

# Enterprise Engineering

An aerial photograph of a modern, multi-story white building complex, likely a university or corporate campus, situated on a hillside overlooking the sea. The building features a curved facade and numerous balconies. In the foreground, there is a green roof with small trees. To the right, a swimming pool and a beach area are visible. The background shows a blue sea and distant mountains under a clear sky.

**Master Class 2014**

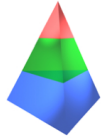
Jan Dietz  
Jan Hoogervorst



**Prelude**

**and**

**Outline**



# Engineering automobiles

## Construction

Assembly of mainly mechanical and electrical parts.

## Operating principle

Rolling on surfaces, being propelled by some power source.

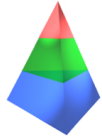
## Power source

engine(s) fuelled by fossil fuels (gasoline, diesel, ...) or electricity.

## Operating Theory

Mechanics (gravity, friction).





# Engineering aircrafts

## Construction

Assembly of mainly mechanical and electrical parts.

## Operating Principle

Gliding on air, being propelled by some power source.

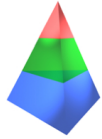
## Power source

engine(s) fuelled by fossil fuels (kerosene).

## Operating Theory

Aerodynamics (lift by wings).





# Engineering enterprises

## Construction

?

## Operating Principle

?

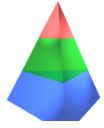
## Power source

?

## Operating Theory

?





# The CIAO! Tree

**APPLICATIONS**

**METHODS**

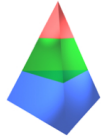
**THEORIES**

**philosophical**

**ontological**

**technological**

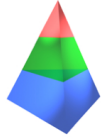
**ideological**



# The EE Theory Framework

<p><b>Ideological Theories</b> <i>selecting the things to make</i> <b>politics</b> EE-theories: <math>\sigma</math>-theory</p>	<p><b>Technological Theories</b> <i>designing and making things</i> <b>analysis and synthesis</b> EE-theories: <math>\beta</math>-theory, <math>\nu</math>-theory</p>
<p><b>Ontological Theories</b> <i>understanding the nature of things and their use</i> <b>explanation and prediction</b> EE-theories: <math>\phi</math>-theory, <math>\delta</math>-theory, <math>\pi</math>-theory, <math>\psi</math>-theory, <math>\tau</math>-theory</p>	
<p><b>Philosophical Theories</b> <i>understanding thinking</i> epistemology, mathematics, phenomenology, logic EE-theories: <math>\omega</math>-theory</p>	



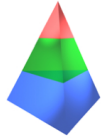


# The importance of a proper theory

**“Whether you can observe a thing or not depends on the theory that you use. It is the theory that decides what can be observed.”**

(Albert Einstein)





# The CIAO! Network



東京工業大学  
Tokyo Institute of Technology



Research Almaden, USA



CTU Prague



Delft University of Technology



TU Lisboa



Moscow and Nizhniy Novgorod, Russia



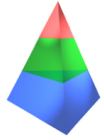
Public Research Centre, Luxembourg



University of St. Gallen

CIAO!





# Outline

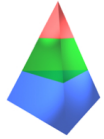
$\delta$ -theory and  $\pi$ -theory

$\psi$ -theory

$\tau$ -theory

$\beta$ -theory





# Outline

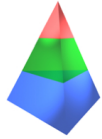
**$\delta$ -theory and  $\pi$ -theory**

$\psi$ -theory

$\tau$ -theory

$\beta$ -theory

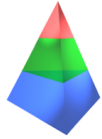




# The EE Theory Framework

<p><b>Ideological Theories</b> <i>selecting the things to make</i> <b>politics</b> EE-theories: <math>\sigma</math>-theory</p>	<p><b>Technological Theories</b> <i>designing and making things</i> <b>analysis and synthesis</b> EE-theories: <math>\beta</math>-theory, <math>\nu</math>-theory</p>
<p><b>Ontological Theories</b> <i>understanding the nature of things and their use</i> <b>explanation and prediction</b> EE-theories: <math>\phi</math>-theory, <math>\delta</math>-theory, <math>\pi</math>-theory, <math>\psi</math>-theory, <math>\tau</math>-theory</p>	
<p><b>Philosophical Theories</b> <i>understanding thinking</i> epistemology, mathematics, phenomenology, logic EE-theories: <math>\omega</math>-theory</p>	





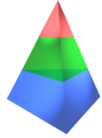
# Ontological theories: the $\delta$ -theory

The  $\delta$ -theory ( $\delta$  is pronounced as DELTA, standing for Discrete Event in Linear Time Automaton) is a theory about the statics, kinematics, and dynamics of state machines.

It provides the basis for an appropriate understanding of what is commonly referred to by terms like “system”, “state”, “event”, and “process”.

The  $\delta$ -theory is rooted in automata theory [Hopcroft and Ullman].





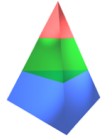
# Ontological theories: the $\pi$ -theory

The  $\pi$ -theory ( $\pi$  is pronounced as PI, standing for Performance in Interaction) is a theory about the ontological essence of discrete event systems.

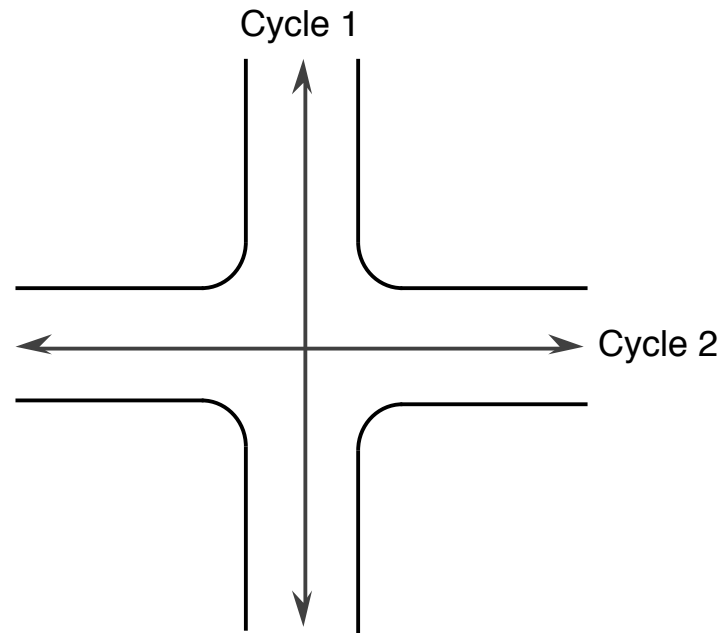
It clarifies and explains the construction and operation of technical, i.e. non-social, systems.

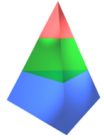
The  $\pi$ -theory is rooted in the  $\delta$ -theory, systemic ontology [Bunge] and discrete event systems [Cassandras and Lafortune]



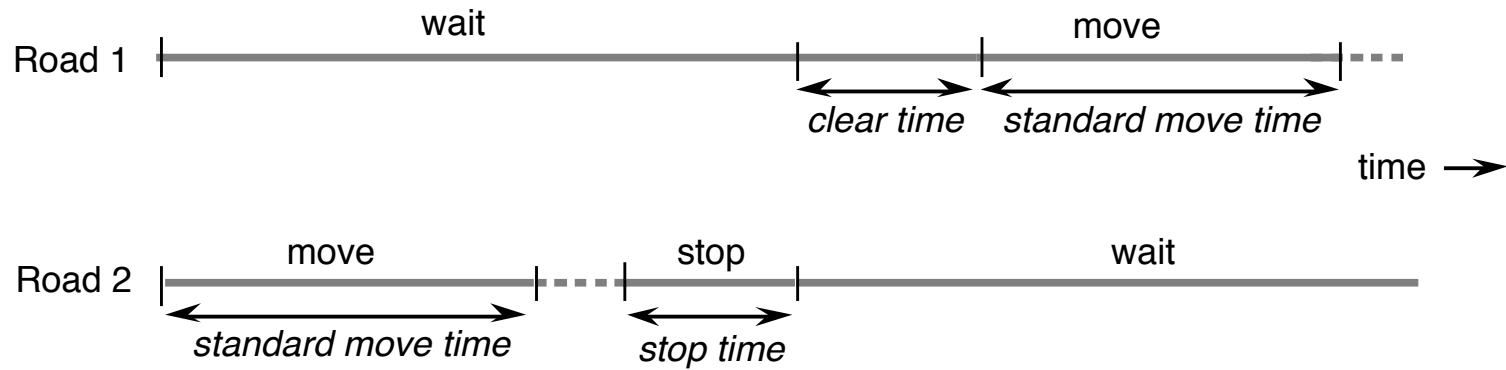


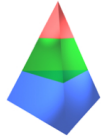
# Example: traffic control system (TCS)





# Functional model of the TCS





## Facts and states

At every point in time, the world of a system is in a particular state. A *state* is defined as a set of *facts*.

The facts contained in a state are elements of the *state base* of the system, being the set of all facts that may belong to a state of the system.

A fact is said to be *current* at the point in time  $t$  if it has been made existent before or at  $t$ , and if it has not been made nonexistent since then.

*Examples of facts:*

phase(1) = wait, phase(2) = move, move\_time(1) = 200,  
move\_time(2) = 240, clear\_time(1) = 8, clear\_time(2) = 8,  
stop\_time(1) = 5, stop\_time(2) = 7





# Acts and agenda

Systems activate each other by generating acts for each other, to be performed at some time. The set of possible acts that a system can deal with is called its *action base*.

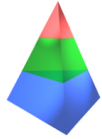
An *agendum* is a pair  $\langle a, t \rangle$  where  $a$  is an act and  $t$  is a point in time. At every moment a system disposes of a set of *agenda*.

The action  $a$  in the agendum  $\langle a, t \rangle$  is said to be *current* at  $t$ .

*Examples of acts:*

let\_pass(1), let\_pass(2)





# The smartie model of a discrete event system

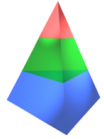
A smartie is defined by a tuple  $\langle \mathbf{S}, \mathbf{M}, \mathbf{A}, \mathbf{R}, \mathbf{T} \rangle$ , where:

- S** : a set of fact types, called the *state base*
- M** : a set of fact types, called the *mutation base*
- A** : a set of act types, called the *action base*
- R** : a set of act types, called the *reaction base*
- T** : a partial function, called the *transition base* :  
$$\mathbf{T} \in \wp \mathbf{A} * \wp \mathbf{S} \rightarrow \wp (\mathbf{R} * \mathbb{D}) * \wp \mathbf{M}$$

In this definition, the union of the extensions of a set of concept types  $C$  (act types or fact types) is denoted as  $\underline{C}$ , and the power set of a set  $X$  is denoted as  $\wp X$ .

Points in time are represented by elements of the set  $\mathbb{T}$ ; the current point in time is denoted by *Now*; (positive) time durations are elements of the set  $\mathbb{D}$ .





# Transition rules

The extension of **T** is a set of *transition rules*  $\langle A, S, R, M \rangle$  where:

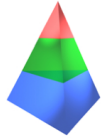
A is the current *action*;  $A \subseteq \underline{\mathbf{A}}$

S is the current *state*;  $S \subseteq \underline{\mathbf{S}}$

R is the current *reaction*; it is a set of pairs  $\langle r, d \rangle$  with  $r \in \underline{\mathbf{R}}$  and  $d \in \underline{\mathbf{D}}$ ; d is the *delay* of the reaction; the action r will become current at time  $\text{Now}+d$

M is the current *mutation*;  $M \subseteq \underline{\mathbf{M}}$





# Activating and conditioning

Smartie  $i$  is *activating* smartie  $j$  if  $\mathbf{R}_i \cap \mathbf{A}_j \neq \emptyset$ .

The new agenda of smartie  $j$  is the symmetric set difference of its current agenda and the current reaction.

