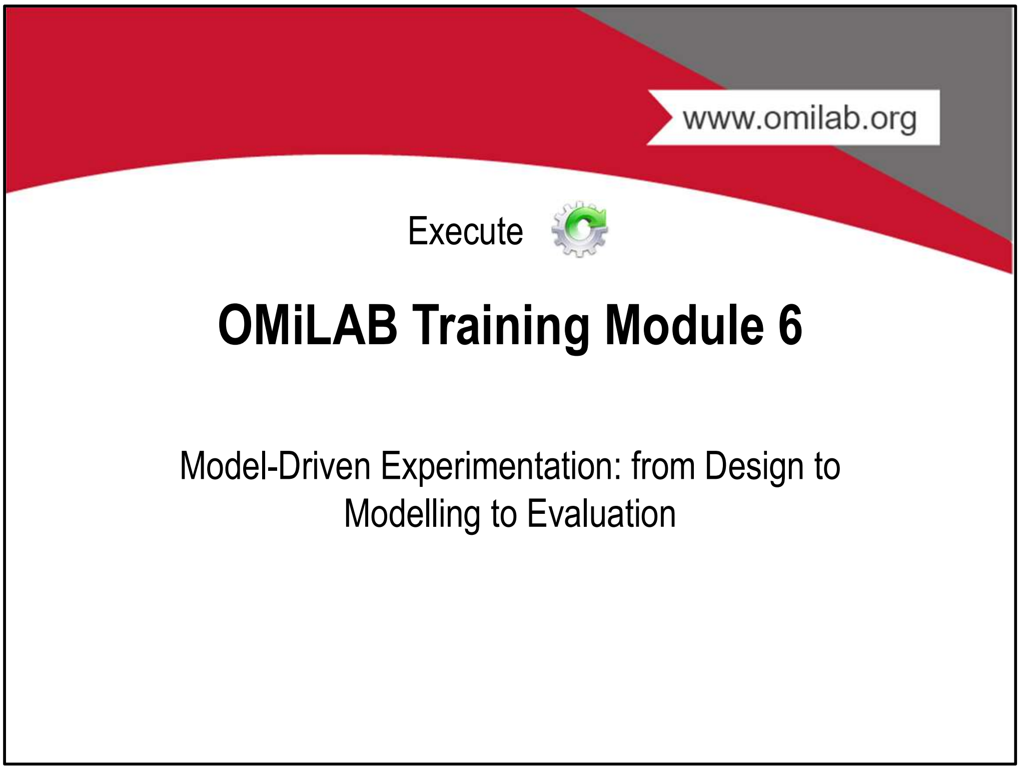
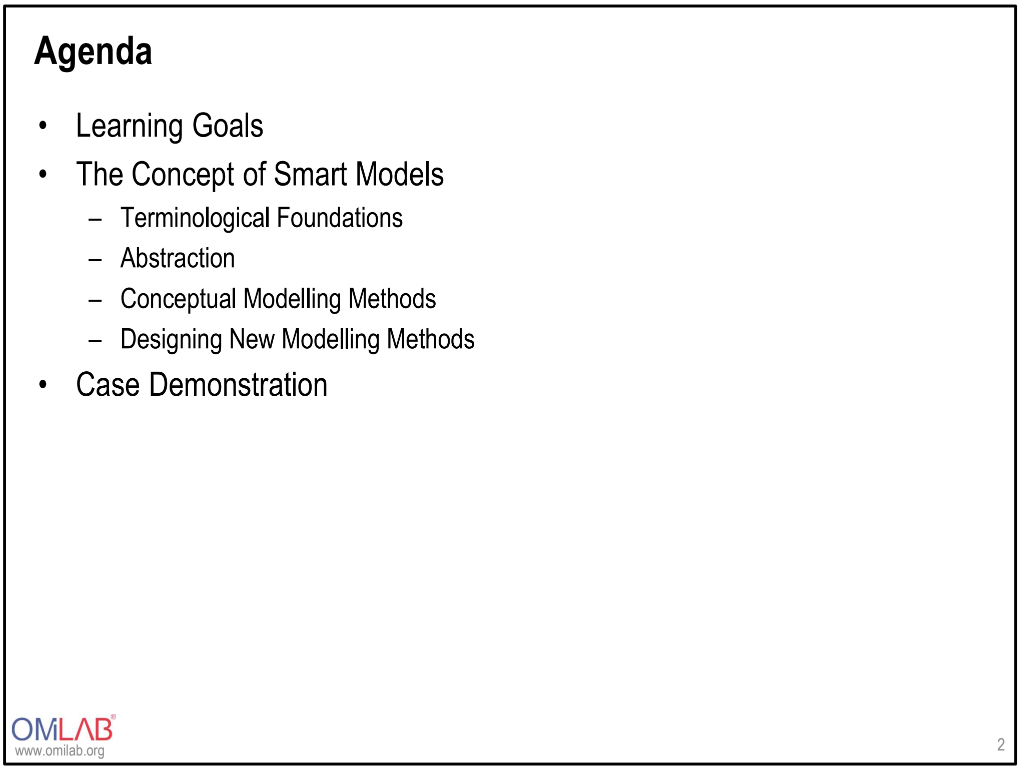


