

# **Modeling perspectives in conceptual and enterprise modeling**

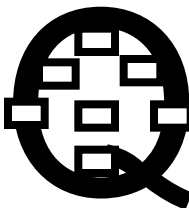
**John Krogstie**

**Professor in information systems**

**IDI, NTNU**

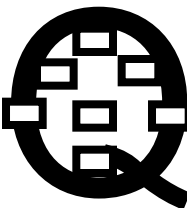
# Overview over lecture

- **Modeling perspective**
  - What is a perspective?
  - Examples of modeling languages of different perspectives
  - Usage of different modeling perspectives in practice in particular relative to enterprise modeling
- **Based on Chap 3.3 in book: Model-based development and evolution of information systems: A quality approach (Chapter can be downloaded from SpringerLink at <http://link.springer.com/book/10.1007/978-1-4471-2936-3/page/1>). Also available directly from Canvas in Manuscript-form**
- **Friday – looking at combination of perspectives (start of 3.5) and other generic mechanisms**



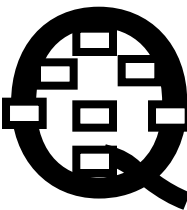
# An overview of conceptual modeling languages

- More than 500 different modeling languages described in literature
- Classifying languages according to modeling perspectives to get an overview
- Today: Overview of perspectives, with brief examples of languages of each perspective, will go in more detail on the languages in some of the perspectives later in the course also looking upon how they can be used in concert



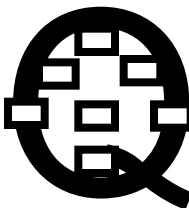
# Discussion

- Why are there so many modeling languages?



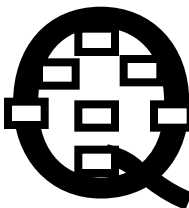
# Modeling perspectives

- **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**

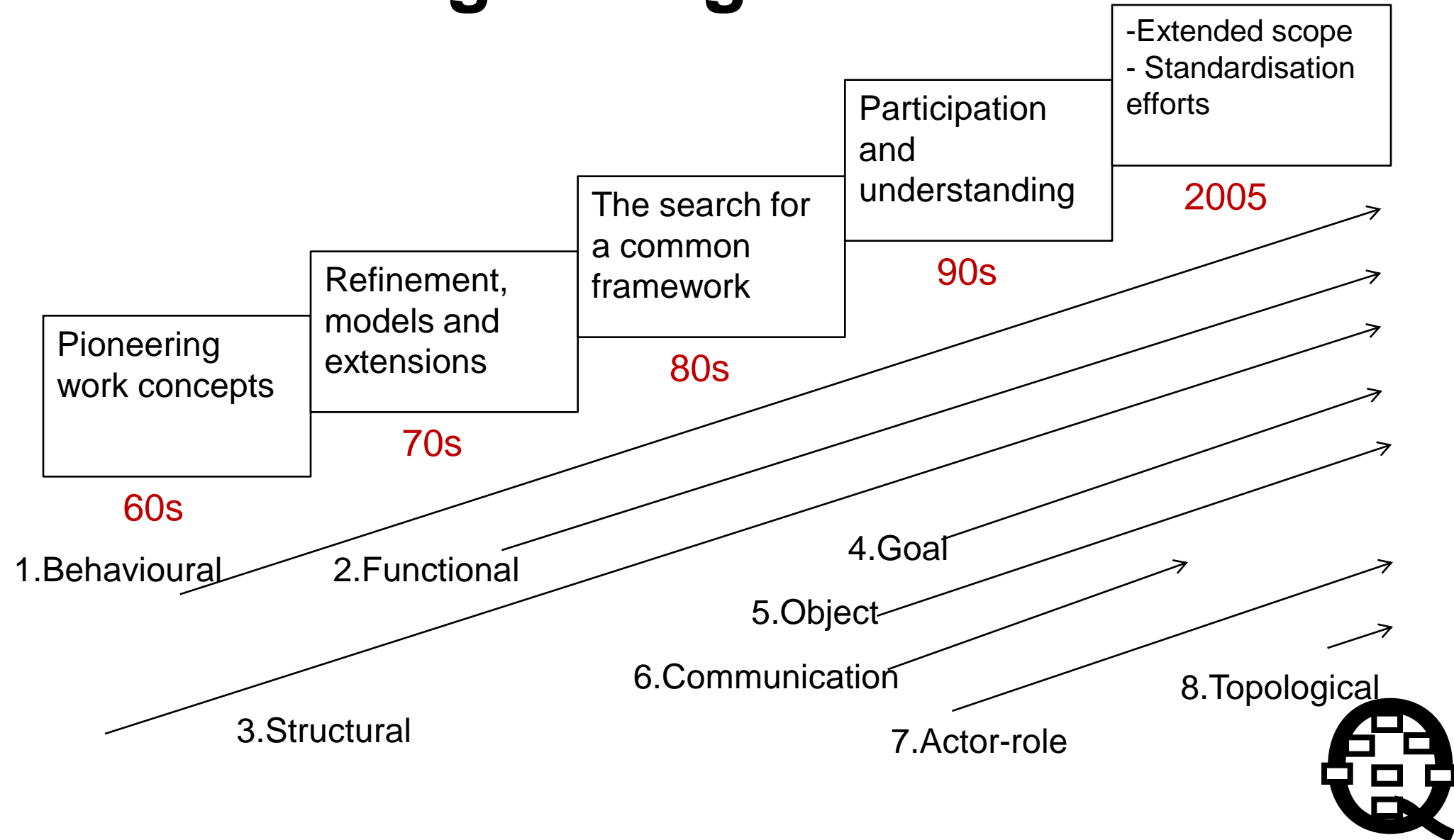


# 8 perspectives/orientations to conceptual and enterprise modeling

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

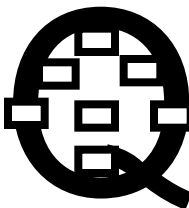


# Modelling during four+ decades



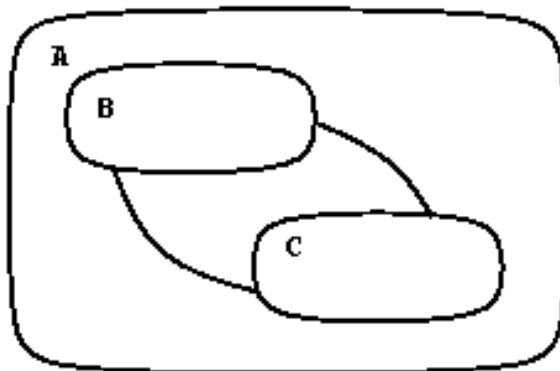
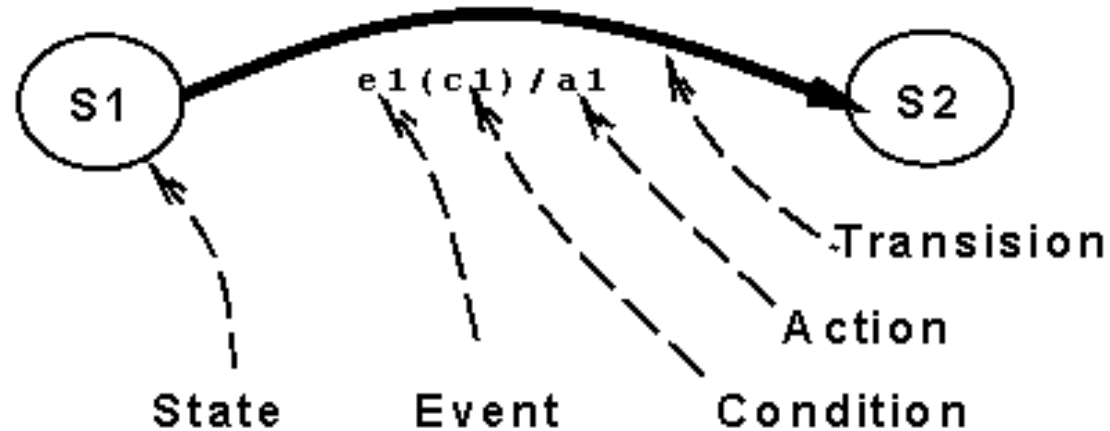
# 1. Behavioral perspective

- **Description of the behavioral dynamics of a system**
- **Main concept: States and transitions between states**
- **Examples**
  - State Transition Diagrams (STD/STM)
  - Statecharts
  - Petri-nets (1961)
  - SDL – Specification and Description Language
  - System Dynamics
- **Different types of state transition diagrams used particularly within real-time systems and telecommunications systems. Petri-net semantics is the base semantics for a number of modern process modeling languages (see below), much applied in Business Process Modelling (BPM)**
- **Also used in enterprise models, but often on a less detailed level**

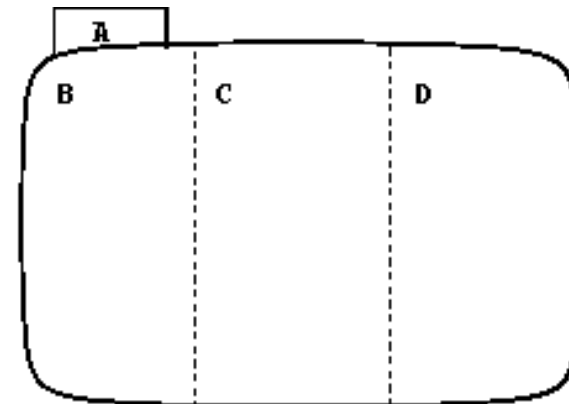




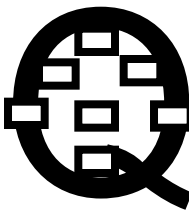
# Example: STD/statecharts



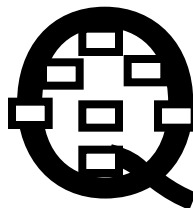
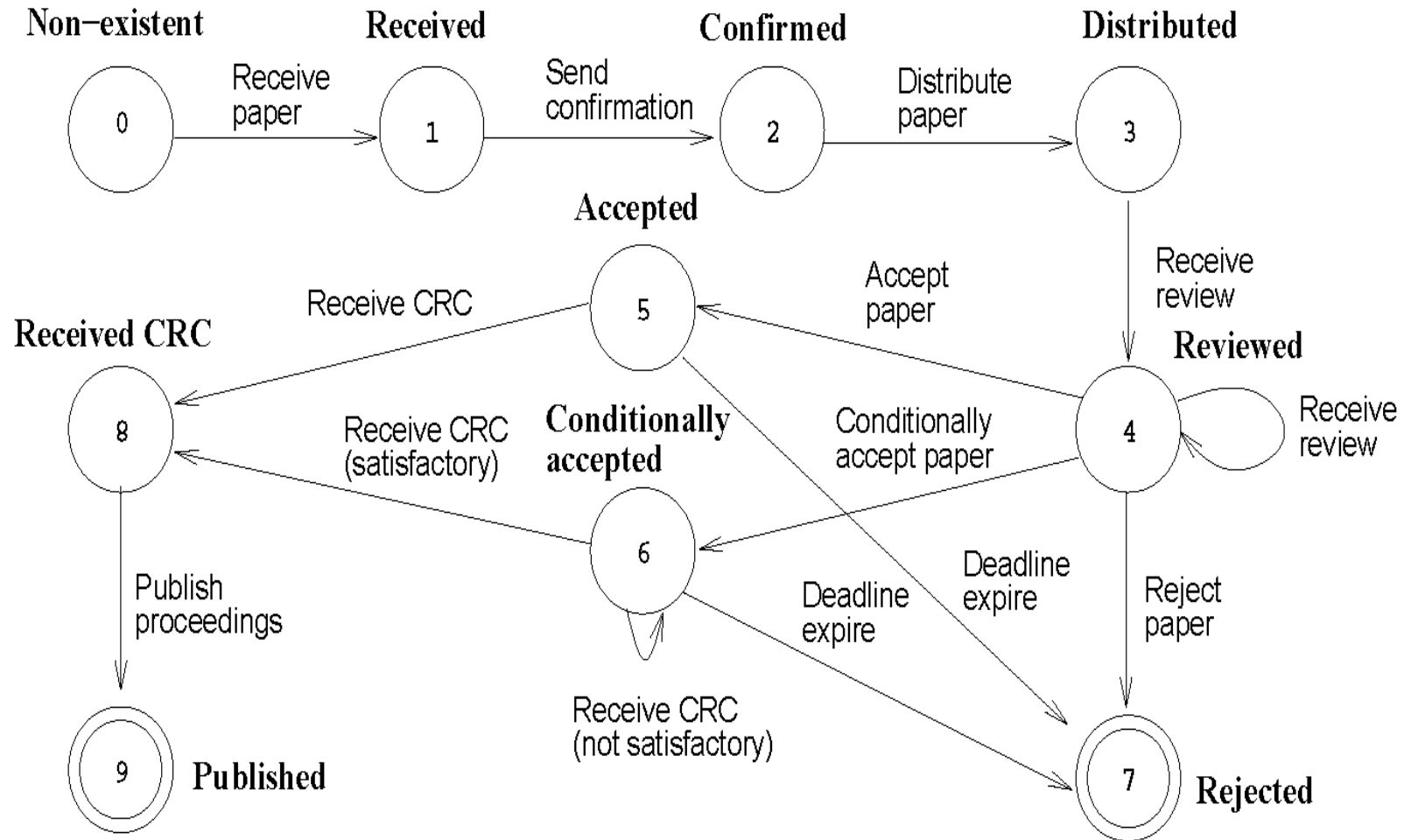
XOR decomposition



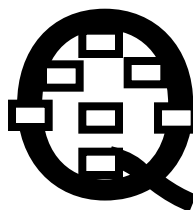
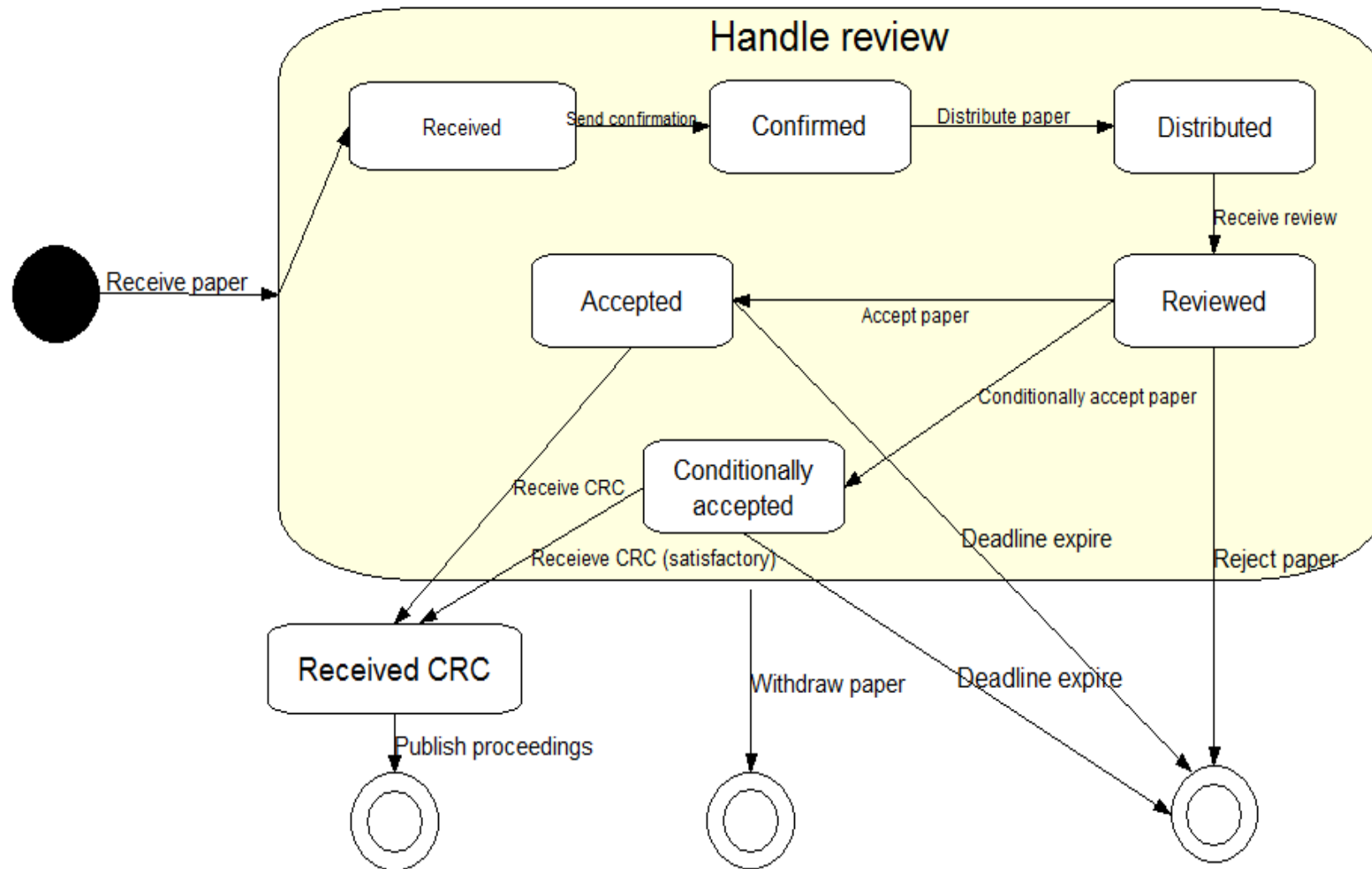
AND decomposition



