www.omilab.org

Foundation

OMiLAB Training Module 4

The Value of Conceptual Models

Learning Goals

- Understand the concept of model value
- Understand the different values that can be enabled by a conceptual model
- Understand the different roles a conceptual model can play in systems analysis and design
- Understand how the value of a conceptual model is dependent on the model engineer on the one side and the modeler on the other

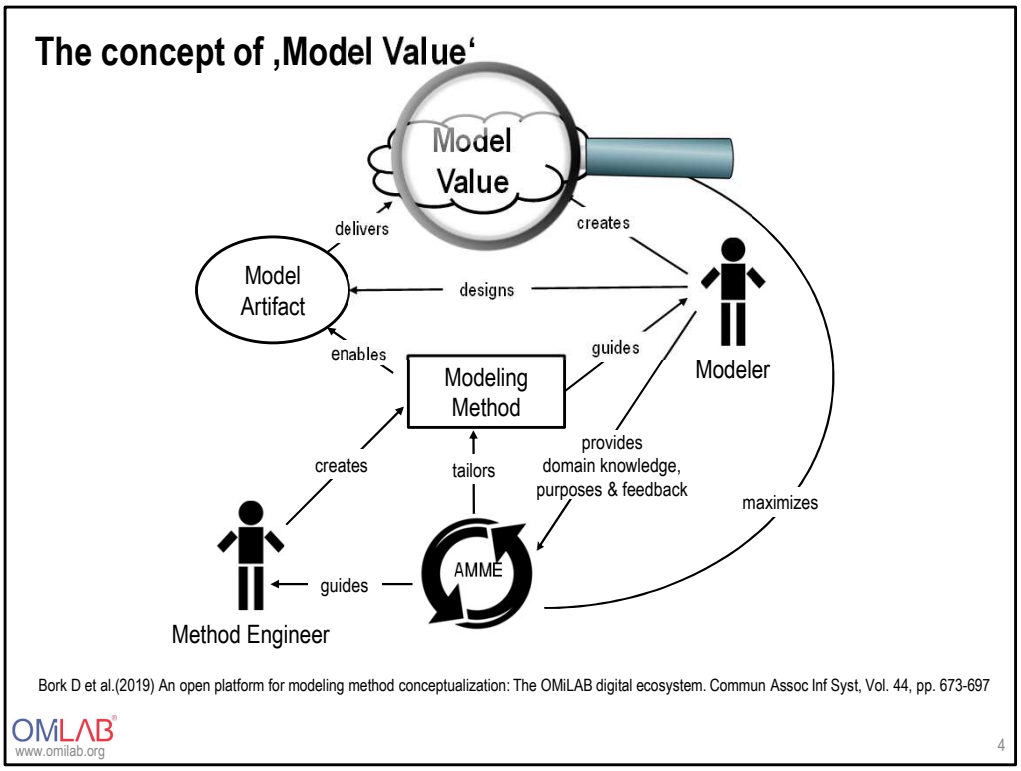
OMiLAB
www.omilab.org

2

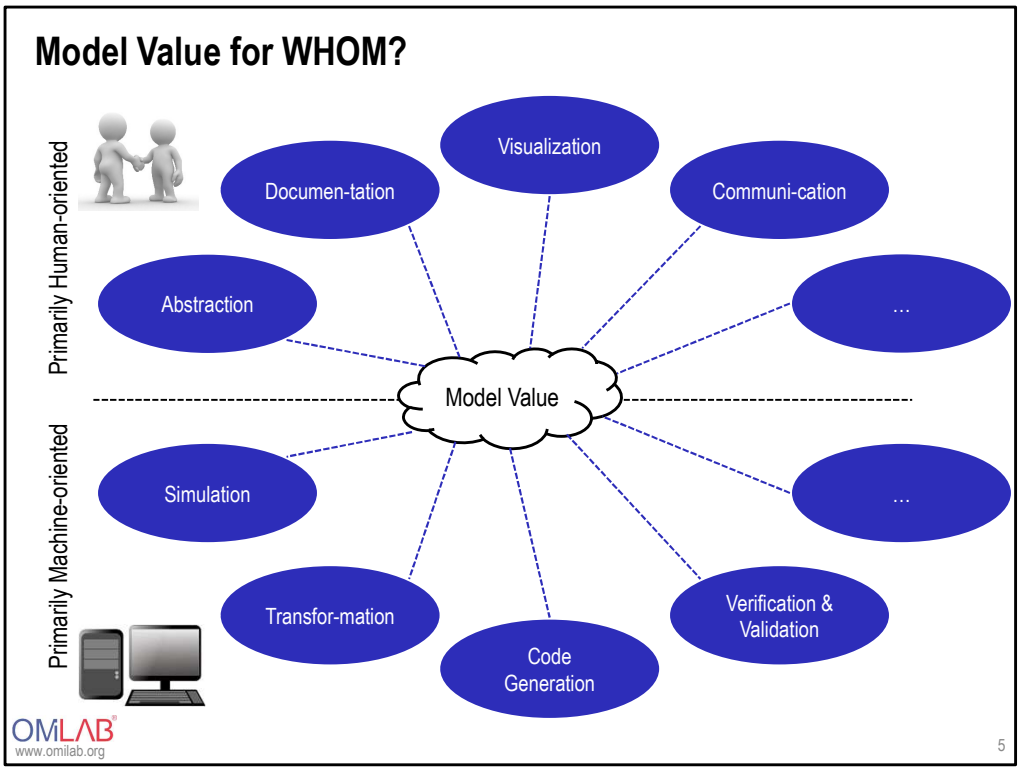
The goal of this training module is to educate in the concept of model value and to show some different forms model value can take.

And whom does it serve?

WHAT IS MODEL VALUE?



- This slide shows the concept of model value and its dependencies
- The concept shows, that model value can only be co-created by the modeler and the method engineer
    - The method engineer creates a modelling method that comprises the seed for model value while being supported/guided by an agile modelling method engineering approach (AMME – to be introduced later).
    - The modeler is guided by this modelling method and designs valid model artifacts with it
    - Only then he or she creates the model value by using the seeded value and instantiating it in the modelling context at hand



- This slide differentiates model value by the stakeholder
  - Humans (the upper hemisphere
  - Machines (the lower hemisphere)
- See the dotted line which should emphasize that the differentiation is not stringent in all cases

Enablers for Human-oriented Model Value

- Because Conceptual Models **apply Abstraction**
  - The complexity of the system under study can be reduced to a cognitively manageable level (in contrast to analysing the original)
    - Model Value: **Abstraction**
    - Model Value: **Documentation**
- Because Conceptual Models are **built with 'Concepts'**
  - Humans have a shared understanding and vocabular (in contrast to using natural language)
    - Model Value: **Communication**
  - Precise and unambiguous documentations can be created (in contrast to using natural language)
    - Model Value: **Documentation**
- Because Conceptual Models are **aiming for human beings**
  - Intuitive representations of abstract and complex structures can be created
    - Model Value: **Visualization**

OMiLAB
www.omilab.org

6

- This slide describes the enables for human-oriented model value
- It explains how different characteristics of conceptual modelling enable specific exemplified model value for humans

Enablers for Machine-oriented Model Value

- Conceptual Modelling methods **can be formalized**
  - The metamodel can be formalized and introduced to a computer system
  - The semantics of the modelling language can be formalized
  - The notation can be standardized
  - Import/export formats can be standardized
  - Mechanisms & algorithms can be implemented on top of the formalized modelling language
- All enables above more of less contribute to the exemplary model values
  - Model Value: **Simulation**
  - Model Value: **Transformation**
  - Model Value: **Coder Generation**
  - Model Value: **Verification & Validation**

