GERA Modelling Framework, Enterprise Models and Modelling Languages

Associate Professor Peter Bernus

Griffith University, Brisbane, Australia

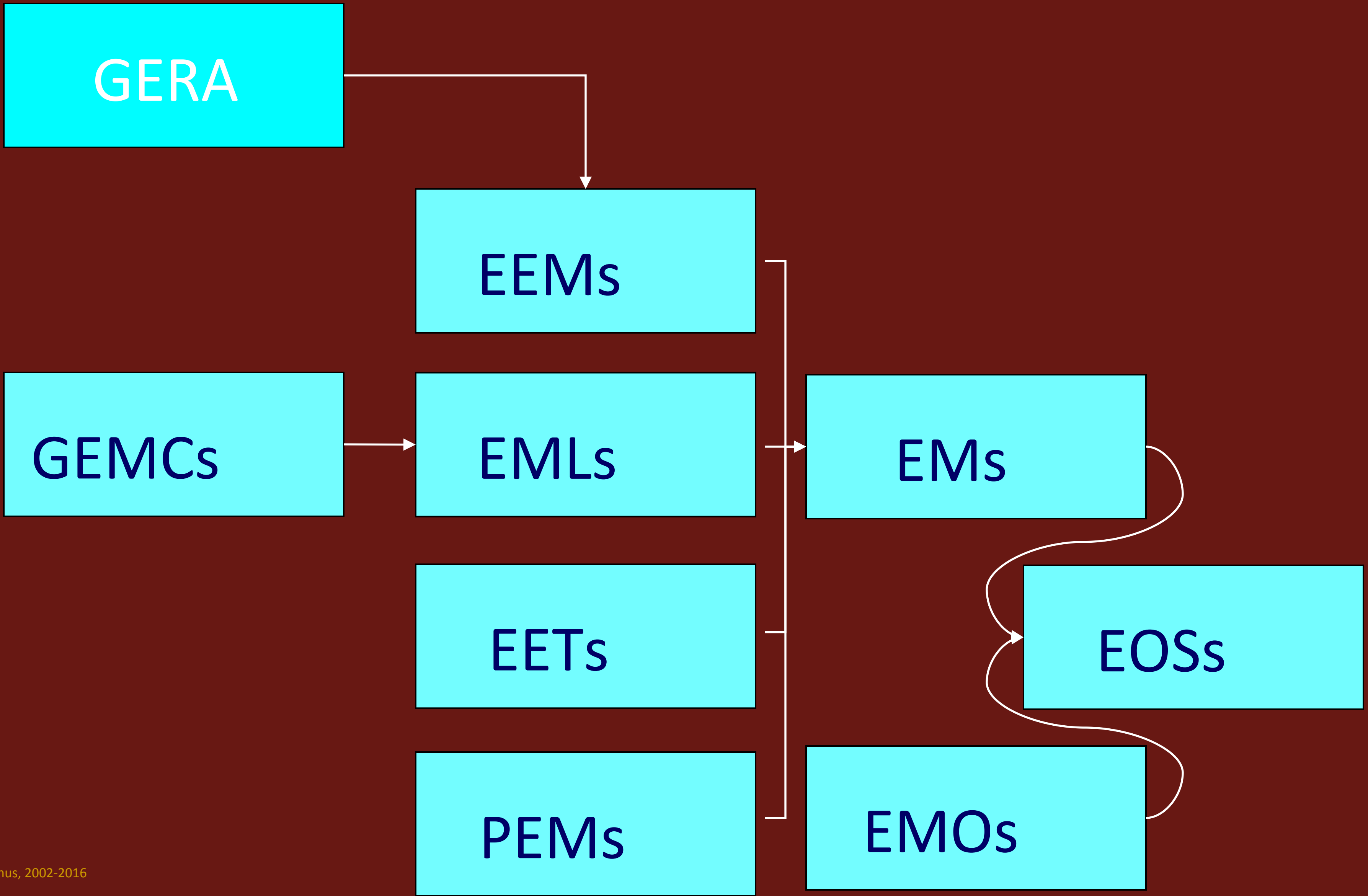
Overview

- The GERA Modelling Framework of GERAM
  (defines the scope of enterprise modelling)

- Enterprise Models
  (of any particular enterprise entity)

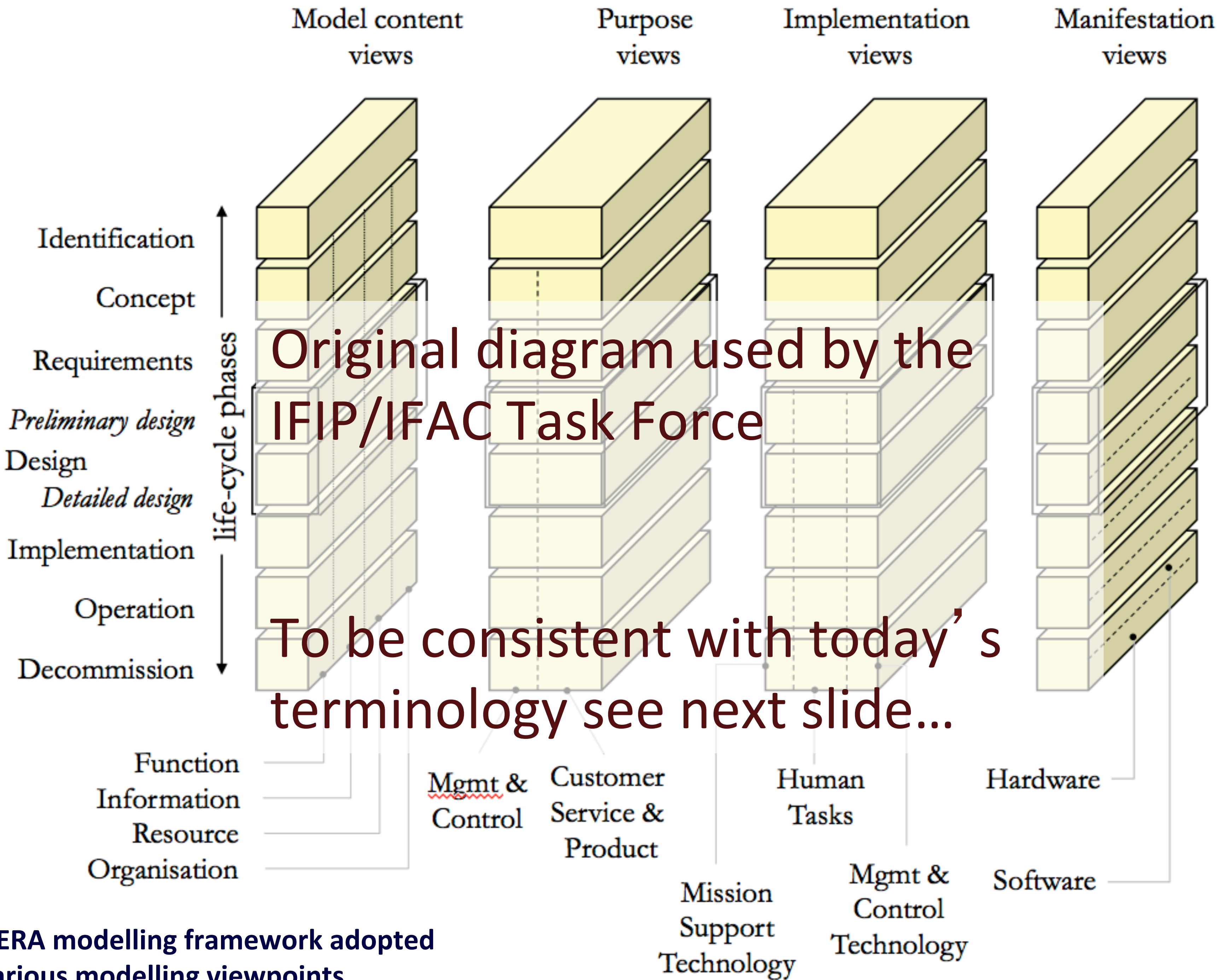
- Enterprise Modelling Languages
  (these can be used to create models)

Recall the Components of GERAM



How is it possible to combine the proposed viewpoints of different proposed modelling frameworks?

- We realise that these frameworks use different criteria to divide / systematise the enterprise descriptions that one might want to create
- None of these systematisations are 'best'
- We can use the combination of these to strengthen their use individually or in combination
- Other useful systematisations may exist
- However, we intended to develop a systematisation that guarantees completeness (why this is so is discussed later)



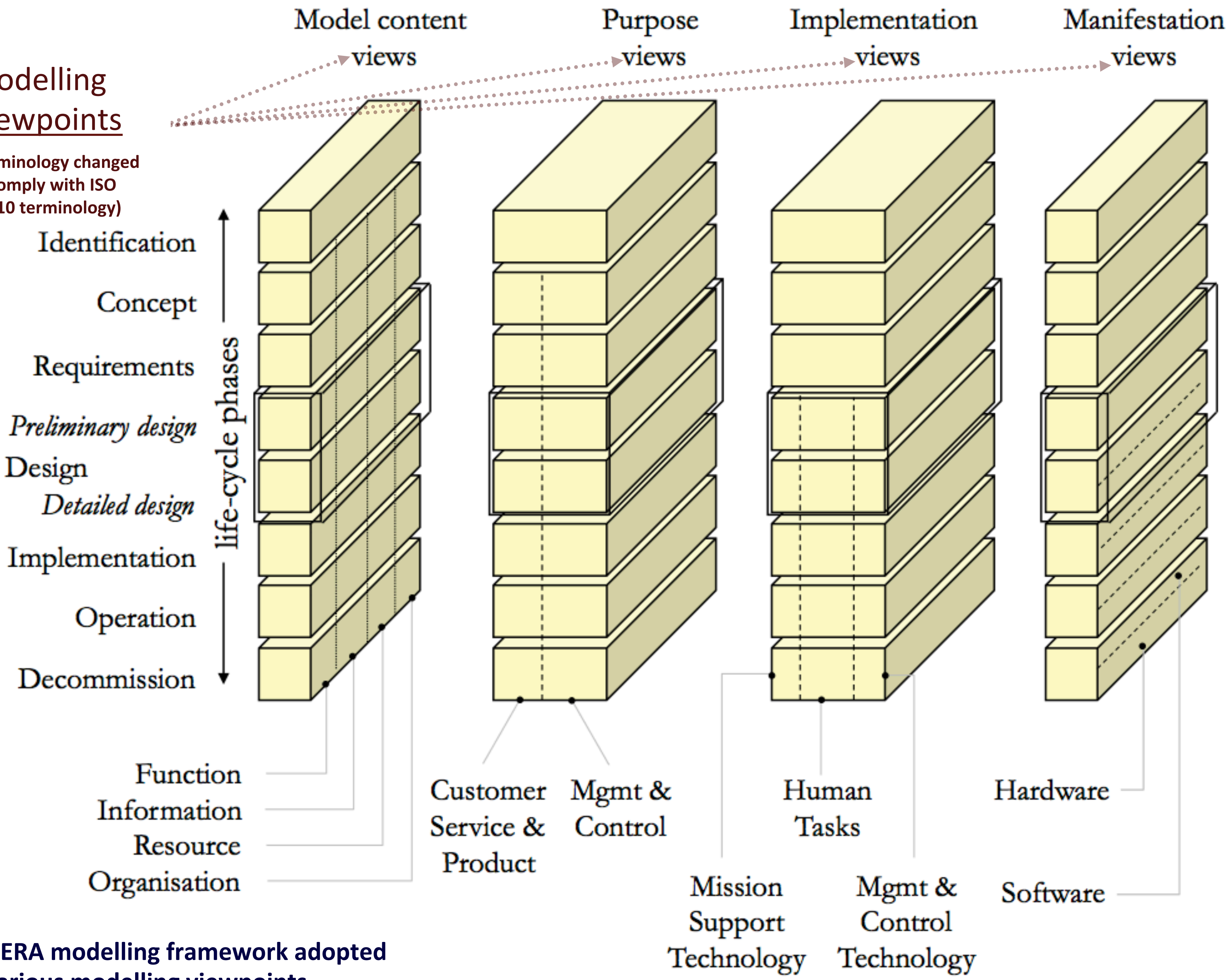
Original diagram used by the IFIP/IFAC Task Force

To be consistent with today's terminology see next slide…

GERA modelling framework adopted various modelling viewpoints …

Modelling Viewpoints

(terminology changed to comply with ISO 42010 terminology)

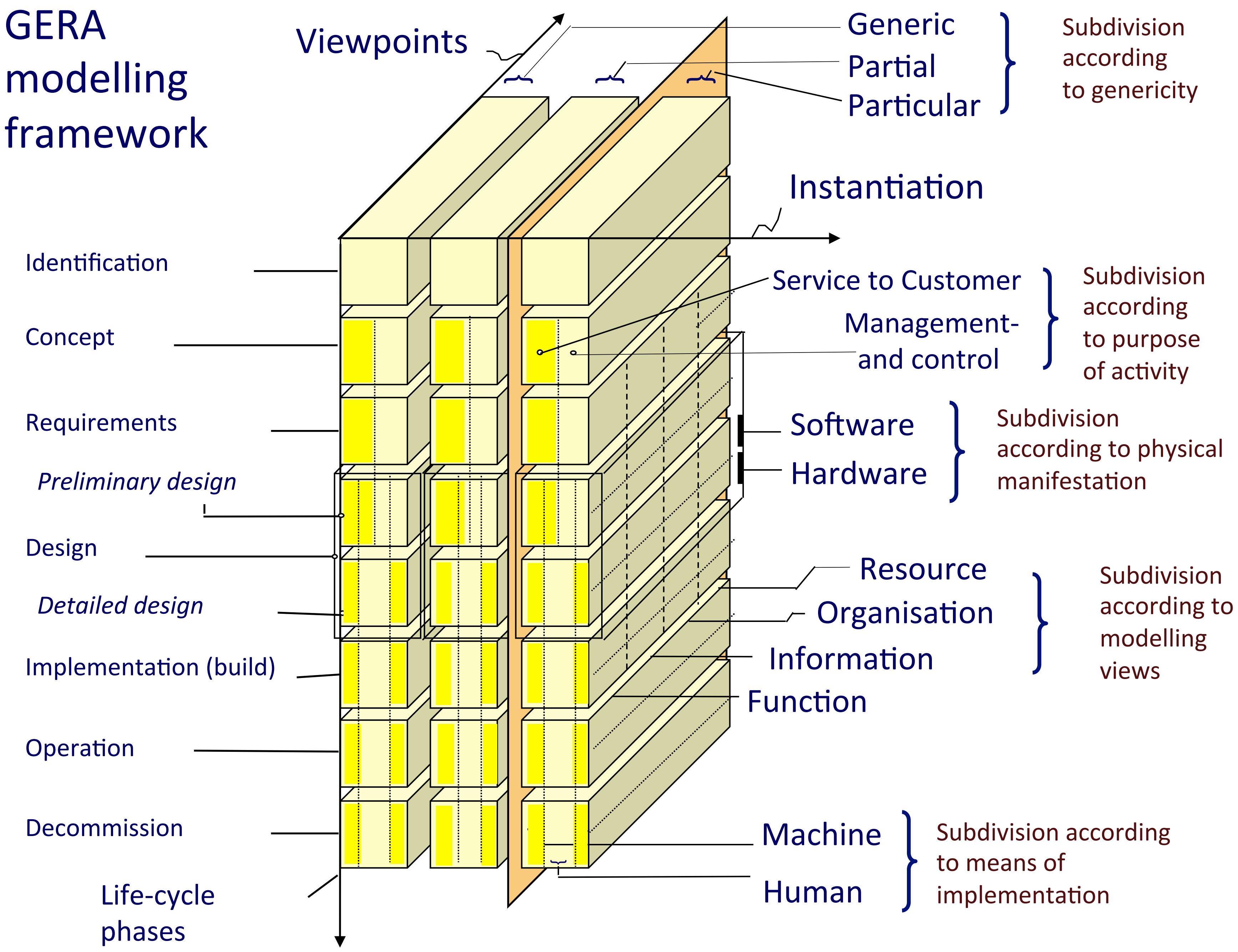


GERA modelling framework adopted various modelling viewpoints

GERA Enterprise Modelling Framework

- This framework combines what several other frameworks have to say about enterprise modelling
- The GERA Enterprise Modelling Framework defines the scope of enterprise modelling

GERA modelling framework



- Notice that the side-views of this cube correspond to the modelling frameworks of the contributing architecture frameworks (PERA, CIMOSA)
- The modelling frameworks of the various known architecture frameworks all map to the GERA modelling framework (there exist mappings onto GERA of the Zachman, DoDAF, TOGAF, ARIS, TOGAF and many others)

EMs

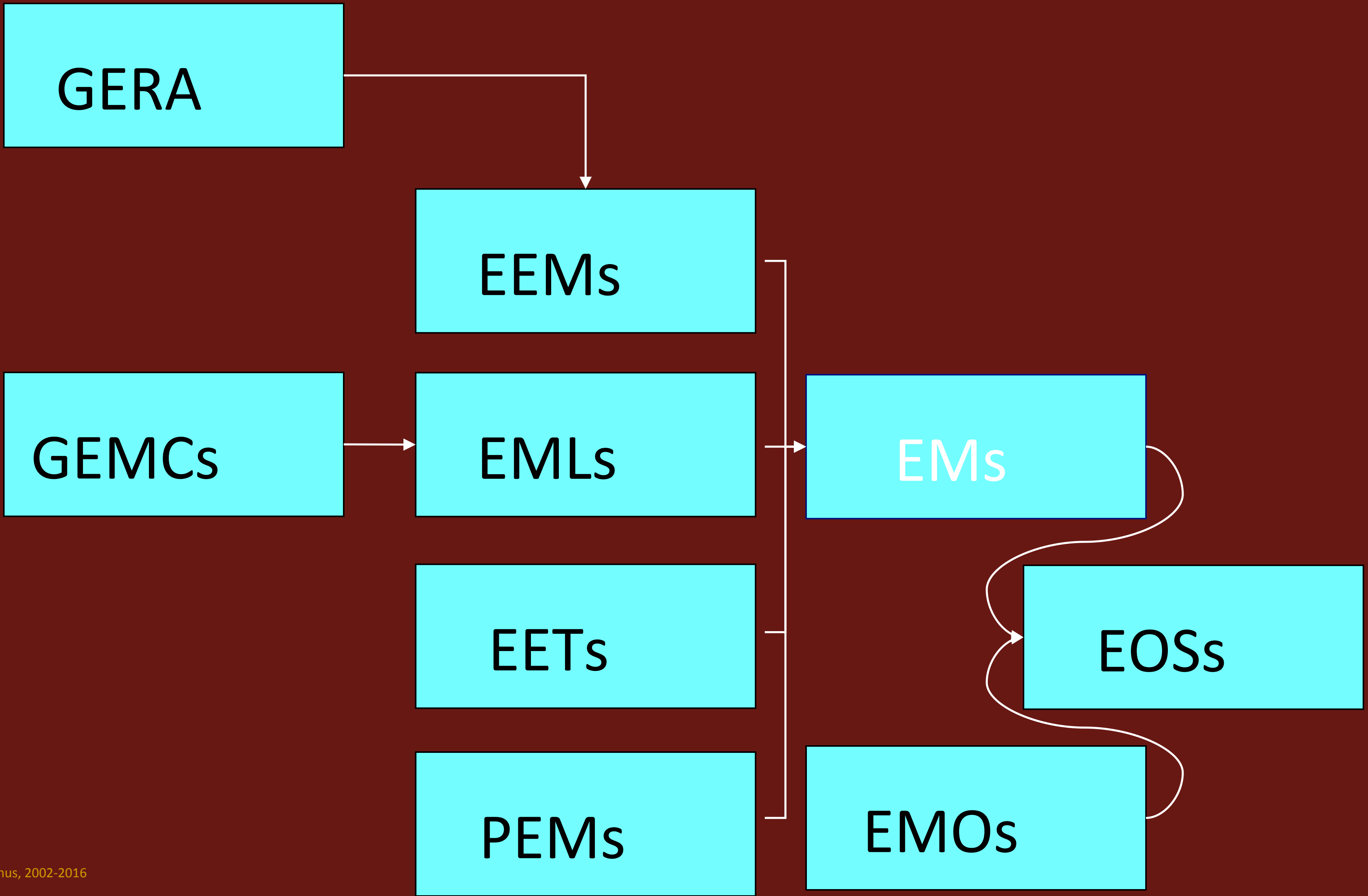
Enterprise Models

represent the particular

enterprise

… these are all those models and descriptions that represent the particular enterprise (entity) across its life-cycle (the same entity is described on different levels of abstraction) and across its life history (the same entity is described in different stages of its life, i.e., models have version history)

The Components of GERAM



- Potentially every area of the GERA modelling framework can be populated with models or descriptions of an enterprise entity, or of its parts)

- Often these models exist in the enterprise for the production facility and for the control system / information system, but not the part of the business processes done by humans and not for the entire system of management (decisions)

- Note that the <u>models are / should be produced to satify some need</u>, i.e., to answer some question (‘stakeholder concern’), which can be described as the stakeholder’s ‘*viewpoint*’

  Different stakeholders need different levels of detail: therefore we should package these extracts of models into ‘*views*’ to satisfy the needs of stakeholders (ISO 42010)