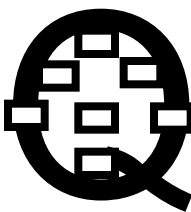
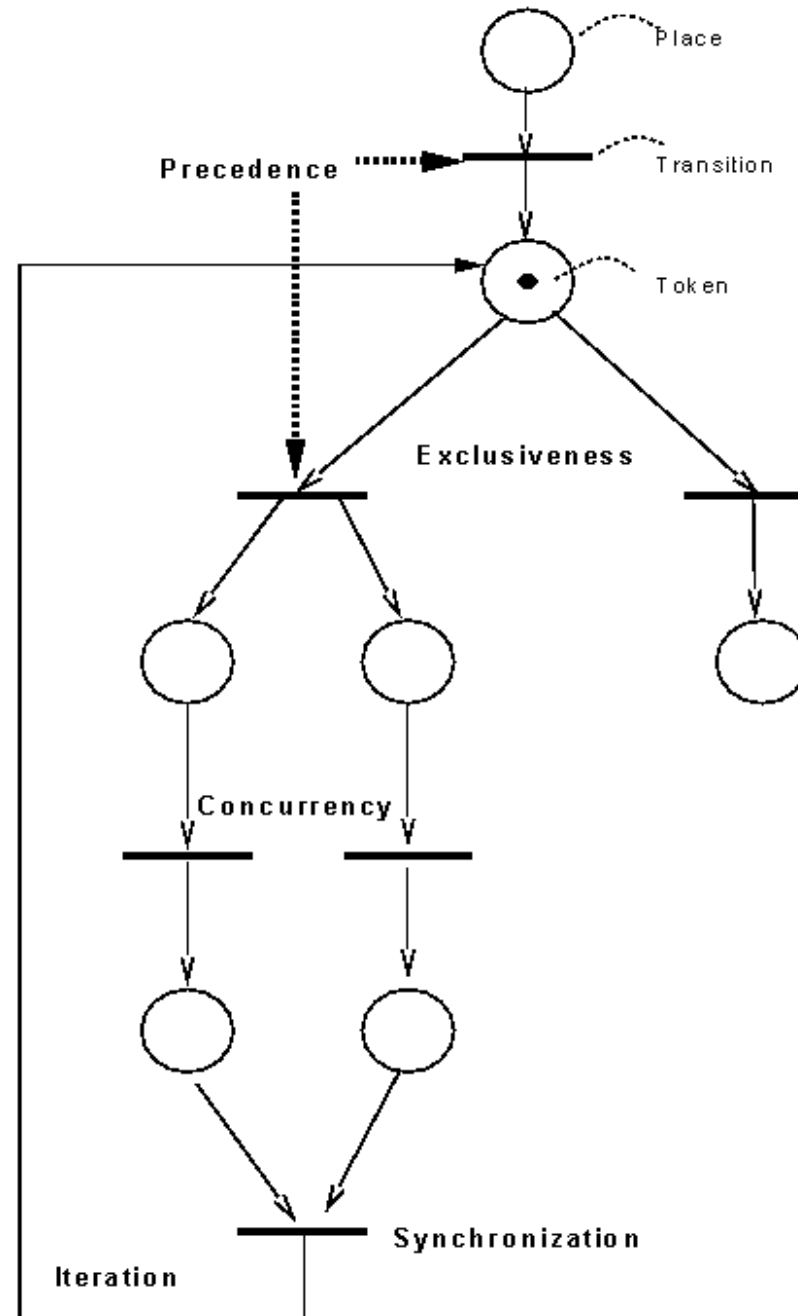
# States in a paper process



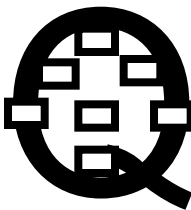
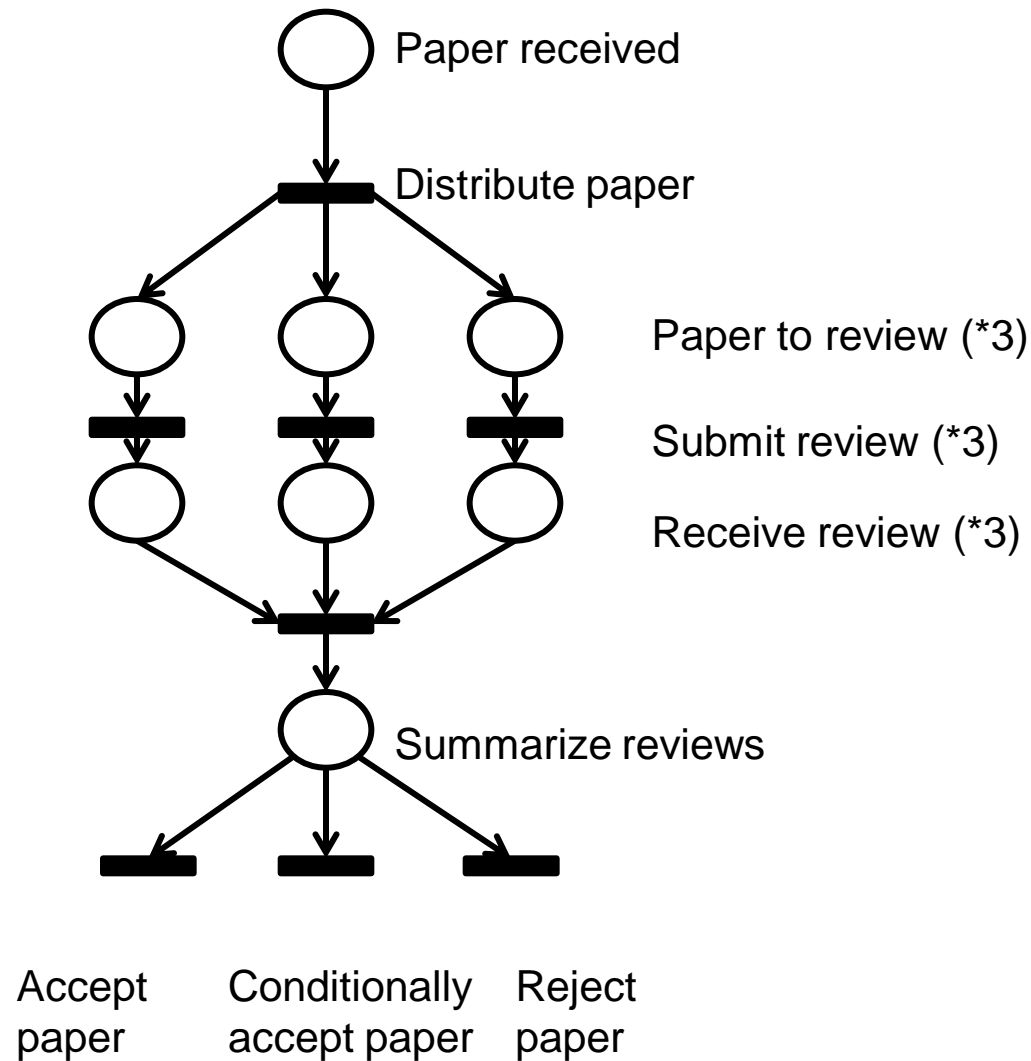
# Example of use of statecharts



# Example: Petri-nett

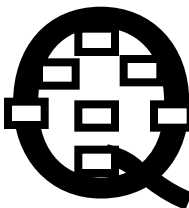


# Improving expressiveness through Petri-nets

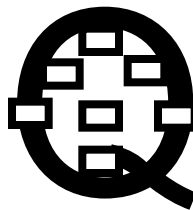
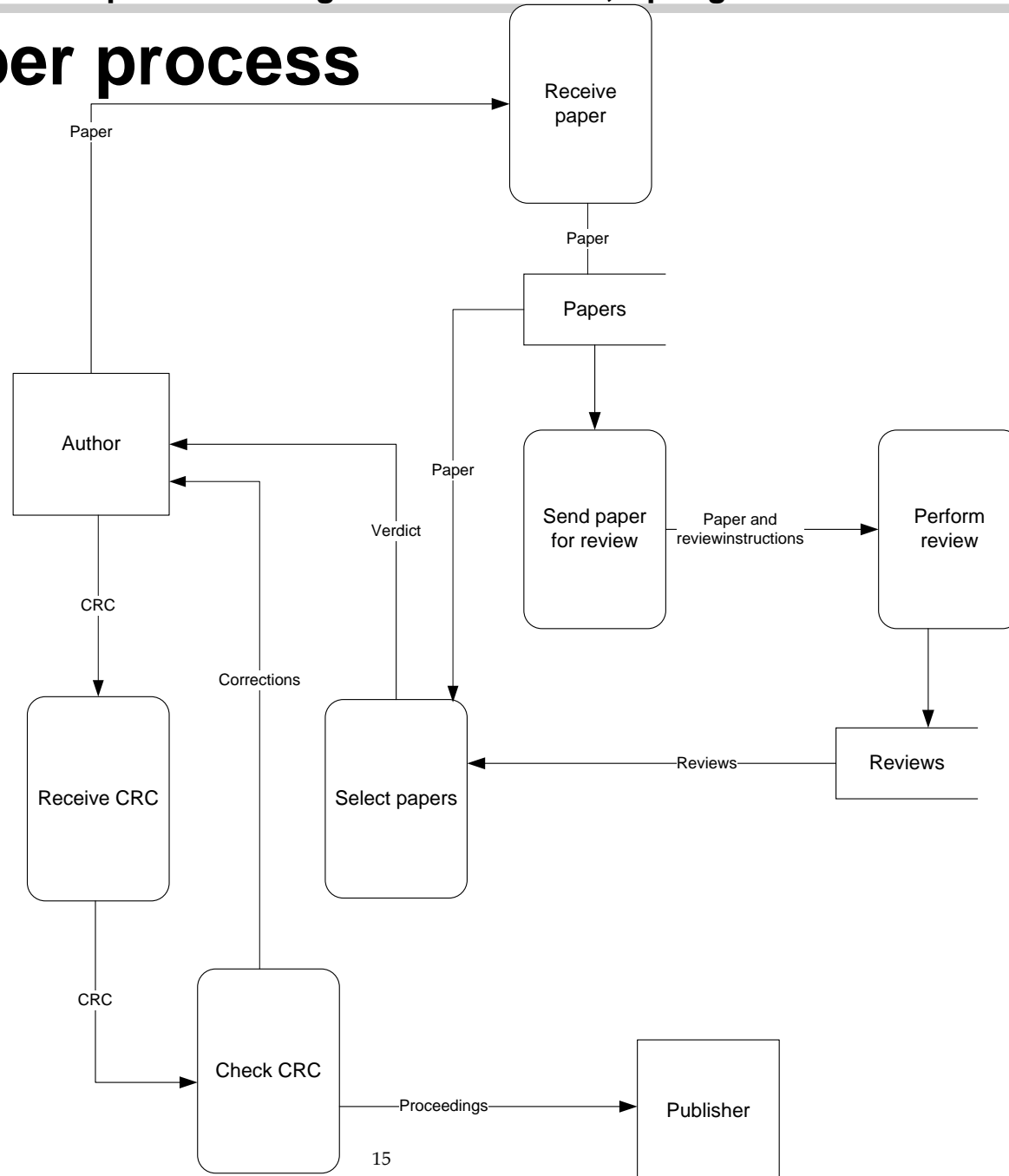


## 2. Functional perspective

- Often called process-oriented (although process modeling can be done according to a number of perspectives)
- Main concept: Transformation (function, process, activity, action, task)
- IPO – Input, Process, Output : Get a static input and transform to another static object
- Aggregation through decomposition
- Paradigm example: DFD
- Many variants and extensions (SA/RT, IDEF/0)
- Many newer languages also include also behavioral modeling (control flow) e.g. BPMN
- Used a lot in process modeling, both for business processes and processes within system applications and workflow modeling

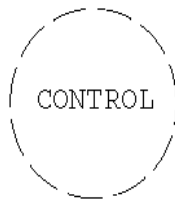


# DFD of paper process



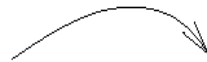
# Example on extension of DFD: SA/RT

1) TRANSFORMATIONS

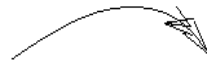


2) DATA FLOWS

DISCRETE DATA



CONTINUOUS DATA



3) EVENT FLOWS

SIGNAL



ACTIVATION



DEACTIVATION

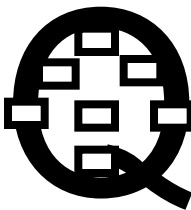


4) STORES

DATA  
STORE



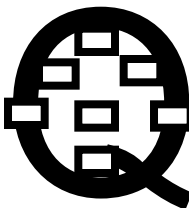
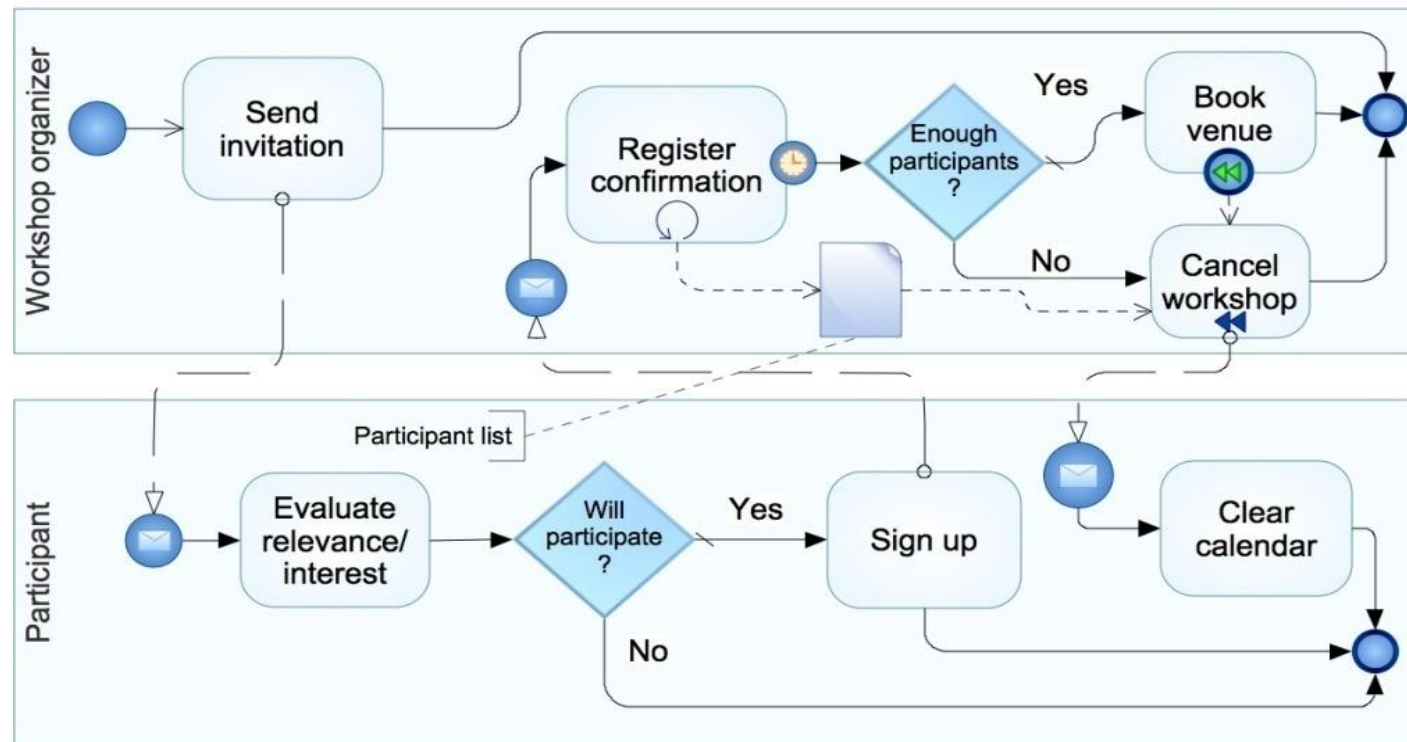
BUFFER





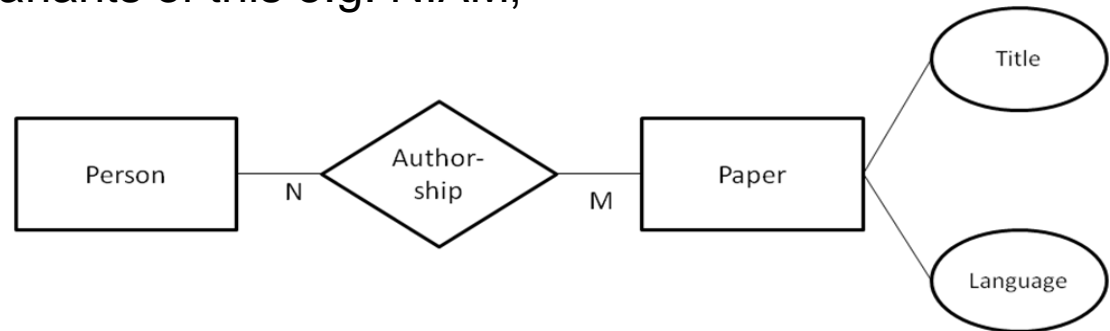
# Combinations of the functional and behavioural perspective

- Functional modelling + control flow enables modeling many of the same mechanisms as e.g. Petri-nets
- UML Activity Diagram, YAWL, EPC, EEML, BPMN...

