Architecture and Architecture Description

Prof. Dr. Knut Hinkelmann

Chapter 2: Enterprise Architecture Description

- What is an Architecture?
- ISO/IEC/IEEE 42010 Systems and Software Engineering — Architecture Description
- Modeling and Meta-Modelling

Architecture – What is it?

- Is this an Architecture?

- No, this NOT the Architecture. This is the RESULT of architecture.
- In the result you can see the Architect's "architecture"
- The result is an implementation, an instance

Adapted from Zachman (2012)

Enterprise – What is it?

- An "Architecture" (for anything) would be the total set of descriptive representations (models) relevant for describing a complex object such that it can be created and that constitute a baseline for changing the object after it has been instantiated.
- Therefore "**Enterprise Architecture**" would be the total set of models relevant for describing an Enterprise, that is, the descriptive representations required
  - to create a (coherent, optimal) Enterprise and
  - to serve as a baseline for changing the Enterprise once it is created.

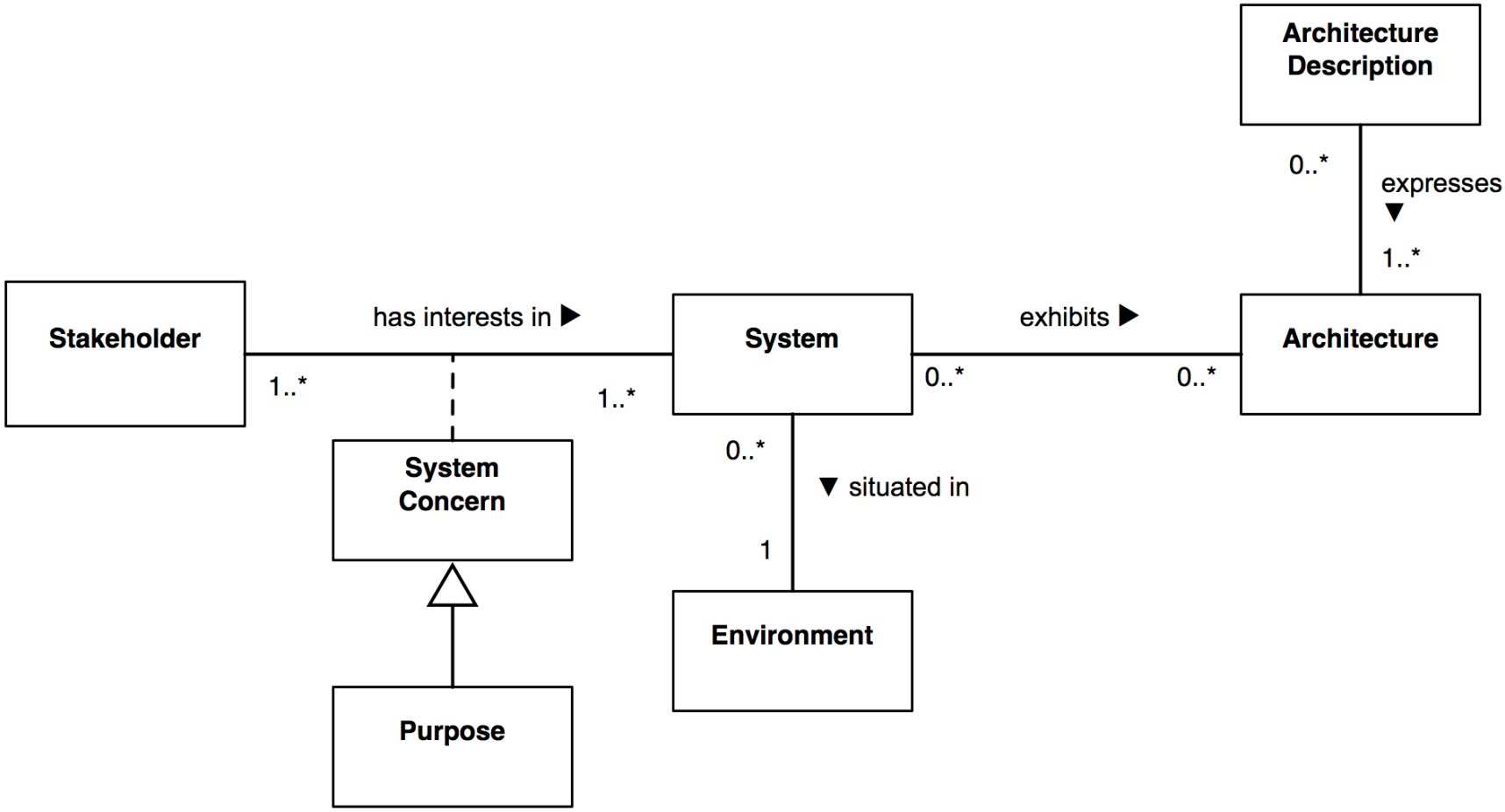
Adapted from Zachman (2012)