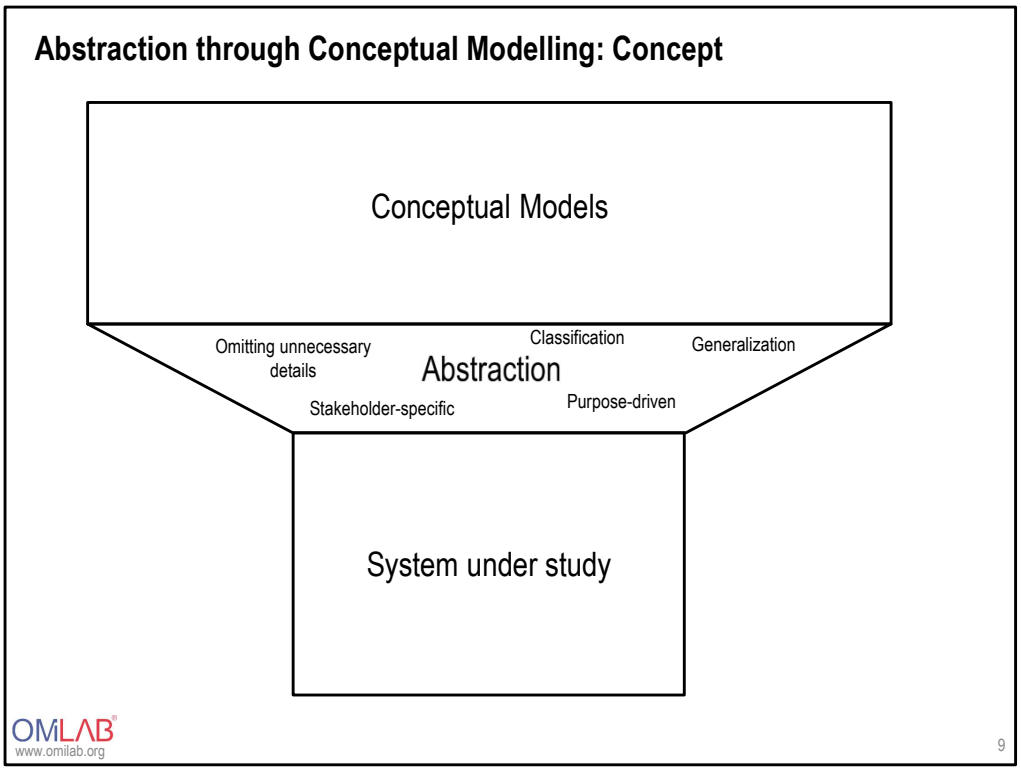
OMiLAB
www.omilab.org

7

- This slide describes the enables for machine-oriented model value
- It explains how different characteristics of conceptual modelling enable specific exemplified model value for machines
- The differentiation is not so clear here as several characteristics need to come together to enable a certain model value

Domain-specific Human-centered abstractions

HOW CAN WE REALIZE MODEL VALUE IN DIFFERENT DOMAINS?



- This slide aims to explain the generic concept of abstraction which comprises
  - At the bottom a system under study, which can be a real system that is observed or a hypotheticla system that is to be designed
  - At the center the concept of abstraction that employs classification, generalization etc.. In order to realize a stakeholder- and purpose-driven reduction of the complexity of the system under study
  - At the top we can then see the product of abstracting from a system under study, the conceptual models
- In the following, this concept is applied and showcased with different disciplines

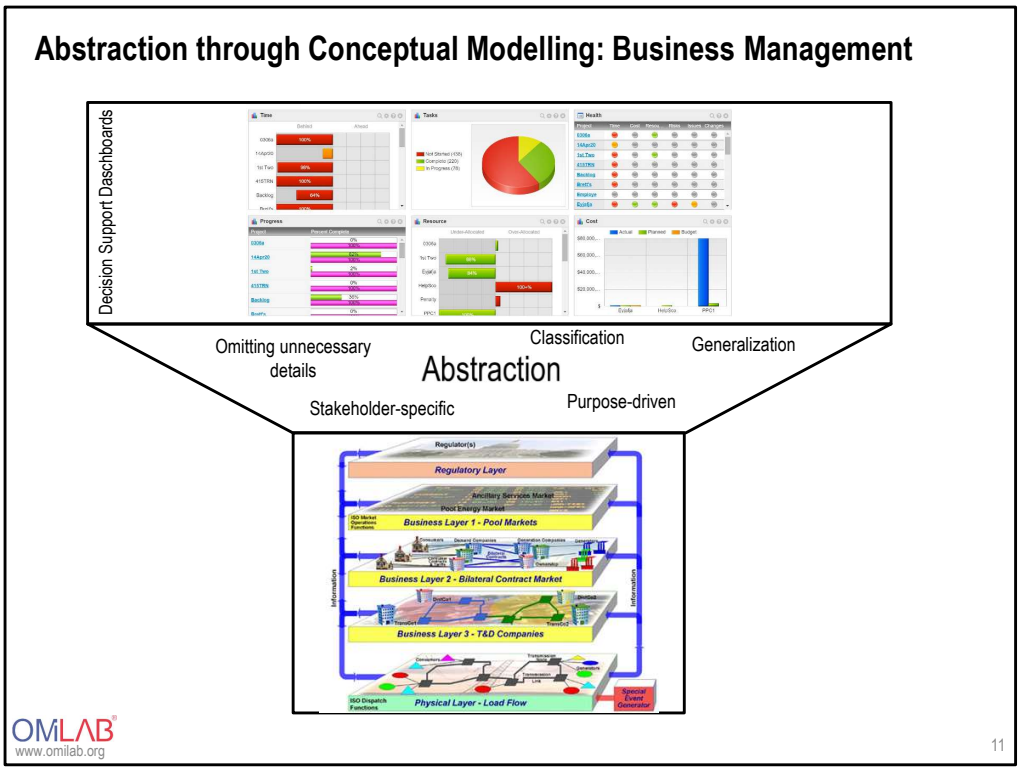
Examples of Conceptual Modelling Abstractions

Discipline	Business Management	Information Science	Information Systems	Computer Science	Engineering Science	Data Science	Artificial Intelligence
Purpose of modelling / Model value	Provide Decision support, e.g., by management dashboards and KPI matrices	Structuring amounts of data by creating a data schema.	Shaping the enterprise processes to increase throughput, quality, and efficiency.	Designing and analysing a software system as an implementation blueprint.	Simulating different production line configurations.	Structuring and integrating heterogeneous data. Models as predictive and explanatory tools.	Knowledge models for comprehension (Explainability problem) and reasoning.
Example Modelling Language Support	Analysis Dashboards based on BPMN, Workflows, and ArchiMate.	ER	BPMN, ArchiMate	UML	Petri Nets	Data Visualization Technique like Alignment Maps	Knowledge Graphs