Smartie  $i$  is *conditioning* smartie  $j$  if  $\mathbf{M}_i \cap \mathbf{S}_j \neq \emptyset$ .

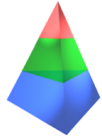
The new state of smartie  $j$  is the symmetric set difference of its current state and the current mutation.

The symmetric set difference  $\Delta$  is defined as follows:

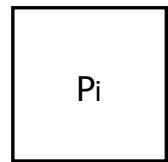
$$A \Delta B = (A \setminus B) \cup (B \setminus A).$$

Its effect is that every element in  $B$  that is not in  $A$  will be 'added', and that every element in  $B$  that is also element in  $A$ , will be 'removed'.

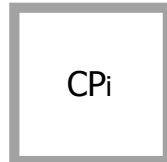




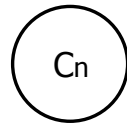
# Legend of the smartienet (1)



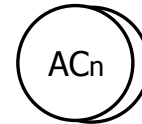
elementary processor  $P_i$



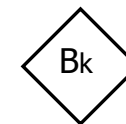
composite processor  $CP_i$



elementary channel  $C_n$



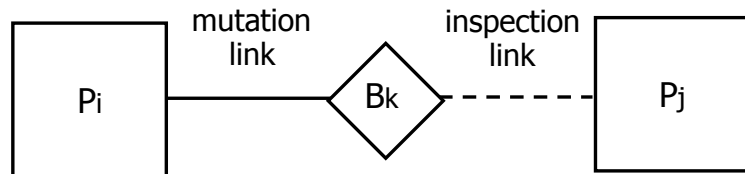
aggregate channel  $AC_n$



elementary bank  $B_k$

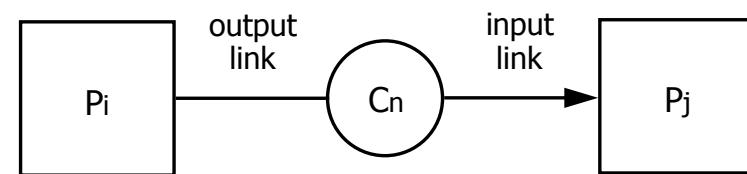


aggregate bank  $AB_k$



processor  $P_i$  *conditions* processor  $P_j$  through bank  $B_k$

$B_k$  is a mutation bank of  $P_i$   
 $B_k$  is an inspection bank of  $P_j$



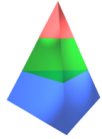
processor  $P_i$  *activates* processor  $P_j$  through channel  $C_n$

$C_n$  is an output channel of  $P_i$   
 $C_n$  is an input channel of  $P_j$

$$M_i \cap S_j \neq \emptyset$$

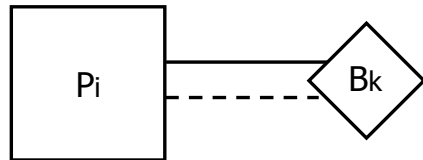
$$R_i \cap A_j \neq \emptyset$$



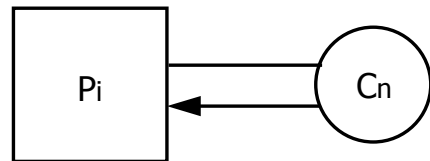


# Legend of the smartienet (2)

$$\mathbf{M}_i \cap \mathbf{S}_i \neq \emptyset$$



processor  $P_i$  *conditions* itself through bank  $B_k$

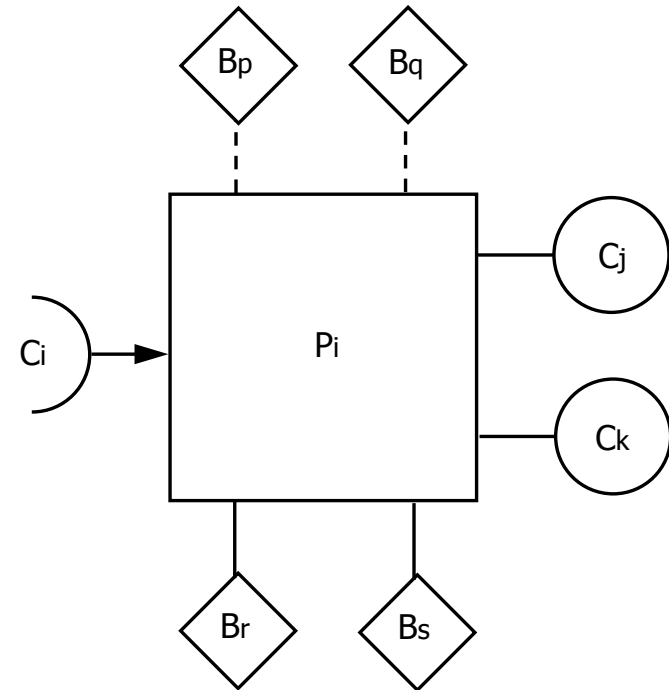


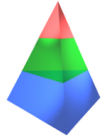
processor  $P_i$  *activates* itself through channel  $C_n$

$$\mathbf{R}_i \cap \mathbf{A}_i \neq \emptyset$$

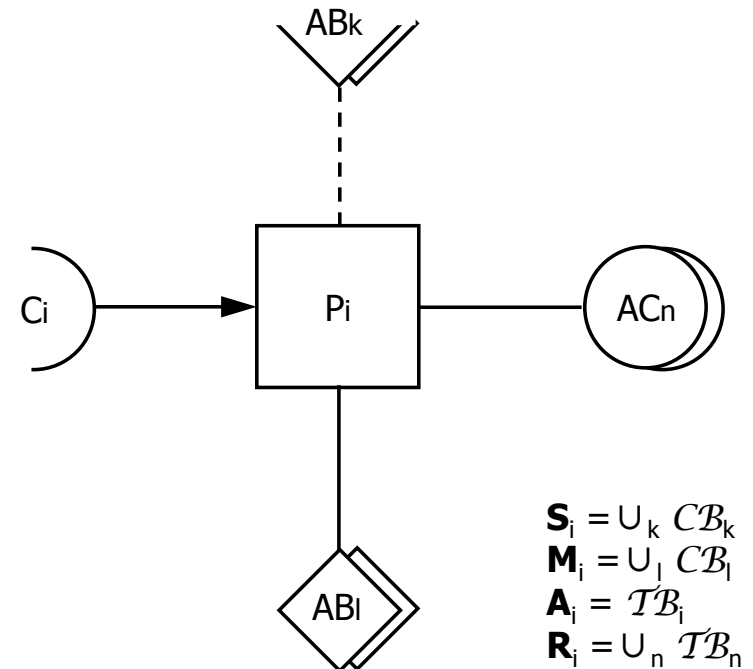
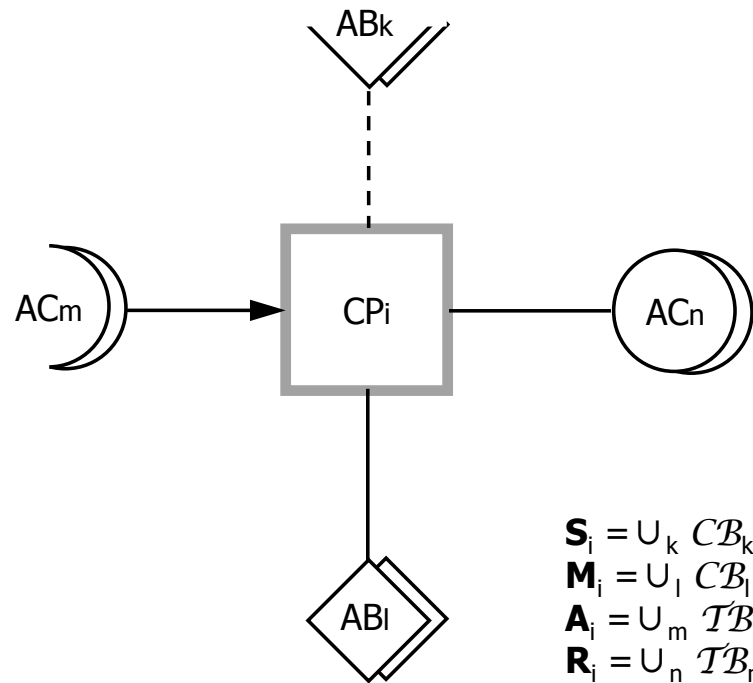
processor  $P_i$  *module*

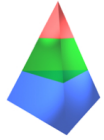
$C_i$  is input channel of  $P_i$   
 $C_j$  is output channel of  $P_i$   
 $C_k$  is output channel of  $P_i$   
 $B_p$  is inspection bank of  $P_i$   
 $B_q$  is inspection bank of  $P_i$   
 $B_r$  is mutation bank of  $P_i$   
 $B_s$  is mutation bank of  $P_i$



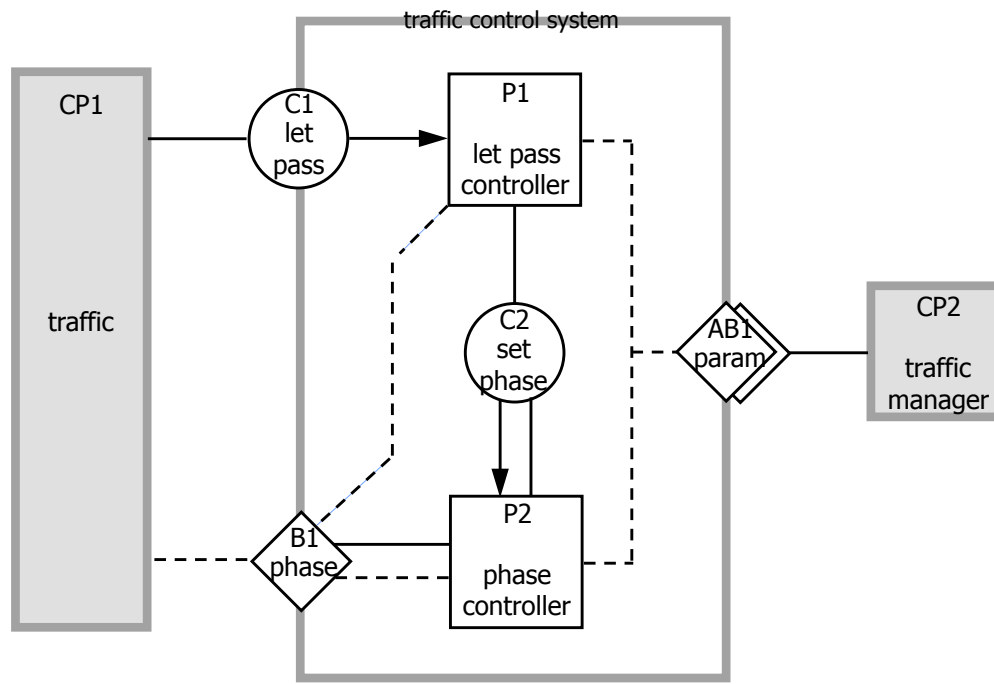


# Legend of the smartienet (3)

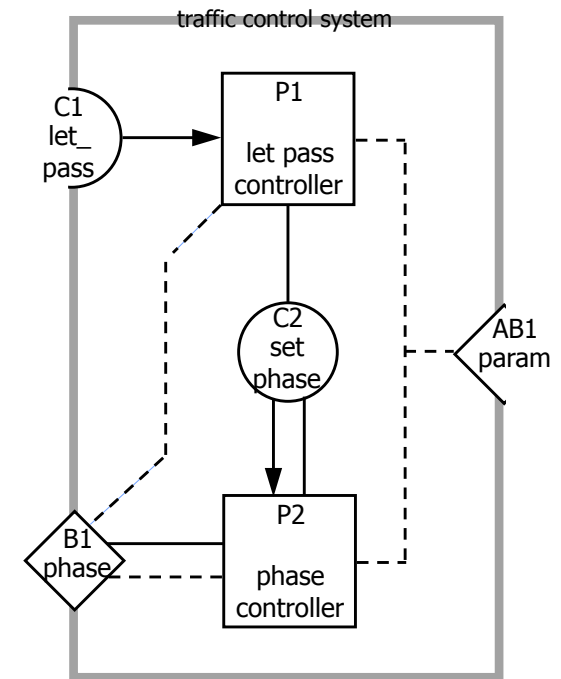




# Detailed smartienet diagram of the TCS



detailed system construction



detailed module construction





## Specification of smartie 1

The first smartie (with kernel P1) is specified as follows:

**S1** = {phase(Cycle), move\_time(Cycle)}

**M1** =  $\emptyset$

**A1** = {let\_pass(Cycle)}

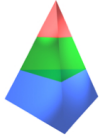
**R1** = {set\_phase(Cycle, Phase)}

The transition base **T1** is specified as follows:

**when** let\_pass(cycle) **occurs**

**if** phase(cycle) = wait **and** phase(other\_cycle) = move  
**then** set\_phase(other\_cycle, stop)  
**with** delay = max(0, (move\_time(other\_cycle) -  
(Now - creation\_time(phase(other\_cycle) = move)))





## Specification of smartie 2

The second smartie (with kernel P2) is specified as follows:

**S2** = {phase(Cycle), stop\_time(Cycle), clear\_time(Cycle)}

**M2** = {phase(Cycle)}

**A2** = {set\_phase(Cycle, Phase)}

**R2** = {set\_phase(Cycle, Phase)}

The transition base **T2** is specified as follows:

**when** set\_phase(cycle,stop) **occurs**

**if** phase(cycle) = move

**then** set\_phase(cycle,wait) **with** delay = stop\_time(cycle);

phase(cycle) := stop

**when** set\_phase(cycle,wait) **occurs**

**if** phase(cycle) = stop

**then** set\_phase(other\_cycle,move) **with** delay = clear\_time(other\_cycle);

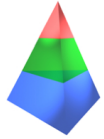
phase(cycle) := wait

**when** set\_phase(cycle,move) **occurs**

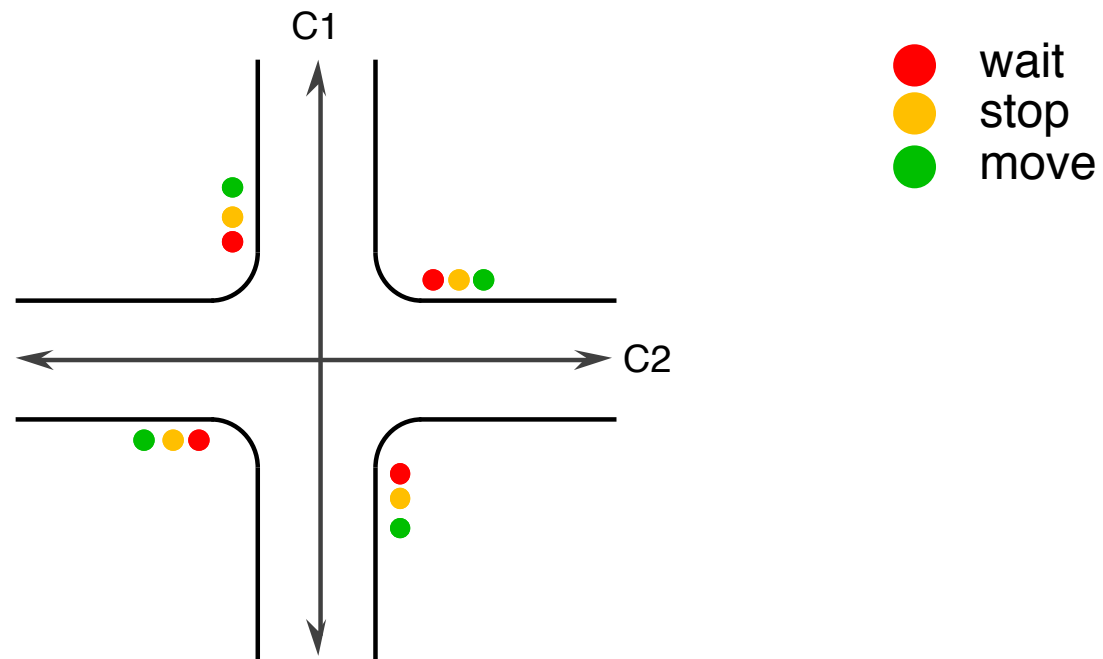
**if** phase(cycle) = wait **and** phase(other\_cycle) = wait

**then** phase(cycle) := move



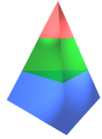


# A possible implementation of the TCS

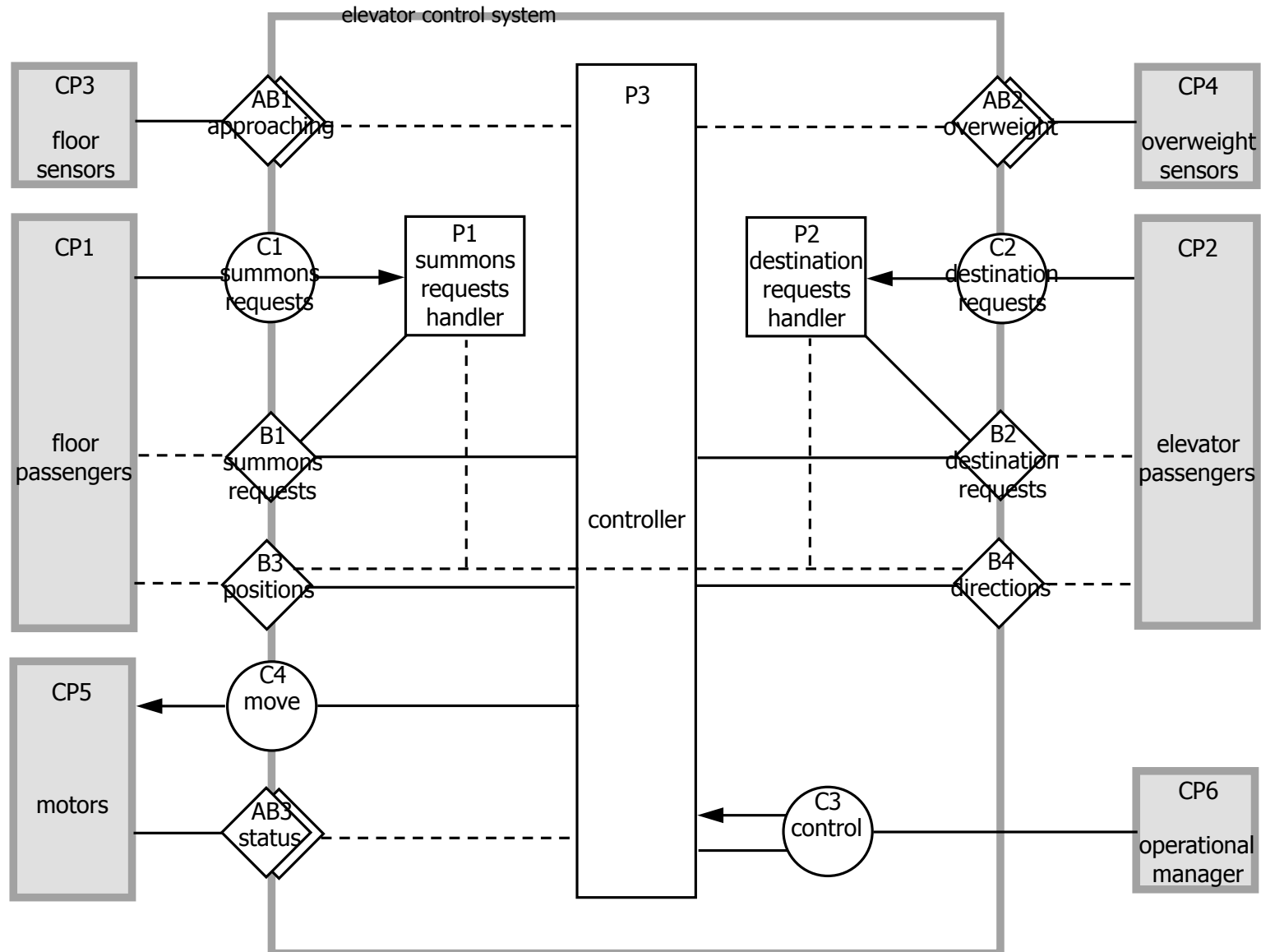


There are sensors in the road to generate let\_pass commands





# The elevator control system (ECS)





# The elevator control system (ECS)

direction(Elevator) = up:

$\exists$ Floor: (destination\_requested(Elevator,Floor) = true **or** summons\_requested(Floor,up) = true)  
**and** Floor > position(Elevator)

direction(Elevator) = down:

$\exists$ Floor: (destination\_requested(Elevator,Floor) = true **or** summons\_requested(Floor,down) = true)  
**and** Floor < position(Elevator)

direction(Elevator) = still:

direction(Elevator)  $\neq$  up **and** direction(Elevator)  $\neq$  down

## Specification of smartie 1

The internal smartie with kernel P1 (summons requests handler) is specified as follows:

**S1** = {summons\_requested(Floor,Direction), position(Elevator), direction(Elevator)}

**M1** = {summons\_requested(Floor,Direction)}

**A1** = {summons\_request(Floor,Direction)}

**R1** =  $\emptyset$

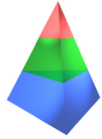
The transition base **T1** is specified as follows:

**when** summons\_request(Floor, Direction) **occurs**

**if** summons\_requested(Floor, Direction) = false **and**  
(**there is no** Elevator **for which** position(Elevator) = Floor **and**  
(direction(Elevator) = Direction **or** direction(Elevator) = still))

**then** summons\_requested(Floor, Direction) := true





# Conclusions $\delta$ -theory and $\pi$ -theory

## *Ontological model*

Interaction – activating

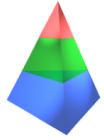
Interstriction – conditioning

Implementation

## *Essential model*

Realisation





# Outline

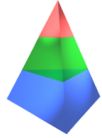
$\delta$ -theory and  $\pi$ -theory

**$\psi$ -theory**

$\tau$ -theory

$\beta$ -theory





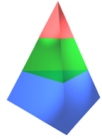
# Ontological theories: the $\psi$ -theory

The  $\psi$ -theory ( $\psi$  is pronounced as PSI, standing for Performance in Social Interaction) is a theory about the ontological essence of social systems.

It clarifies and explains the construction and operation of organisations.

The  $\psi$ -theory is rooted in the  $\pi$ -theory, speech act theory [Austin, Searle], social action theory [Habermas], and information systems theory [Langefors].





# The $\psi$ -theory

The  $\psi$ -theory (PSI stands for Performance in Social Interaction) consists of two parts: the general  $\psi$ -theory and the special  $\psi$ -theory.

The **general  $\psi$ -theory** is a theory of human cooperation. Therefore, it is also called the **human face** or **front side** of the  $\psi$ -theory.

The **special  $\psi$ -theory** clarifies the consequences of the general  $\psi$ -theory for the systems approach to organisations. Therefore, it is also called the **system face** or **back side** of the  $\psi$ -theory.

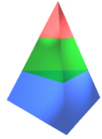




## The general $\psi$ -theory

- The operating principle of organisations is that *subjects* (human beings) enter into and comply with *commitments* regarding the production of *products*.
- Commitments are raised and dealt with in *transactions*. These are interaction structures of *coordination acts/facts* between two actors, concerning a *production act/fact*. One subject is the *initiator* of the transaction and the other is the *executor*.
- The effect of a coordination act is the creation of a coordination fact, which is an *event* (state change) in the *coordination world* of the organisation.
- The effect of a production act is the creation of a production fact, which is an *event* (state change) in the *production world* of the organisation.