ISO/IEC/IEEE 42010 Systems and Software Engineering — Architecture Description

- International standard for architecture descriptions of systems and software.
- The original IEEE 1471 specified requirements on the contents of **architecture descriptions** of systems.
  - An architecture description (AD) expresses the architecture of a system of interest
- ISO/IEC/IEEE 42010 adds definitions and requirements on **architecture frameworks** and **architecture description languages** (ADLs)

http://www.iso-architecture.org/ieee-1471/faq.html

ISO/IEC/IEEE 42010
A Conceptual Model of Architecture Description



http://www.iso-architecture.org/ieee-1471/cm/

n|w

Key Ideas of ISO/IEC/IEEE 42010:
System

- ISO/IEC/IEEE 42010 is about *System Architecture*
- The Standard, however, takes no position on the question, *What is a system?*
- The term "*system*" could refer to an enterprise, a product line, a service, a subsystem, or software.
- Systems can be man-made or natural. Users of the Standard are free to employ whatever *system theory* they choose.
- The premise of the Standard is, *For a system of interest to you, the Standard provides guidance for documenting an architecture of that system.*

http://www.iso-architecture.org/ieee-1471/cm/

n|w

Key Ideas of ISO/IEC/IEEE 42010:
Architecture

- "*Architecture*" names that which is fundamental about a system; the set of essential properties of a system which determine its form, function, value, cost, and risk.
- That which is **fundamental** to a system takes several forms:
  - its **elements**: the constituents that make up the system;
  - the **relationships**: both internal and external to the system; and
  - the **principles of its design and evolution**.

http://www.iso-architecture.org/ieee-1471/cm/

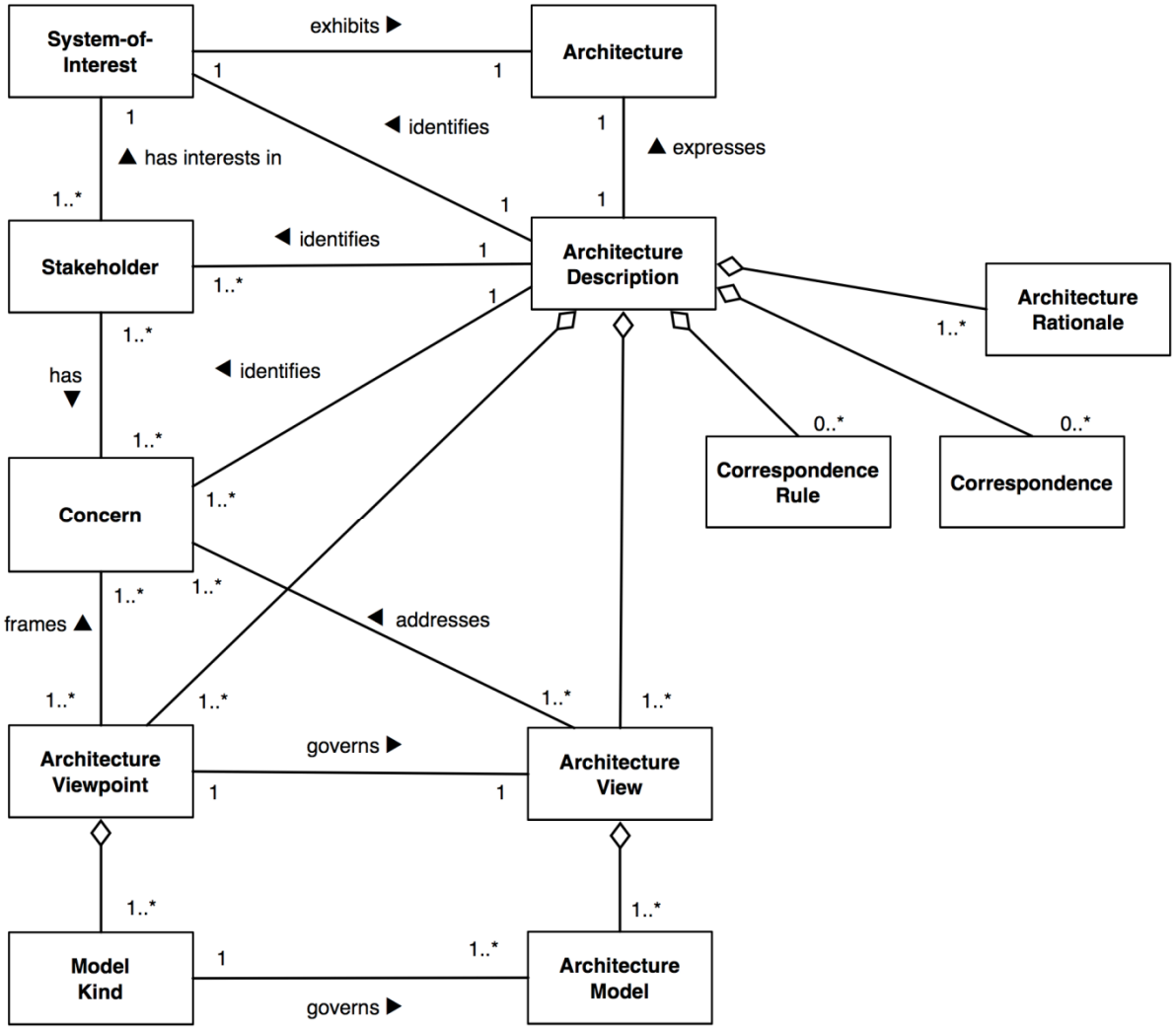
n|w

Key Ideas of ISO/IEC/IEEE 42010:
Architecture and Architecture Description

- An architecture is a *conception of a system* – i.e., it is in the human mind. An architecture may exist without ever being written down.
- An *architecture description* (AD) is an artifact that expresses an Architecture to share with others.
  - An AD is what is written down as a concrete work product. It could be a document, a repository or a collection of artifacts used to define and document an architecture
  - Architects and other system stakeholders use Architecture Descriptions to understand, analyze and compare Architectures, and often as "blueprints" for planning and construction.

http://www.iso-architecture.org/ieee-1471/cm/

The Core of Architecture Description



http://www.iso-architecture.org/ieee-1471/cm/

n|w

Key Ideas of ISO/IEC/IEEE 42010

Environment

- Every System exists in its *Environment*, it acts upon that Environment and vice versa.
- A System's Environment determines the range of influences upon the system. It include developmental, operational, technical, political, regulatory, and *all other influences which can affect the architecture*.
- The environment of a system is understood through the identification of the *stakeholders* of the system and their concerns.