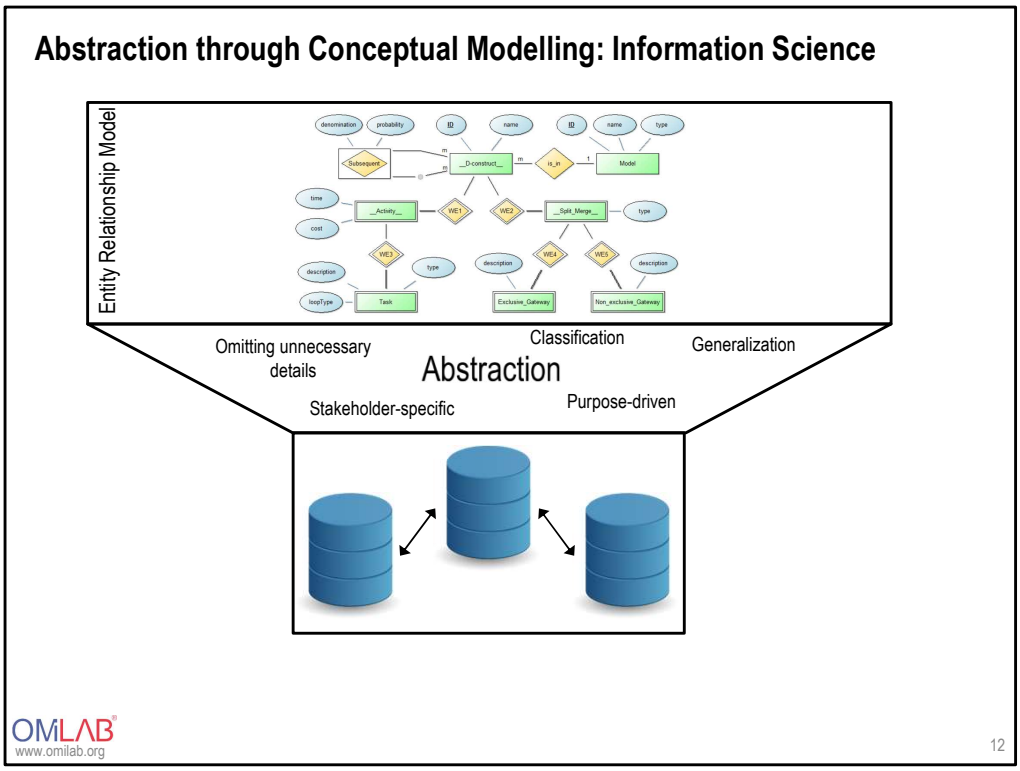
OMiLAB
www.omilab.org

10

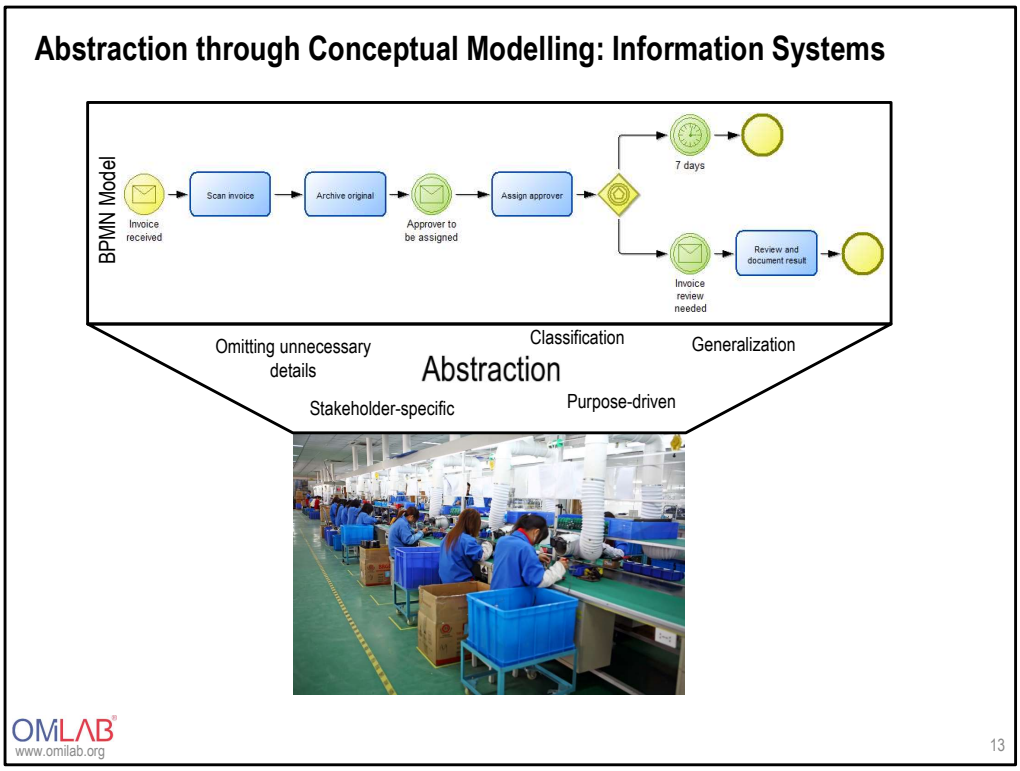
* This slide provides an overview of examples how abstraction is applied in different conceptual modelling disciplines



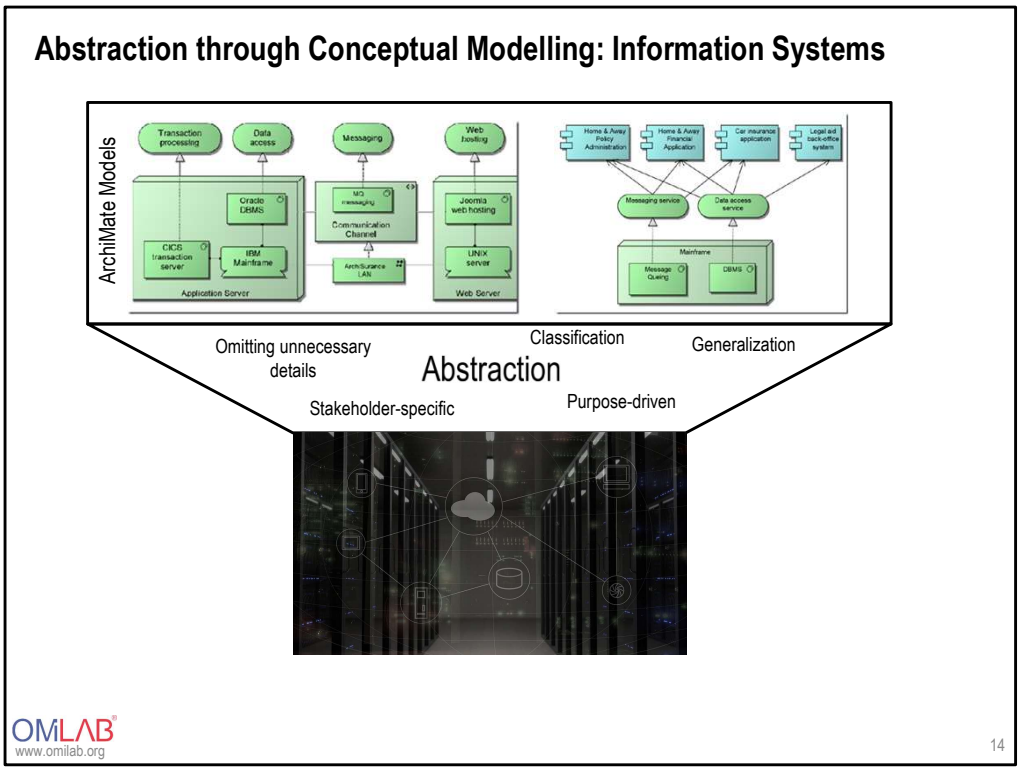
- This example shows abstraction as being applied to enterprise ecosystems
- Heterogeneity and complexity of enterprises is abstracted while creating a homogenized and high-level decision support system by means of a management dashboard
- This dashboard still has the notion of processes and resources and its data is fed from conceptual models



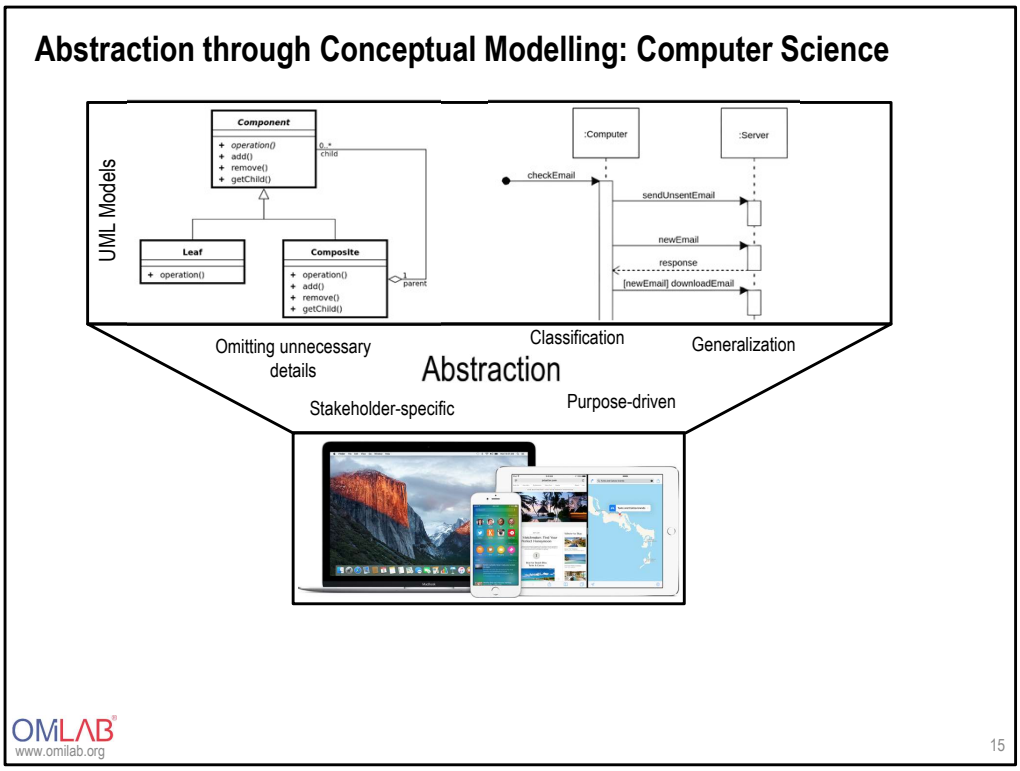
- This example shows abstraction as being applied to data
- Heterogeneity and complexity of data is abstracted while creating a homogenized schema of the data in form of an Entity-Relationship model



