www.omilab.org

Execute 

OMiLAB Training Module 2

Fundamental Conceptual Modelling Languages
using Bee-Up

Theoretical Foundations

Agenda

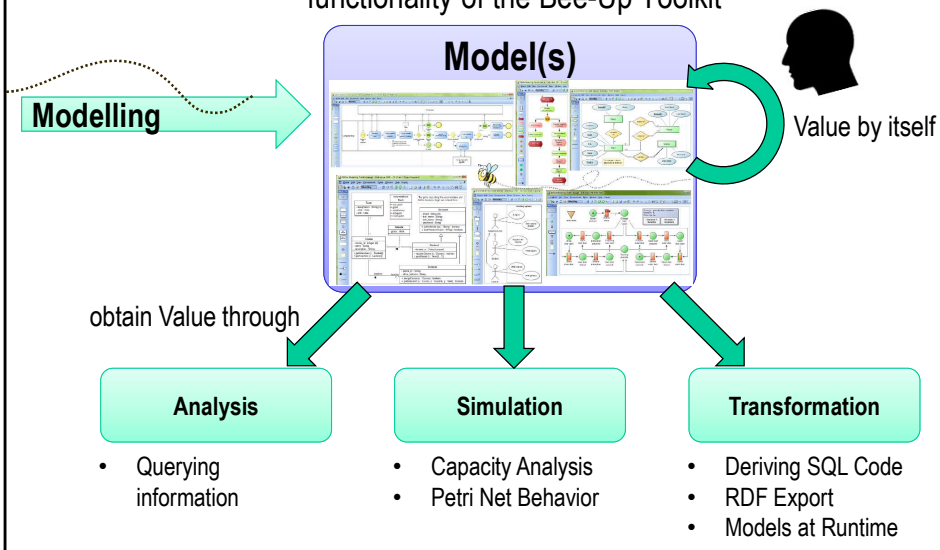
1. Learning Goals
2. Motivation for Models
3. What is Bee-Up
4. Why Bee-Up
5. Modelling Languages in Bee-Up
6. Modelling in Bee-Up
7. Lessons Learned

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 lead 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

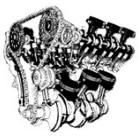


- Querying information
- Capacity Analysis
- Petri Net Behavior
- Deriving SQL Code
- RDF Export
- Models at Runtime

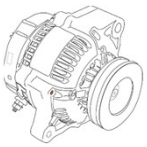
OMiLAB
www.omilab.org

4

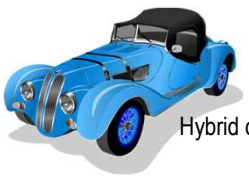
Hybridization – Example



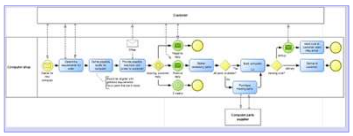
Combustion engine



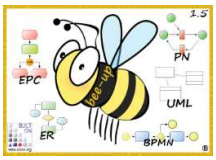
Electric engine



Hybrid car

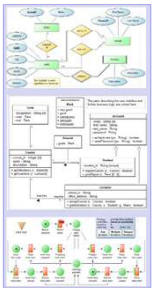


BPMN



EPC

+ ER
+ UML
+ PN
…



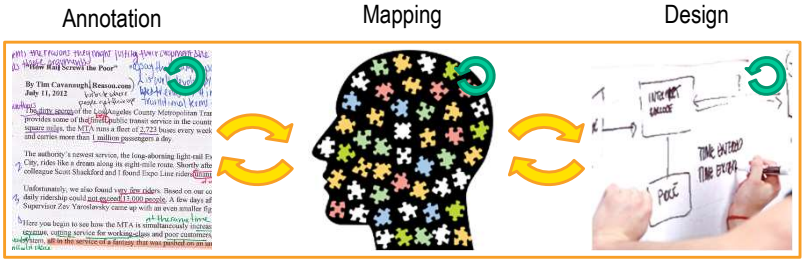
OMiLAB
www.omilab.org

5

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

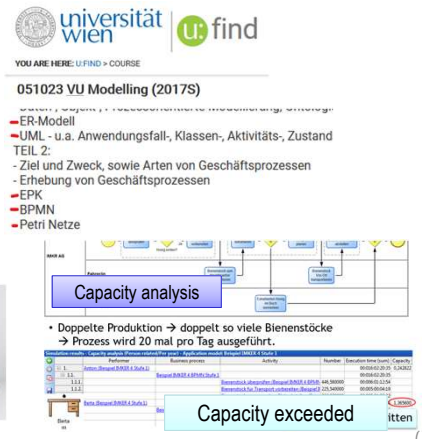


Why Bee-Up exists?

- 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

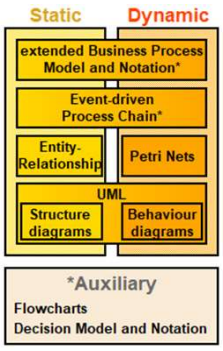
OMiLAB
www.omilab.org

What is Bee-Up? – The Modelling Languages

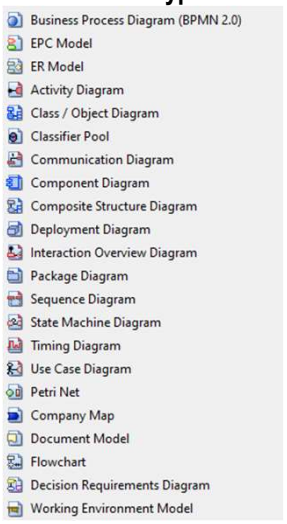
- **B**PMN – Business Process Model and Notation
- **E**PC – Event-driven Process Chains
- **E**R – Entity Relationship

  |
- **U**ML – Unified Modeling Language
- **P**etri Nets

- + Auxiliary
  - DMN
  - Flowcharts
  - …



Actual Model Types:



OMiLAB
www.omilab.org

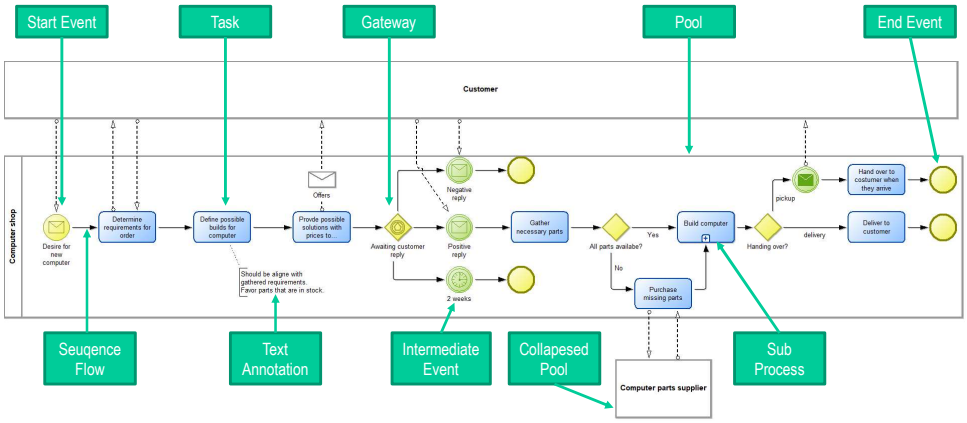
8

MODELING LANGUAGES IN BEE-UP

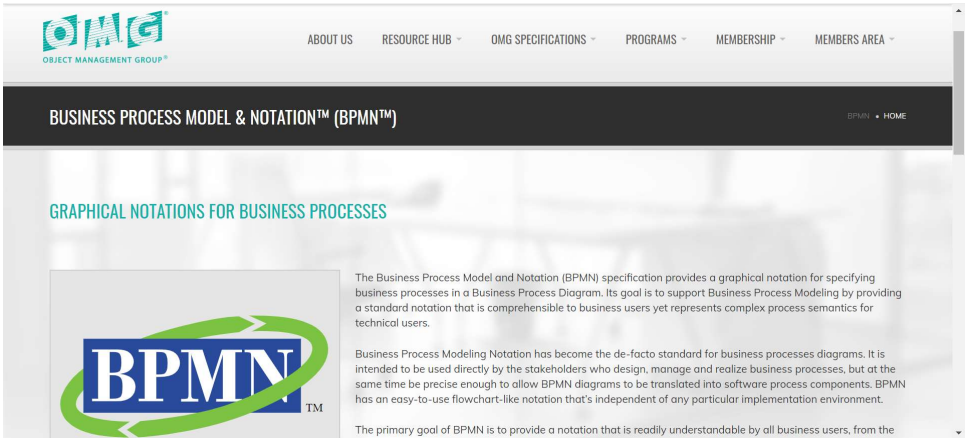
Business Process Model and Notation (BPMN)