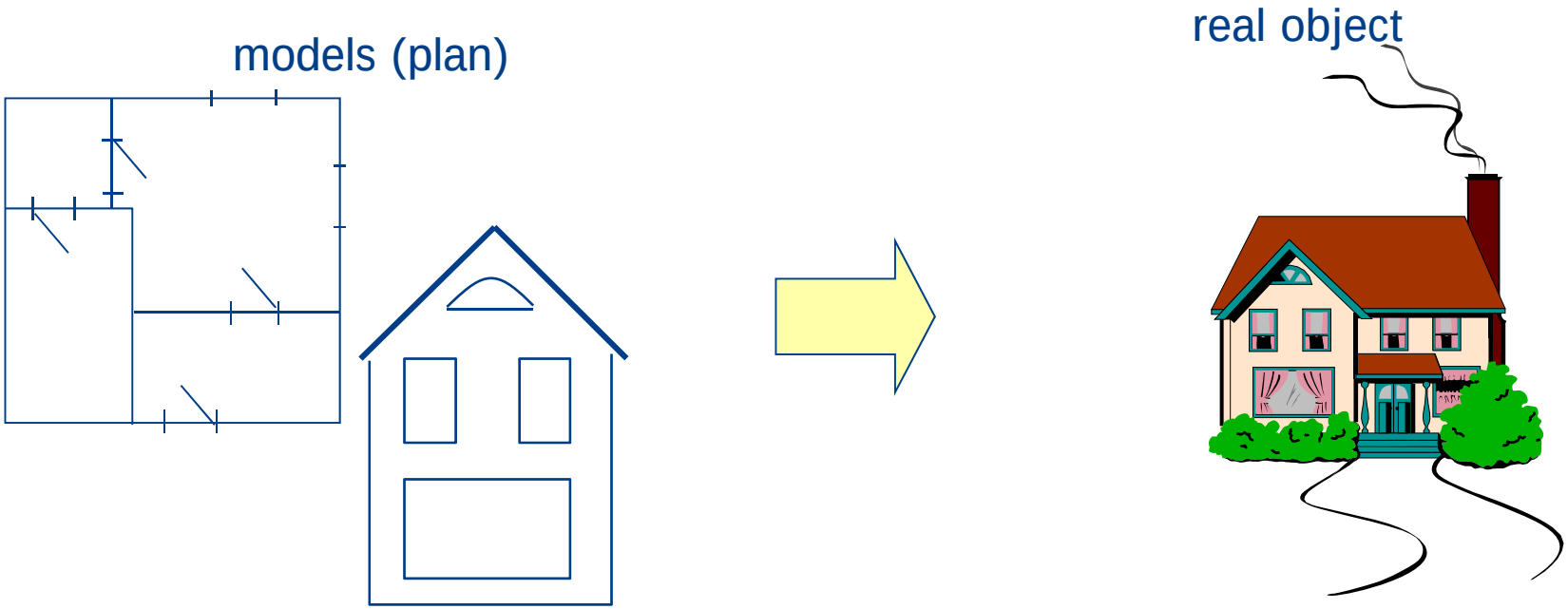
n|w

Key Ideas of ISO/IEC/IEEE 42010
Stakeholder an Concerns

- The environment of a system is understood through the identification of the *Stakeholders* of the system and their *Concerns*.
- A *Concern* is any interest in the system.
  - **Examples of System Concerns**: agility, behavior, business goals, business strategy, complexity, customer experience, flexibility, functionality, maintainability, purpose, quality of service, regulatory compliance, security, structure.
- *Stakeholders* are individuals, groups or organizations holding concerns for the System.
  - **Examples of Stakeholders**: client, owner, user, operator, maintainer, developers, suppliers, regulator, auditor, architect.

Models

- An *Architecture Description* consists of one of several *Architecture Models*
- A Model is a reproduction of a relevant part of reality which contains the essential aspects to be investigated.



Rationale for Modelling

- Models provide abstractions of an object or a system that allow engineers to reason about that system by ignoring extraneous details while focusing on relevant ones.
- All forms of engineering rely on models to understand complex, real-world systems.
- Models are used in many ways:
  - predict system qualities
  - reason about specific properties when aspects of the system are changed
  - communicate key system characteristics to various stakeholders
- There are different models of a systems which are specific for the different kinds of interests and uses

(Brown 2004)