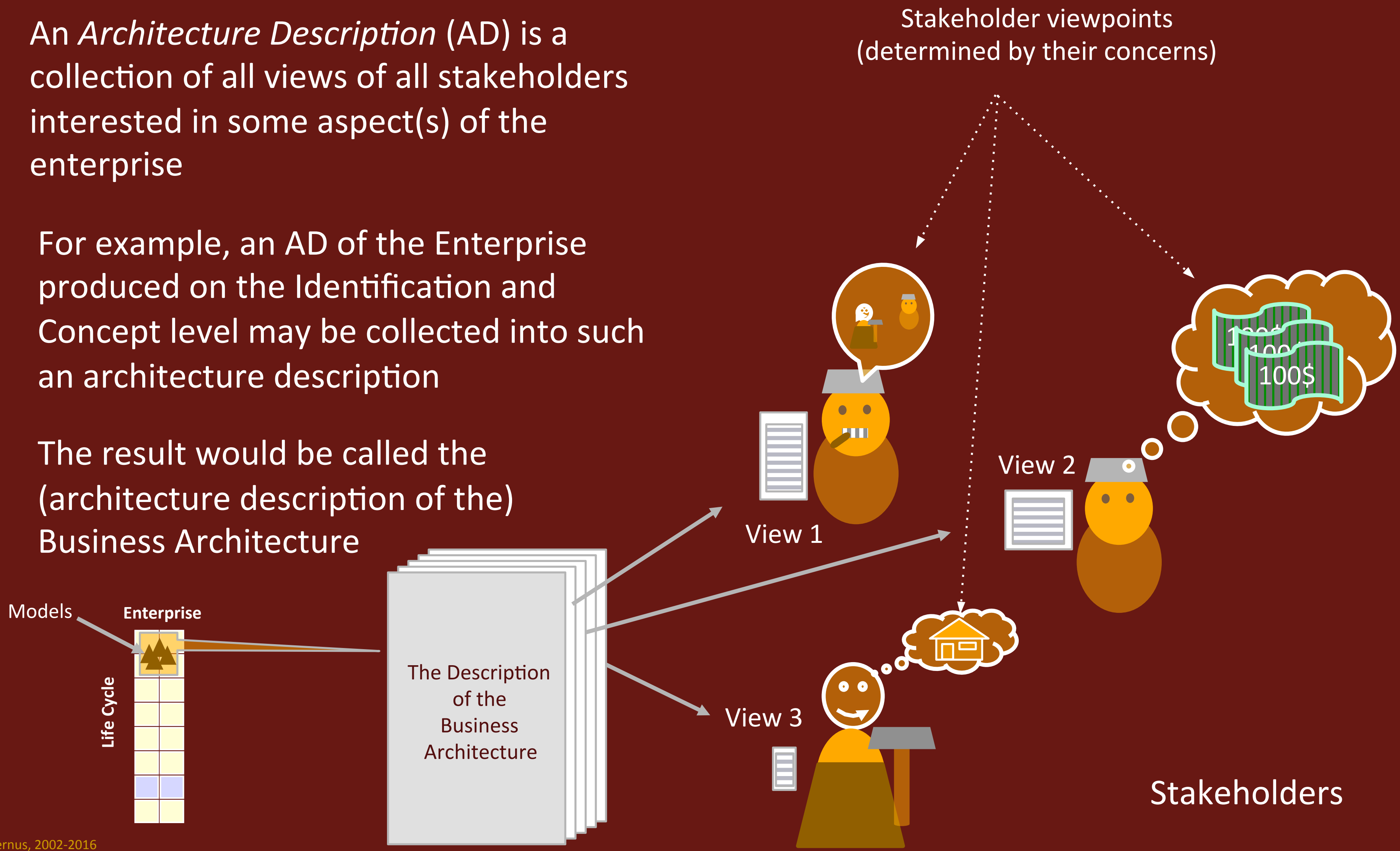
- Enterprise Modelling Tools can help deliver / display such customised views

Example

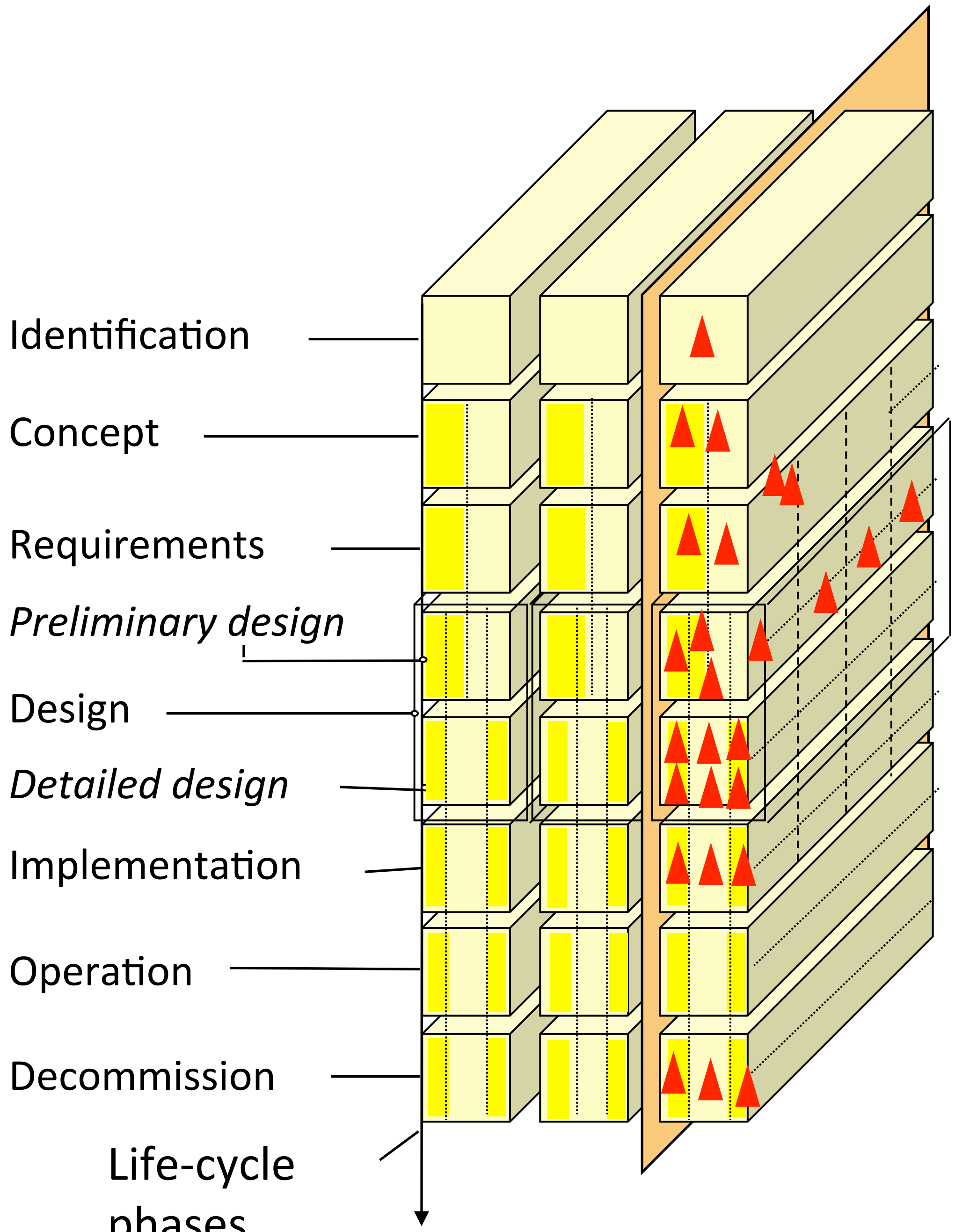
- Views can be packaged into 'Architecture Descriptions'
- An *Architecture Description* (AD) is a collection of all views of all stakeholders interested in some aspect(s) of the enterprise

For example, an AD of the Enterprise produced on the Identification and Concept level may be collected into such an architecture description

The result would be called the (architecture description of the) Business Architecture



Stakeholder viewpoints (determined by their concerns)
Models
Enterprise
Life Cycle
The Description of the Business Architecture
View 1
View 2
View 3
100$
Stakeholders



The modelling framework can be filled with models (generic, partial and particular) for any enterprise entity / entity type

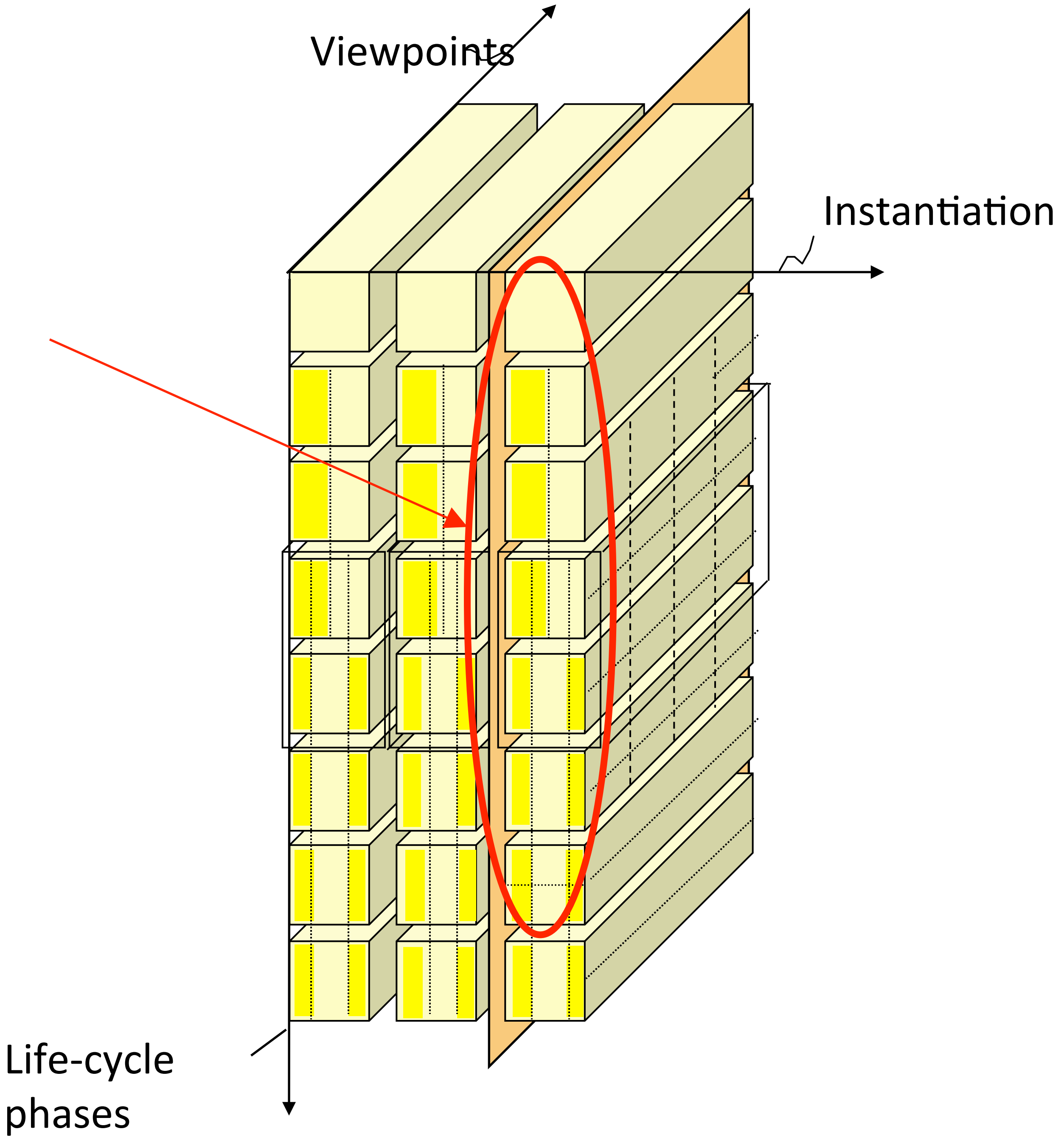
What modelling language to use for which area is *not* pre-defined by the framework

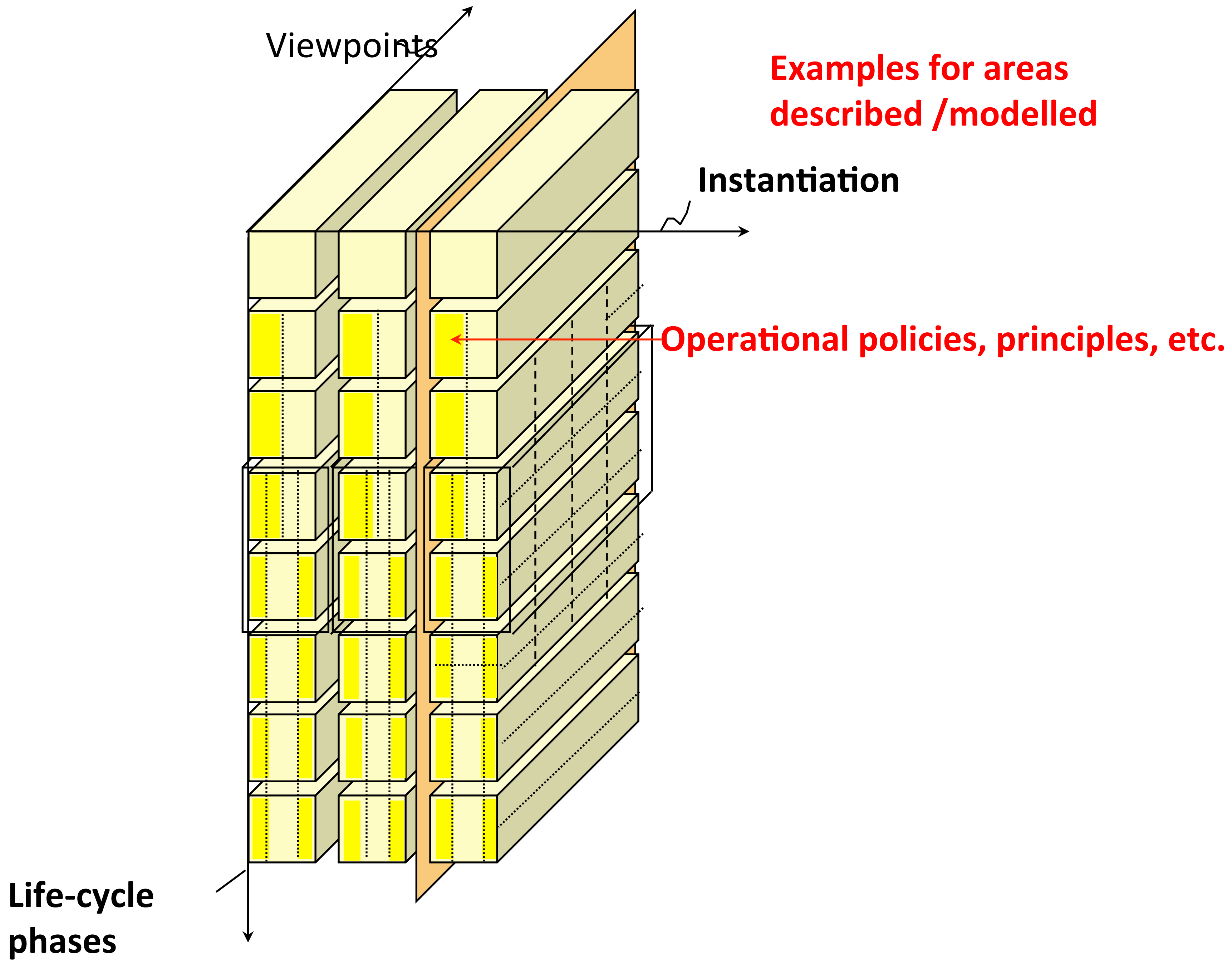
The selection of modelling language depends on the given entity, the types of processes in the entity, stakeholder competencies, the purpose of modelling (what questions need to be answered by the model to answer stakeholder concerns) and the availability of modelling tools

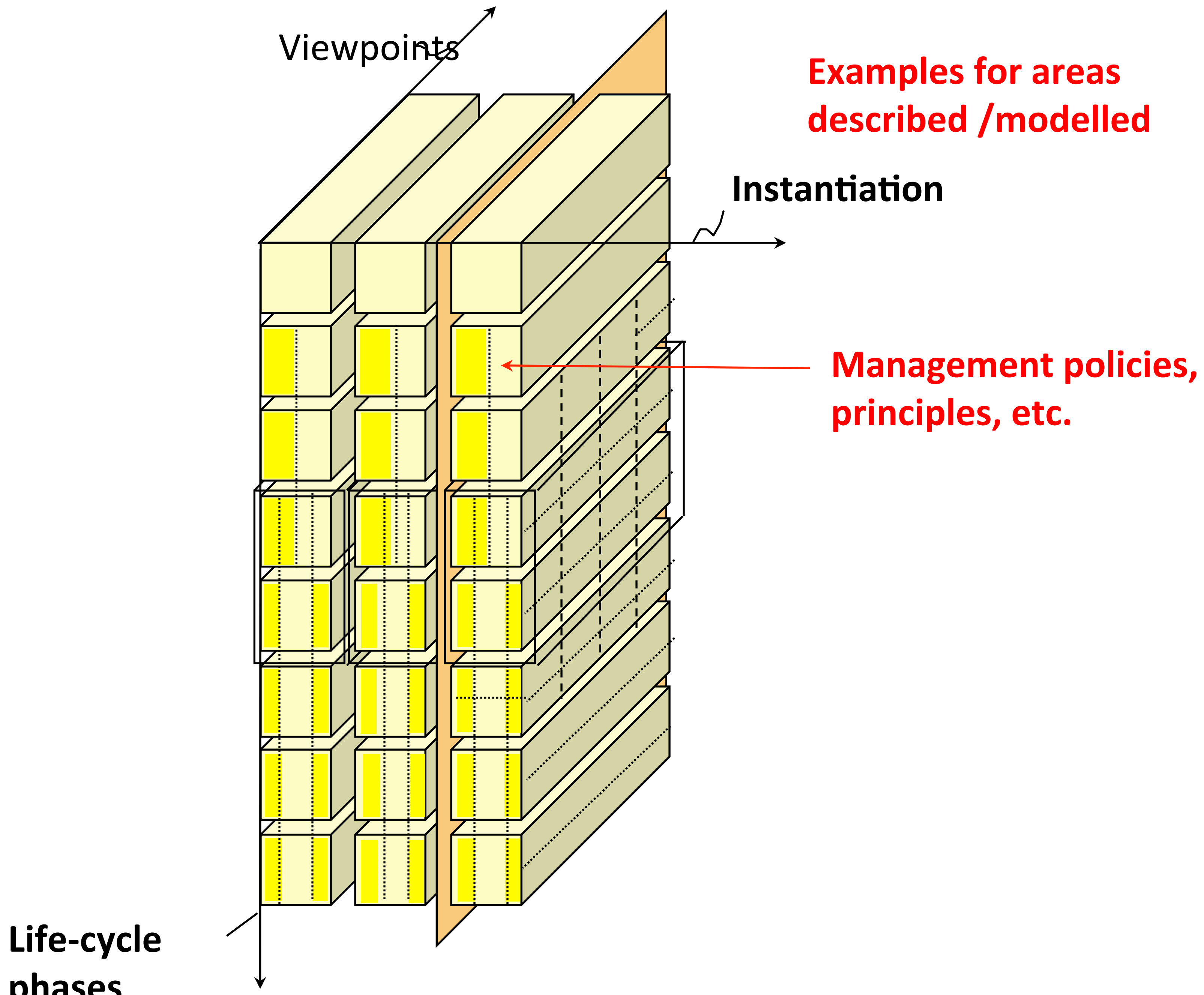
Use of enterprise models

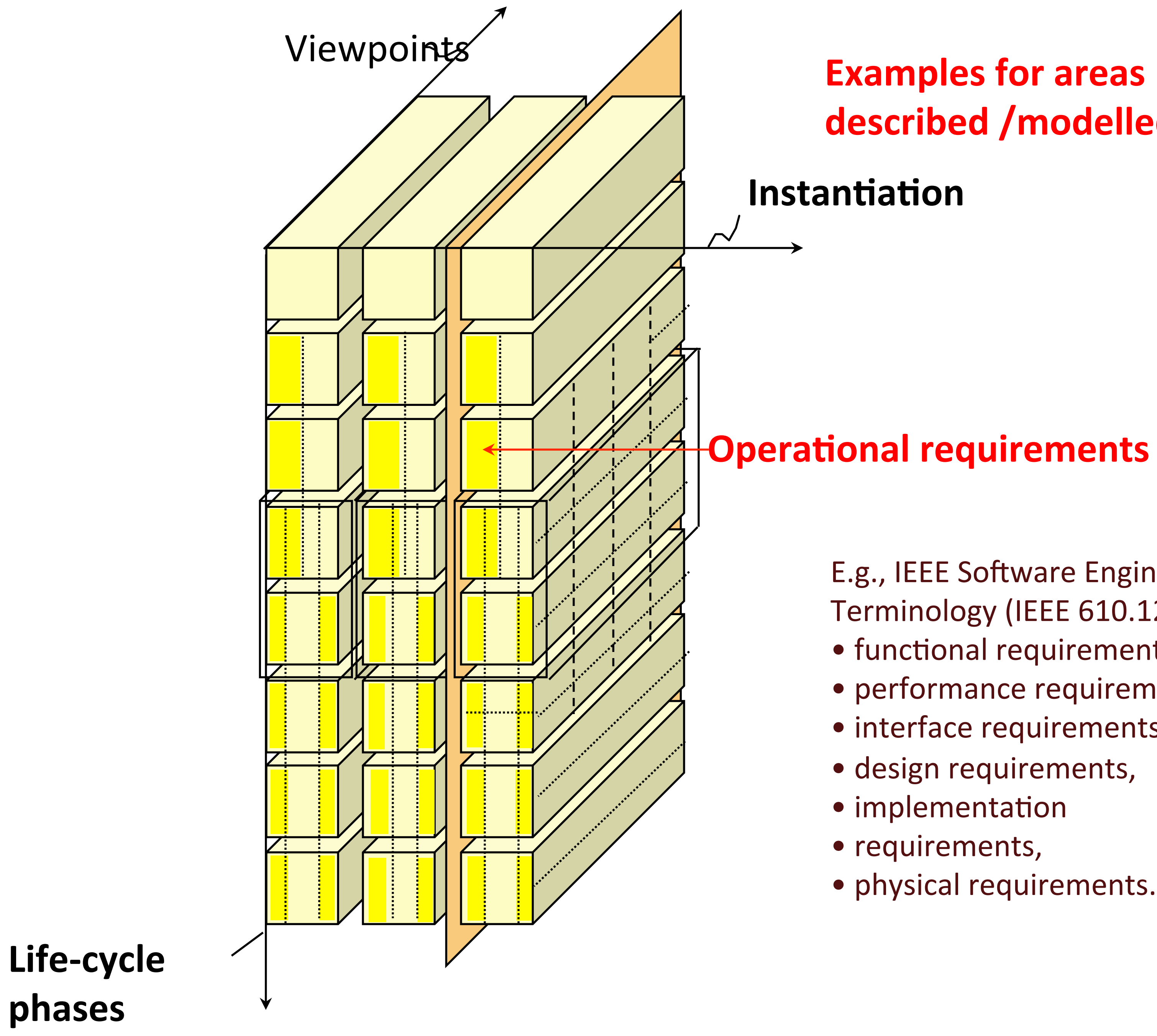
- Support the enterprise engineering process (and in general support decision making about the change)
- Used as an explicit, common reference to support common understanding among groups of people
- Used for design decision making by representing explicit properties of a system or used to calculate / optimise implicit properties of the system (e.g. by simulation or other calculation)
- Model based control: support the execution of business processes (e.g. workflow implementation of some business processes), support decision making during the operation of the entreprise