- 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


Start Event
Task
Gateway
Pool
End Event
Customer
Computer shop
Desire for new computer
Determine requirements for order
Define possible builds for computer
Offers
Provde possible solutions with prices to...
Should be aligne with gathered requirements. Favor parts that are in stock.
Awaiting customer reply
Negative reply
Positive reply
2 weeks
Gather necessary parts
All parts available?
Yes
No
Purchase missing parts
Build computer
Handing over?
pickup
delivery
Hand over to costumer when they arrive
Deliver to customer
Seuqence Flow
Text Annotation
Intermediate Event
Collapesed Pool
Computer parts supplier
Sub Process

BPMN – Further Information

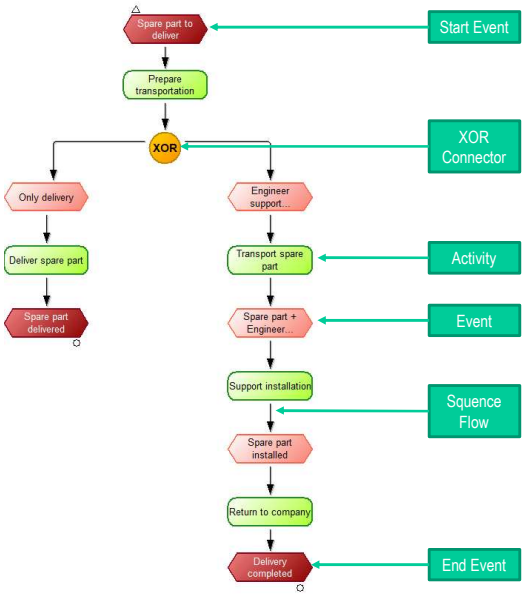


https://www.omg.org/bpmn/index.htm & http://www.bpmn.org/

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



OMiLAB
www.omilab.org

14

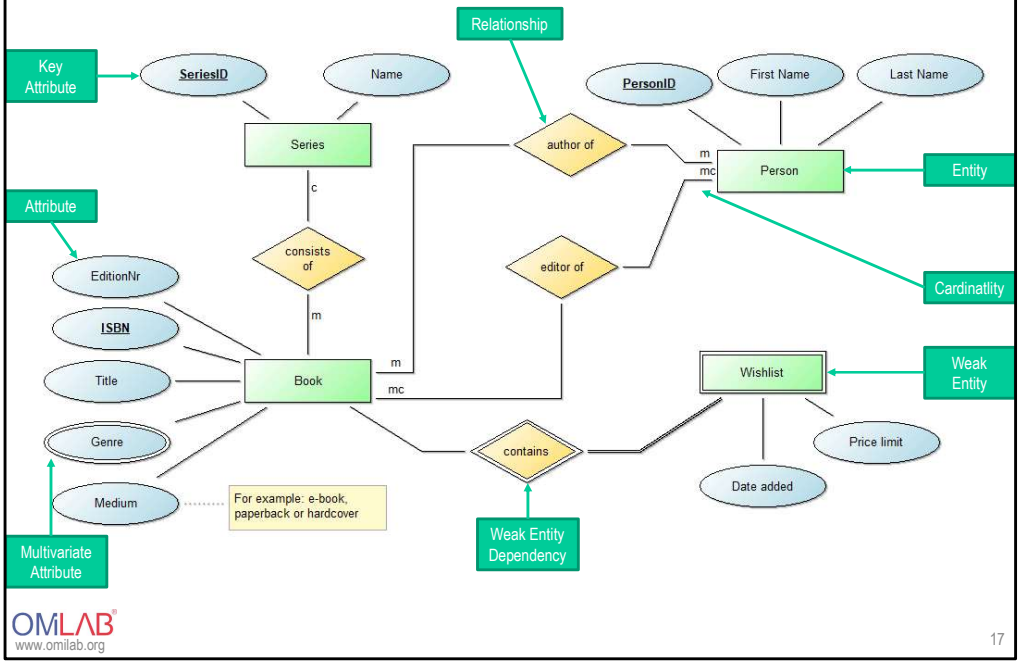
EPC – Further Information

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

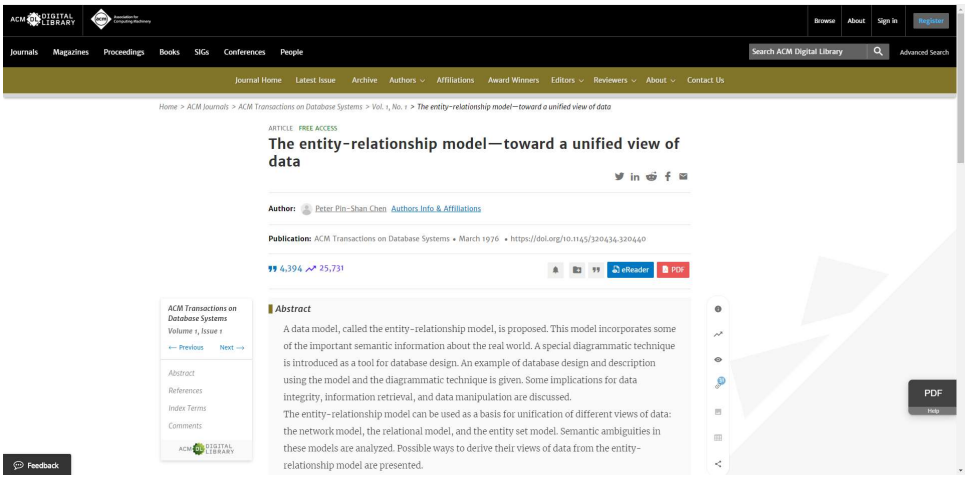
Entity Relationship Model (ER)

- Entity Relatinship 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


Relationship
Key Attribute
SeriesID
Name
PersonID
First Name
Last Name
Series
author of
m
mc
Person
Entity
Attribute
c
consists of
editor of
Cardinatlity
EditionNr
ISBN
m
Book
m
mc
Wishlist
Weak Entity
Title
Genre
contains
Price limit
Date added
Medium
For example: e-book, paperback or hardcover
Weak Entity Dependency
Multivariate Attribute
OMLAB
www.omilab.org
17

ER – Further Information



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

Unified Modeling Language (UML)

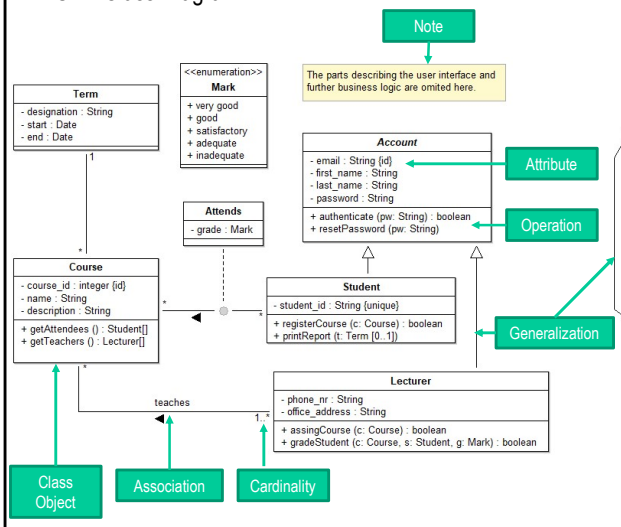
- Object-oriented concepts were first used in the 1960s in the programming language SIMULA.
- Object-oriented programming languages first became widespread through Smalltalk, Ada, C++ and Java (see graphic in section "2.3.4 Object-oriented models").
- Object-oriented analysis and design methods from 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