# The transaction process

In the **proposition phase**, the actors discuss the *product to be produced*, and try to come to agreement

In the **execution phase**, the executor *produces some product*

In the **result phase**, the actors discuss the *product that has been produced*, and try to come to agreement

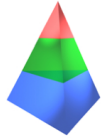


Asking for flowers  
Ordering a book  
Applying for membership

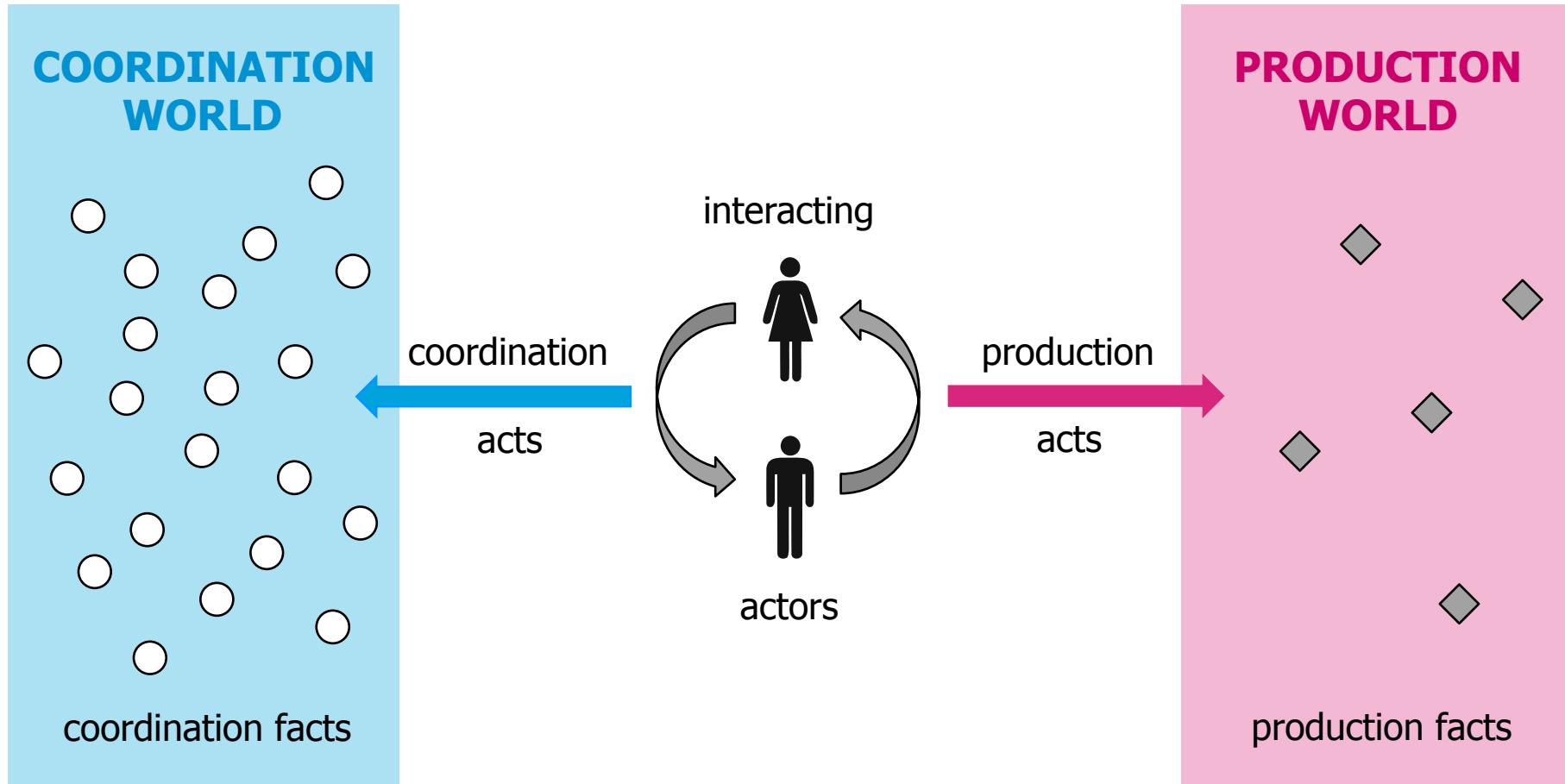
Creating  
Deciding  
Judging

Having got the flowers  
Having got the book  
Having become member



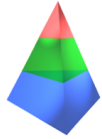


# Coordination and production acts/facts

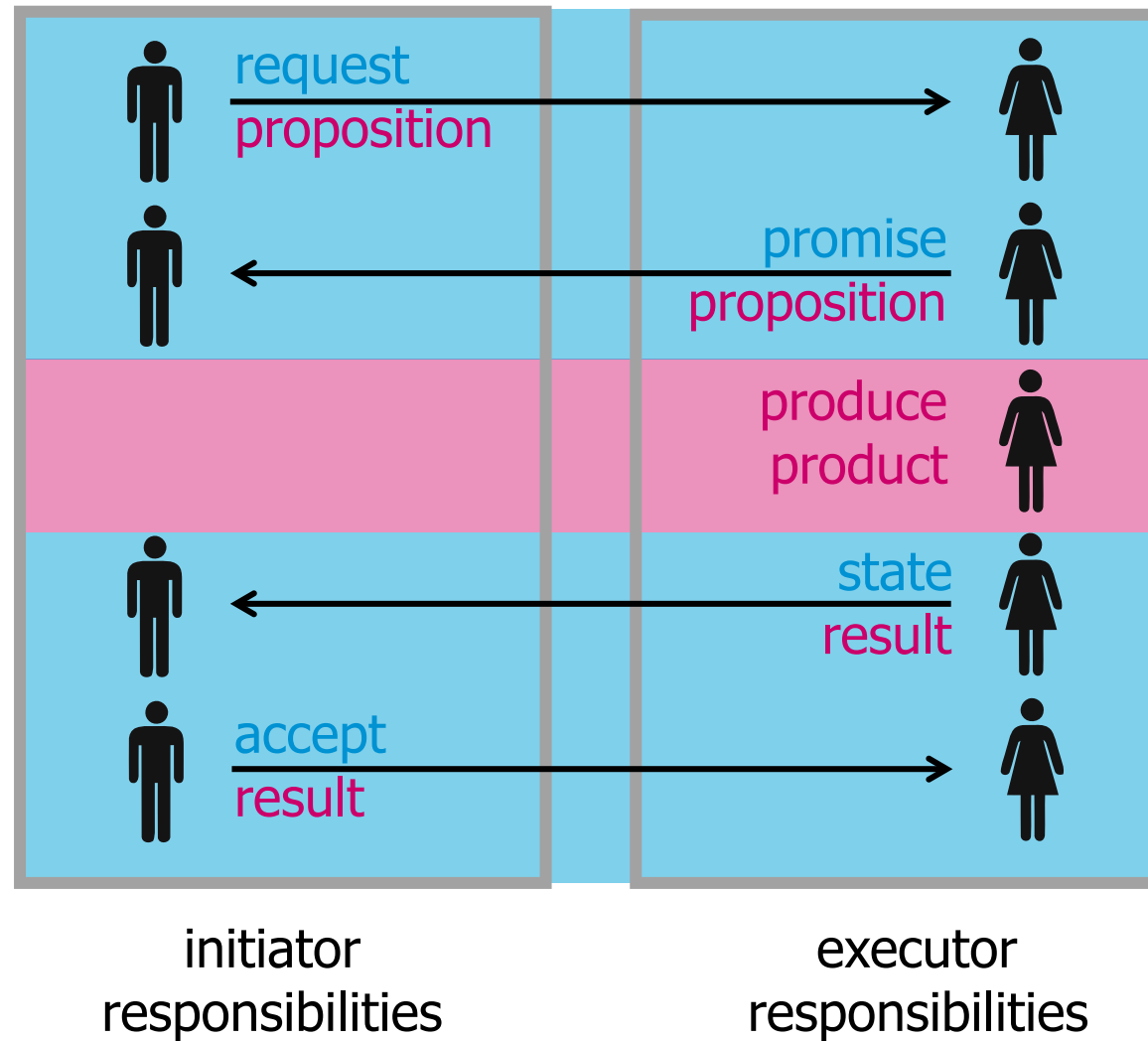


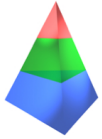
These state changes occur according to the universal transaction pattern.  
Did you see the pattern?