(Particular) Enterprise Models



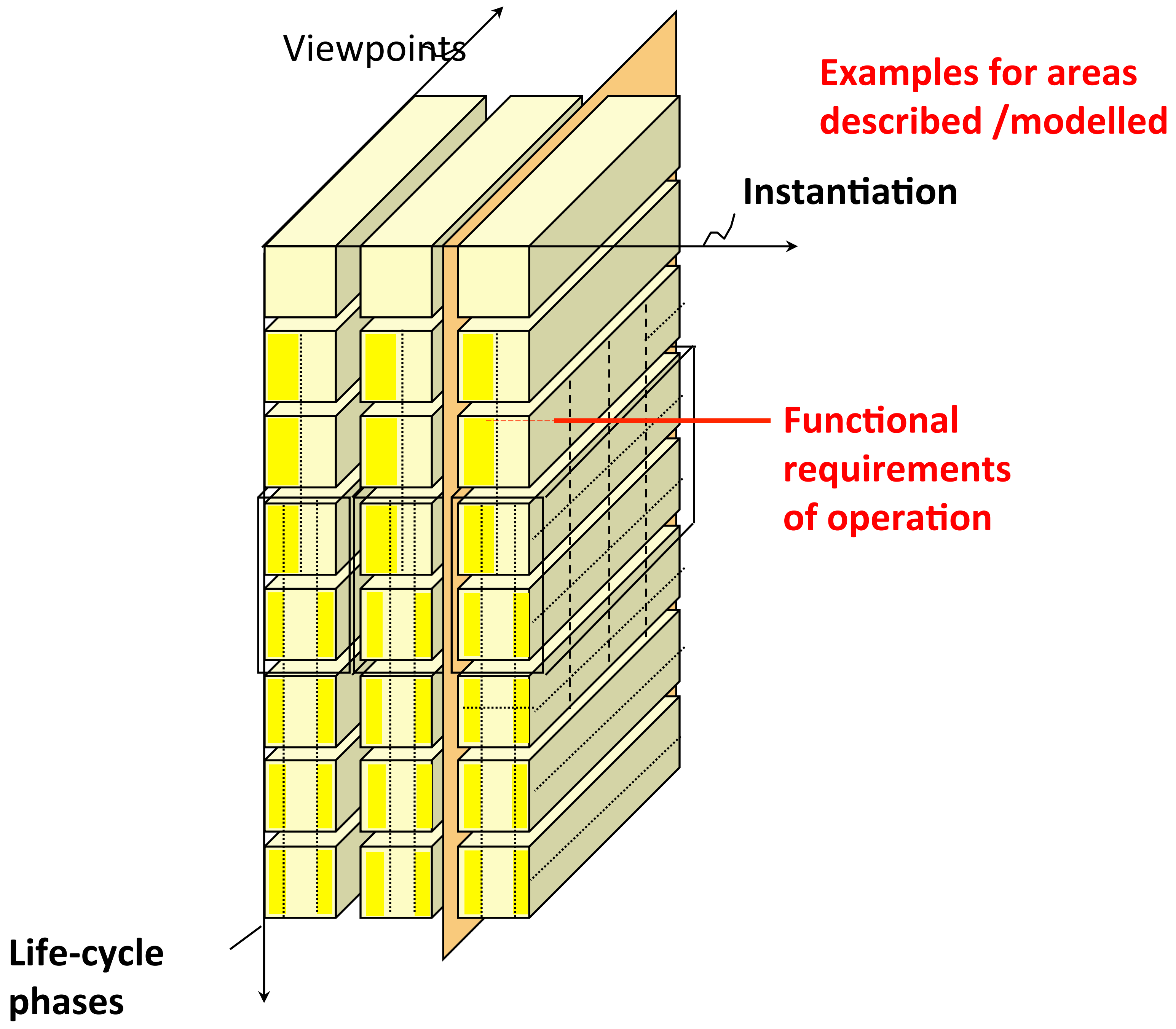

Viewpoints
Examples for areas described /modelled
Instantiation
Operational policies, principles, etc.
Life-cycle phases

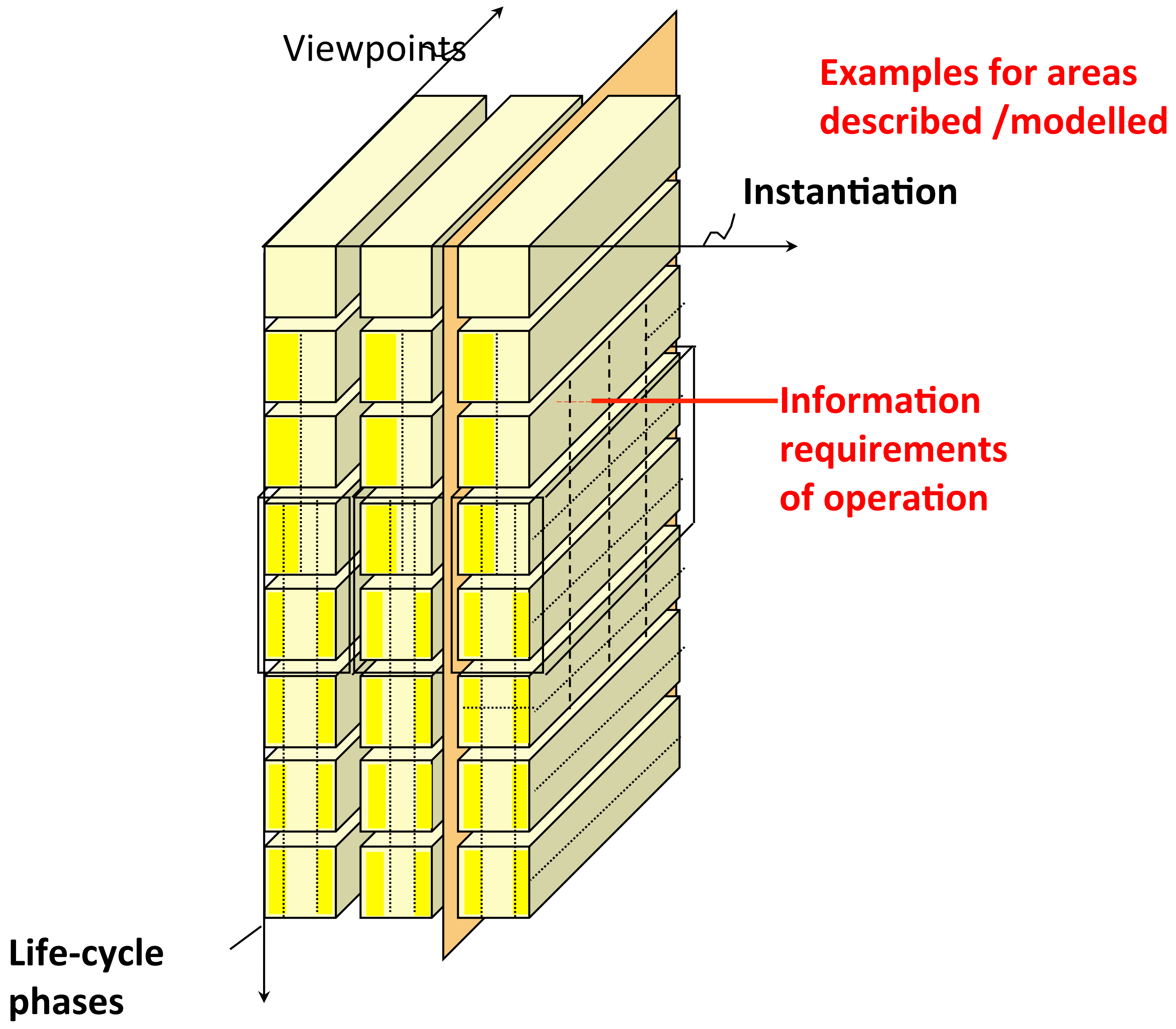

Viewpoints
Examples for areas described /modelled
Instantiation
Management policies, principles, etc.
Life-cycle phases

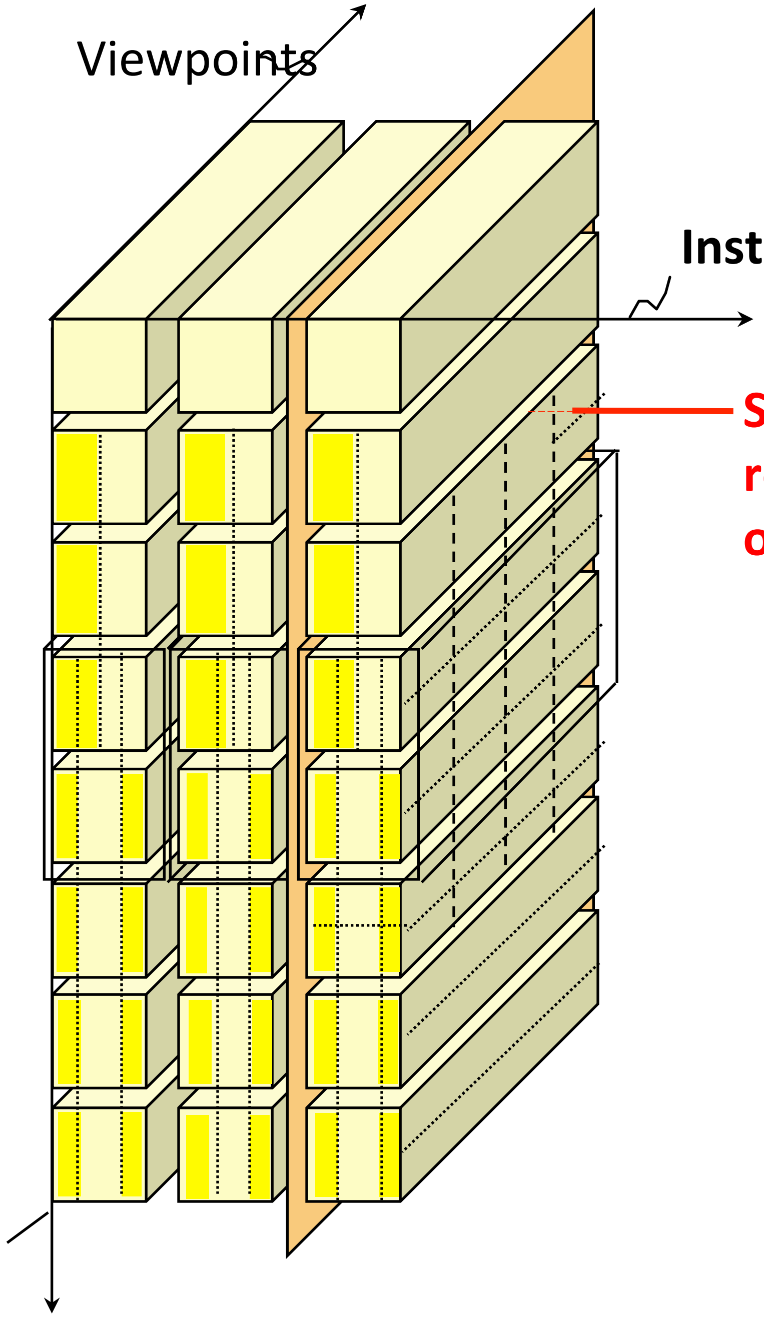


E.g., IEEE Software Engineering Terminology (IEEE 610.12:1990)

- functional requirements,
- performance requirements,
- interface requirements,
- design requirements,
- implementation
- requirements,
- physical requirements.

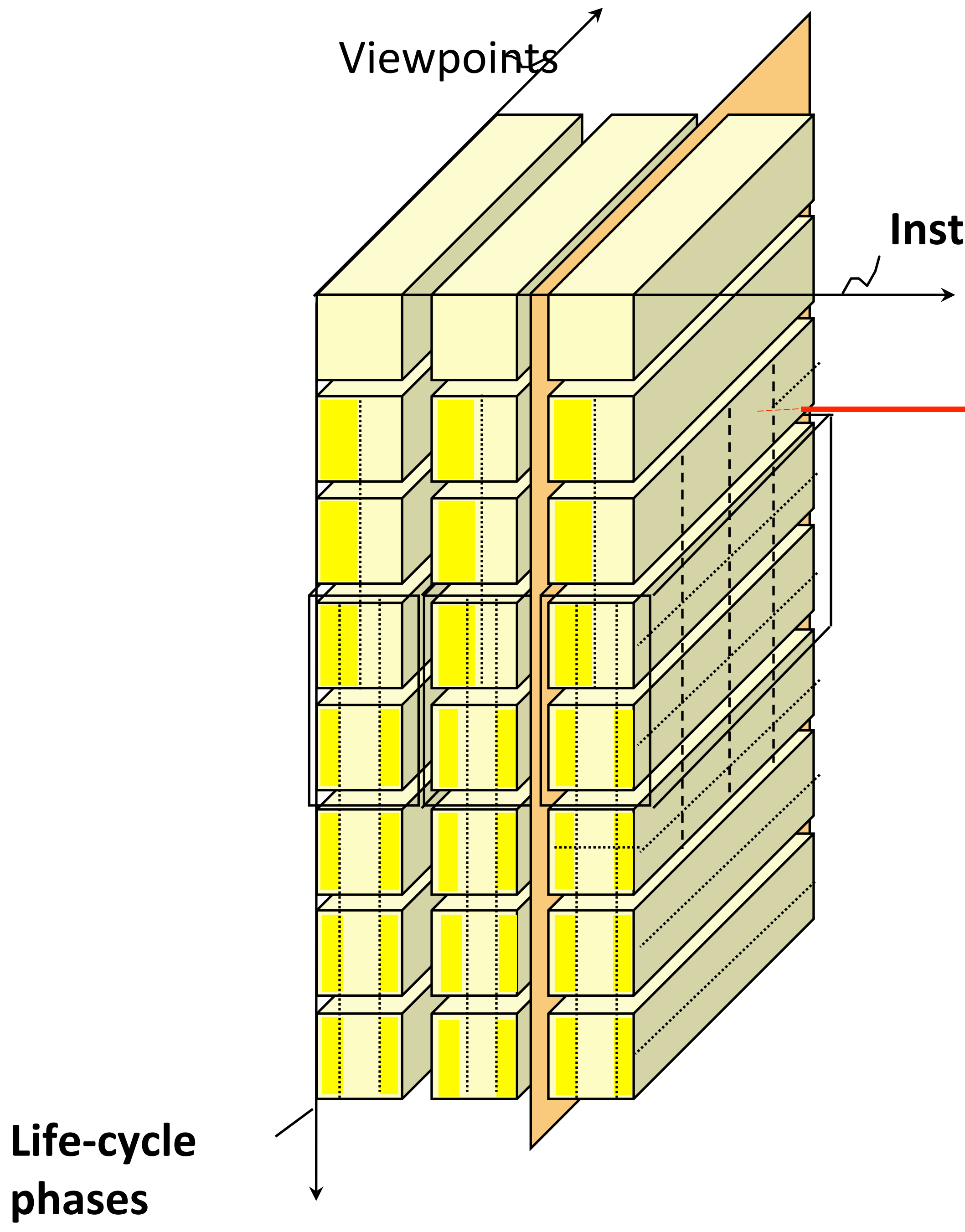

Viewpoints
Examples for areas described /modelled
Instantiation
Functional requirements of operation
Life-cycle phases


Viewpoints
Examples for areas described /modelled
Instantiation
Information requirements of operation
Life-cycle phases