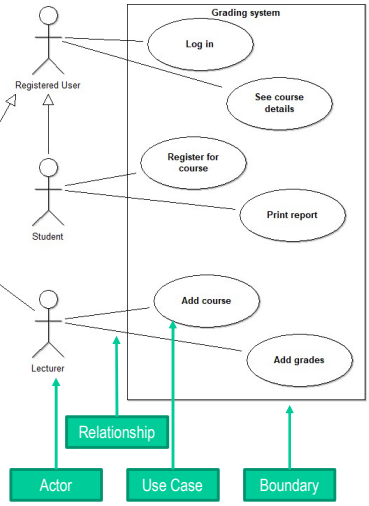
Structure diagrams (static structures: class structures up to structuring of entire systems and architectures)	Behaviour diagrams (sequences between static parts)
• Class Diagram • Object Diagram • Component Diagram • Profile Diagram • Compositional Structure Diagram • Distribution Diagram • Package Diagram	• Activity Diagram • Use Case Diagram • Status Diagram • Sequence Diagram • Interaction Overview Diagram • Communication Diagram • Time Diagram

UML

UML Class Diagram



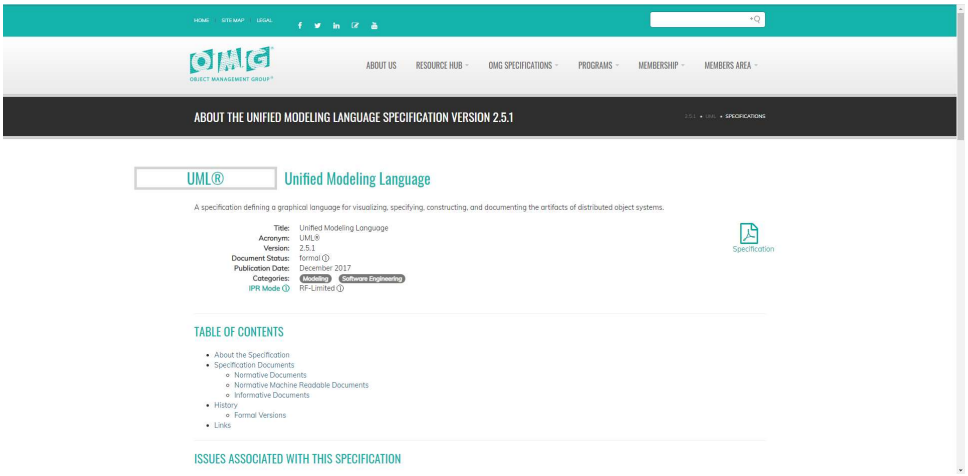
UML Use Case Diagram



OMiLAB
www.omilab.org

21

UML – Further Information

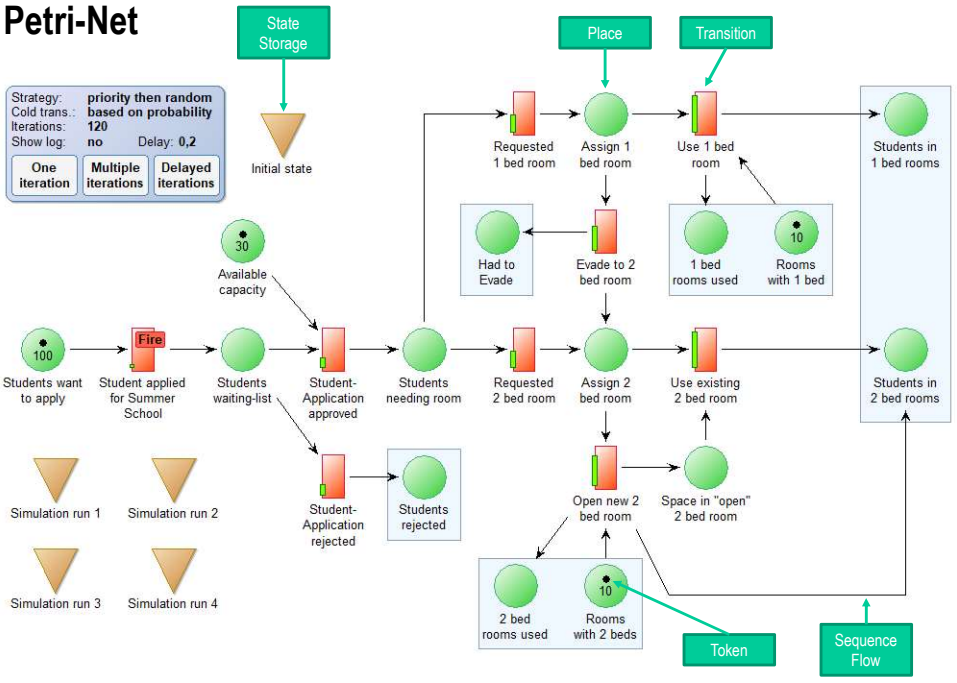


https://www.omg.org/spec/UML/About-UML/ & https://www.uml.org/

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

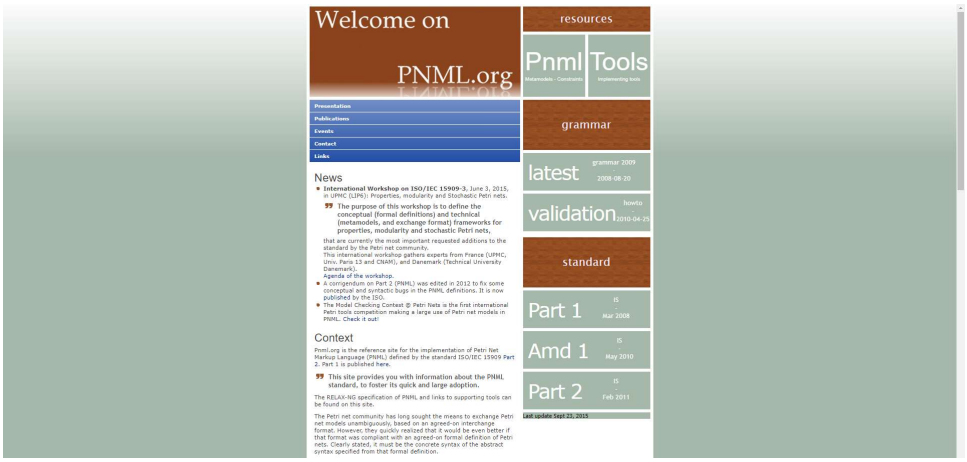
Petri-Net



OMLAB
www.omilab.org

24

Petri-Net – Further Information

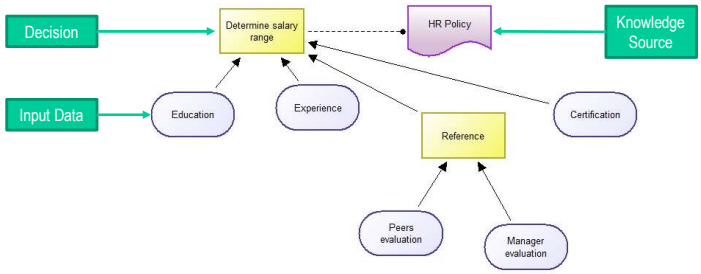


http://www.pnml.org/