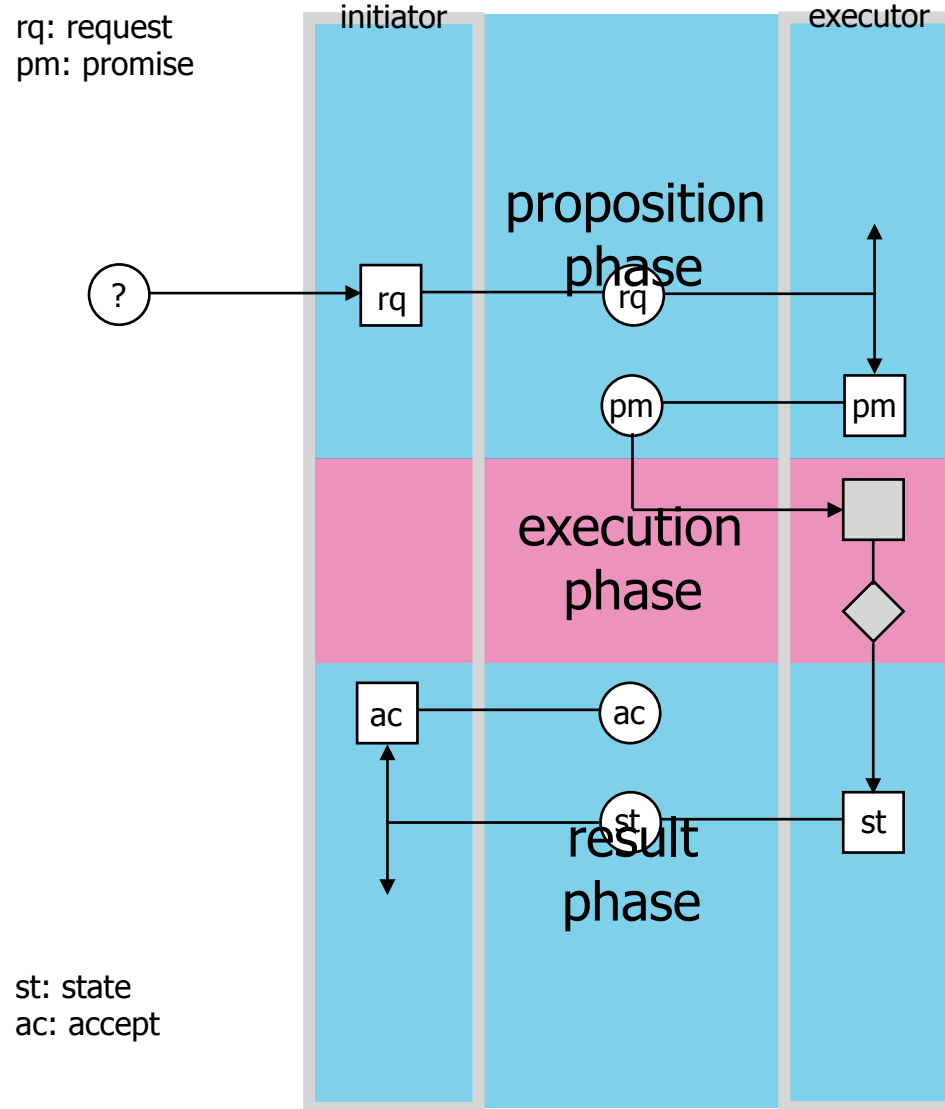


# The basic transaction pattern



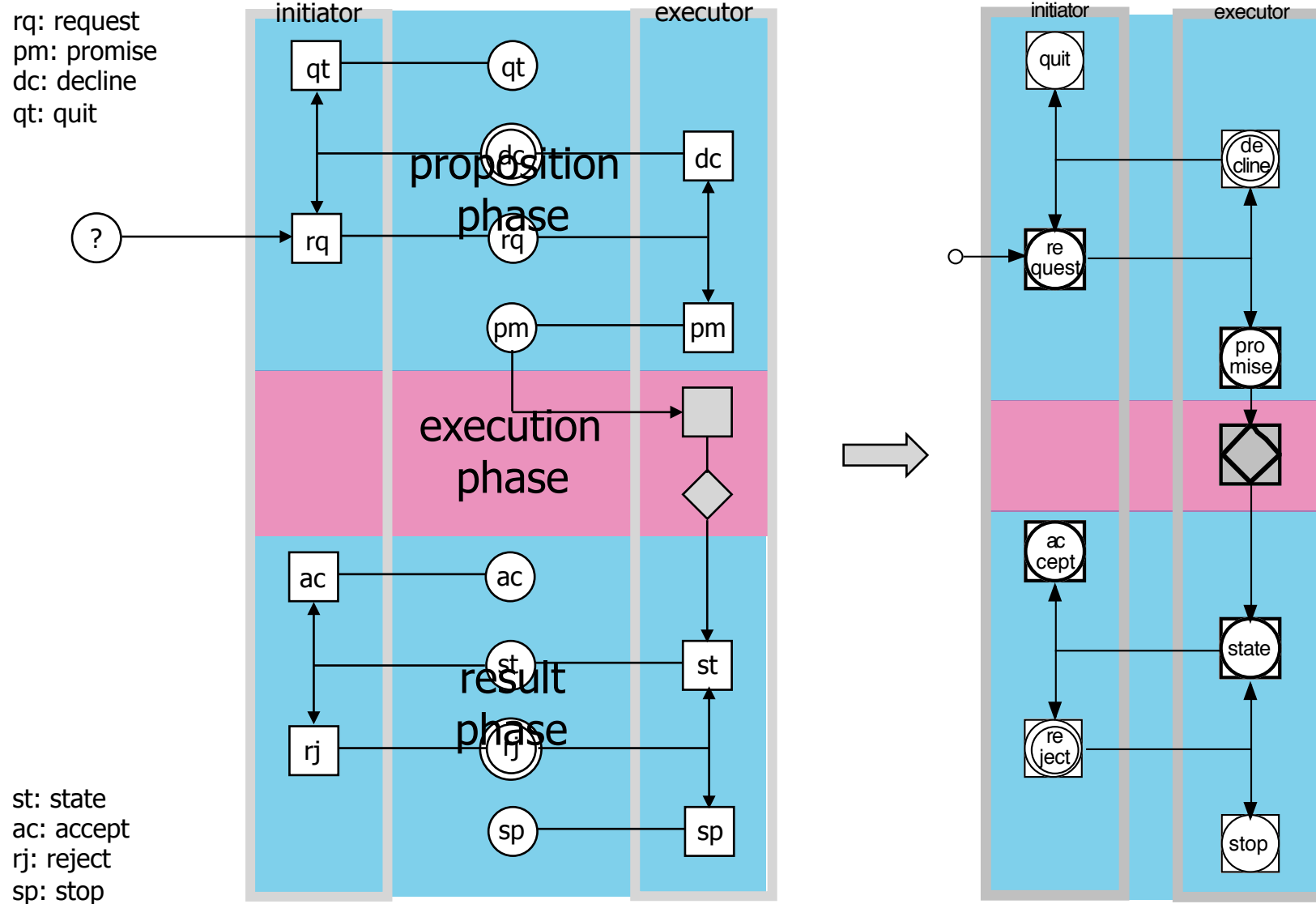


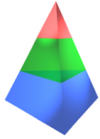
# The basic transaction pattern



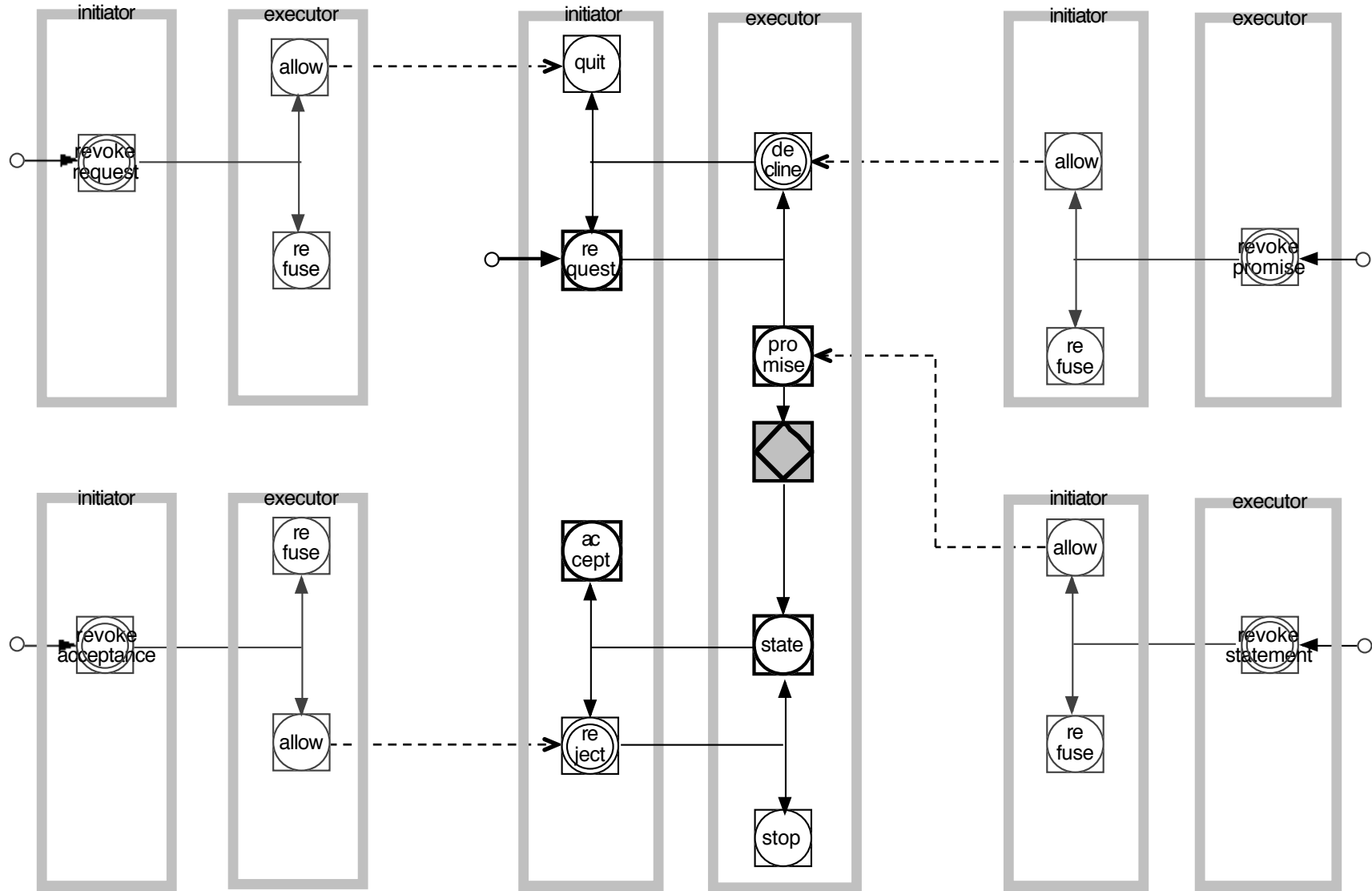


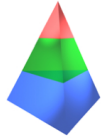
# The standard transaction pattern





# The complete transaction pattern





# Validity claims in coordination acts

According to Jürgen Habermas' Theory of Communicative Action, the performer of a coordination act raises three *validity claims* towards the addressee. The addressee has to accept all of them in order to let the coordination act be successful.

## Claim to justice (G: Richtigkeit, NL: juistheid)

Has the performer the authority to perform the coordination act?

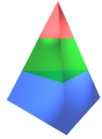
## Claim to sincerity (G: Wahrhaftigkeit, NL: oprechtheid)

Is the performer sincere in performing the coordination act?

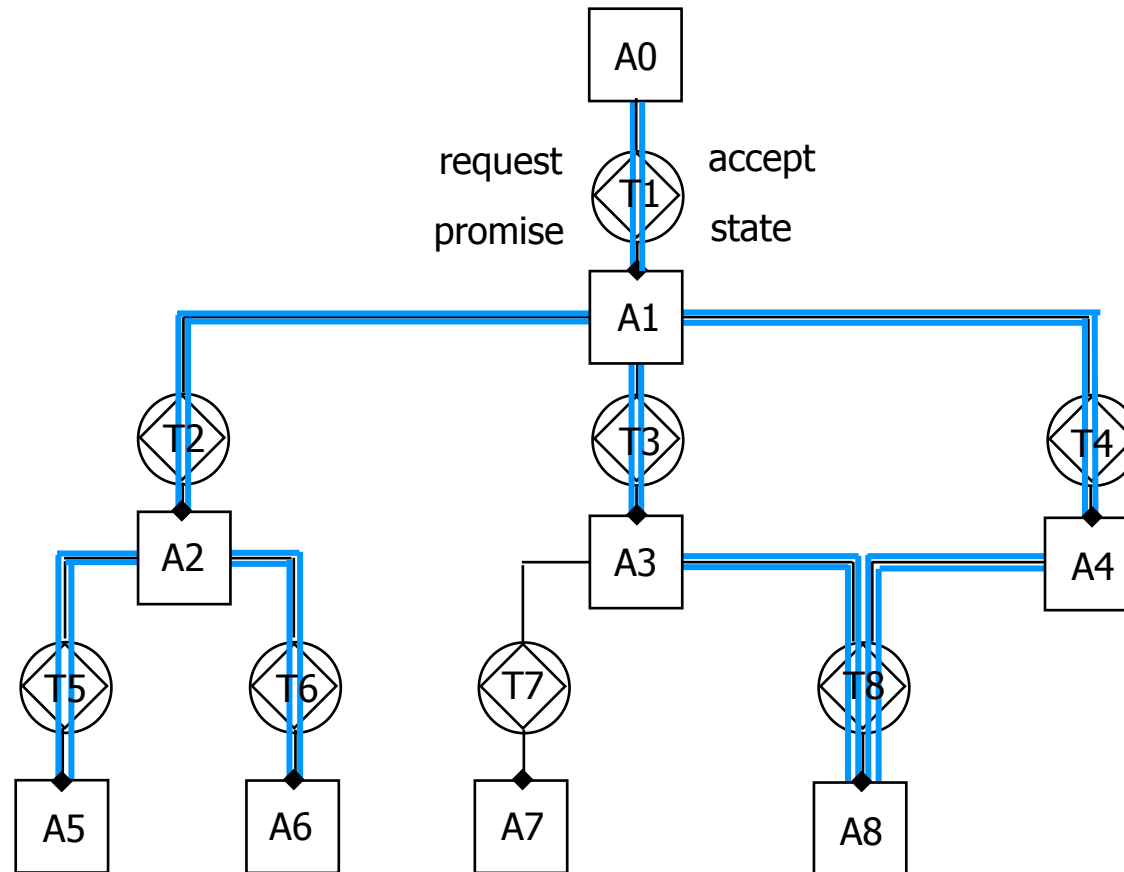
## Claim to truth (G: Wahrheit, NL: waarheid)

Does the product exist or can it be produced?





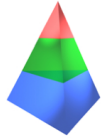
# Business process



**In order to produce P1, A1 needs a P2, a P3 and a P4! And ...**

NOTE: component transactions may also be carried out in parallel.