Architecture Views and Viewpoints

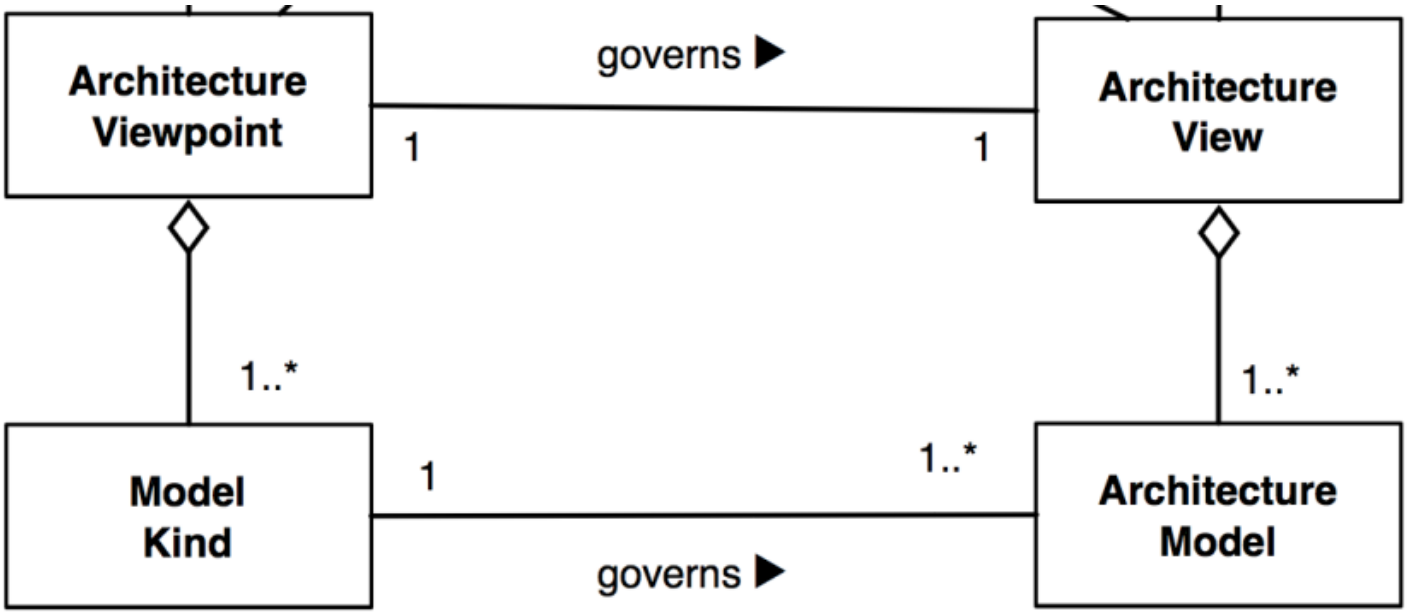
- Not everyone is interested in everything. Views and Viewpoints are a means to specify which part of an Architecture Description is of relevance
- *View*: Part of an architecture description that
  - addresses a set of related *concerns* and
  - is addressed to a set of *stakeholder*
- *Viewpoint* specifies a view
  - prescribes the concepts, models, analysis techniques, and visualizations that are provided by the view

A *view* is what you see and
a *viewpoint* is where you are looking from

What is and what is not shown in a view depends on the scope of the viewpoint and on what is relevant to the concerns of the stakeholders

Source: ArchiMate 2.0 Specification, chapter 8, http://pubs.opengroup.org/architecture/archimate2-doc/chap08.html

Views, Viewpoints, Model Kinds and Models



ISO/IEC/IEEE 42010

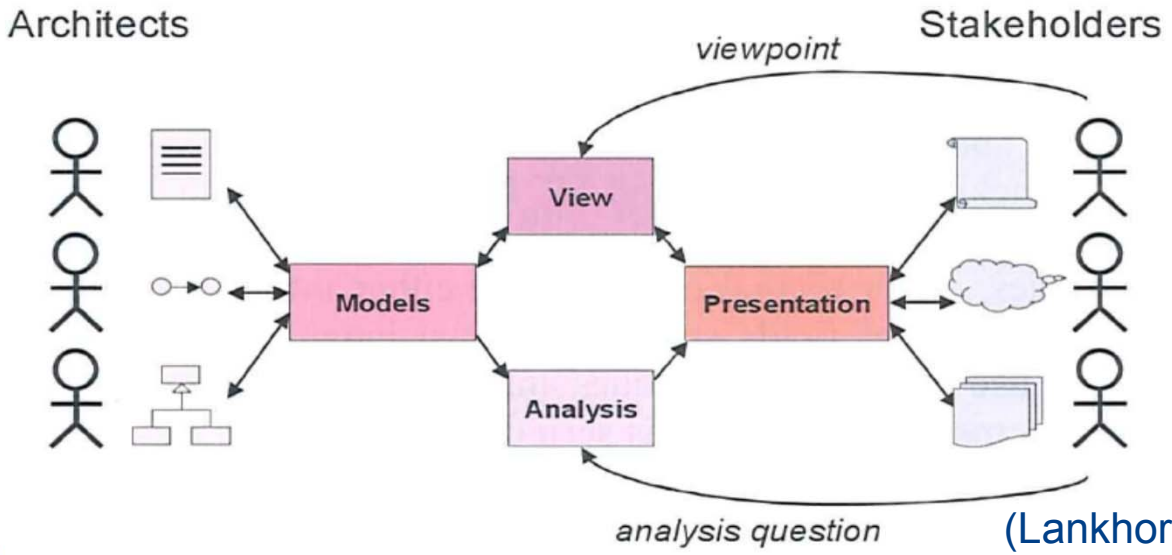
Architecture Viewpoints and Views

- An *Architecture View* expresses the Architecture of the System from the perspective of one or more Stakeholders to address specific Concerns, using the conventions established by its viewpoint.
- An Architecture View consists of one or more *Architecture Models*.
- An *Architecture Viewpoint* is a set of conventions for constructing, interpreting, using and analyzing one type of Architecture View.
  - Examples of viewpoints: operational, systems, technical, logical, deployment, process, information.

http://www.iso-architecture.org/ieee-1471/cm/

Communicating about Architecture

- Viewpoints are designed for the purpose of communicating certain aspects of an architecture.
- Viewpoints are a means to focus on particular aspects of the architecture;
- the aspects are determined by the concerns of the stakeholder with whom communication takes place.
- The architect informs the stakeholders, and the stakeholders give feedback on the presented aspects.