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

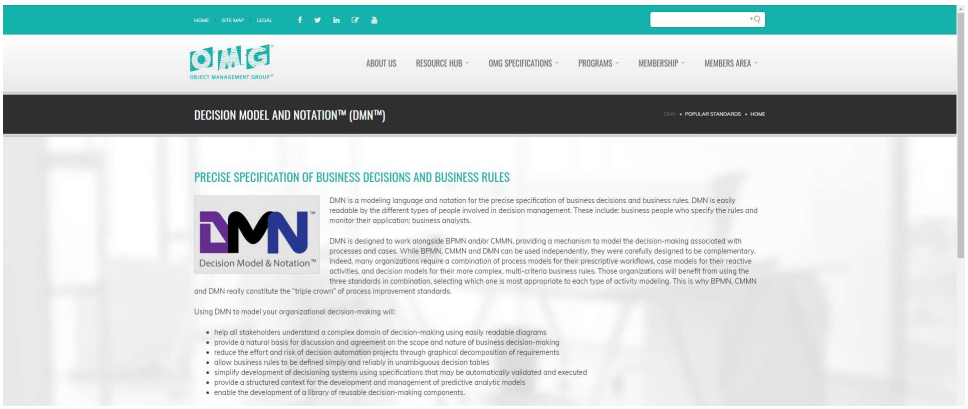


Based on https://why-what-how.com/lms/mod/page/view.php?id=578

OMiLAB
www.omilab.org

27

DMN – Further Information

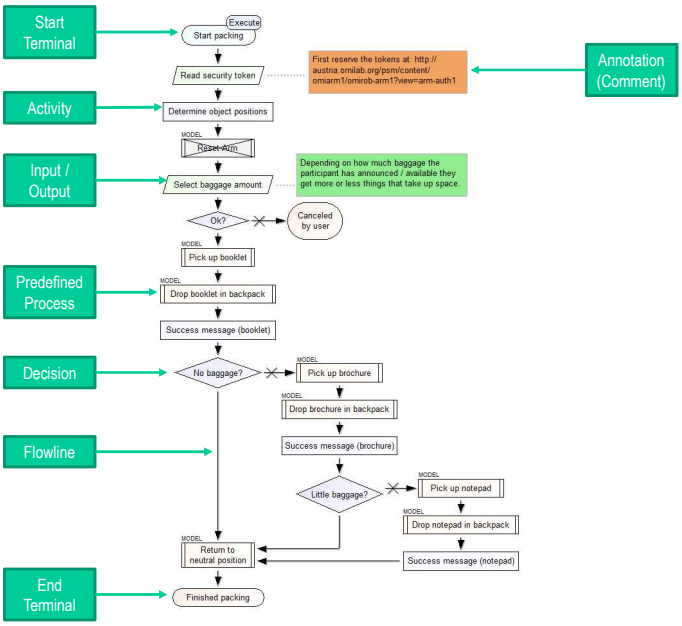


https://www.omg.org/dmn/

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



OMiLAB
www.omilab.org

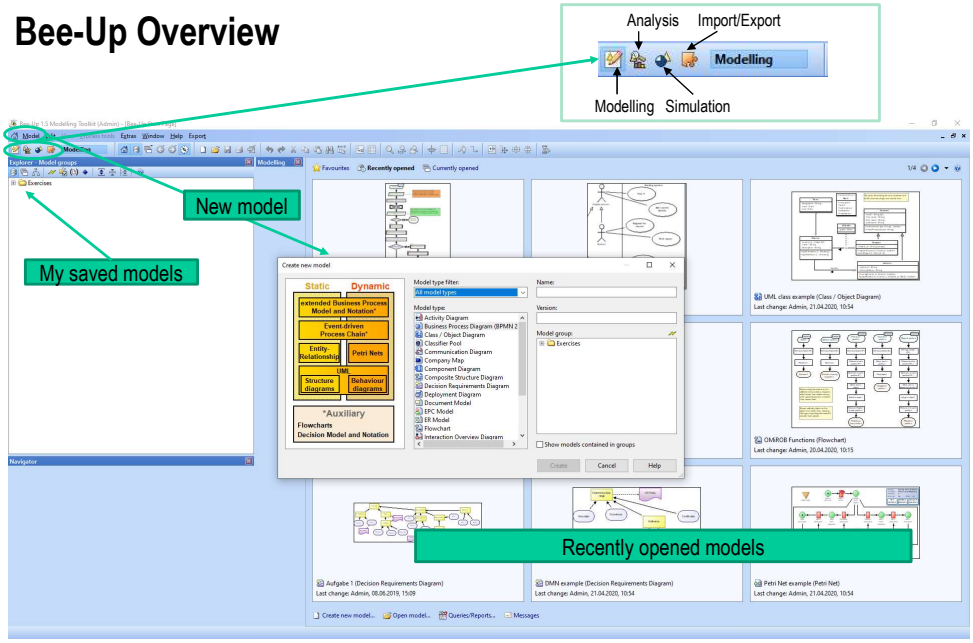
30

Flowchart – Further Information

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

MODELLING IN BEE-UP

Bee-Up Overview

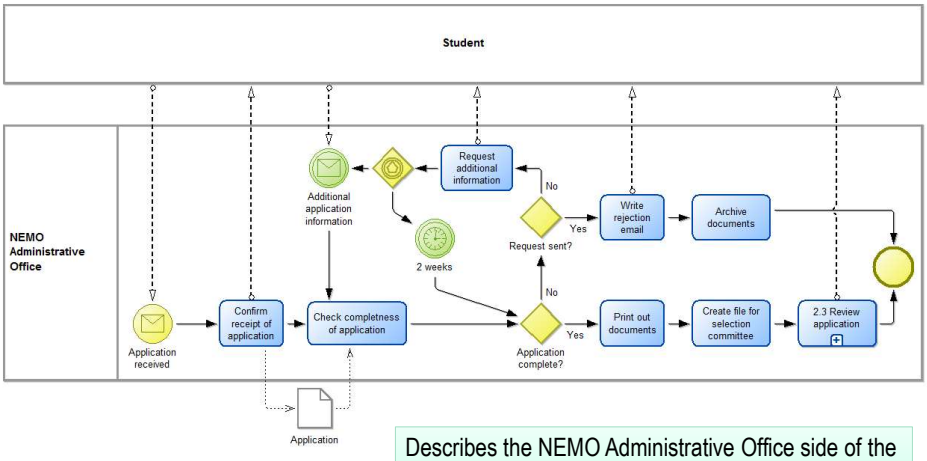


OMiLAB
www.omilab.org

Find out who handles application data

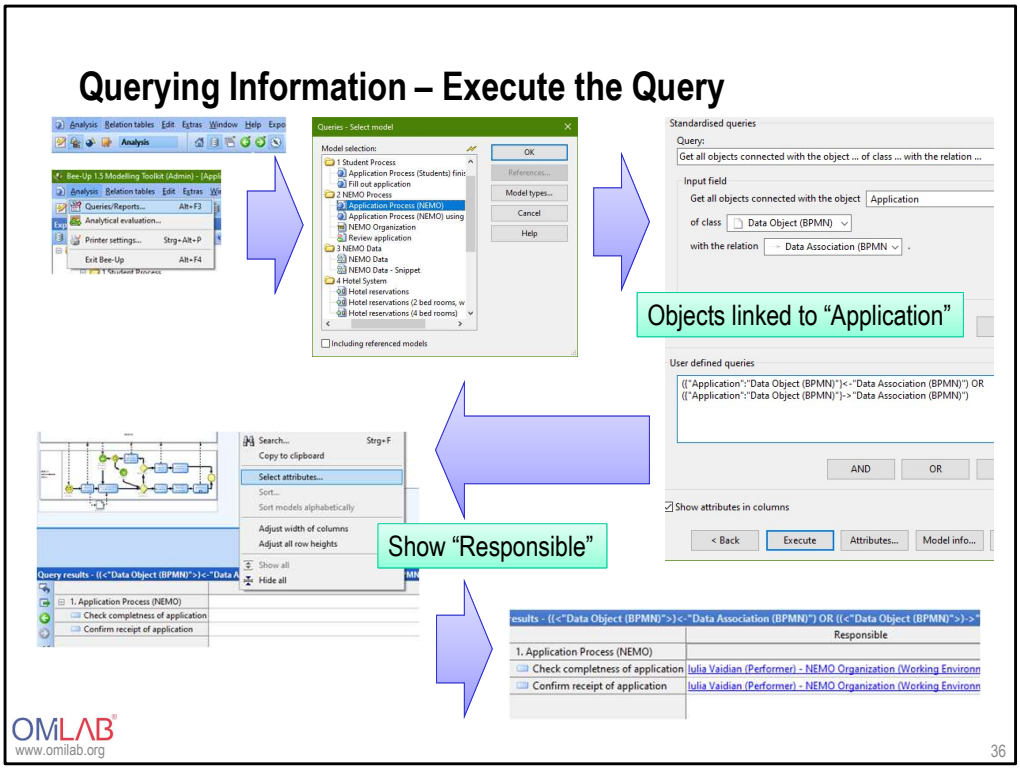
QUERYING INFORMATION

Querying Information – Used Model



Describes the NEMO Administrative Office side of the application submission

OMiLAB
www.omilab.org



Query:

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

({"Application":"Data Object (BPMN)"}->"Data Association (BPMN)")