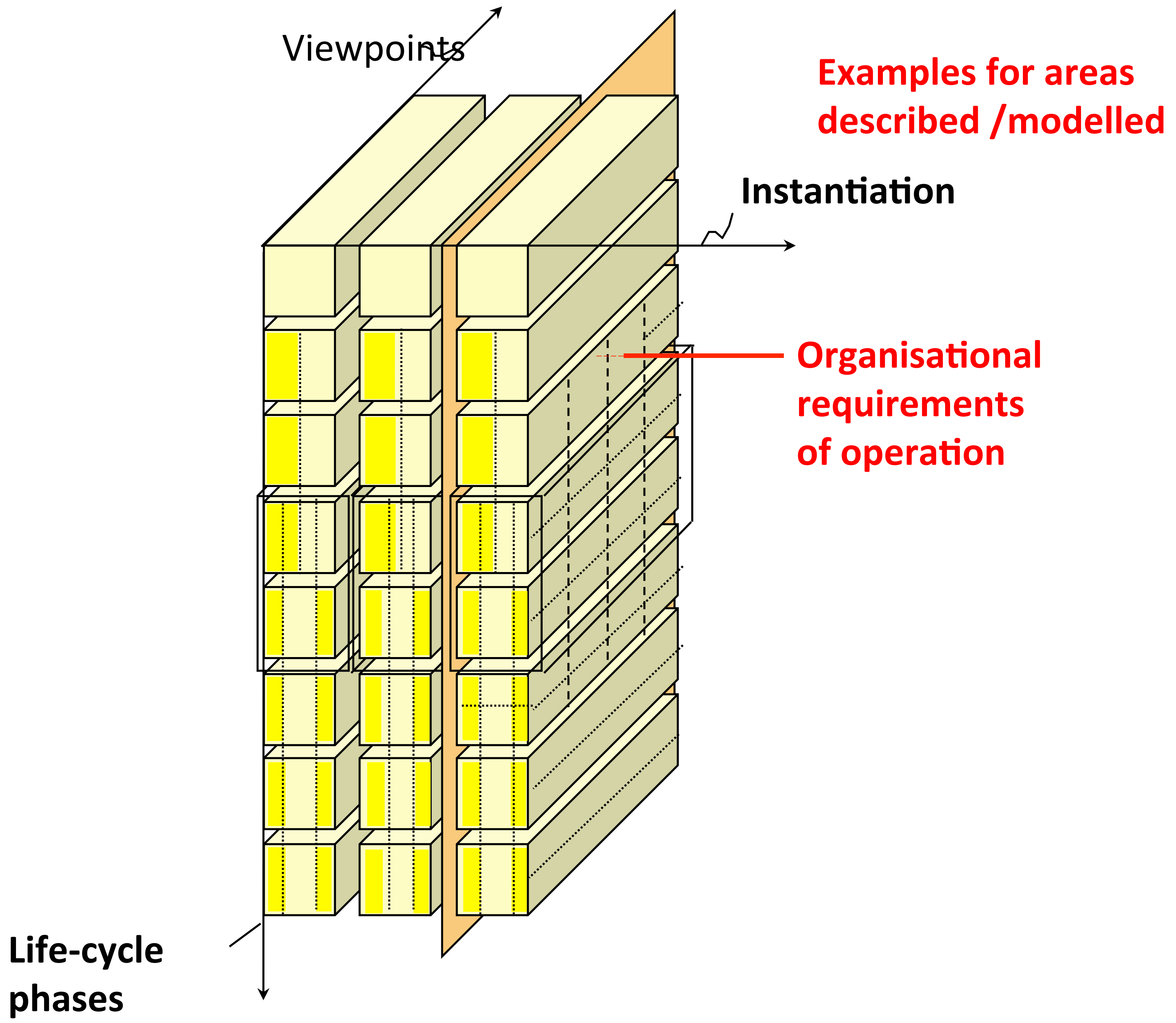


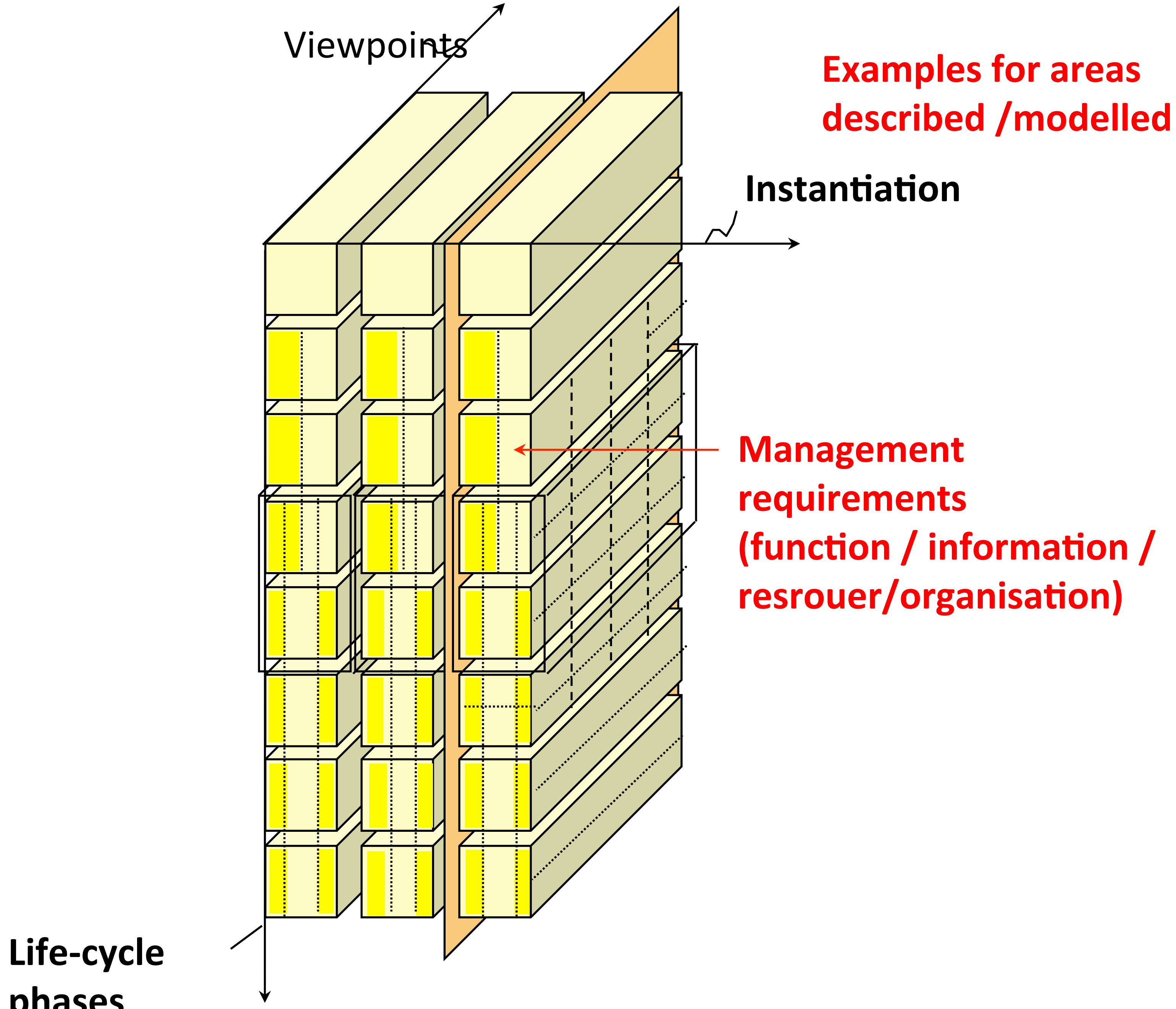
Examples for areas described /modelled

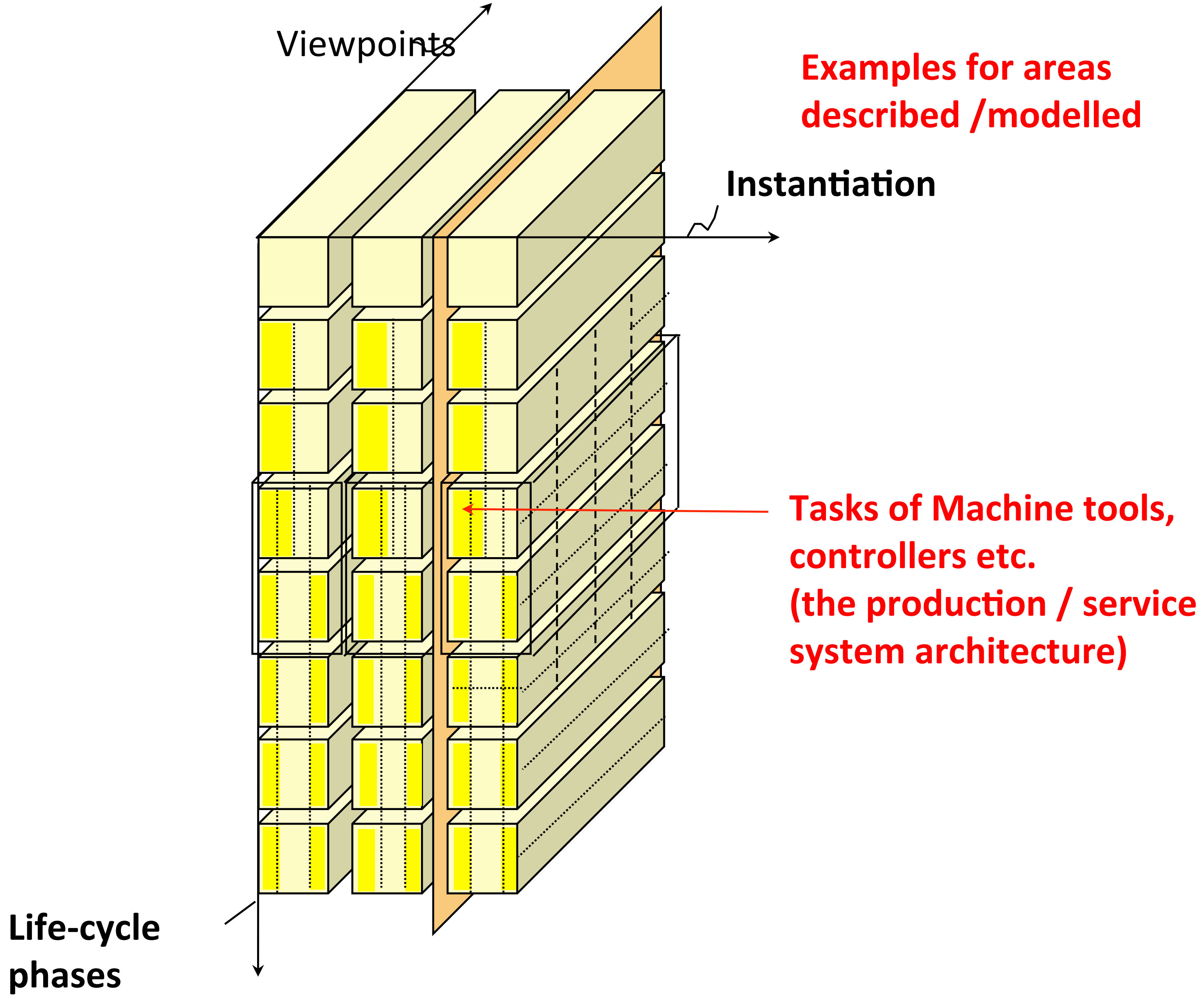
Sw resource requirements of operation

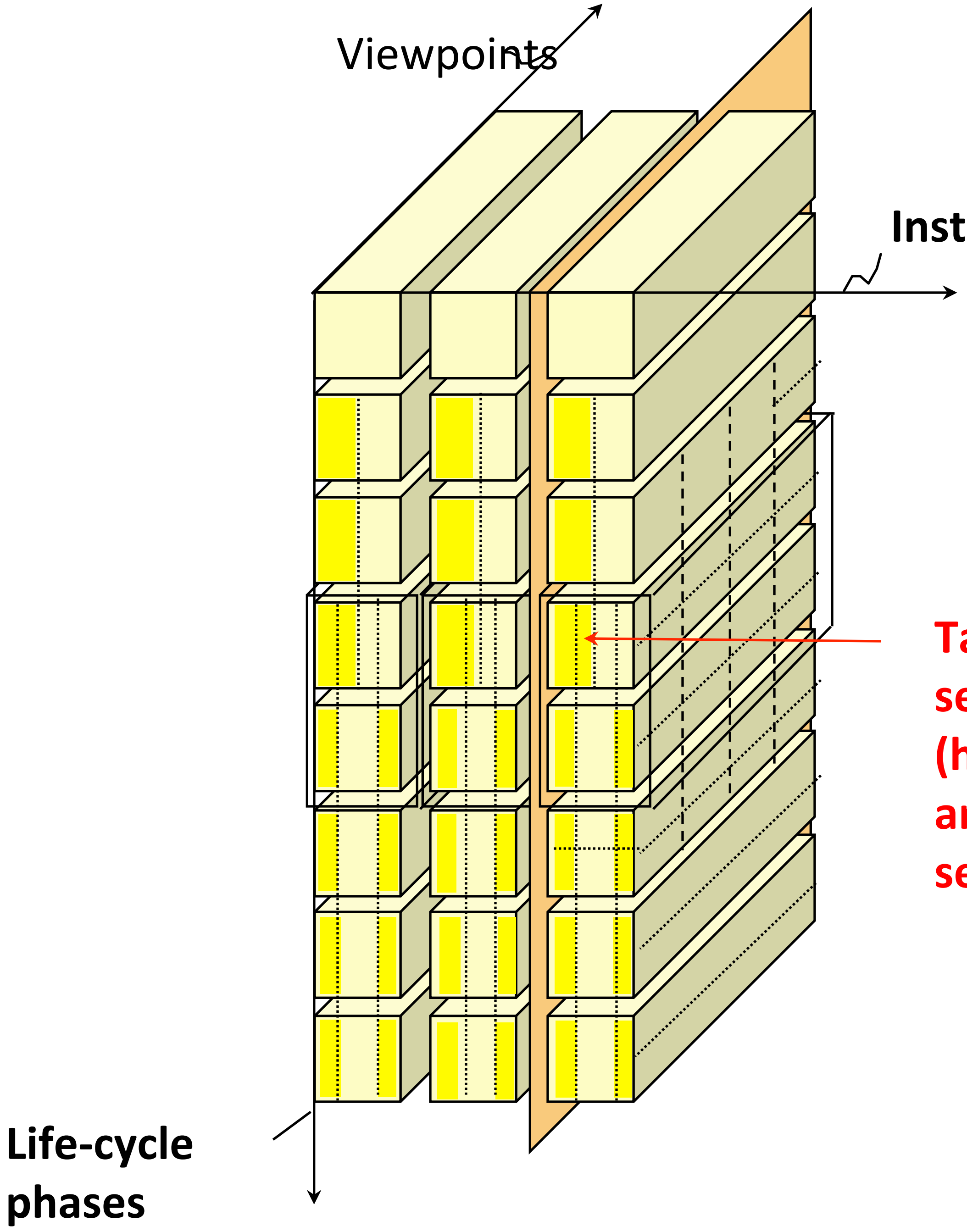


Examples for areas described /modelled


Viewpoints
Examples for areas described /modelled
Instantiation
Organisational requirements of operation
Life-cycle phases

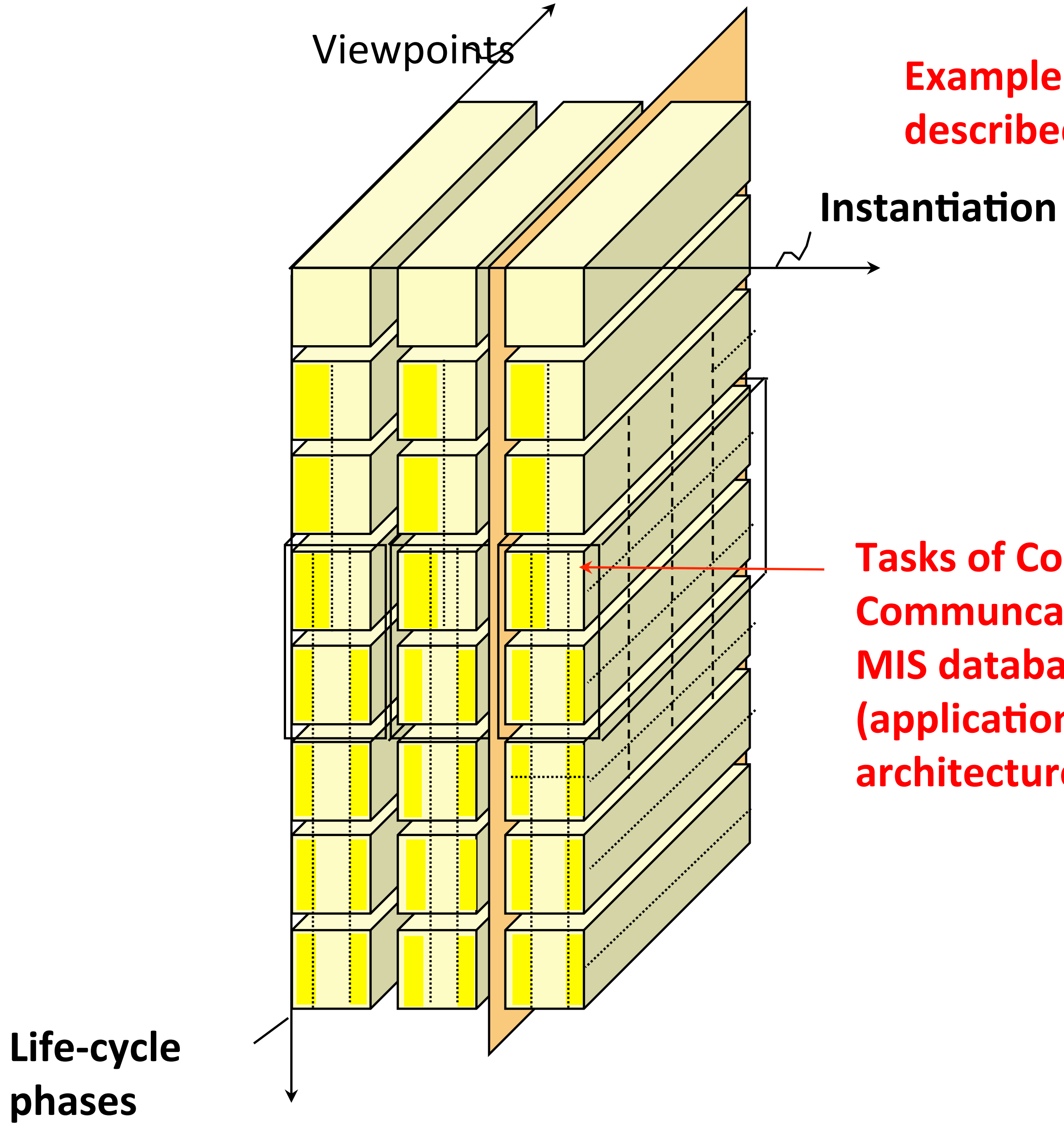

Viewpoints
Examples for areas described /modelled
Instantiation
Management requirements (function / information / resrouer/organisation)
Life-cycle phases


Viewpoints
Examples for areas described /modelled
Instantiation
Tasks of Machine tools, controllers etc. (the production / service system architecture)
Life-cycle phases



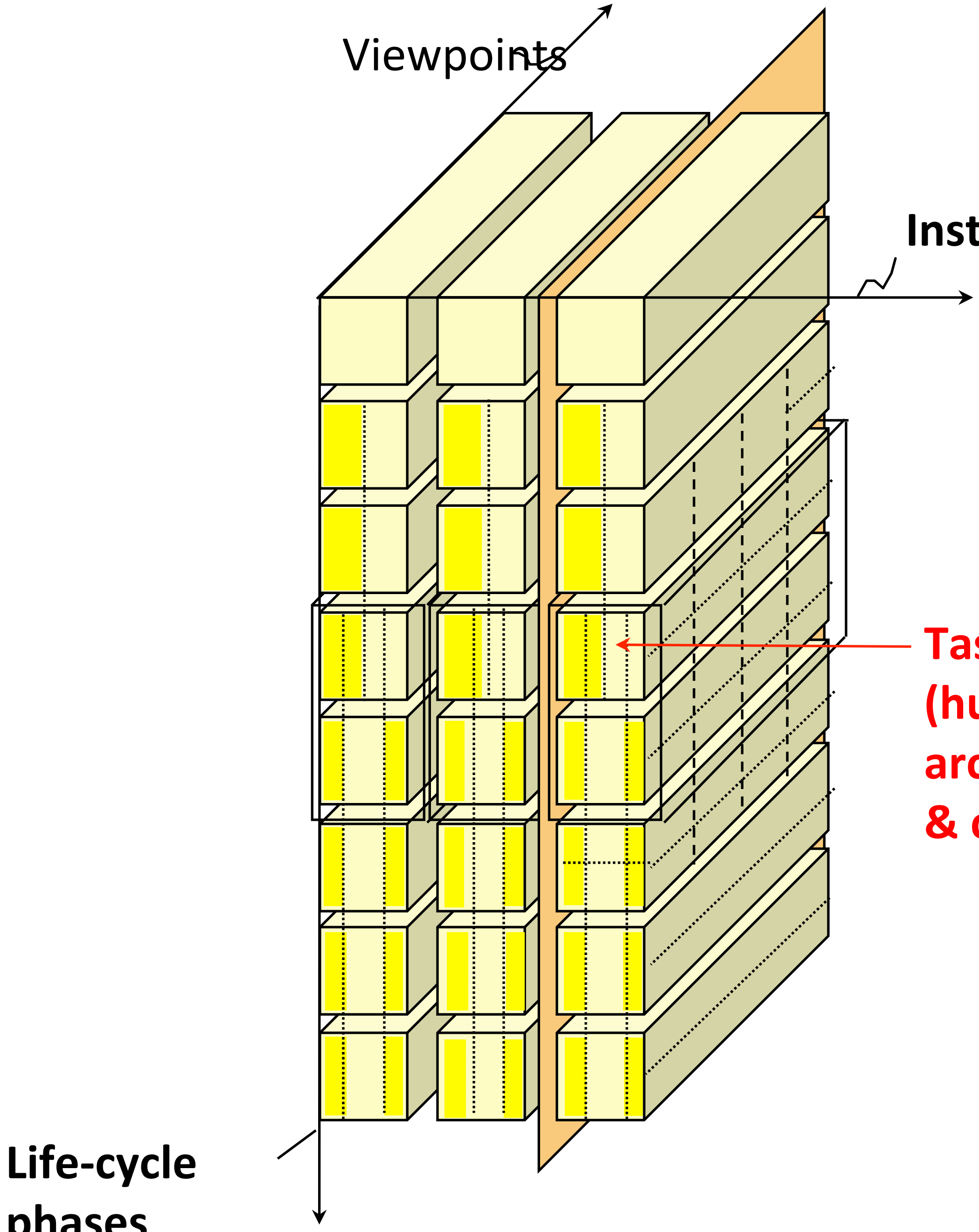
Examples for areas described /modelled

Tasks of shop floor workers / service delivery personnel (human-organisational architecture of the shop floor / service area)



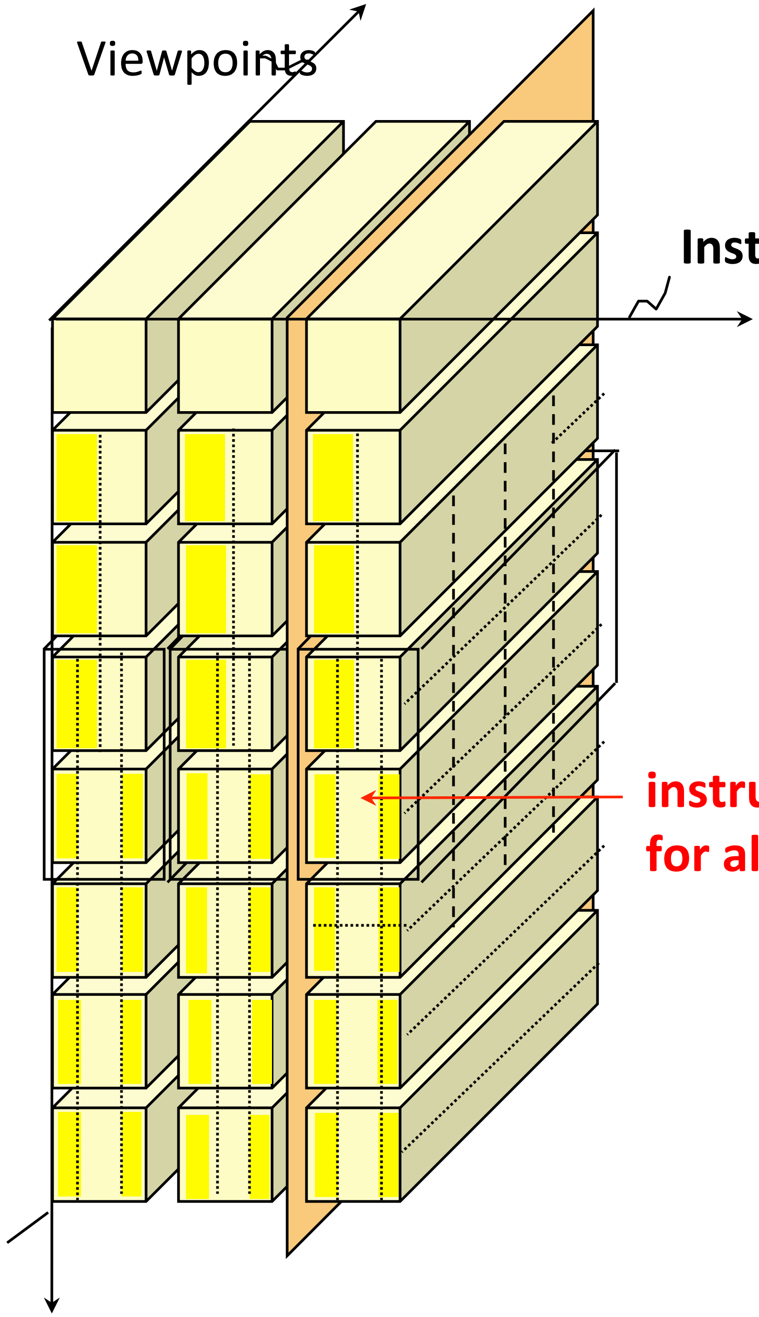
Examples for areas described /modelled

Tasks of Control & Communcation systems, MIS database, DSS, ERP, etc. (application- and information architecture)



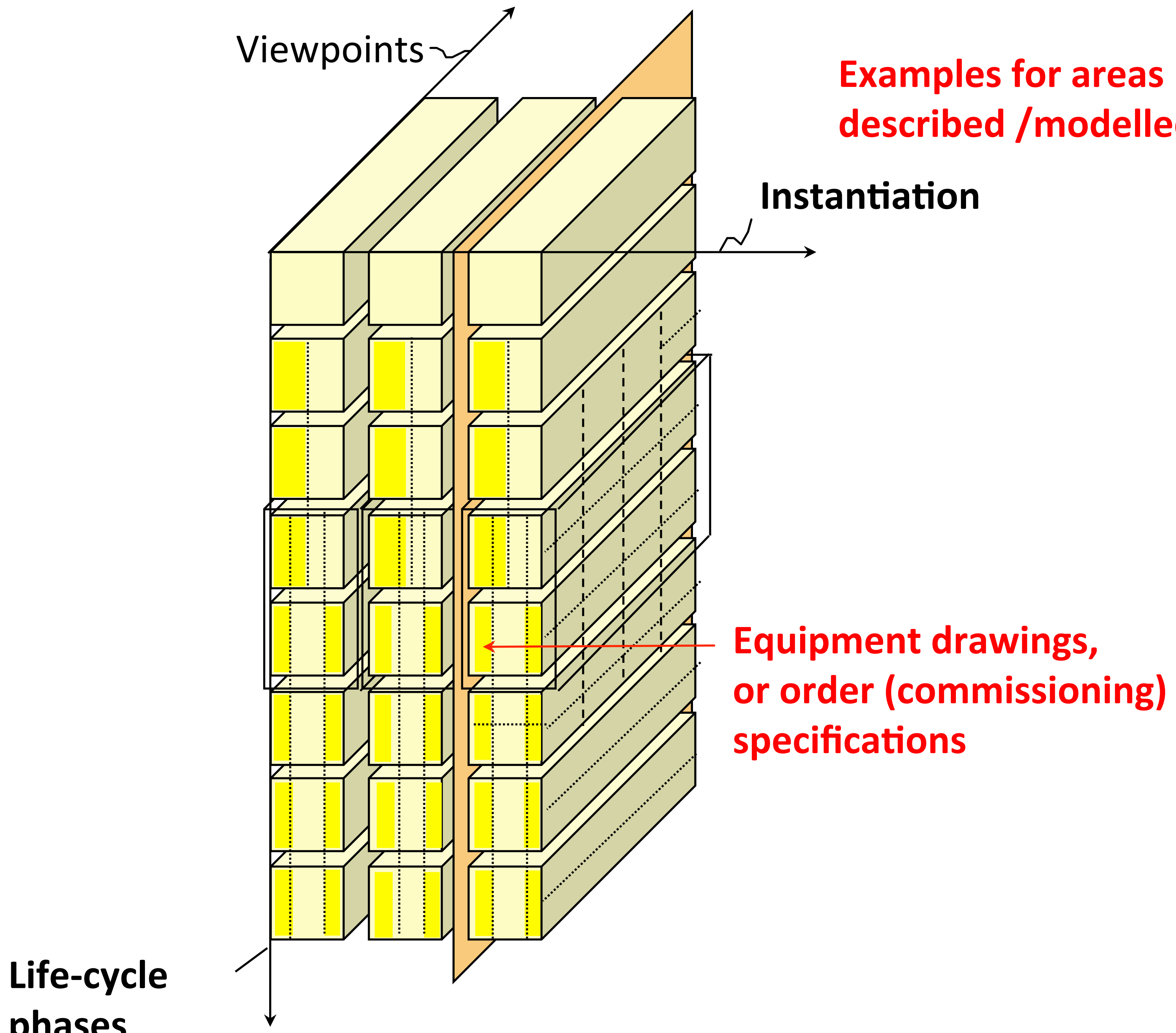
Examples for areas described /modelled

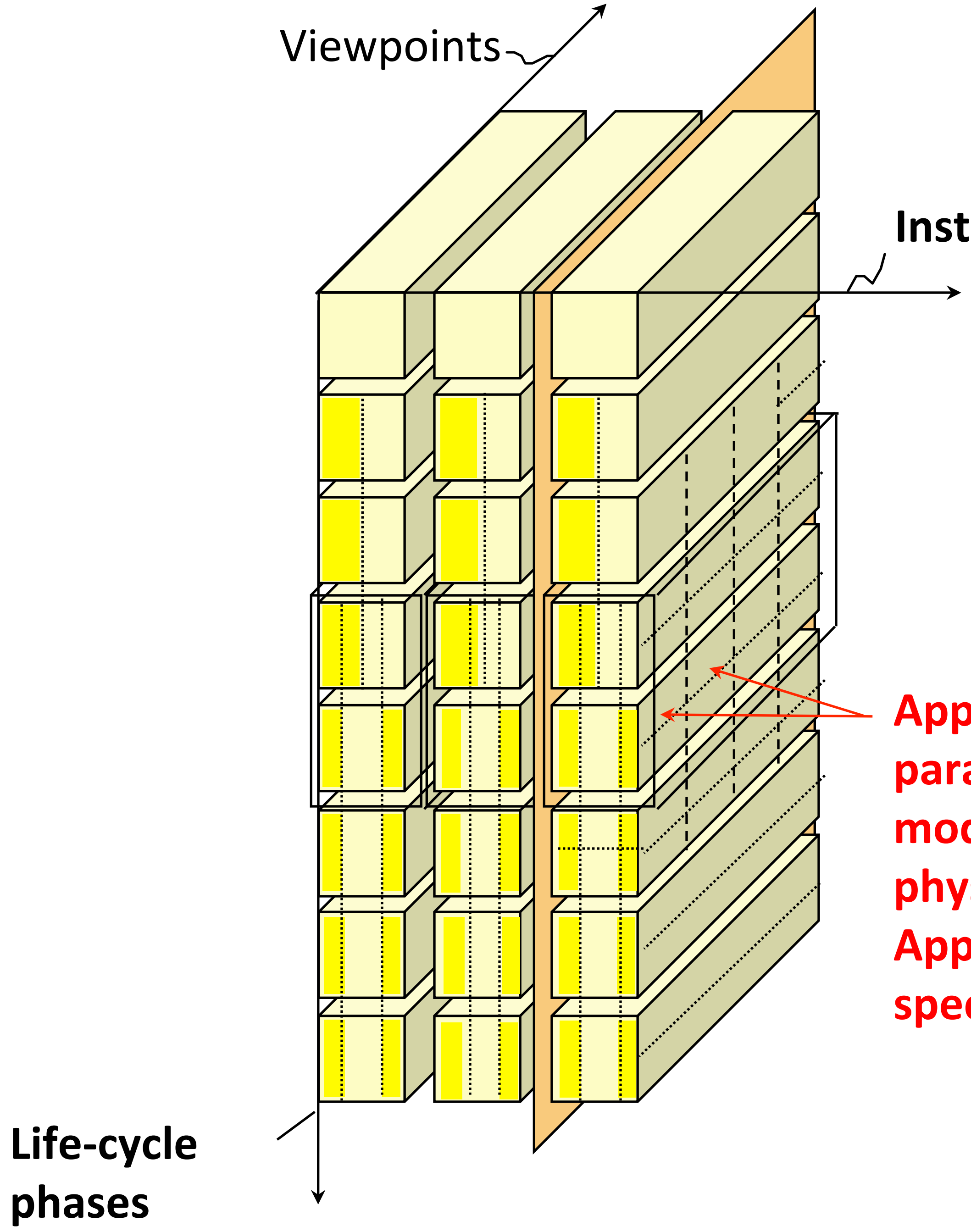
Tasks of Management personnel (human-organisational architecture of the management & control system)