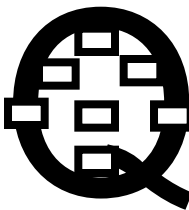


### 3. Structural perspective

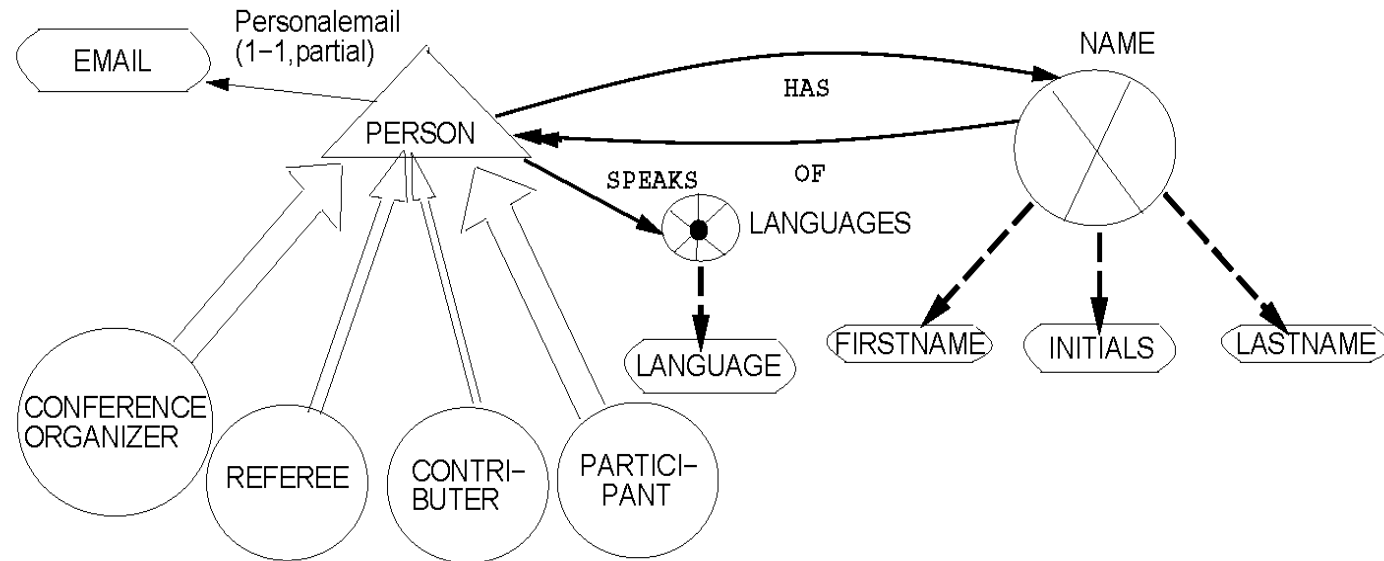
- Static description of the structure of the passive elements in a domain
- Main concept: Entity (Phenomena, concept, thing...)
- Examples
  - Data-modelling (e.g. ER and variants of this e.g. NIAM, ORM...)
  - Semantic nets
  - Conceptual graphs



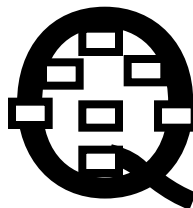
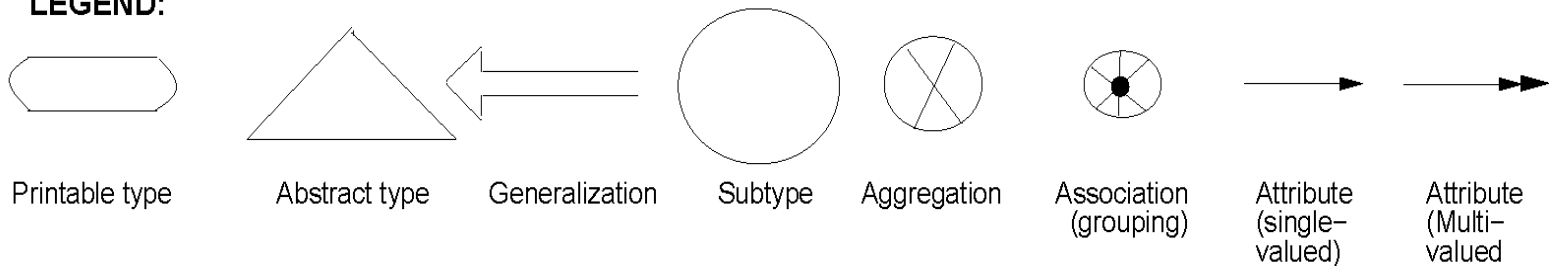
- Especially data modeling is used a lot in practice.
- Enterprise models of the structural perspective often less detailed than database models



# Example: GSM

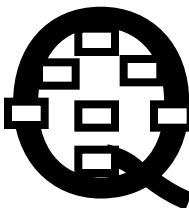


## LEGEND:

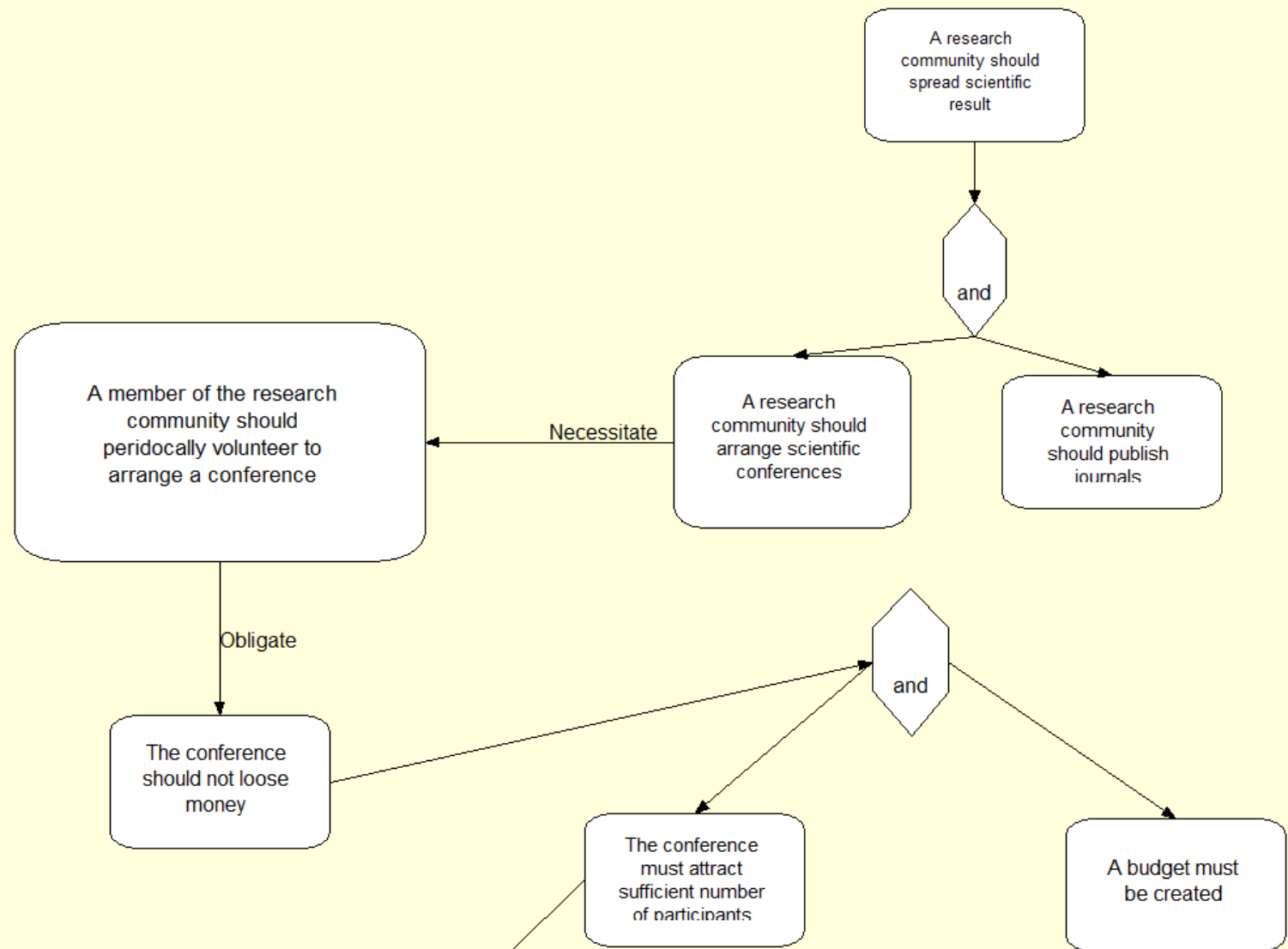


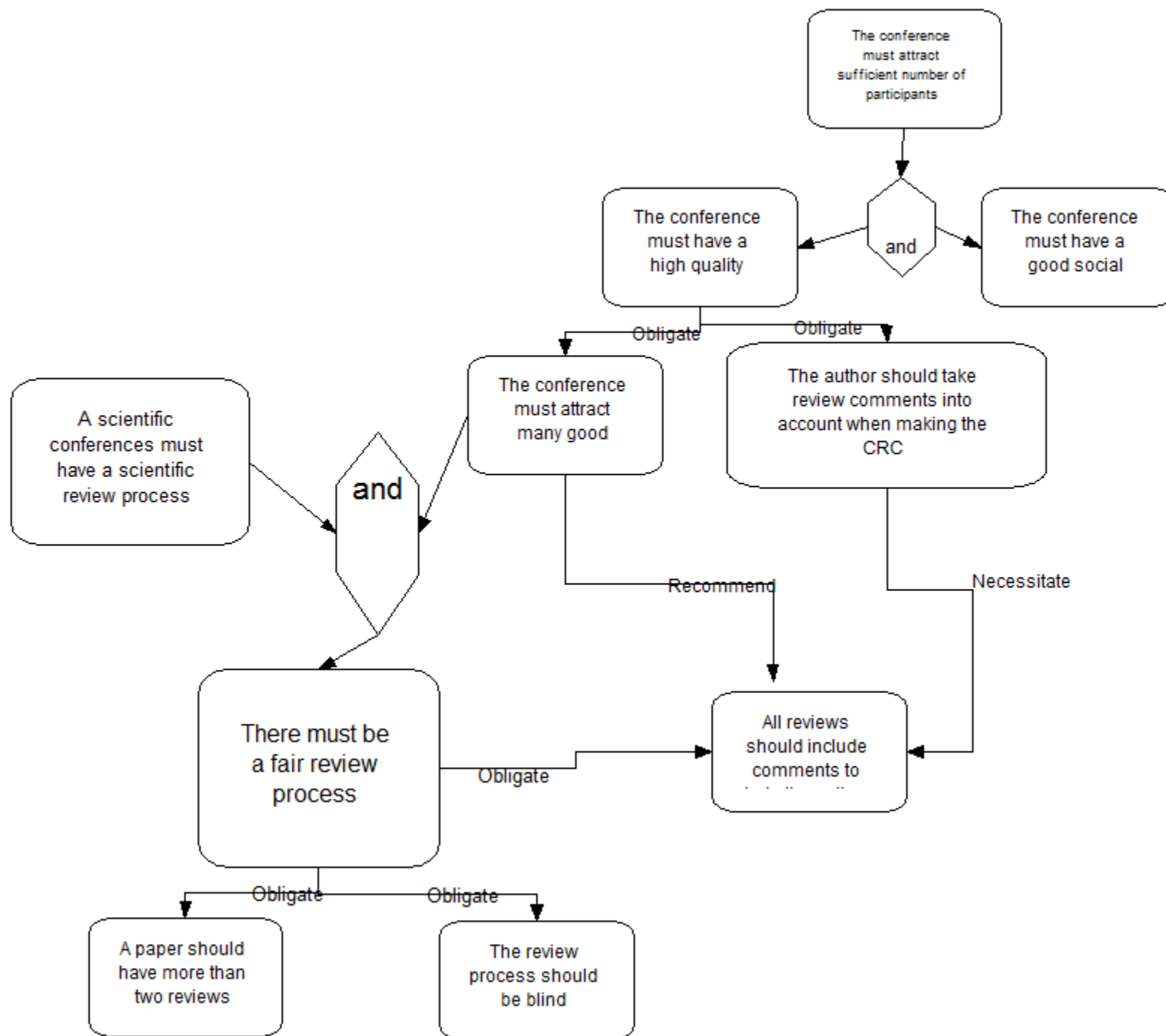
## 4. Goal and Rule perspective

- Description of goals/means connections
- Main concept: Rule (goal, constraint)
- A rule is something that influences the actions of a set of actors.
- Standard form: IF condition THEN action
- Examples:
  - Rule hierarchies (goal-oriented modeling) see e.g. i\* and EEML
  - Business rules systems
  - Tempora : Combination of structural, functional/behavioral and goal orientation
- **Usage** : Goal hierarchies in requirements engineering, modeling of motivation, laws and policies in enterprise models. Use of rule-based systems “behind the scenes” in applications

