TDT4252/DT8802 Enterprise Modelling and Enterprise Architecture

Sobah Abbas Petersen/John Krogstie

Adjunct Associate Professor / Professor

sap@idi.ntnu.no / krogstie@idi.ntnu.no

TDT4252 Summary Lecture
Spring 2014

Purpose of this lecture

- To present a summary of the TDT4252 course for Spring 2014.

Learning Goals (from Study Plan)

- Theoretical insights into different modeling perspectives, languages and techniques for creating models of :
  - Information systems
  - Enterprises
- Practical skills in
  - Analysing situations for modelling
  - Creating good models
- The course will introduce the ideas of Enterprise Modelling and Enterprise Architecture and provide a holistic view of modelling.

Course Outline

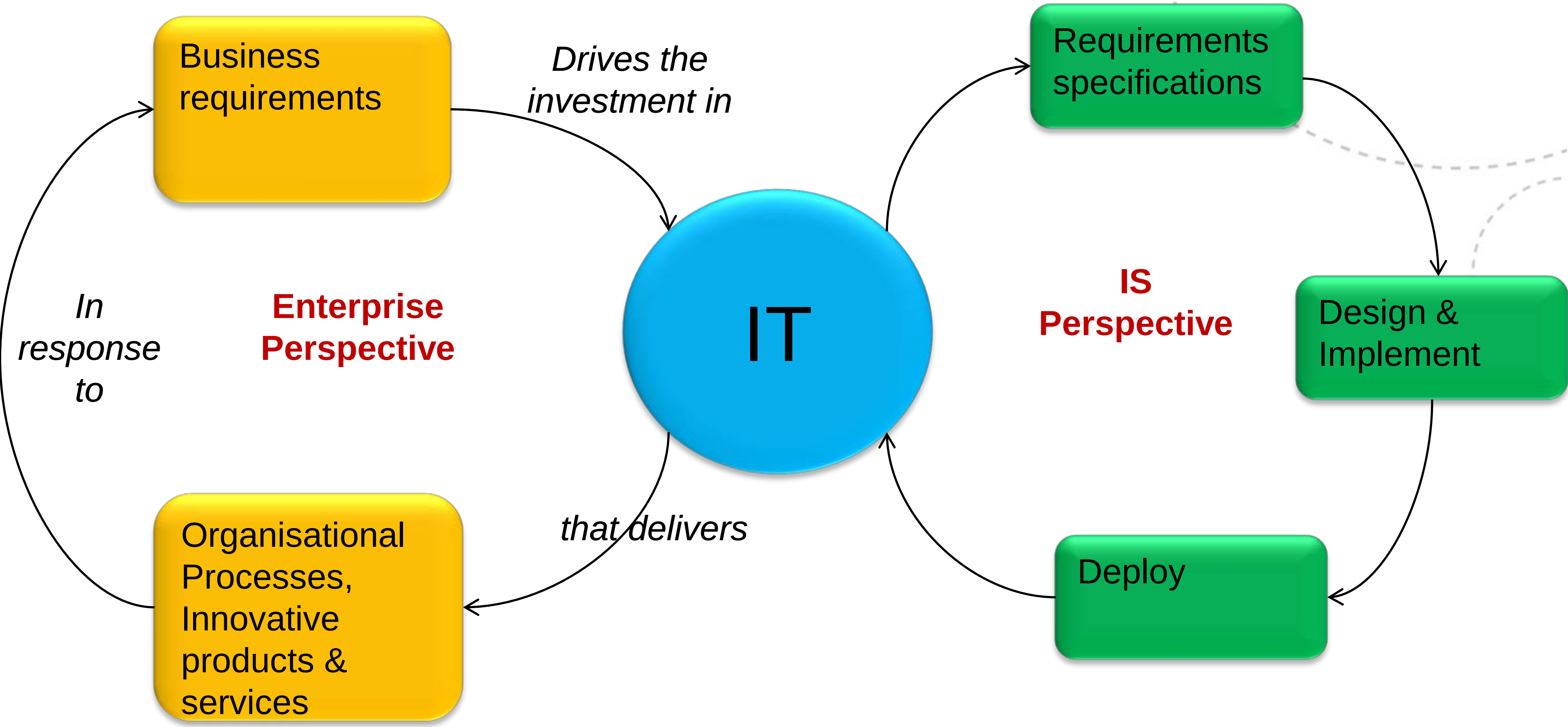
- The course will consist of the following:
  - Perspectives of modelling and different modelling approaches and languages.
  - Active Knowledge Modelling (AKM)
  - Enterprise Modelling
  - Enterprise Architectures
- Assignments
  - There will be one mandatory modelling assignment (Term Paper)
- Evaluation
  - A written exam – 65%
  - Assignment – 35%

Scope of the course

- Different perspectives of modelling.
- Different kinds of modelling
  - IS modelling: requirements, goals, actors
  - Process Modelling
  - Product Modelling
  - Active Knowledge Modelling
  - Enterprise Modelling
  - Enterprise Architectures
- We will use practical examples of models.
- We will look at how the different types of models relate to one another to create Enterprise Models and Enterprise Architectures.

Business Strategy and IS

Business objectives sets IT priorities!



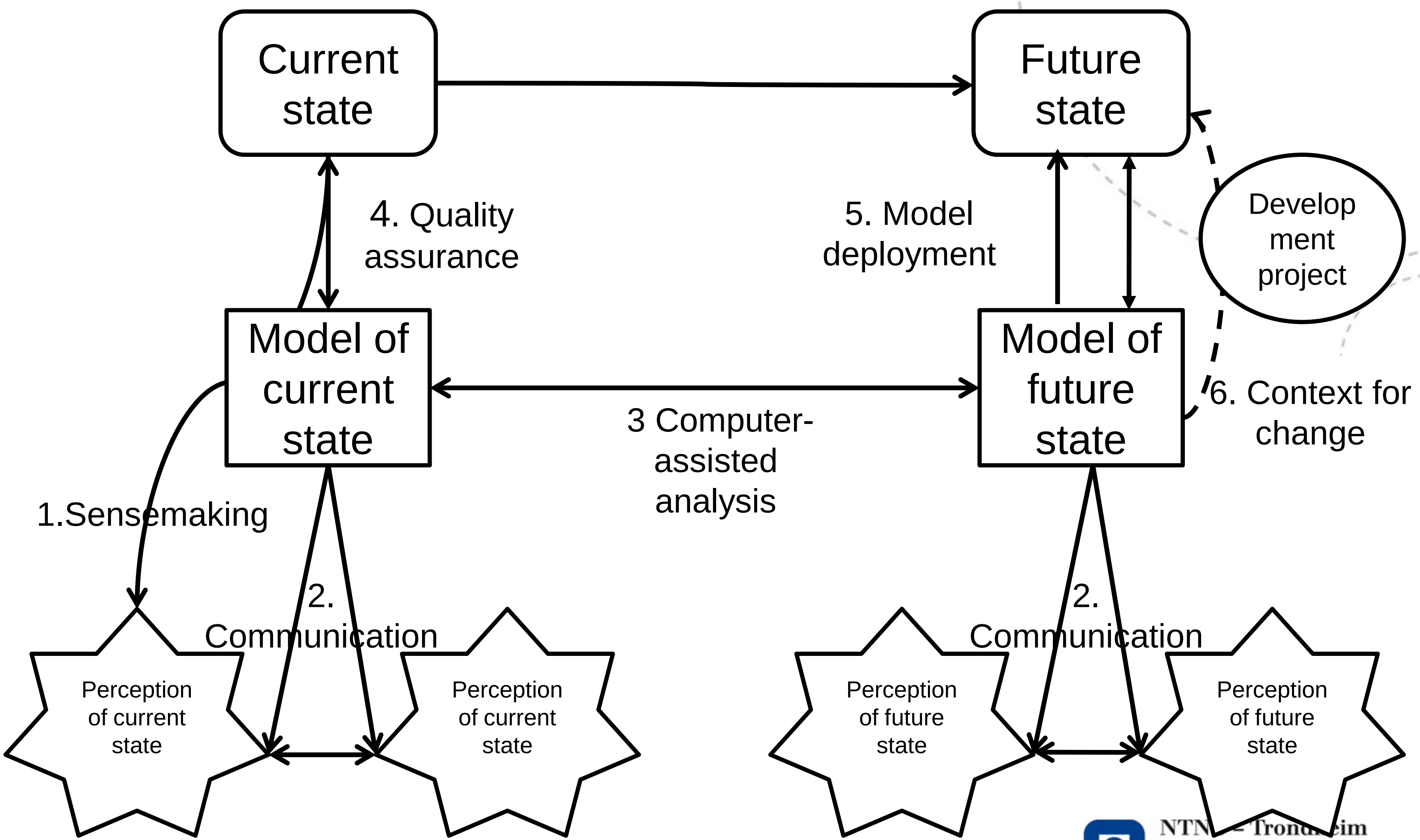
TDT4252 Summary Lecture
Spring 2014

NTNU – Trondheim
Norwegian University of Science and Technology

Purpose of the Model

- Before creating a model, it is important to understand the purpose of modelling and the purpose that would be served by the model that is created. This determines:
  - The design and focus of the model.
  - The perspectives of the model.
  - The modelling language and approach selected.
  - The modelling application.
  - The presentation of the model to the users.

Usage of modeling



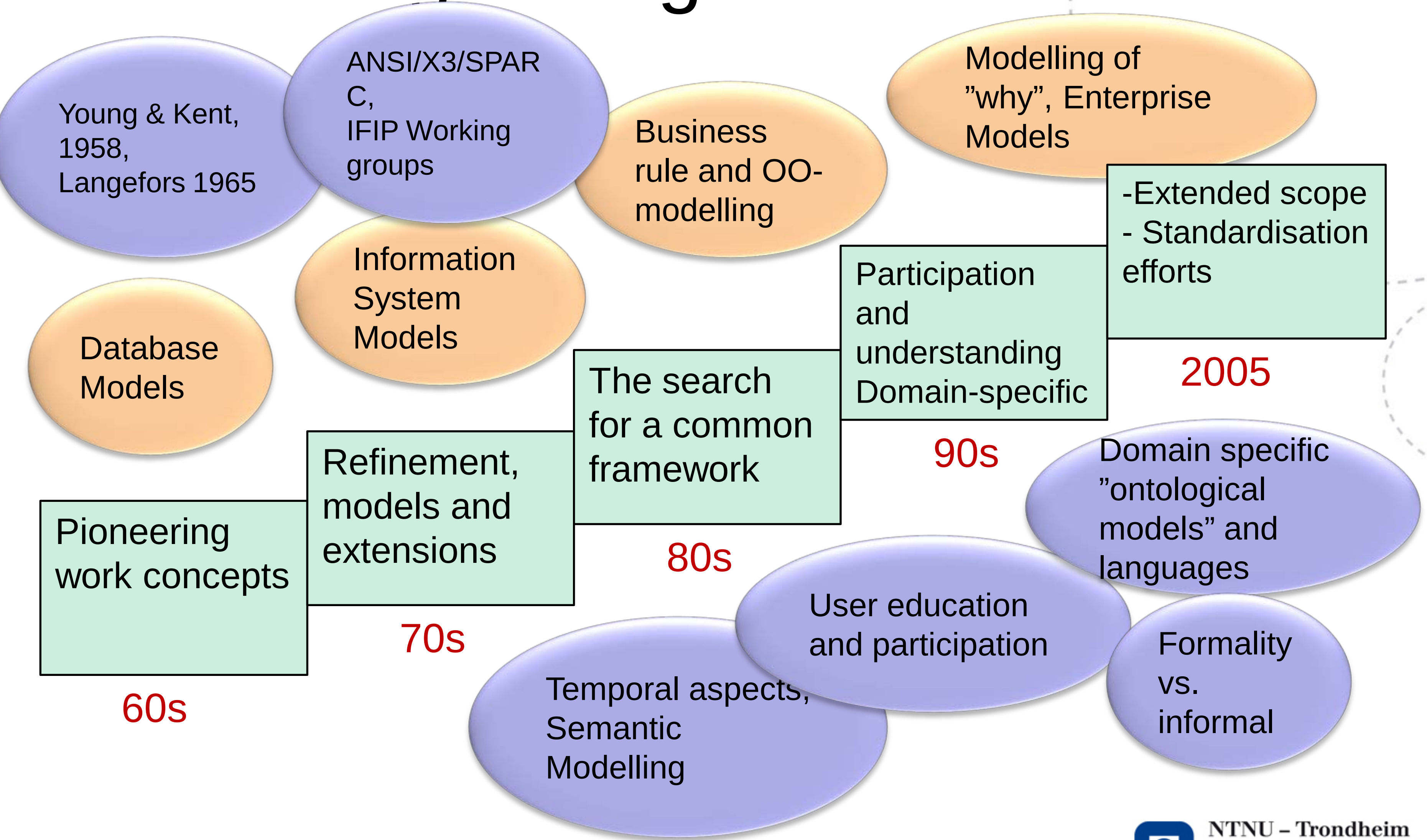
NTNU – Trondheim
Norwegian University of Science and Technology

A Historical Perspective on Conceptual Modelling

Syllabus: Parts of 2.2 and 2.3 in Krogstie: Model-based development and evolution of information systems: A quality approach

Additional material (not syllabus): Janis A. Bubenko jr: From Information Algebra to Enterprise Modelling and Ontologies – a Historical Perspective on Modelling for Information Systems in Conceptual Modelling in Information Systems Engineering. Krogstie, John; Opdahl, Andreas Lothe; Brinkkemper, Sjaak (Eds.)

Modelling during four+ decades



Young & Kent, 1958, Langefors 1965

ANSI/X3/SPARC, IFIP Working groups

Database Models

Information System Models

Business rule and OO-modelling

Modelling of ”why”, Enterprise Models

Pioneering work concepts

60s

Refinement, models and extensions

70s

The search for a common framework

80s

Participation and understanding Domain-specific

90s

-Extended scope - Standardisation efforts

2005

Temporal aspects, Semantic Modelling

User education and participation

Domain specific ”ontological models” and languages

Formality vs. informal

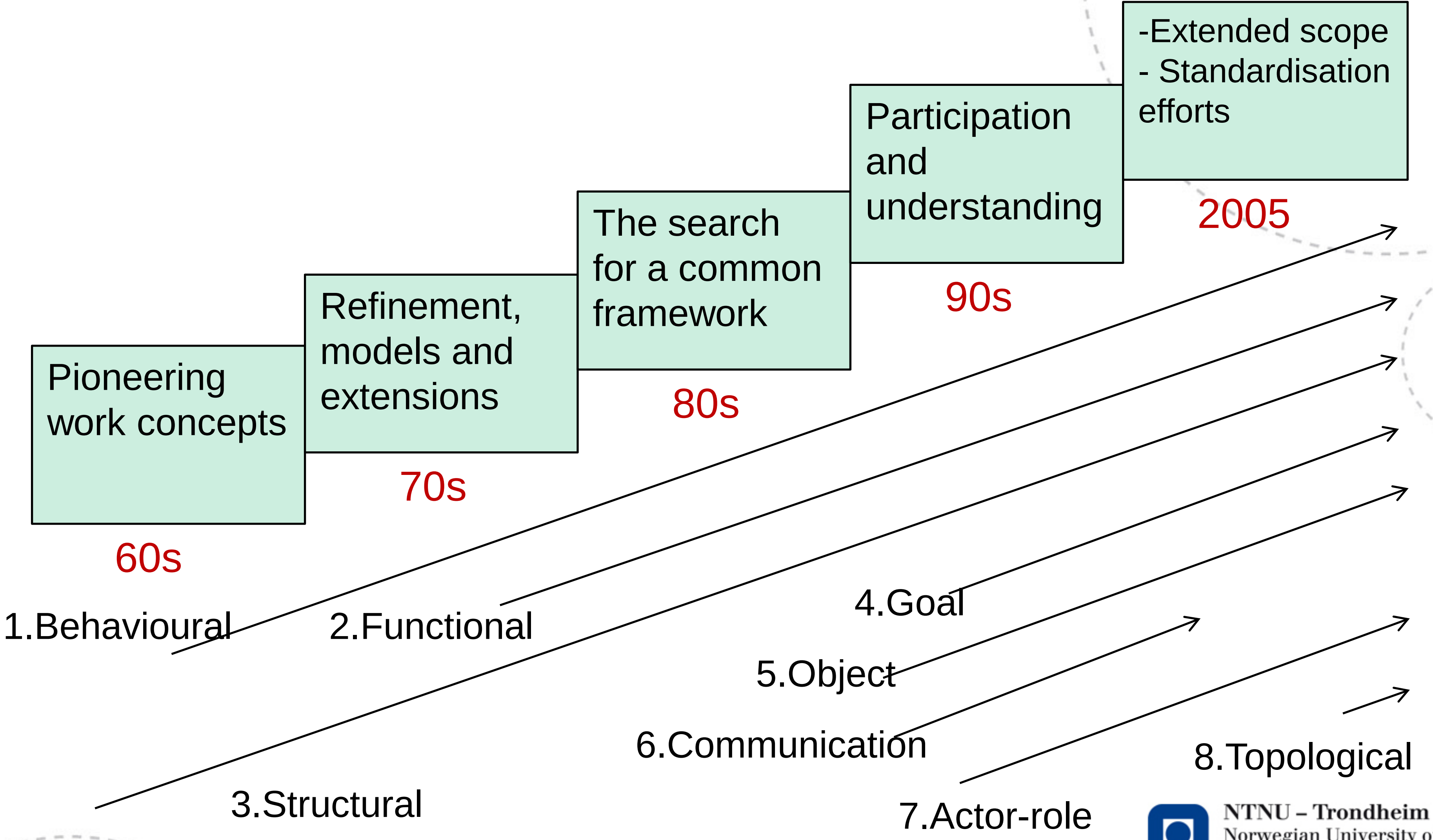
Modeling perspectives

- A modeling perspective: What concepts are important, what is emphasized, and what is ignored
  - What are the fundamental concepts
  - What aspects are explicitly represented
    - As nodes, relationships, or properties
  - What is visualized?
  - What is modeled first?
- Modeling languages with different perspectives can support the same concepts

8 perspectives/orientations to conceptual and enterprise modeling

1. Behavioral
2. Functional
3. Structural
4. Goal and rule-oriented
5. Object-oriented
6. (Social) communication
7. Actor/role-oriented
8. Topological

Modelling during four+ decades



Pioneering work concepts

60s

Refinement, models and extensions

70s

The search for a common framework

80s

Participation and understanding

90s

-Extended scope
- Standardisation efforts

2005

1.Behavioural

2.Functional

3.Structural

4.Goal

5.Object

6.Communication

7.Actor-role

8.Topological

Actor-role modelling

- Actor-role oriented modeling, **introduction to i*, GRL, UCM**

Based on the following articles:

- A01: Yu: “Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering"(Proc. RE'97)
- A02:Liu and Yu: “Designing Information Systems in Social Context: A Goal and Scenario Modeling Approach" Information Systems 29(2):187-203

Actor-oriented analysis

- Actors (persons, departments, organizations, …)
  - Focus on who and why
  - Improve understanding of needs
  - Improve structure of requirements
- Example *i* (GRL)*
  - Both actor-oriented and goal-oriented modeling
- Why i*?
  - Broad set of usage experiences by many people
  - Several large examples of usage of the technique for industrial applications.
  - Standardized as part or Requirements engineering -technique together with use case maps

Intentional Actors and relationships

A01: Introduction to i*: motivation

- Requirements engineering (RE) traditionally: WHAT, not why ,
- But there are problems in the analyses before the requirements are established:
  - **WHY** is the system built?
  - **WHO** needs it?
- i.e.
  - Understand the problem domain
  - Give users support to think about the requirements
  - Enable changes in the business process
  - Improve traceability

- **i* for early-phase RE**

A01: central concepts

- Actor
  - Perform task with a purpose (intentional)
  - Have goals, skills, responsibilities
  - Is dependent on other actors to achieve own goals
- Dependency in relation to
  - Resource (must get from another actor)
  - Task (that another actor must perform)
  - Goal (that another actor must achieve)
  - Soft-goal (that another actor must achieve)
- The above concepts are modelled in a Strategic Dependency Model (SD)

A01: Strategic Dependency (1)