www.omilab.org

Execute

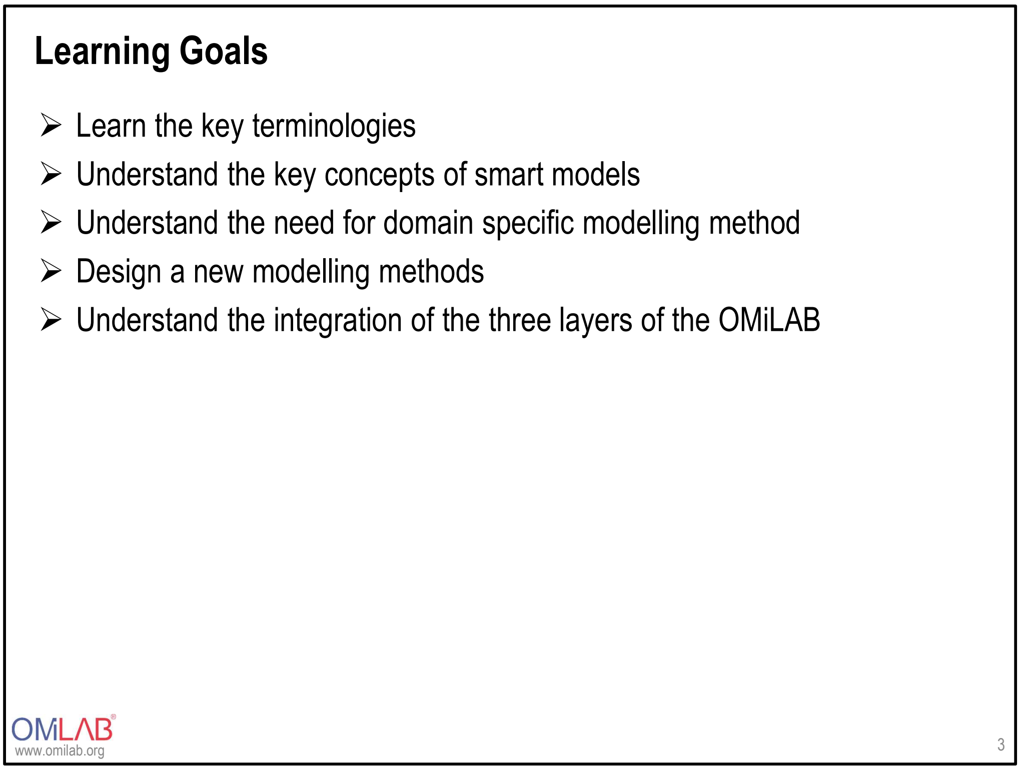
OMiLAB Training Module 6

Model-Driven Experimentation: from Design to Modelling to Evaluation



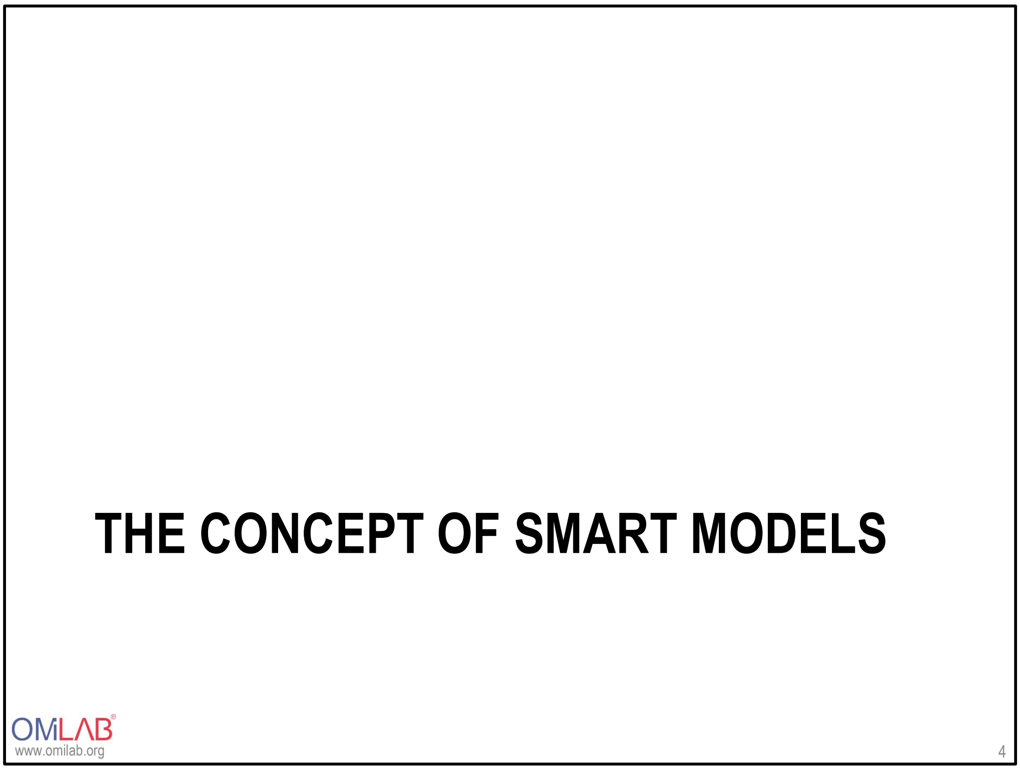
Agenda

- Learning Goals
- The Concept of Smart Models
  - Terminological Foundations
  - Abstraction
  - Conceptual Modelling Methods
  - Designing New Modelling Methods
- Case Demonstration



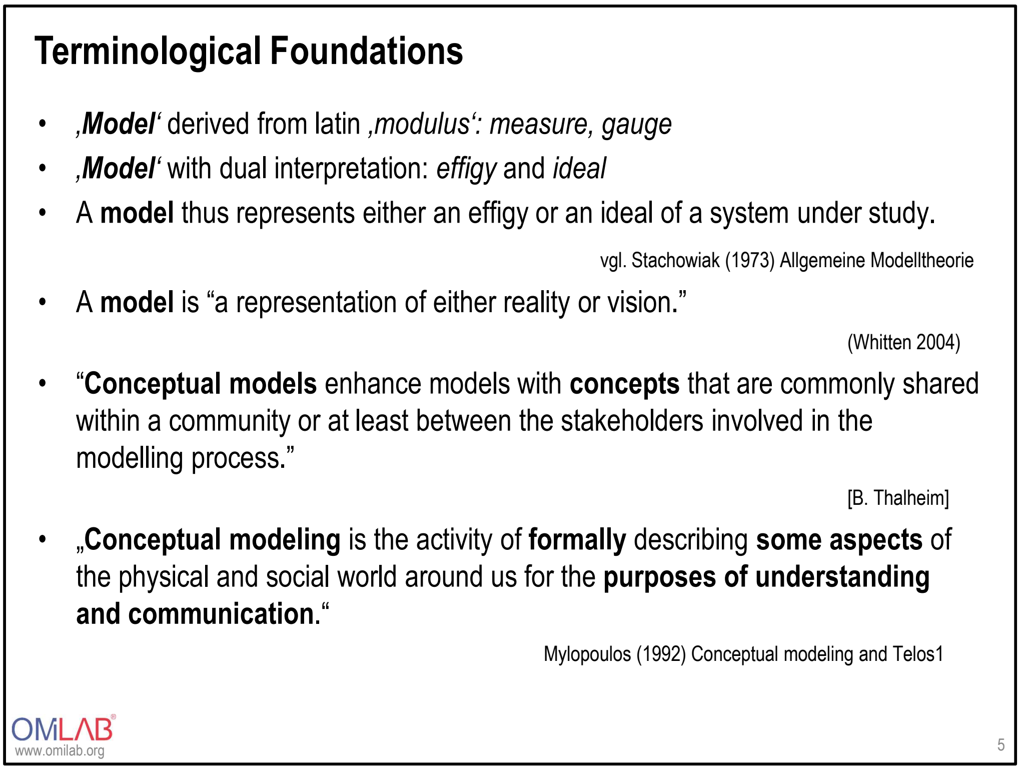
Learning Goals

- Learn the key terminologies
- Understand the key concepts of smart models
- Understand the need for domain specific modelling method
- Design a new modelling methods
- Understand the integration of the three layers of the OMiLAB



THE CONCEPT OF SMART MODELS

www.omilab.org



Terminological Foundations

- ‚***Model***‘ derived from latin *‚modulus‘: measure, gauge*
- ‚***Model***‘ with dual interpretation: *effigy* and *ideal*
- A **model** thus represents either an effigy or an ideal of a system under study.

vgl. Stachowiak (1973) Allgemeine Modelltheorie

- A **model** is “a representation of either reality or vision.”

(Whitten 2004)

- “**Conceptual models** enhance models with **concepts** that are commonly shared within a community or at least between the stakeholders involved in the modelling process.”