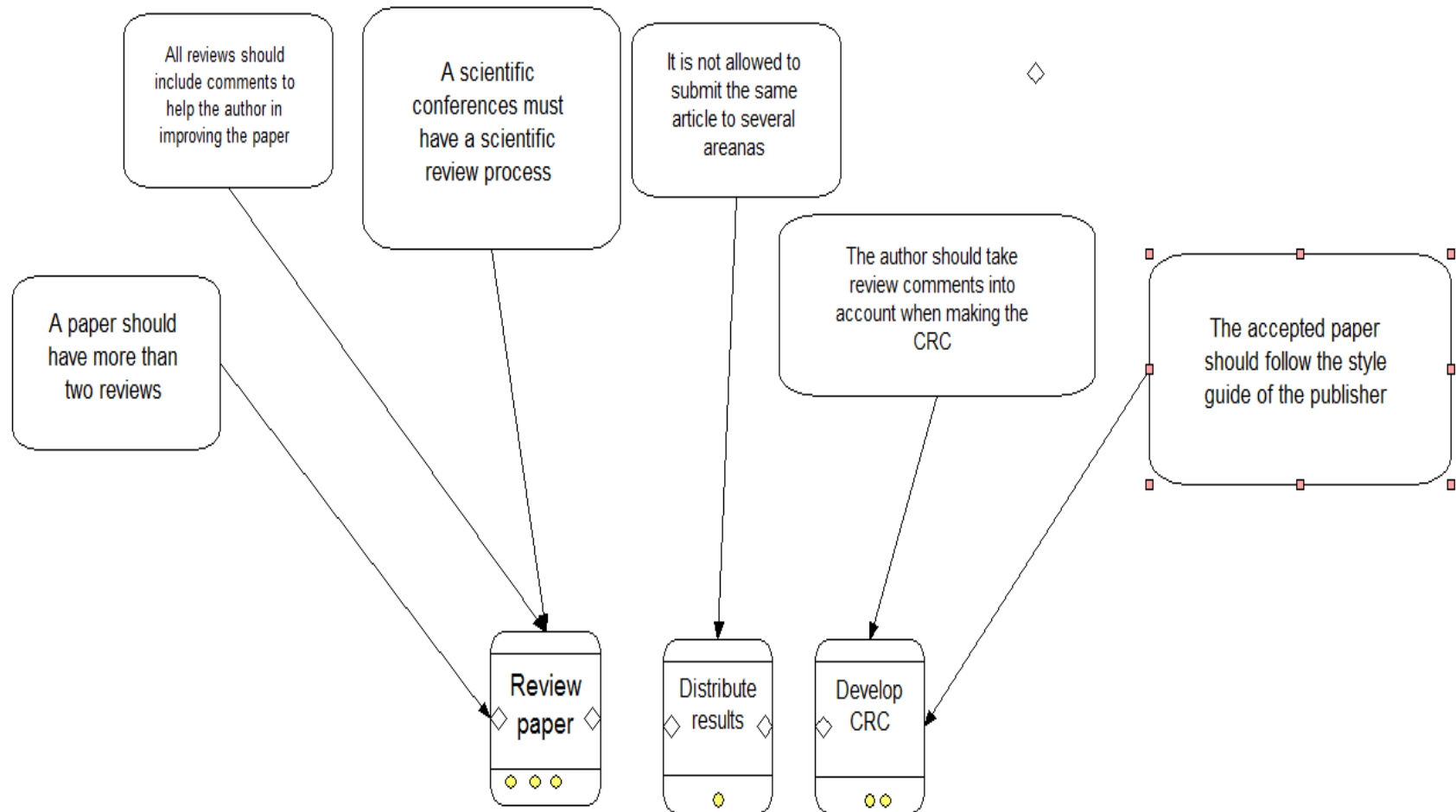


# Goal-hierarchies (example EEML)



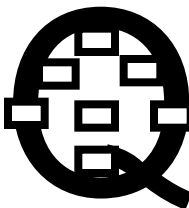


# Leaf goals relative to tasks (EEML example)



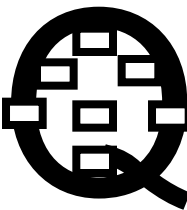
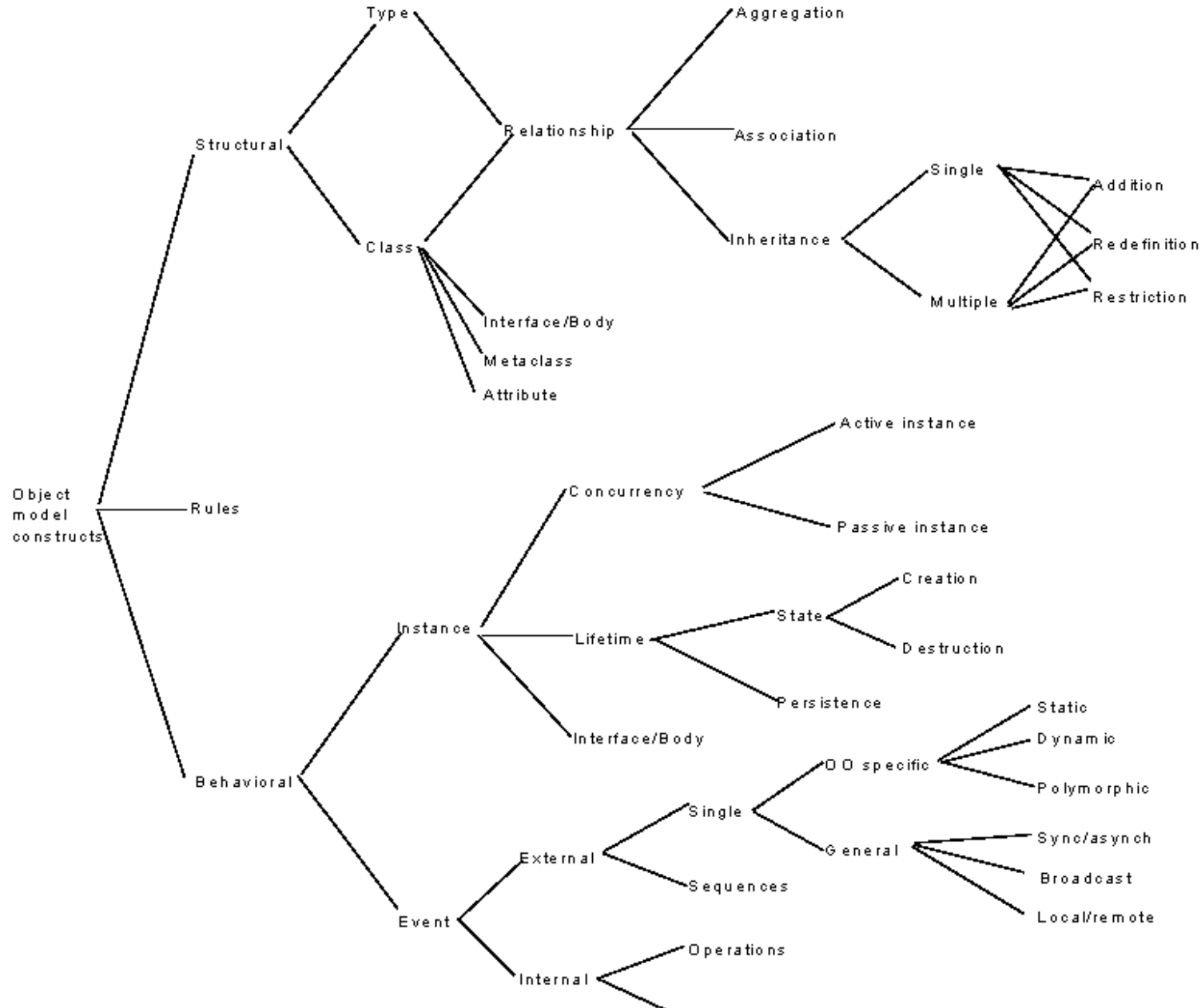
## 5. Object perspective

- Description of the world as autonomous, communicating objects
- Main concept: Object. An entity with a unique identity with a local state only accessible by sending messages to the interface of the object.
- Examples: UML, OML, OMT, Shlaer/Mellor, OOA...
- Basis from design and programming of object oriented systems
- Used more as object-oriented development has become more common (particularly UML and extensions of UML)
- Modeling on the CIM-level - Computational independent models could be used for enterprise modeling

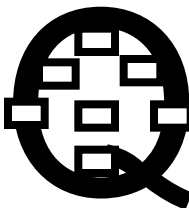
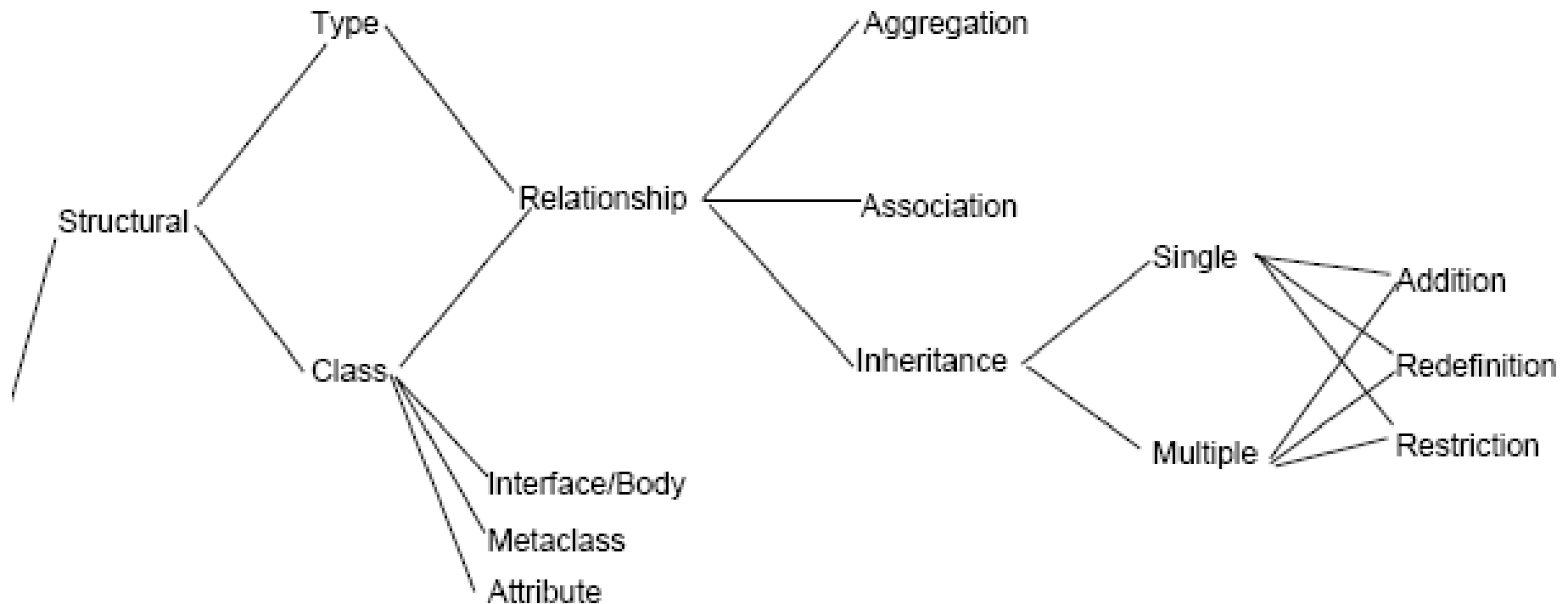




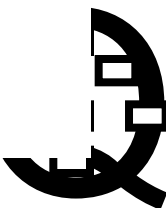
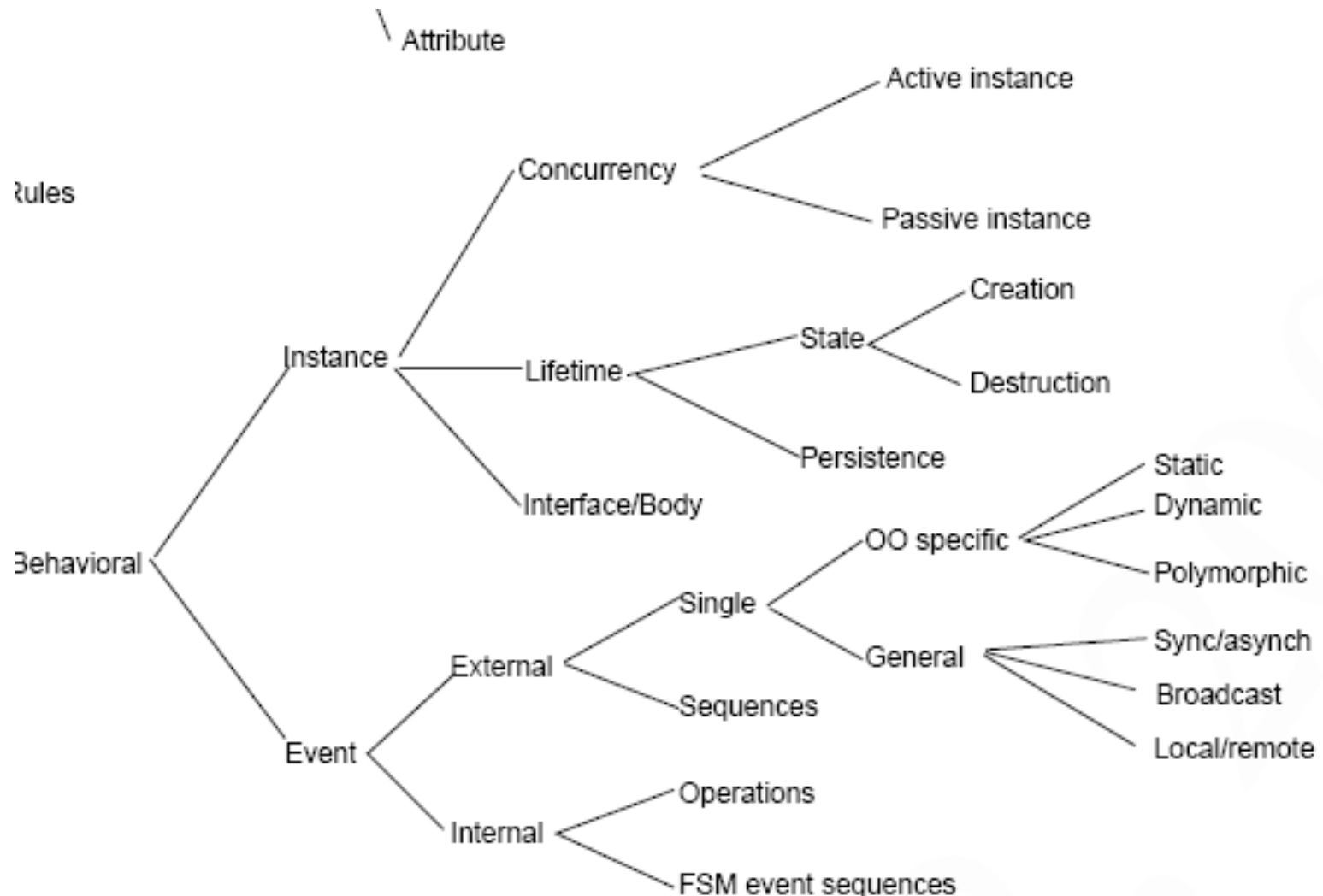
# Concepts represented in OO-modeling



# Structural aspects in object-orientation

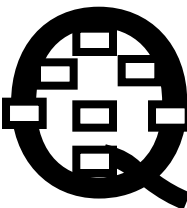


# Dynamic aspects in object-orientation



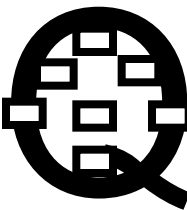
## 6. Social Communication perspective

- Description of the (social) communication structure
- Also called the language action perspective (LAP)
- Main concepts: Speech Acts and Conversations
- Based on philosophy of language (Austin, Searle, Habermas)
- Different types of speech acts which are put together into conversations
- Specifically used in workflow modeling and CSCW/Groupware (but diminishing popularity)



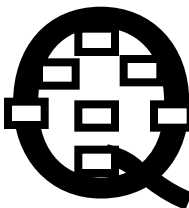
# Speech acts - Illocution

- **Three distinct acts**
  - **Locution:** The physical expression
  - **Illocution:** The intention of the locution
  - **Perlocution:** Acts following the illocution



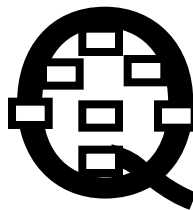
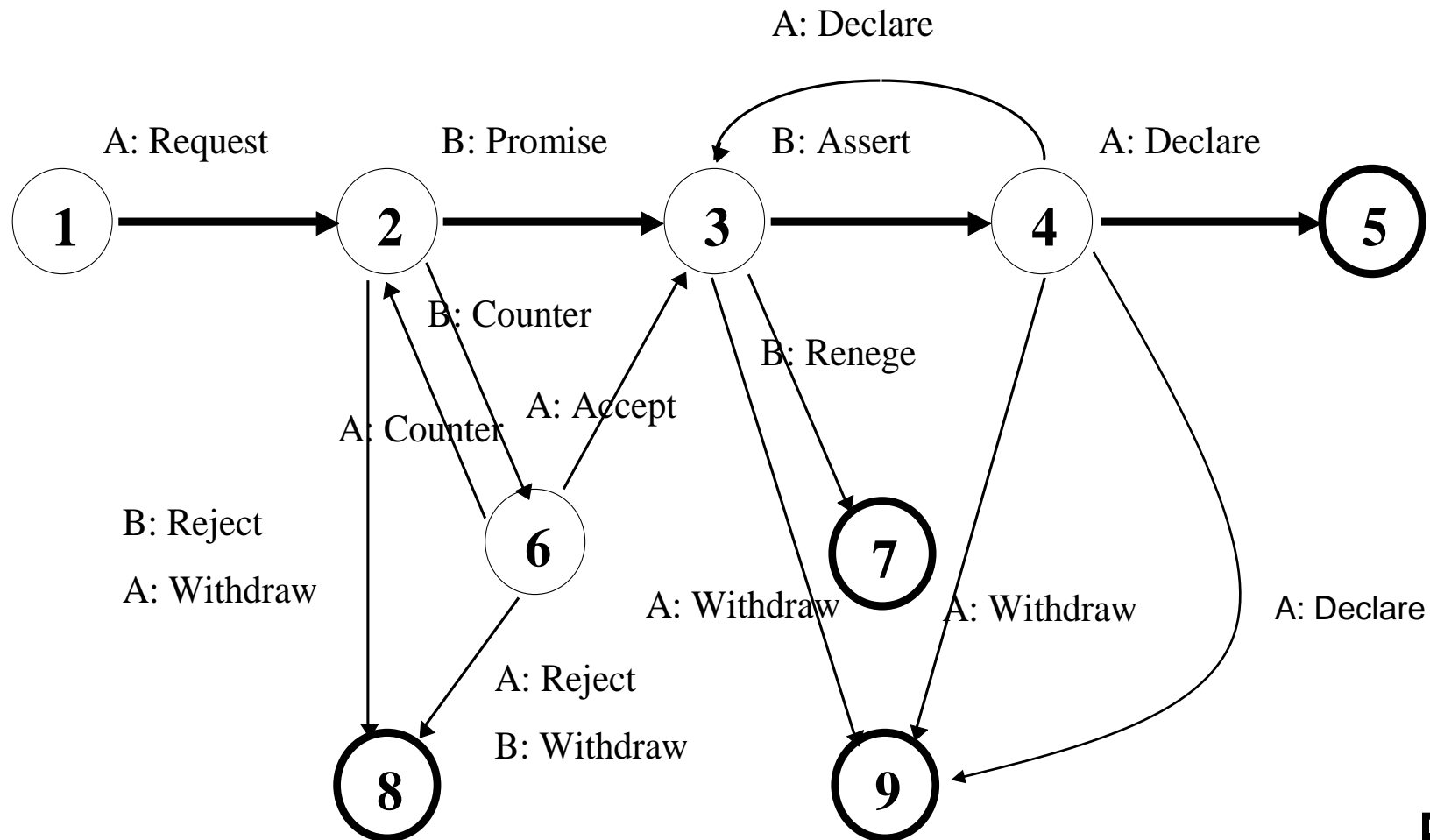
# Categories of Speech Acts (Searle)

- **Directive - Conference organizer to researchers:**  
"Please send us your paper before 1. March"
- **Commissive- Researcher:** "I promise to send you the paper before the deadline"
- **Assertive- Researcher:** "Please find attached my paper"
- **Declarative - Conference organizer to researcher:**  
"Your paper is accepted"
- **Expressive - Conference organizer to researcher:**  
"Congratulations"