(Lankhorst et al. 2005, p. 4)

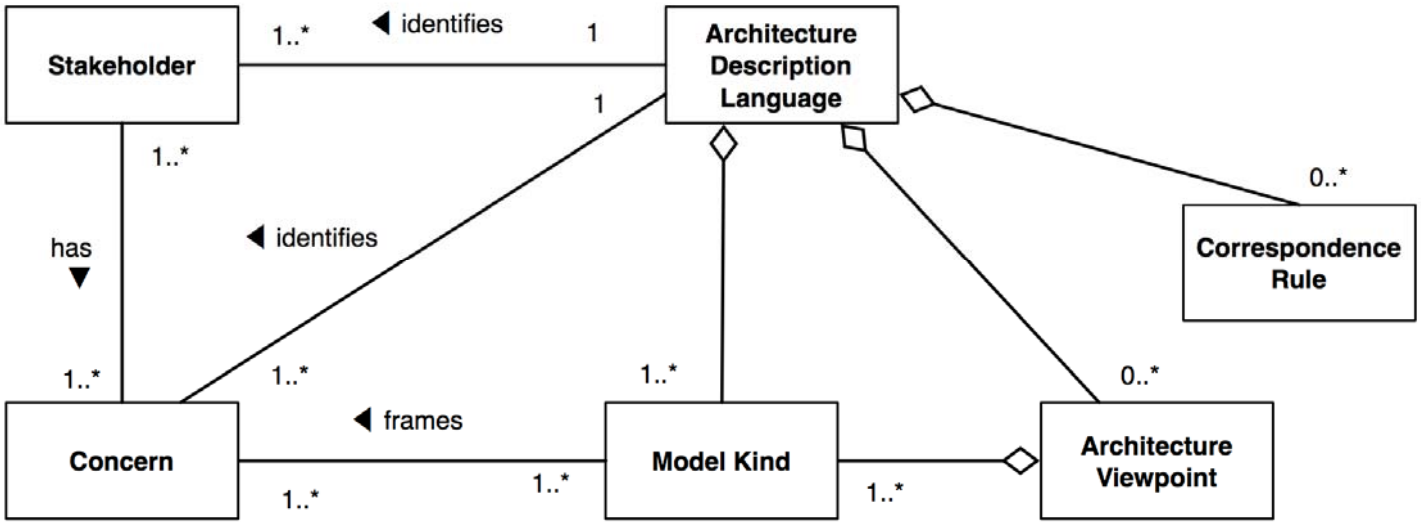
Architecture Models

- A view is a part of the Architecture Description; it is comprised of *Architecture Models*.
- An *Architecture Model* is constructed in accordance with the conventions established by its Model Kind, typically defined as part of its governing viewpoint.
  - Examples of Models: The model of the order process of the company, the model of the customer data, the organisation of a specific company
- A *Model Kind* defines the conventions for a type of Architecture Model.
  - Examples of model kinds are process models, organisation model, data models

http://www.iso-architecture.org/ieee-1471/cm/

Architecture Description Language

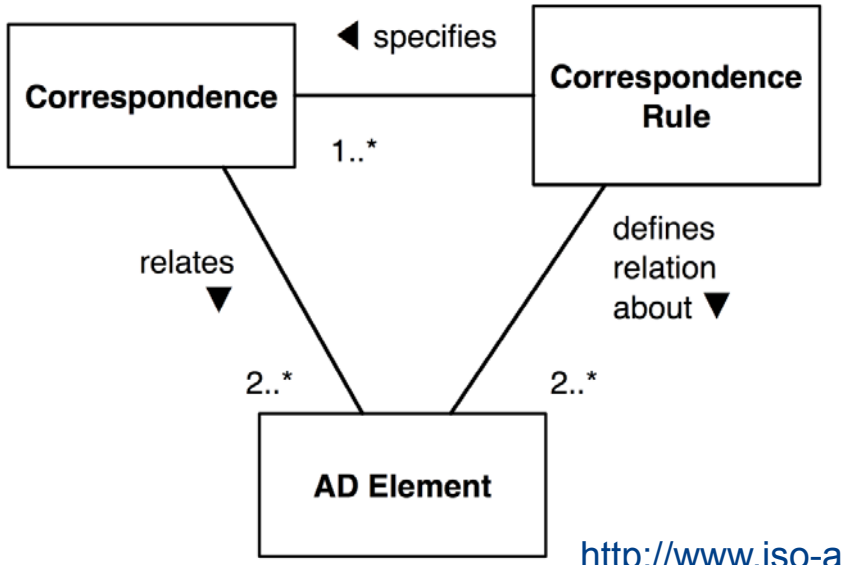
- An *Architecture Description Language (ADL)* is any form of expression for use in Architecture Descriptions. An ADL might include a single Model Kind, a single viewpoint or multiple viewpoints.
  - Examples of ADLs ،



http://www.iso-architecture.org/ieee-1471/cm/

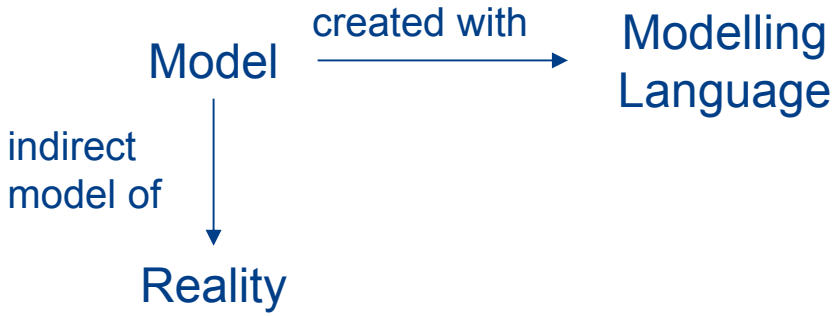
AD Elements and Correspondences

- Any item in an Architecture Description (AD) is considered an *AD Element*.
- *Correspondences* express a relation between AD Elements.
- *Correspondence Rules* enforce relations within an Architecture Description or between Architecture Descriptions.