Examples for areas described /modelled

instructions, role descriptions for all personnel

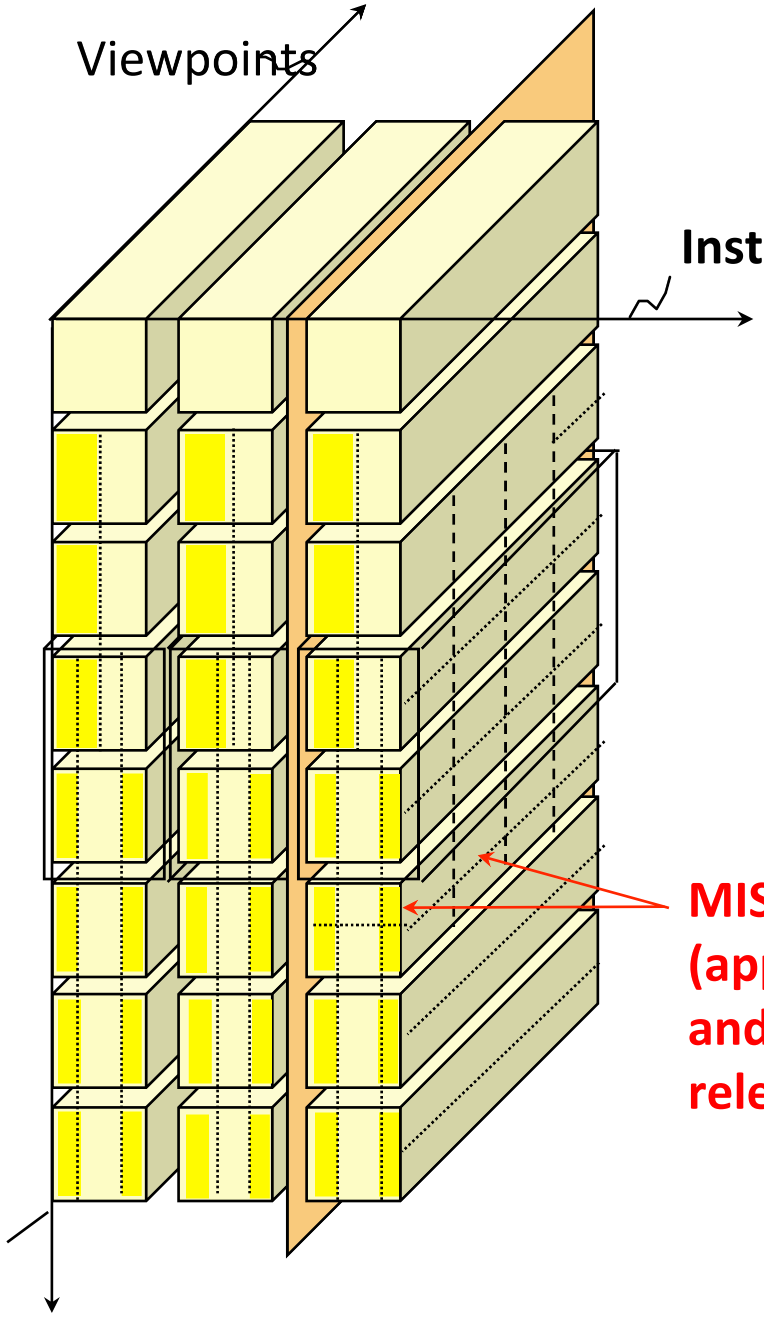

Viewpoints
Examples for areas described /modelled
Instantiation
Equipment drawings, or order (commissioning) specifications
Life-cycle phases



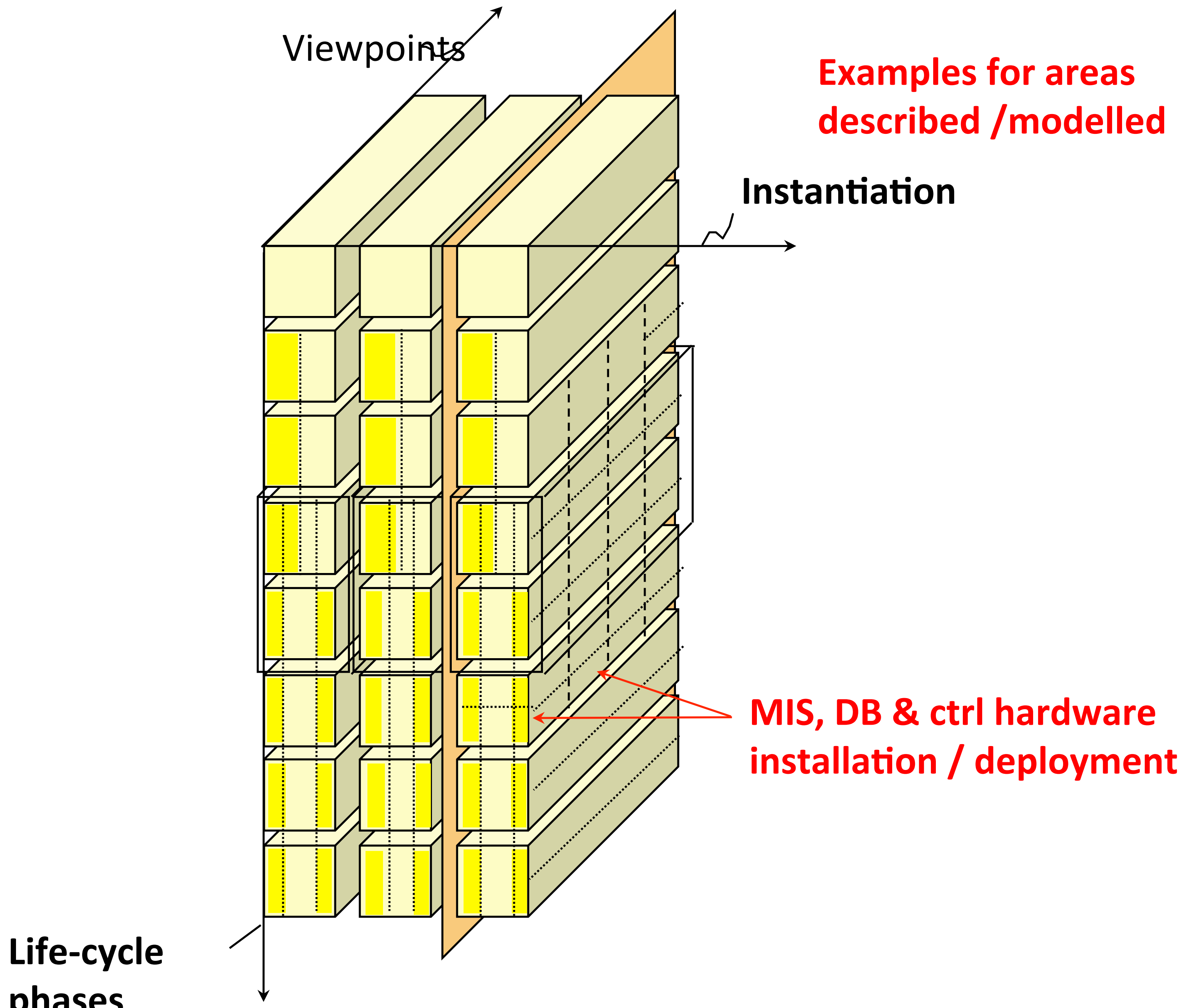
Examples for areas described /modelled

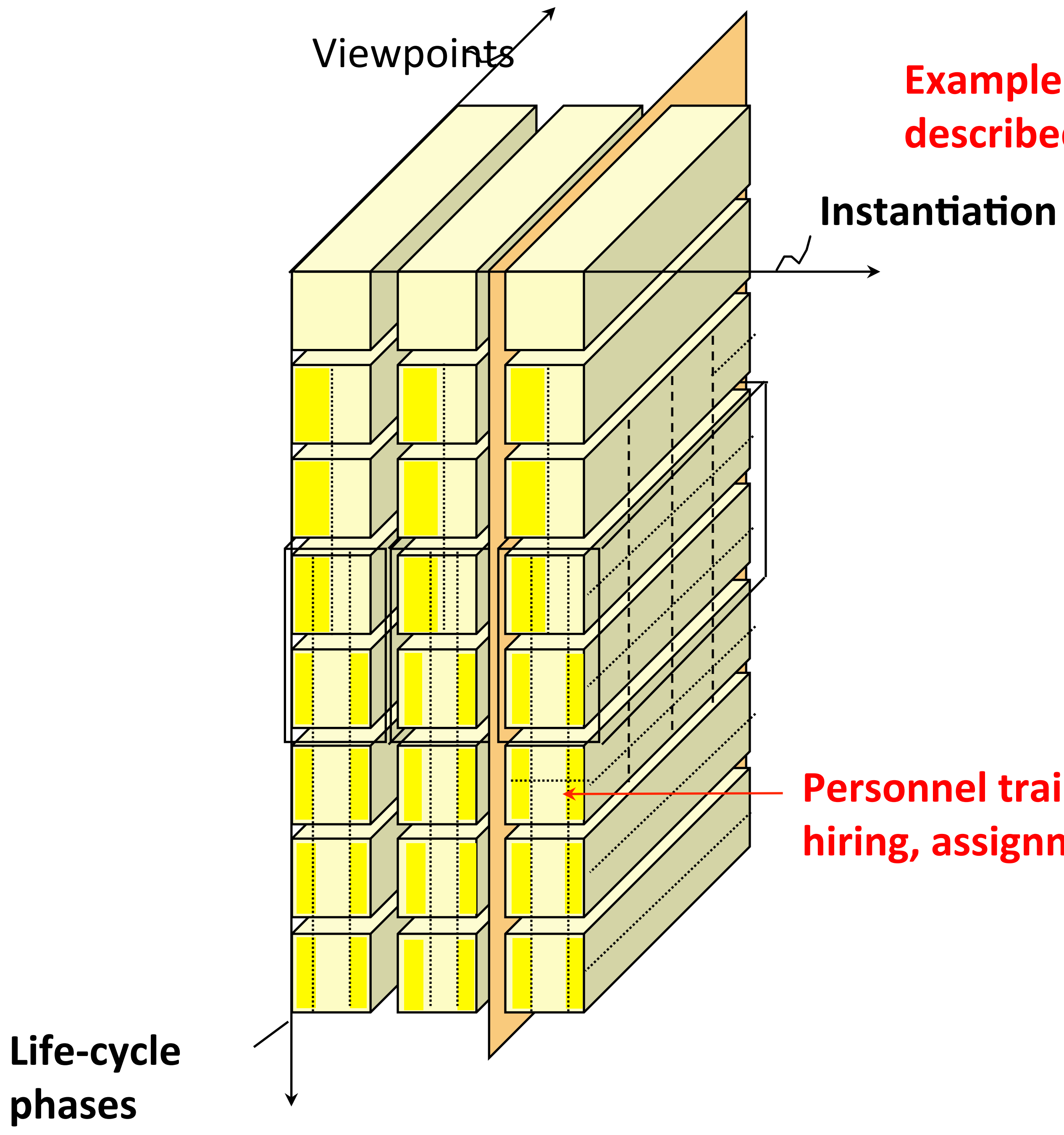
Application coding / parametric configuration of ERP modules, Database coding (SQL, physical schema design) and/or Application / Database product specifications

Examples for areas described /modelled



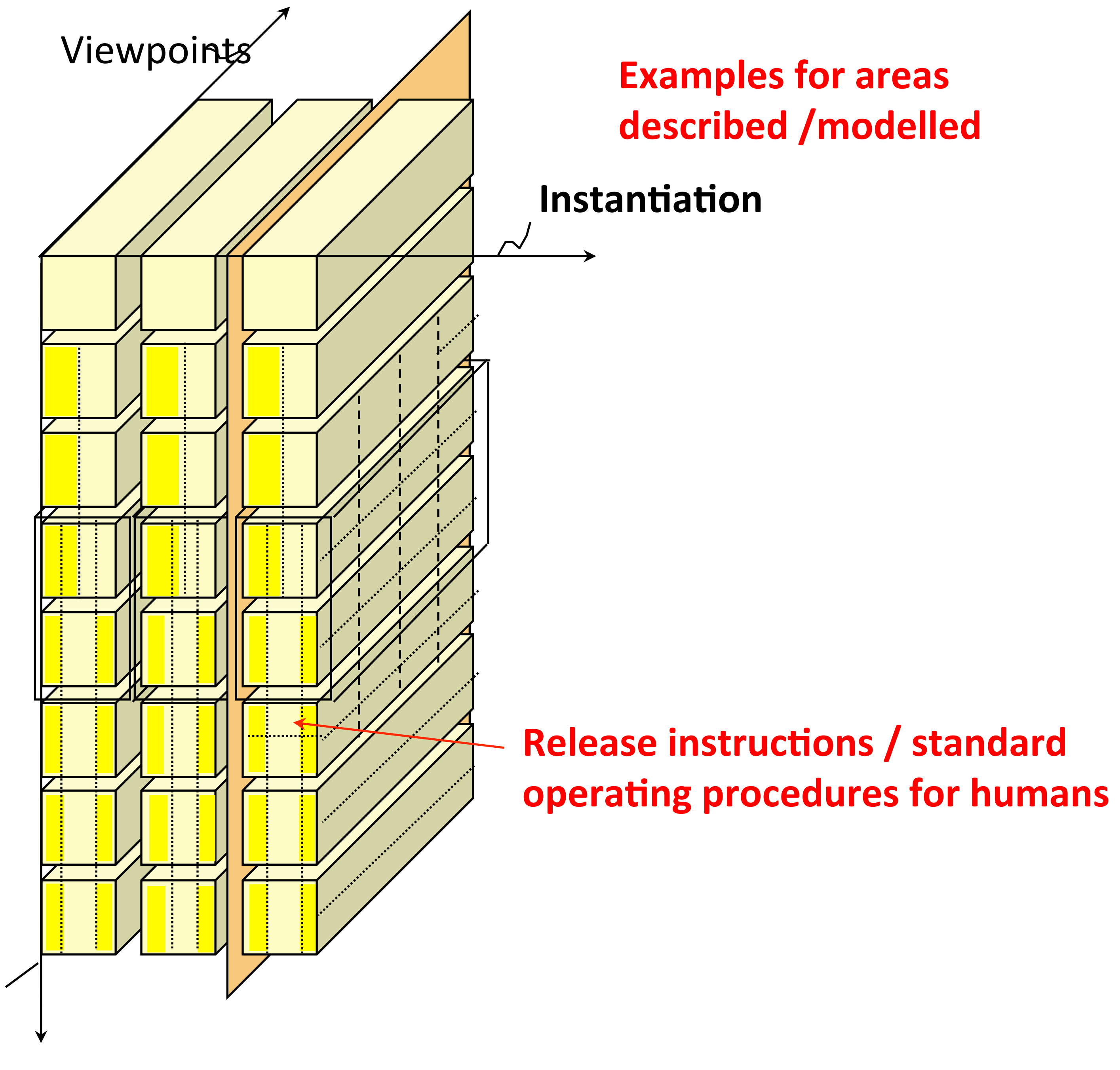
MIS & ctrl software (applications and database deployment & release to operation)

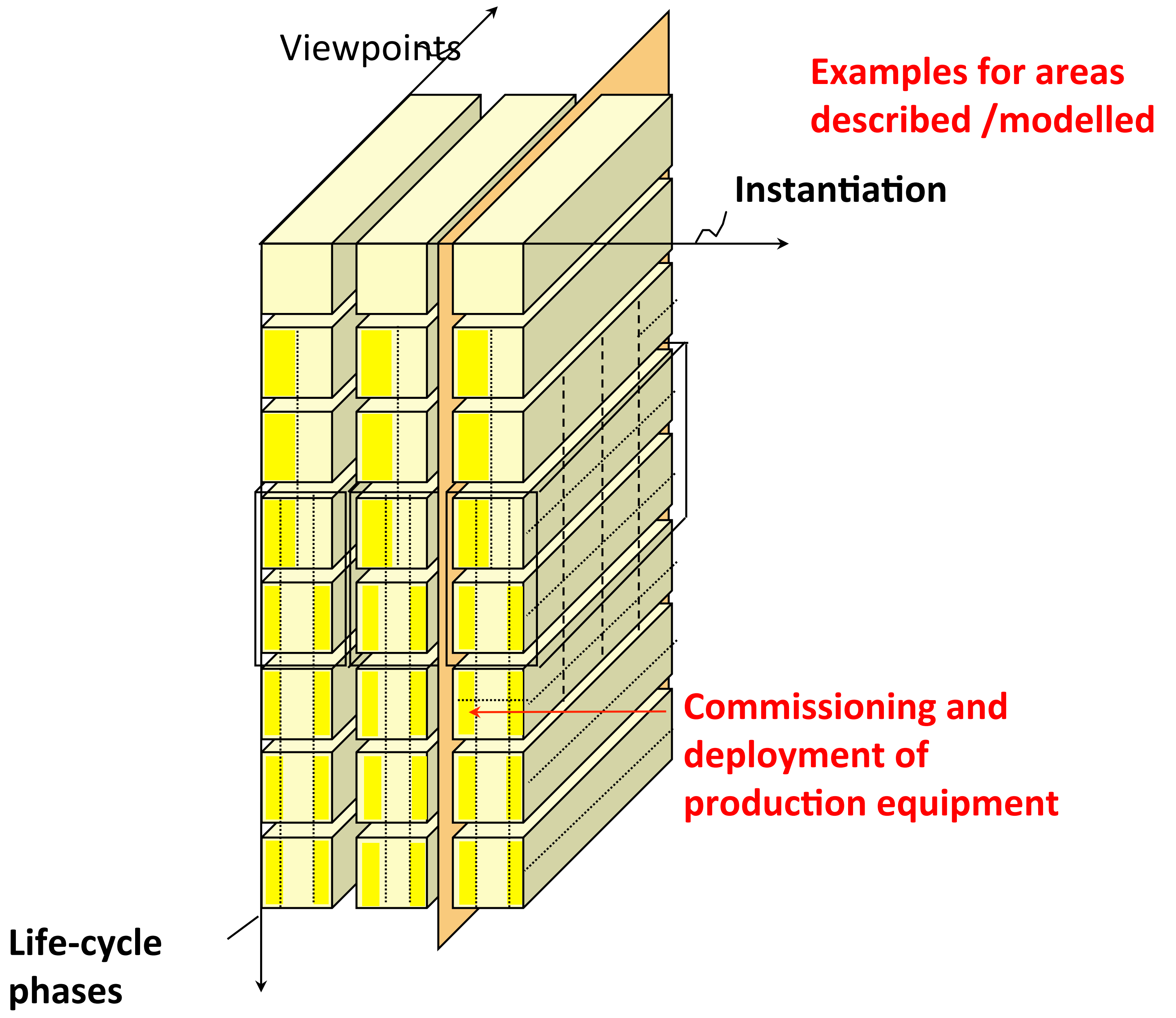

Viewpoints
Examples for areas described /modelled
Instantiation
MIS, DB & ctrl hardware installation / deployment
Life-cycle phases