[B. Thalheim]

- „**Conceptual modeling** is the activity of **formally** describing **some aspects** of the physical and social world around us for the **purposes of understanding and communication**.“

Mylopoulos (1992) Conceptual modeling and Telos1

OMILAB

www.omilab.org

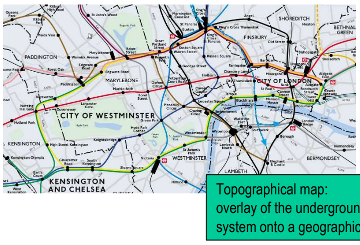
5

“A model is typically a schematic description of a system, theory, or phenomenon of an origin that accounts for known or inferred properties of the origin and may be used for further study of characteristics of the origin.” [B. Thalheim]

An artefact that is acknowledged by an observer as being a representation of a domain for a specific purpose
Sources: Stachowiak, Thalheim, Ogden and the FRISCO report

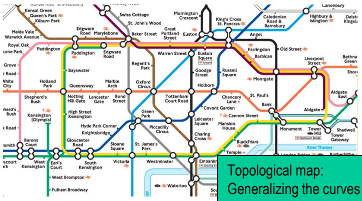
Abstraction is key to Modelling

- **Simplification**
  - — Withdrawing or removing something
  - — Leaving out of consideration one or more properties of a complex object



Topographical map:
overlay of the underground system onto a geographical map

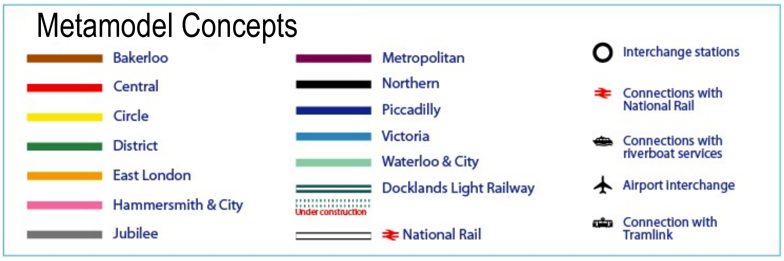
- **Generalization**
  - — Formulating general concepts by abstracting common properties of instances
  - — A general concept formed by extracting common features



Topological map:
Generalizing the curves to just horizontal, vertical and diagonal lines and where the distances between stations were no longer proportional

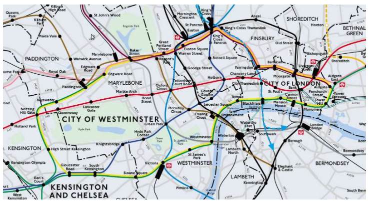
Cf. Jeff Kramer, Is Abstraction the key to computing?
Image sources: https://thatsmaths.files.wordpress.com/2013/01/tubemap-topog-city.png
https://www2.b3ta.com/host/creative/53465/1200582535/mapbig.gif

Abstraction is key to Metamodelling

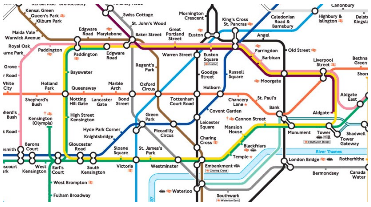


Metamodel Concepts

- 🟫 Bakerloo
- 🟥 Central
- 🟨 Circle
- 🟩 District
- 🟧 East London
- Hammersmith & City
- Jubilee
- Metropolitan
- ⬛ Northern
- 🟦 Piccadilly
- Victoria
- Waterloo & City
- Docklands Light Railway
- Under construction
- National Rail
- ○ Interchange stations
- Connections with National Rail
- Connections with riverboat services
- ✈ Airport interchange
- Connection with Tramlink



Modeling

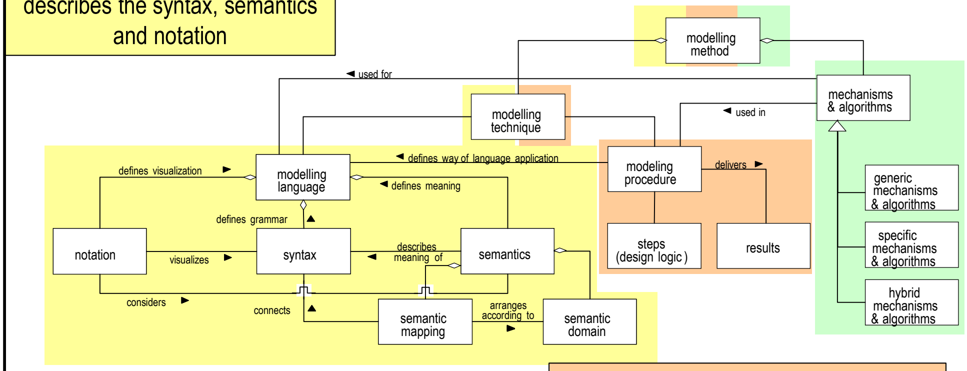


Abstraction

Conceptual Modelling Methods

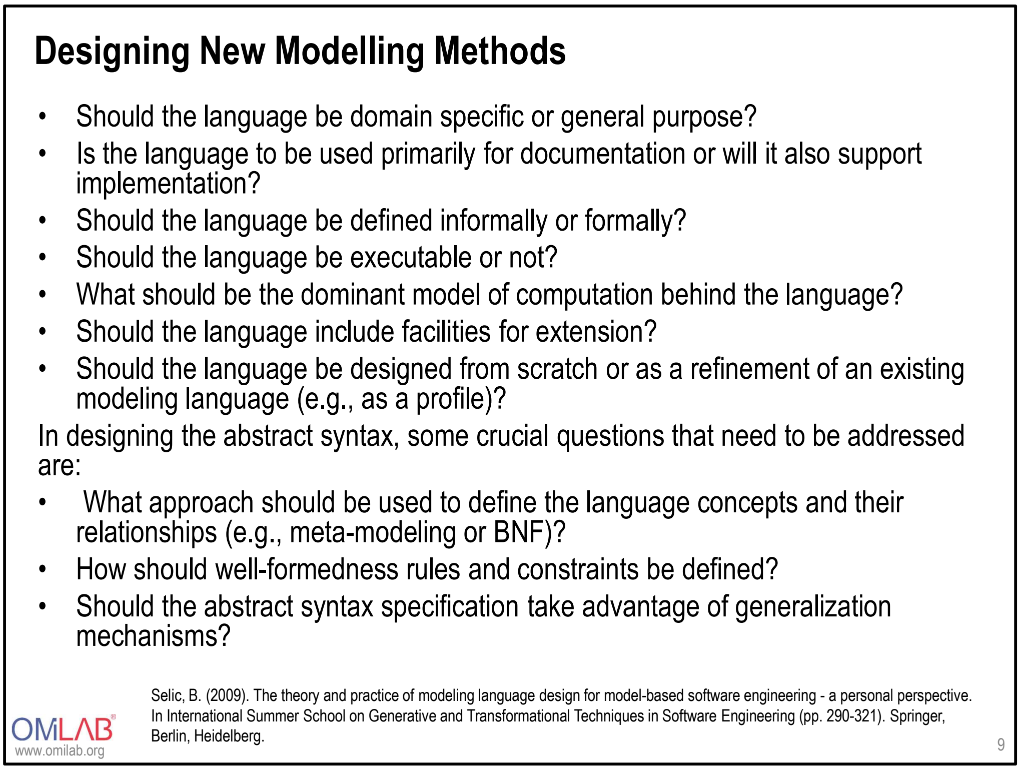
algorithms and mechanisms that provide *“functionality to use and evaluate”* models described by a modelling language

the ***modelling language*** that describes the syntax, semantics and notation



the ***modelling procedure*** that describe how to create valid models

Reference: Karagiannis, D., Kühn, H.: „Metamodelling Platforms“. In Bauknecht, K., Min Tjoa, A., Quirchmayer, G. (Eds.): Proceedings of the Third International Conference EC-Web 2002 – Dexa 2002, Aix-en-Provence, France, September 2002, LNCS 2455, Springer, Berlin/Heidelberg, p. 182 ff.