Intentional relationships among organisational Actors

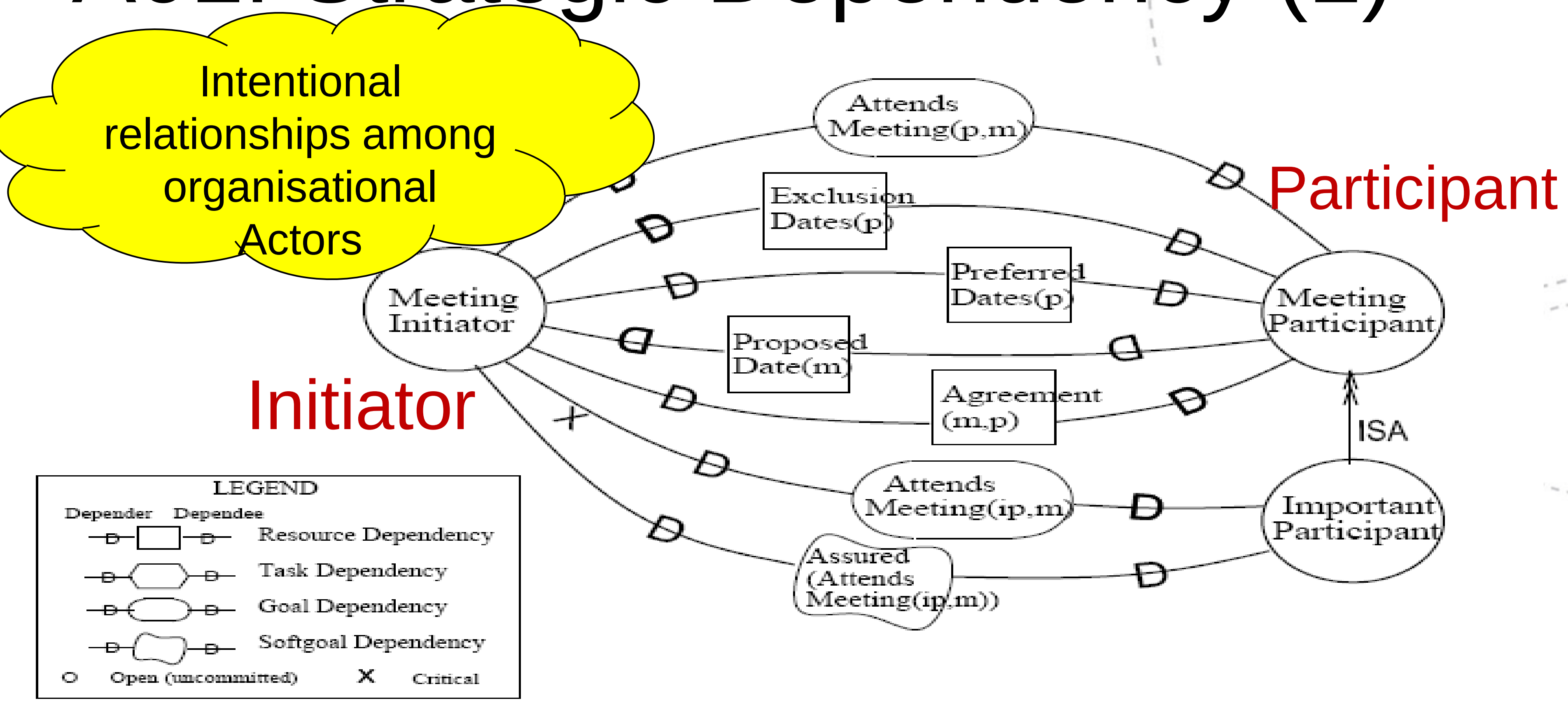
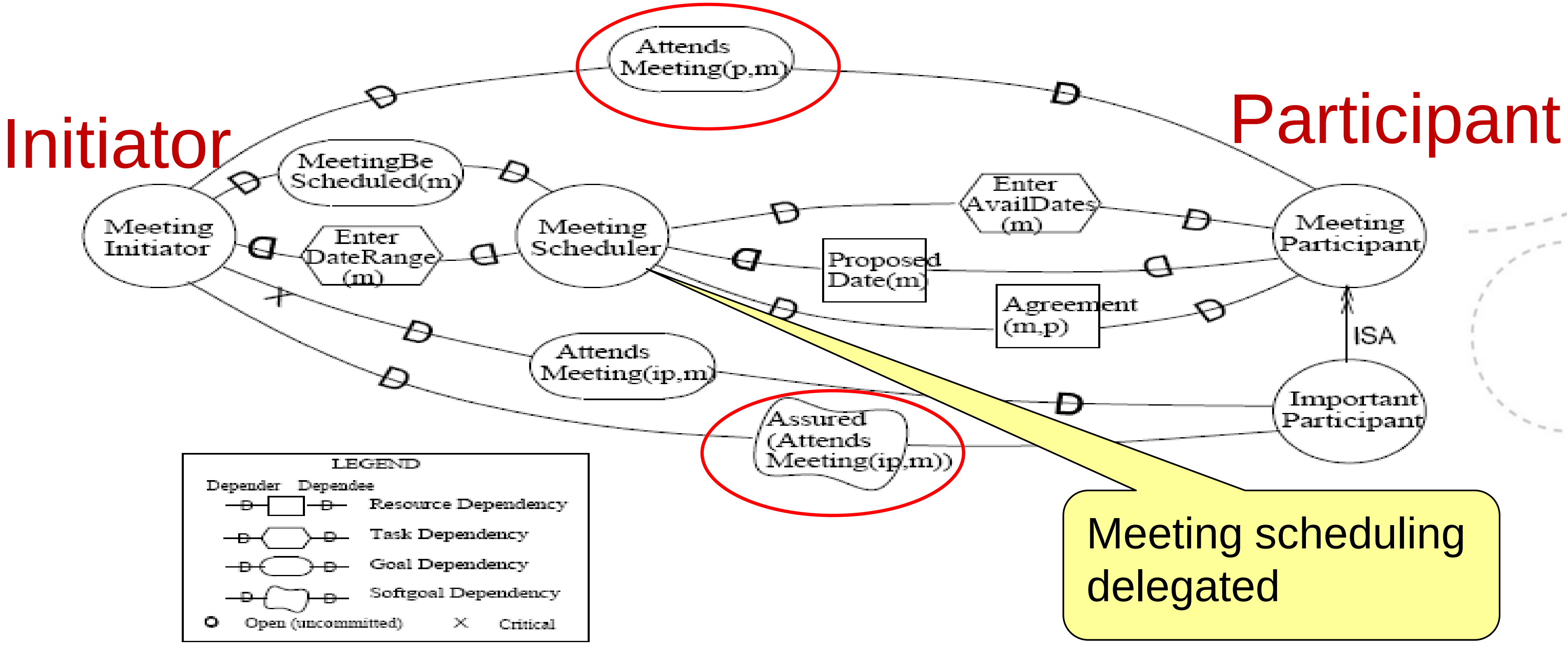


Figure 1: Strategic Dependency model for meeting scheduling, without computer-based scheduler

A01: Strategic Dependency (2)



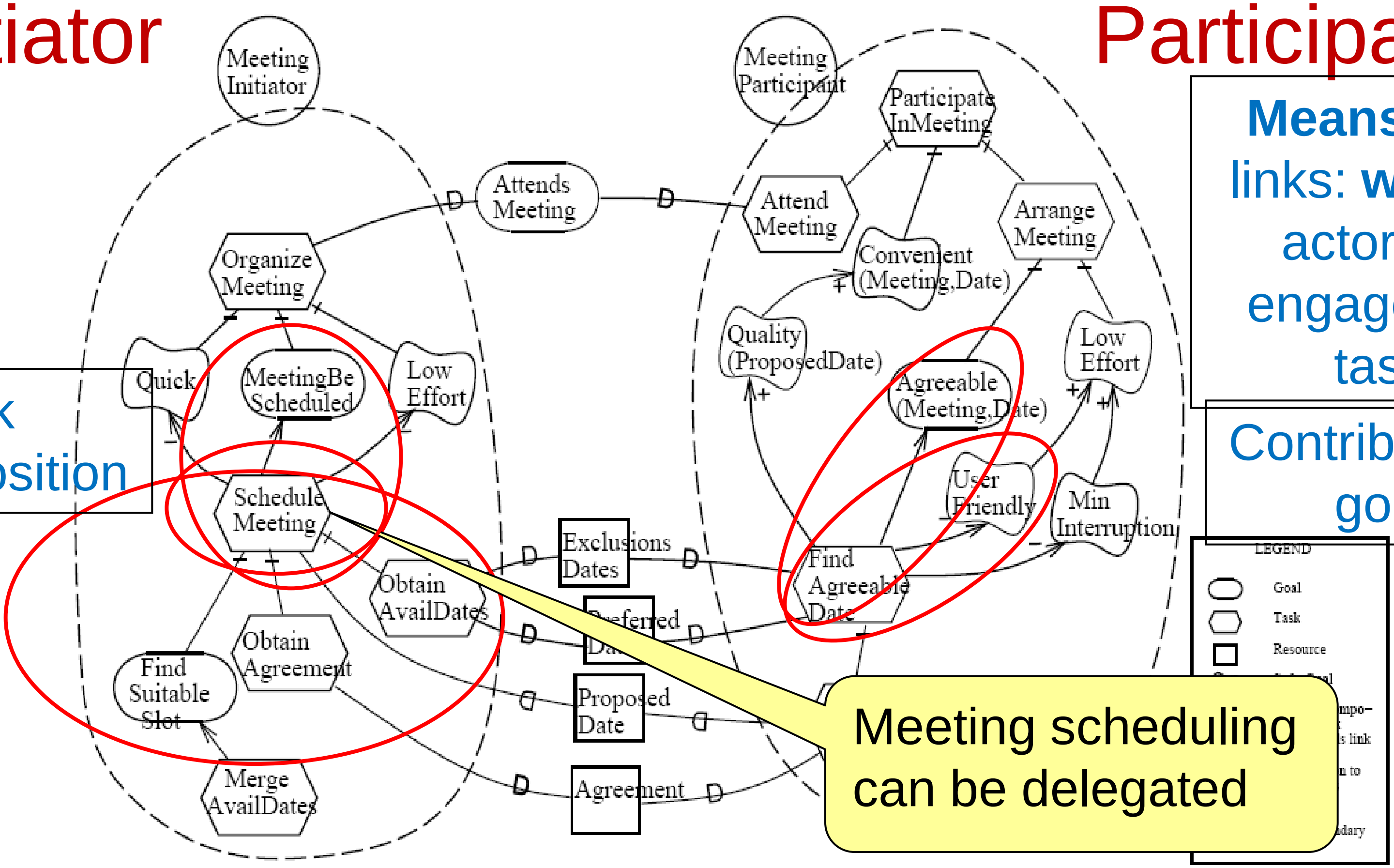
Meeting scheduling delegated

Figure 2: Strategic Dependency model for meeting scheduling with computer-based scheduler

A01: Strategic Relationship Model (1)

- **Strategic Rationale Model (SR)**
  - “Blowing up” the actor or looking “inside” the actor, to model internal intentional relationships.
  - Allows modelling of stakeholder interests and rationales.
  - Show different goals of each actor.
  - Different relationships between goals
    - Contribution (+, -), means-goal hierarchy, decomposition

A01: Strategic Rationale Model



Initiator

Participant

Task decomposition

Means-end links: **why** an actor will engage in a task

Contribution to goals

Meeting scheduling can be delegated

Figure 3: A Strategic Rationale model for meeting scheduling, before considering computer-based meeting scheduler

A04: Main Concepts in GRL

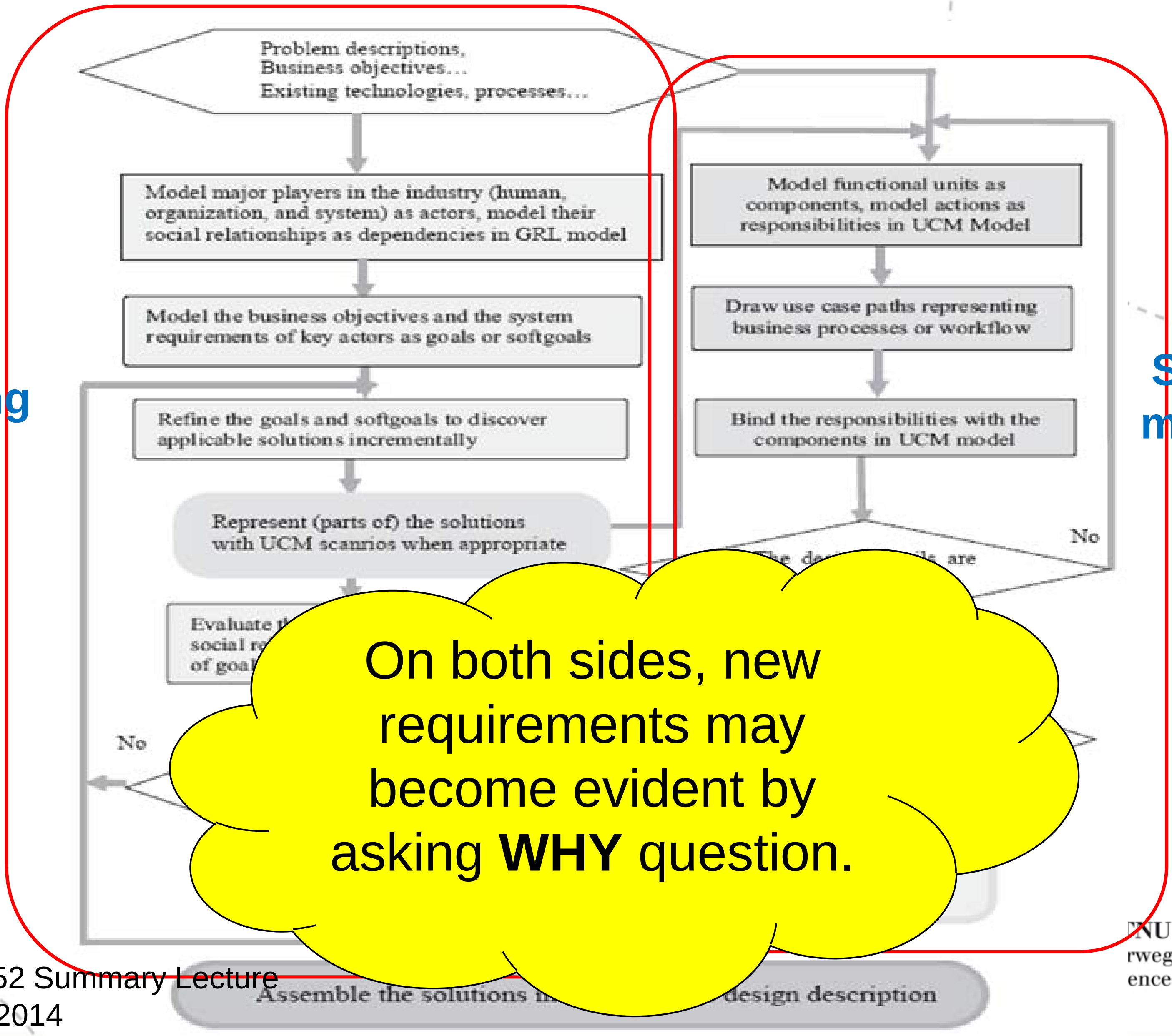
- **Goal**: to depict business objectives and system requirements (functional and non-functional).
- **Tasks**: to represent different ways to achieve goals.
- **Means-end reasoning**: to explore alternative solutions.
- **Social context**: modelled in terms of dependency relationships among the agents.

A04: i* & UCM

- Goal and scenario modelling can be done in parallel:
  - **Goal-modeling** – identification of alternatives and trade-offs in requirements engineering.
  - **Scenario-modeling** – snapshots of possible design solutions or fragments of a solution (partial and incomplete).
- Interaction between the modelling processes:
  - Design-alternatives in the goal modeling is explored in scenarios in UCM.
  - New goals might be elicited with ”why”-questions in relation to UCM.

A02: Process

GRL modelling



Problem descriptions,
Business objectives…
Existing technologies, processes…

Model major players in the industry (human, organization, and system) as actors, model their social relationships as dependencies in GRL model

Model the business objectives and the system requirements of key actors as goals or softgoals

Refine the goals and softgoals to discover applicable solutions incrementally

Represent (parts of) the solutions with UCM scanrios when appropriate

Evaluate t… social re… of goal…

No

Model functional units as components, model actions as responsibilities in UCM Model

Draw use case paths representing business processes or workflow

Bind the responsibilities with the components in UCM model

No

The de… …ls are

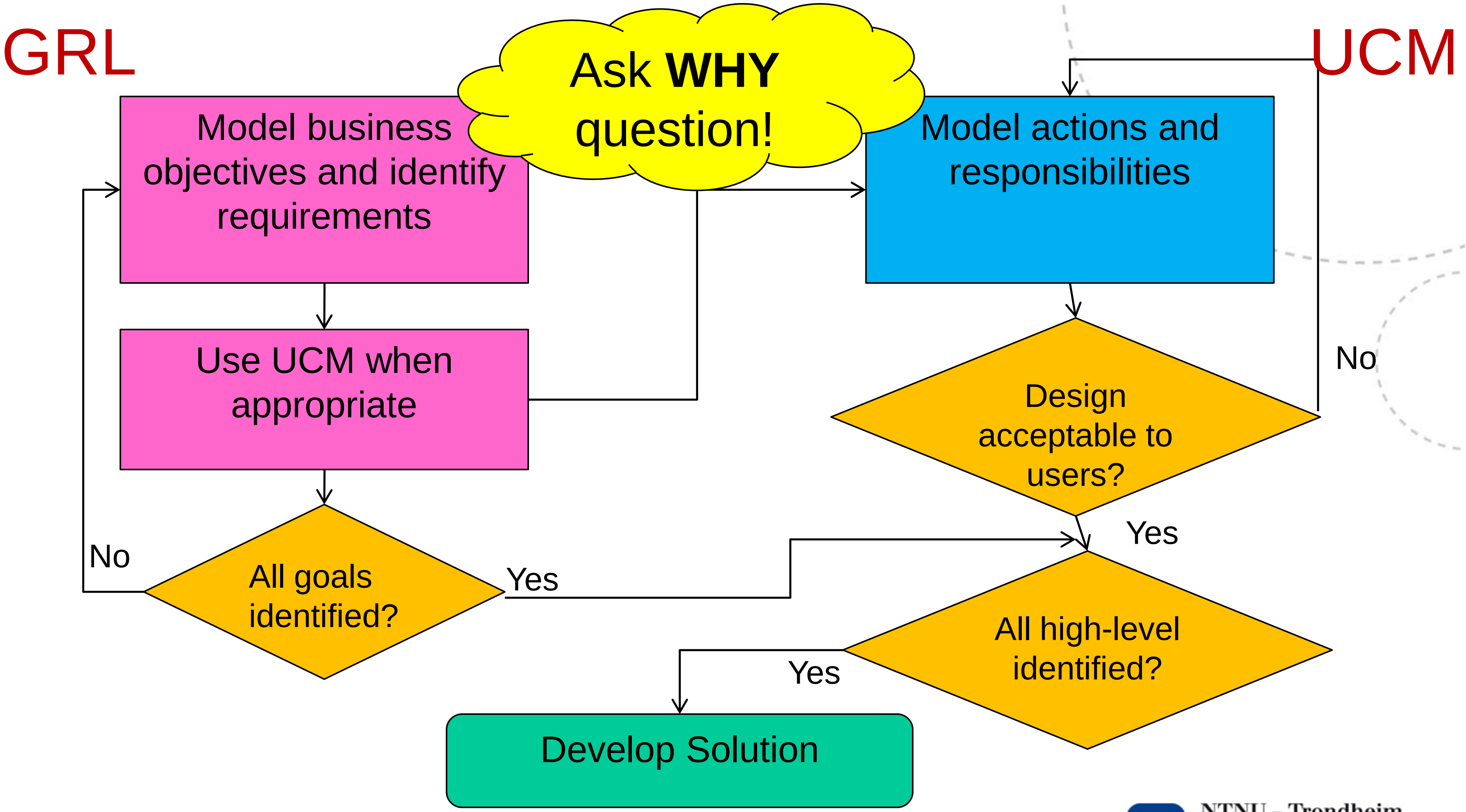
Assemble the solutions in… …design description

On both sides, new requirements may become evident by asking **WHY** question.

Scenario modelling (UCM)

…NU – Trondheim
…rwegian University of
…ence and Technology

GRL and UCM: Parallel & Interactions


GRL
UCM
Ask **WHY** question!
Model business objectives and identify requirements
Use UCM when appropriate
All goals identified?
No
Yes
Model actions and responsibilities
Design acceptable to users?
No
Yes
All high-level identified?
Yes
Develop Solution

A02: UCM Central Concepts

- Central concepts
  - Start points (preconditions, causes)
  - End points (postconditions, effects)
  - Responsibilities (tasks to be performed)
  - Components (objects in the system)
  - Use case path: connect start points, responsibilities and end points
  - Decomposition
  - Control-flow: OR-join. OR-fork, AND-join, AND-fork, timer, abort, failure points, shared responsibilities