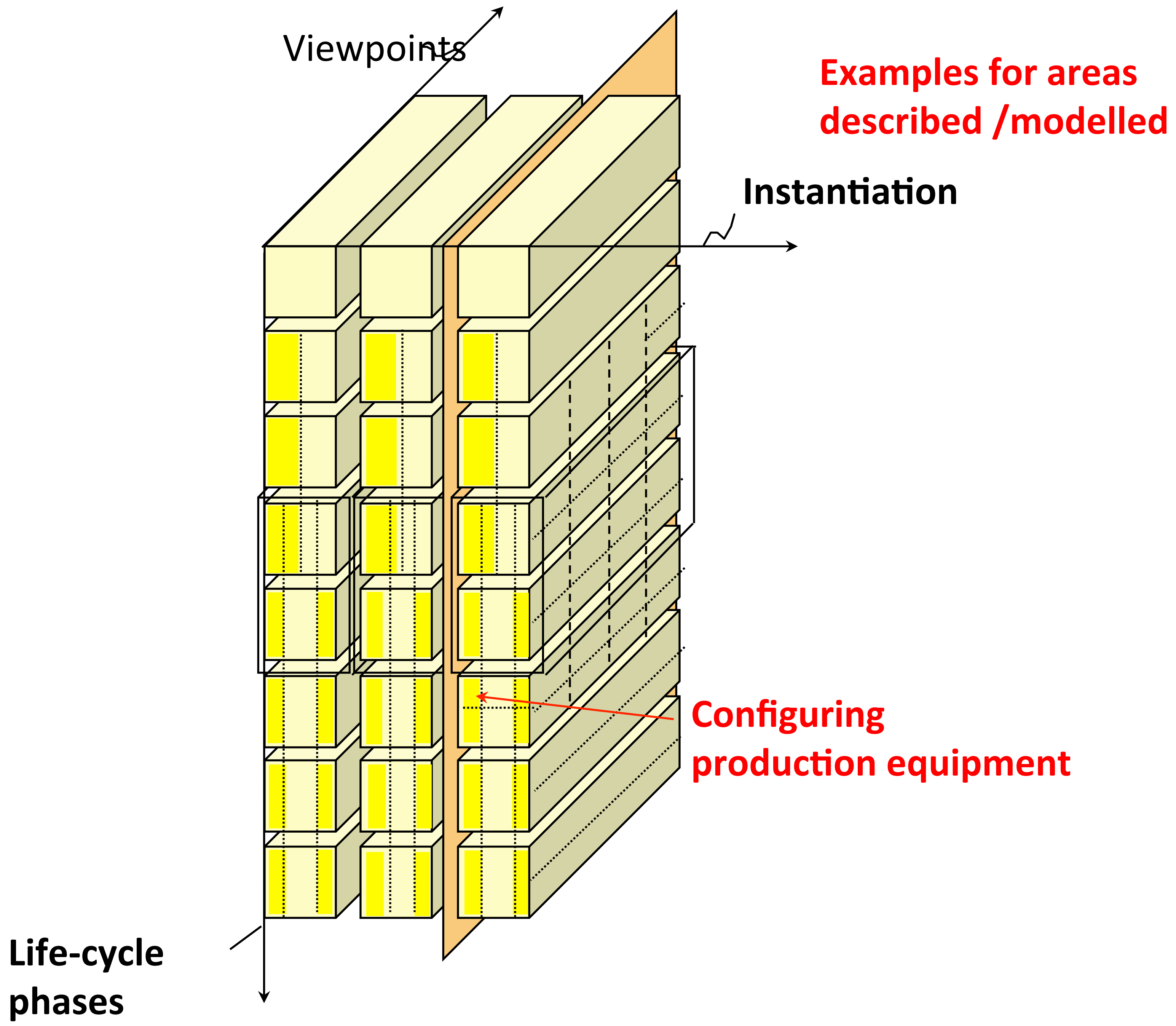


Examples for areas described /modelled

Personnel training, hiring, assignment to roles


Viewpoints
Examples for areas described /modelled
Instantiation
Release instructions / standard operating procedures for humans
Life-cycle phases


Viewpoints
Examples for areas described /modelled
Instantiation
Commissioning and deployment of production equipment
Life-cycle phases

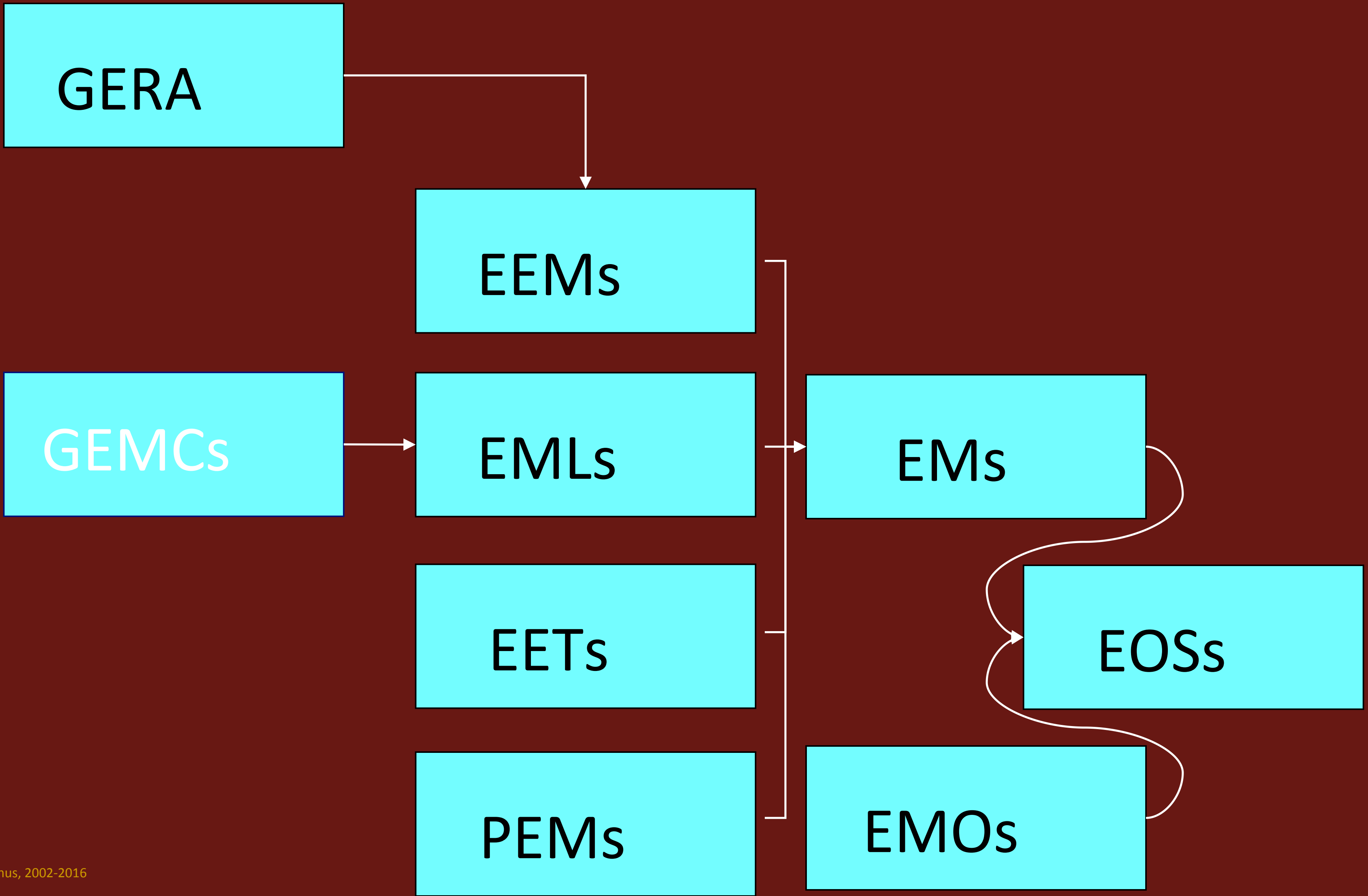

Viewpoints
Examples for areas described /modelled
Instantiation
Configuring production equipment
Life-cycle phases

GEMCs

Generic Enterprise Modelling Concepts

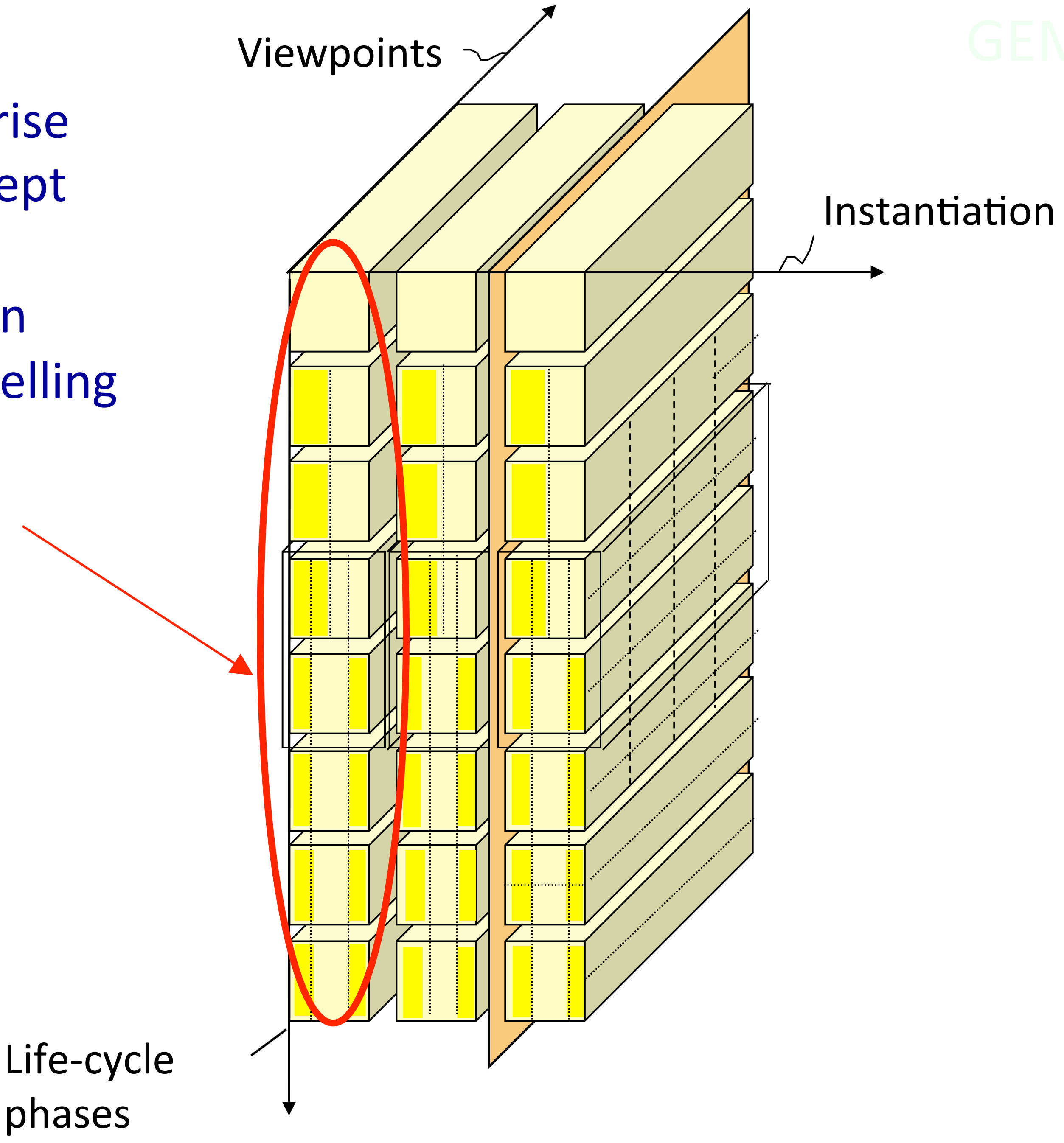
ISO 15704: GEMCs define the meaning of enterprise modelling languages (EMLs) (define the meaning of ‘modelling constructs’)

[Note: in ISO 42010: EMLs are called Architecture Description Languages (ADLs)]


GERA
EEMs
GEMCs
EMLs
EMs
EETs
EOSs
PEMs
EMOs

GEMCs

Generic enterprise modelling concept definitions (the concepts used in enterprise modelling languages)



GEMCs

For end users

less formal

- *Glossary and examples*
- *Metaschema*
- *Ontological theories*

more formal

For tool developers

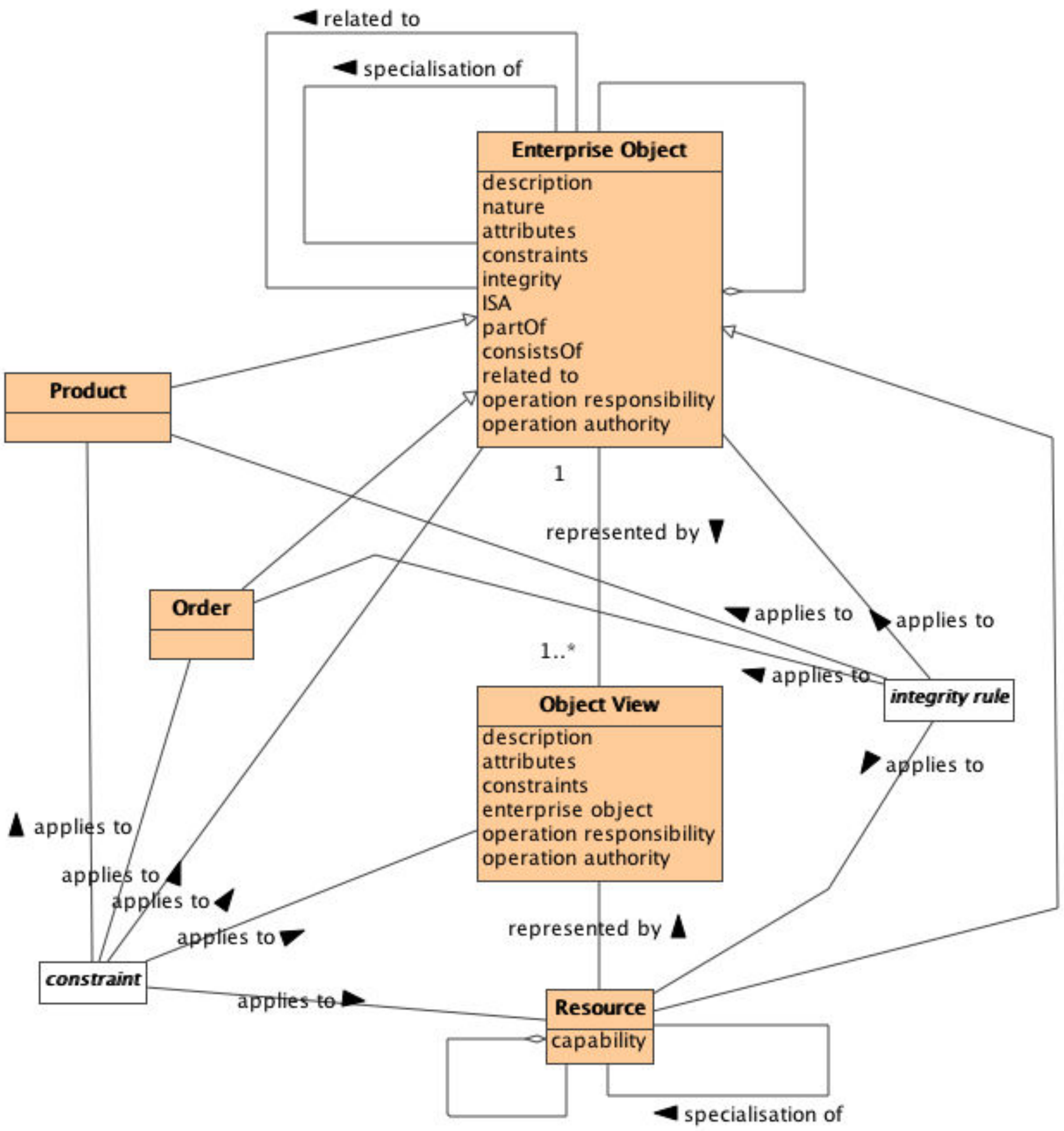
The meaning of language constructs
may take various forms at different levels
of formality

In the standards

- The Object Management Group releases / maintains many modelling languages, mainly for modelling IT systems (UML, SysML, ...) and define the semantics ousing metaschemata using a language called Meta Object Facility (MOF) (ISO/IEC 19508:2014)
- ISO/IEC 15414:2015 Information technology — Open distributed processing — Reference model – Enterprise language (for ODP systems)
- ISO 19440:2007 Enterprise Integration – *Constructs for Enterprise Modelling* developed jointly by CEN/TC 310/WG 1 and ISO TC184 SC5 WG1
- ISO 19440:2007 includes *metaschemata* (expressed as UML Class diagrams) of the required modelling language constructs (Annex C to 19440)
- This particular set of languages were specifically developed to contain constructs for *enterprise modelling for model based control* of manufacturing processes
- Some Architecture Frameworks define their own metaschema (or metamodel) for many constructs promoted or required by the AF (e.g. NATO AF NMM)
- In the following slides we show *one* example (ISO 19440) of such metaschema
- In reality, for *every* modelling language in use there should exist a corresponding semantic definition in form of metamodel <u>and</u> ontological theory! Unfortunately there exist too many competing metaschemata ☹ and too few ontological theories to go with them ☹