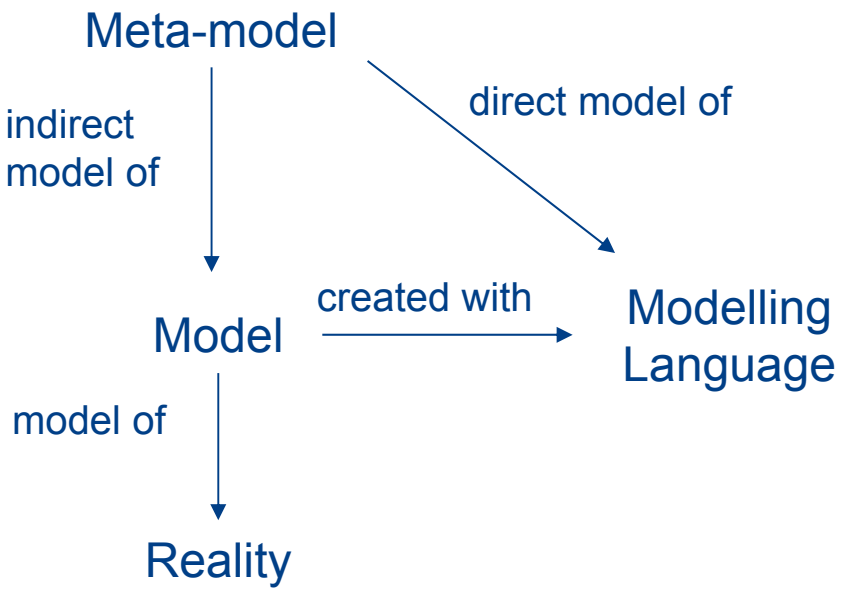
http://www.iso-architecture.org/ieee-1471/cm/

Modelling Language



- A modelling "language" specifies the building blocks (elements) from which a model can be made.
- There can be different types of modelling languages, depending on the kind of model
  - graphical model
  - textual description
  - mathematical model
  - conceptual model
  - physical model

Meta-model



A meta-model defines the modelling language, i.e. the building blocks that can be used to make a model. It defines the

- object types that can be used to represent a model
- relations between object types
- attribtues of the object types
- meaning of the object types
- rules to combine object types and relations

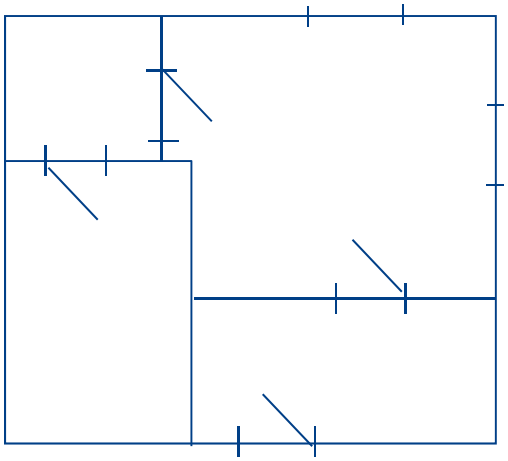
- There typically is a meta-model for each Model Kind

Model and Meta-Model in Architecture

real object

house

model



architect's drawing (plan)

meta-model
(modelling language)

object types:

wall

door

window

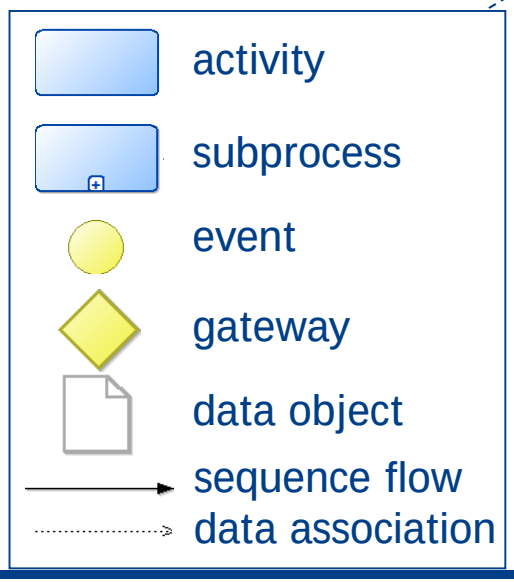
rules:

- a door is adjacent to a wall on both sides
- Windows are on outer walls.

Illustration: Meta-model and Model for Processes

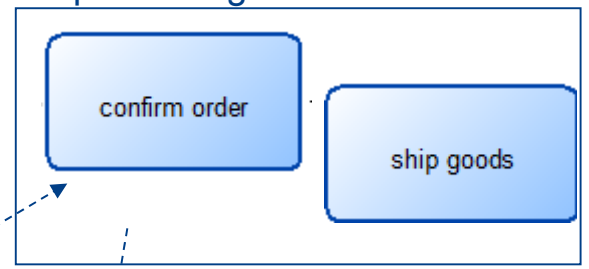
Meta-model:
A process model consists of object types for active elements «activity» and «subprocess», control elements «events», «gateways» artifacts «data object» and the relation types «sequence flow» and «data association». The elements have attributes and there are rules how the elements can be combined.