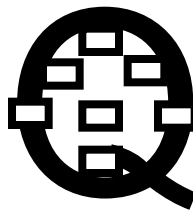
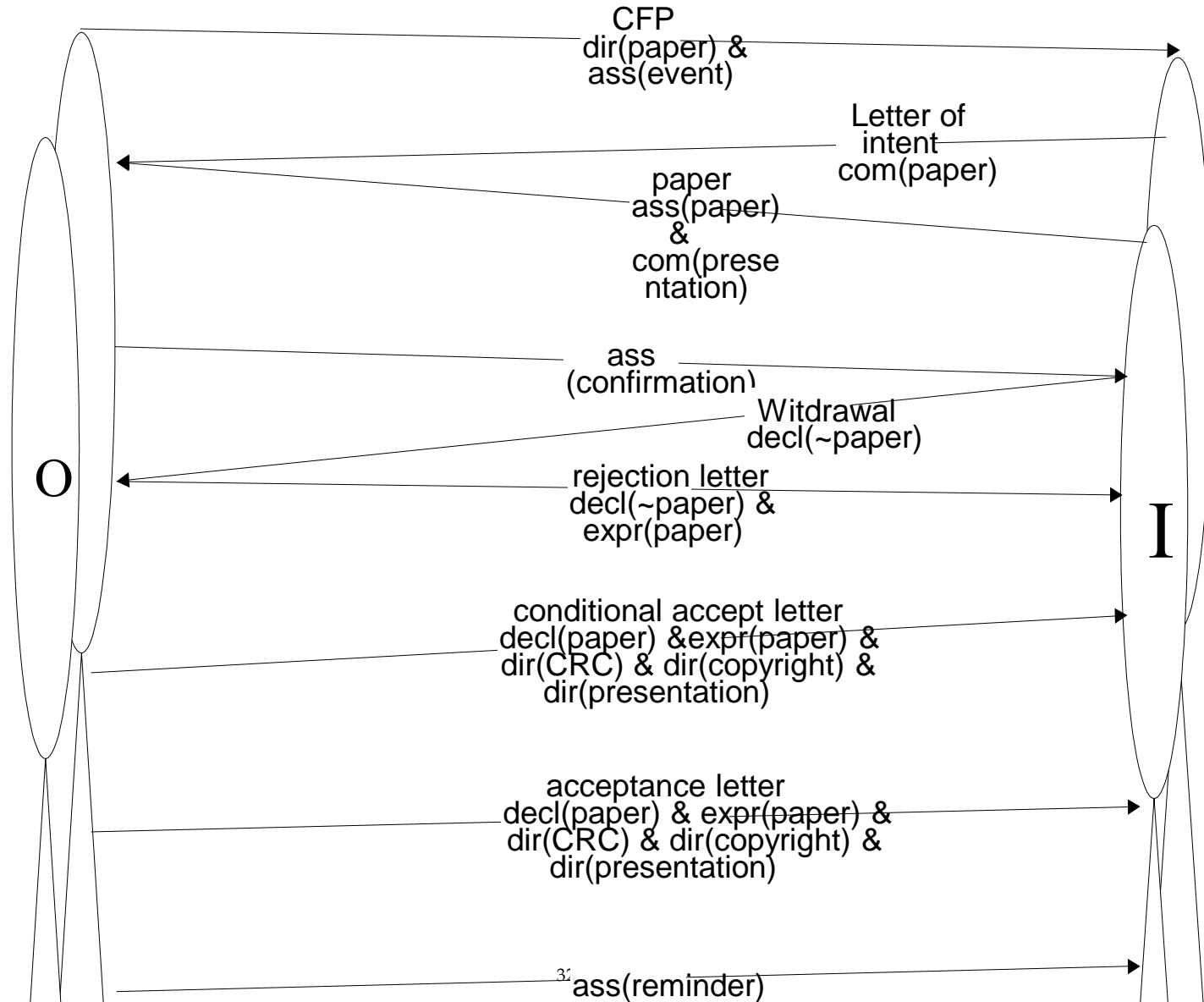


# Winograd & Flores

## Conversation for Action



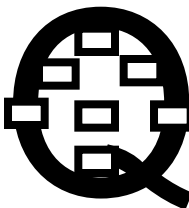
# Speech acts relative to paper process



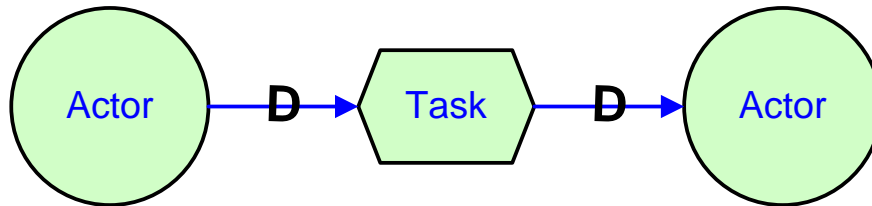


## 7. Actor and role perspective

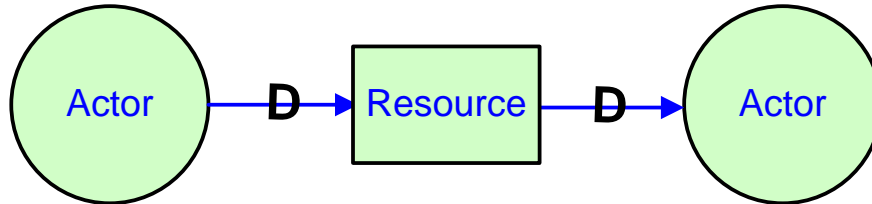
- Description of the *active* parts in an organizational or system structure
- Main concepts: Actors and roles
- Proposed definition
  - An actor is a phenomena that influence the history of another actor
  - A role is the behavior that is expected by an actor (by other actors) when filling the role
- Based originally on
  - work in AI (Intelligent agents)
  - object-oriented programming
  - Modeling of organizational structures, group dynamics, social network analysis etc.
  - Economic theory
- More on this perspective later in the course (i\* and as part of e.g. Archimate and EEML)



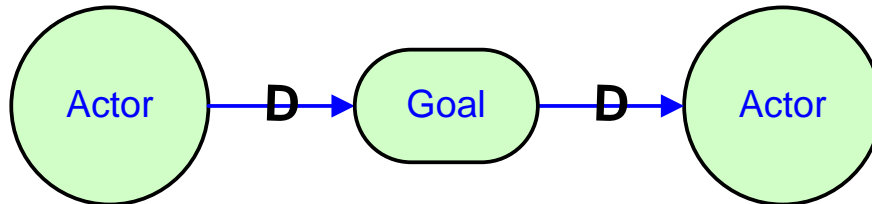
# I\* dependencies



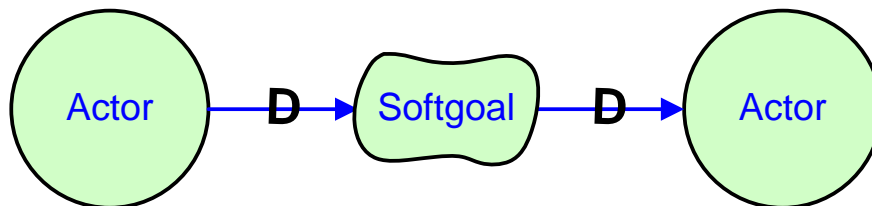
Task Dependency



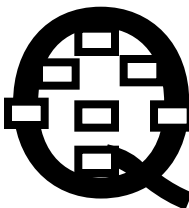
Resource Dependency



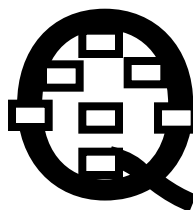
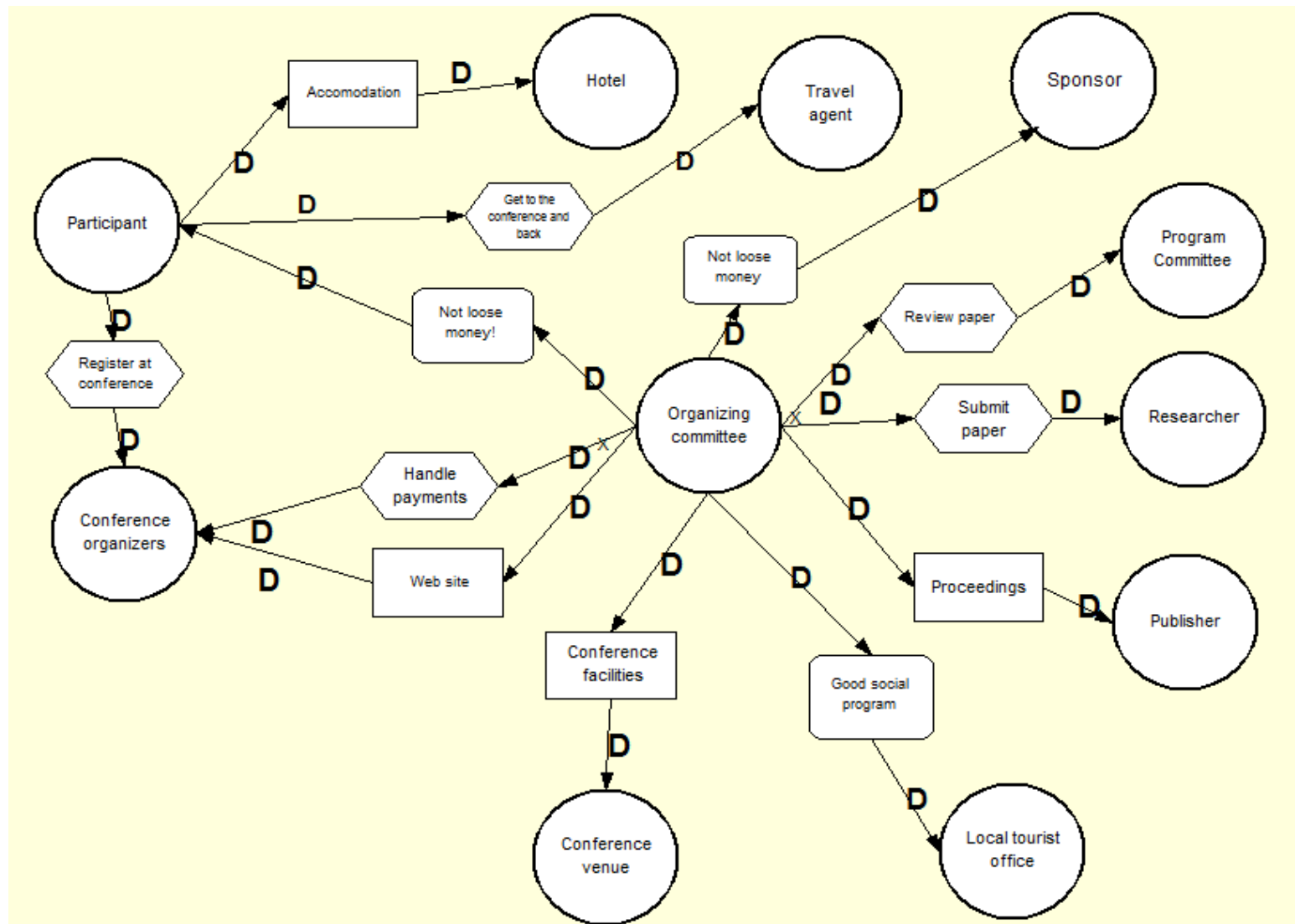
Goal Dependency



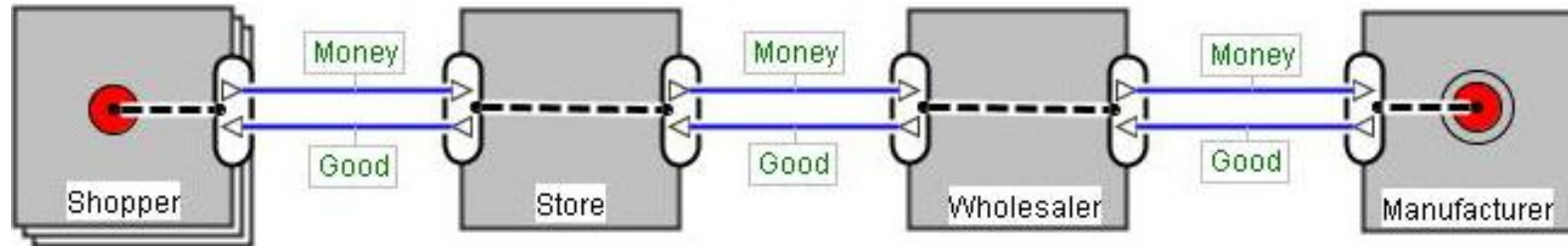
Soft-goal Dependency



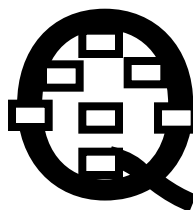
# Dependencies between actors in conference arrangements



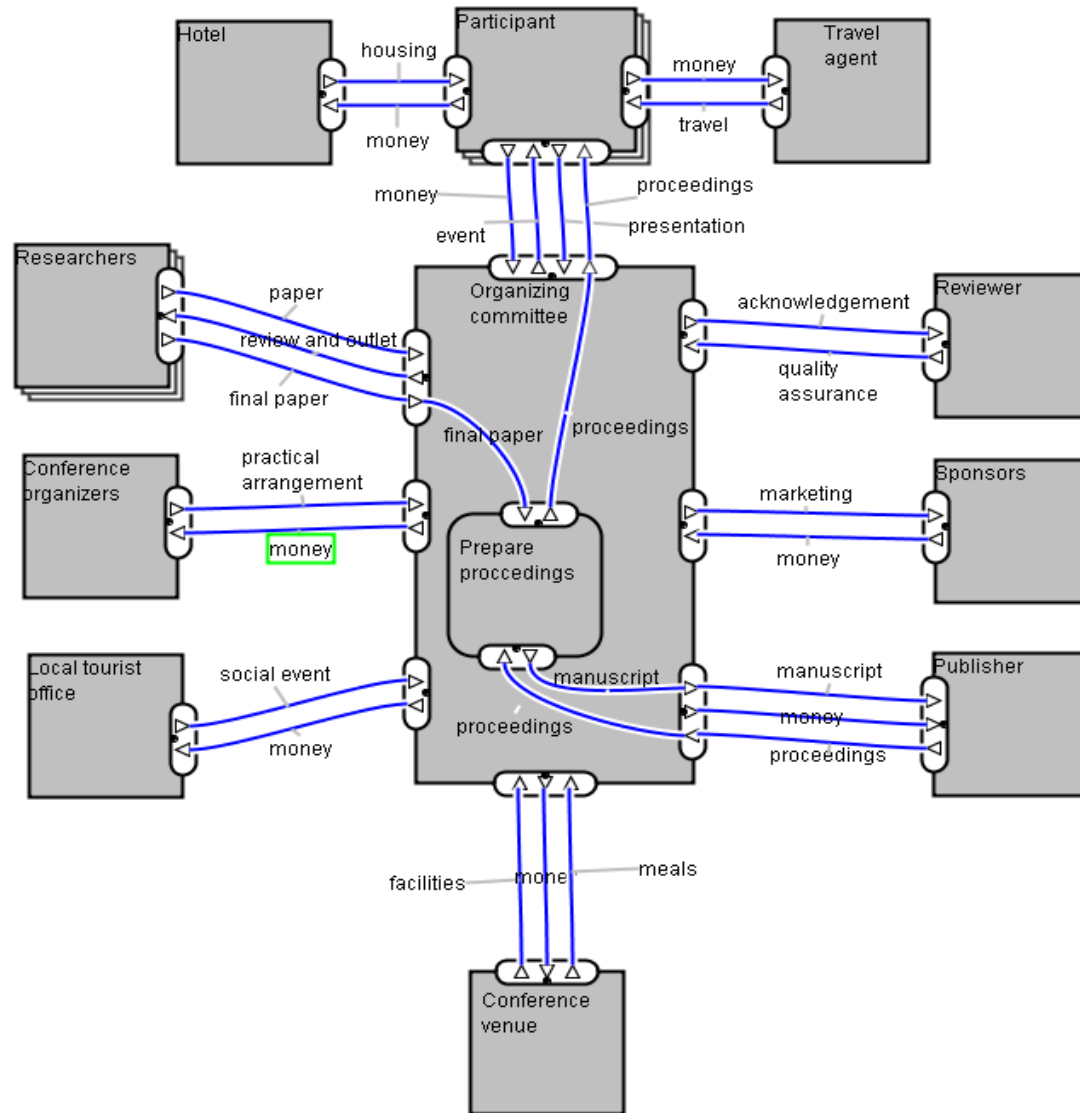
# e<sup>3</sup>value



Legend	Actor	Value interface	Value port	Value Exchange	Start stimulus	Connect. element	End stimulus