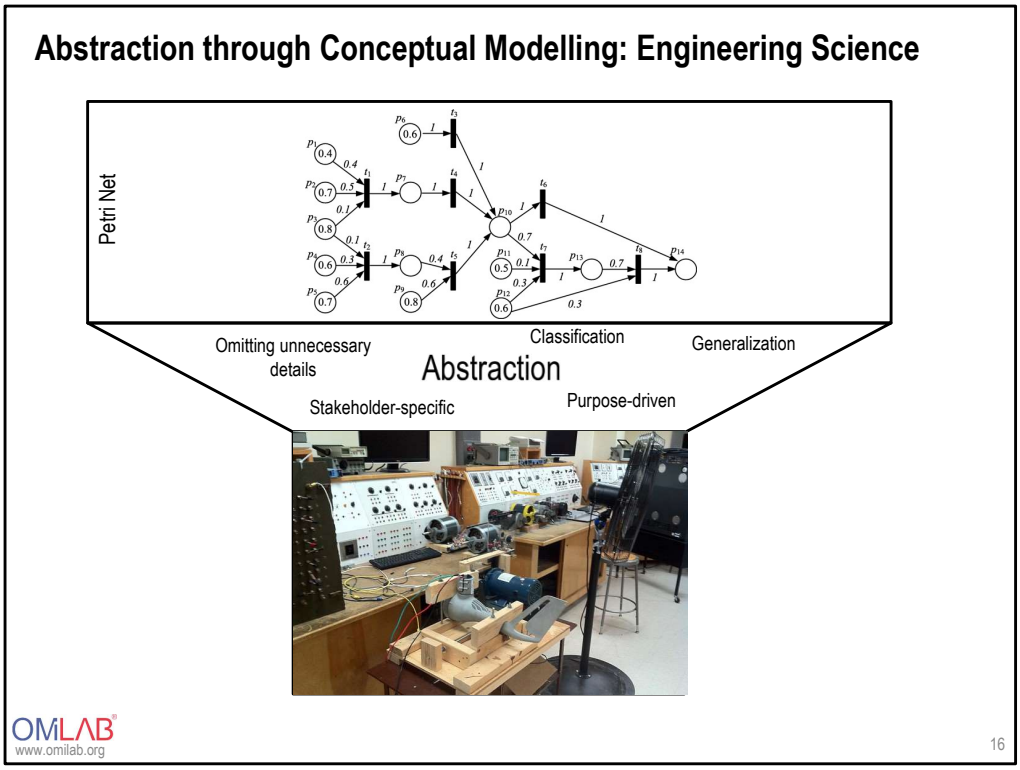
- This example shows abstraction as being applied to production processes
- Heterogeneity and complexity of the process is abstracted while creating a sequence of steps to be performed while creating a product



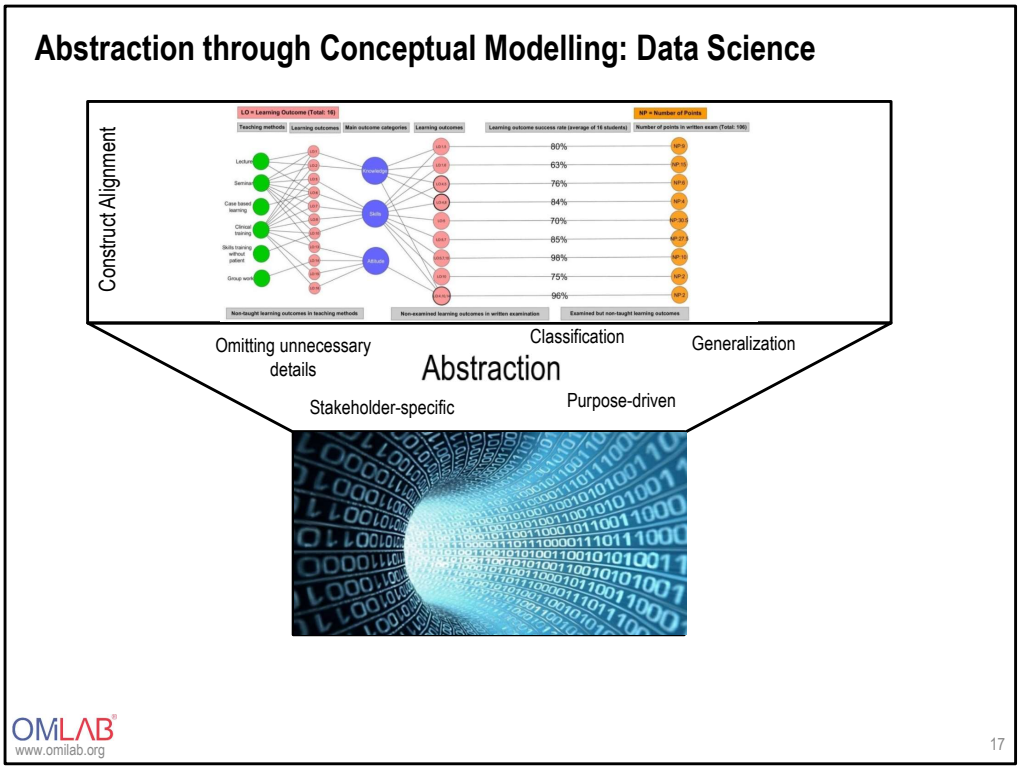
- This example shows abstraction as being applied to IT infrastructures
- Heterogeneity and complexity of current IT infrastructures is abstracted while creating a homogenized model of the IT landscape
    - The example shown here uses ArchiMate, the de-facto industry standard for Enterprise Architecture modelling that also comprises an Application and Technology layer viewpoint



- This example shows abstraction as being applied to computer systems and software
- Heterogeneity and complexity of current computer systems is abstracted
- The example shown here uses the UML family of languages as the de-facto industry standard for software and systems modelling

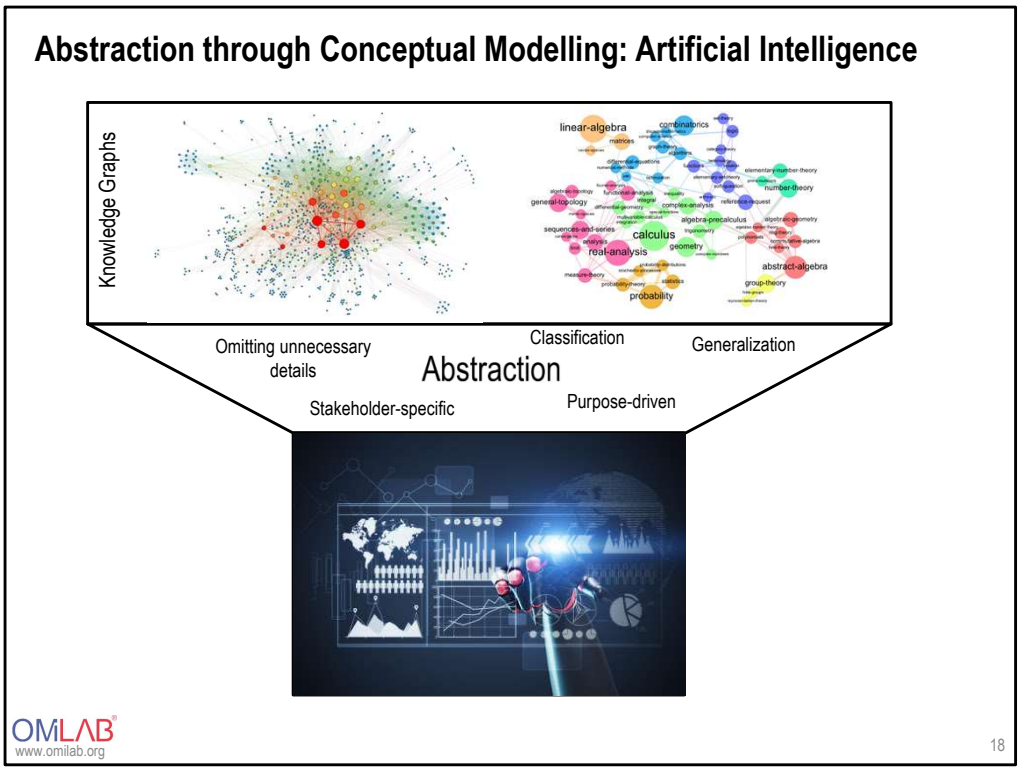


- This example shows abstraction as being applied to engineered systems
- The example shown here uses the Petri Net language to specify the dynamics/operation of the system