Modeling Language:
Syntax (appearance) and semantics of meta-model elements



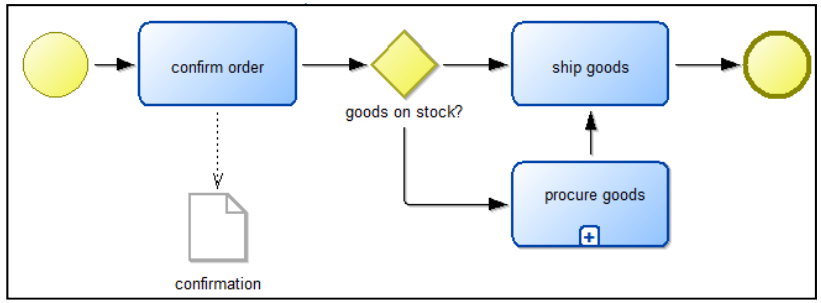
Objects:
Instances of object types representing real entities

Object „confirm order" is an instance of the object type "activity"



Instance of "confirm order" in a particular model

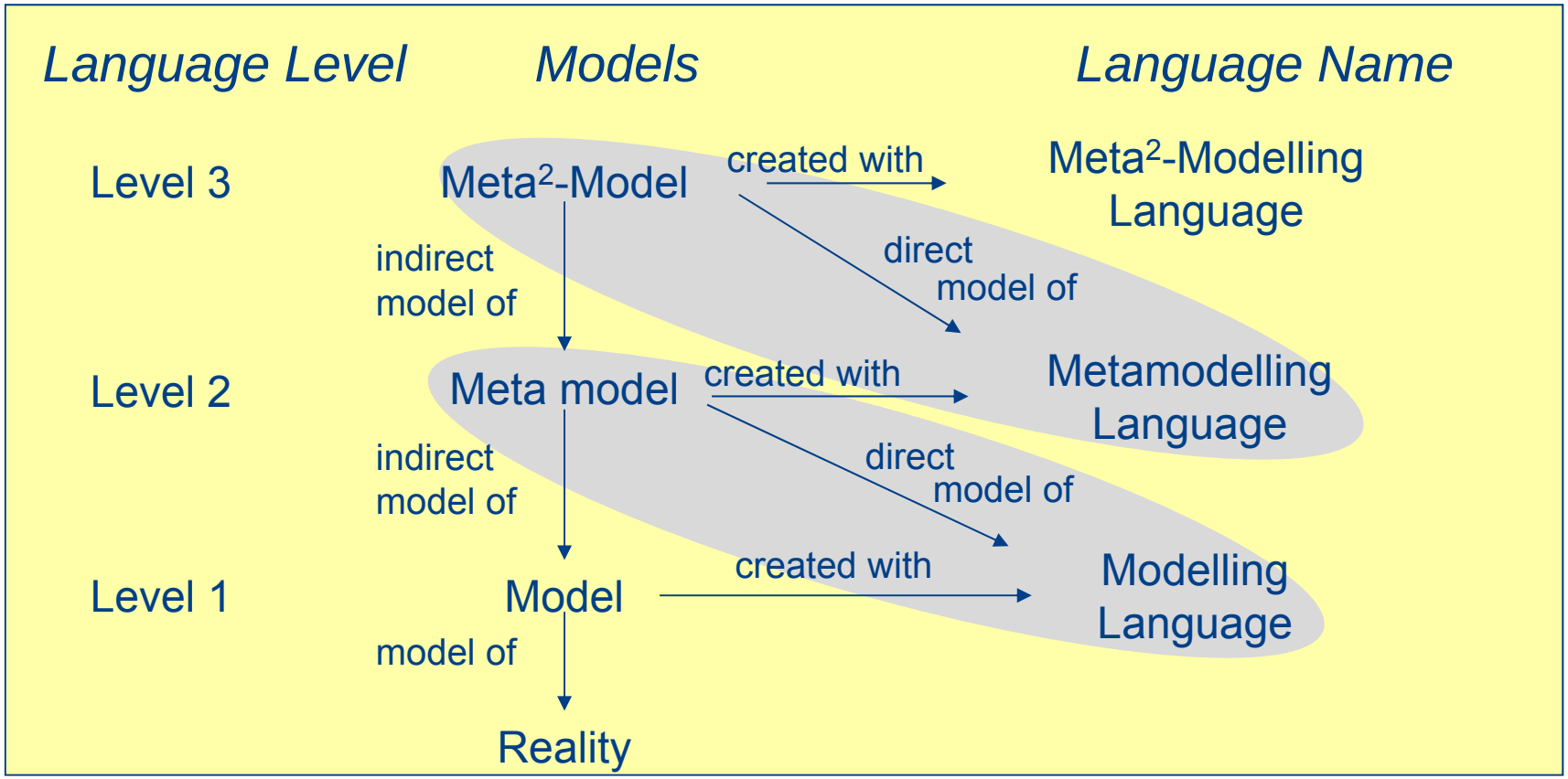
Model:



combination of objects representing a process

Meta Model Hierarchy

The meta-model must again be described in some language, which has to be specified in a meta-model