# The special $\psi$ -theory

The special  $\psi$ -theory takes the systems approach to organisations. Being the back side, PSI is read backwards (ISP), with two meanings:

## Intelligent System Partitioning

The three human abilities (performa, informa, and forma) can also be applied to production. This leads to *partitioning* an organisation in three aspect organisations: *B-organisation* (B from Business), *I-organisation* (I from Information) and *D-organisation* (D from Document and Data).

## Integrated System Perspectives

The ontological model of an organisation is the integration of four sub models or *perspectives* on the whole: *Construction Model* (CM), *Process Model* (PM), *Fact Model* (FM), and *Action Model* (AM).





# Intelligent System Partitioning

*creating  
deciding  
judging*

**B-organisation**

*remembering  
recalling  
computing*

**I-organisation**

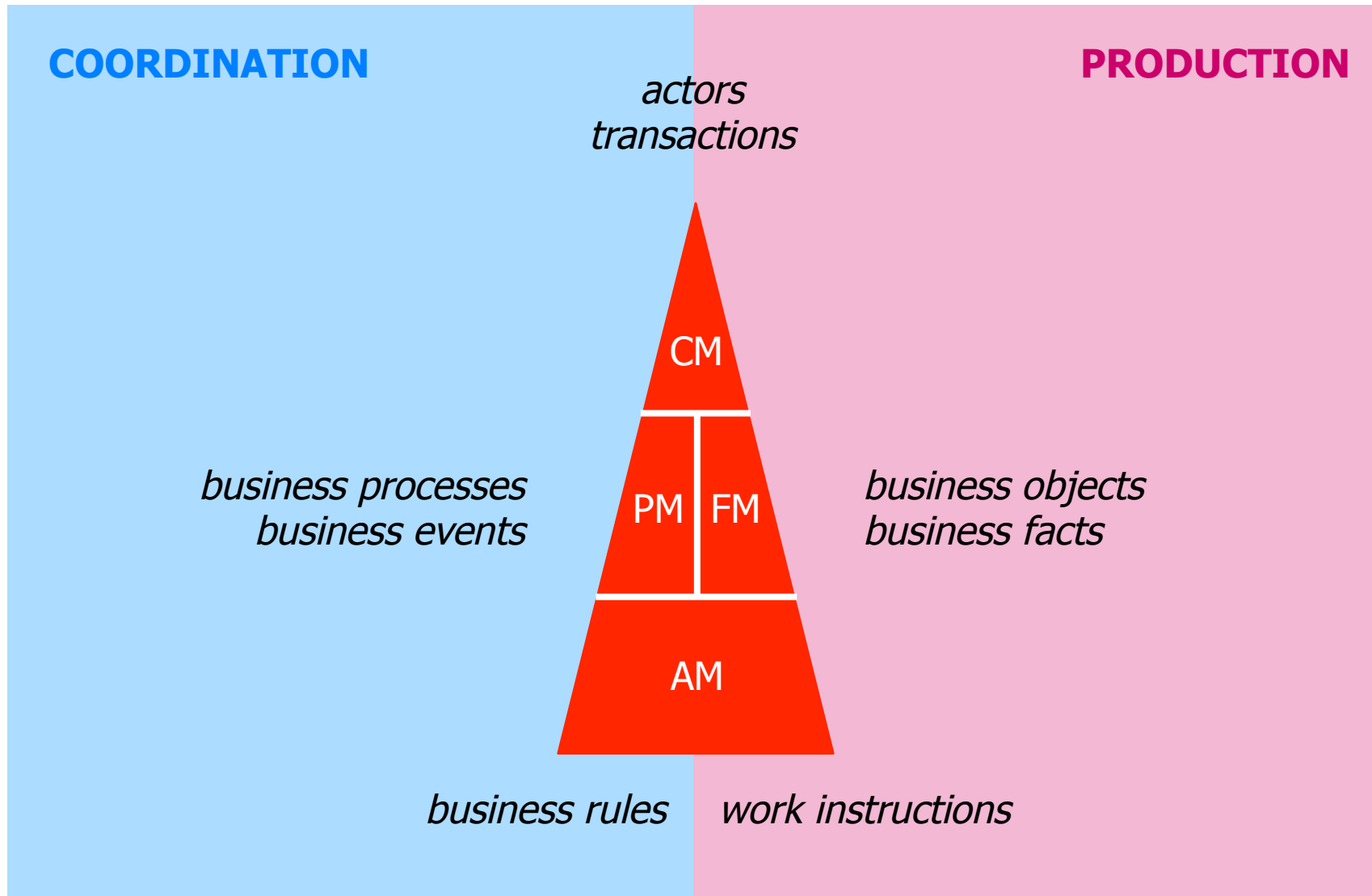
*storing  
retrieving  
transmitting  
copying*

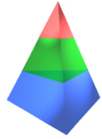
**D-organisation**





# Integrated System Perspectives





# Genotype and phenotype of humans

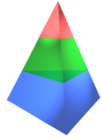
In biology, a distinction is made between the *genotype* and the *phenotype* of organisms



The phenotypes of identical twins may differ considerably (notably in the course of time)

Conversely, people with different genotypes may have quite similar phenotypes.





# Genotype and phenotype of organisations

Also regarding organisations, a distinction can be made between genotype and phenotype

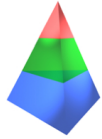
The **genotype** of an organisation is defined as its **essential model**

The **phenotype** of an organisation is defined as the **realisation** and **implementation** of the essential model

**Realisation** is devising the I-organisation and the D-organisation of the essential model

**Implementation** is allocating technological means to actor roles, and to coordination and production acts/facts





# The phenotype of a bank (1)



How can I help you, sir?

I want to withdraw money

*request*

From your current account?

Yes

How much do you want?

400 euro please

*employee fills out a form*

If you sign here please

*client signs the form*

One moment please

*promise*

*employee issues banknotes*

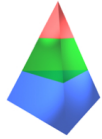
Here you are, sir

*state*

Thank you

*accept*





## The phenotype of a bank (2)



Welcome to the ING bank

Please insert your card

*client inserts card*

Enter your PIN please

*client keys the PIN*

Choose the amount please

*client presses € 400*

Take your card please

*client takes the card*

Your money is being counted

*banknotes are produced*

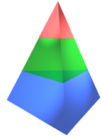
Take your money please

*client takes the banknotes*

*request  
promise*

*state  
accept*





# The crispie model of an organisation

Hereafter, a formal definition of a crispie is presented, fully based on the formal definition of a smartie [JDM-7]. In this definition, the union of the extensions of a set of concept types (act types or fact types)  $C$  is denoted as  $\underline{C}$ , and the power set of a set  $X$  is denoted as  $\wp X$ . Points in time are represented by elements of the set  $\mathbb{T}$ ; the current point in time is denoted by Now; (positive) time durations are elements of the set  $\mathbb{D}$ .

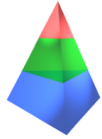
A crispie is formally defined as a tuple  $\langle C, R, I, S, P \rangle$ , where

- C** : a set of C-fact types, called the *coordination base*
- R** : a set of action rules, called the *rule base*
- I** : a set of intentions, called the *intention base*
- S** : a set of C-fact types and P-fact types, called the *state base*
- P** : a set of product kinds, called the *product base*

$$R : \wp \underline{C} * \wp \underline{S} \rightarrow \wp (I * \underline{P} * \mathbb{T} * \mathbb{D})$$

A crispie is called *elementary* if the C-facts in its coordination base all regard one and the same product kind. Crispies that are not elementary, are called *composite*.





# The crispie model of an organisation

The components **C**, **S**, and **P** have been explained above. The intention base **I** comprises all intentions that are contained in the complete transaction pattern (Cf. Figure 2.3). The rule base **R** can conveniently be represented by its extension, i.e. the set of *action rules* ( or business rules) of the form  $\langle C, S, \langle i, p, pt, sd \rangle \rangle$  where:

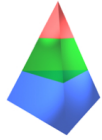
$C$  is the *event* that is going to be responded to;  $C \subset \underline{\mathbf{C}}$ . Note that  $\underline{\mathbf{C}} \subset \mathbf{I} * \underline{\mathbf{P}}$

$S$  is a set of C-facts and P-facts, called the *state*;  $S \subset \underline{\mathbf{S}}$ .

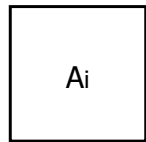
$\langle i, p, pt, sd \rangle$  is the *response*; it is a set of tuples  $\langle i, p, pt, sd \rangle$  where  $i$  is the intention of the created C-fact ( $i \in \mathbf{I}$ ) and  $p$  is the product that the C-fact is concerned about ( $p \in \underline{\mathbf{P}}$ ). The production time of  $p$  is  $pt$  ( $pt \in \mathbb{T}$ ), and the settlement time of the C-fact  $st = \text{Now} + sd$  ( $sd \in \mathbb{D}$ ).

Then Mary gets this event to respond to:  $\langle \text{request}, \text{membership \#387 is started}, 20130401, st-r \rangle$ , created by John. The time  $st-r$  is the settlement time of the request. Let us assume that the current state  $S$  allows Mary to promise. Her response consists of the tuple  $\langle \text{promise}, \text{membership \#387 is started}, 20130401, st-p \rangle$ , in which  $st-p$  is the settlement time of the promise. This is an event to which Mary has to respond. Let us assume that the current state  $S$  allows Mary to produce the product (the execute act, followed by the state act). Then her response consists of the tuple  $\langle \text{state}, \text{membership \#387 is started}, 20130401, st-s \rangle$ , in which  $st-s$  is the settlement time of the state. This is the event to which John has to respond. Let us assume that the current state  $S$  allows him to accept. Then his response will consist of the tuple  $\langle \text{accept}, \text{membership \#387 is started}, 20130401, \text{null} \rangle$ , in which  $\text{null}$  is the settlement time of the accept. Since the accept fact is a terminal state in the transaction process, its settlement time is irrelevant; that's why it gets the value *null*.





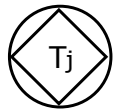
# The crispinet diagram



elementary actor role  $A_i$



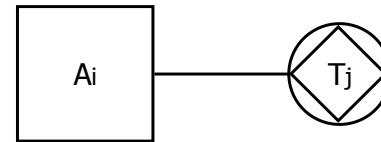
composite actor role  $CA_k$



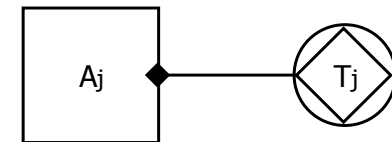
transaction kind  $T_j$



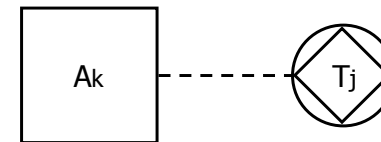
aggregate transaction kind  $AT_i$



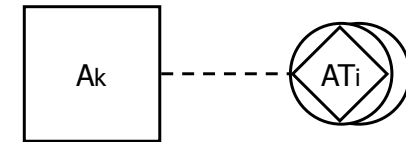
$A_i$  is an initiator role of the transaction process  $T_j$