- This example shows abstraction as being applied to (Big) Data
- The example shown here uses the Construct Alignment approach that structures data relatioships along multiple facets

Upper Picture from paper: Visual analytics in healthcare education: exploring novel ways to analyze and represent big data in undergraduate medical education
Lower Picture Licensed by CC BY-ND



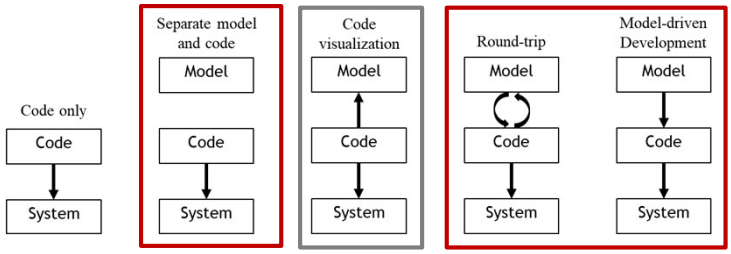
- This example shows abstraction as being applied in artificial intelligence
- The example shows knowledge graphs which were genertеated through AI systems
    - The AI works as a black box but produced and abstraction of its results such that human beings can comprehend it

What is the relationship between Conceptual Models, Code, and System?

WHICH MODEL VALUE CAN BE ENABLED BY MACHINES?

Different kinds of ‚Model Value‘

- Primarily Human-oriented, e.g.,
  - Abstracting from complexity
  - Documentation of system under study
  - Communication amongst different stakeholders
  - Validation of adherence to standards or compliance rules
- Primarily Machine-oriented, e.g.,
  - Analysis of even very complex models
  - Simulation of models and model alternatives
  - Model transformation
  - Model-driven Development



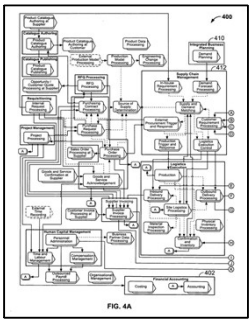
OMiLAB www.omilab.org

Figure adapted from "Domain-Specific Modeling enabling full code generation", Kelly and Tolvanen (2008)

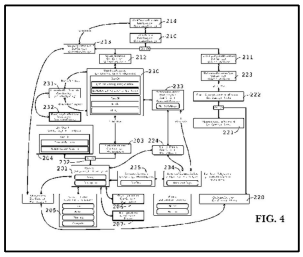
20

- This slide now picks up where we left concerning model value and extends the previously discussed intrinsic value by more emergent aspects that are enabled by mechanisms & algorithms
- The different kinds of model value are not also positioned along the different roles of a conceptual model

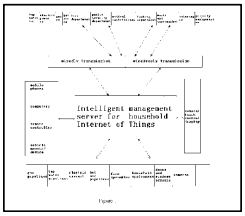
Separate Model and Code: Models for explaining complex systems (Patents)



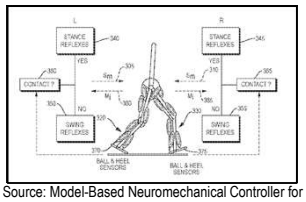
Source: Software model business process variant types, US 8312416 B2



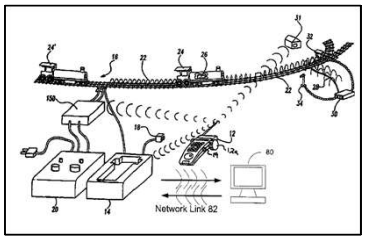
Source: Semantic model for insurance software components US 8650043 B1



Source: Intelligent management system for household internet of things US 20140244825 A1



Source: Model-Based Neuromechanical Controller for a Robotic Leg US 20100324699 A1



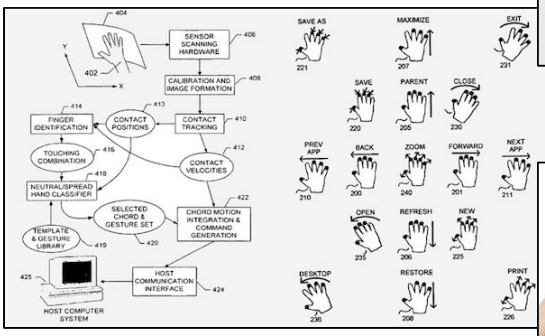
Source: Model train control system US 8154227B1

OMiLAB
www.omilab.org

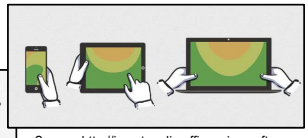
21

- This is the first slide that shows an example where code and model are separated.
- The images are taken from different – mostly software – patents.
    - Software per se (source code) cannot be patented, whereas a conceptual model of the software can!

Separate Model and Code: Models for UI simulation



Source: http://images.fastcompany.com/upload/apple-multitouch1.jpg



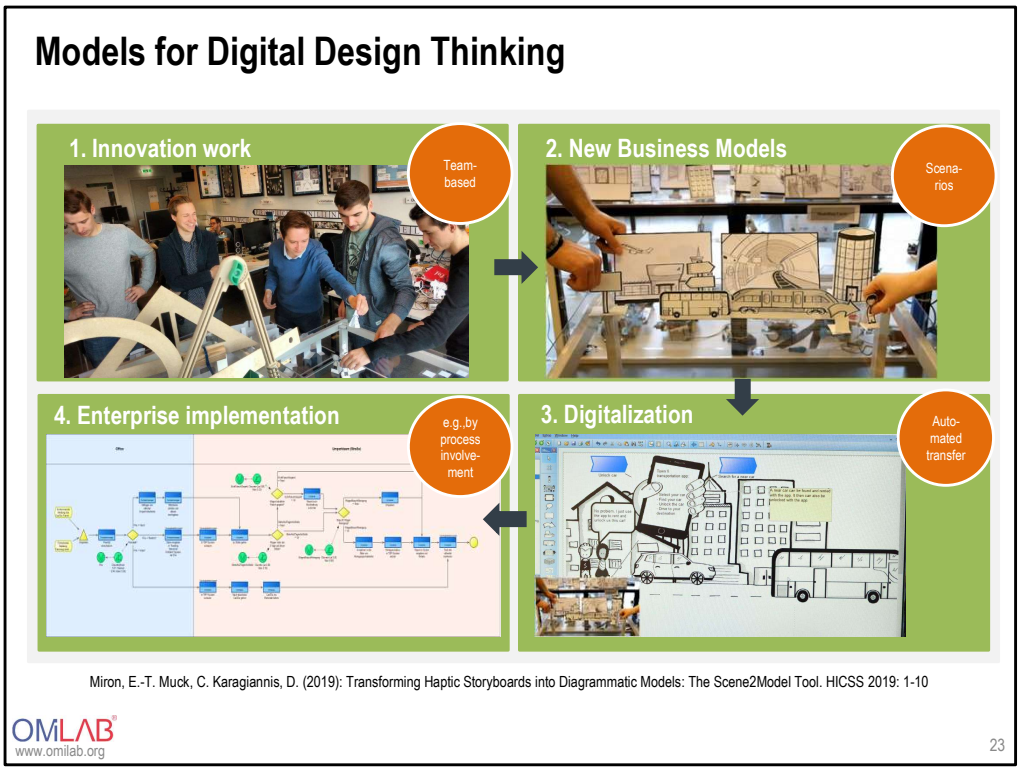
Source: http://insertmedia.office.microsoft.com