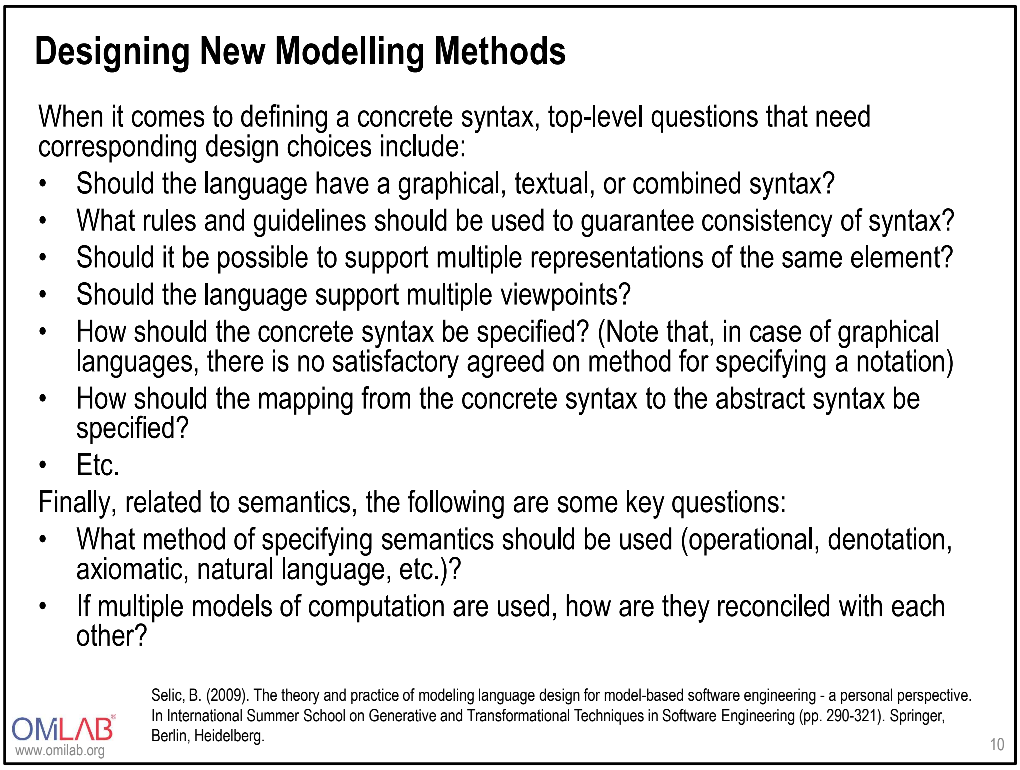
Designing New Modelling Methods

- Should the language be domain specific or general purpose?
- Is the language to be used primarily for documentation or will it also support implementation?
- Should the language be defined informally or formally?
- Should the language be executable or not?
- What should be the dominant model of computation behind the language?
- Should the language include facilities for extension?
- Should the language be designed from scratch or as a refinement of an existing modeling language (e.g., as a profile)?

In designing the abstract syntax, some crucial questions that need to be addressed are:

- What approach should be used to define the language concepts and their relationships (e.g., meta-modeling or BNF)?
- How should well-formedness rules and constraints be defined?
- Should the abstract syntax specification take advantage of generalization mechanisms?

Selic, B. (2009). The theory and practice of modeling language design for model-based software engineering - a personal perspective. In International Summer School on Generative and Transformational Techniques in Software Engineering (pp. 290-321). Springer, Berlin, Heidelberg.



Designing New Modelling Methods

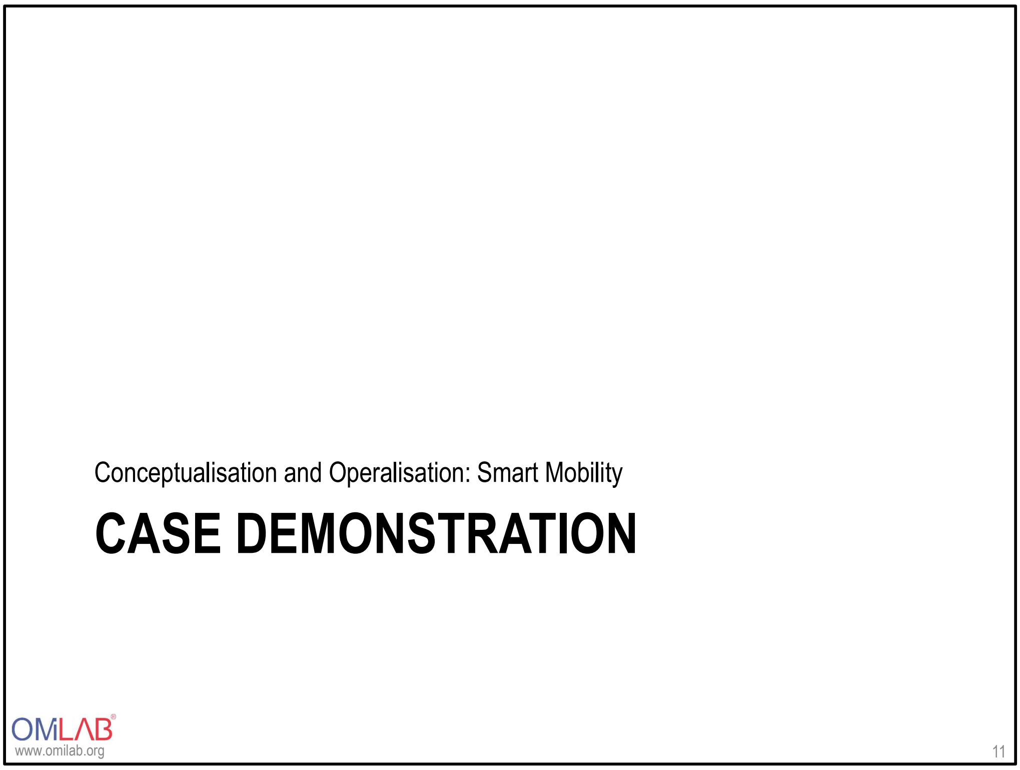
When it comes to defining a concrete syntax, top-level questions that need corresponding design choices include:

- Should the language have a graphical, textual, or combined syntax?
- What rules and guidelines should be used to guarantee consistency of syntax?
- Should it be possible to support multiple representations of the same element?
- Should the language support multiple viewpoints?
- How should the concrete syntax be specified? (Note that, in case of graphical languages, there is no satisfactory agreed on method for specifying a notation)
- How should the mapping from the concrete syntax to the abstract syntax be specified?
- Etc.