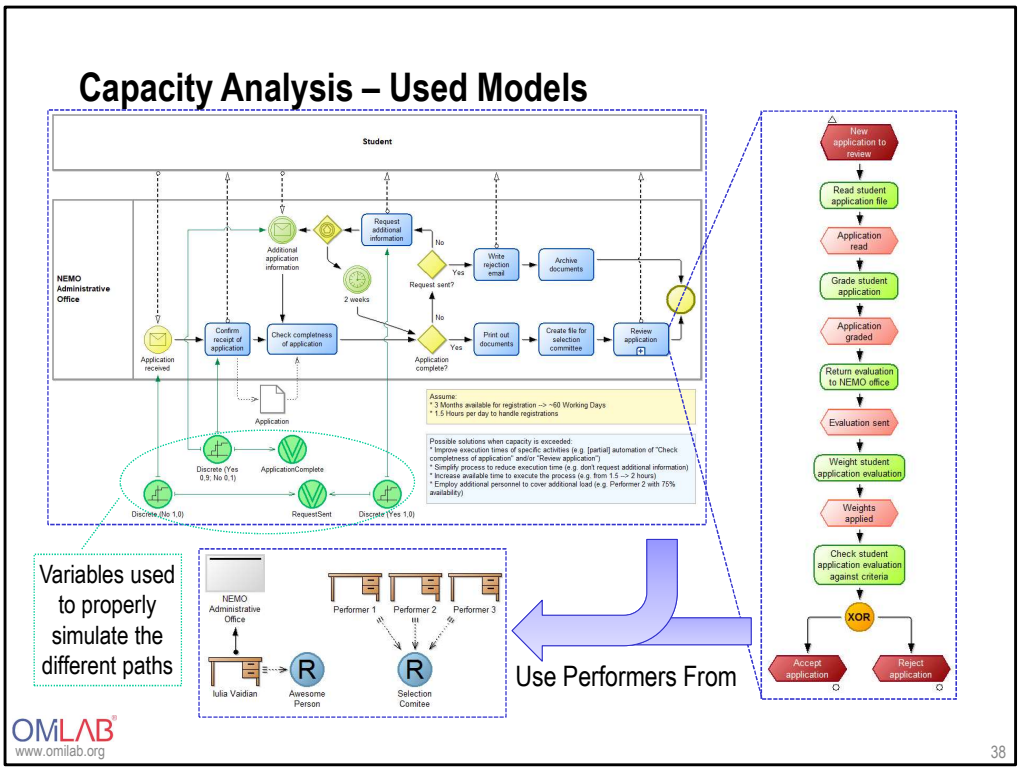
Alternative Query:

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

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

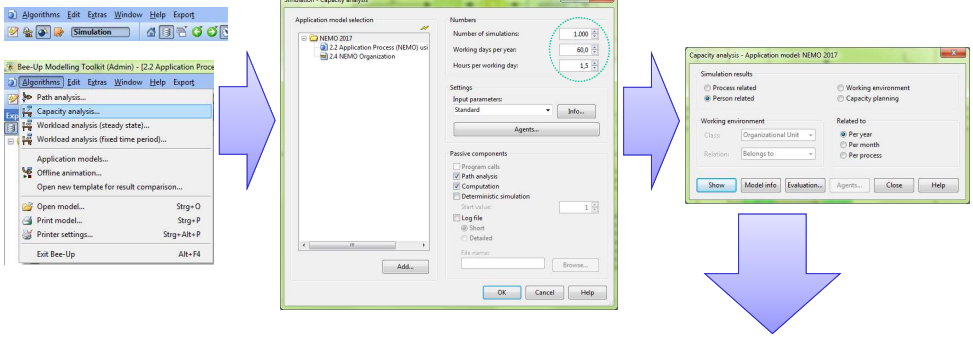
Determine load on employees for a process

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)				00:044:01:03:36	0,745111
1.1.		2.2 Application Process (			00:044:01:03:36	
1.1.1.			Print out documents (2.2 Application Proce	192,400000	00:004:00:24:48	
1.1.2.			Create file for selection committee (2.2 App	192,400000	00:008:00:49:36	
1.1.3.			Request additional information (2.2 Applica	17,800000	00:000:00:35:36	
1.1.4.			Confirm receipt of application (2.2 Applicat	200,000000	00:002:00:20:00	
1.1.5.			Write rejection email (2.2 Application Proce	7,600000	00:000:00:15:12	
1.1.6.			Archive documents (2.2 Application Proces	7,600000	00:000:00:15:12	
1.1.7.			Check completness of application (2.2 Appl	212,800000	00:011:01:14:00	
1.1.8.			Weight student application evaluation (2.3 I	192,400000	00:010:01:02:00	
1.1.9.			Check student application evaluation again	192,400000	00:006:00:37:12	
2.	Performer 1 (2.4 NEMO Organization)				00:010:01:04:48	0,178667
2.1.		2.2 Application Process (			00:010:01:04:48	
2.1.1.			Read student application file (2.3 Review ap	56,600000	00:006:00:26:00	

~75%

C

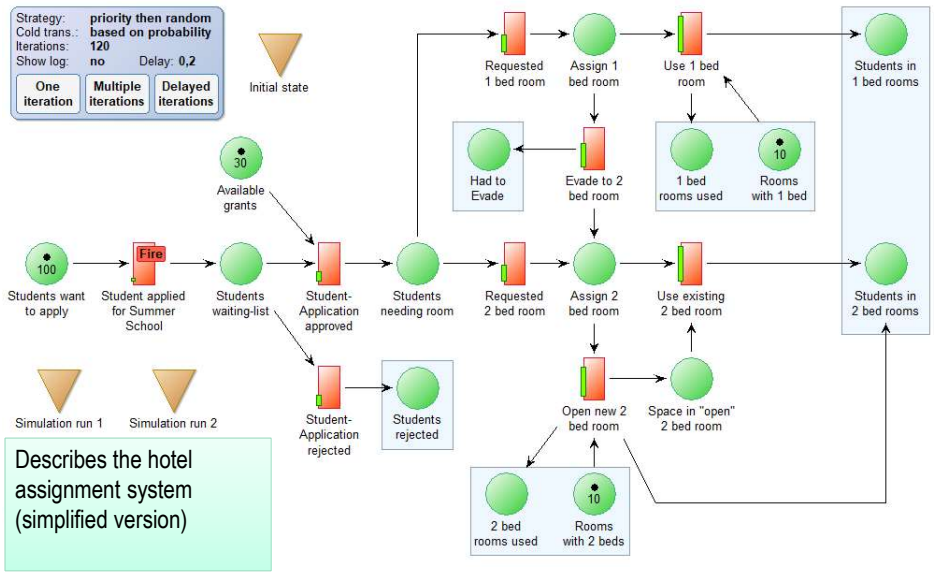
OMiLAB
www.omilab.org

39

Analyze behavior of the modelled system

PETRI NET SIMULATION

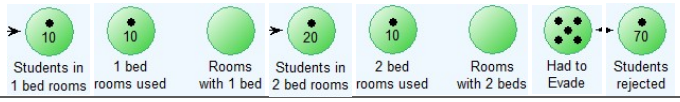
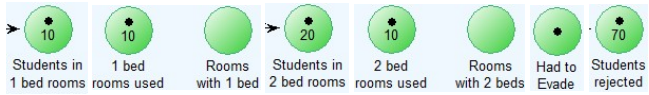
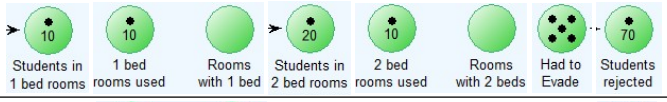
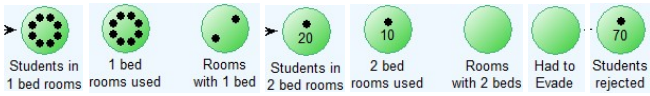
Petri Net Simulation – Used Model



OMiLAB
www.omilab.org

41

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

OMiLAB
www.omilab.org

42

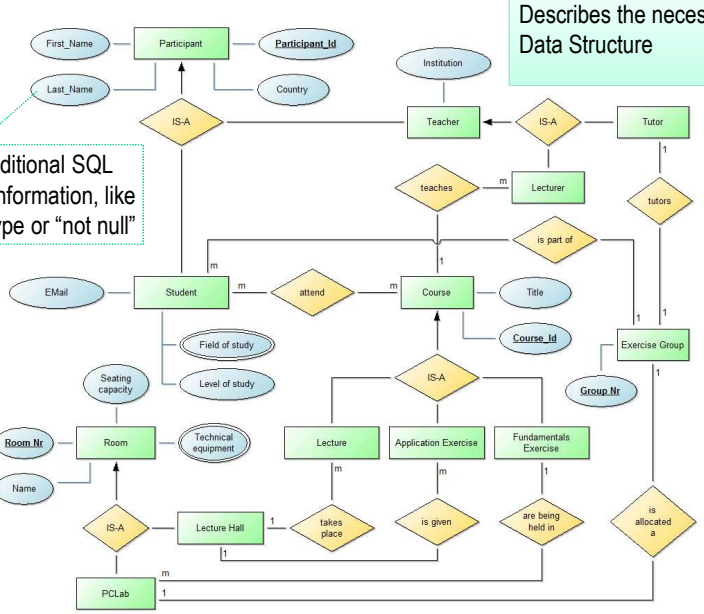
Automatically generate SQL code from a model describing the data structure