i* & UCM

i*/GRL:
Goal structure
& alternatives

UCM: Scenario

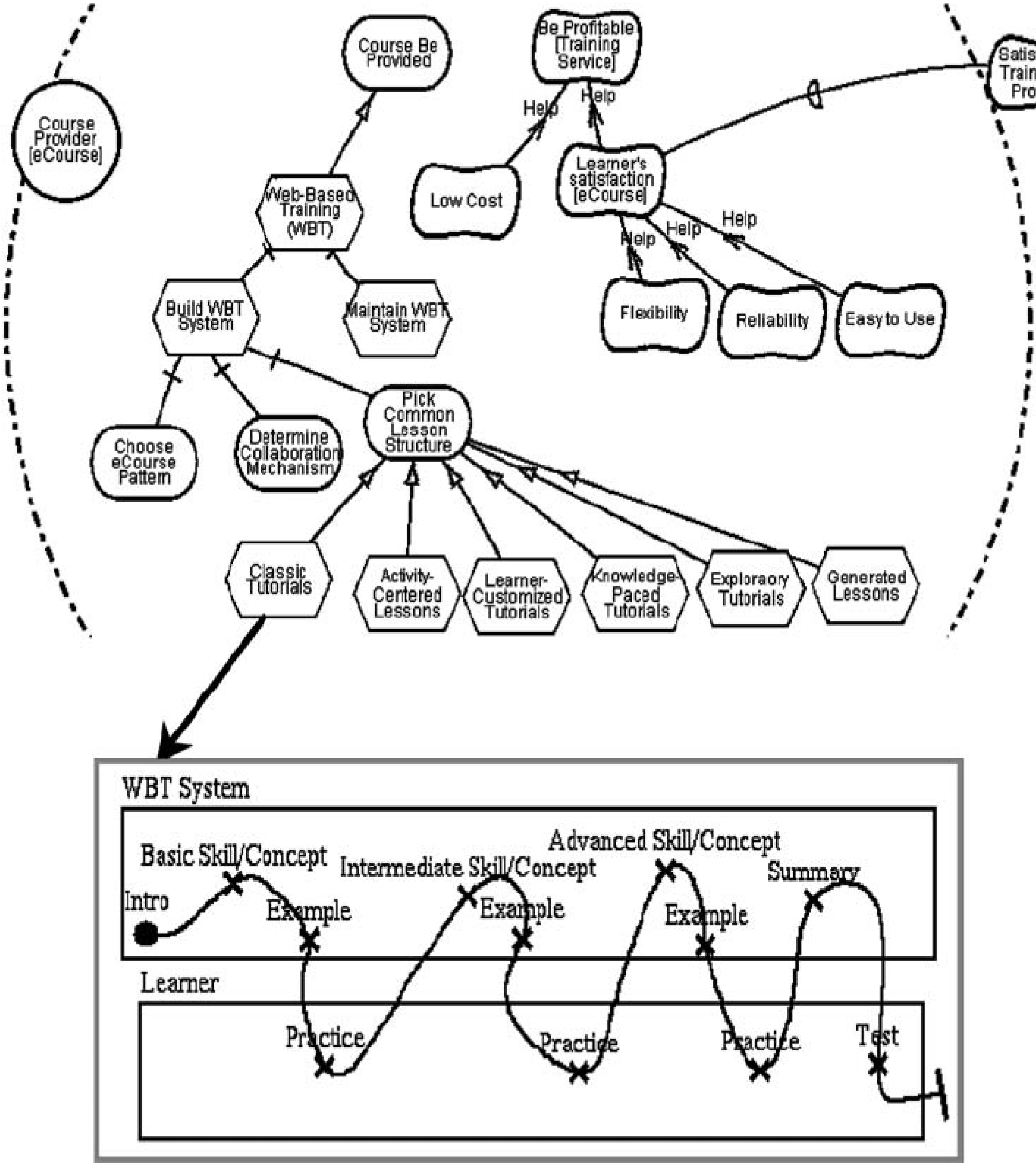


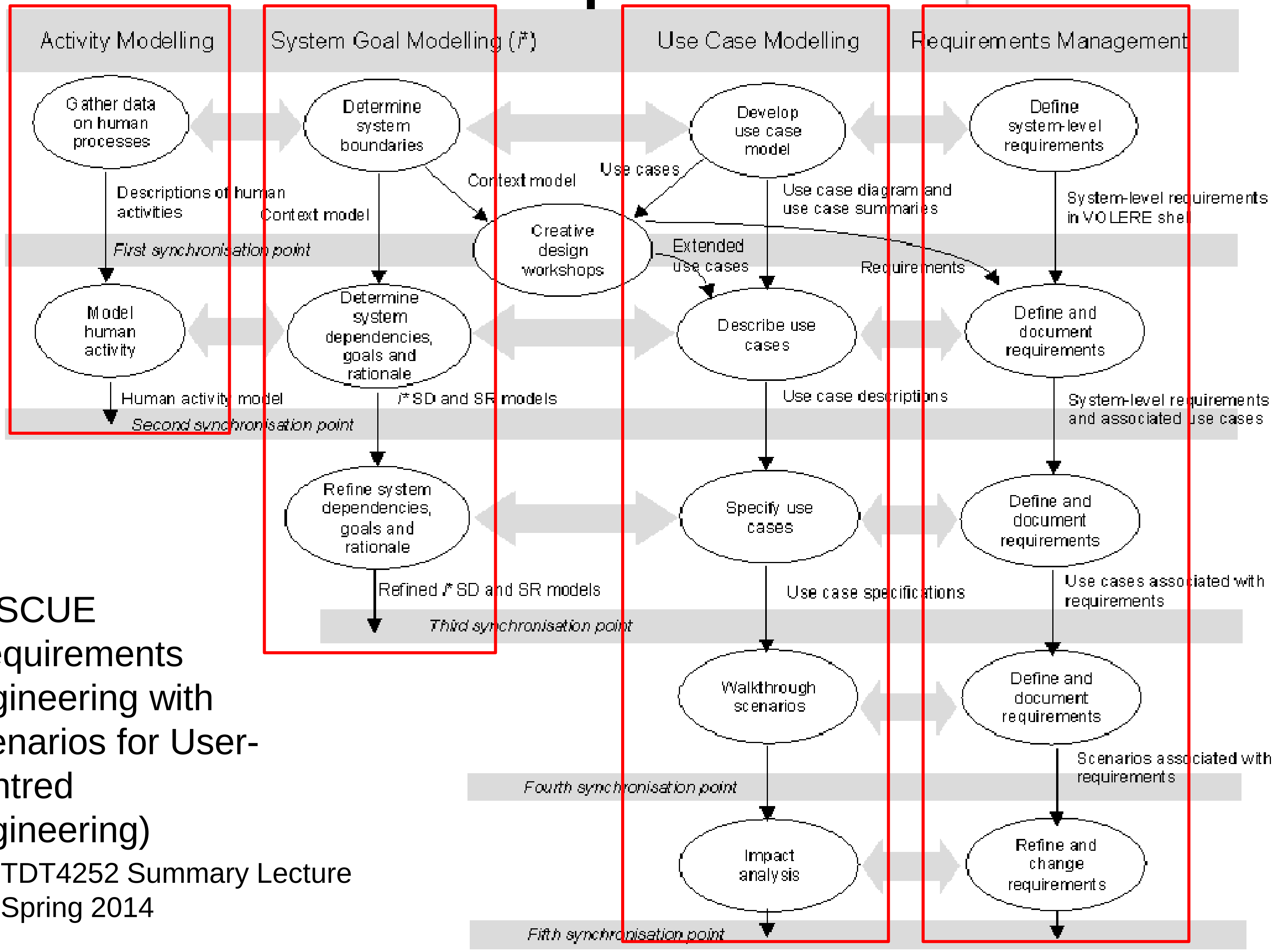
Fig. 9. Design alternatives and the corresponding scenarios.

A03: Maiden et al. Model-driven Requirements Engineering: Synchronising Models in an Air Traffic Management Case Study

A03 (Maiden et al)

- i* and other techniques used in a large project: Air traffic Management (ATM)
- Motivation:
  - Large socio-technical systems needs to be analyzed from a number of perspectives.
  - Need to use different modelling techniques.
  - Scalability of techniques.
  - It must be possible to synchronize the models
    - Conflicts, omissions, ambiguities (semantic quality of the overall model)
    - Compare to the need for synchronization of other models (requirements, design, code)
- The paper present an overall method RESCUE (Requirements Engineering with Scenarios for User-Centred Engineering)
  - Usage of several modelling techniques in concert
    - Use cases, i*, human activity modeling, requirements statements

The RESCUE process


Activity Modelling
System Goal Modelling (i*)
Use Case Modelling
Requirements Management
Gather data on human processes
Determine system boundaries
Develop use case model
Define system-level requirements
Descriptions of human activities
Context model
Context model
Use cases
Use case diagram and use case summaries
System-level requirements in VOLERE shel
Creative design workshops
First synchronisation point
Extended use cases
Requirements
Model human activity
Determine system dependencies, goals and rationale
Describe use cases
Define and document requirements
Human activity model
i* SD and SR models
Use case descriptions
System-level requirements and associated use cases
Second synchronisation point
Refine system dependencies, goals and rationale
Specify use cases
Define and document requirements
Refined i* SD and SR models
Use case specifications
Use cases associated with requirements
Third synchronisation point
Walkthrough scenarios
Define and document requirements
Scenarios associated with requirements
Fourth synchronisation point
Impact analysis
Refine and change requirements
Fifth synchronisation point

RESCUE (Requirements Engineering with Scenarios for User-Centred Engineering)

TDT4252 Summary Lecture
Spring 2014

Summary, from i* to other models (A01, A02, A03)

- i* is primarily for early RE/analysis
  - Actors, actor dependencies
  - Goal/task hierarchies
  - To understand the problem domain and the organization
  - Both for modeling as-is and to-be situations
- Can be connected to
  - ‘Prior’ informal models (Human Activity Model)
  - Later requirements specification (use cases/VOLERE requirements template)
  - Evaluation of design-alternatives (Use Case Maps)
- More experiences needed

Process Modelling: Overview

- Process Modelling: SeeMe and IDEF0

Based on the following articles:

- SeeMe in a Nutshell, Thomas Herrmann http://www.imtm-iaw.ruhr-uni-bochum.de/imperia/md/content/seeme/seeme_in_a_nutshell.pdf
- Menzel, Christopher, Mayer, Richard J. The IDEF Family of Languages. (pages 1-11 only) http://cmenzel.org/Papers/idef-family.pdf

Business Processes

- A business process is a collection of related, structured activities or tasks that produce a specific service or product (serve a particular goal) for a particular customer or customers. (Ref: Wikipedia)

Goal

- A business process is a sequence (or partially ordered set) of enterprise activities, execution of which is triggered by some event and will result in some observable or quantifiable result. (Ref: Vernadat, 1996)

Think as *processes* instead of *functions* and *procedures*!

History

- Flow charts
- Control flow diagrams
- Gantt Charts
- Pert charts
- SADT/IDEF
- UML (Unified Modelling Language)
- SeeMe
- EEML
- BPMN (Business Process Modelling Notation)

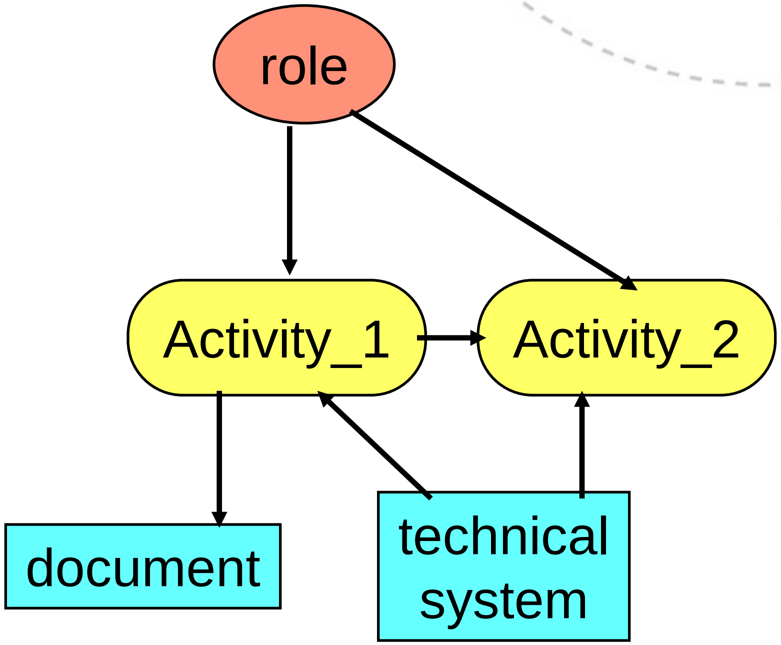
SeeMe Basics

Basic concepts/perspectives

Role – filled by single persons or ad-hoc or official groups like a departments. A role typically have a set of rights and responsibilities.

Activity – activities which are carried out by roles. They usually use entities or modify them.

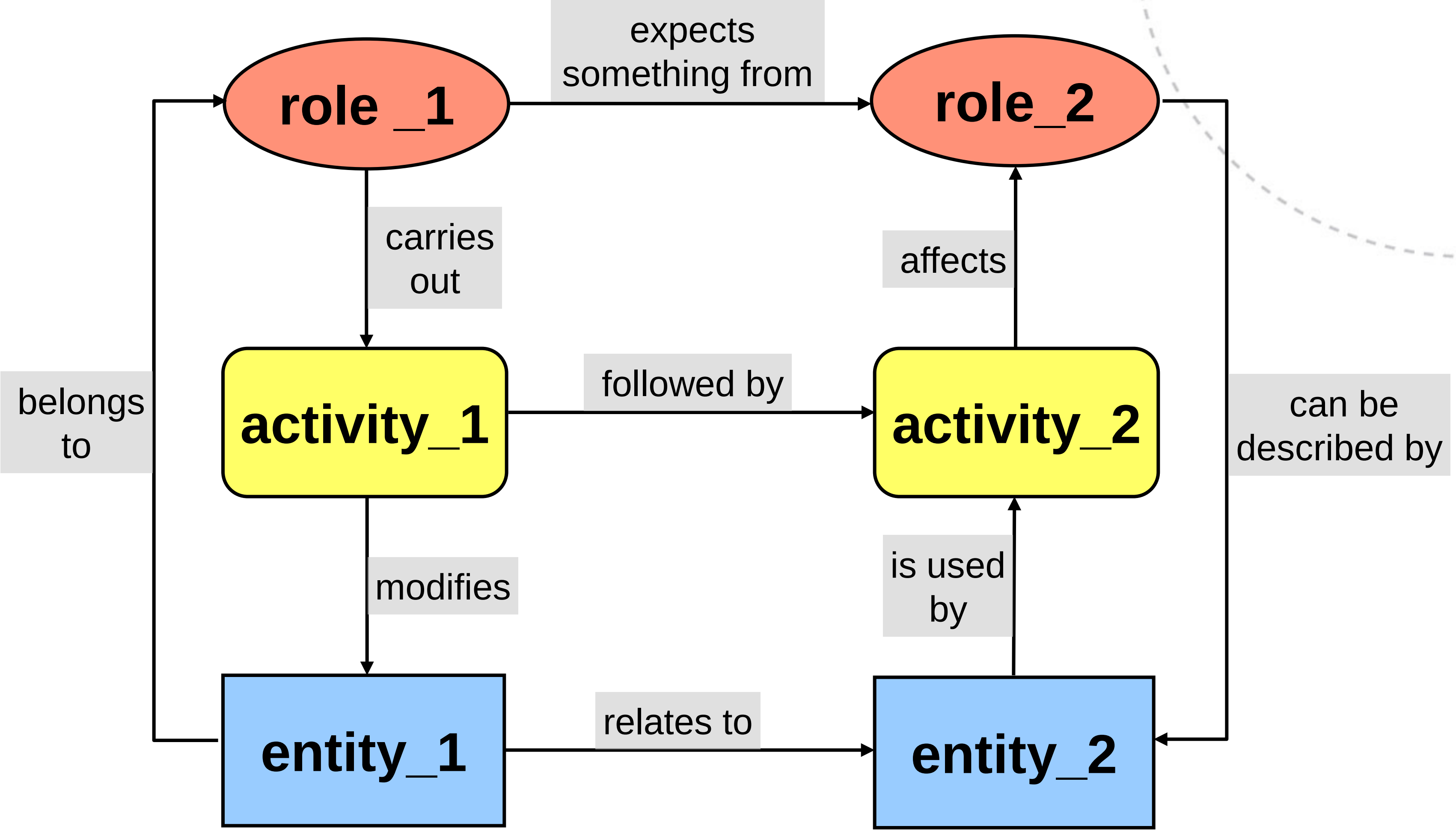
Entity – static aspects of the setting relevant for the process (e.g. data, systems)



Compare to Archimate (later):
Active (role), Behaviour (activity), Passive (entity)

SeeMe Basics

Relations

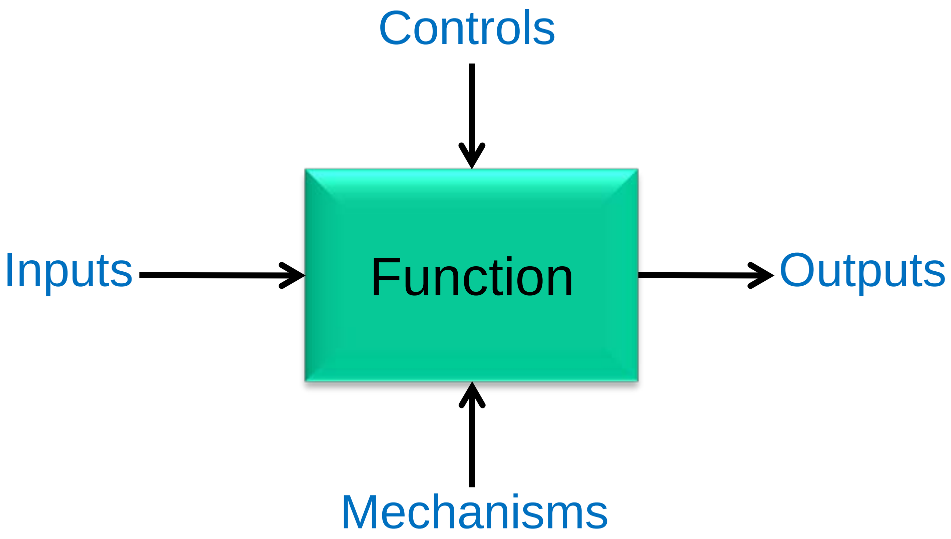


The standard definitions can be over-ridden by attaching alternative labels to the relations.

IDEF0: Syntax

- A model of a function at the highest level of inputs, outputs, controls and mechanisms.

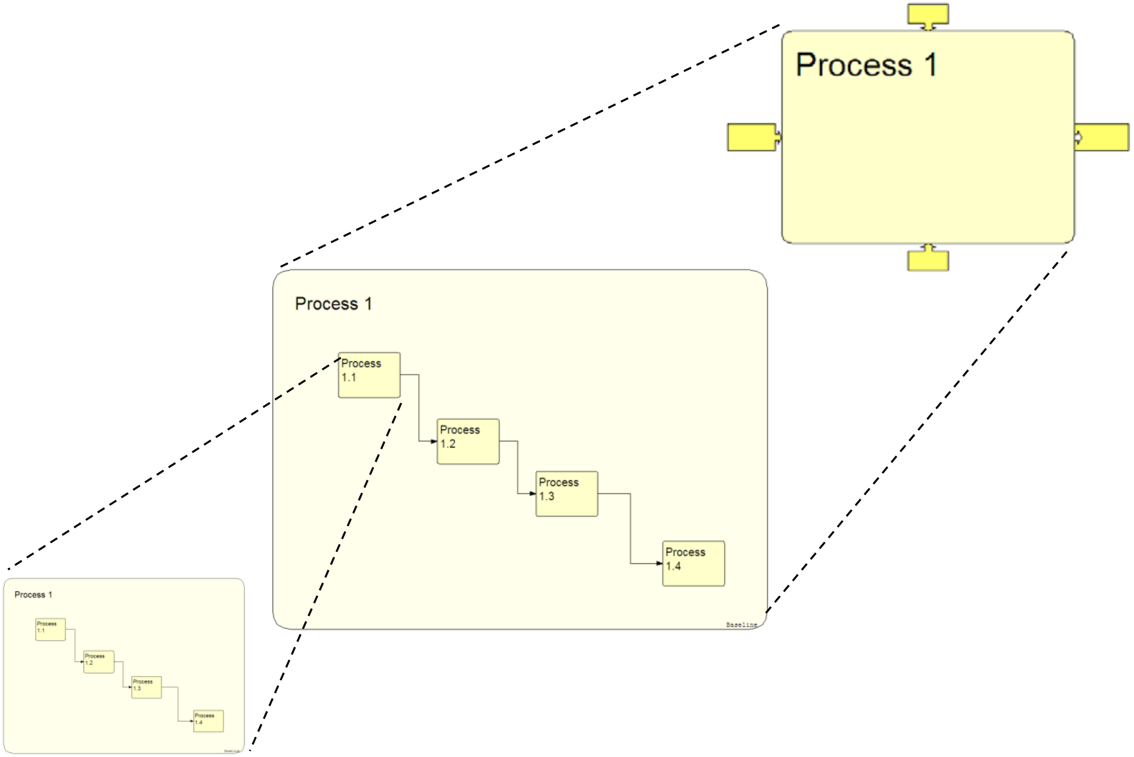
ICOMs



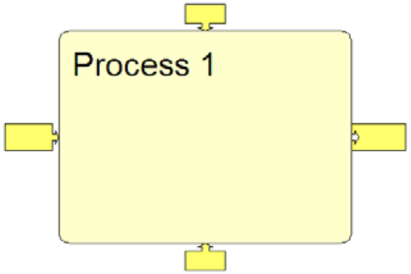
•Inputs: items that trigger or are transformed in the activity
•Controls: guide or regulate the activity
•Mechanisms: resources used to perform the activity
•Outputs: results of the activity or items processed or transformed

IDEF0: Decomposition

- The top level is called a context.