Finally, related to semantics, the following are some key questions:

- What method of specifying semantics should be used (operational, denotation, axiomatic, natural language, etc.)?
- If multiple models of computation are used, how are they reconciled with each other?

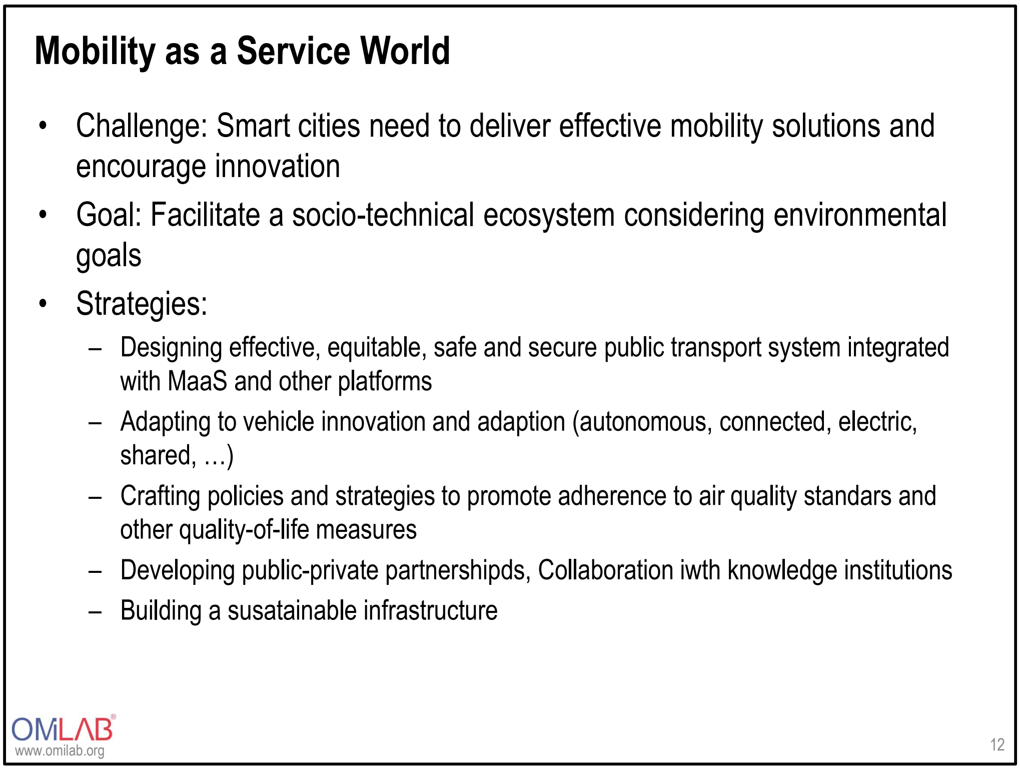
Selic, B. (2009). The theory and practice of modeling language design for model-based software engineering - a personal perspective. In International Summer School on Generative and Transformational Techniques in Software Engineering (pp. 290-321). Springer, Berlin, Heidelberg.



Conceptualisation and Operalisation: Smart Mobility

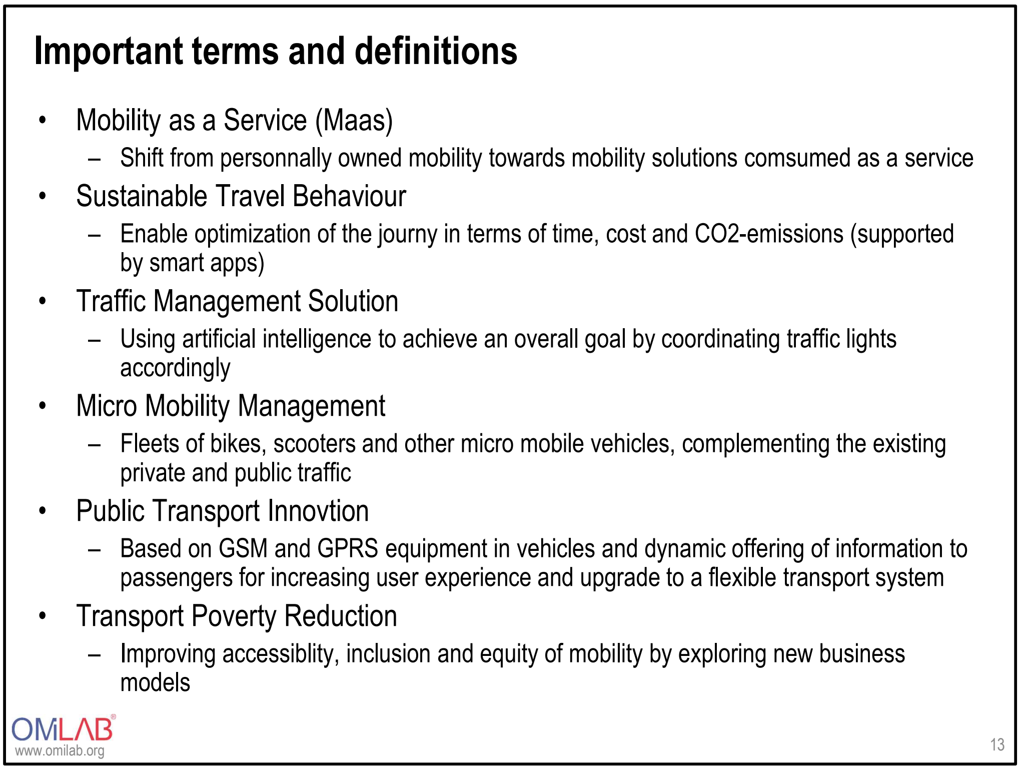
CASE DEMONSTRATION

OMiLAB®
www.omilab.org



Mobility as a Service World

- Challenge: Smart cities need to deliver effective mobility solutions and encourage innovation
- Goal: Facilitate a socio-technical ecosystem considering environmental goals
- Strategies:
  - Designing effective, equitable, safe and secure public transport system integrated with MaaS and other platforms
  - Adapting to vehicle innovation and adaption (autonomous, connected, electric, shared, …)
  - Crafting policies and strategies to promote adherence to air quality standars and other quality-of-life measures
  - Developing public-private partnershipds, Collaboration iwth knowledge institutions
  - Building a susatainable infrastructure