DERIVING SQL

Deriving SQL – Used Model

Describes the necessary NEMO Data Structure

Has additional SQL relevant information, like the datatype or “not null”



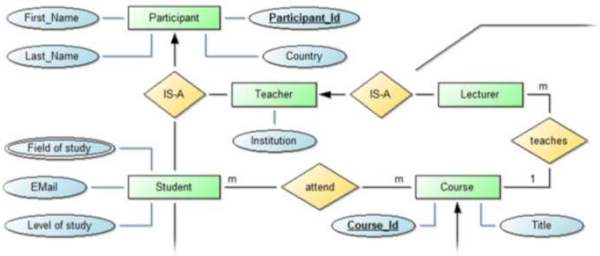
OMiLAB
www.omilab.org

44

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);
```



OMiLAB
www.omilab.org

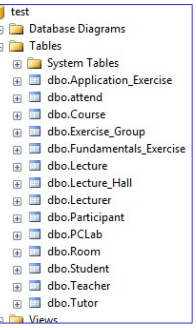
Deriving SQL – Running the Code

Using MS SQL Server Management Studio

```
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
```

Messages

```
Command(s) completed successfully.
```



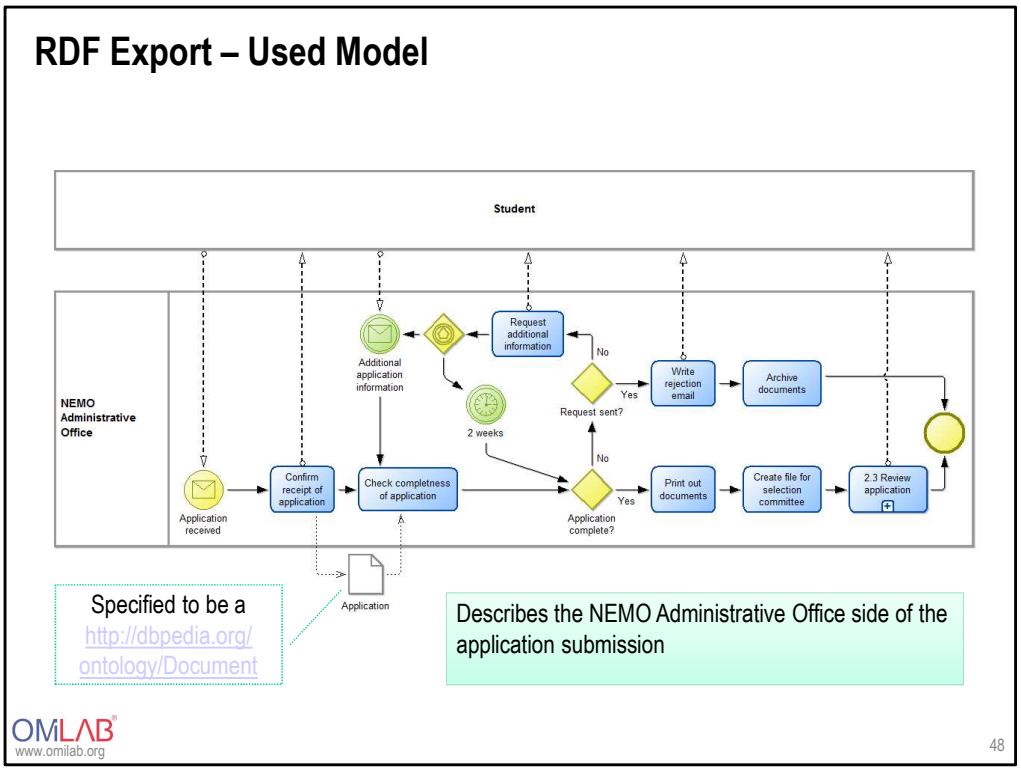
OMiLAB
www.omilab.org

46

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



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_URI       <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 "00:000:00:01:00" ;
    mm:a_For_compensation "No" ;
    mm:a_Global_task    "No" ;
    mm:a_Loop_type      "Not specified" ;
    mm:a_Monitoring     "No" ;
    mm:a_Resting_time   "00:000:00:00:00" ;
    mm:a_Task_type      "Not specified" ;
    mm:a_Transport_time "00:000:00:00:00" ;
    mm:a_Waiting_time   "00:000:00:00:00" ;
    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-Iulia_Vaidian> ;
    cv:a_Name           "Confirm receipt of application" .

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

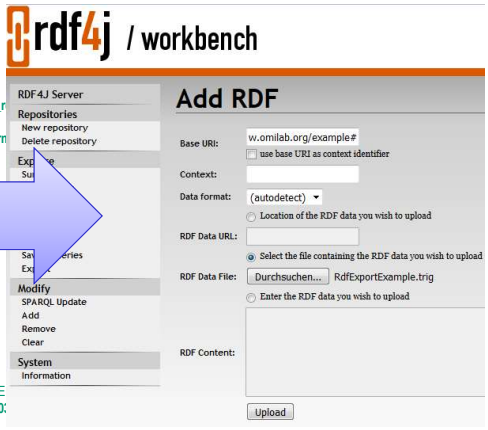
OMiLAB
www.omilab.org

49

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

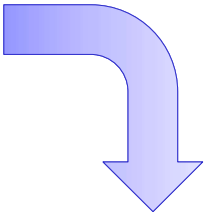
RDF Export – TriG Code Snippet