Source: http://news.preisgenau.de/wp-content/uploads/2011/03/ipad-multitouch.jpg

OMiLAB
www.omilab.org

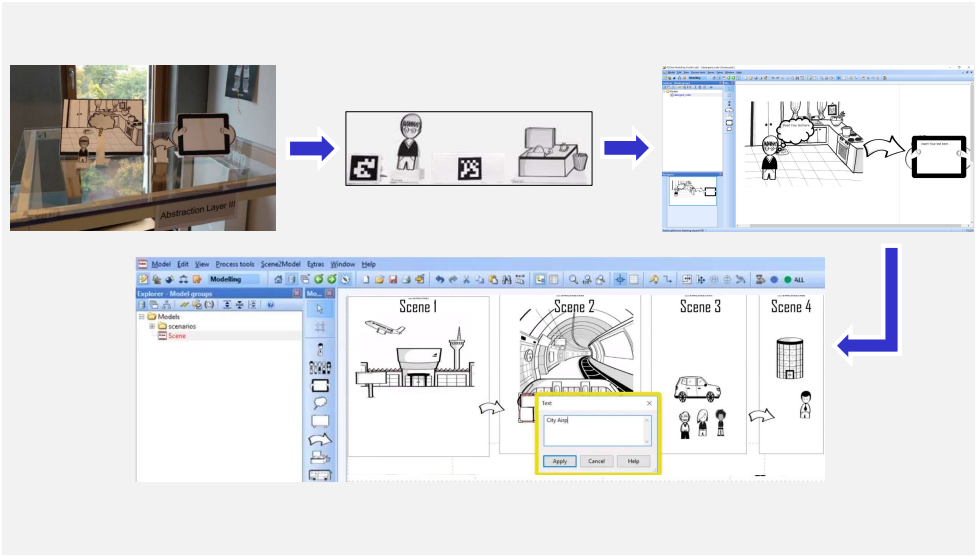
22

- This is another example of the model from code separate dimension
- Here, conceptual models are used to specify and simulate the user interface (UI) of mobile apps on different screen sizes.
- Software engineers can use these kind of models to pre-test and simulate e.g., the reachability of interaction elements for the user.



- Emphasize the role and the added value of conceptual modeling in design thinking
  - Digitizing the design thinking artefact
  - Enabling further enrichement of the design thinking artefact
  - Enabling computerized processing of the design thinking artefact
    - Compared to the status-quo where pictures are made which cannot be further processed
- Point the interested reader to a publication that has more details about Scene2Model

Models for Digital Design Thinking

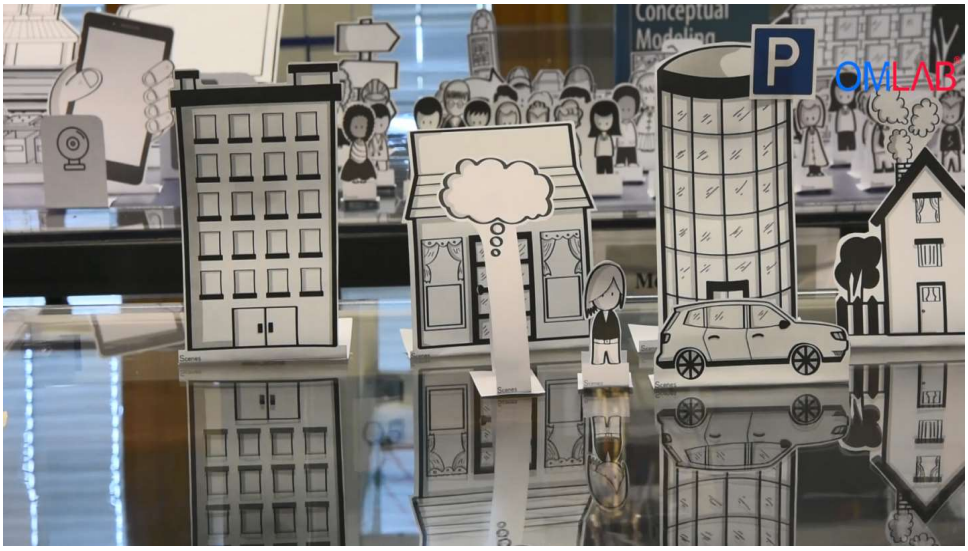


OMiLAB
www.omilab.org

24

- This slide shows in greater detail how the digitization of the design thinking artefact is realized in the Scene2Model approach
- SAP Scene elements are printed and cut out to enable the haptic experience during the design thinking workshop
- Ne win the Scene2Model approach is the attachement of an QR code to the SAP scene elements
- This QR code can be tracked automatically ba a camera and, once the workshop has finished with one scene, can be digitized using the Scene2Model tool
- Within the tool, the digital scene and its elements can be further enriched with semantics and linked to other conceptual modelling langauges like Processes

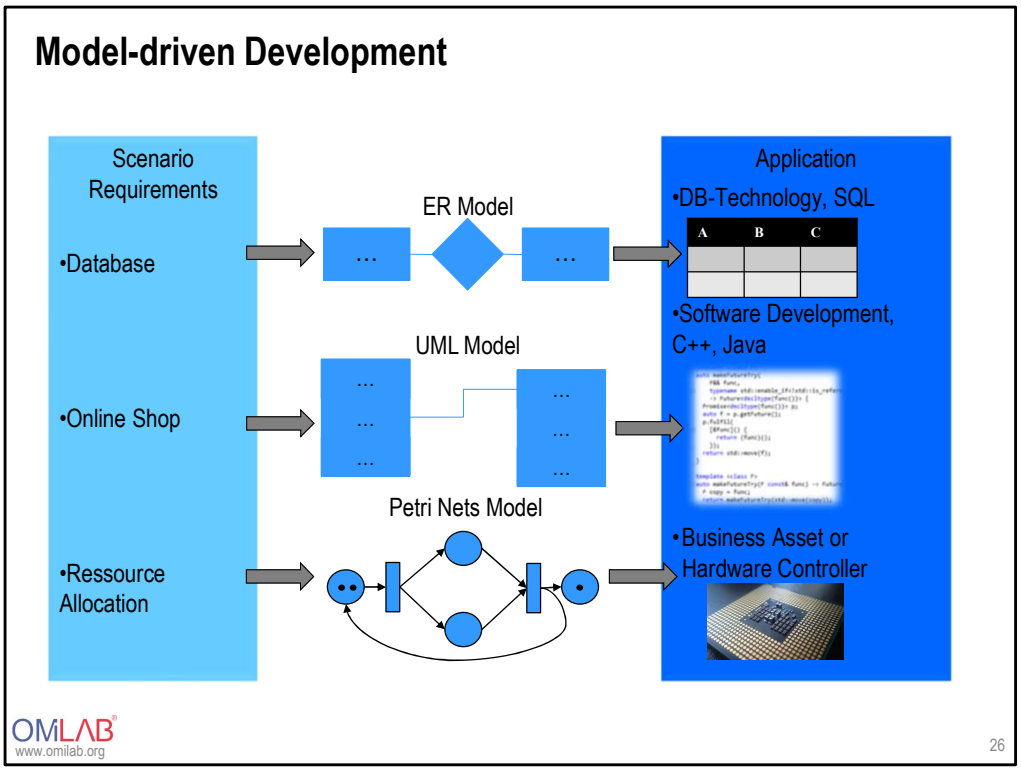
Models for Digital Design Thinking



OMiLAB
www.omilab.org

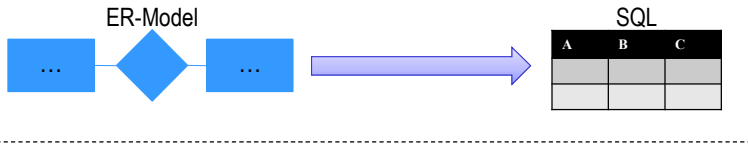
25

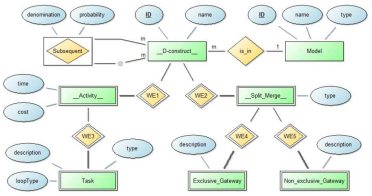
- Show the video “Module4_Target-S2M_OMiLAB” and discuss the different phases of a Scene2Model based Design Thinking workshop
- 1. Haptic and creative workshop to ideate
- 2. Digitization using Scene2Model and further enrichment and processing of the model



- Describe the added value of conceptual modelling in these three model-drivne development scenarios
  - Instead of writing low-level technical code of a dedicated database management system
    - Design ER models, enrich them with specific attributes and generate the code automatically from the models
  - Instead of writing lots of lines of program code of a dedicated software system
    - Design UML models, enrich them with specific attributes and generate parts of the code automatically from the models
  - Instead of writing low-level technical code for ressource allocation or synchronous communication
    - Design and simulate Petri Net models
- All three examples will be discussed in detail in the following three slides.