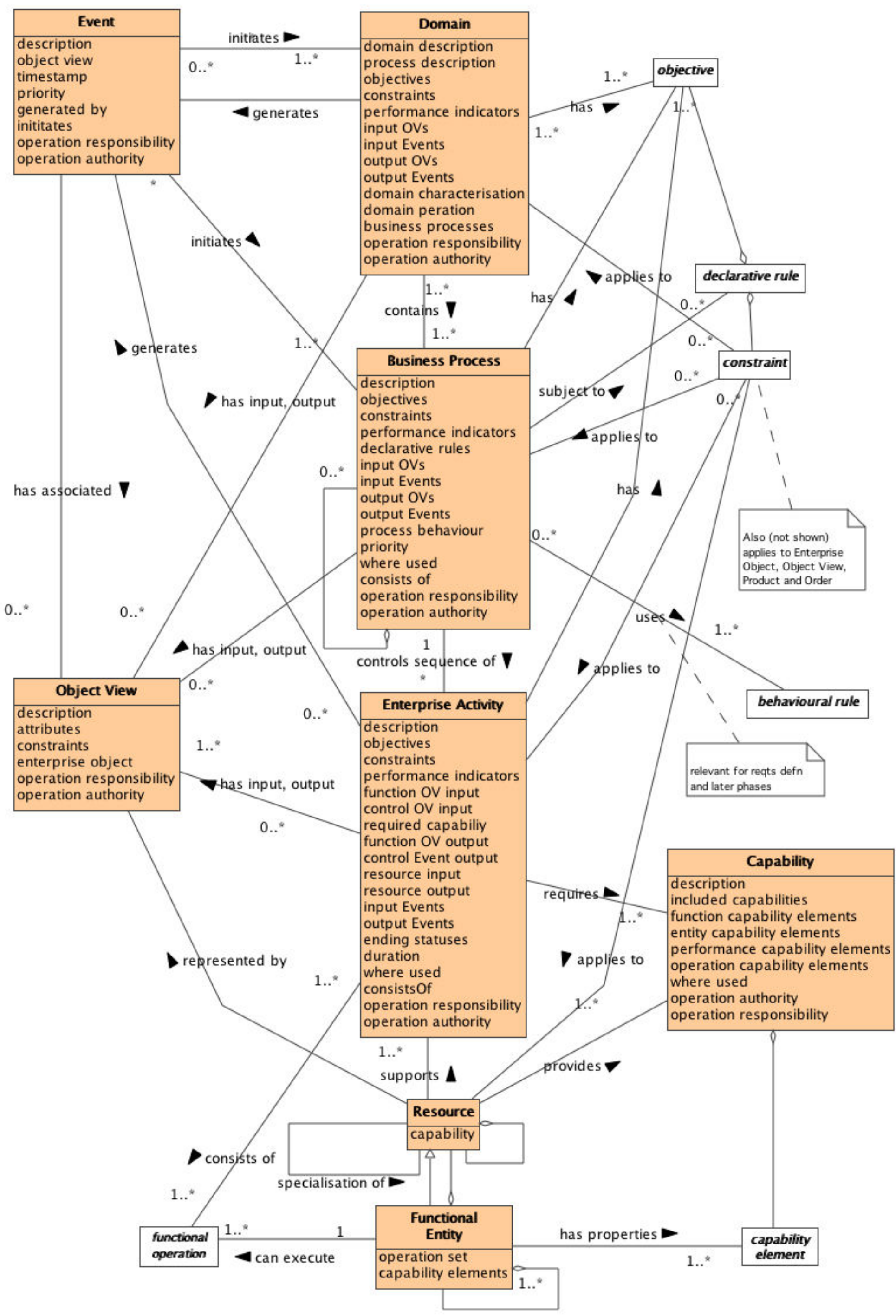
ISO 19440:2007

Information aspect



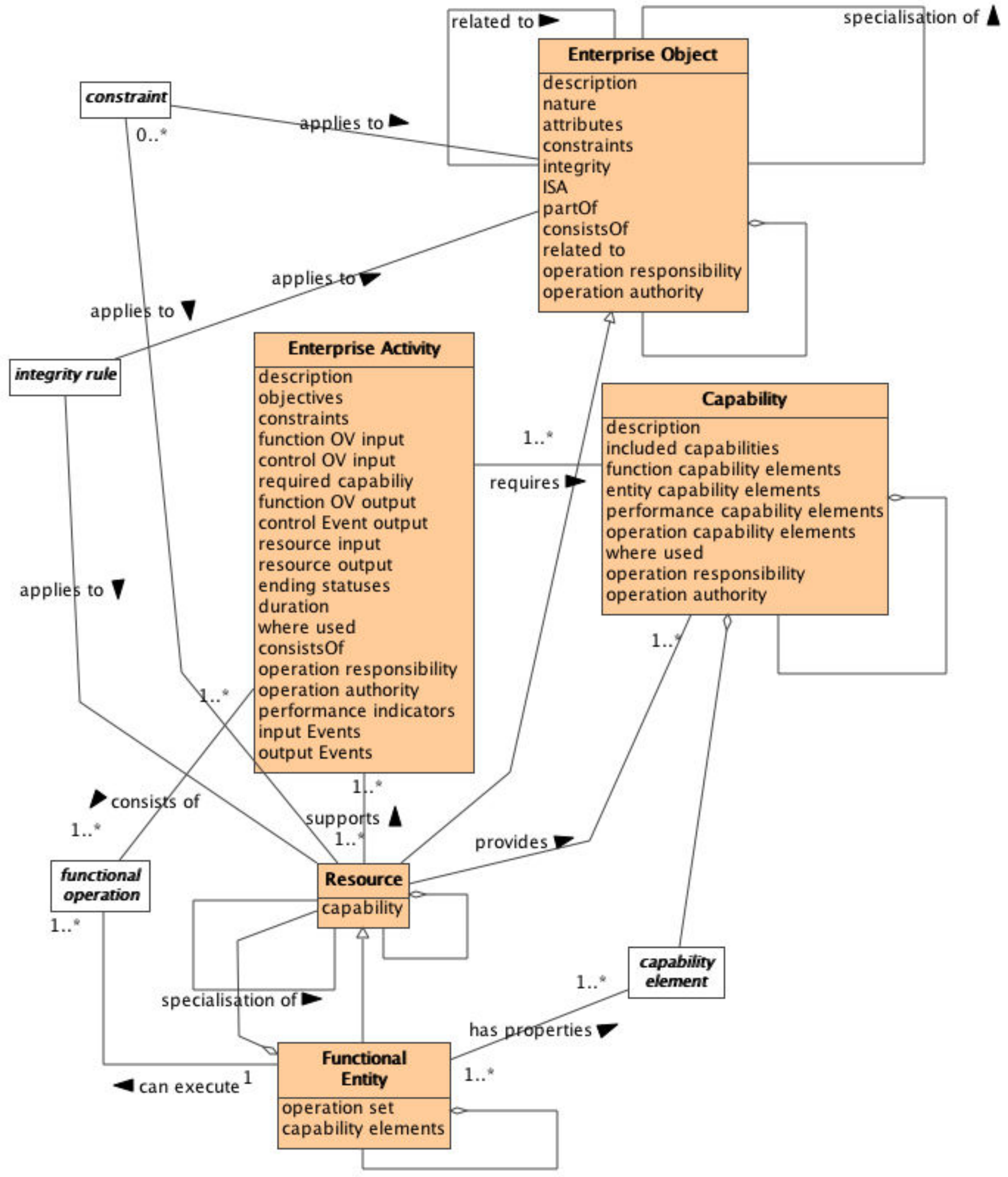
ISO 19440:2007

Function-aspect

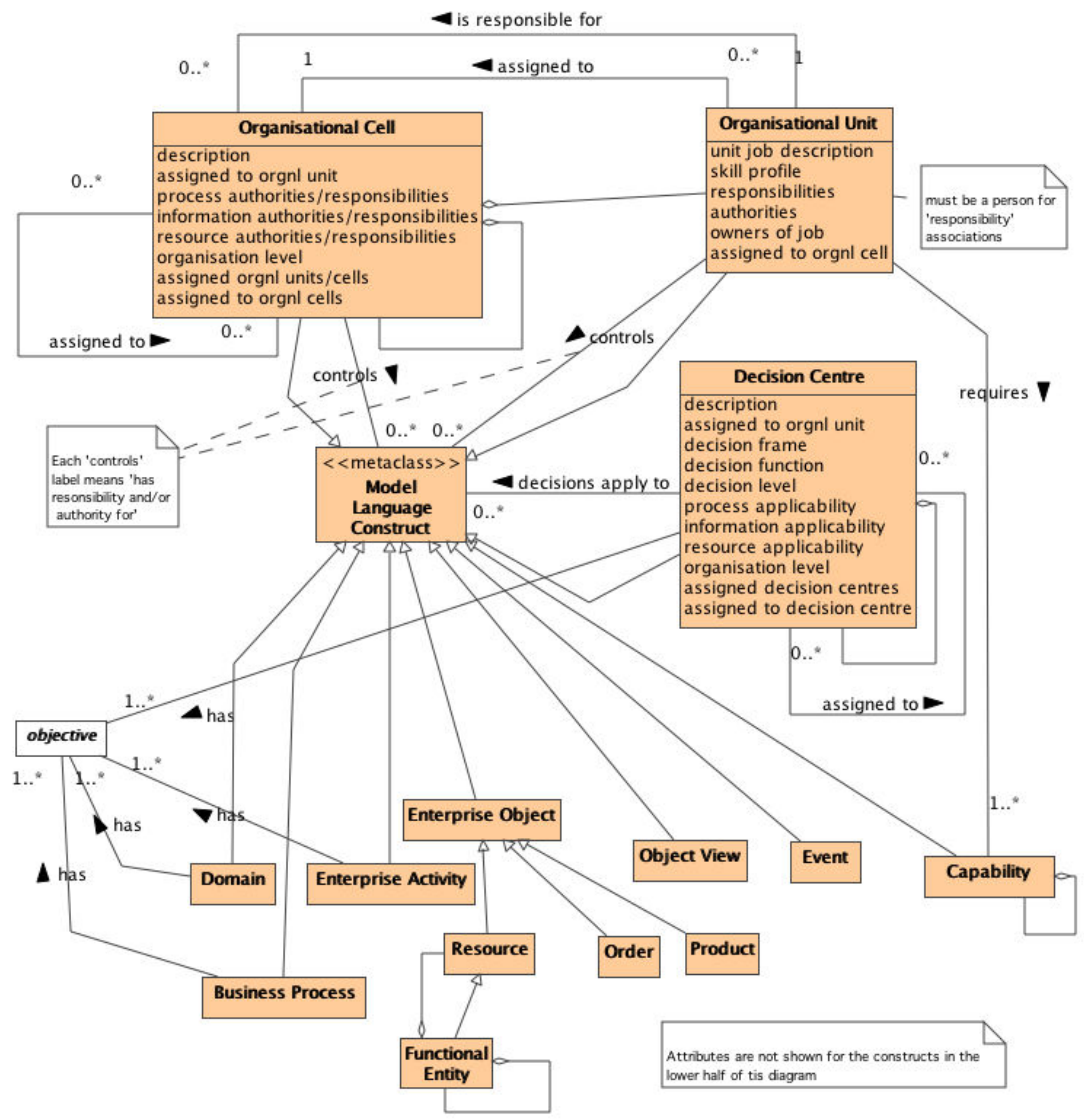


ISO 19440:2007

Resource aspect



Organisationa/ decisional aspect



Some other relevant standard documents

1. ISO IS 10303-11:1994 EXPRESS Language (of historical interest, demonstrates that language expressive power must match stakeholder concern, or the language is not adequate for modelling, e.g. simple OO modelling language was insufficiently expressive for the purpose)

2. ISO/IEC 15414 – Open Distributed Processing (ODP) [see OMG]

3. ISO 18629 – Process Specification Language (PSL)

   PSL is a *formal model of processes in First Order Logic* (process ontology)
   http://www.mel.nist.gov/psl/ontology.html

4. ISO/IEC 15909 – High-level Petri nets (the archetype of most process modelling languages used by the IT industry)

5. DoDAF has recommendations of languages to be used for each of its many viewpoints

6. A very long list of 'industry standard' modelling languages exist but precise semantics is rarely defined …

EMLs

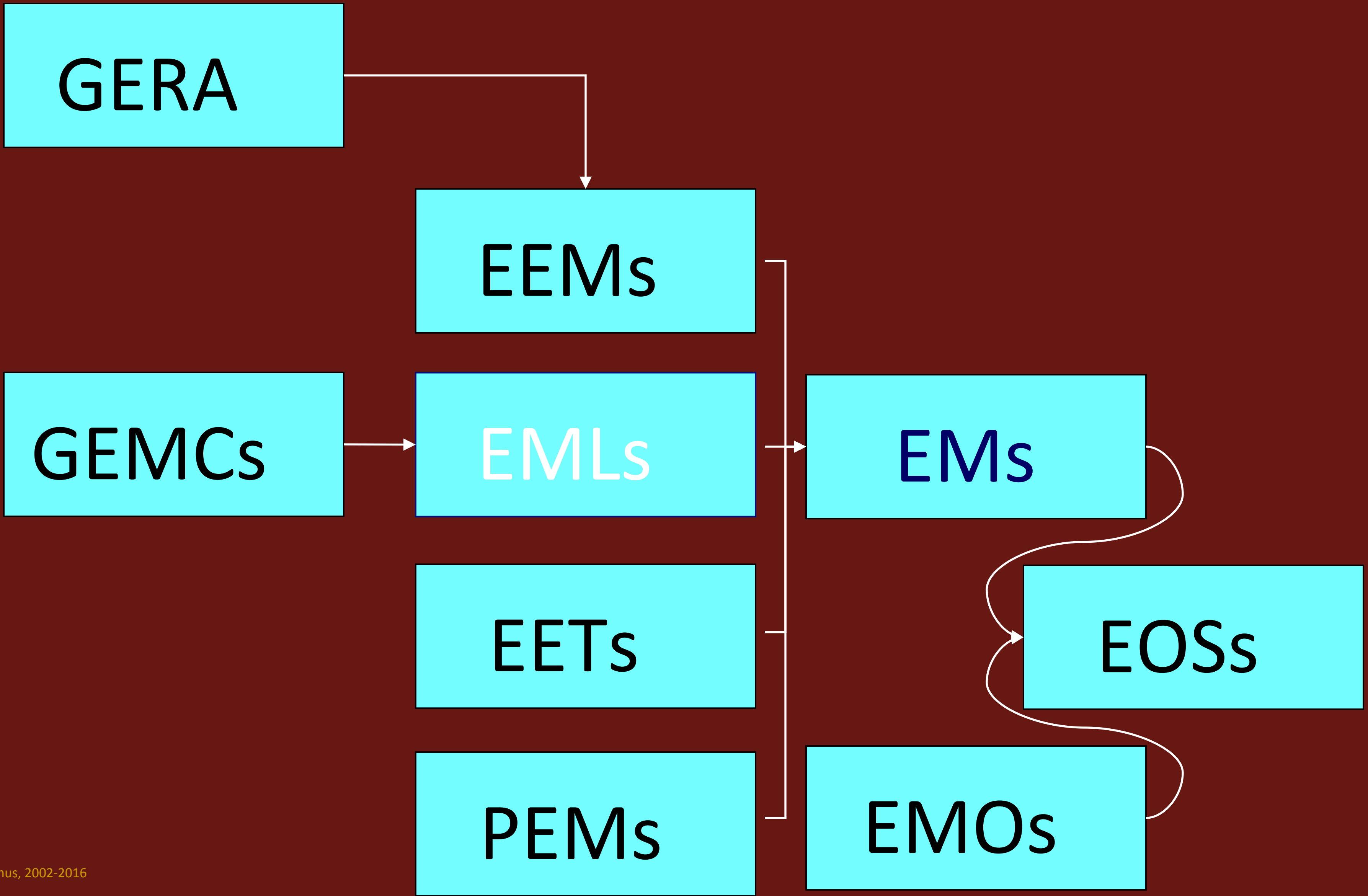
Enterprise Modelling Languages

provide modelling constructs for modelling of human role, processes and technologies

Ideally all areas in the modelling framework would need suitable language<u>s</u>
for every type of stakeholder concern – some languages are formal, some are not...

We need to know the 'ususal suspects'

The Components of GERAM



Available languages

- For each <u>area</u> of the GERA modelling framework one can identify <u>several</u> languages that exist and are in use today in industry!
- GERA modelling framework only identifies such areas
- Depending on the aim of modelling, languages of various expressive power are needed (as determined by the stakeholder concern that need to be responded to by way of creating models, views of which models can be analysed, presented, discussed etc.)

- E.g. ISO DIS 19440 is integrated set of modelling constructs (languages that originated from CIMOSA) for business process modelling, suitable to express models that can be used for model based control (e.g., for manufacturing applications)

- For the ‘downstream’ models (detailed design) a great variety of modelling languages exist, e.g., geometric modelling, simulation languages etc.

- Practically no formal modelling languages exist for the identification and concept level, although a long list of very simple models are in use

Note that we not only design the processes that can be implemented by a machine (computer or production machinery), but also processes that are performed by humans (and this is often overlooked, or the human is treated as a machine in the process)

There are *many* modelling languages…

- The intention of developing PSL and 19440 was that they should be expressed as views of a common ontology / metaschema

- This would allow *interoperation* of modelling tools (exchange of models between them)

- Functional requirements modelling:
  IDEF0, CIMOSA Requirements constructs,
  UML Use Case diagrams and Activity Diagrams, ...

- Behavioural modelling / Process requirements modelling:
  IDEF3, CIMOSA Requirements constructs, Petri Nets, FirstStep,
  ARIS EPC, PSL, BPMN, Behaviour Trees, UML collaboration or sequence diagrams, …

- Function/Process design modelling
  BPEL, CIMOSA Requirements constructs, Simulation languages
  (WS-)BPEL workflow language, Programming Languages,…

- Data requirements modelling:
  ER, OOA/D, UML class diagrams, …

- Data Design modelling:
  SQL, Programming Languages' data constucts, …

- Data Implementation modelling
  SQL, XML, …

what is the difference?

Differences...

- *Level of abstraction … Select according to the life cycle phase*
- *Expressive power* …. Select depending on the analysis / design task (i.e., what kind of question can a model answer?)
- *Rigour* … Is the meaning of modelling constructs well defined?
- *Ease of use* … do people of various backgrounds understand it in the same way? Is good graphic representation available (usability, readability,…)?
- *Availability of support tool*
- *Extendable* … can one define new constructs / extend existing ones, is there a ‘metamodelling’ capability?

Some languages

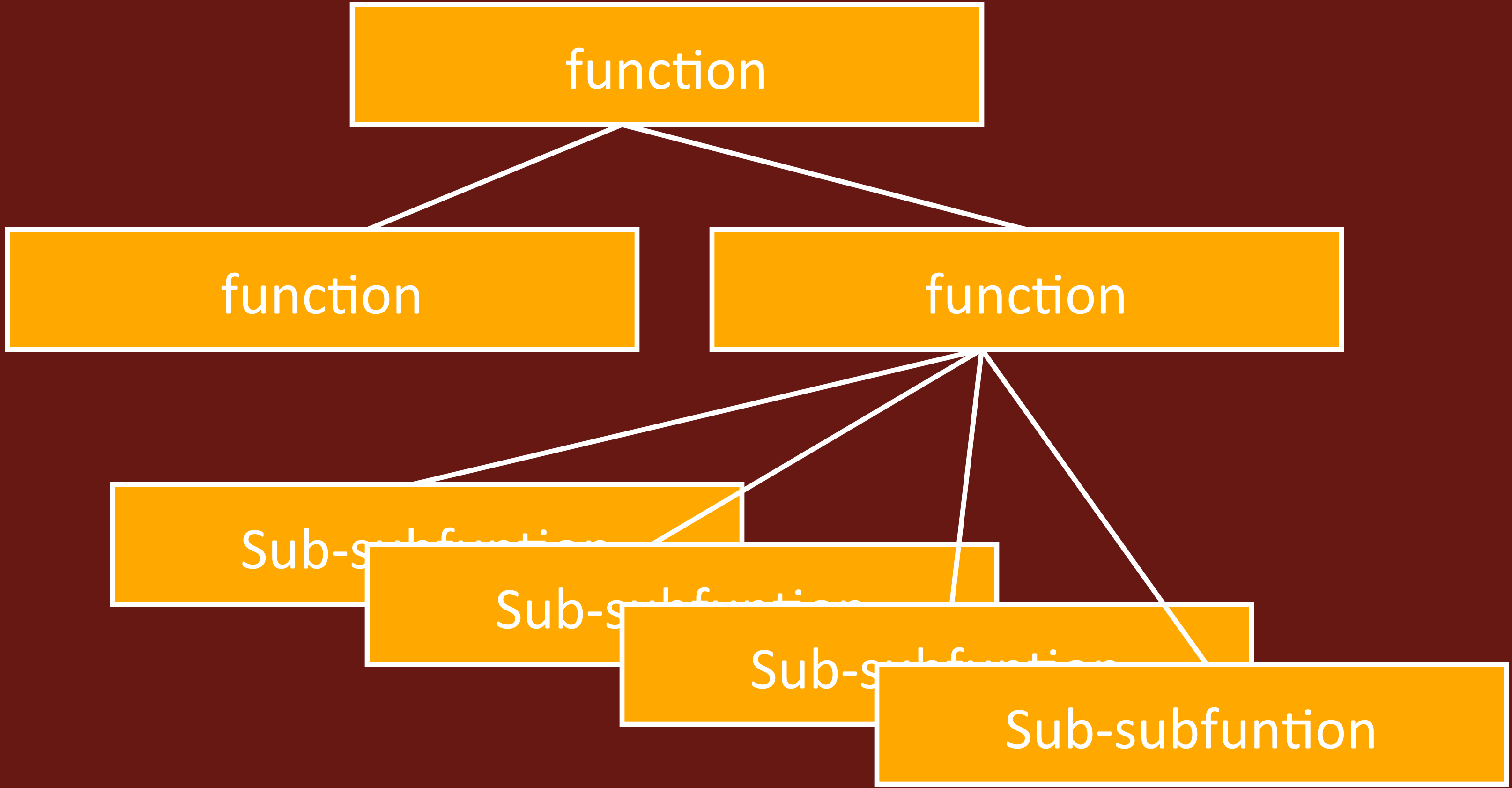
- We shall group languages by the life-cycle phase and then subgroup by viewpoints

  E.g., Requirements specification life cycle phase – function / behaviour modelling viewpoints

Requirements specification life-cycle phase – function / behaviour modelling viewpoint

- Functional decomposition

Sometimes this all we need to know: what are the functions in the enterprise? (e.g., can be used to scope the tasks of organisational units, or to describe what functions do certain SW applications support). Example: HL7 Functional Model in Health Care



Function List

1. Overarching (OV)

OV.1 Overarching Criteria

2. Care Provision (CP)

CP.1 Manage Clinical History

CP.2 Render externally-sourced Information

CP.3 Manage Clinical Documentation

CP.4 Manage Orders

CP.5 Manage Results

CP.6 Manage Medication, Immunization and Treatment Administration

CP.7 Manage Future Care

CP.8 Manage Patient Education & Communication

CP.9 Manage Care Coordination & Reporting

3. Care Provision Support (CPS)

CPS.1 Record Management

CPS.2 Support externally-sourced Information

CPS.3 Support Clinical Documentation

CPS.4 Support Orders

CPS.5 Support for Results

CPS.6 Support Treatment Administration

CPS.7 Support Future Care

CPS.8 Support Patient Education & Communication

CPS.9 Support Care Coordination & Reporting

CPS.10 Manage User Help

4. Administration Support (AS)

AS.1 Manage Provider Information

AS.2 Manage Patient Demographics, Location and Synchronization

AS.3 Manage Personal Health Record Interaction

AS.4 Manage Communication

AS.5 Manage Clinical Workflow Tasking

AS.6 Manage Resource Availability

AS.7 Support Encounter/Episode of Care Management

AS.8 Manage Information Access for Supplemental Use

AS.9 Manage Administrative Transaction Processing

5. Population Health Support (POP)

POP.1 Support for Health Maintenance, Preventative Care and Wellness

POP.2 Support Population-Based Epidemiological Investigation

POP.3 Support for Notification and Response

POP.4 Support for Monitoring Response Notifications Regarding a Specific Patient's Health

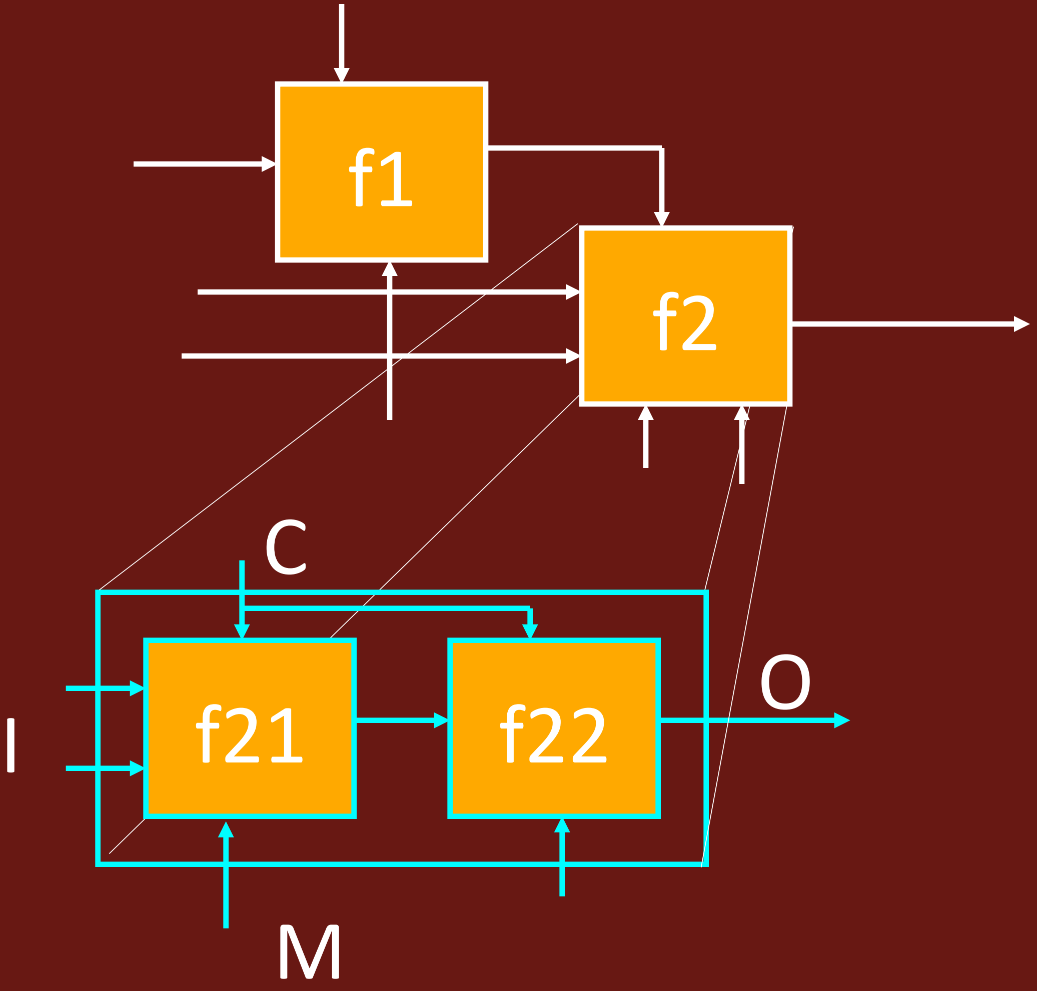
POP.5 Donor Management Support

...