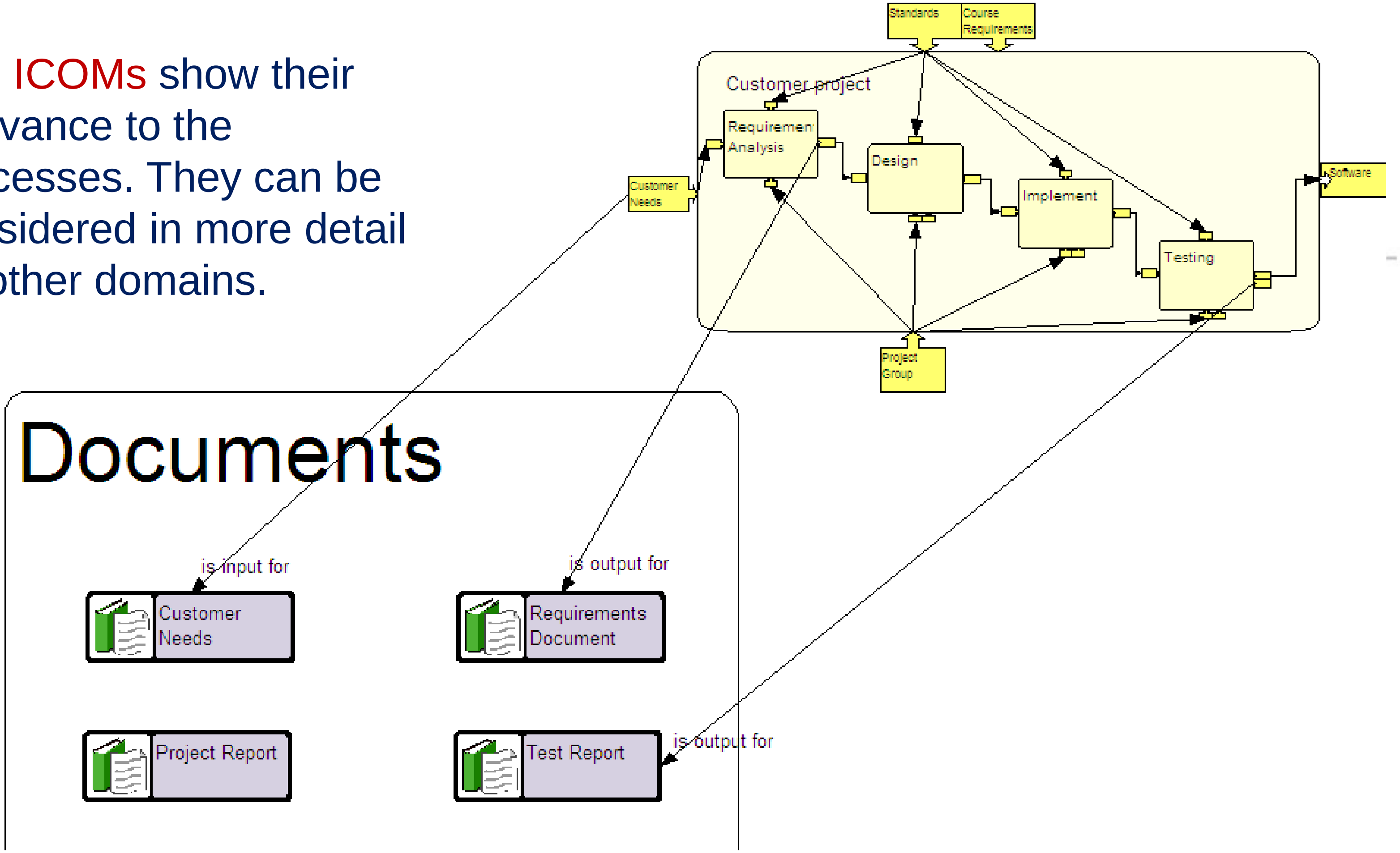
IDEF0: ICOMs



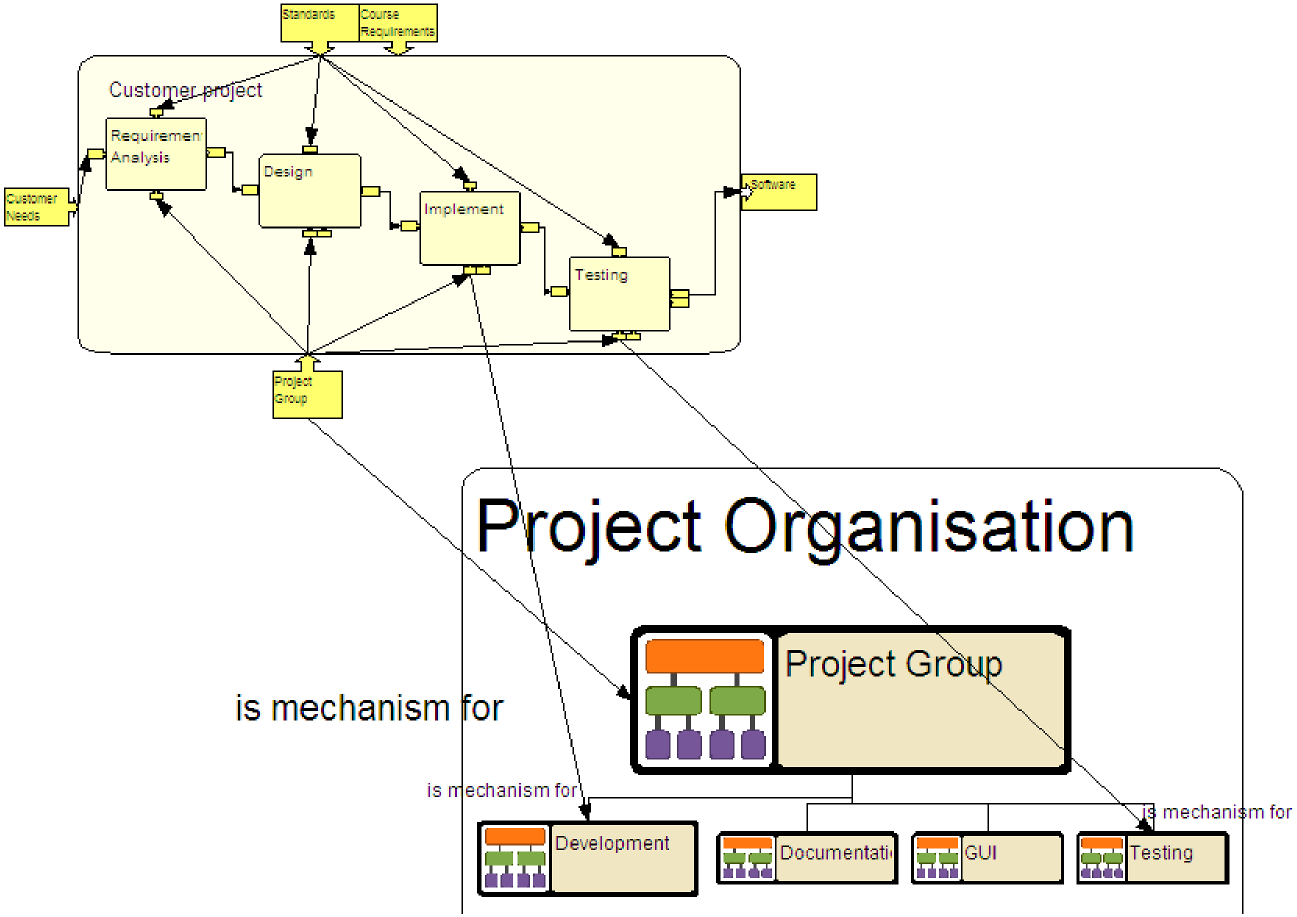
- Input:
  - Can be a trigger
  - Input that is transformed to output.
- Control
  - Guide or regulate activity
  - !!! Distinction between input and control: inputs change, controls remain unchaged.
- Mechanism: resources needed to perform activity
  - People
  - Equipment, IT
  - Financial resources
- Outputs
  - Results of a performing the activity

IDEF0 Model in Metis (1)

The ICOMs show their relevance to the processes. They can be considered in more detail as other domains.



IDEF0 Model in Metis (2)



Introduction to Enterprise Modelling

Based on the following articles:

A5: Enterprise Project: The Enterprise Ontology, http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html

A6: Fox, M. S. and Gruninger, M. 1998. Enterprise Modelling. AI Magazine, Fall.109-121.

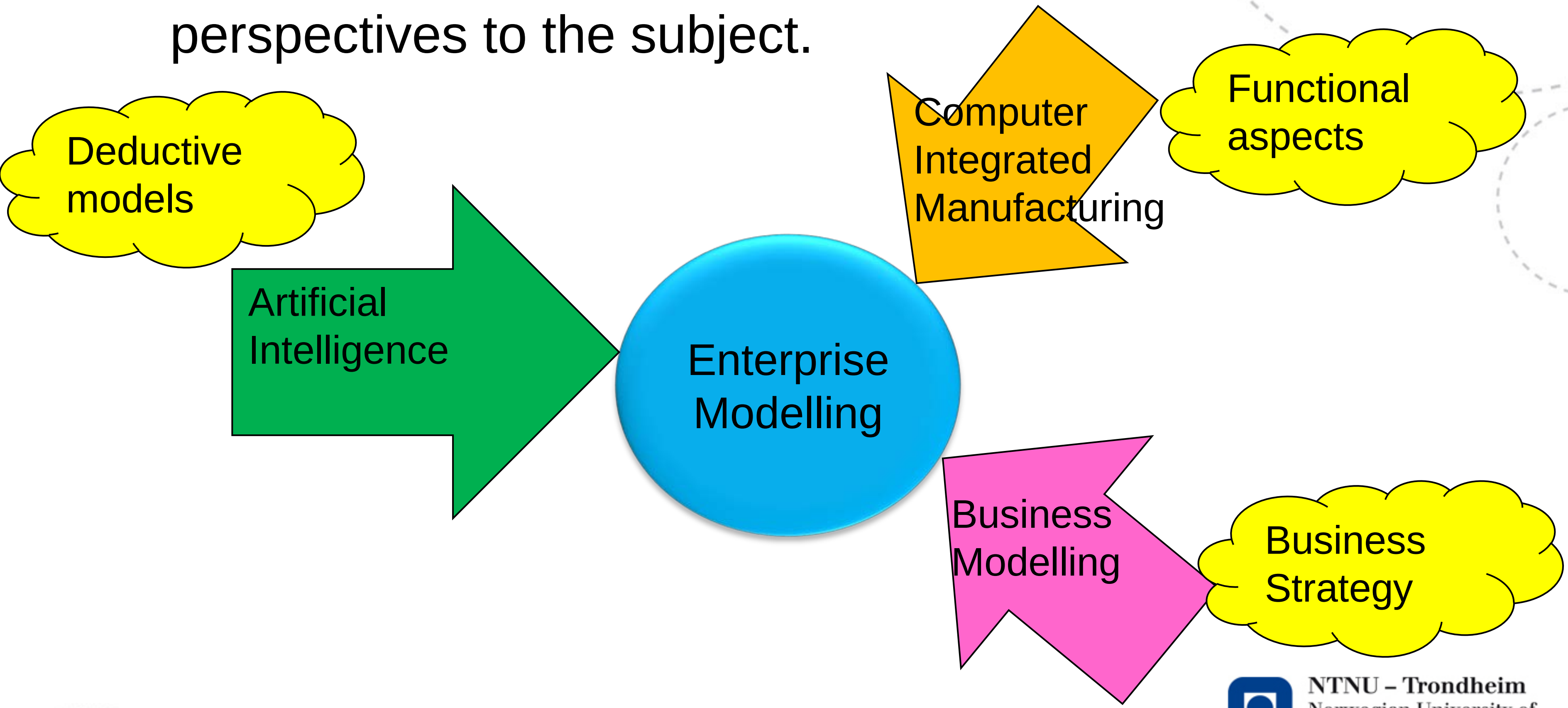
Additional Reading:

Vernadat, F. B. (1996), Chapter 3: Enterprise Modelling. Chapman and Hall, pp. 69-117. ISBN: 0 412 60550 3.

Lillehagen and Krogstie (2008), Chapter 4: State of the Art of Enterprise Modelling. Springer-Verlag, Berlin, Heidelberg. pp. 91-118.

Points of View

- Enterprise Modelling has been of interest to several research areas, and they all bring in different perspectives to the subject.



Enterprise Modelling: Definitions

- A computational representation of the structure, activities, processes, information, resources, people, behaviour, goals and constraints of a business, government, or other enterprise. (Fox & Gruninger)
- Enterprise modelling is the set of activities or processes used to develop the various parts of an enterprise model to address some desired modelling finality. (Vernadat)
- A collective name for the use of models in Enterprise Engineering and Enterprise Integration. (Bernus)

Enterprise Modelling: Purpose

Remember we said that modelling is not always for IT systems design…….

- To represent and understand how the enterprise works.
- To capitalise acquired knowledge and know-how for later use
- To rationalise and secure information.
- To (re)design and specify a part of an enterprise (functional behavioural, information, organisational or structural aspects).
- To analyse some aspects of the enterprise (economic analysis, oranisational, qualitative, etc.)
- To simulate the behaviour of some parts of the enterprise.
- To make better decisions about enterprise operations and organisation
- To control, coordinate or monitor some parts of the enterprise.

It's all about the ENTERPRISE, not just an IT system!

Ref: Vernadat

NTNU – Trondheim
Norwegian University of Science and Technology

Scope of Enterprise Modelling

- Enterprise Modelling is concerned with the what, how, when and who aspects of an enterprise.
  - What – operations performed.
  - How – defines the enterprise behaviour, the way things are done.
  - When – enforces the notion of time as being an essential component of the model.
  - Who – the resources or agents performing operations of the business process.
  - How much – economic aspects.
  - Where – logistics.
- Four basic aspects to be considered:
  - Functional, behavioural, information and organisational.

Ref: Vernadat

Role of an Enterprise Model

- Towards supporting model-driven enterprise design, analysis and operation.

Perspectives	Examples
Design	An Enterprise Model should provide the language used to define an enterprise. It should support the possibility to explore alternative designs or models. Need to reason about alternative designs, e.g. can a process be performed in different ways? Can a goal be achieved in a different way?
Analysis	Need to be able to detect the impact of changes. E.g. how will the purchase of a machine affect the activities? How will changing the processes affect the resource consumption?
Operation	The Enterprise Model must be able to represent what is planned, what might happen and what has happened. It must supply the information necessary to support the operations of an enterprise.

Enterprise Modelling: Challenges

- All enterprise functions do not share the same Enterprise Models – correspondence problem.
  - Each enterprise model may have the same concepts (e.g. activity), but they may use different names for them (operation vs. task).
  - Some translation of concepts is required for communication.
  - The concepts and representation lack adequate specifications of what the concepts (terminology) mean (semantics).
- Legacy systems – enterprises have their data in independent systems.
- Cost of designing, building and maintaining an enterprise model is huge.

GEM: Generic Enterprise Model

- An object library that defines the classes of objects that are generic across a type of enterprise, such as manufacturing or banking, and can be used (that is instantiated) in defining a specific enterprise.
- A GEM is composed of:
  - A set of object classes structured as a taxonomy, with subclass and super-class relationships.
  - For each object class, a set of relations linking it to other object classes, plus a definition of the intended meaning of the relation.
  - For each object class, a set of attributes plus a definition of the intended meaning of each attribute.

Benefits of GEM

- Predefined object library – allows the modeller to quickly move on to model instantiation.
- Path for growth – Benefit from the experience of others.
- Shared conceptualisation – by adopting a GEM the other parts of the organisation stand a greater chance of understanding what is represented in the Enterprise Model.

Commonsense Enterprise Models (1)

- The usefulness of an instantiated GEM is determined by the functions it can support, e.g. scheduling, forecasting, etc.
- We should be able to query the model and obtain answers to support the organisation.
- Where does the GEM end and inference begin?
- Three types of queries:
  - Factual (direct retrieval of information, surface-level processing, e.g. relational database).
  - Expert (requires that the information system has extensive knowledge, deep-level processing, supports some reasoning).
  - Common Sense – requires that the information system be able to deduce answers to questions that one would normally assume can be answered if one has common sense understanding of the enterprise.

Commonsense Enterprise Models (2)

- Commonsense queries:
  - Ability to answer queries as though the model has a common sense understanding of the enterprise.
  - Such an understanding often represents knowledge about the enterprise acquired over a relative short time and does not require expert knowledge.
  - Examples of such knowledge: organisational structure, roles, goals and resources would enable the deduction of what resources a person might allocate based on his/her role in the organisation.
  - We refer to it as shallow-level processing: retrieval that requires a small number of deductions to answer the query.
    - Requires a set of rules of deduction, axioms.

Benefits of Enterprise Modelling

- To build a common enterprise culture and shared vision to be communicated through the enterprise via the model, used as a common language.
- To capitalise enterprise knowledge and know-how to build an enterprise memory, which becomes a part of the enterprise assets.
- To support decision making concerning enterprise improvement or control.

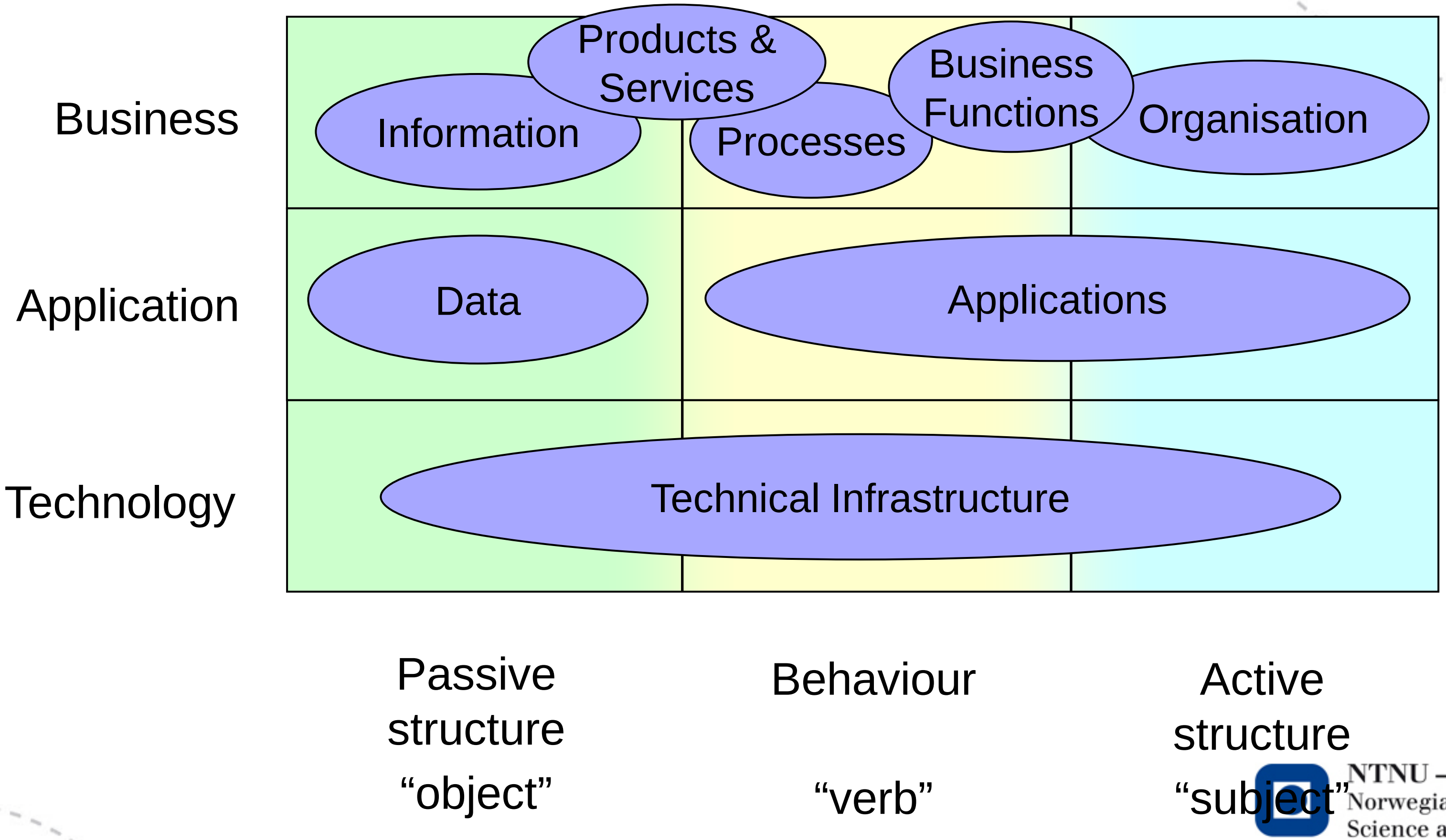
Ref: Vernadat

Main Traits of ArchiMate

- *A Lean* language:
  - just enough concepts, not bloated to include everything possible
  - 80/20 rule
- Well-founded concepts & models give *precision*
  - clear communication about architectures
  - get away from the ‘fuzzy pictures’ image
- Links to *existing approaches*
  - UML, BPMN, TOGAF
- International *vendor-independent standard*
  - The Open Group
- *Tool support*
  - Many tools available

NTNU – Trondheim
Norwegian University of Science and Technology

Layers, aspects, and viewpoints in ArchiMate



Business

Application

Technology

Products & Services

Information

Processes

Business Functions

Organisation

Data

Applications

Technical Infrastructure

Passive structure
“object”

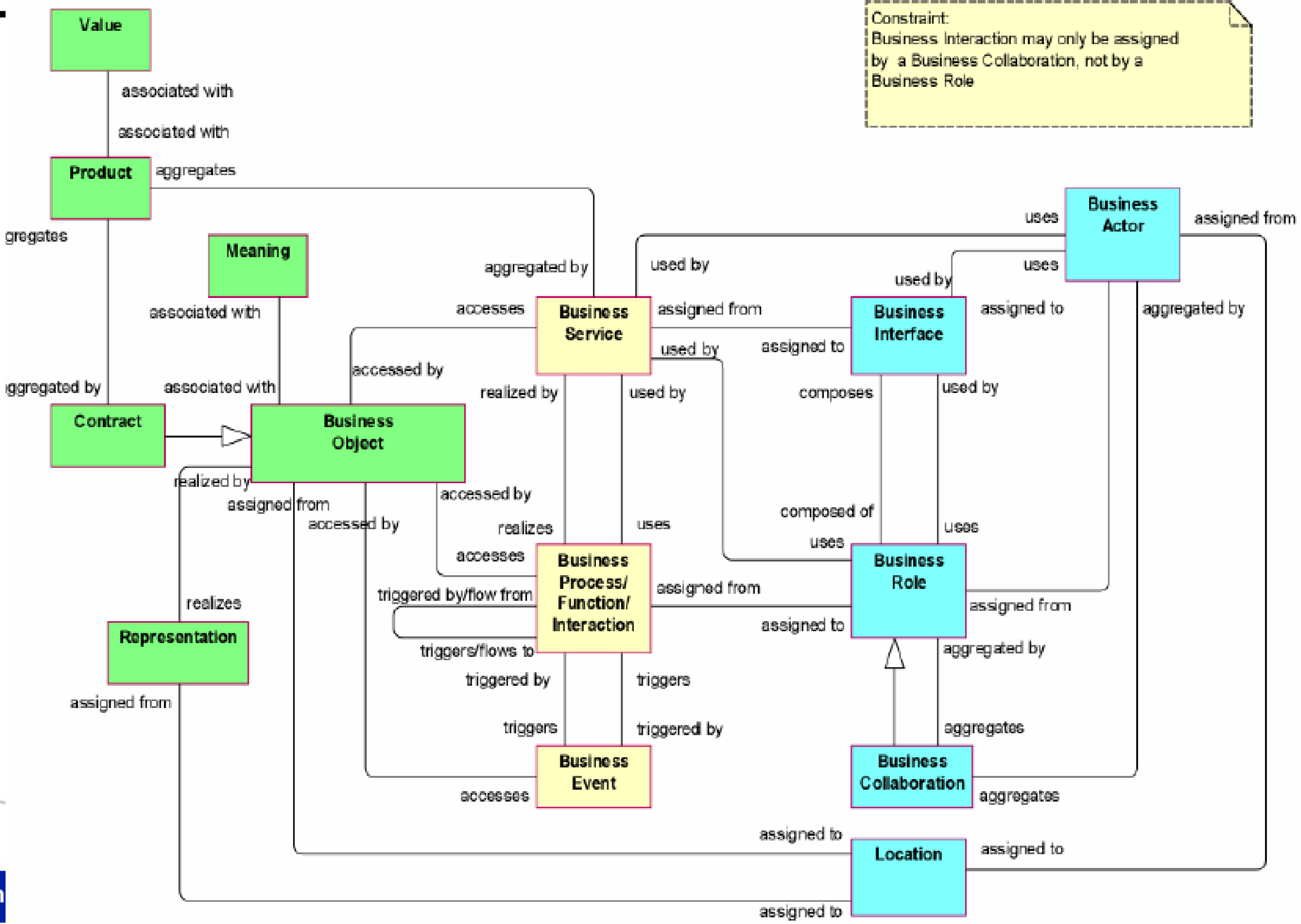
Behaviour
“verb”

Active structure
“subject”