```
<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_URI        <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   "00:000:00:01:00" ;
    mm:a_For_compensation   "No" ;
    mm:a_Global_task      "No" ;
    mm:a_Loop_type        "Not specified" ;
    mm:a_Monitoring       "No" ;
    mm:a_Resting_time     "00:000:00:00:00" ;
    mm:a_Task_type        "Not specified" ;
    mm:a_Transport_time   "00:000:00:00:00" ;
    mm:a_Waiting_time     "00:000:00:00:00" ;
    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-Iulia_Vaidian>
    cv:described_in  <http://www.omilab.org/example#Working_Environment_Model-2_4_NEMO_Organization> .
}
```



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/registration#Application>	<http://www.omilab.org/example#Performer-37003-Julia_Vaidian>	"Julia Vaidian"

OMiLAB
www.omilab.org

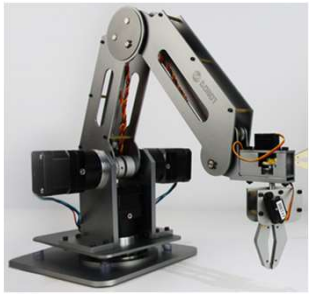
51

Packing the backpack using a robotic arm

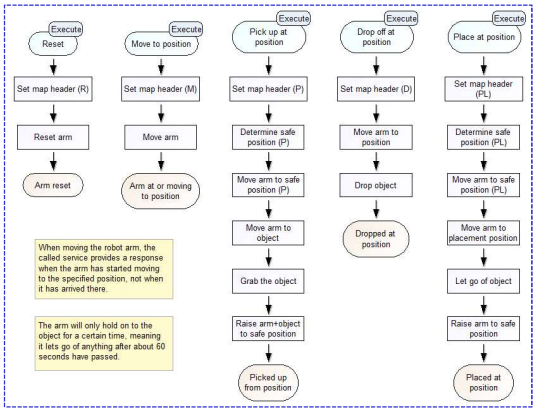
MODELS AT RUNTIME

Transformation

Models for Runtime – Used Models



Described Through

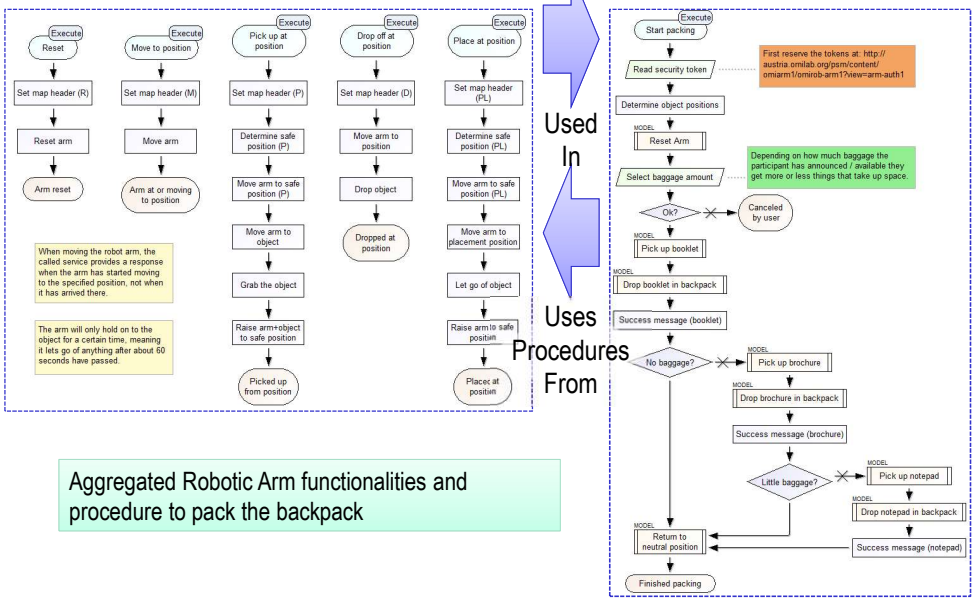


Available capabilities:

- Rotate
- Grab on
- Grab off

A Model aggregating basic capabilities to useful functionalities

Models at Runtime – Used Models

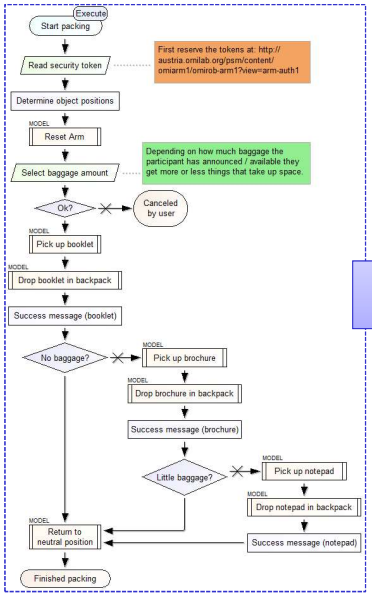


Aggregated Robotic Arm functionalities and procedure to pack the backpack

OMiLAB
www.omilab.org

54

Models at Runtime – Execution



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

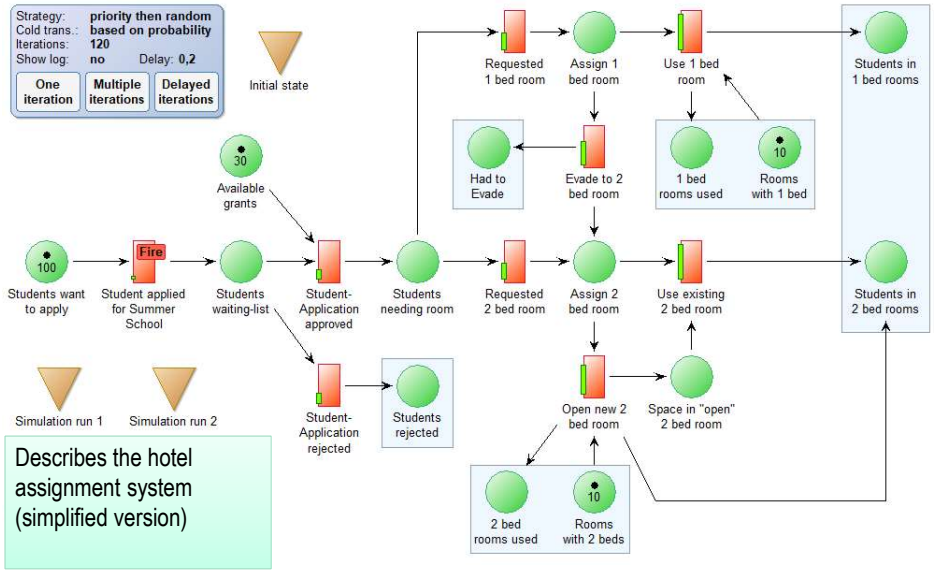
OMiLAB
www.omilab.org

55

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

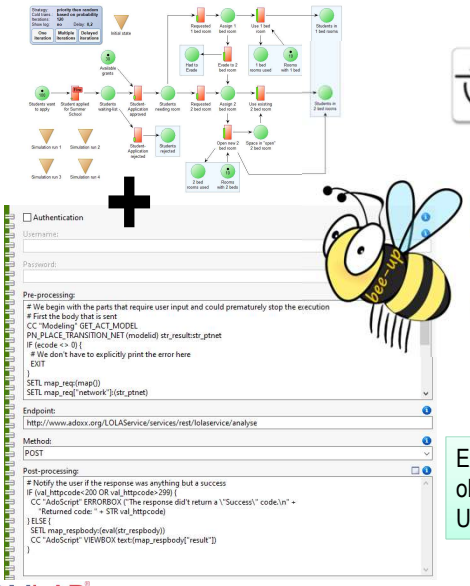
EXTENDING BEE-UP

Using an External Service – Used Model



OMiLAB
www.omilab.org

Using an External Service – Setup



LoLA Petri Net Verification

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

The implementation depends on conc anguage.

REST service call

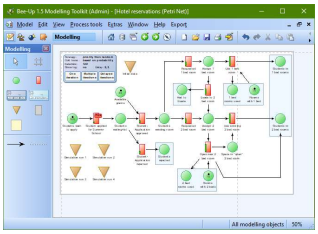
Server running Web-Service

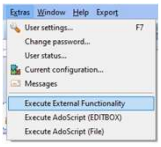
Execution details defined in an object (Attribute Profile) in the Bee-Up tool.

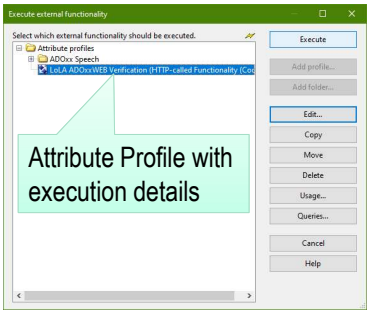
OMiLAB
www.omilab.org

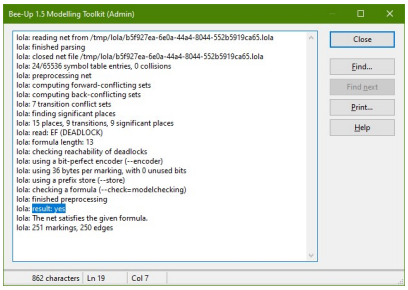
58

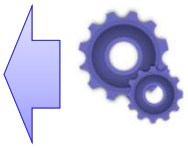
Using an External Service – Execution

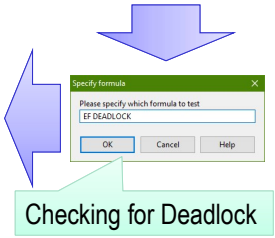








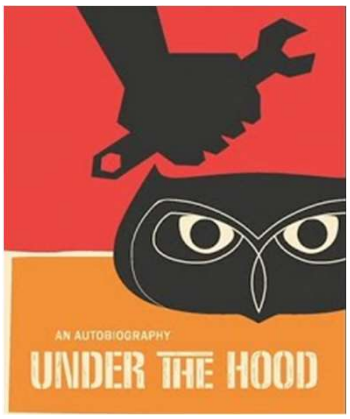




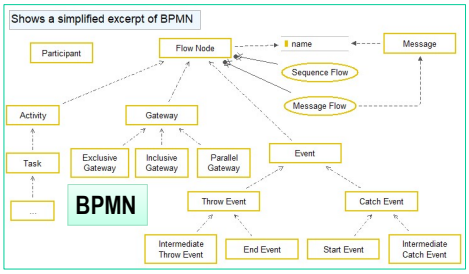
OMiLAB
www.omilab.org

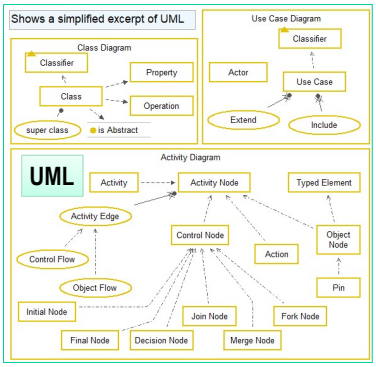
What makes Bee-Up tick and how

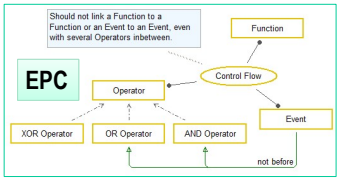
UNDER THE HOOD

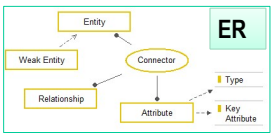


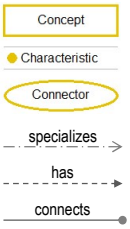
Under the Hood – Modelling Languages

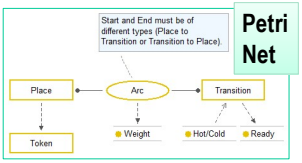








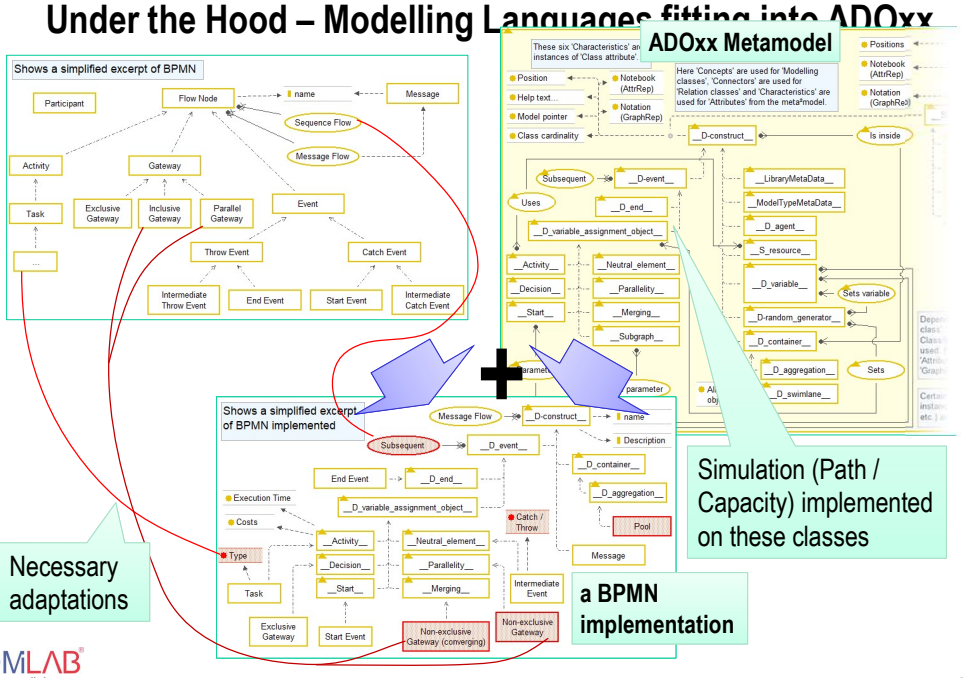




OMiLAB
www.omilab.org

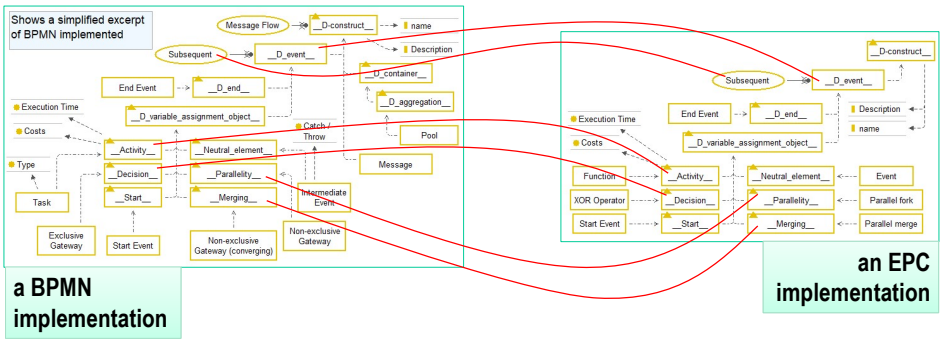
61

Under the Hood – Modelling Languages fitting into ADOxx



Under the Hood – Fitting BPMN and EPC and …

Use same (super-)classes to allow uniform simulation



a BPMN implementation

an EPC implementation

…

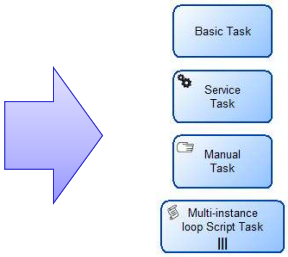
OMiLAB
www.omilab.org

63

Under the Hood – Deciding additional details

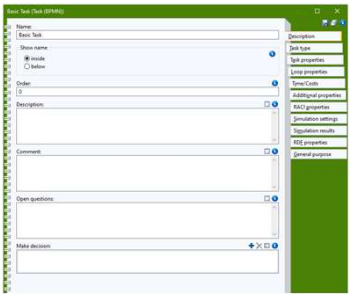
```
# Draw the gradient
CLIP_ROUNDRECT x:(tabx0) y:(taby0) w:(tabw) h:(tabh) rx:0.2cm ry:0.2cm
GRADIENT_RECT x:(tabx0) y:(taby0) w:(tabw) h:(tabh) style:downdiag
              color1:(col1FFm) color2:(col1FTo)
CLIP_OFF