Model-driven Development - 1: ER-Model to SQL Code





+ SQL-specific Informations

SQL Code Generation

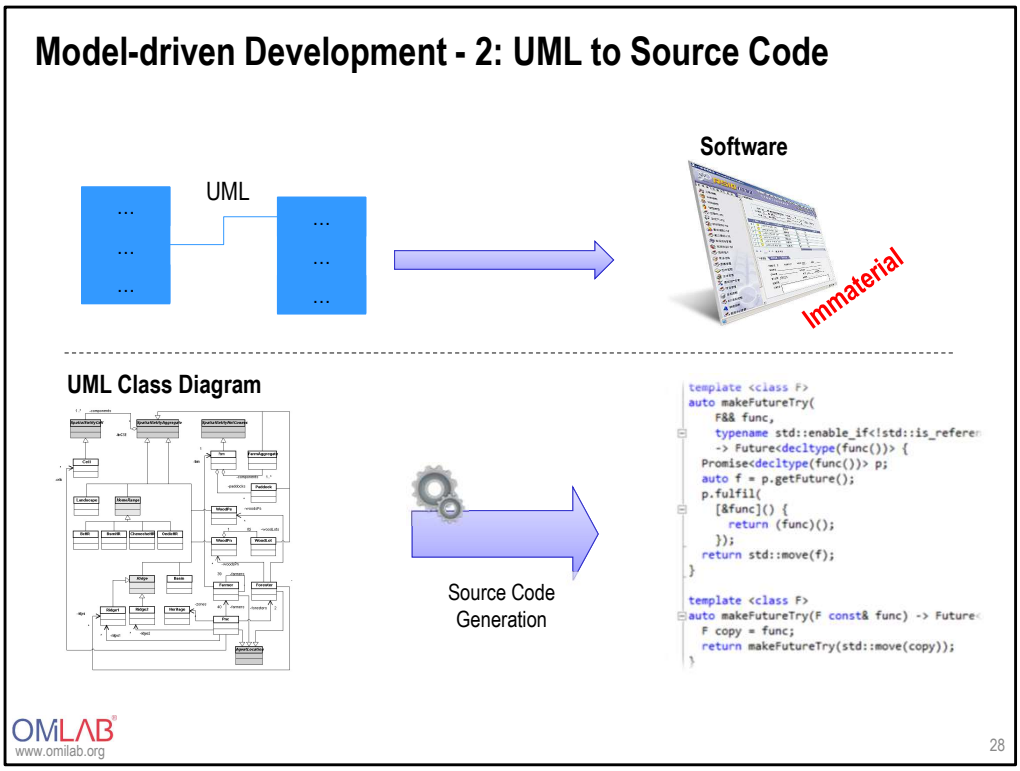
```
CREATE TABLE Subsequent (
  source_ID VARCHAR(32) NOT NULL,
  target_ID VARCHAR(32) NOT NULL,
  denomination VARCHAR(128),
  probability FLOAT,
  CONSTRAINT pk_Subsequent PRIMARY KEY (source_ID,target_ID)
);

CREATE TABLE __D_construct__ (
  ID VARCHAR(32) NOT NULL,
  name VARCHAR(128),
  model_ID VARCHAR(32),
  CONSTRAINT pk___D_construct__ PRIMARY KEY (ID)
);

CREATE TABLE Model (
  ID VARCHAR(32) NOT NULL,
  name VARCHAR(128),
  type VARCHAR(128),
  CONSTRAINT pk_Model PRIMARY KEY (ID)
);

ALTER TABLE Subsequent ADD CONSTRAINT fk_Subsequent_source___D_construct__ FOREIGN KEY
(source_ID) REFERENCES __D_construct__(ID);
ALTER TABLE Subsequent ADD CONSTRAINT fk_Subsequent_target___D_construct__ FOREIGN KEY
(target_ID) REFERENCES __D_construct__(ID);
ALTER TABLE __D_construct__ ADD CONSTRAINT fk___D_construct___model_Model FOREIGN KEY
(model_ID) REFERENCES Model(ID);
```

OMiLAB www.omilab.org

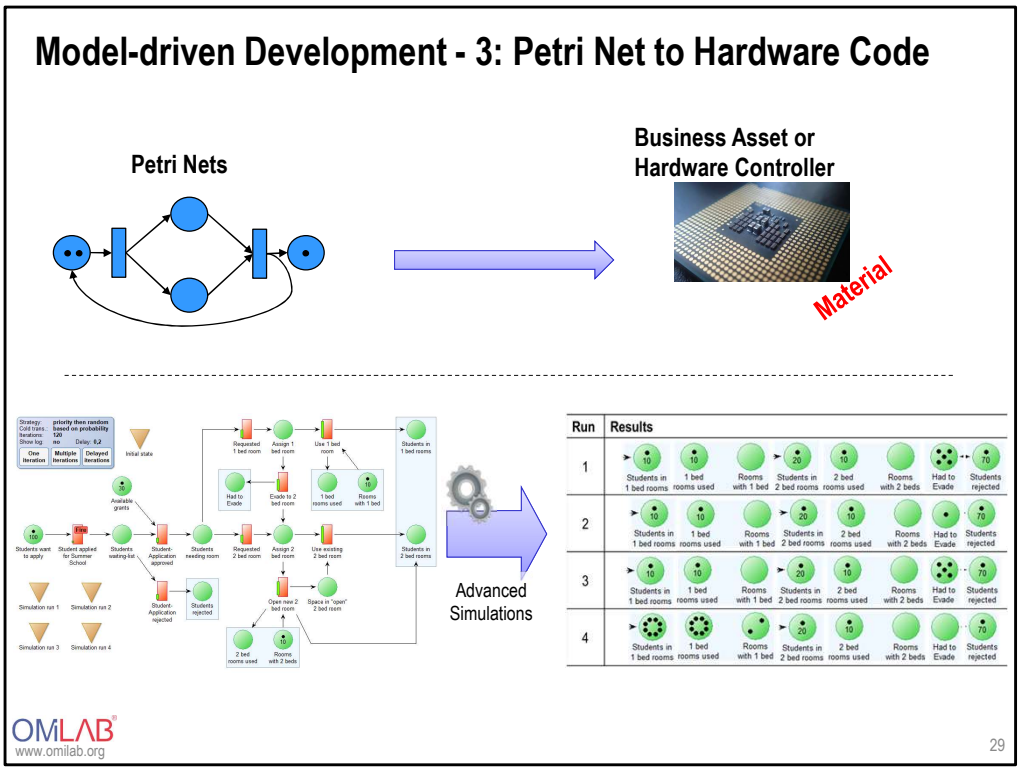
http://vienna.omilab.org/repo/files/Bee-Up/The_IMKER_Case_Study.pdf

27

- Describe the SQL generation by exemplifying that, e.g.,
  - For ever Entity _D_Construct_ in the ER model a CREATE TABLE statement is generated
  - For the Relationship Subsequent in the ER model a CREATE TABLE statement is generated
  - For every defined Primary Key attribute a corresponding PRIMARY_KEY statement is generated
- For supporting further database management systems and dialoects, only a new generator needs to be implemented



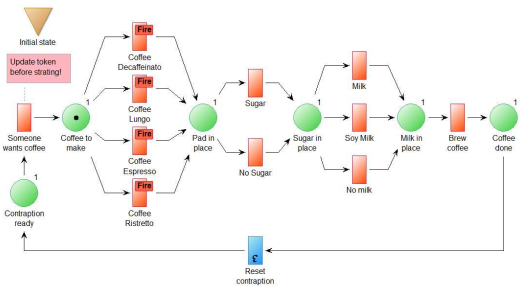
- Describe that parts of the source code can be generated from the UML model
  - For ever class in the UML class diagram model a class in e.g., Java is generated
- For supporting further programming languages, only a new generator needs to be implemented



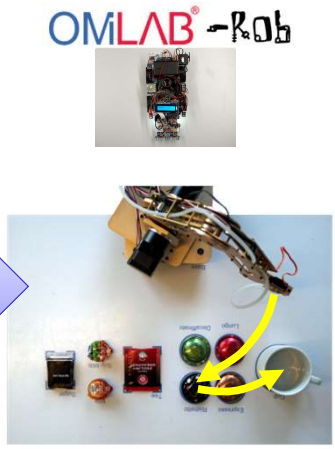
- When the system to be build has scarce ressources that need to be allocated or used exclusively, Petri nets are a very powerful modelling language
- With Petri nets states and transitions can be modeled as well as the ressource consumption and production
    - Using the mathematical sound foundations of Petri Nets, manifold interesting simulations can be executed to validate and analyse the model
    - These validation and analysis capabilities are very powerful as detecting errors early is a means of saving costs in the end