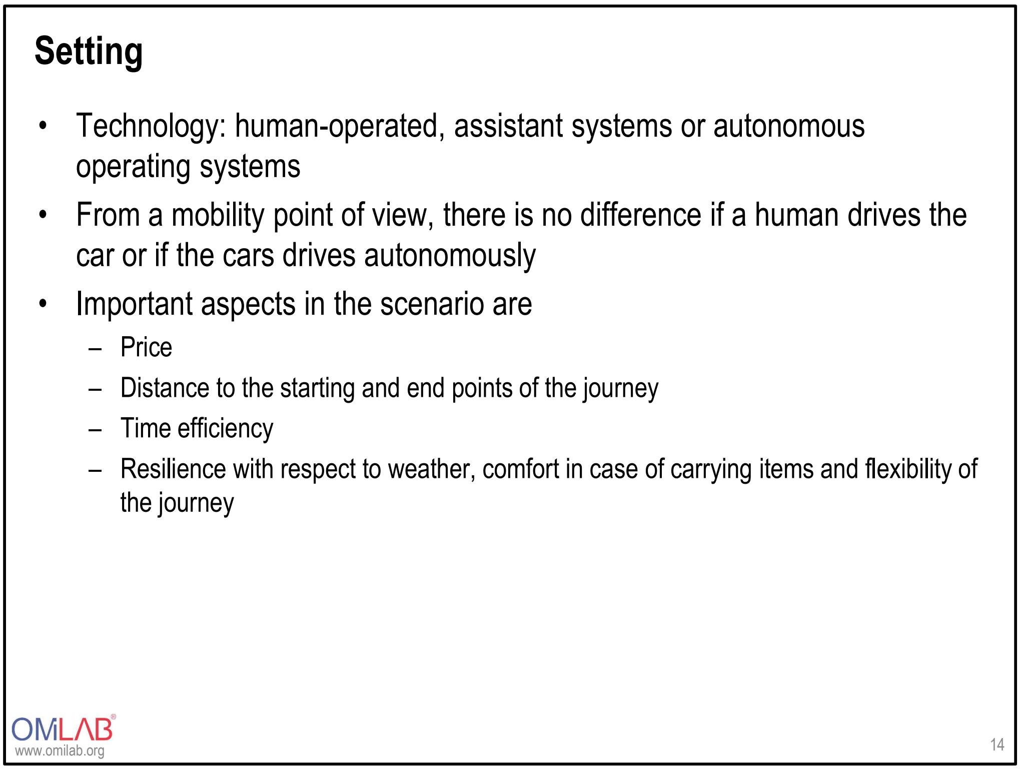


Important terms and definitions

- Mobility as a Service (Maas)
  - Shift from personnally owned mobility towards mobility solutions comsumed as a service
- Sustainable Travel Behaviour
  - Enable optimization of the journy in terms of time, cost and CO2-emissions (supported by smart apps)
- Traffic Management Solution
  - Using artificial intelligence to achieve an overall goal by coordinating traffic lights accordingly
- Micro Mobility Management
  - Fleets of bikes, scooters and other micro mobile vehicles, complementing the existing private and public traffic
- Public Transport Innovtion
  - Based on GSM and GPRS equipment in vehicles and dynamic offering of information to passengers for increasing user experience and upgrade to a flexible transport system
- Transport Poverty Reduction
  - Improving accessiblity, inclusion and equity of mobility by exploring new business models

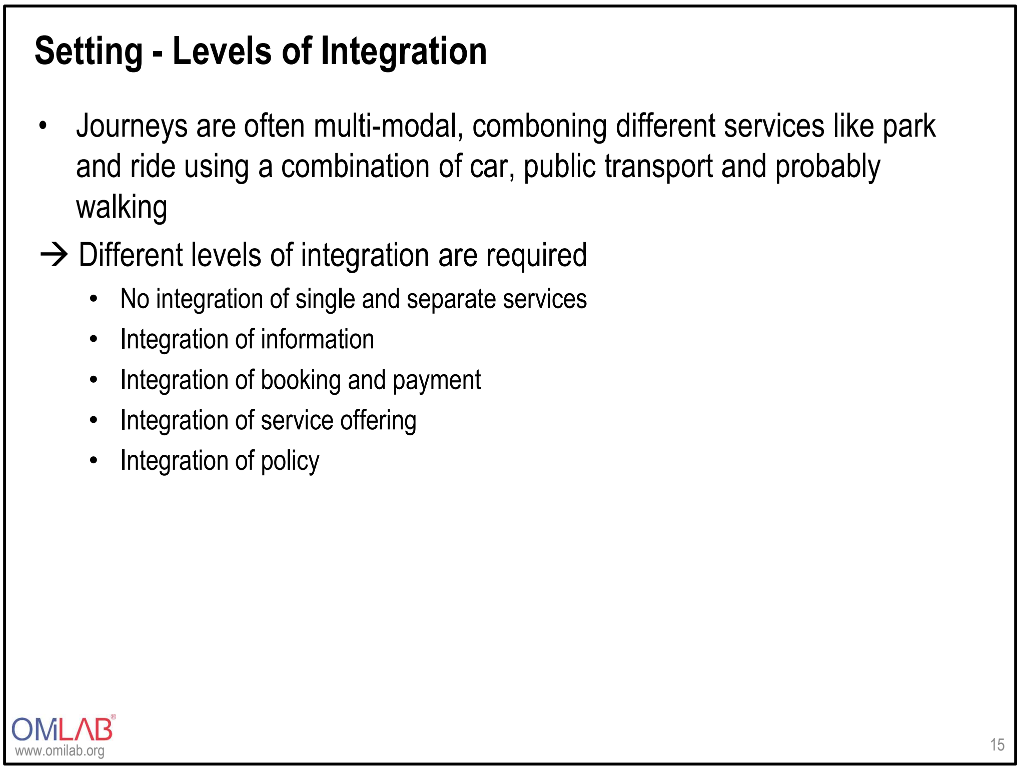
OMiLAB®
www.omilab.org

GSM =
GPRS =



Setting

- Technology: human-operated, assistant systems or autonomous operating systems
- From a mobility point of view, there is no difference if a human drives the car or if the cars drives autonomously
- Important aspects in the scenario are
  - Price
  - Distance to the starting and end points of the journey
  - Time efficiency
  - Resilience with respect to weather, comfort in case of carrying items and flexibility of the journey