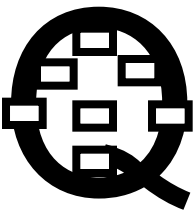


# Example from conference case in e<sup>3</sup>value

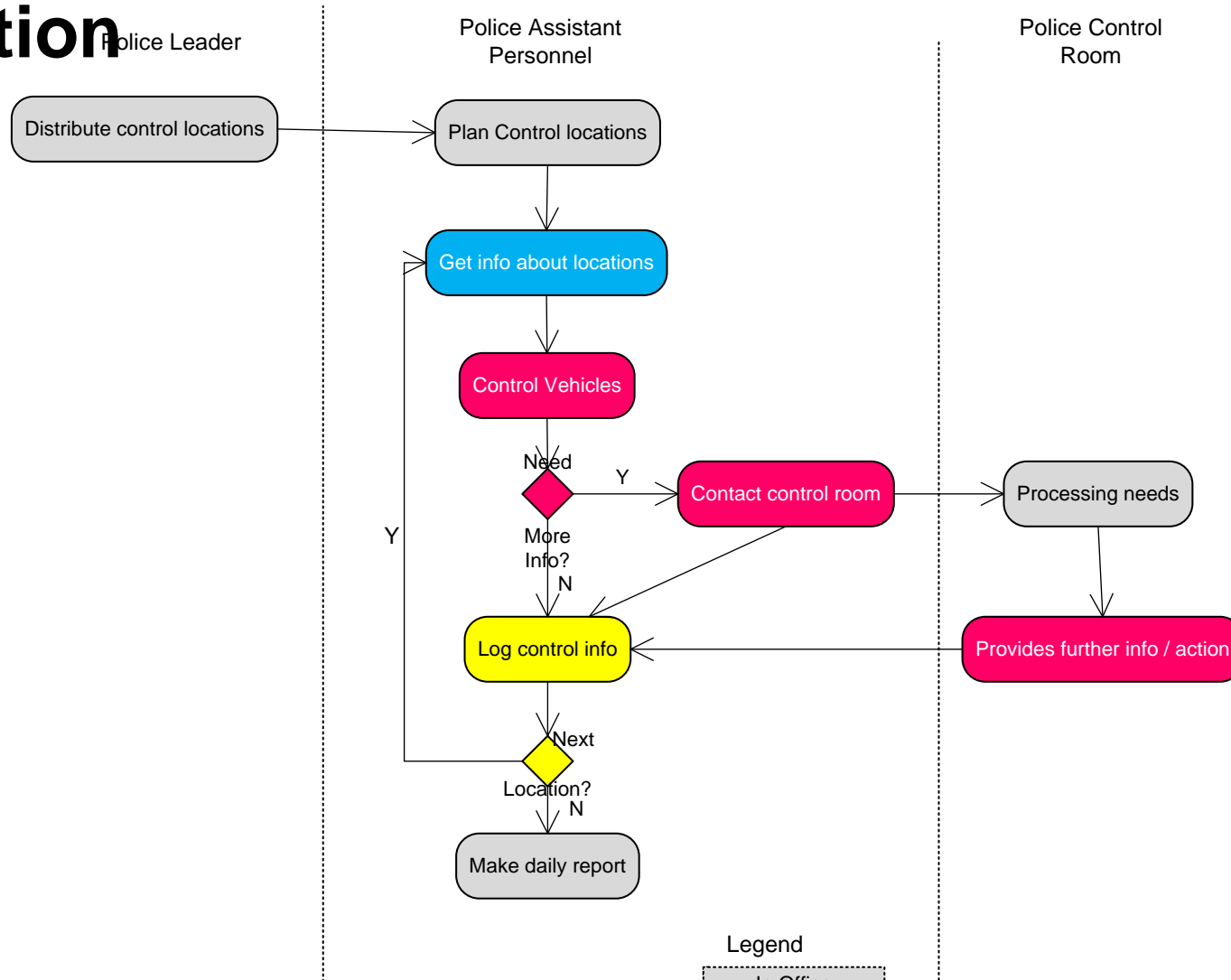


## 8. Topological perspective

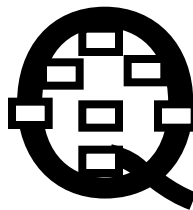
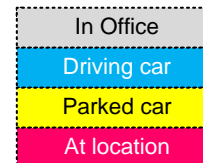
- **Relates to the topological ordering between the different concepts (where)**
- **Place (type of space e.g. Auditorium) vs. Space (e.g. R93).**
- **Limited support of modelling this aspect in conceptual modelling**
- **Included on a high level in enterprise architecture and enterprise modelling (e.g. Zachman, Archimate)**
- **'Where' is increasingly relevant in organisations**
  - Outsourcing
  - Supply Chain Management/Logistics
  - Virtual organizations
  - Apps and Mobile information systems
- **And it is possible to utilize 'where' to a larger degree (also real time) to know where users, equipment and goods should be, are or where at a certain time**
  - Tracking (RFID, UWB, GPS, GSM, WiFi, Ultrasound...)
  - Internet of Things (IoT)



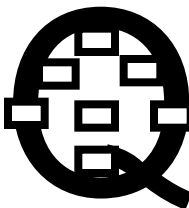
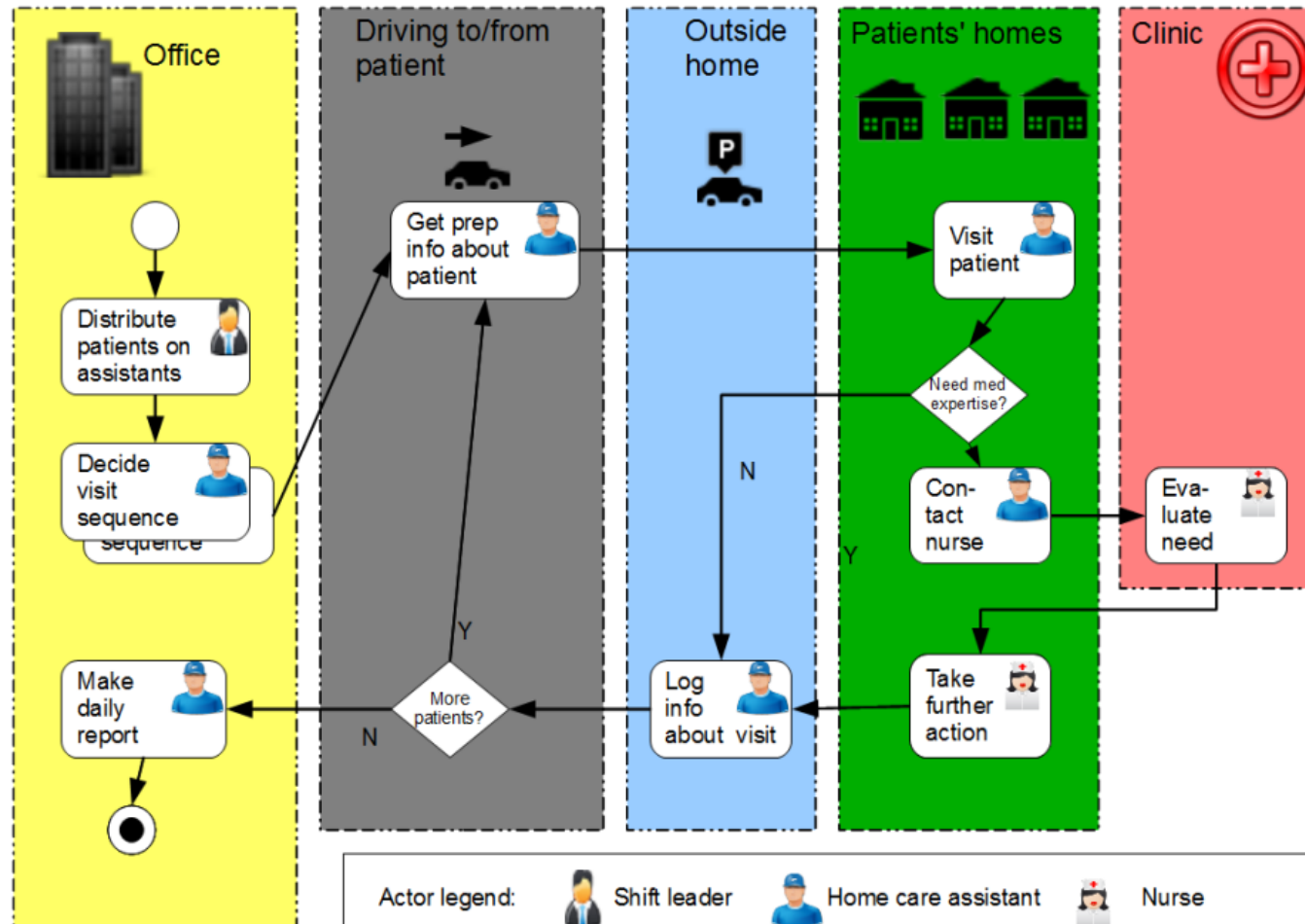
# Place-oriented modelling - representing location



## Legend

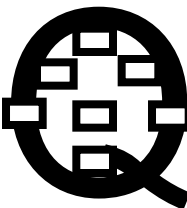
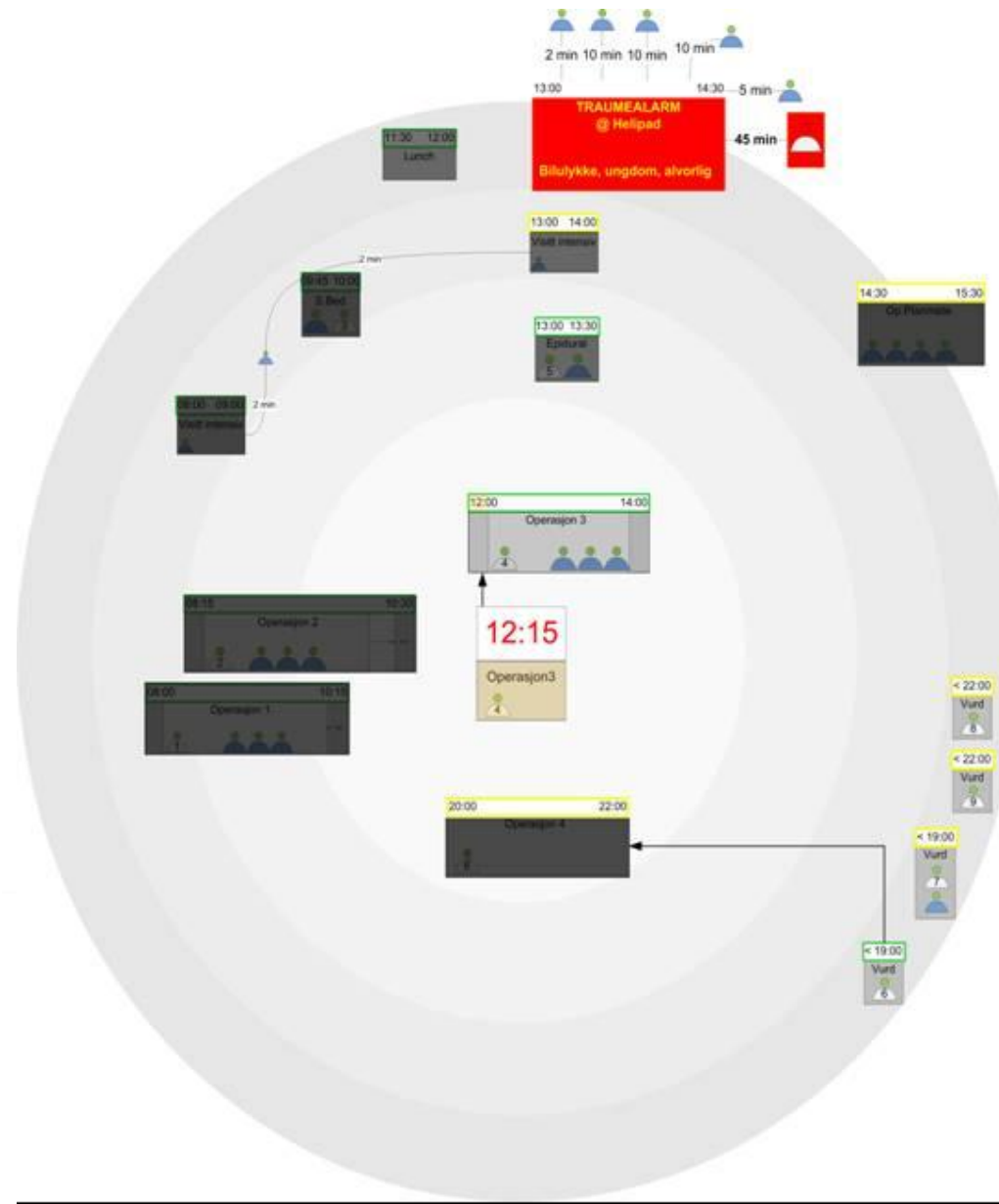


# Place-oriented modelling – structure according to location

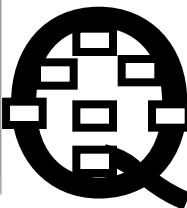
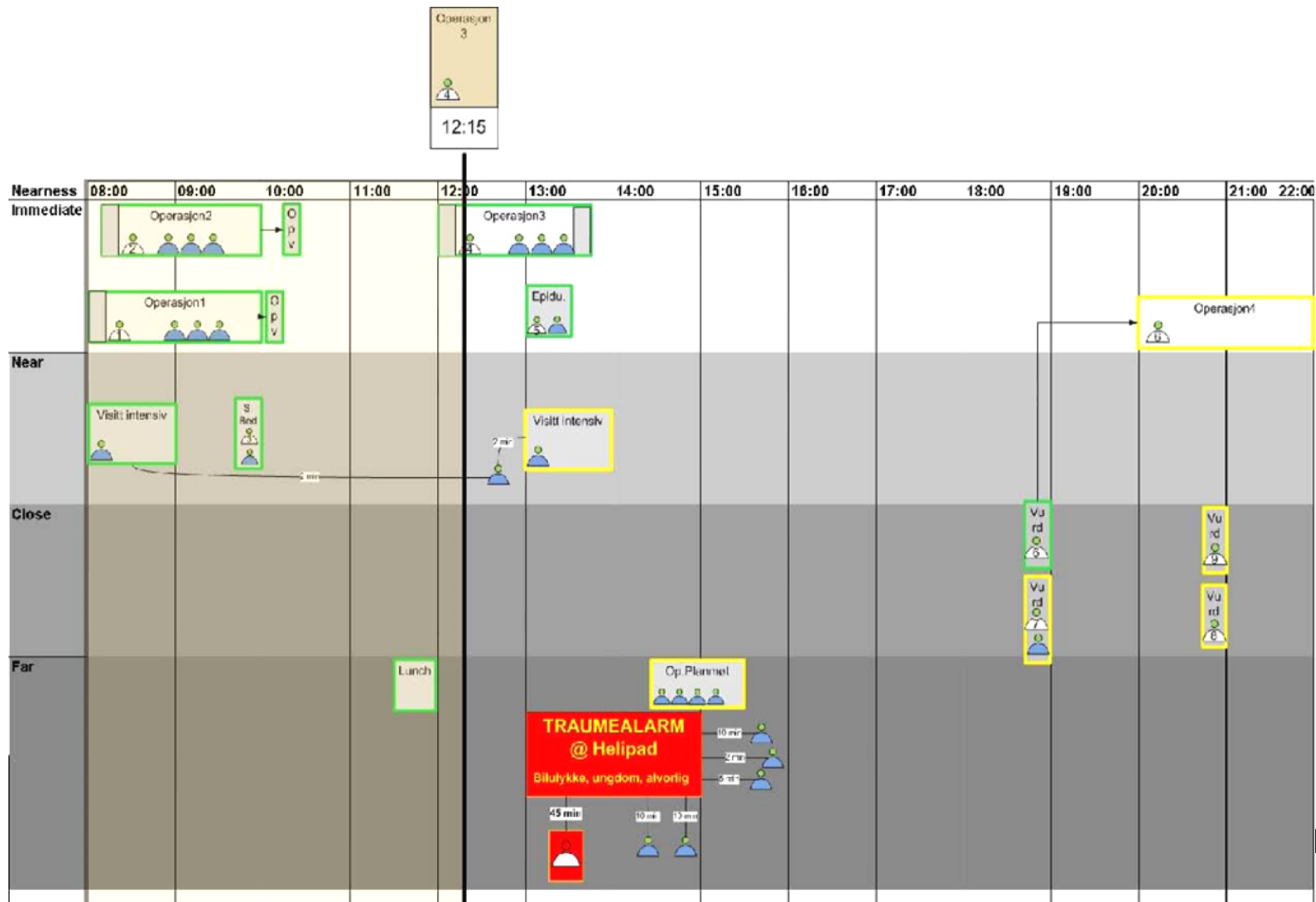




# Space-oriented modelling



# Space and time combine with conceptual information information



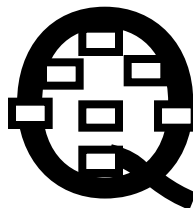
# No language perspective covers all situations/domains in itself