Archimate aspects vs the perspectives to modelling

- Structural - Passive aspect. More detailed using e.g. UML class diagrams
- Functional - Behaviour aspect
- Behavioral - Only limited control flow, more detailed using e.g. BPMN
- Rule-oriented - Motivational extension
- Object-oriented - Not directly, more details using e.g. UML
- Social communication – Speech acts mentioned under passive structure (meaning)
- Actor/role-oriented – Active aspect
- Topological - Location (as part of the active aspect)

Business layer meta-model



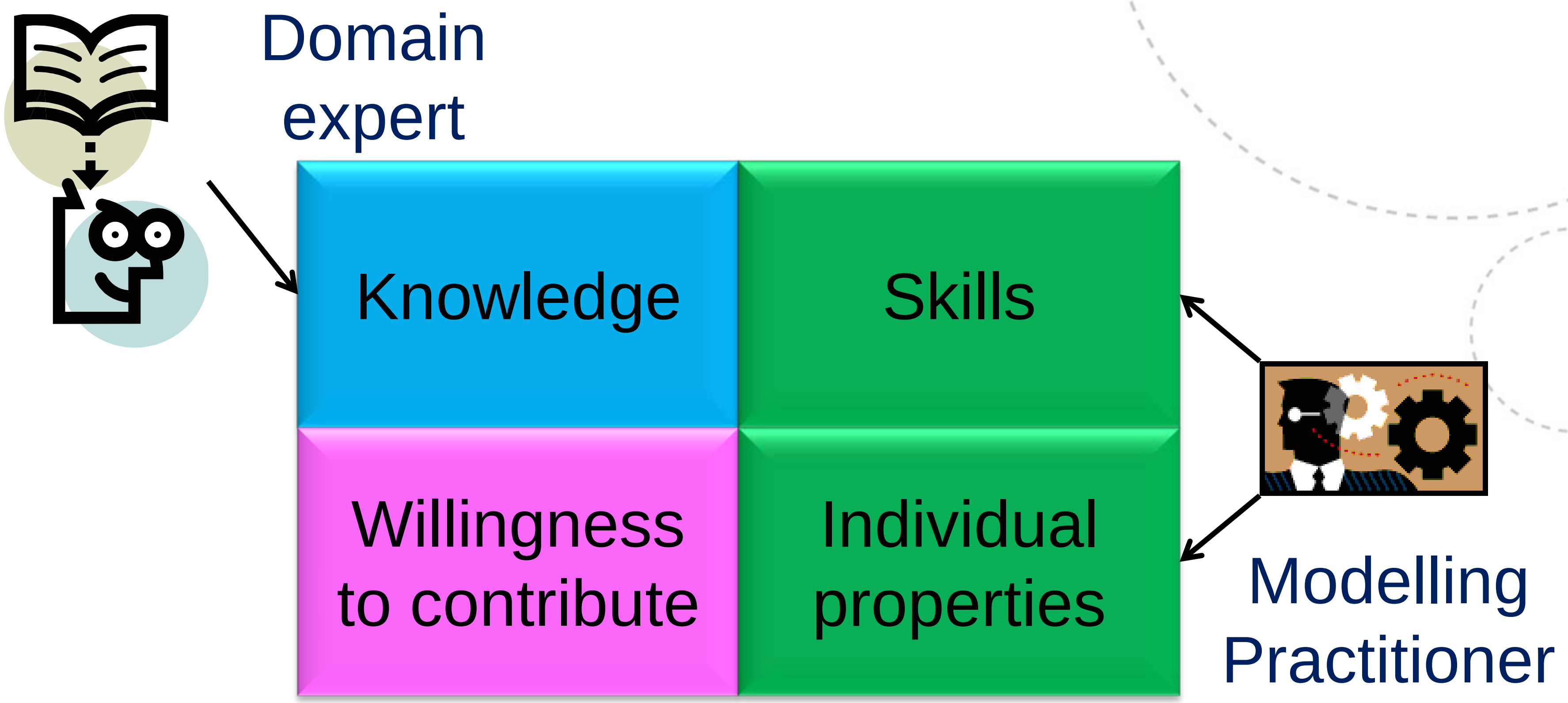
Constraint:
Business Interaction may only be assigned by a Business Collaboration, not by a Business Role

- Competences for Enterprise Modelling
  - Mainly chapter 2.4 in Krogstie: Model-based development of information systems (+ 2.1.4 and 2.1.5)
  - Additional material: A. Persson and J. Stirna, Towards Defining a Competence Profile for the Enterprise Modelling Practitioner. P. van Bommel et al. (eds.), PoEM 2010, LNBIP 68, pp. 232-245.

Methods

- There are several methods for Enterprise Modelling.
- Consultative approach: analysts create model; stakeholders are consulted in order to validate the model.
- Participatory approach: stakeholders meet in modelling sessions.
  - Groups led by a facilitator.
  - Participation is consensus-driven
  - Models created collaboratively.

Concept of Competency



Domain expert

Knowledge

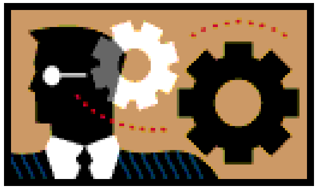
Skills

Willingness to contribute

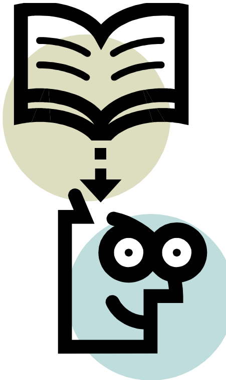
Individual properties

Modelling Practitioner

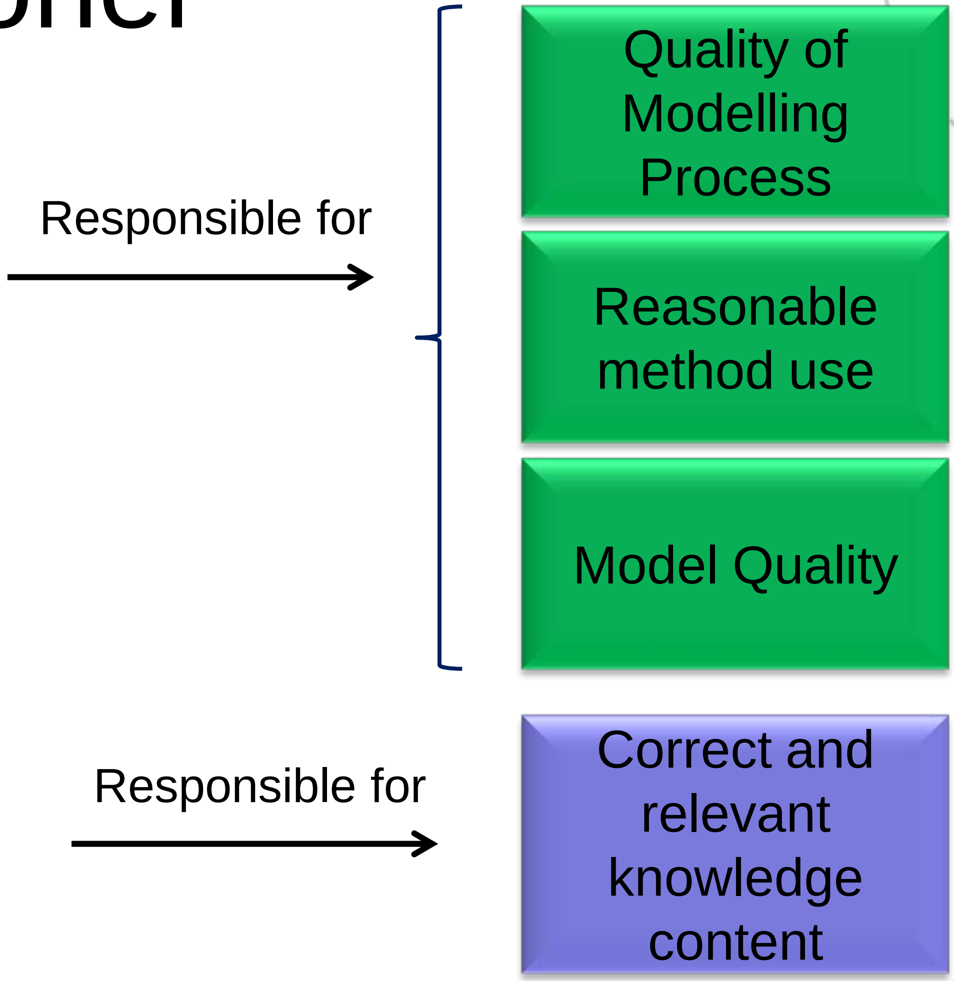
Domain expert and Modelling Practitioner



Modelling Practitioner



Domain expert



Responsible for

Quality of Modelling Process

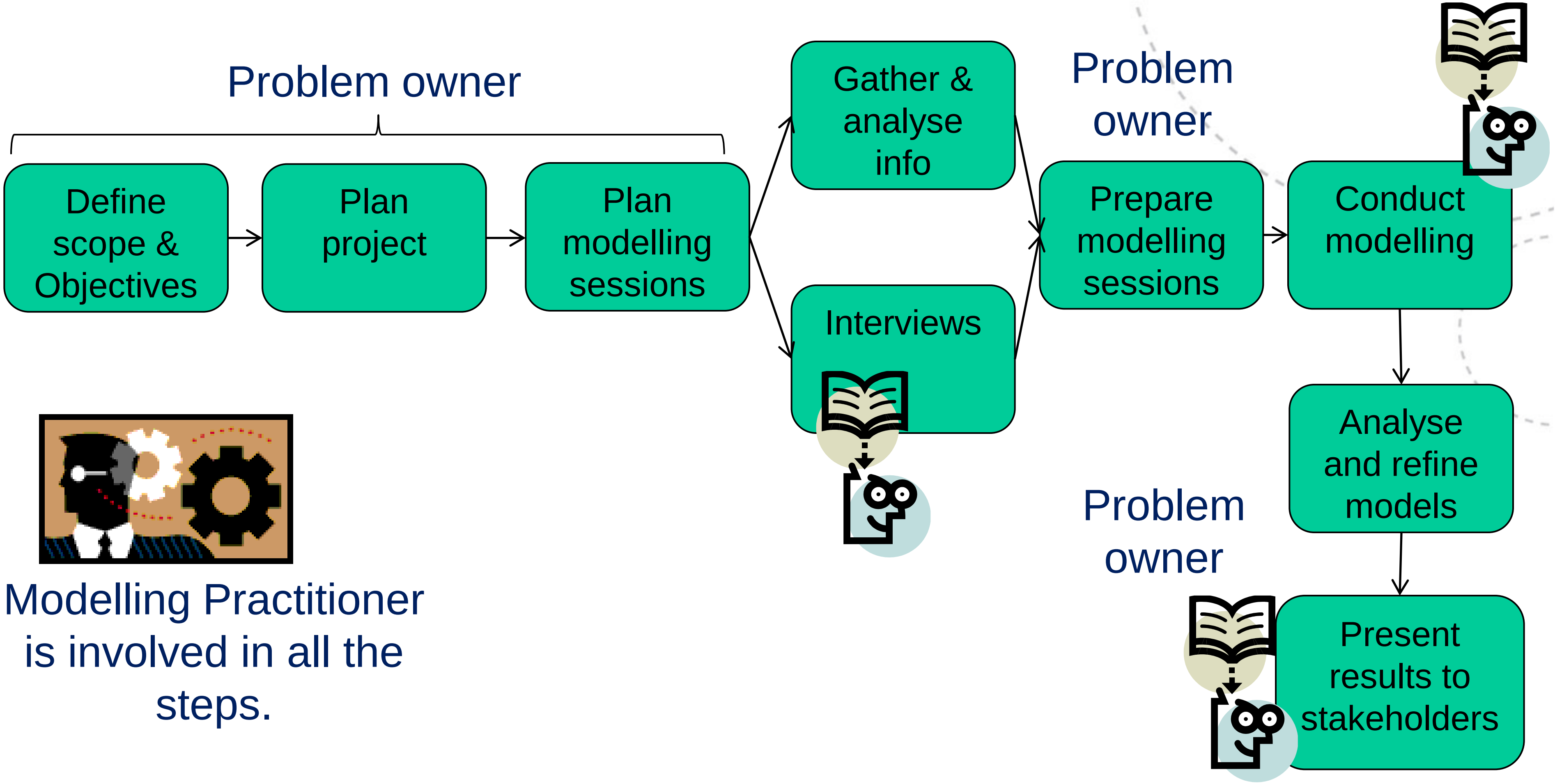
Reasonable method use

Model Quality

Responsible for

Correct and relevant knowledge content

Enterprise Modelling Process



Problem owner

Define scope & Objectives

Plan project

Plan modelling sessions

Gather & analyse info

Interviews

Problem owner

Prepare modelling sessions

Conduct modelling

Analyse and refine models

Problem owner

Present results to stakeholders

Modelling Practitioner is involved in all the steps.

Model Evaluation

- We have looked at several ways to evaluate models:
  - Competency questions: Can the model answer queries as though it has common sense about the enterprise? (Fox & Gruninger)
    - A8: S. A. Petersen (2003) “Using Competency Questions to Evaluate an Agent-based Model for Virtual Enterprises”.
  - Characteristics of Enterprise Models(Fox & Gruninger).
  - Principles of Enterprise Models (Vernadat)
  - SEQUAL (Krogstie)
    - A9 Krogstie, J. Using EEML for Combined Goal and Process Oriented Modeling: A Case Study EMMSAD 2008

A8: Evaluating Enterprise Models

Characteristic	Description
Functional completeness	Can the DEM represent the information necessary for a function to perform its task?
Generality	To what extent is the DEM shared between the diverse activities in the enterprise, e.g. finance and manufacturing?
Efficiency	Does the DEM support efficient reasoning?
Perspicuity	Is the DEM easily understood by users so that it can be applied consistently and interpreted across the enterprise?
Precision granularity	Is there a core set? Does the representation support reasoning at various levels of abstraction and detail?
Minimality	Does the DEM contain the minimum number of objects necessary?

Enterprise Modelling Principles (1)

- Principle of separation of concerns:
  - It would be unrealistic to consider an enterprise as a whole. It must be analysed piece by piece.
- Principle of functional decomposition:
  - Major functions structured as sub-functions.
- Principle of Modularity:
  - To facilitate management of change.
- Principle of Genericity:
  - Important to define standard building blocks as generic classes to factor common descriptive attributes and behaviours.
- Principle of reusability:
  - To reduce modelling efforts and increase modularity.

Ref: Vernadat

Enterprise Modelling Principles (2)

- Principle of separation of functionality and behaviour:
  - Functionality: things to be done; behaviour: the way things are done.
- Principle of process and resource decoupling:
  - Consider separately the things being done (processes) and the agents performing them (resources).
- Principle of conformity:
  - Deals with syntax and semantics of the model to really accurately represent what it’s supposed to model.

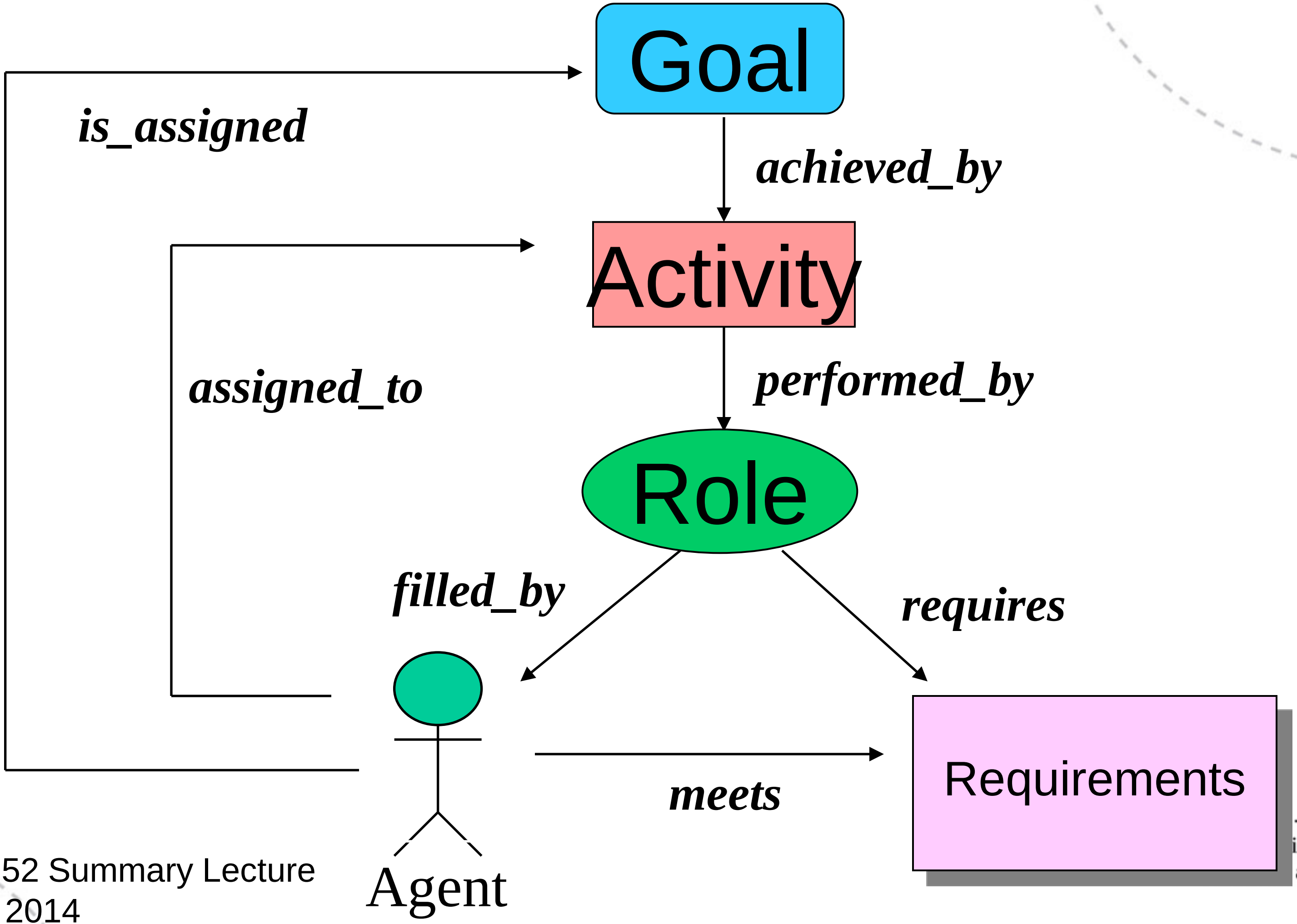
Ref: Vernadat

Enterprise Modelling Principles (3)

- Principle of model visualisation:
  - To easily communicate models, e,g. simple graphical formalisms.
- Principal of simplicity vs. adequacy:
  - The modelling language to be rich enough to express what needs to be expressed, yet simple enough so that users can easily learn to use it.
- Principle of management of complexity:
  - Must permit the representation of systems of great complexity.
- Principle of rigor of representation:
  - The model must neither be ambiguous nor redundant.
- Principle of separation of data and control:
  - To support real-time systems. The process is operated not by the availability of data, but by events.

Ref: Vernadat

Agent-based Model of a VE



Agent-based Model: Contents

Entity	Attributes	Relationship
Goal, e.g. Produce some pens	goal(goal_name, product_area, deadline, max_cost).	achieved_by(goal, activity).
Activity, e.g. Design pens	activity(name, start_date, completion_date).	performed_by(activity, role).
Agent, e.g. Michael	agent(name, address, goals, skills, availability).	filled_by(role, agent).

Example: What are the skills required to perform a particular activity?

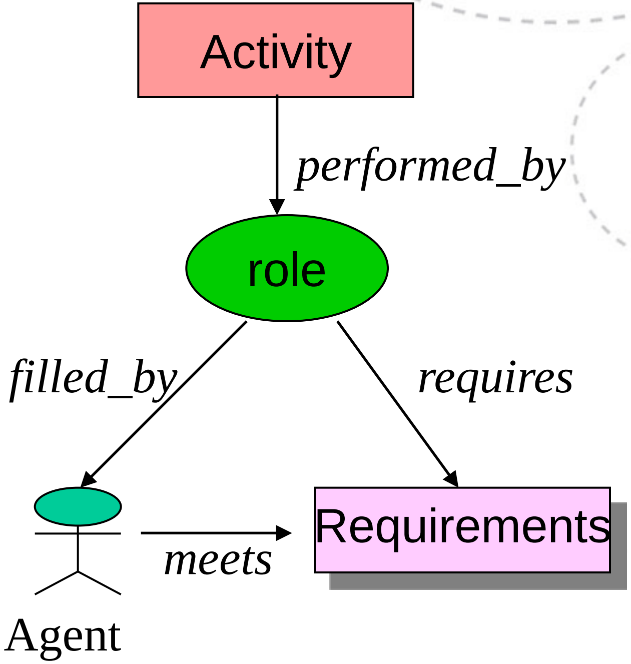
performed_by(activity, role).

filled_by(role, agent).