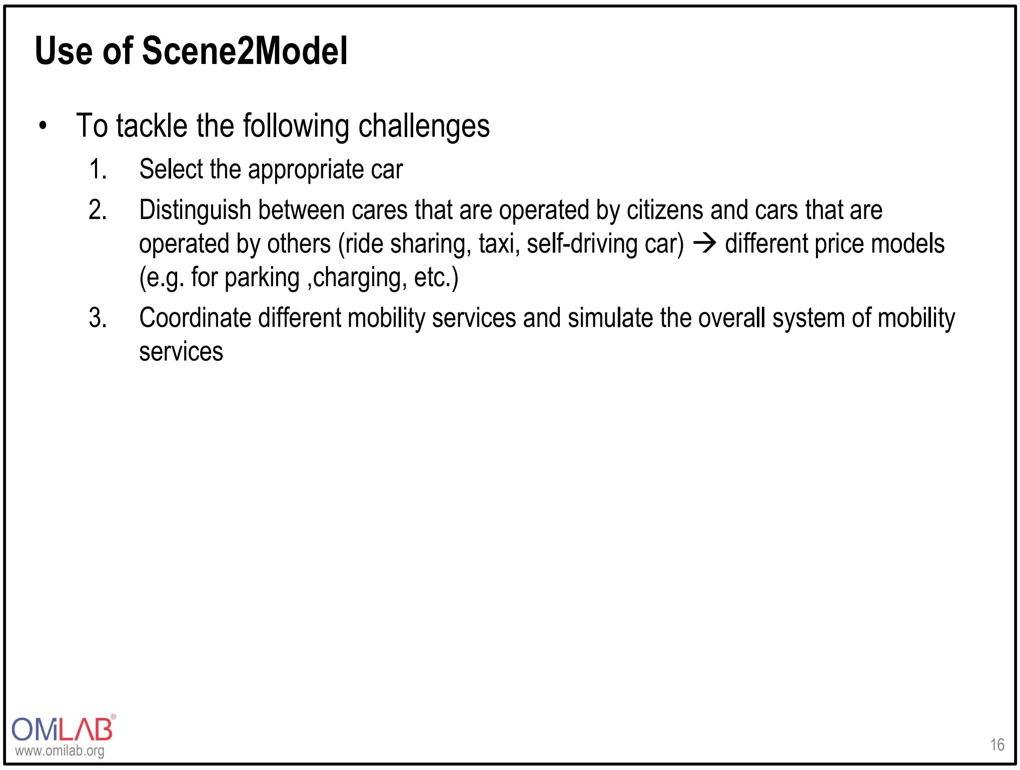


Setting - Levels of Integration

- Journeys are often multi-modal, comboning different services like park and ride using a combination of car, public transport and probably walking

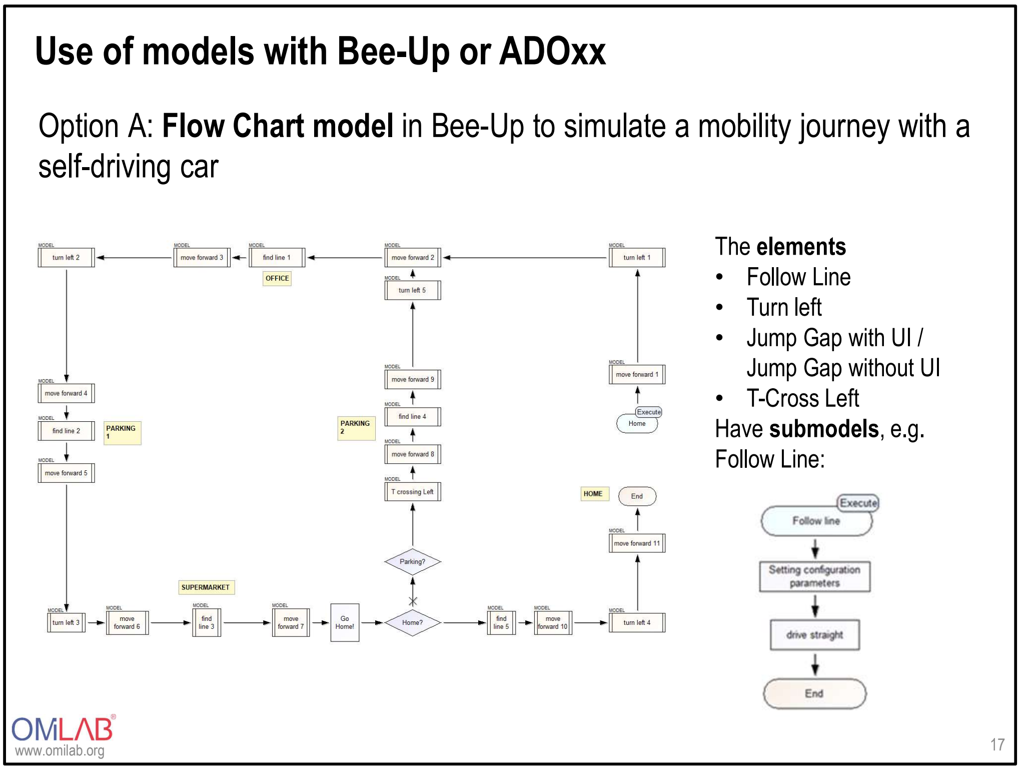
→ Different levels of integration are required

  - No integration of single and separate services
  - Integration of information
  - Integration of booking and payment
  - Integration of service offering
  - Integration of policy



Use of Scene2Model

- To tackle the following challenges
  1. Select the appropriate car
  2. Distinguish between cares that are operated by citizens and cars that are operated by others (ride sharing, taxi, self-driving car) → different price models (e.g. for parking ,charging, etc.)
  3. Coordinate different mobility services and simulate the overall system of mobility services



Use of models with Bee-Up or ADOxx

Option A: **Flow Chart model** in Bee-Up to simulate a mobility journey with a self-driving car

The **elements**

- Follow Line
- Turn left
- Jump Gap with UI / Jump Gap without UI
- T-Cross Left

Have **submodels**, e.g. Follow Line:

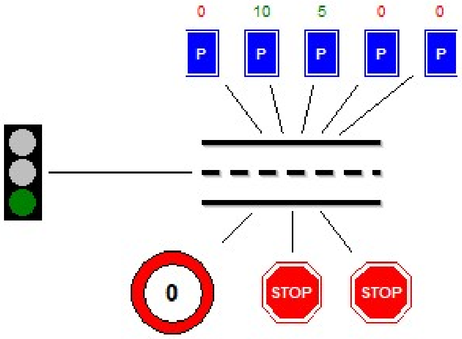
OMiLAB
www.omilab.org

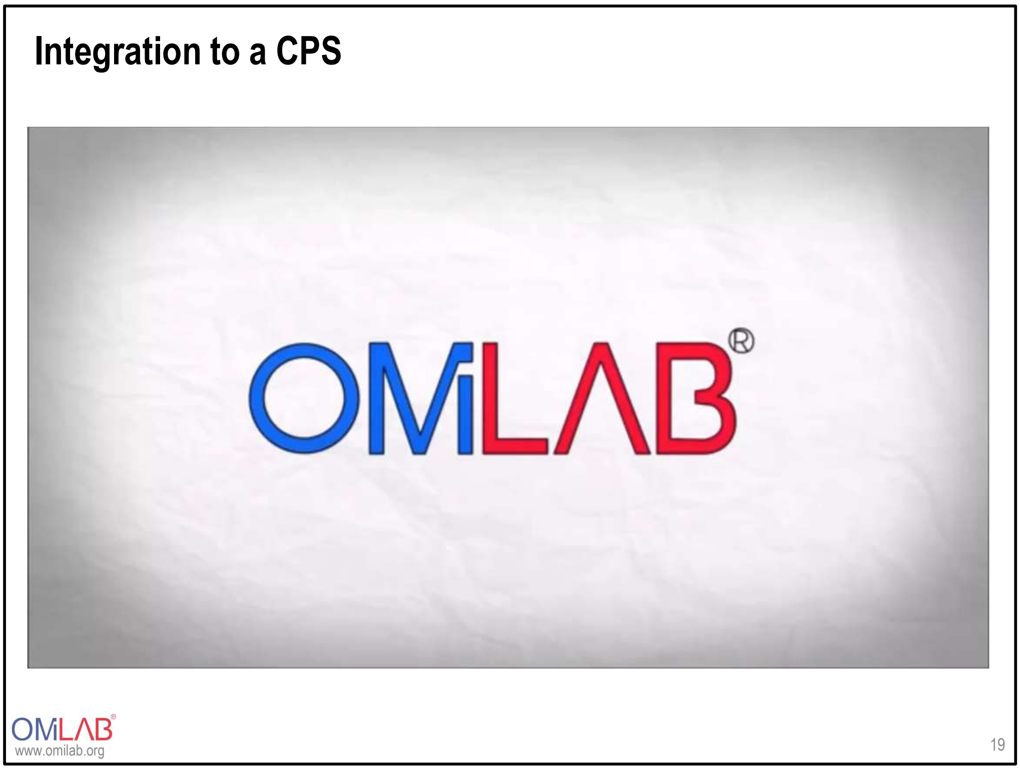
17

The figure presents the individual actions that have to be performed by the mBot using a flow chart that is laid-out in a street map-oriented way. The start is at the right side of the figures, with the start object called “Home”. There is the “Execution” button that starts the communication with the mBot.

Use of models with Bee-Up or ADOxx

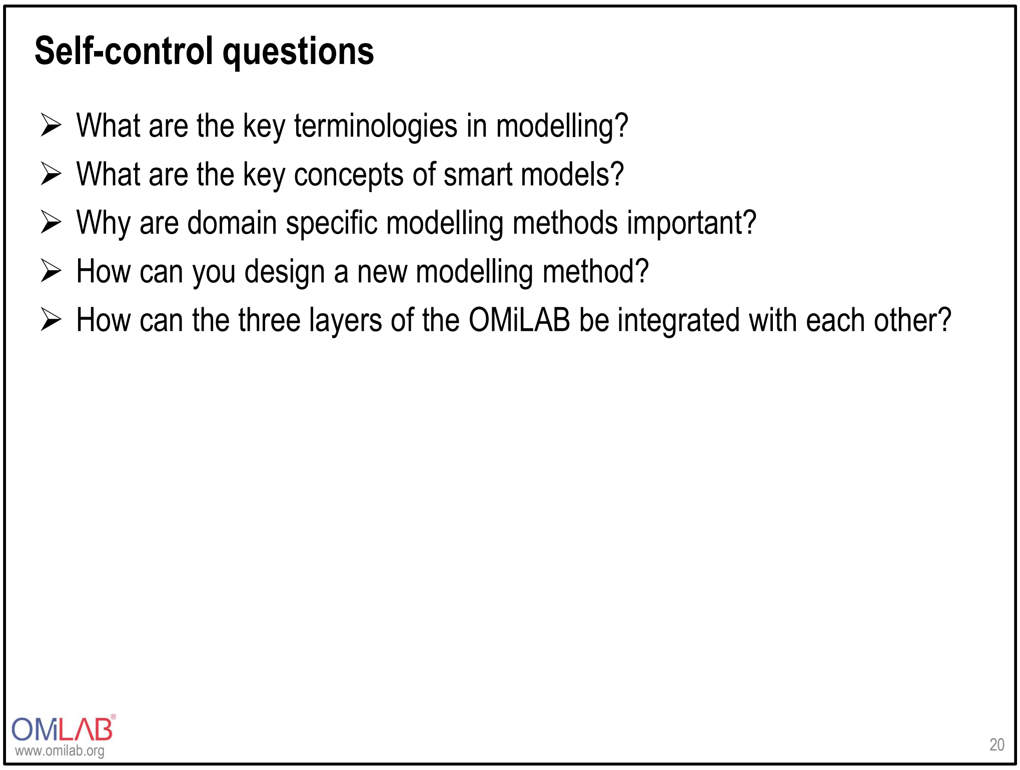
Option B: **Own modelling method** in ADOxx to simulate smart parking





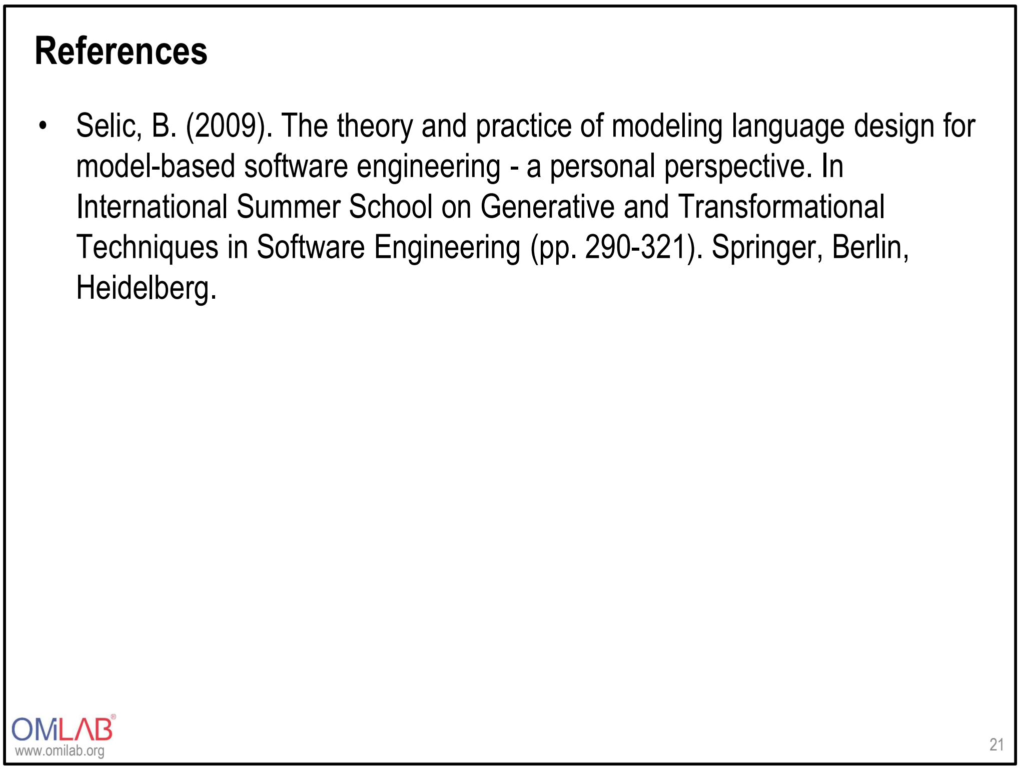
Integration to a CPS

Show the video “Module6_ExperimentNemoCar_OMiLAB” and discuss it.



Self-control questions

- What are the key terminologies in modelling?
- What are the key concepts of smart models?
- Why are domain specific modelling methods important?
- How can you design a new modelling method?
- How can the three layers of the OMiLAB be integrated with each other?



References

- Selic, B. (2009). The theory and practice of modeling language design for model-based software engineering - a personal perspective. In International Summer School on Generative and Transformational Techniques in Software Engineering (pp. 290-321). Springer, Berlin, Heidelberg.