Example: extract from ISO/HL7 10781 - Electronic Health Record System Functional Model

Requirements specification life-cycle phase – function / behaviour modelling viewpoint

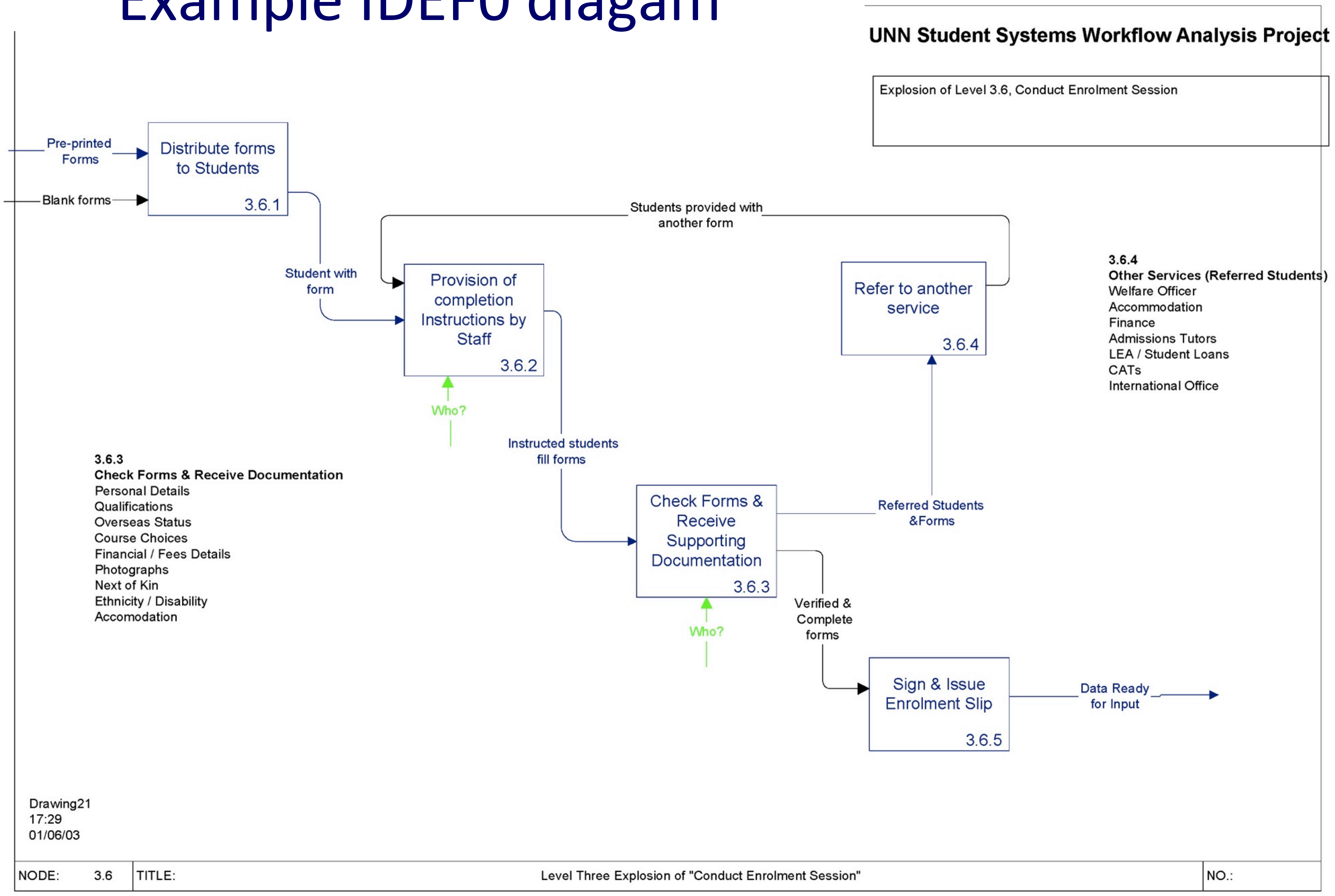
- Activity Modelling – IDEF0, Data Flow Diagrams
  - Abstracts from time
  - Expresses functional decomposition
  - Inputs, outputs, controls, mechanisms (resources)
  - I/O relationships



Activity

'Concept' (information, event, material, resource)

Example IDEF0 diagam


UNN Student Systems Workflow Analysis Project
Explosion of Level 3.6, Conduct Enrolment Session
Pre-printed Forms
Blank forms
Distribute forms to Students 3.6.1
Student with form
Students provided with another form
Provision of completion Instructions by Staff 3.6.2
Who?
Refer to another service 3.6.4
Instructed students fill forms
Check Forms & Receive Supporting Documentation 3.6.3
Who?
Referred Students &Forms
Verified & Complete forms
Sign & Issue Enrolment Slip 3.6.5
Data Ready for Input
Drawing21
17:29
01/06/03
NODE: 3.6
TITLE: Level Three Explosion of "Conduct Enrolment Session"
NO.:

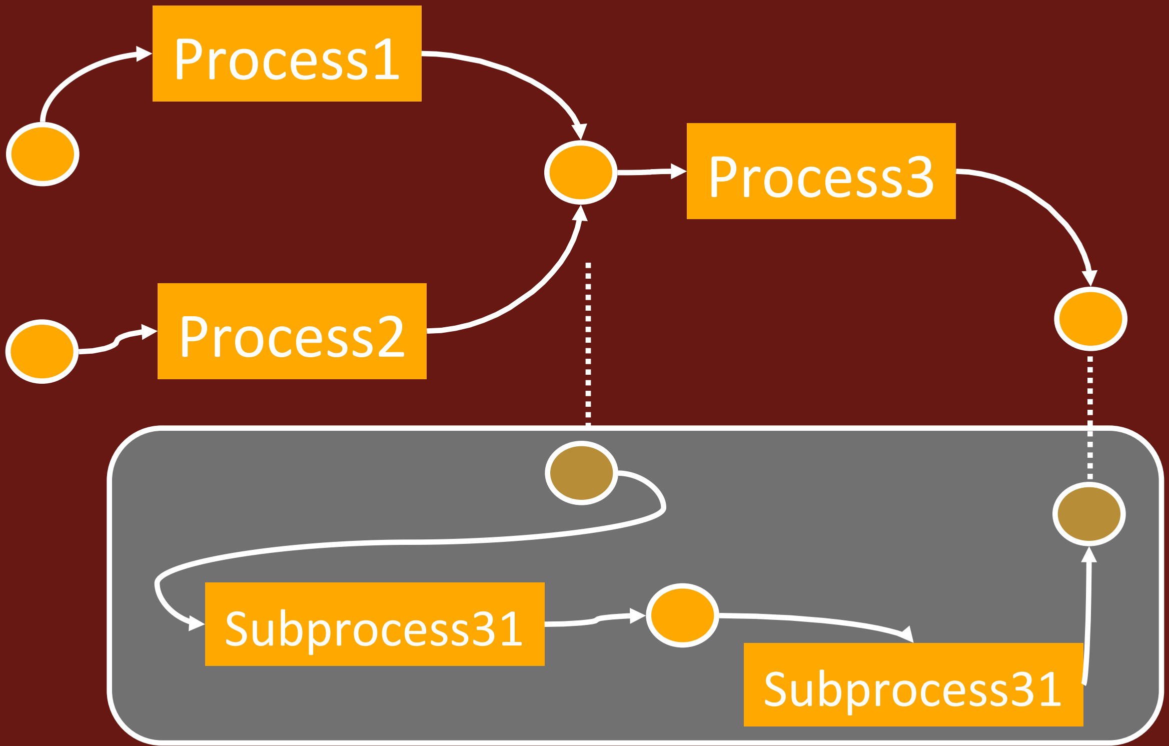
3.6.4
Other Services (Referred Students)
Welfare Officer
Accommodation
Finance
Admissions Tutors
LEA / Student Loans
CATs
International Office

3.6.3
Check Forms & Receive Documentation
Personal Details
Qualifications
Overseas Status
Course Choices
Financial / Fees Details
Photographs
Next of Kin
Ethnicity / Disability
Accomodation

Source: http://www.jiscinfonet.ac.uk/tools/idef0/

Requirements specification life-cycle phase – function / behaviour modelling viewpoint

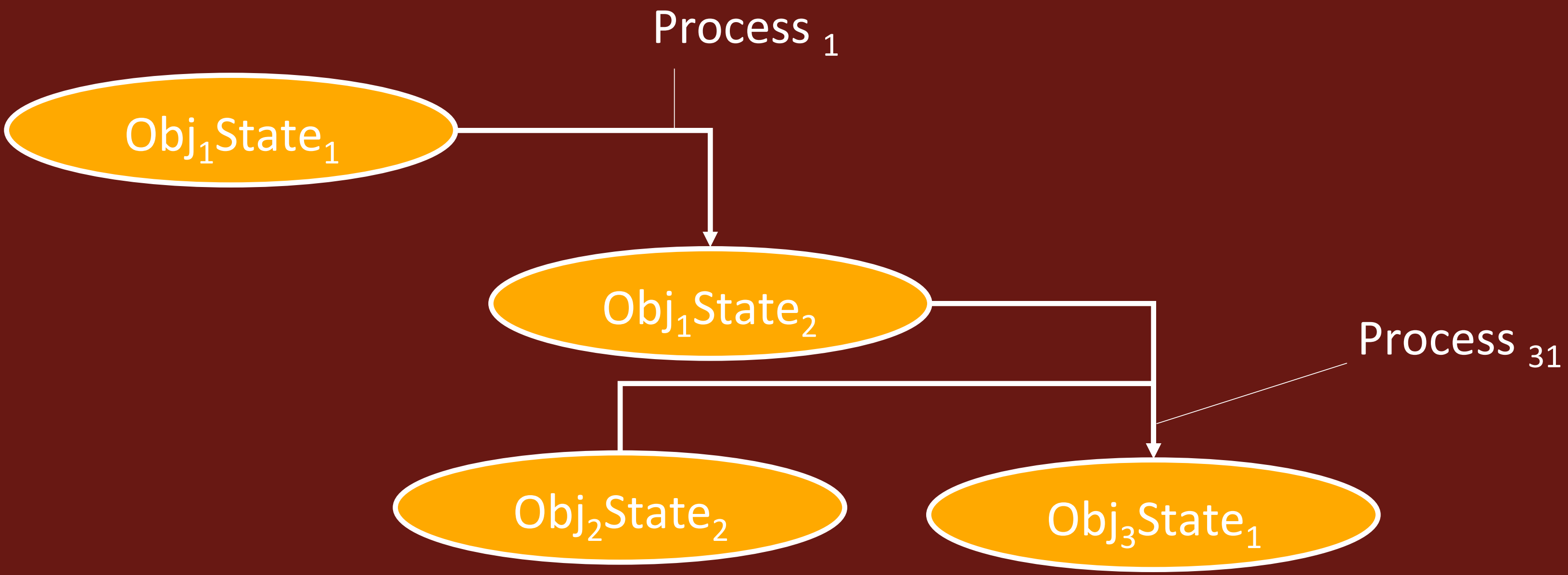
- Behavioural modelling – Petri Nets, … (BPMN, CIMOSA function modelling constructs/FirstStep, IEM, ARIS Event Driven Process Chains, IDEF3, UML collaboration diagram, Behaviour Trees, …)
  - Define sequences of events (behaviour)
  - Functional decomposition
  - Some of these languages can also express I/O rel' ships
  - Some additional attributes (language depending)



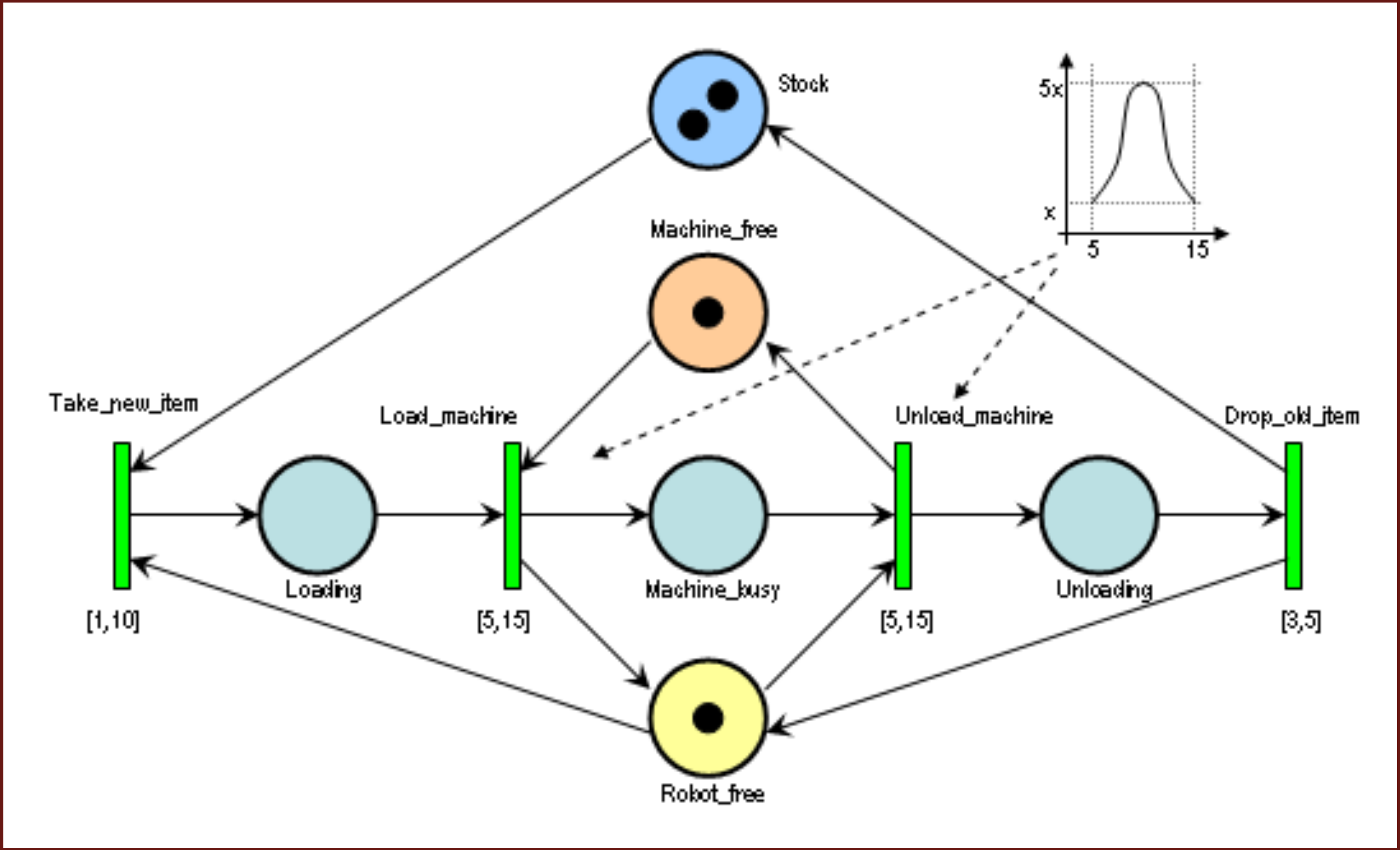
event ● process ▬

- Behavioural modelling of functions is often accompanied by state transition diagrams (e.g. IDEF3 / UML have State transition diagrams / Harel State Charts respectively)

E.g.
Process$_1$ transforms Object$_1$ from State$_1$ [precondition] to State$_2$ [postcondition];
Process$_{32}$ needs Object$_1$ to be in State$_2$ [precondition] and causes
Object$_2$ to reach State$_2$ and Object$_3$ to reach State$_1$ [postcondition]

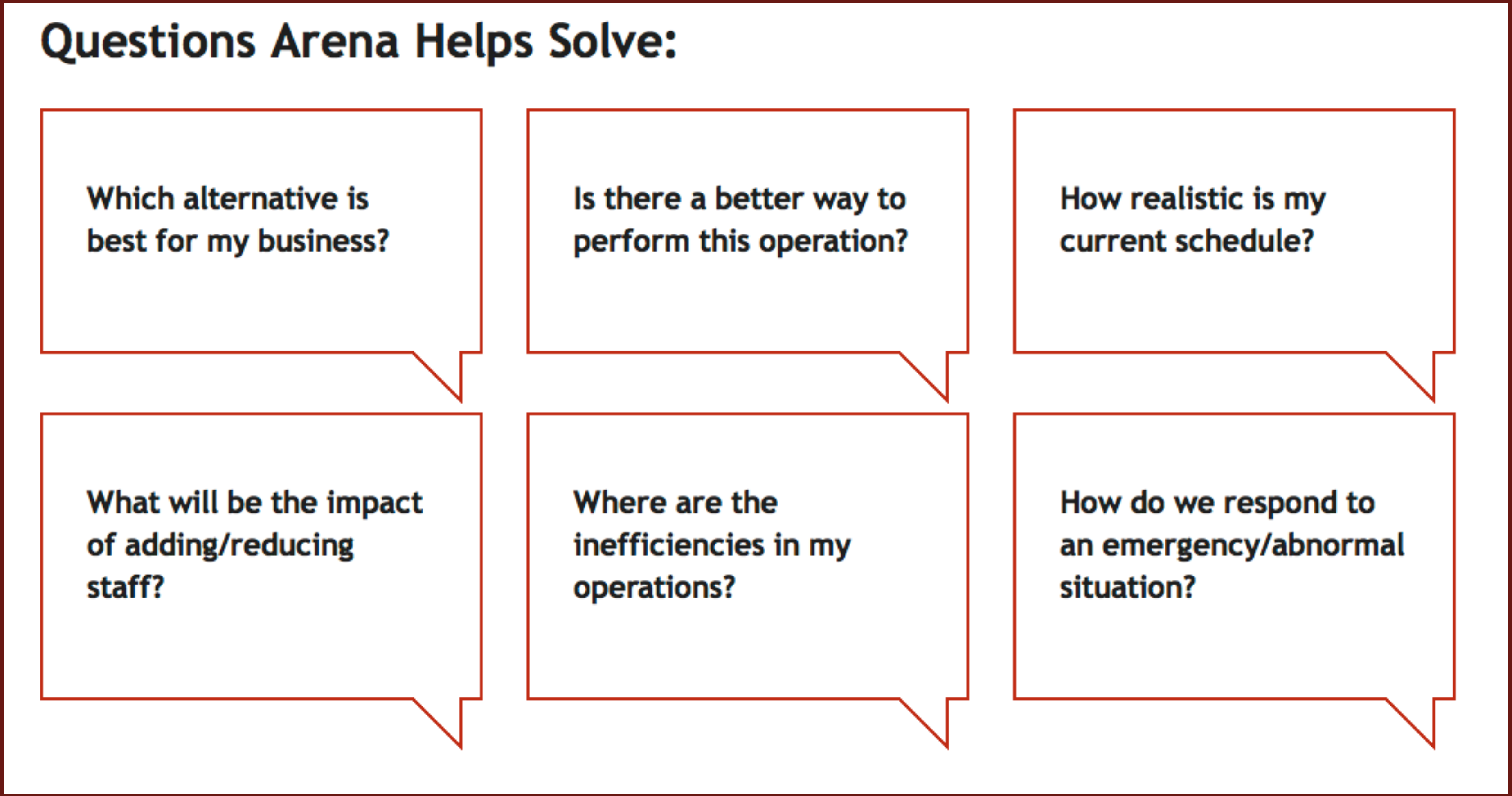


Example (somewhat simplistic) Petri net



Source: http://dainf.ct.utfpr.edu.br/~maziero/doku.php/software:arp_manual_english

Simulation languages / tools
Example: ARENA for discrete event simulation



Questions Arena Helps Solve:

Which alternative is best for my business?

Is there a better way to perform this operation?

How realistic is my current schedule?

What will be the impact of adding/reducing staff?

Where are the inefficiencies in my operations?

How do we respond to an emergency/abnormal situation?

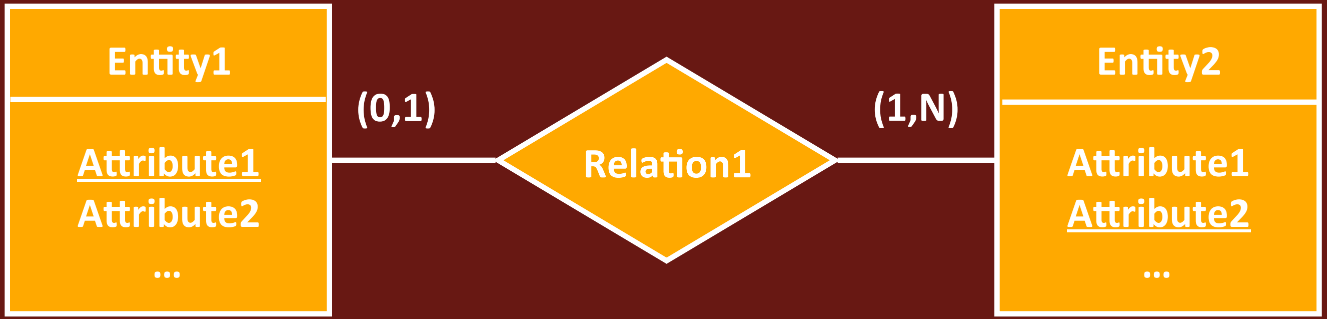
Simulation is able to answer questions like…

- Compare process alternatives for performance optimisation
- Compare process costs, throughput, cycle times, equipment utilization and resource availability
- Simulation and testing of process changes
- Calculate impact of uncertainty and variability on system performance
- Visualise the above using 2D / 3D animations
- …

ARENA is one of the most frequently used modelling tools by industrial engineers (see details at https://www.arenasimulation.com/)

Requirements specification life-cycle phase – information modelling view

- Information modelling languages: Entity-relationship modelling (<u>ER, IDEF1X</u>) and object modelling (<u>ORM, UML class diagram, EXPRESS, CIMOSA object view constructs, IEM</u>). Note: ER and OO are *almost identical* here – object modelling is not the same as object oriented design where both the information and behavioural aspects are designed using the same model!



Many notations exist, some more rich then the other

(0,1) : participation/ cardinality constraint

Requirements specification life-cycle phase – resource modelling view

- Few languages allow detailed specification of resource requirements and resource capabilities – e.g. CIMOSA resource modelling constructs *do* (see ISO DIS 19440 for constructs)

  However, if using other languages (IDEF0, 3, ARIS EPC, etc.)

  – resource *requirements* can be represented as text annotations to the function/process models

  – resource *capabilities* as text annotations to the resource as an object

- UML Component diagrams + text annotation

Requirements specification life-cycle phase – organisational modelling view

- Depending on the level of detail the organisation needs to be modelled, there are several choices – CIMOSA, ARIS, GRAI-Grids
- The essence of the organisational view is the *definition of responsibilities*
- However, on the requirements level we do not decide on the aggregation and mapping of responsibilities (functions) into roles (that may be played by machines or humans), but *we state the requirements that constrain such mapping*
- *Organisation* is in fact a relation between tasks (functions) and resources and as such is only decided on the architectural design level

Architectual Design level

- Function – IDEF0,3,CIMOSA, IEM, ARIS EPC, Workflows
- Information – Relational Schema, XML, Object Schema,
- Resource – CIMOSA, IEM, ARIS res. view., Component diagram …
- Organisation – CIMOSA, GRAI-Net, ARIS org view, … etc

+ various domain-dependent engineering & financial & HR models

Note: on this level the functional requirements become aggregated and assigned to hw/sw/human resources

Instead of adding new detail to the function/ information, we develop the *system architecture* (i.e., how the system is made up of its constituents and implements its function / information.

To do this, we develop a more detailed structure of resources (human and machine) and allocate these to implement functions / (store/transfer) information

We must make sure that the required capabilities & capacities of resources match the actual resource capabilities & capacities: *the engineering & financial & HR models used must be able to prove that the above aggregation and mapping is valid*

Example: architectural design level of the organisational viewpoint

- *Human roles* on the architectural design level
  - An organisational chart is a very simple model, mapping human resource to a functional decomposition, but is not suitable for analysing the quality of the mapping
  - It is better to map roles to a more detailed representation of the function to be able to prove that he mapping is good (see some methods later in the *'decisional modelling' seminar*)
  - Note also that normally only part of the organisation is defined up-front, some of the organisation is dynamically created / re-created as roles get temporarily created and assigned
- Similarly, it is necessary to aggregate machine functions into *machine roles*
  - The architectural design principles regarding what is the best way to do this are discussed in the *'architectural design' seminar*
  - Component diagram is a way to express the role aggregations and interfaces
  - Some of the architectural design is dynamically created (but the scope of possible dynamic structures must still be designed, see *'virtual enterprise challenge' seminar*)

Detailed Design level

- Function – IDEF0/3, CIMOSA/ IEM implementation level constructs, BPEL, or other Workflow Modelling Languages, Programming languages
- Information Relational Schema, XML, Various programming languages
- Resource – CIMOSA, IEM, ARIS, Organisation – CIMOSA, GRAI-Net
- Organisation – extensions to the existing mapping

+ various domain-dependent engineering & financial & HR models

On this level more detail is added, enough for deployability / commissioning / puchasing ...

Note: detailed design of software is *coding*.
(the ‘build’ LC phase corresponds to release to operation)

Summary

- GERA Modelling Framework
  (defines the scope of modelling)
- Enterprise Models
  Generic concepts (GEMCs) – metaschema / ontology
  Particular models (what they capture)
- Modelling Languages
  (overview of the variety available for use – which can also be a source of confusion…)

The end