$A_j$  is the executor role of the transaction process  $T_j$



$A_k$  has access to the transaction bank of  $T_j$



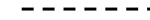
$A_k$  has access to the transaction banks of  $AT_i$



initiator link



executor link

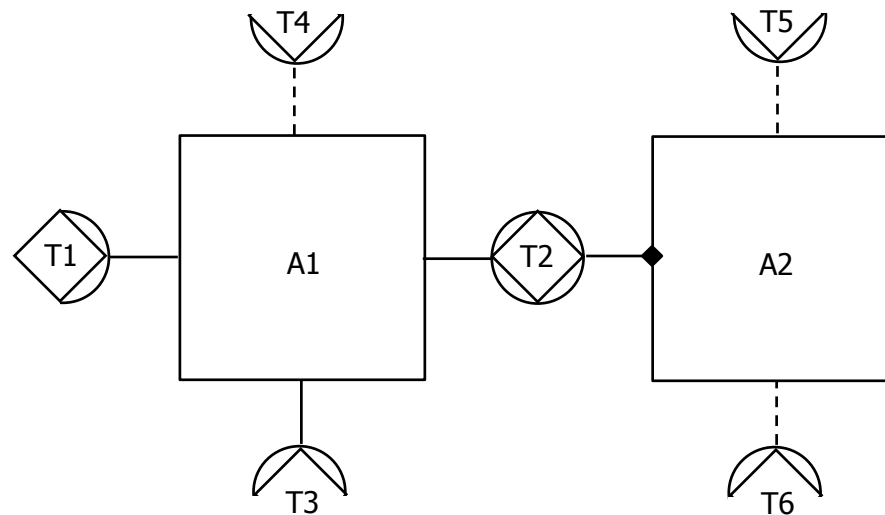
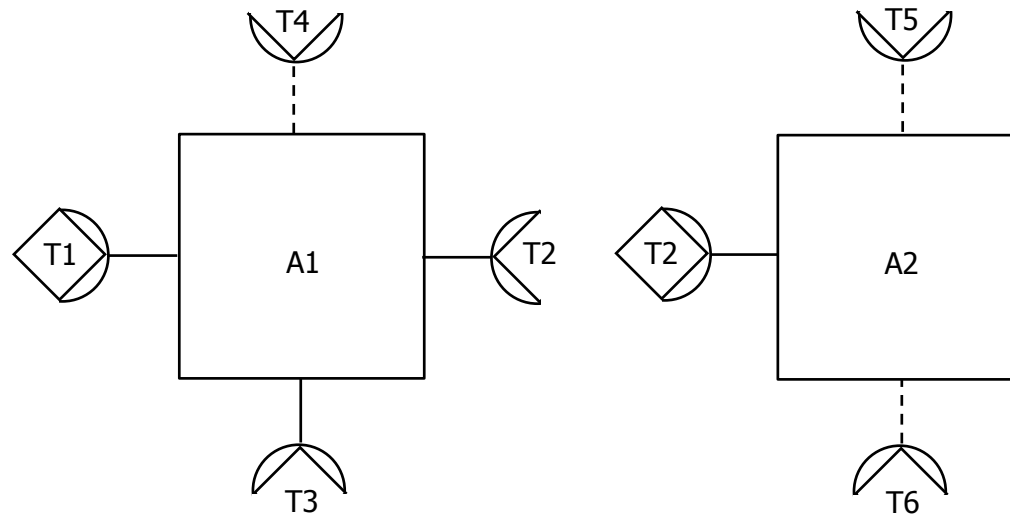


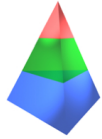
information link



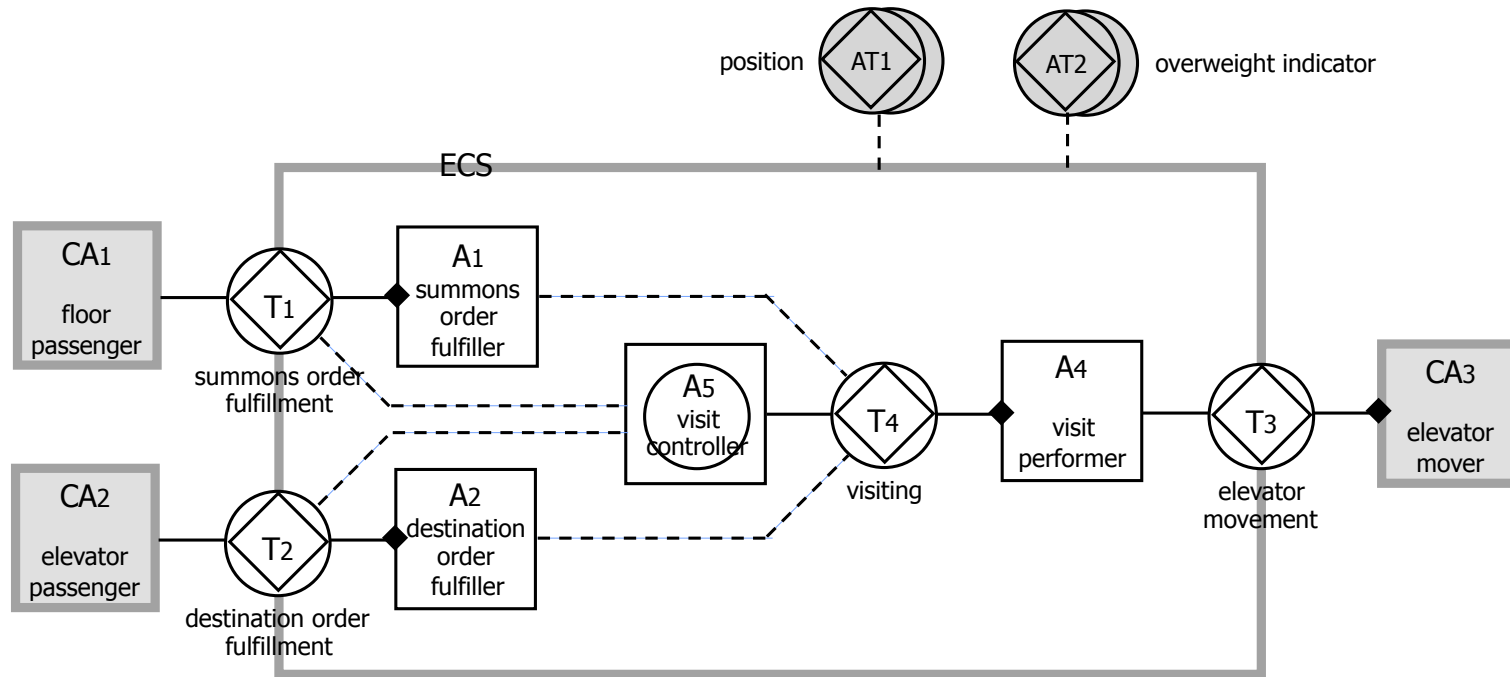


# Constructs in crispienets



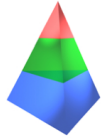


# Crispienet of the ECS



transaction kind	product kind
T1 summons order fulfillment	P1 summons order for Floor in Direction is fulfilled
T2 destination order fulfillment	P2 destination order for Elevator to Floor is fulfilled
T3 elevator movement	P3 Elevator is set to move in Direction
T4 visiting	P4 Elevator visits Floor in Orientation
T5 visit control	P5 visit control for Period is done





# Rule base of crispie A2

**when** destination order fulfillment **for** (Elevator, Floor) is requested  
**if** overweight indicator **of** Elevator **is false**  
**then**

**if** position **of** Elevator **is lower than** Floor  
**then** **if** orientation **of** Elevator **is up or** orientation **of** Elevator **is idle**  
**then** promise destination order fulfillment **for** (Elevator, Floor)  
**else** decline destination order fulfillment **for** (Elevator, Floor)

**else-if** position **of** Elevator **is higher than** Floor  
**then** **if** orientation **of** Elevator **is down or** orientation **of** Elevator **is idle**  
**then** promise destination order fulfillment **for** (Elevator, Floor)  
**else** decline destination order fulfillment **for** (Elevator, Floor)

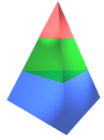
**else-if** position **of** Elevator **is equal to** Floor  
**then** decline destination order fulfillment **for** (Elevator, Floor)

**when** destination order fulfillment **for** (Elevator, Floor) is promised  
request visiting **for** (Elevator, Floor, Orientation)  
**with** Orientation **is** orientation **of** Elevator

**when** destination order fulfillment **for** (Elevator, Floor) is promised  
**while** **there is some** Orientation  
**for which** visiting **for** (Elevator, Floor, Orientation) is accepted  
execute destination order fulfillment **for** (Elevator, Floor)  
state destination order fulfillment **for** (Elevator, Floor)

**when** visiting **for** (Elevator, Floor, Orientation) is stated  
**then** accept visiting **for** (Elevator, Floor, Orientation)





# Outline

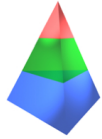
$\delta$ -theory and  $\pi$ -theory

$\psi$ -theory

**$\tau$ -theory**

$\beta$ -theory





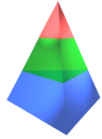
# Ontological theories: the $\tau$ -theory

The  $\tau$ -theory ( $\tau$  is pronounced as TAO, standing for Teleology, Affordances, Ontology) is a theory about subjects (having purposes) and objects (having properties) and the possible relationships between them.

It clarifies and explains such terms as “function”, “construction”, “value”, and “experience”.

The  $\tau$ -theory is rooted in teleology [Kant, Jung, Hegel], affordance theory [Gibbs], ontology [Bunge], and mereology [Simons].





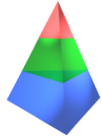
# The $\tau$ -theory

$\tau$  (**TAO**) stands for **Teleology Affordance Ontology**

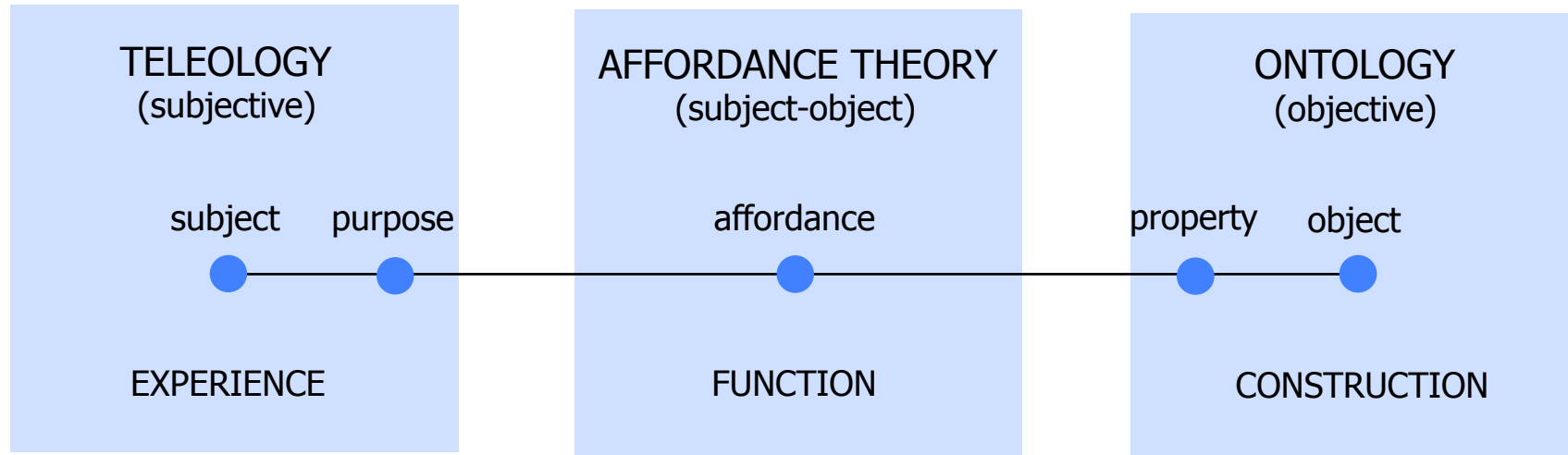
- The word “teleology” is composed of “telos” and “logos”; it is about explaining the behavior of *subjects* from studying their *purpose(s)*.
- The word “ontology” is composed of “ontos” and “logos”; it is about studying the *existence* (nature, essence) of *objects*.
- The word “affordance”<sup>1</sup> refers to the usefulness of an *object* for a *subject* in the light of his/her purpose.
- The  $\tau$ -theory explains the important difference between *function* and *construction*.