- **Different approaches for combining different perspectives**

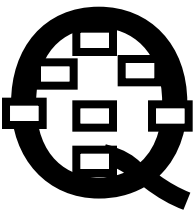
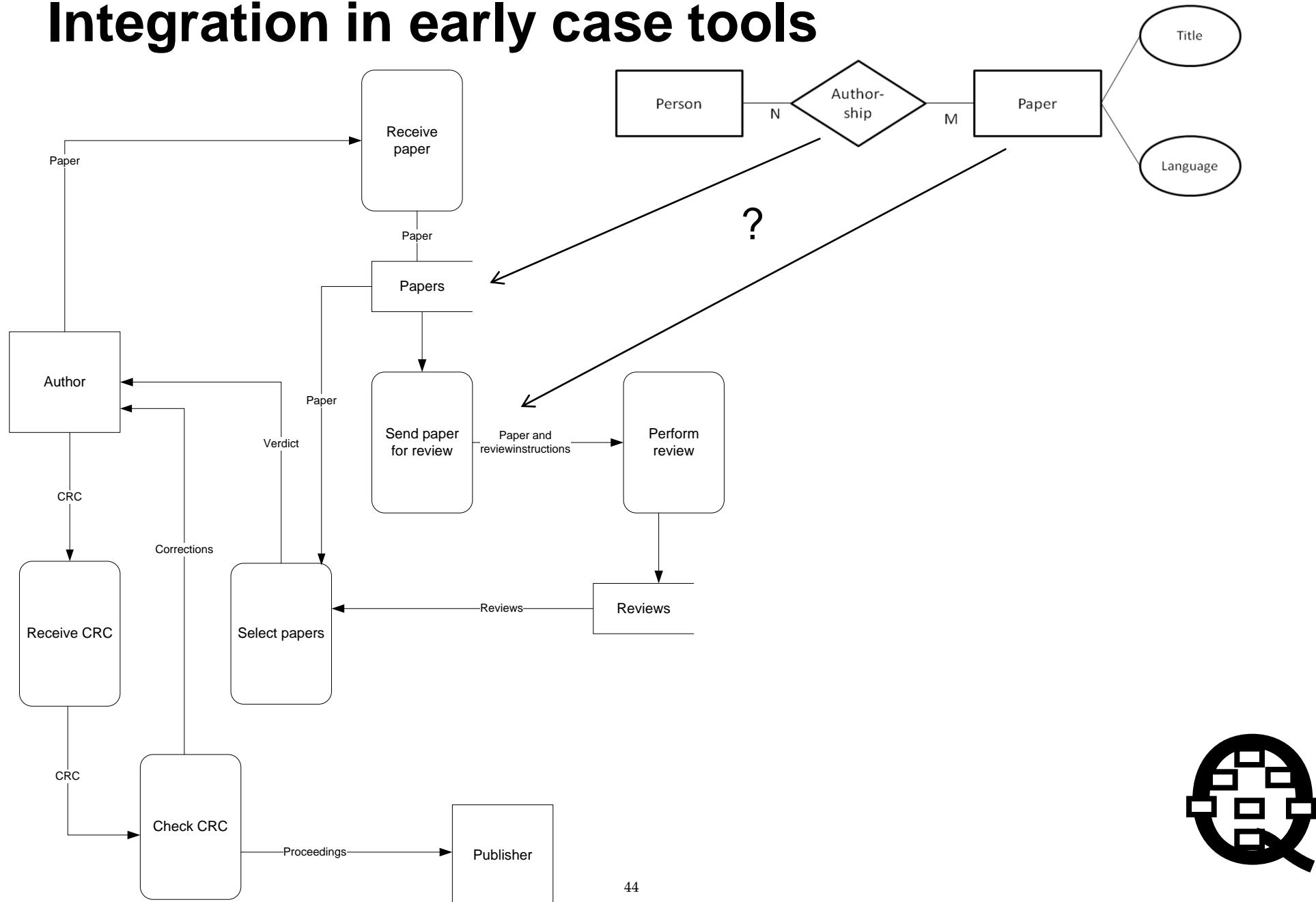
- 1 Use many different languages in parallel without explicit integration (**early CASE-tools**, Visio)
- 2 Integrate existing languages (**UML**, EEML, MEAF)
- 3 Develop completely new approaches (***i\****, e<sup>3</sup>Value, SeeMe, **Archimate**)
- 4 Develop framework to be able to create and adapt languages on an as-needed basis (Metaedit, **METIS**/Trous Architect)

- **Perspective-less modeling**

- Facet modeling (Opdahl and Sindre 1997)
- **GEMAL** (Krogstie 2012)









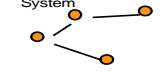





























# Integration in early case tools



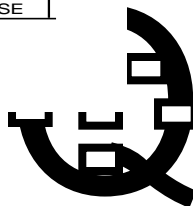
# Zachman's Enterprise Architecture Framework

## ENTERPRISE ARCHITECTURE - A FRAMEWORK <sup>TM</sup>

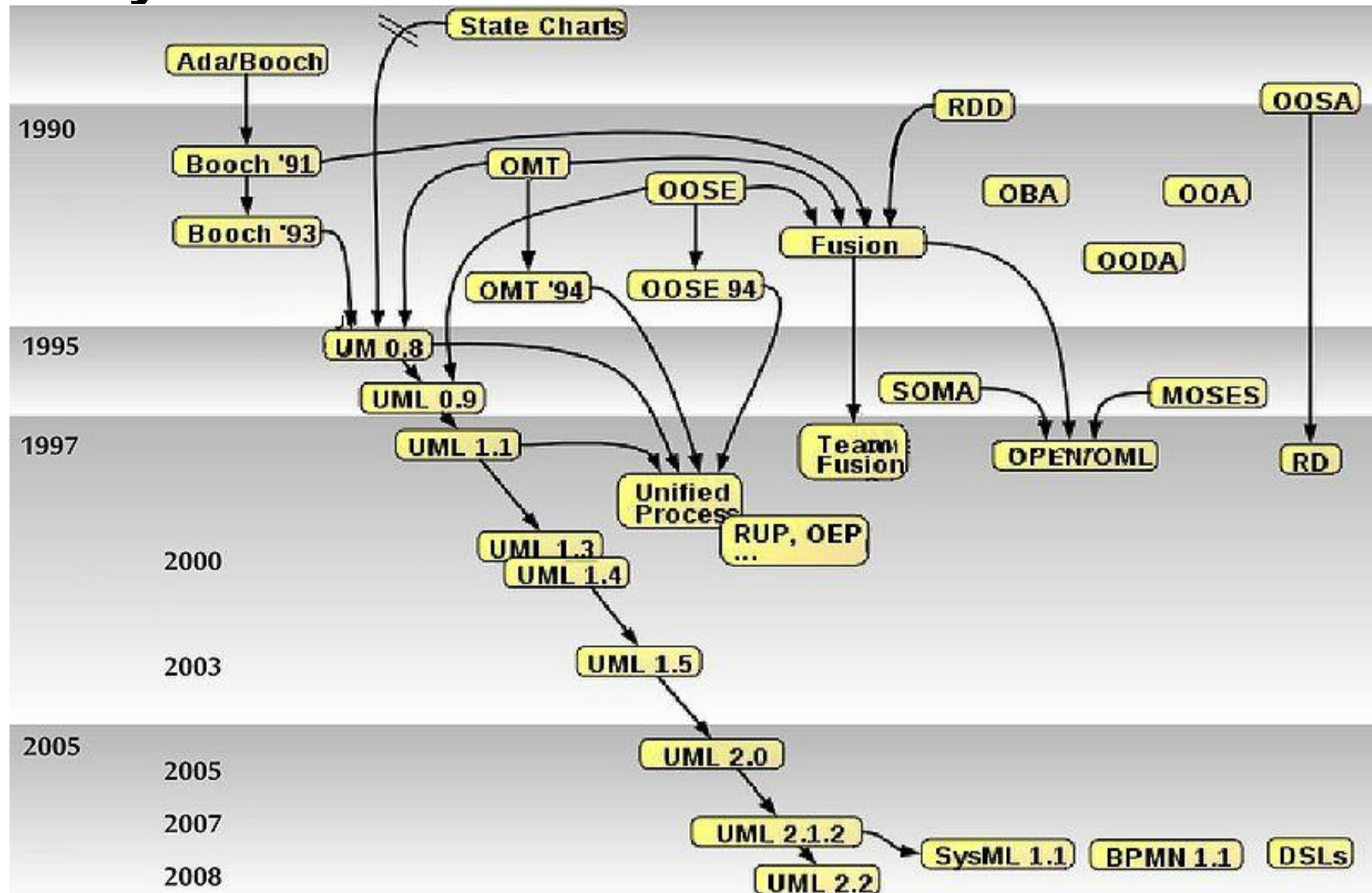
	DATA <i>What</i>	FUNCTION <i>How</i>	<b>Perspectives/aspects</b>		TIME <i>When</i>	MOTIVATION <i>Why</i>	
SCOPE (CONTEXTUAL)	List of Things Important to the Business 	List of Processes the Business Performs 	List of Locations in which the Business Operates 	List of Organizations Important to the Business 	List of Events Significant to the Business 	List of Business Goals/Critical Success Factor 	SCOPE (CONTEXTUAL)
<i>Planner</i>	ENTITY = Class of Business Thing 	Function = Class of Business Process 	Node = Major Business Location 	People = Major Organizations 	Time = Major Business Event 	Ends/Mean=Major Bus. Goal/Critical Success Factor 	<i>Planner</i>
ENTERPRISE MODEL (CONCEPTUAL)	e.g. Semantic Model  Ent = Business Entity Rein = 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)
<i>Owner</i>	e.g. Logical Data Model  Ent = Data Entity Rein = 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)
<i>Designer</i>	e.g. Physical Data Model  Ent = Segment/Table/etc. Rein = 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)
<i>Builder</i>	e.g. Data Definition  Ent = Field Rein = 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 REPRESENTATIONS (OUT-OF-CONTEXT)
<i>Sub-Contractor</i>							
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

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

21



# History of UML



# Integration of existing languages, e.g. UML

## ■ Structural

- Class diagram (without methods). The perspective is well supported, although not as well as in languages made specifically for this purpose

## ■ Object-oriented

- Object and Class diagrams (with methods)
- Sequence and collaboration diagrams

## ■ Functional

- Use cases for informal modeling
- Activity diagrams

## ■ Behavioral

- State, sequence and collaboration diagrams

## ■ Rule oriented

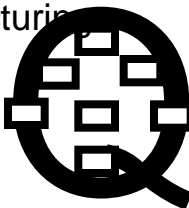
- No goal-hierarchies, difficult to represent rules across classes. Individual rules in an additional language (OCL)

## ■ Actor/role oriented

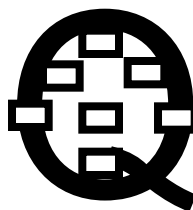
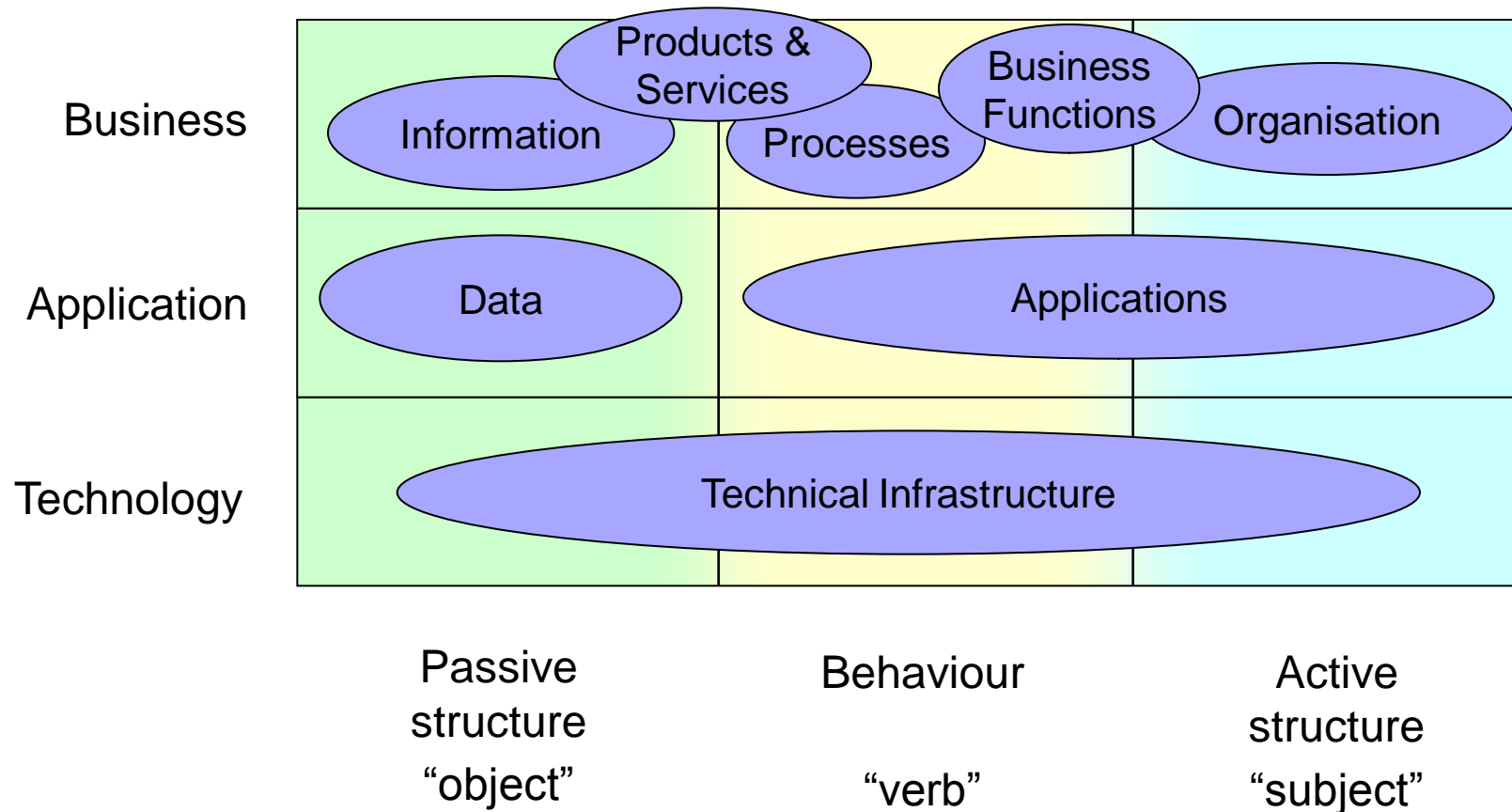
- Limited coverage
- Actors in use case diagrams, swimlanes in activity diagrams, but few relationships for structuring actors and roles

## ■ Social communication and Topological

- Not covered



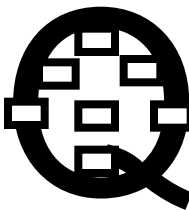
# Layers and perspectives/aspects in Archimate

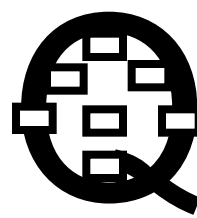
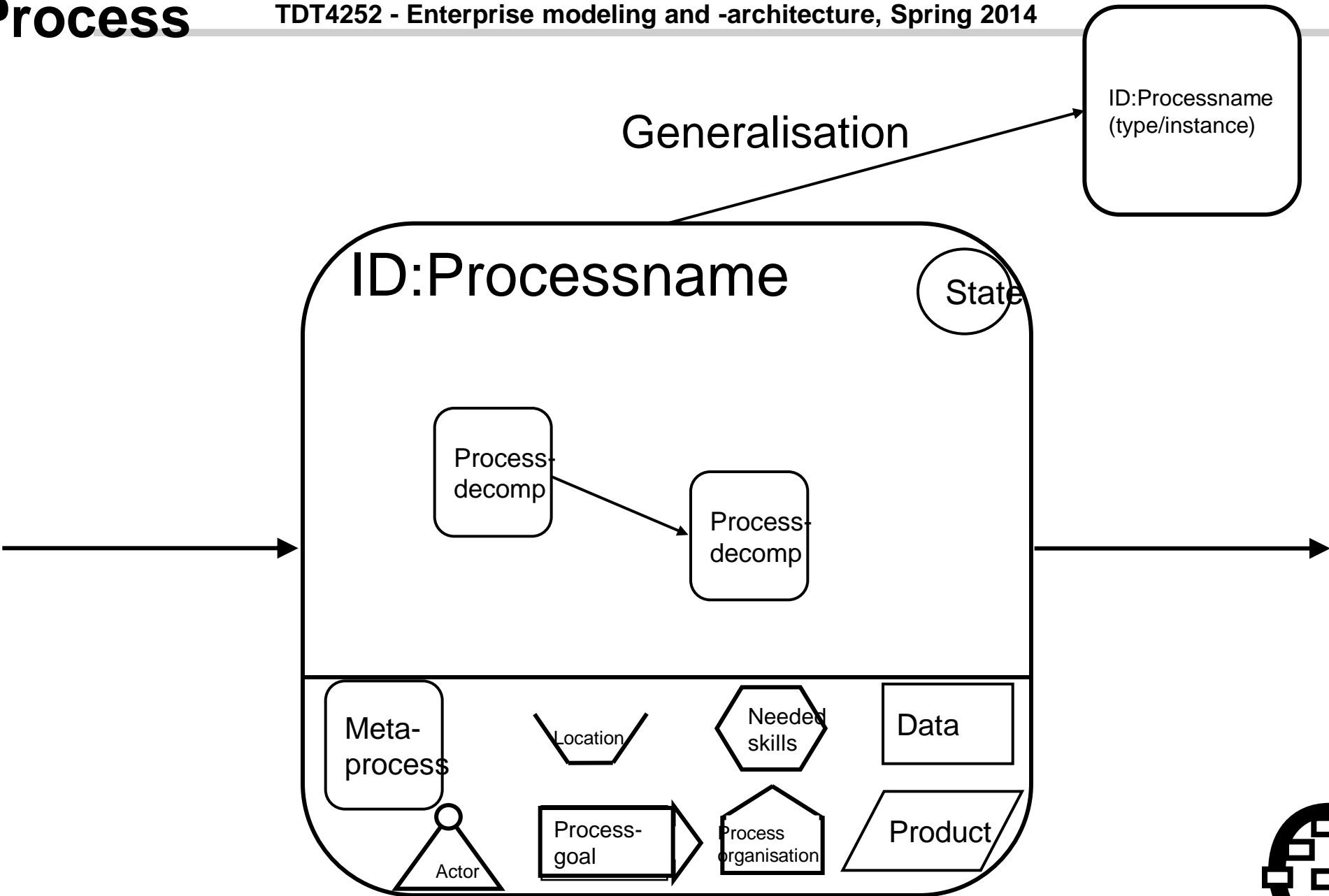