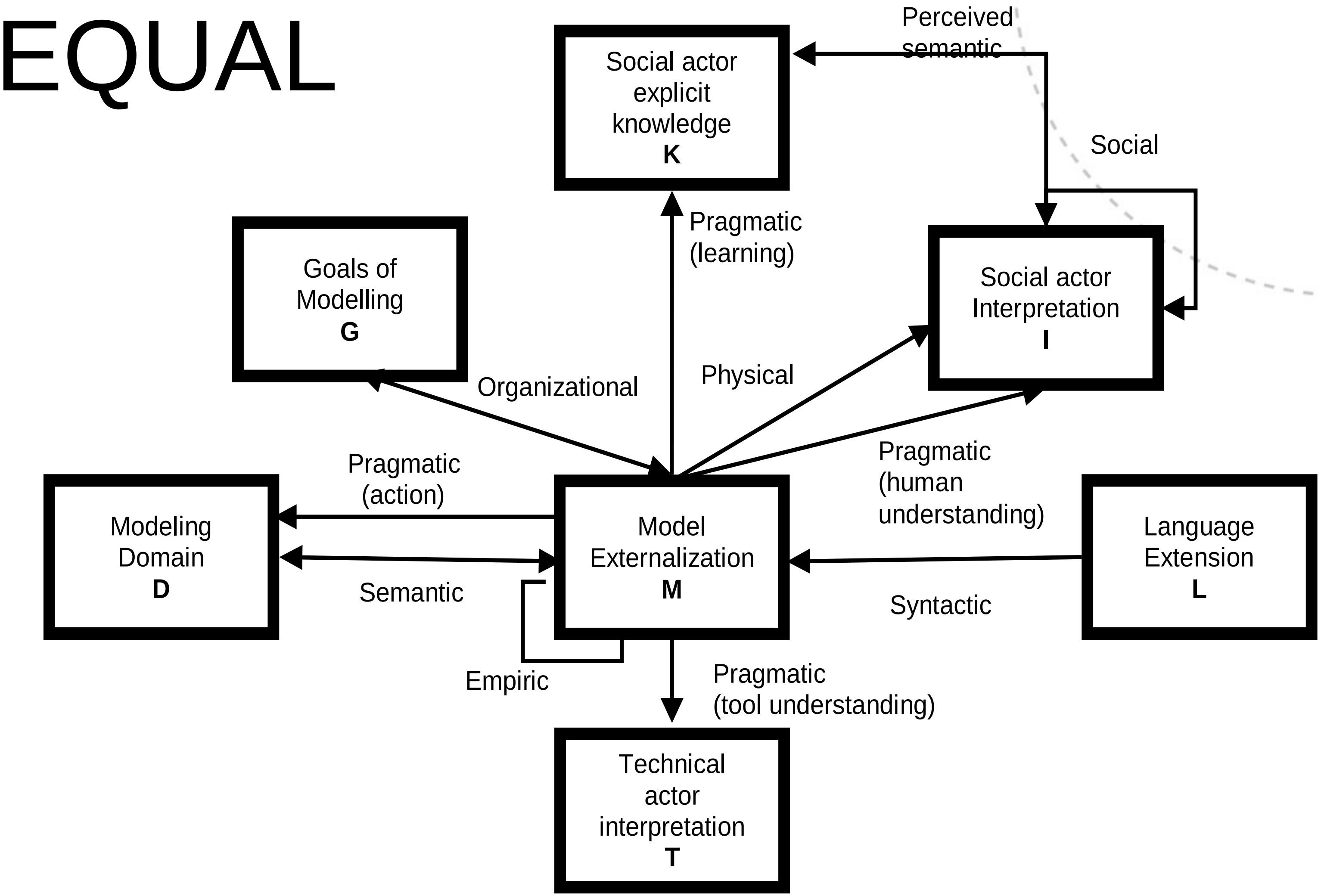
requires(role, requirements(Skills,
    availability, price)).

if (performed_by(activity, role) and
    requires(role, requirements(Skills,
    availability, price))

then (skills required for activity = Skills).



Evaluation framework – SEQUAL



Interoperability

- Definition:
  - Interoperability is a property of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, without any restricted access or implementation.
  - The ability of two or more systems or components to exchange information and to use the information that has been exchanged (IEEE).
- Initially defined for IT systems, considering exchange of information. But this is not enough.

Product Based Interoperability

- Based on Lillehagen/Krogstie Chapter 5

Interoperability

- Syntactic Interoperability:
  - If two or more systems are capable of communicating and exchanging data, they are exhibiting syntactic interoperability. Specified data formats, communication protocols and the like are fundamental. XML or SQL standards are among the tools of syntactic interoperability.
- Semantic Interoperability:
  - The ability to automatically interpret the information exchanged meaningfully and accurately in order to produce useful results as defined by the end users of both systems. To achieve semantic interoperability, both sides must refer to a common information exchange reference model.
- Business Interoperability:
  - The ability for diverse business processes to work together.

EA Bridges Strategy and Implementation

Architecture

- Business architecture
- Information architecture
- Solution architecture
- Technology architecture

Business Strategy

- Business drivers
- Business goals
- Business policy
- Trend analysis

Implementation

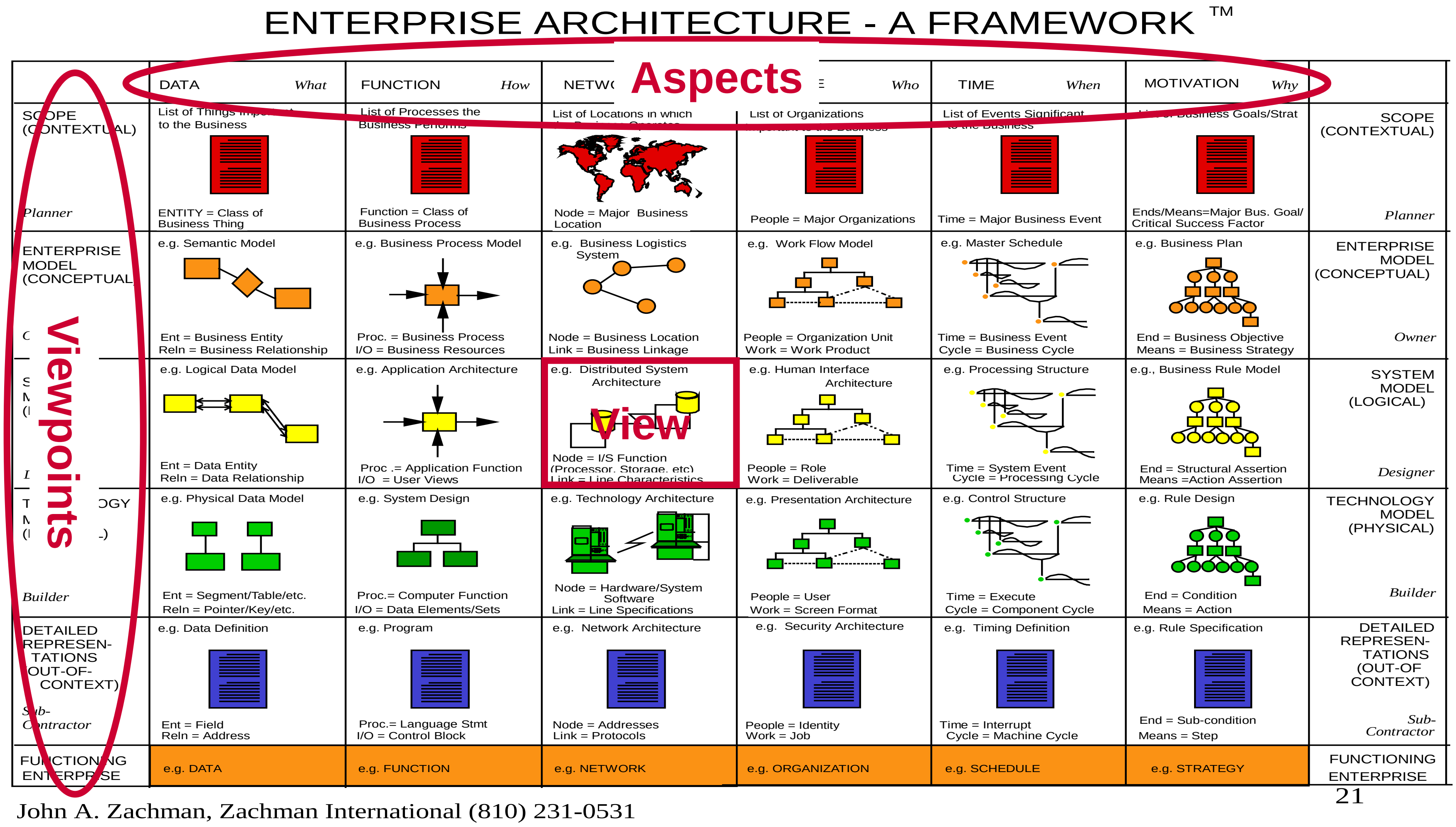
- Business processes
- Application systems
- Tech infrastructure
- Organizational structure

The bridge between strategy & implementation

Enterprise Architecture

- We have looked at the following:
  - Zachman -> Taxonomy
  - TOGAF -> Process
  - FEA -> Methodology
  - Gartner -> Practice
- How do these compare?

Zachman’s EA Framework



ENTERPRISE ARCHITECTURE - A FRAMEWORK ™

	DATA *What*	FUNCTION *How*	NETW…	…E *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Function = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organizations	List of Events Significant to the Business Time = Major Business Event	List of Business Goals/Strat Ends/Means=Major Bus. Goal/ Critical Success Factor	SCOPE (CONTEXTUAL) *Planner*
ENTERPRISE MODEL (CONCEPTUAL) C…	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy	ENTERPRISE MODEL (CONCEPTUAL) *Owner*
S… M… (I… *L…*	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. Application Architecture Proc .= Application Function I/O = User Views	e.g. Distributed System Architecture Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means =Action Assertion	SYSTEM MODEL (LOGICAL) *Designer*
T… …OGY M… (I… …-) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. System Design Proc.= Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture Node = Hardware/System Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action	TECHNOLOGY MODEL (PHYSICAL) *Builder*
DETAILED REPRESEN- TATIONS (OUT-OF- CONTEXT) *Sub- Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. Program Proc.= Language Stmt I/O = Control Block	e.g. Network Architecture Node = Addresses Link = Protocols	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN- TATIONS (OUT-OF- CONTEXT) *Sub- Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

21

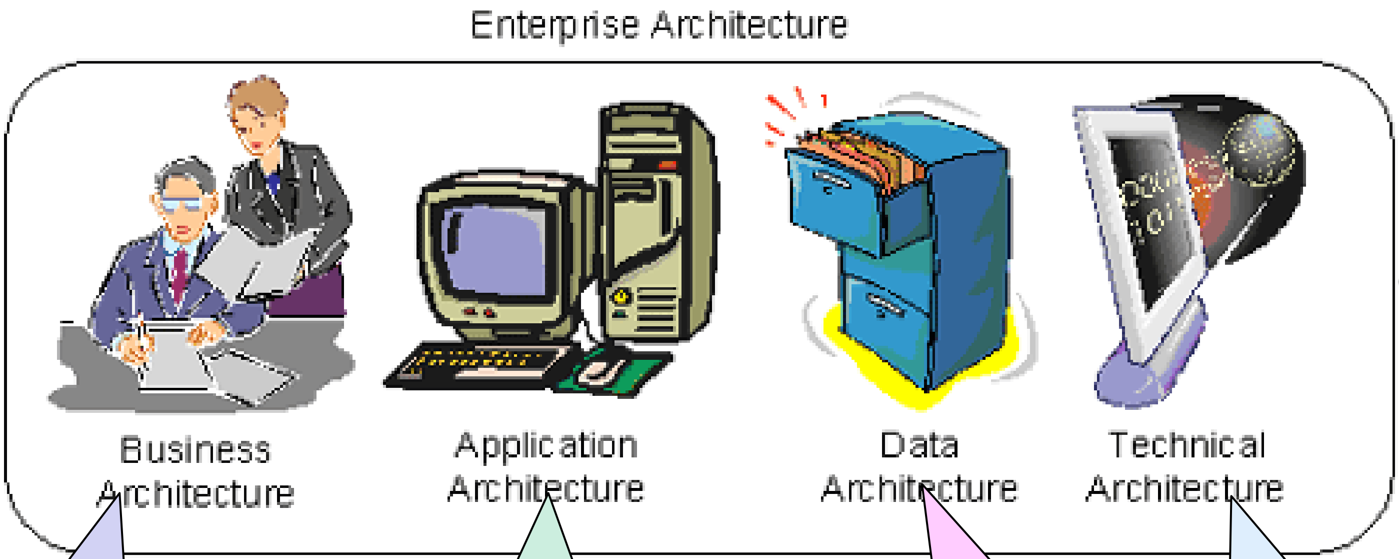
John A. Zachman, Zachman International (810) 231-0531

TOGAF – consists of

- An Architectural Development Method (ADM)
- Foundation Architecture
  - A Technical Reference Model (TRM)
  - A Standards Information Base (SIB)
  - Building Blocks Information (BBIB)
- Resource Base contains advice on:
  - Architecture views, IT Governance, Business scenarios, Architecture patterns, etc.

Greenslade, 2000-2002

TOGAF’s Enterprise Architecture



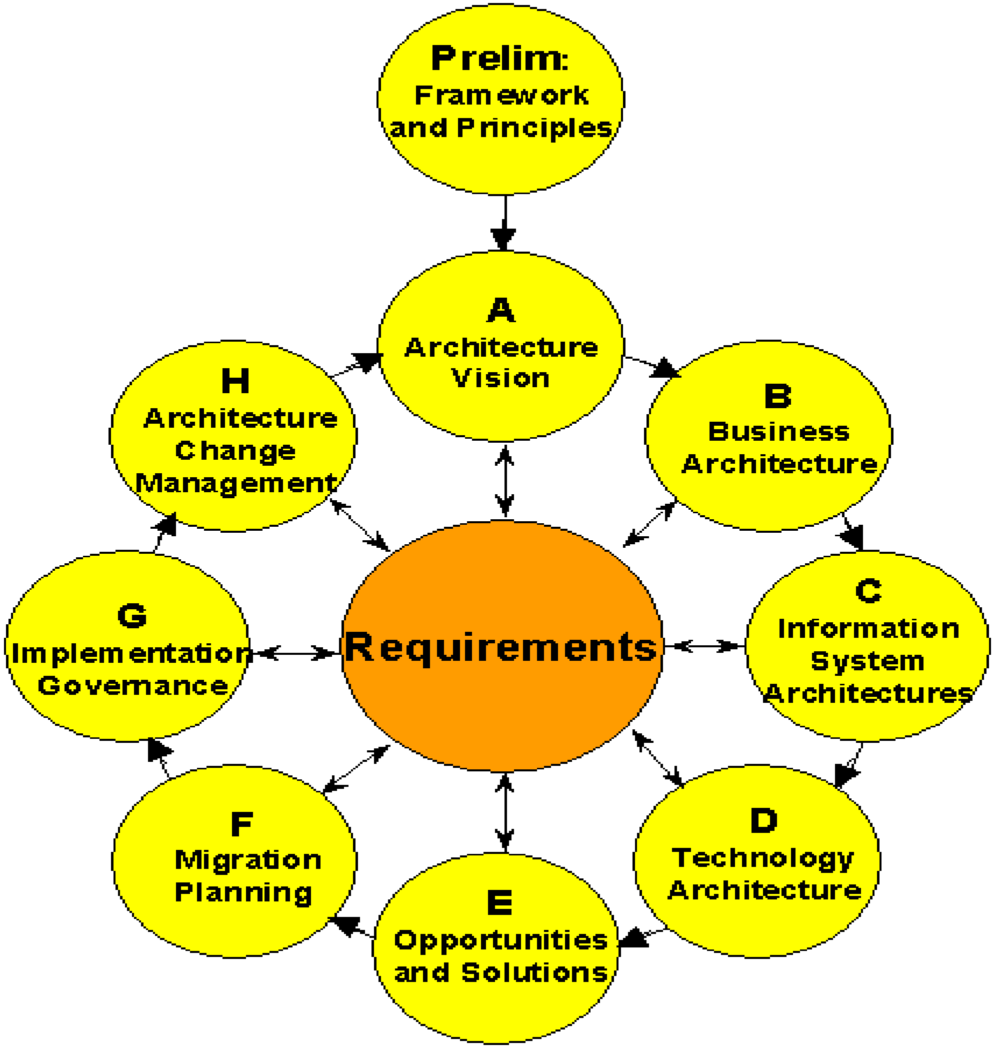
Describes the processes the business uses to meet its goals.

Describes how specific applications are designed and how they interact with each other.

Describes how the enterprise datastores are organised and accessed.

Describes the hardware and software infrastructure that supports applications and their interactions.

Architecture Development Cycle - ADM



Gartner’s 4 Architectural Viewpoints

Business Architecture

- Defines and describes the current- and future- state models of business activities (processes, assets and organization structure)

Information Architecture

- Defines and describes the current- and future- state models of the information value chain, key information artifacts (concepts), information flows

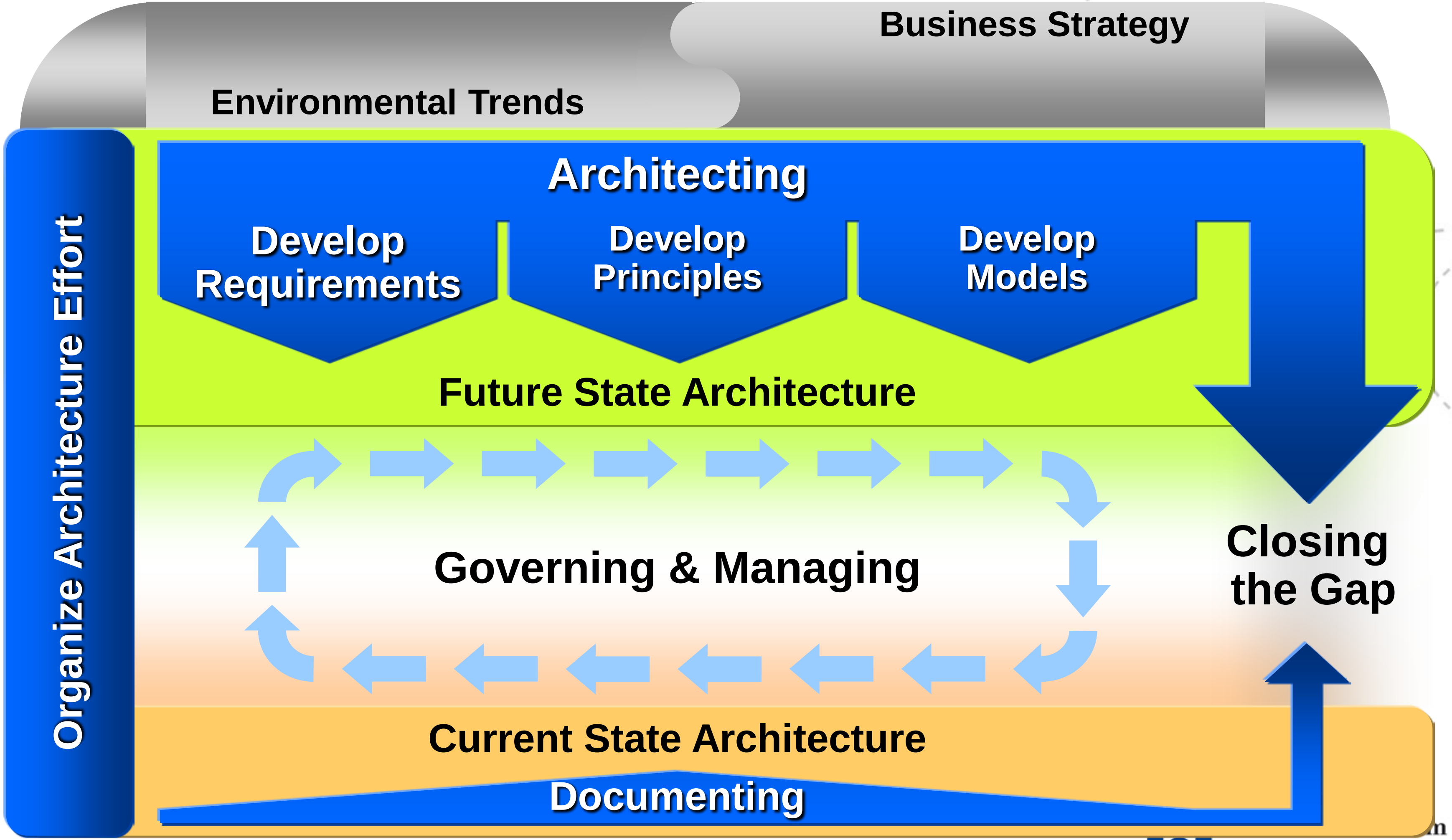
Technology Architecture

- Defines and describes the current- and future- state models of the infrastructure and technology platforms required for the solution architecture and which enables rapid engineering, solutions development and technical innovation

Solution Architecture

- Combining and reconciling (integration) the loosely coupled and often conflicting viewpoints of the primary stakeholders into a unified architecture
- Having divided to conquer, we must reunite to rule
- SA is a consistent architectural description of a specific enterprise solution
- An intersection of viewpoints

Gartner Enterprise Architecture Process Model

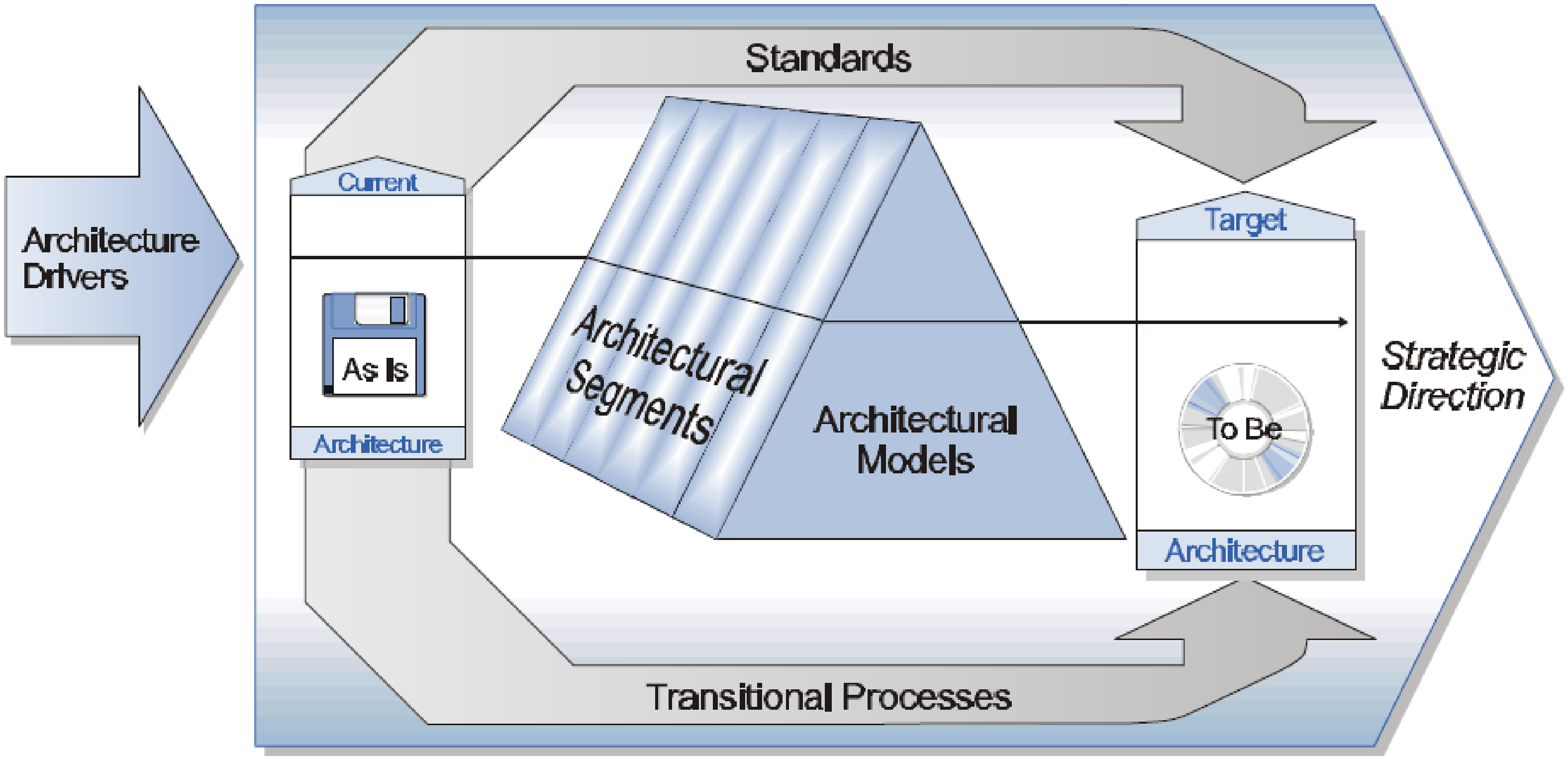


FEA Process (1)

- FEA Process is primarily focussed on creating a segment architecture for a subset of the overall enterprise.
- Segment architecture development process:
  - Step 1: Architectural analysis.
  - Step 2: Architectural definition.
  - Step 3: Investment and funding strategy.
  - Step 4: Program management plan and execute projects.

