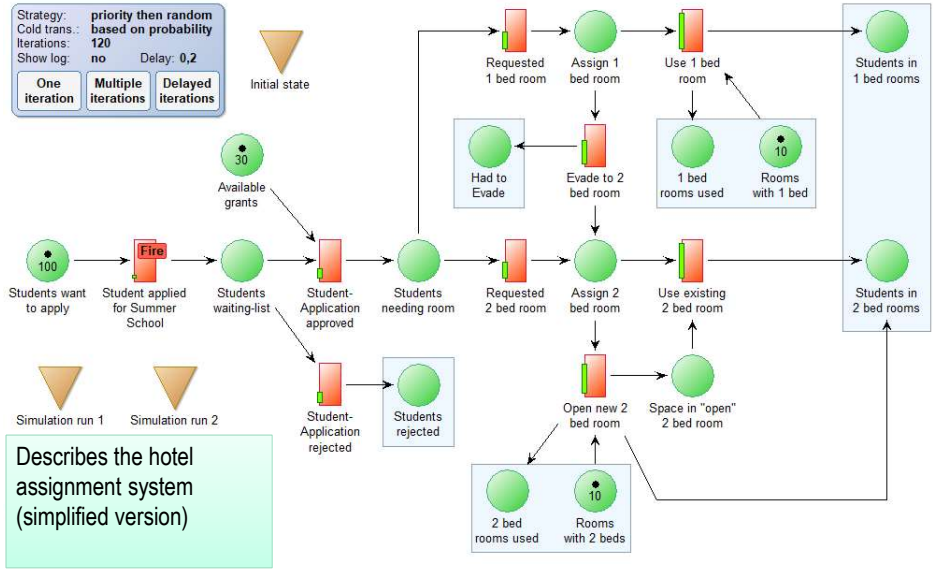
<http://austria.omilab.org/psm/content/bee-up/info?view=scenarios>

Robotic Arm: <http://austria.omilab.org/psm/content/omiarm1/info>

How you can use custom functionalities and what we are working on

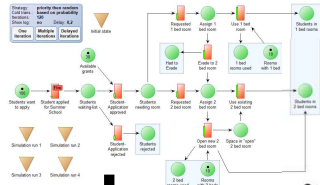
EXTENDING BEE-UP

Using an External Service – Used Model



Describes the hotel assignment system (simplified version)

Using an External Service – Setup



LoLA Petri Net Verification

The LoLA ADOxxWEB API is an implementation of a webservice wrapper (/lola/index.html). To enable integration of the LoLA functionality, the LoLA t