Round-trip Engineering: Robotic Process Automation

Robotic Process Automation based on Petri Nets




```
HTTP_SEND_REQUEST (str_roboturl + "positionXYZ") str_method:("PUT") ...
HTTP_SEND_REQUEST (str_roboturl + "positionXYZ") str_method:("PUT") ...
HTTP_SEND_REQUEST (str_roboturl + "grabOn") str_method:("POST") ...
...
```



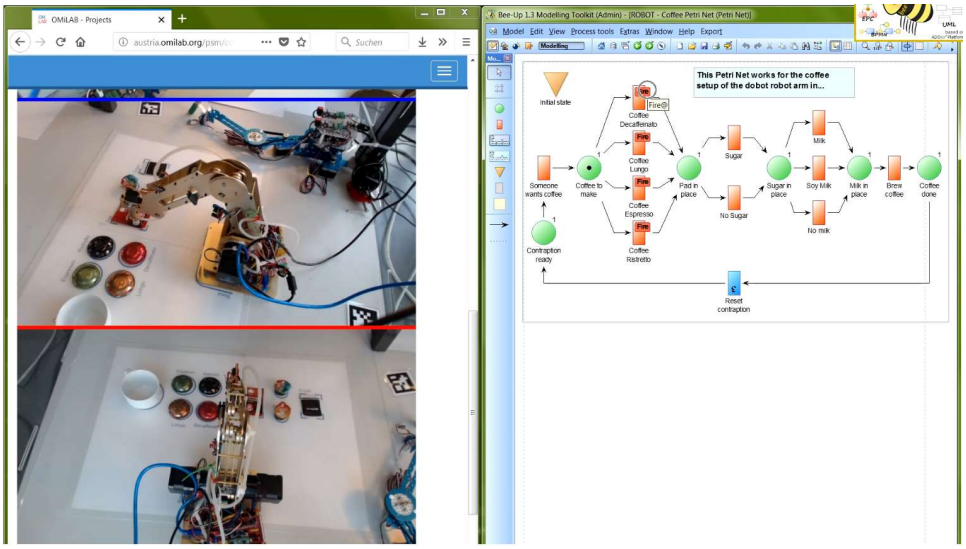
Model: Petri Net in Bee-Up
Robot-Arm in OMiLAB http://austria.omilab.org/psm/content/omiarm1/info

OMiLAB
www.omilab.org

30

- This slide shows a scenario, where a conceptual model, in this case a Petri Net model is enriched with attributes that enable the communication between the modelling tool and a Dobot arm
- When executing the Petri net model in the Bee-Up tool, the tool performs HTTP requests as exemplified in the slide in order to steer the robotic arm
- Consequently, one can create a Cyber-physical System using conceptual modelling and robotics without considering all technical details.
    - Conceptual modelling here also raises the abstraction level of designing and executing processes using Robotics

Round-trip Engineering: Robotic Process Automation



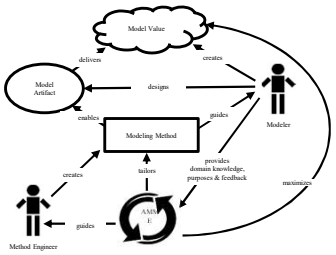
https://austria.omilab.org/psm/content/bee-up/info

OMiLAB
www.omilab.org

31

- Describe what can be seen in the video “Module4_Bee-Up-Omiarm_OMiLAB”.
- Emphasize that this is all available in the openly available Bee-Up tool and that the corresponding hardware can be ordered as an OMiLAB package.

Wrap-Up



- Model value is very
  - **Heterogeneous**
  - **Domain**- and **Stakeholder-specific**
- Model value can be enabled
  - By the **Method Engineer** who encodes it in the metamodel
  - By the **Modeller** who generates value while creating models
  - By the **Modelling tool** that executes mechanisms & algorithms to automatically process the models
- Model value can address
  - **Different Stakeholders** involved
  - **Different Purposes** of conceptual modelling

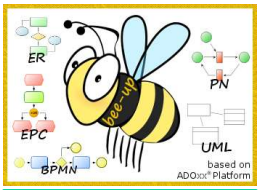
OMiLAB
www.omilab.org

32

- Wrap up the whole module by emphasizing again the co-creation aspect of model value
- Additionally, proper tooling is very important when aiming to increase the model value by means of advanced mechanisms & algorithms
- Model value – as the generic concept of value – is stakeholder specific. As such, model value can only be measure from a user perspective.
  - This user can be the modeler, but in larger enterprises modeler and model user might also be different persons.

The Bee-Up Conceptual Modelling Tool

- Provides an **integrated modelling environment** for **heterogeneous languages**
- Provides **advanced mechanisms & algorithms to process models**

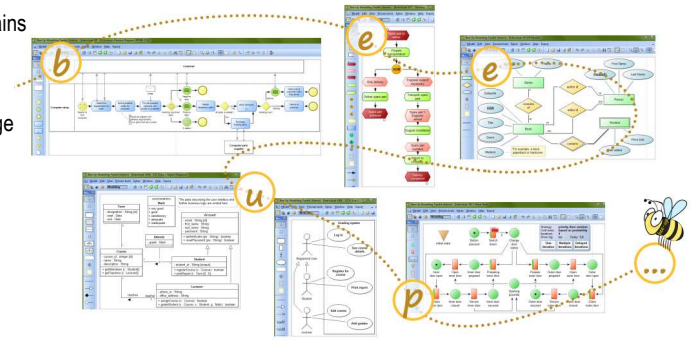


www.omilab.org/**bee-up**

A collection of "classical" modelling languages employed in different domains, e.g. software and systems modelling, business process modelling, and data modelling.

The tool **supports university teachers in basic conceptual modelling courses**. It currently supports the following modelling languages:

BPMN – Business Process Model and Notation
EPC – Event-driven Process Chains
ER – Entity Relationship
|
UML – Unified Modeling Language
Petri Nets



OMiLAB
www.omilab.org

33

- This slides aims to position Bee-Up as a tool that provides mutliple means of abstraction including some of the previously mentioned ones, i.e.,
  - Processes -> BPMN, EPC, Petri Net
  - Data -> ER
  - Application Systems -> UML
- Emphasize that the tool is freely available and is used at multiple universities in teaching conceptual modelling
- Emphasize that on the tool homepage there is lots of additional information liek tutorial videos and case studies amongst others.

Self-control questions

- What is meant by ‚Model Value'?
- What is the intrinsic value of models?
- Which further model value can be generated through mechanisms & algorithms?
- Which roles can a conceptual model play in the design of a database?
- How can Petri nets simulations help in the construction of consumer producer systems?
- Which roles can a conceptual model play in Robotic Process Automation?

References

- Bork D. et al. (2019) An open platform for modeling method conceptualization: The OMiLAB digital ecosystem. Commun Assoc Inf Syst Vol. 44, pp. 673-697
- Brenner, W., Karagiannis, D., Kolbe, L. et al. (2014). User, use & utility research. Business & Information Systems Engineering, 6(1), 55-61.
- Burzynski, P., Karagiannis, D. (2020) bee-up - A Teaching Tool for Fundamental Conceptual Modelling. Modellierung (Companion) 2020: pp. 217-221
- Karagiannis, D. (2019): Teaching Conceptual Modeling with the OMiLAB: The Value of Models, In: Strecker et al. Five Inspiring Course (Re-) Designs Examples of Innovations in Teaching and Learning BISE, Bus Inf Syst Eng.
- Karagiannis, D., Buchmann, R.A., Burzynski, P. Reimer, U., Walch, M. (2016) Fundamental Conceptual Modeling Languages in OMiLAB. Domain-Specific Conceptual Modeling 2016: pp. 3-30
- Kelly, S., & Tolvanen, J. P. (2008). Domain-specific modeling: enabling full code generation. John Wiley & Sons.
- Miron, E.-T. Muck, C. Karagiannis, D. (2019): Transforming Haptic Storyboards into Diagrammatic Models: The Scene2Model Tool. HICSS 2019: 1-10
- Walch, M., Karagiannis, D. (2017) Service-Driven Enrichment for KbR in the OMiLAB Environment. ICServ 2017: pp. 164-177