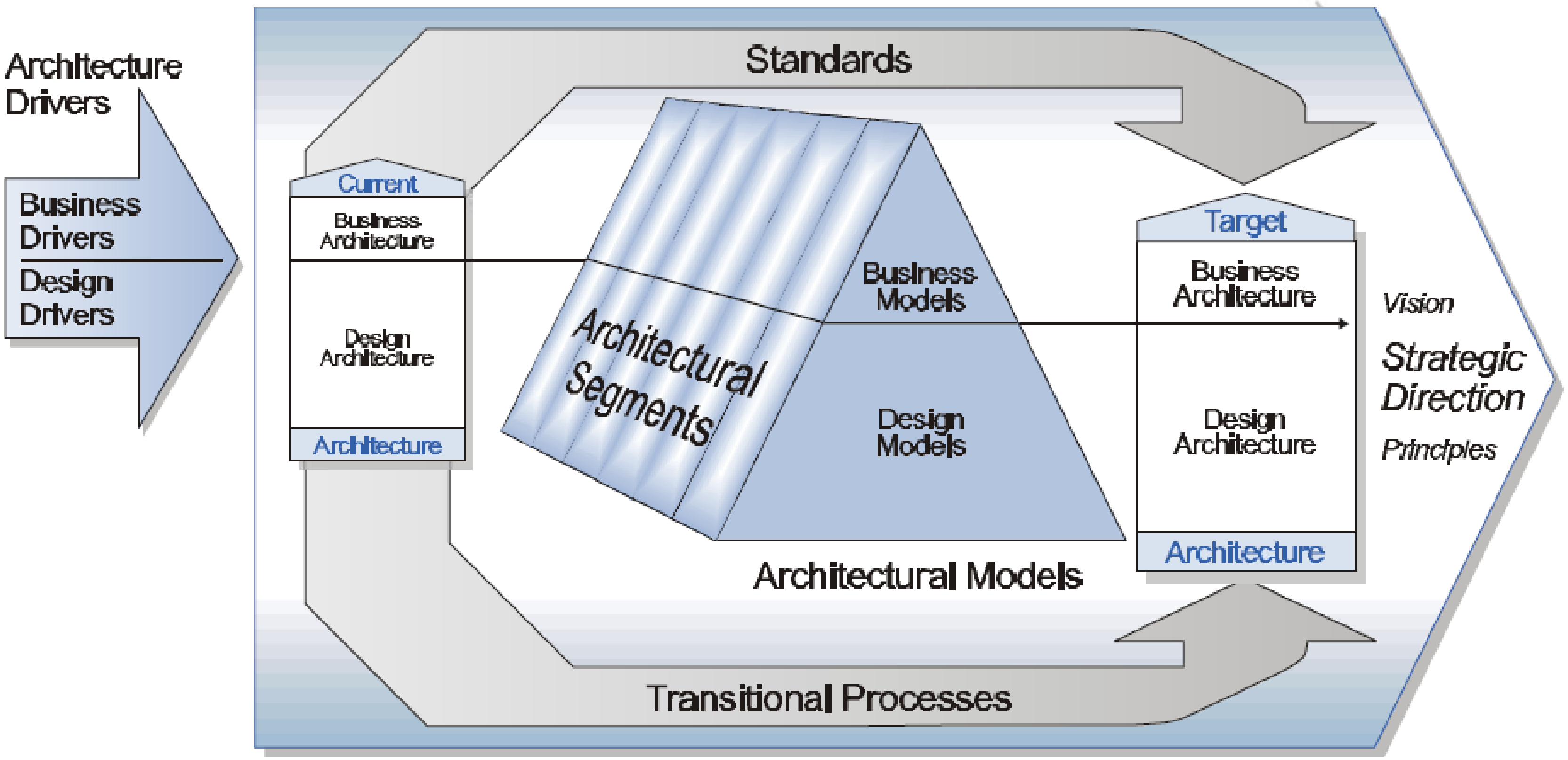
FEA Process, Level I

High level



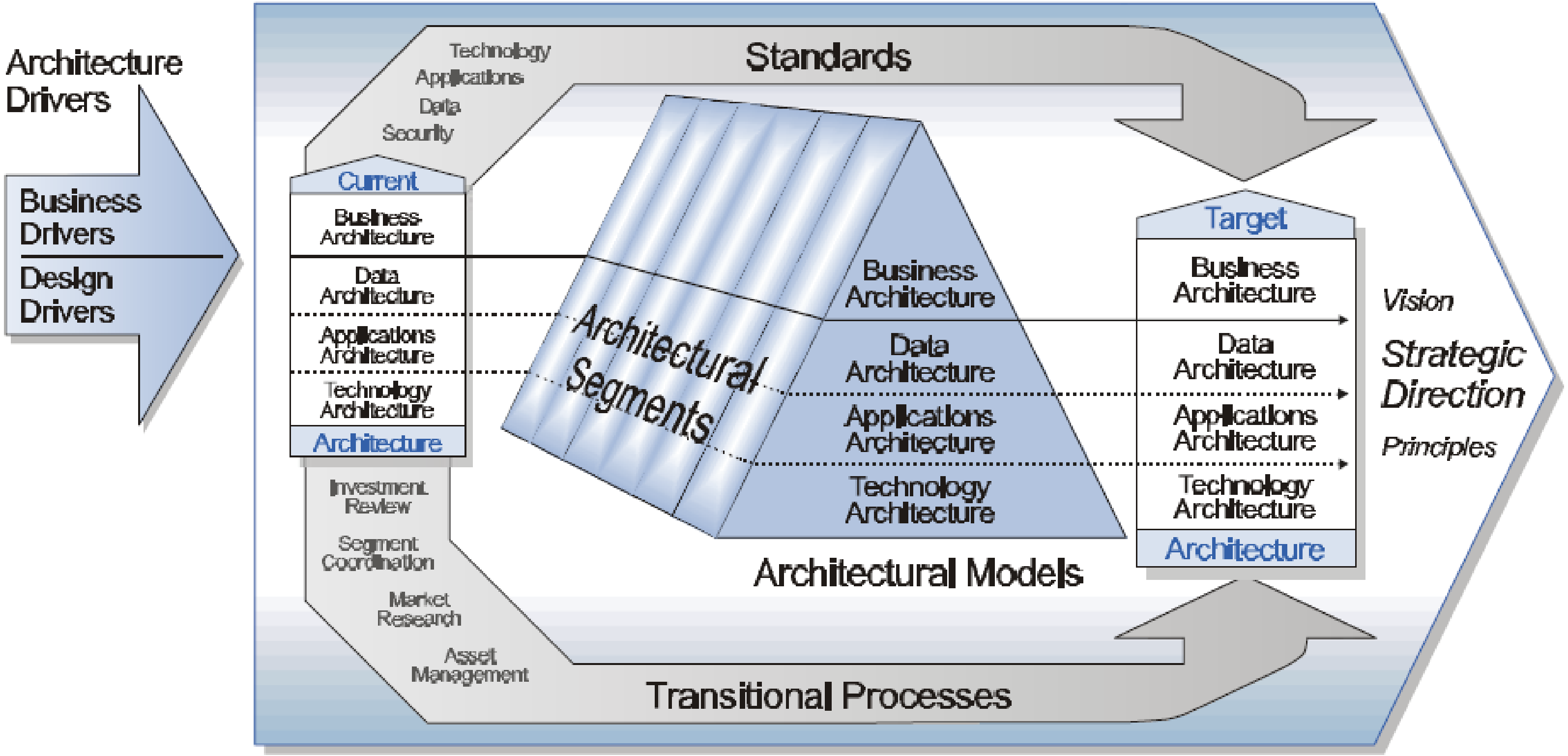
FEA Process – Level II

More detail – the business and design pieces of the architecture and how they are related.

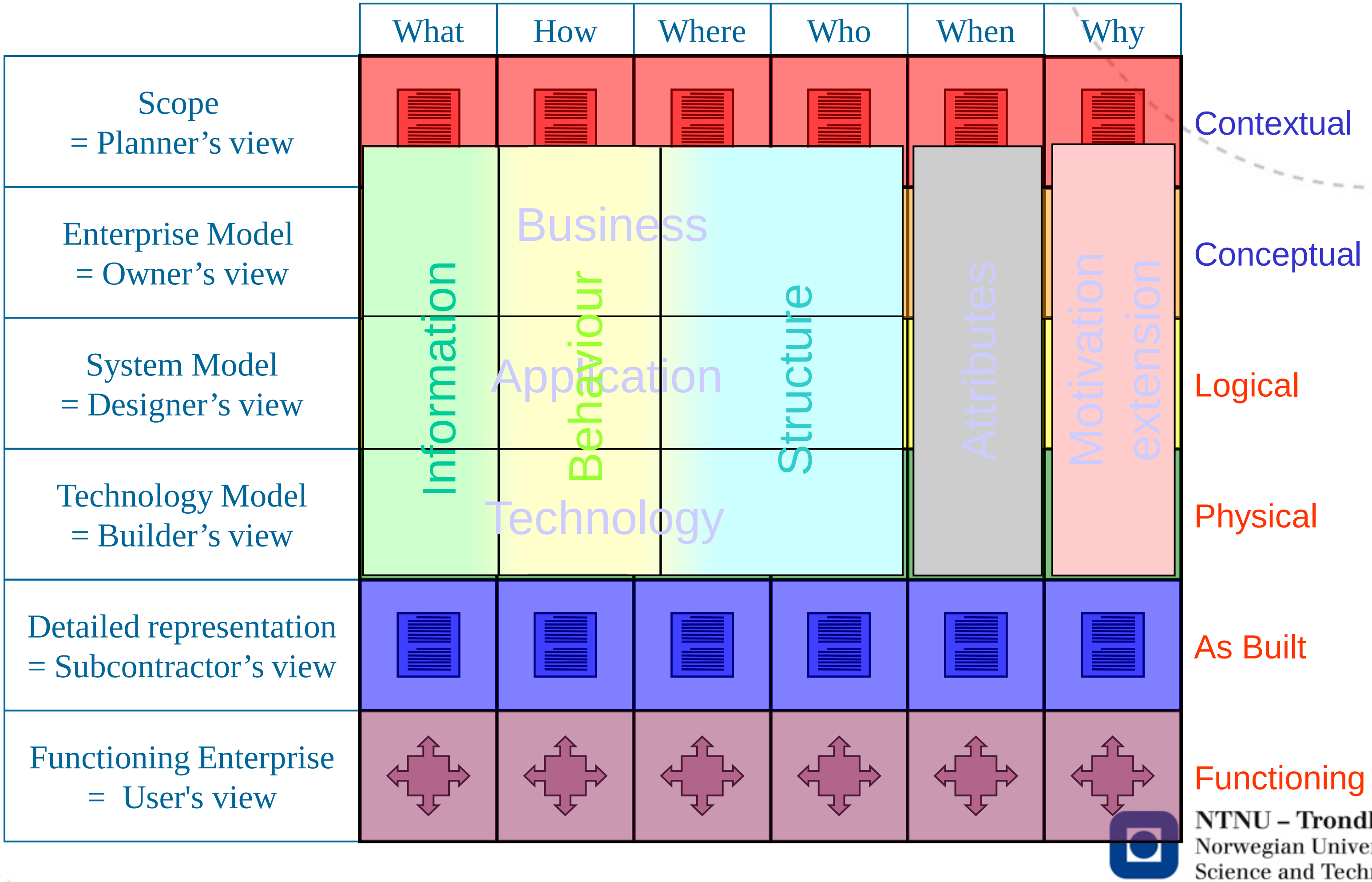


FEA Process – Level III

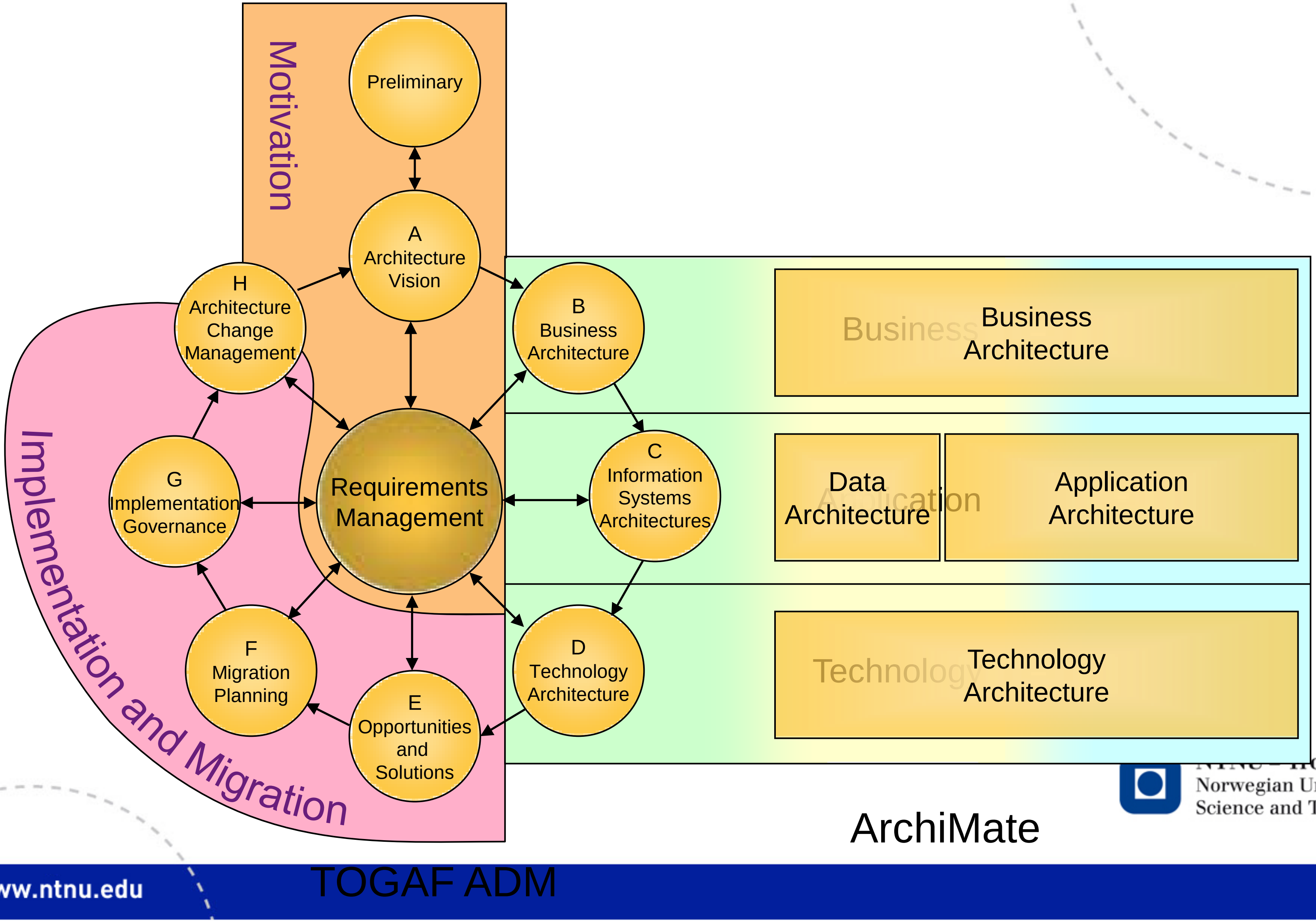
Expand the design pieces of the framework to show the 3 design architectures: data, application and technology.



ArchiMate and Zachman



TOGAF, ArchiMate and ArchiMate Extensions

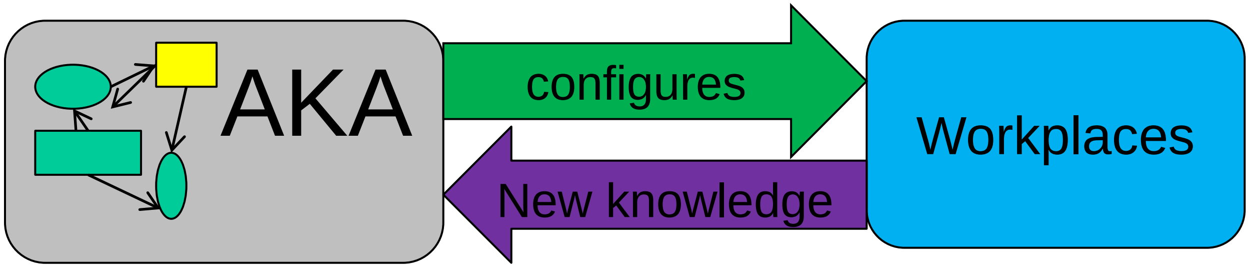


Active Knowledge Modelling and Architecture (1)

- Purpose: to provide product designers, engineers, architects and other stakeholders a common language and workplace contents for building interoperable, collaborative and reusable services and knowledge elements.
- Early phases of projects lack adequate support:
  - Effective and holistic design.
  - Conceptual design of products.
  - Collaborative design.
  - Means to capture the design rationale and the knowledge of the designers and engineers.
- Designers and engineers must feel ownership of the knowledge.

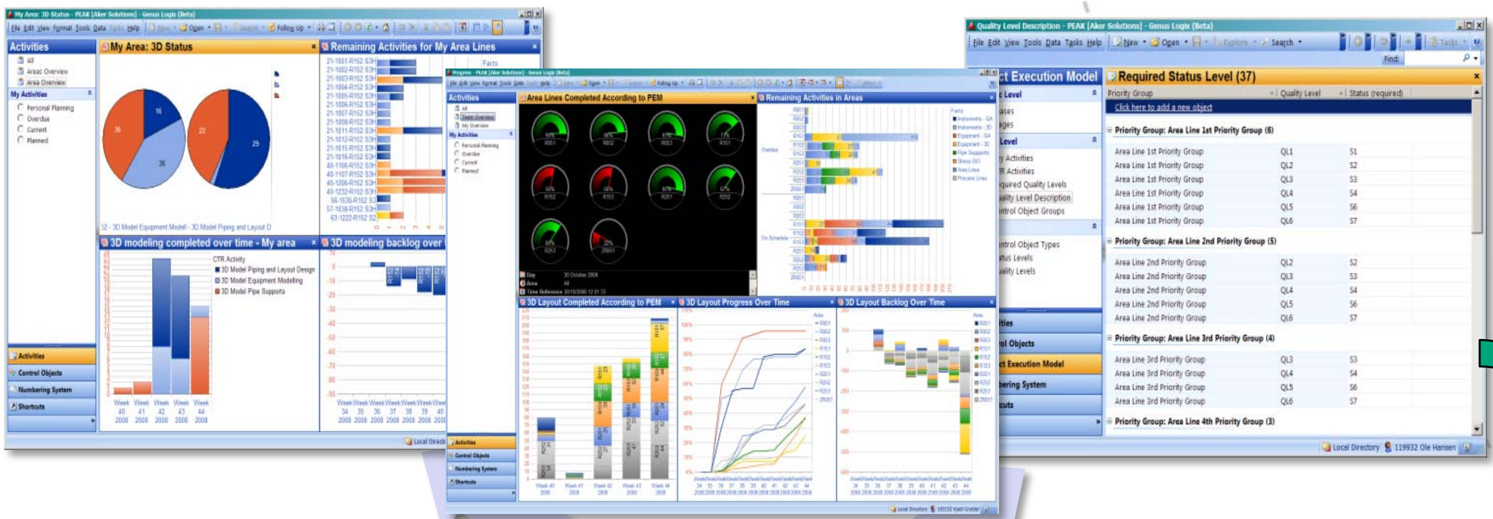
Active Knowledge Architecture (2)

- Designers and engineers must have a workplace and the services that evolve with knowledge that is created and aggregated.
- The data and knowledge stored in and reactivated from an Active Knowledge Architecture (AKA).
- Active: implies that AKA's contents (roles, task patterns, information structures, etc.) will automatically configure the workplaces. Work-centric data created in the workplaces are automatically folded back into the AKA.