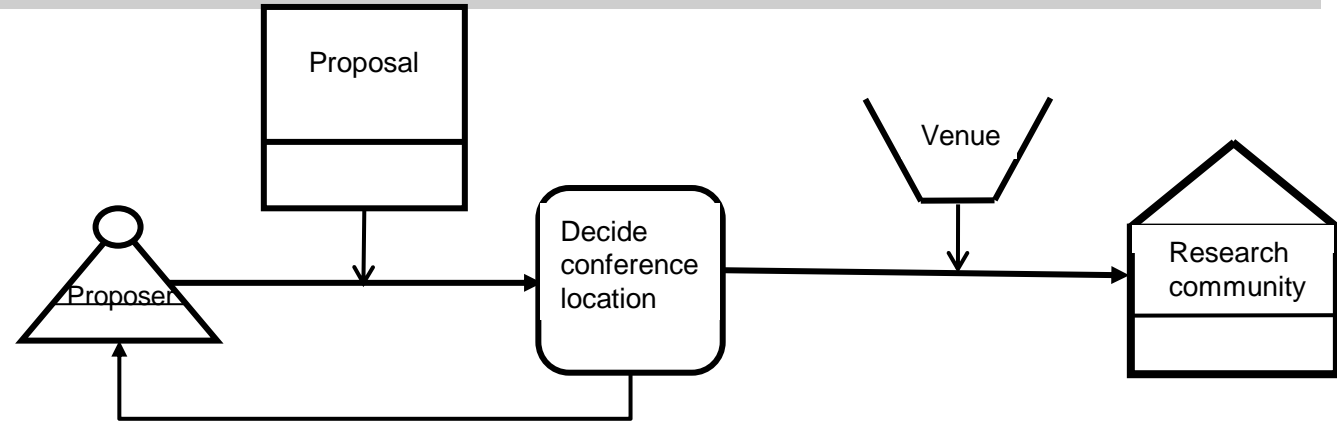
# Modelling perspectives(facets) in GEMAL – General Enterprise Modeling and Activation Language

- Goal
- Process
- Organisation
- Product
- System
- Person
- Capability
- Location
  
- All main concepts (facets of thing) on both type and instance level
  - Same symbol on both type and instance level, instance-level concepts gray background
  - E.g. Personrole and person
- All concepts can be looked upon in *context* of all others. E.g. a personrole in the context of an organization is a position in this organization
- All can be linked through generalization and aggregation hierarchies
- Meta-level structures possible for all concepts
- Additional relationships between concepts (Precede, communicate, support,... Annotated by any of the 8 concepts on type or instance level)
- Properties on all concepts, such as name, description, modality, also properties that are context relative (e.g. start/end-time which indicates the relevant lifetime of the concept)
- 8 concepts, 4 generic abstraction mechanisms, 5 generic relationships freely combined-> Potentially 1000 + concepts

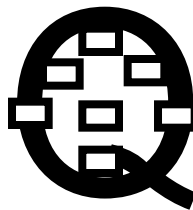
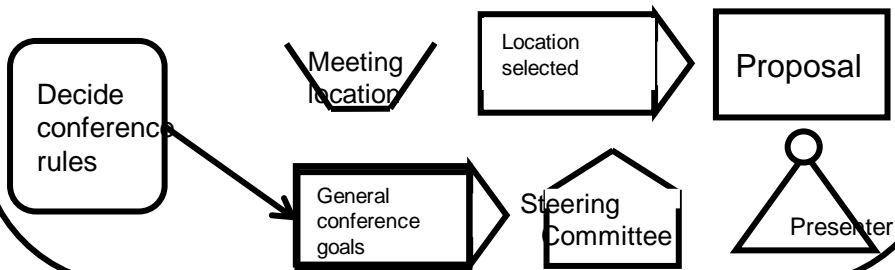
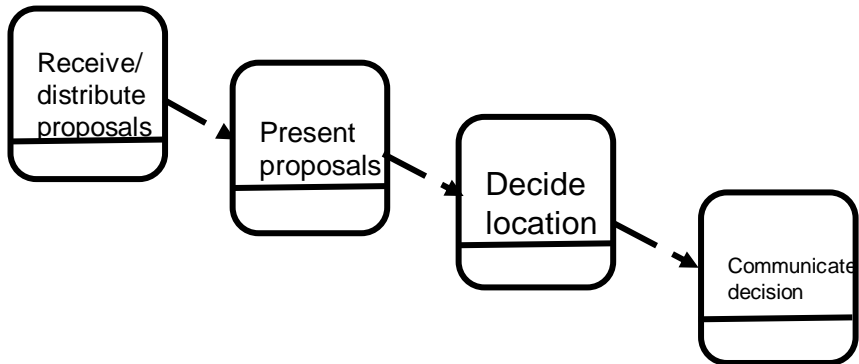




# Process

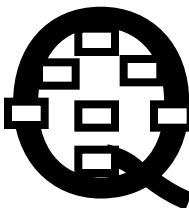


## Decide conference location



# Central problems of multi-perspective approaches

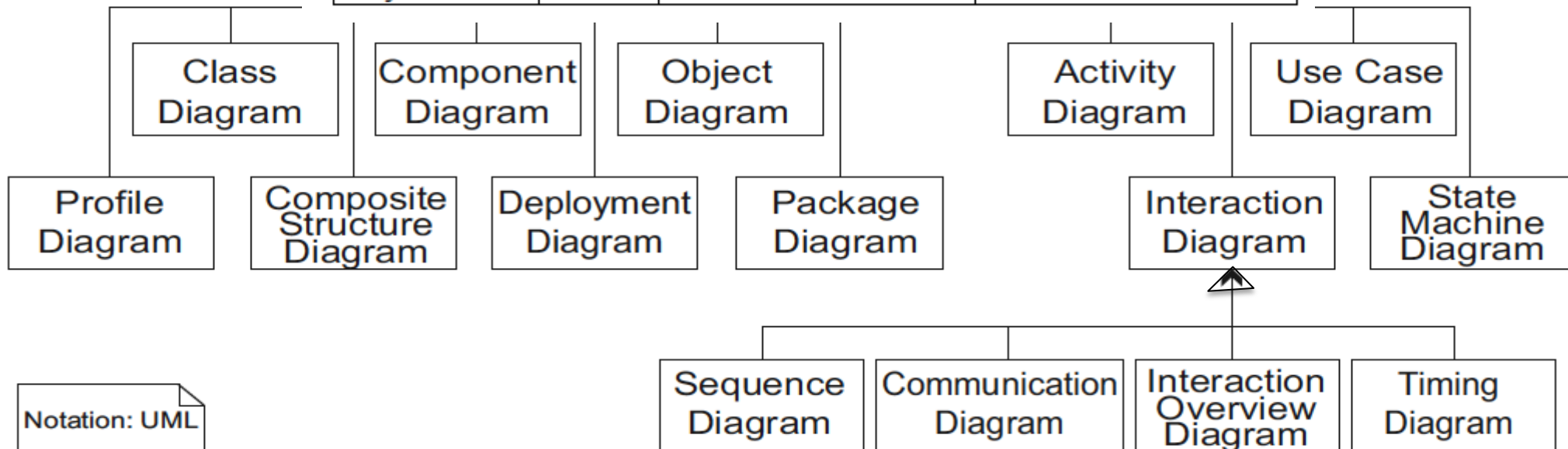
- **Structuring the overall language in sub-languages to create sub-models**
- **Alignment of models in different languages**
  - Integrated meta-model
- **How to see the connection between different sub-models and diagrams?**
  - Tool support
- **Consistent notation**
- **Does it make the language more difficult to learn?**
  - "core" UML vs. extensions and profiles
  - Viewing mechanisms in tools



# Overview of the UML diagrams

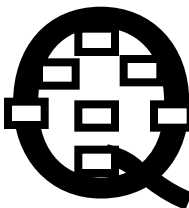
Table 1: UML Overall Results (non domain specific)

Construct	Mean	Standard Deviation	% "Yes" for Kernel
<i>Class</i>	<i>1.00</i>	<i>0.00</i>	<i>100.0%</i>
<i>Use Case</i>	<i>1.61</i>	<i>0.79</i>	<i>90.9%</i>
<i>Sequence</i>	<i>1.73</i>	<i>0.70</i>	<i>95.5%</i>
<i>Statechart</i>	<i>1.81</i>	<i>0.51</i>	<i>100.0%</i>
Component	2.31	0.70	31.8%
Activity	2.41	0.55	27.3%
Collaboration	2.57	0.87	22.7%
Deployment	2.69	0.75	9.1%
Object	3.00	0.86	9.1%

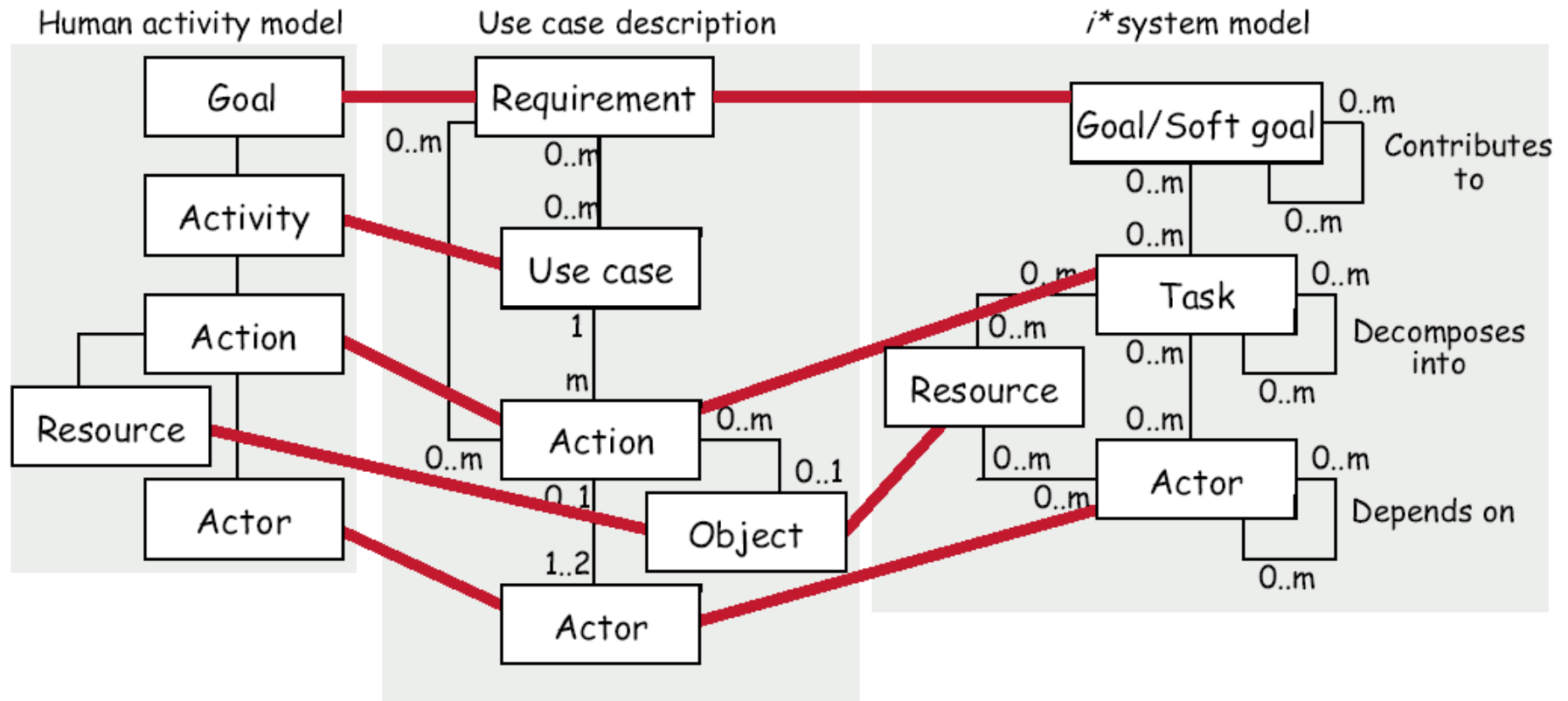


# Overview of UML Diagrams

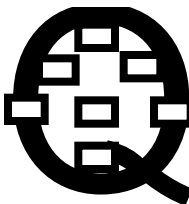
- There is no official UML diagram overview or diagram grouping.
- Although UML models and the repository underlying all diagrams are defined in UML, the definition of diagrams (i.e. special views of the repository) are relatively free
- UML lacks a summary diagram, and the relationships between different diagram types are unclear



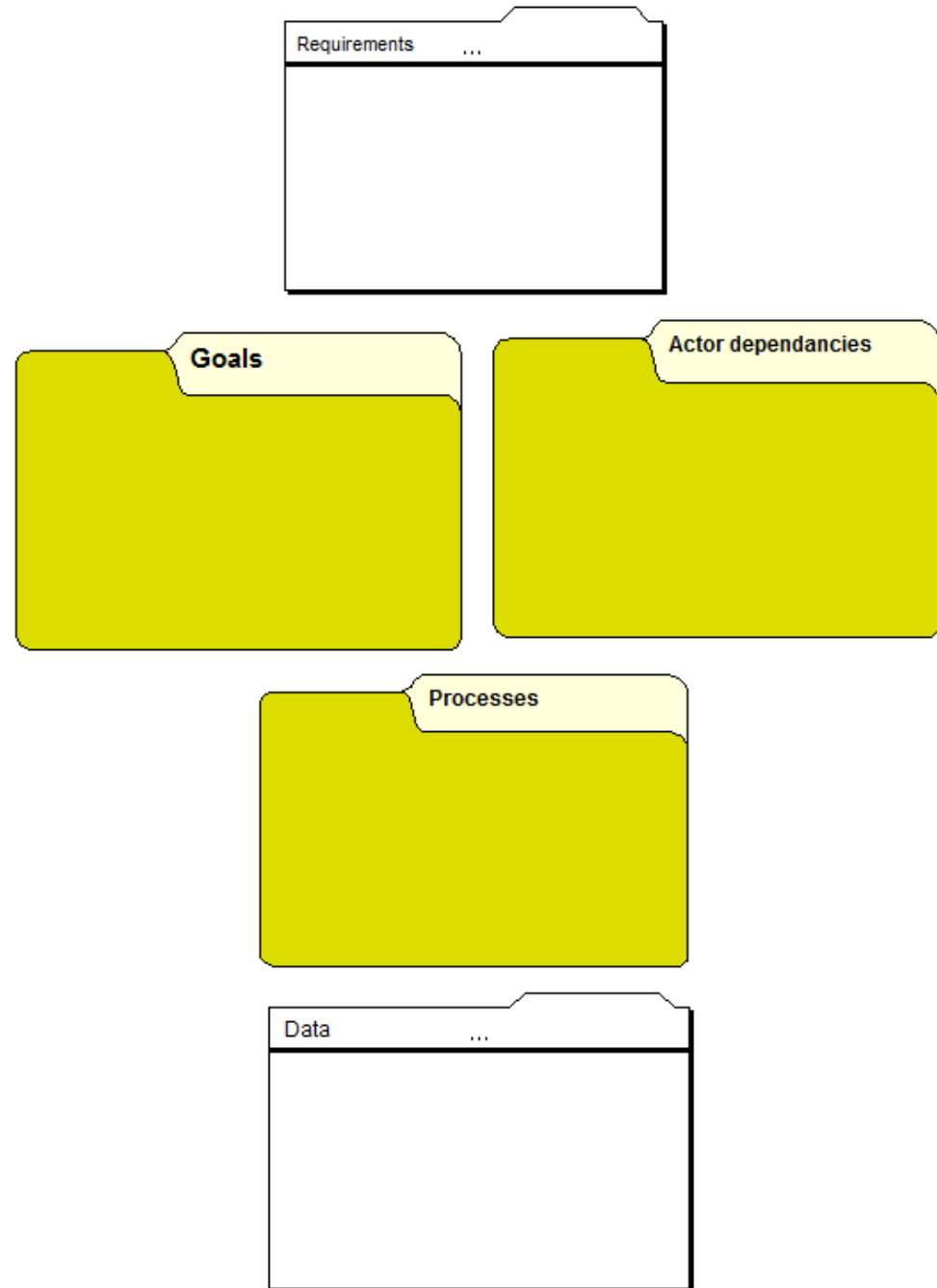
# RESCUE Concept Meta-model



**Fig. 5.** RESCUE concept meta-model as a UML class diagram showing mappings between constructs in the 3 model types.



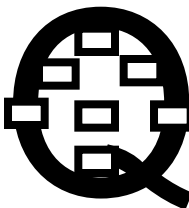
# Folders structuring sub-models in METIS





# Summary

- It is possible to model ones perception of the reality from many different perspectives
- A model expressed in a language of a specific perspective emphasize a certain way of structuring
- There is no perspectives that is best in all cases
- Need often to combine perspectives in an integrated manner
- What to include depends on the goal/purpose of modeling, domain and the stakeholders of the modeling



# **Modeling perspectives in conceptual and enterprise modeling**

**Final questions ?**

**John Krogstie**