# Draw the basic shape
IF (tasktype = "Call activity") {
  PEN w:0.12cm color:(col1Brd)
} ELSE {
  PEN w:0.06cm color:(col1Brd)
}
FILL style:null
ROUNDRECT x:(tabx0) y:(taby0) w:(tabw) h:(tabh) rx:0.2cm ry:0.2cm

# Draw the icon denoting the type
IF (tasktype = "Send") {
  BITMAP "db:\\mfb_bpmn20_send_icon.png" x:(tabx0+0.1cm) y:(taby0+0.1cm) w:0.
} ELSIF (tasktype = "Receive") {
  IF (instanciating = "No") {
    BITMAP "db:\\mfb_bpmn20_receive_icon.png" x:(tabx0+0.1cm) y:(taby0+0.1cm)
  } ELSE {
    BITMAP "db:\\mfb_bpmn20_receive_process_icon.png" x:(tabx0+0.1cm) y:(taby
  }
} ELSIF (tasktype = "User") {
  BITMAP "db:\\mfb_bpmn20_user_icon.png" x:(tabx0+0.1cm) y:(taby0+0.1cm) w:0.
} ELSIF (tasktype = "Manual") {
  BITMAP "db:\\mfb_bpmn20_manual_icon.png" x:(tabx0+0.1cm) y:(taby0+0.1cm) w:
} ELSIF (tasktype = "Service") {
  BITMAP "db:\\mfb_bpmn20_service_icon.png" x:(tabx0+0.1cm) y:(taby0+0.1cm) w
} ELSIF (tasktype = "Script") {
```

Graphical Representation




```
NOTEBOOK

AVAL sTaskType:"Task type"
AVAL sLoopCharacteristic:"Loop type"

#------------------
CHAPTER "Description"
ATTR "Name"
ATTR "Show name" # inside, below
ATTR "Order"
ATTR "Description" lines:5
ATTR "Comment" lines:5
ATTR "Open questions" lines:5
ATTR "Make decision" lines:3

#------------------
CHAPTER "Task type"
ATTR "Task type" ctrltype:radio # Not specified, Service, Send, Receive, User,
Manual, Business rule, Script, Call activity
ATTR "Global task" ctrltype:check checked-value:"Yes" unchecked-value:"No"
      enabled:((sTaskType = "User") OR (sTaskType = "Manual") OR (sTaskType =
"Business rule") OR (sTaskType = "Script"))

#------------------
CHAPTER "Task properties"
GROUP "Object relevance"
  ATTR "Auditing" ctrltype:check checked-value:"Yes" unchecked-value:"No"
  ATTR "Monitoring" ctrltype:check checked-value:"Yes" unchecked-value:"No"
```

Attribute Representation

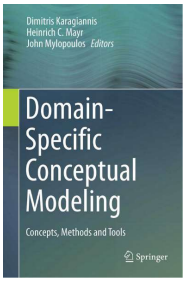


OMiLAB
www.omilab.org

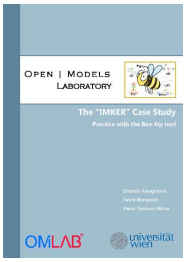
Self-control questions

- 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