AKA: An example

Workplaces
for roles


My Area: 3D Status
Remaining Activities for My Area Lines
3D modeling completed over time - My area
3D modeling backlog over
Area Lines Completed According to PEM
Remaining Activities in Areas
3D Layout Completed According to PEM
3D Layout Progress Over Time
3D Layout Backlog Over Time
Quality Level Description
Required Status Level (37)
Priority Group: Area Line 1st Priority Group (6)
Priority Group: Area Line 2nd Priority Group (5)
Priority Group: Area Line 3rd Priority Group (4)
Priority Group: Area Line 4th Priority Group (3)
Activities
Control Objects
Numbering System
Shortcuts

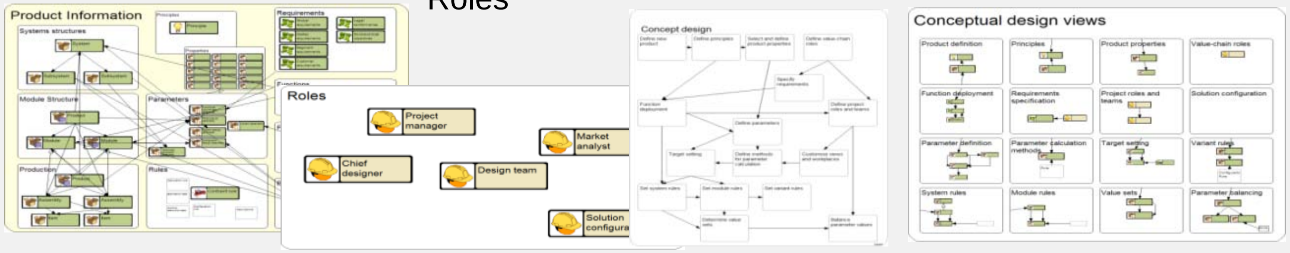
Active
Knowledge
Architecture

Products

Organization
Roles

Processes, Tasks

Systems, Views



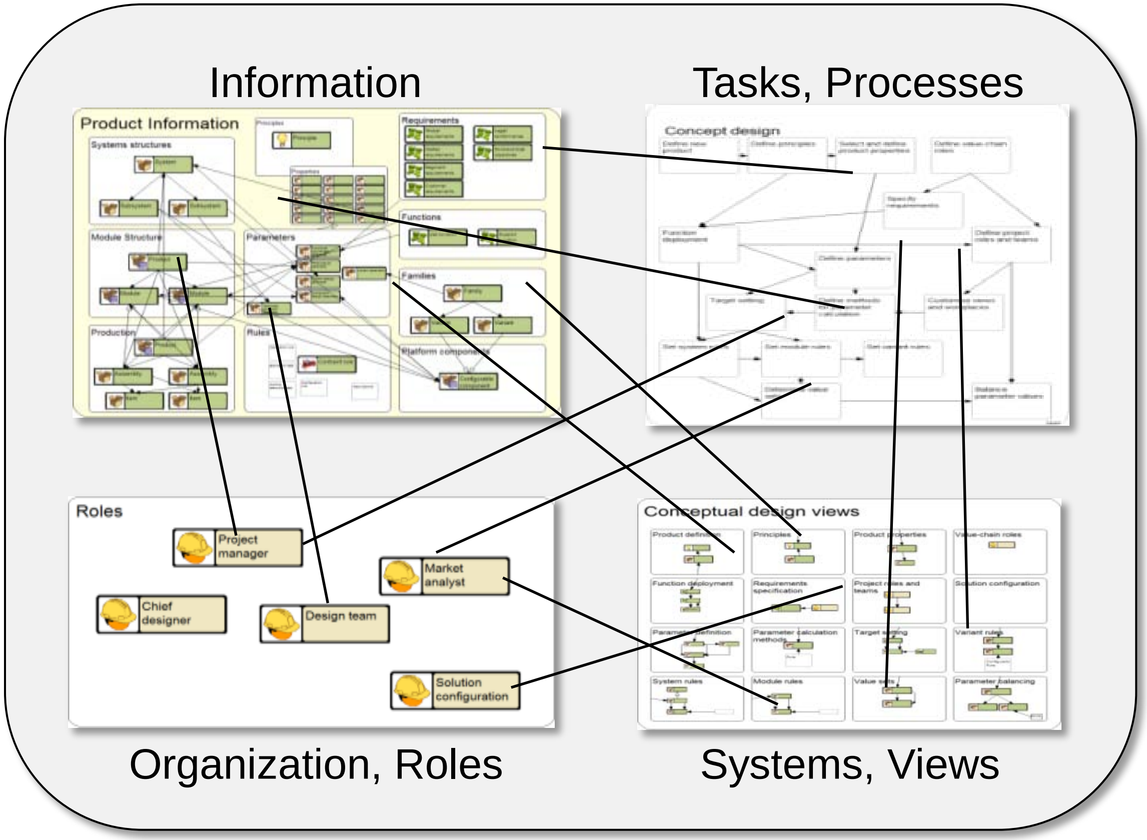
Courtesy of Commitment AS

What is an Active Knowledge Model?

- To be an active model:
  - A visual model must first and foremost be available to the users of the operational information system at the time of execution.
  - The model must automatically influence the behaviour of the computerised work system support and workplace.
  - The model must be dynamically extended and adapted; users must be supported in changing the model to fit their local needs, enabling tailoring of the work environment's behaviour.
- Active knowledge Modelling is capturing knowledge involved in building workplaces, in supporting work execution and knowledge generated by work execution.
- An active knowledge Model must support reflective views of the knowledge aspect.

Main Concept: AKA

- Visual modelling replaces programming.
- Learning by doing, practise and experimenting.



Right operation,

by the right person,

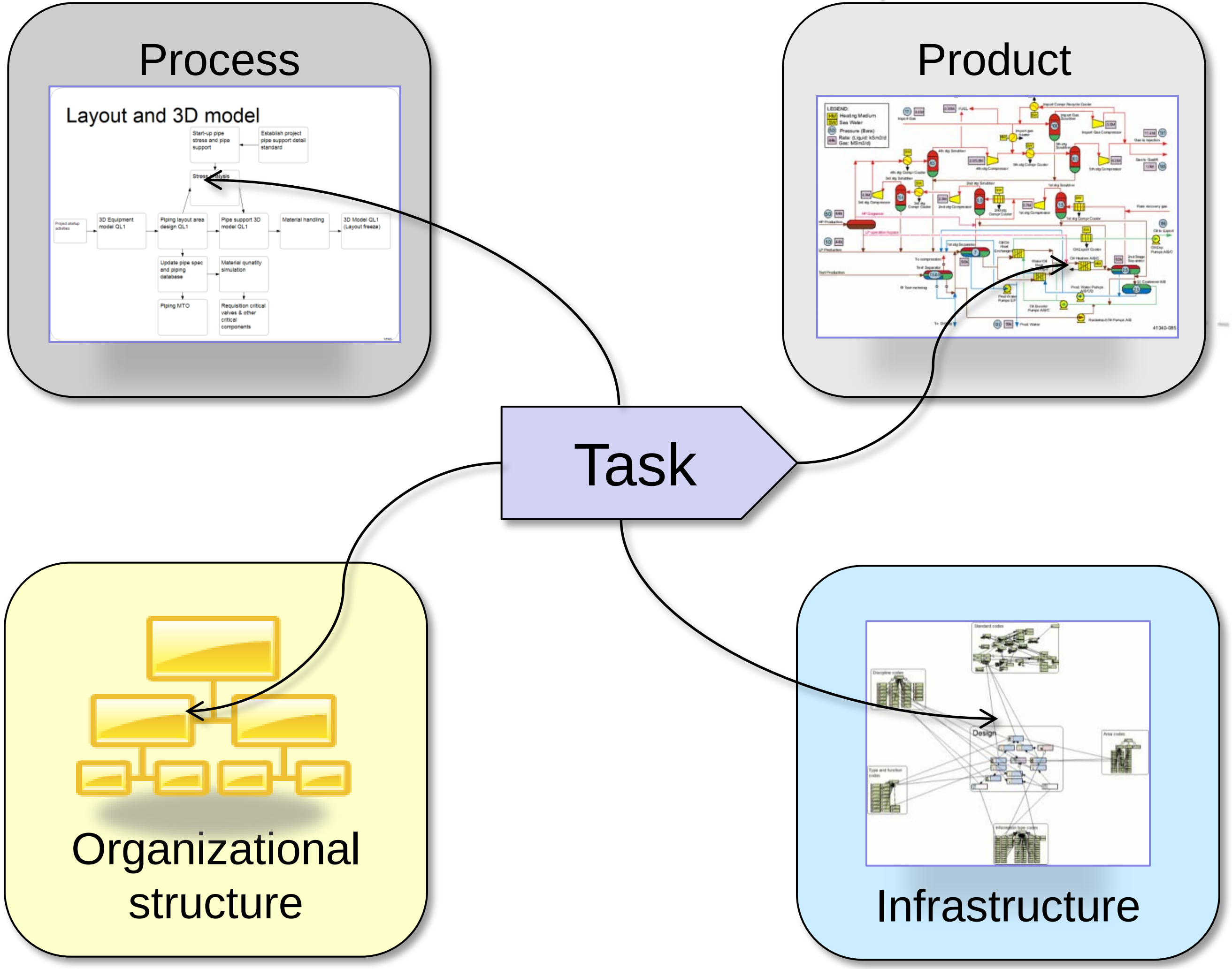
at the right time,

with the right effect.

Courtesy of Commitment AS

Inter-related operations

Operational task patterns constructed from all four surrounding dimensions, and bottom up by users



Process

Product

Task

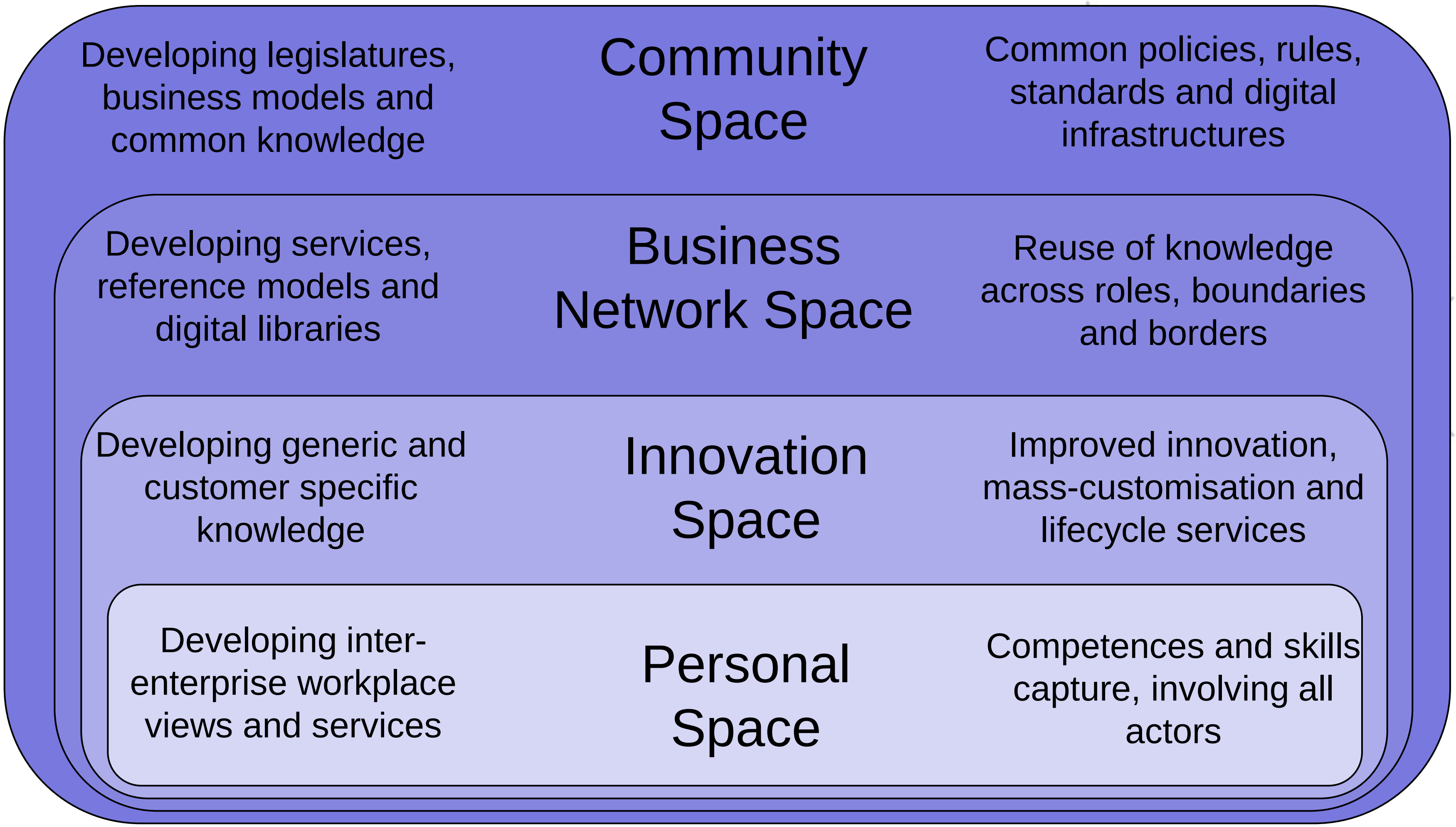
Organizational structure

Infrastructure

Layout and 3D model
Start-up pipe stress and pipe support
Establish project pipe support detail standard
Stress analysis
Project startup activities
3D Equipment model QL1
Piping layout area design QL1
Pipe support 3D model QL1
Material handling
3D Model QL1 (Layout freeze)
Update pipe spec and piping database
Material qunatity simulation
Piping MTO
Requisition critical valves & other critical components
LEGEND
Design

Courtesy of Commitment AS

Enterprise Knowledge Spaces (2)



Developing legislatures, business models and common knowledge

Community Space

Common policies, rules, standards and digital infrastructures

Developing services, reference models and digital libraries

Business Network Space

Reuse of knowledge across roles, boundaries and borders

Developing generic and customer specific knowledge

Innovation Space

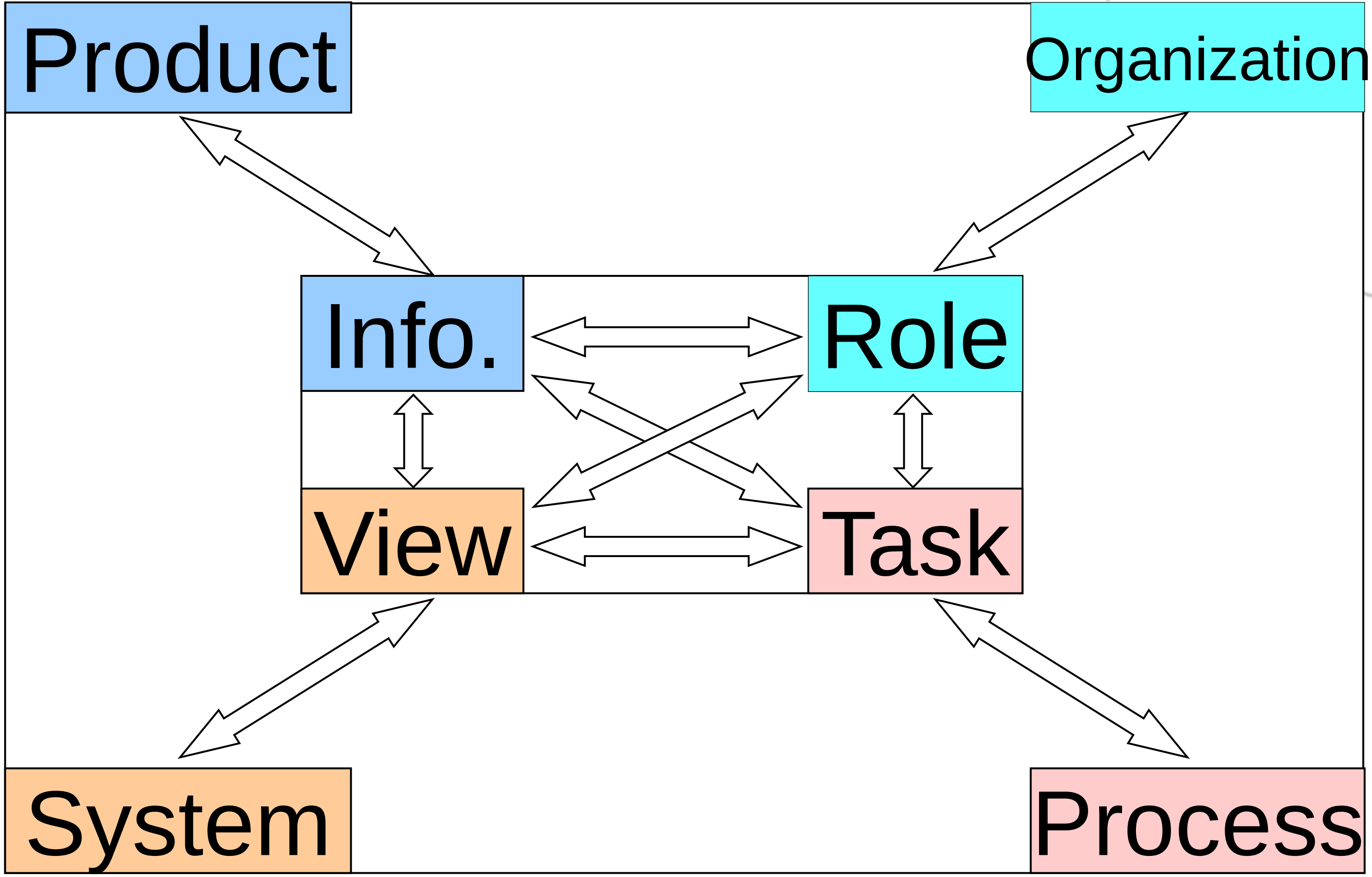
Improved innovation, mass-customisation and lifecycle services

Developing inter-enterprise workplace views and services

Personal Space

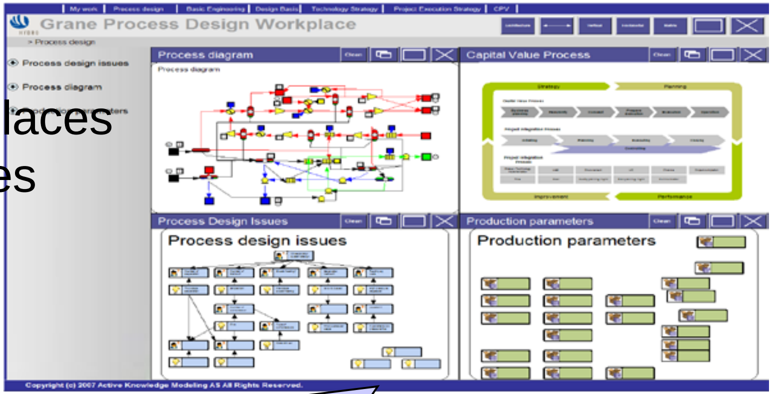
Competences and skills capture, involving all actors

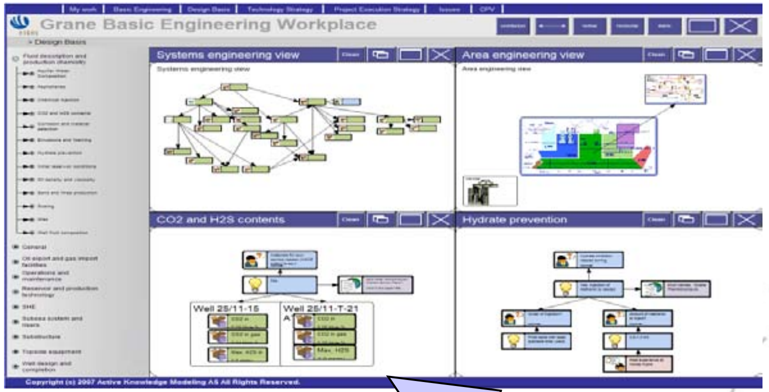
Reflection Across Knowledge Spaces

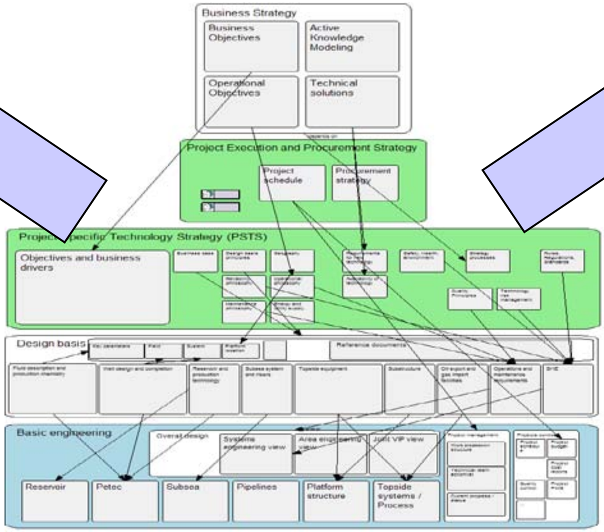


CVW – Configurable Visual Workplaces (1)

Workplaces for roles


Grane Process Design Workplace
My work | Process design | Basic Engineering | Design Basis | Technology Strategy | Project Execution Strategy | CPV
Process design issues
Process diagram
Process diagram
Capital Value Process
Process Design Issues
Process design issues
Production parameters
Production parameters
Copyright (c) 2007 Active Knowledge Modeling AS All Rights Reserved.


Grane Basic Engineering Workplace
My work | Basic Engineering | Design Basis | Technology Strategy | Project Execution Strategy | Issues | CPV
Design Basis
Systems engineering view
Area engineering view
CO2 and H2S contents
Well 25/11-15
Well 25/11-T-21
Hydrate prevention
Copyright (c) 2007 Active Knowledge Modeling AS All Rights Reserved.



Business Strategy

Business Objectives

Active Knowledge Modeling

Operational Objectives

Technical solutions

Project Execution and Procurement Strategy

Project schedule

Procurement strategy

Project Specific Technology Strategy (PSTS)

Objectives and business drivers

Design basis

Reference documents

Basic engineering

Overall design

Systems engineering view

Area engineering view

Joint VIP view

Reservoir

Petec

Subsea

Pipelines

Platform structure

Topside systems / Process

Active Knowledge Architecture

TDT4252 Summary Lecture
Spring 2014

NTNU – Trondheim
Norwegian University of Science and Technology

What next?

- Exam date: Wednesday, 21 May 2013, 09:00 - 13:00hrs
- Q&A session before the exam ?

- **Good luck with your exams!**