The implementation depends on concepts from our language.



```

 Authentication
Username:
Password:

Pre-processing:
# We begin with the parts that require user input and could prematurely stop the execution
# First the body that is sent
CC "Modeling" GET_ACT_MODEL
PN PLACE_TRANSITION_NET (modelid) str_result:str_ptnet
IF (code != 0) {
# We don't have to explicitly print the error here
EXT
}
SETL map_req(mapID)
SETL map_req["network"]:(str_ptnet)

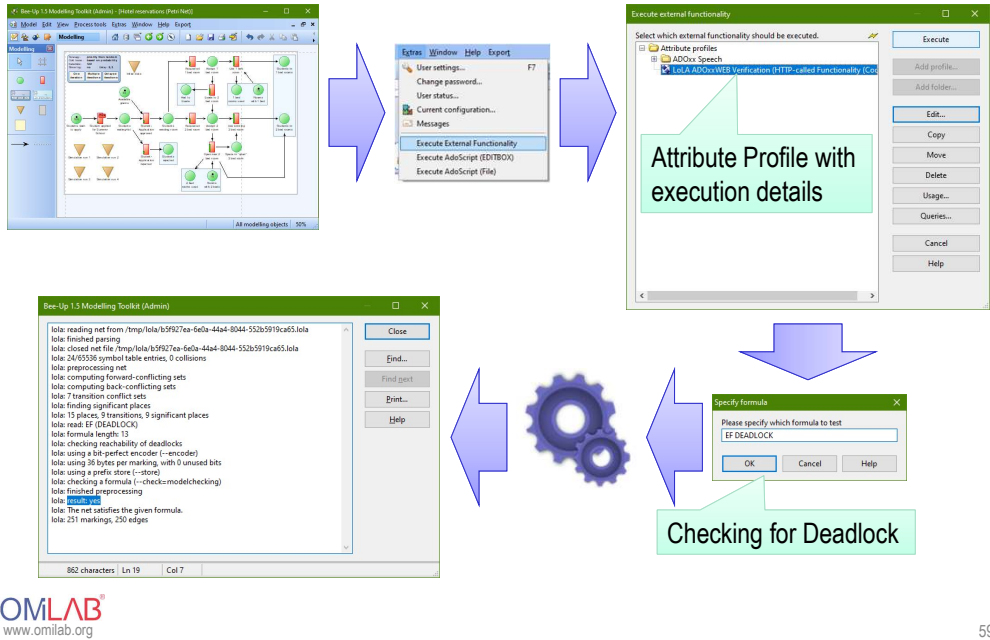
Endpoint:
http://www.adoxx.org/LOLAService/services/rest/lolaservice/analyse

Method:
POST

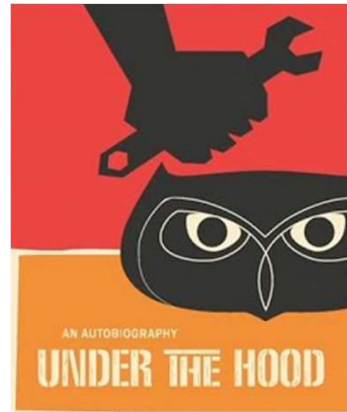
Post-processing:
# Notify the user if the response was anything but a success
IF (val_httpcode=200 OR val_httpcode=200) {
CC "AddScript" ERRORBOX ("The response did't return a \"Success\" code!\n" -
"Returned code: " + STR val_httpcode)
} ELSE {
SETL map_resbody(eval(str_resbody))
CC "AddScript" VIEWBOX text(map_resbody["result"])
}
    
```

Execution details defined in an object (Attribute Profile) in the Bee-Up tool.

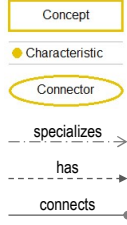
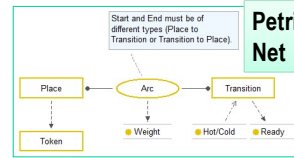
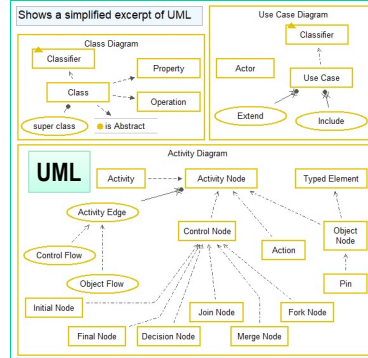
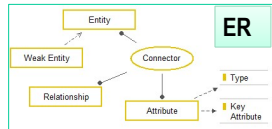
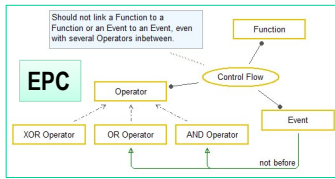
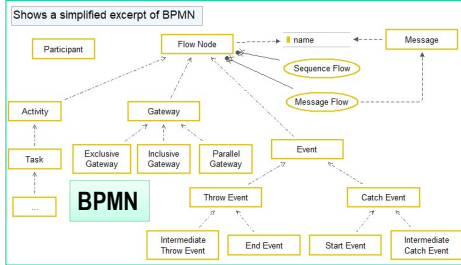
Using an External Service – Execution