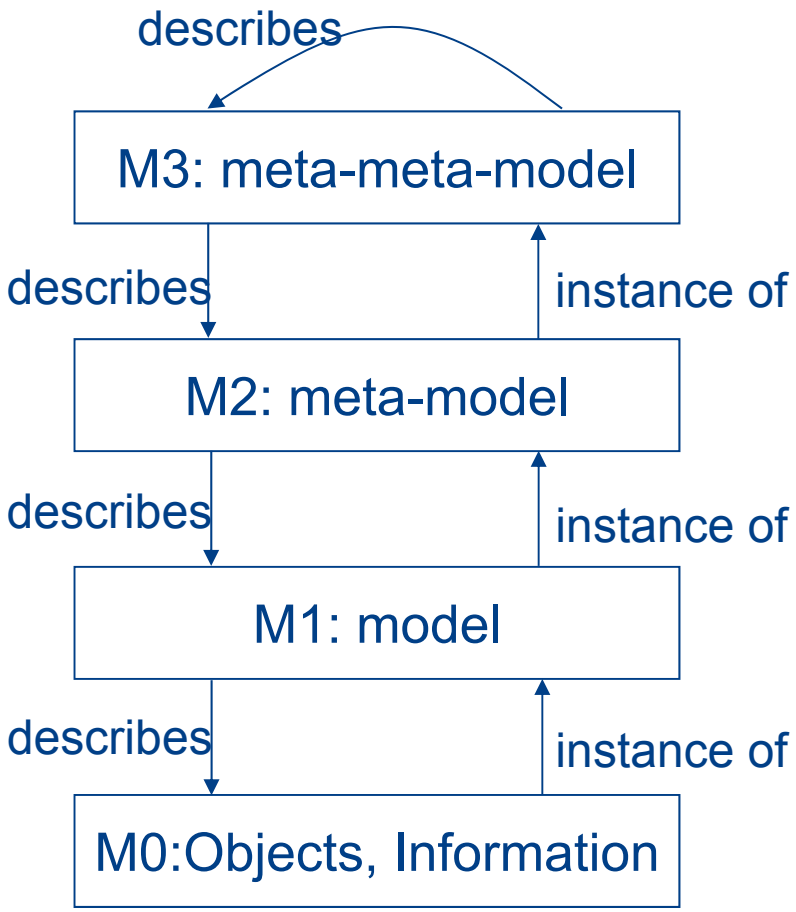


Often the meta-model and the modeling language are unified and not distinguished.

MOF – Meta Object Facility

- The Meta Object Facility (MOF) is an OMG meta-modeling standard.
- MOF is itself a *meta-meta-model*, a specification describing how one may build meta-models.
- MOF is closely based on Unified Modeling Language (UML):
  - Meta-models are represented with class diagrams of UML
- MOF defines the theoretical underpinnings of the XML Metadata Interchange (XMI)
  - XMI is a standard syntax for the Exchange of Models

The OMG Model Stack

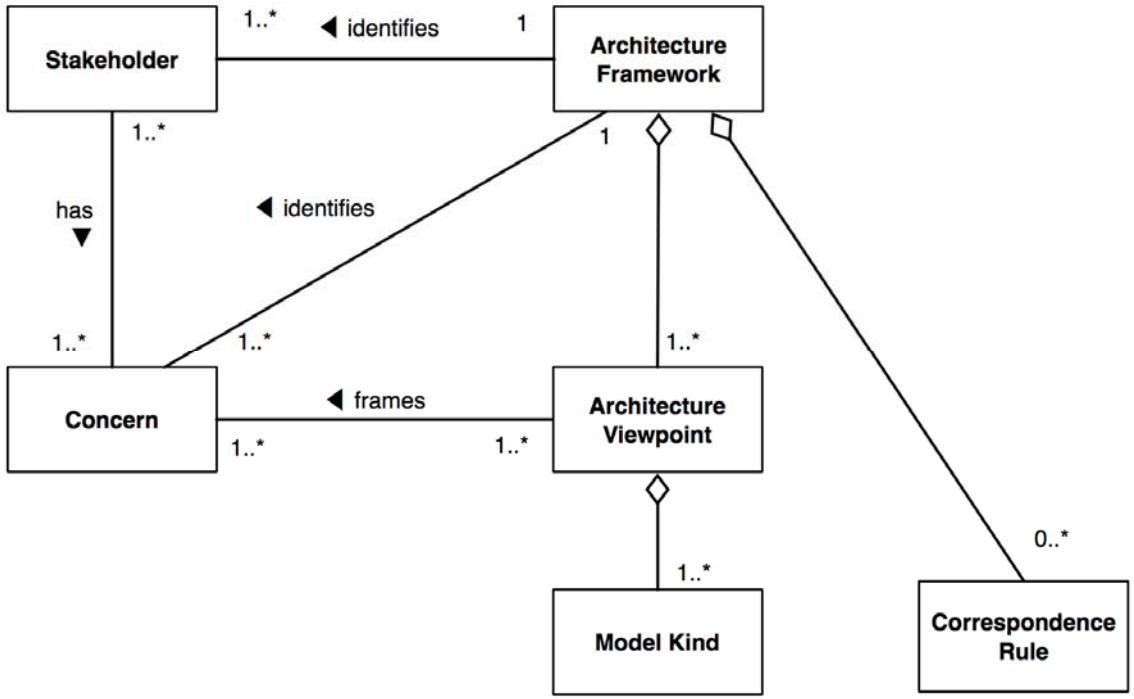


The Meta Object Facility (MOF) distinguishes four levels:

- M0 is the basic data, the lifeblood of the business
  - the customer name "Peter Miller", the price "$291.70".
- M1 is the metadata: schemas and interfaces describing the structure of the data.
  - a table customer with a name column
- M2 is the meta-model, or the "IT language" - specifying the concepts of the modelling language
  - "A relational database has tables, each table has zero or more columns".
- M3 is the MOF specification itself, which allows us to draw the boxes-and-arrows of UML

Architecture Framework

- An *Architecture Framework* establishes a common practice for creating, interpreting, analyzing and using architecture descriptions within a particular domain of application or stakeholder community.



http://www.iso-architecture.org/ieee-1471/cm/