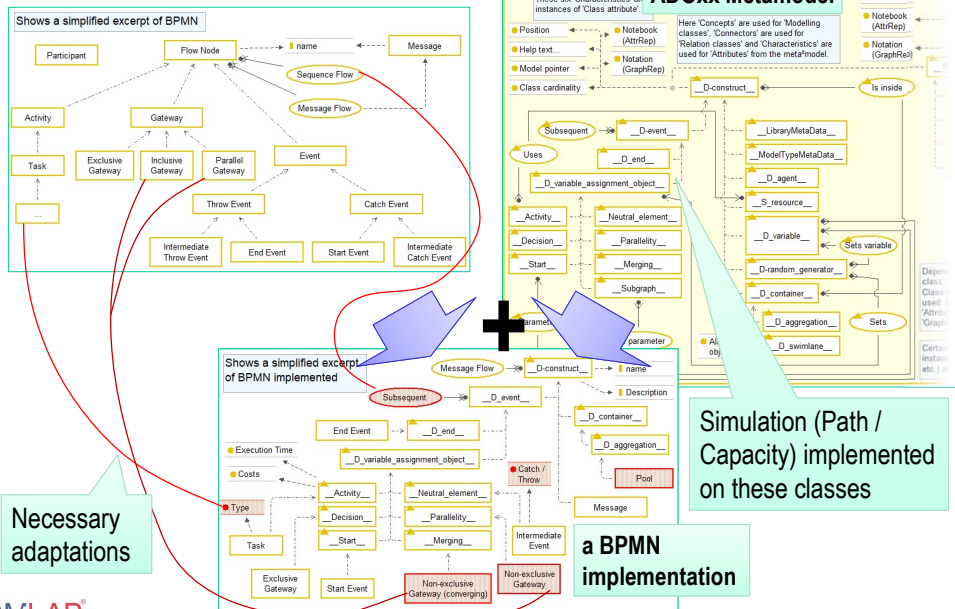
What makes Bee-Up tick and how
UNDER THE HOOD



Under the Hood – Modelling Languages



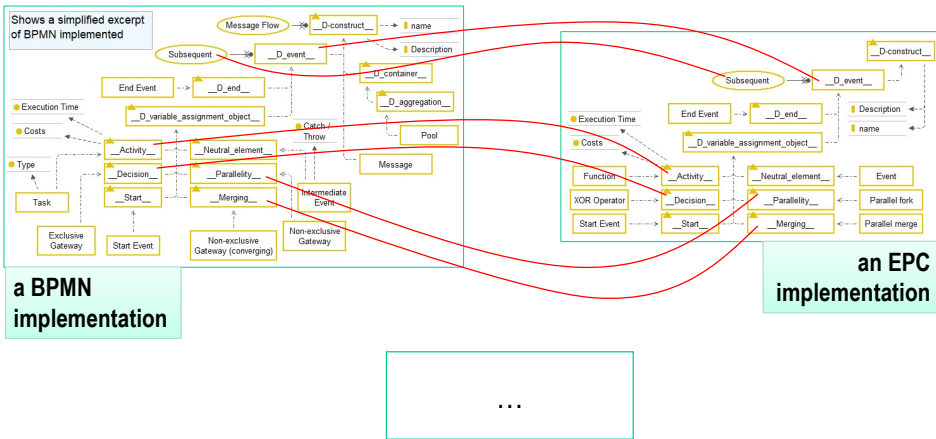
Under the Hood – Modelling Languages fitting into ADOxx



OMILAB
www.omilab.org

Under the Hood – Fitting BPMN and EPC and ...

Use same (super-)classes to allow uniform simulation



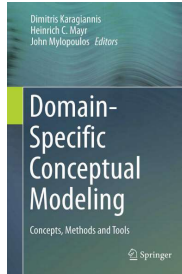
a BPMN implementation

an EPC implementation

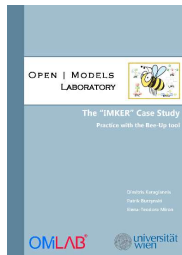
Self-control questions

- Which modelling languages does Bee-Up comprise?
- How do the modelling languages in Bee-Up work?
- How to query information
- How to run a capacity analysis
- How to simulate a Petri Net
- How to derive SQL code from an ER diagram
- How to perform a RDF export
- How to extend Bee-Up using external services

Craving for more?



Detailed information on the conceptual background are also available in D. Karagiannis, R. Buchmann, P. Burzynski, U. Reimer, M. Walch (2016) “*Fundamental Conceptual Modeling Languages in OMILAB*” in “*Domain-Specific Conceptual Modeling*”, Springer (2016)



Check out the “IMKER” case study on <http://www.omilab.org/bee-up> under Publications

Download Bee-Up for free at:
<http://www.omilab.org/bee-up>