1) Gibson, J.J.: *The Ecological Approach to Visual Perception*, Chapter 8. Boston: Houghton Mifflin, 1979





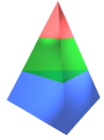
# Affordances



In their pursuit of satisfying needs, subjects do not primarily perceive objects but the *affordances* (potential usages) they may offer.

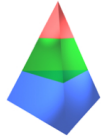
Example: if you (*subject*) want to sit (*purpose*), you may perceive that you can sit (*affordance*) on a tree-stump (*object*), because the height of its surface (*property*) fits your purpose.



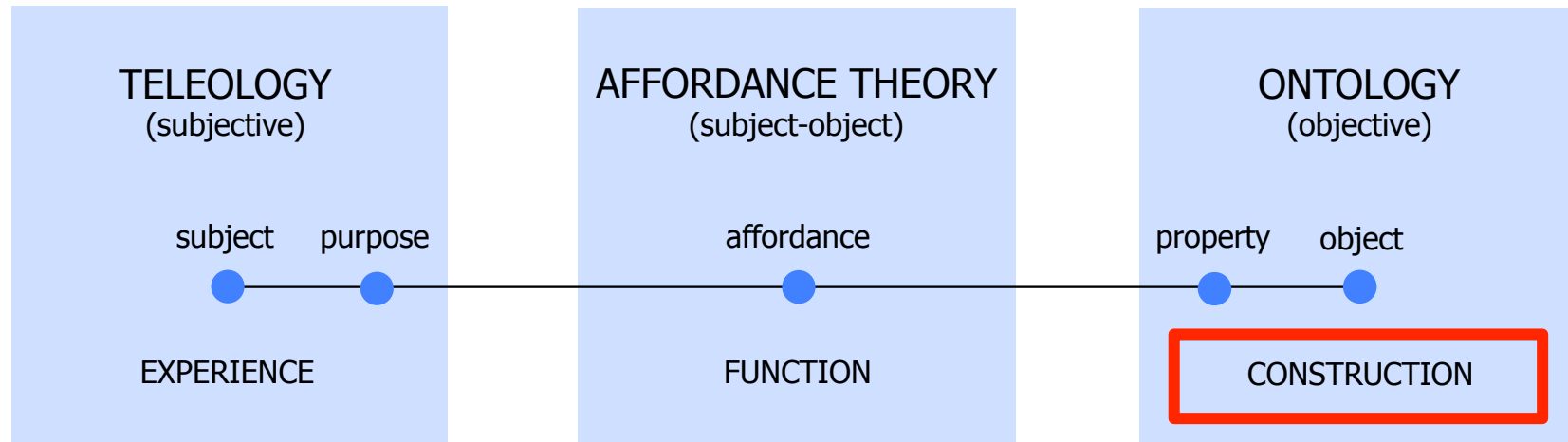


# The affordances of a chair





# Construction (1)



Taking the *construction* perspective, i.e. disregarding the affordances they may offer, one perceives *objects* and their *properties*.

This perception is independent of the observing subject: ontology is *objective*. It is only dependent on the applied *ontological theory* (the 'mental glasses' the subject has put on), like the  $\psi$ -theory.





# Construction (2)

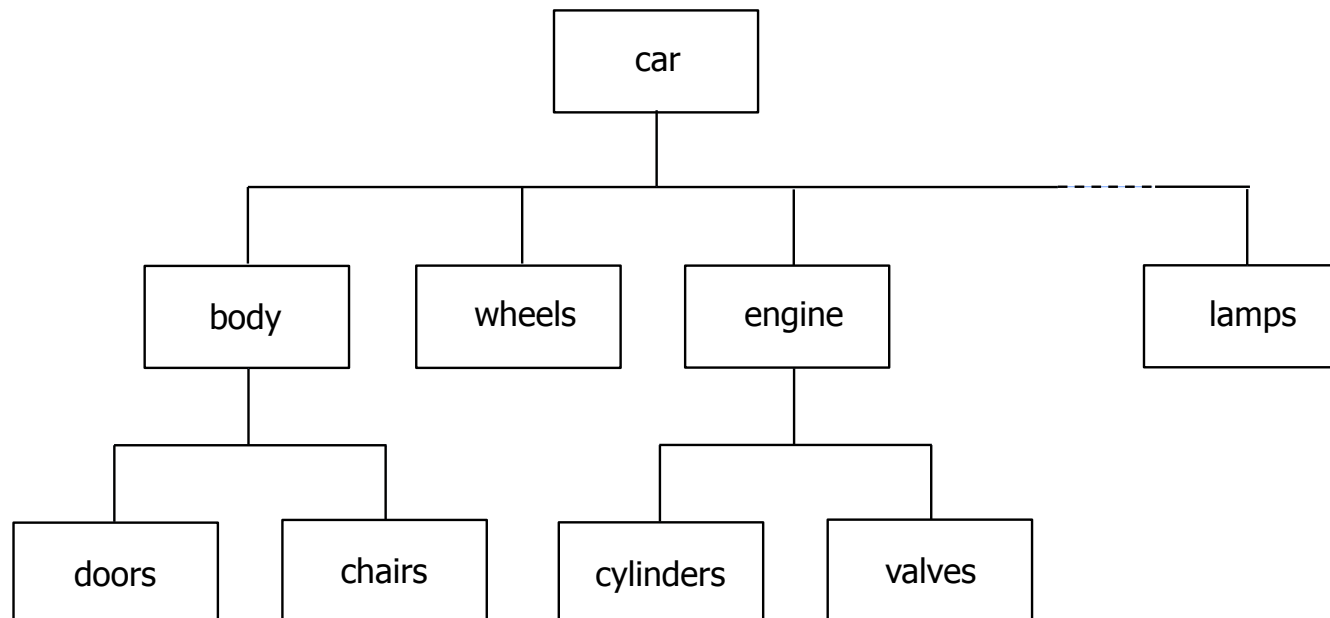


*the mechanic's perspective*

**construction** :  
the components and their  
mutual bonds

**operation** :  
the manifestation of the  
construction in the course of time

***constructional (de)composition***





## Construction (3)

The **construction** of a system **is** something **objective**. In a literal sense, a system is its construction.

Because constructional (conceptual) models of systems show 'openly' their construction, they are called **white-box models**.

*Examples:*

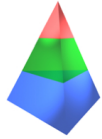
A DEMO model of an organisation

A BPMN model of a work flow

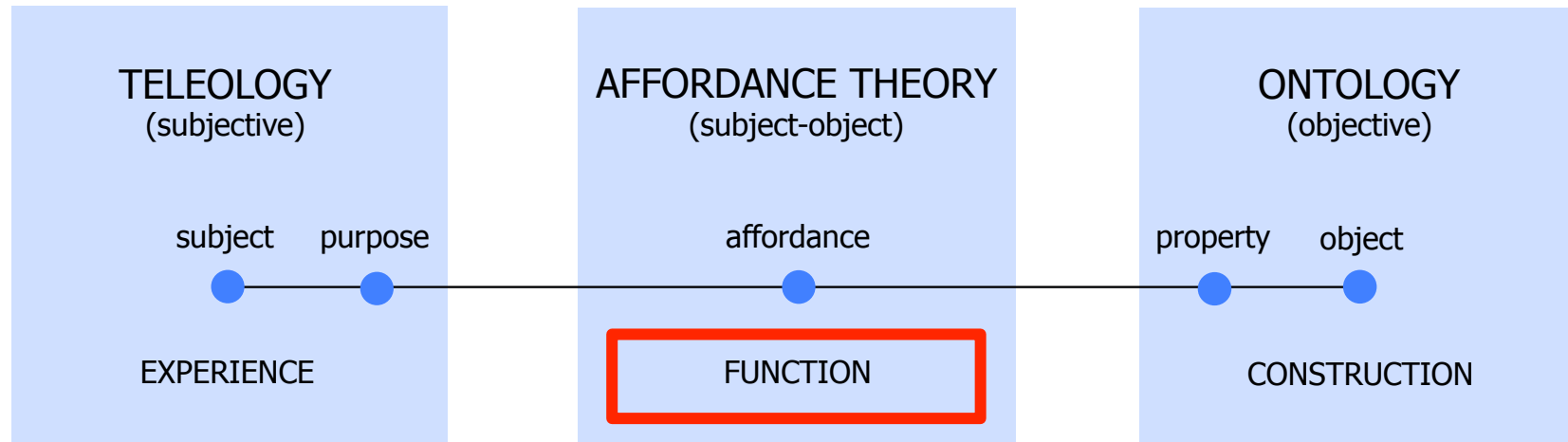
A UML Object Diagram of a software system

**There is only *one* correct constructional model of a system!**





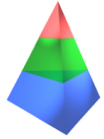
# Function (1)



Next to using natural objects, subjects also create objects (*artefacts*). They are designed and made with some affordance in mind. This affordance is commonly called the *function* of the artefact.

Examples: the function of a chair is to sit on, and the function of a table is to sit at.





## Function (2)

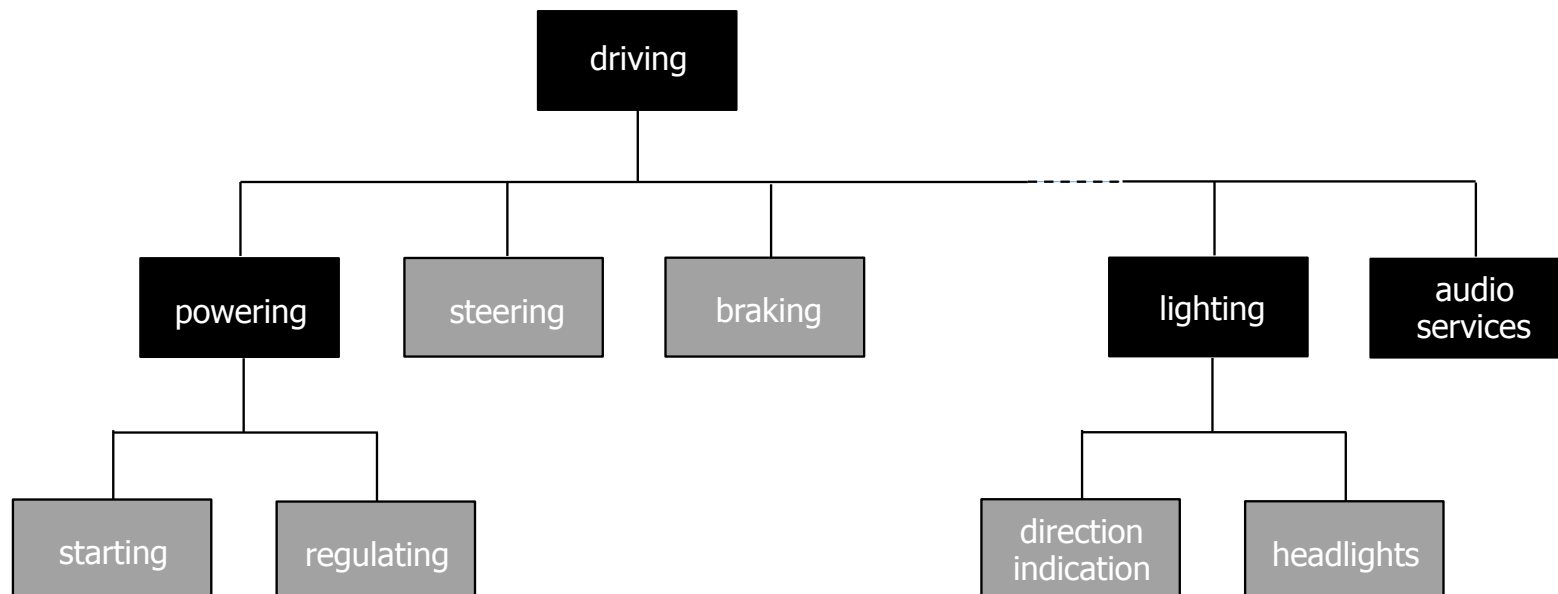


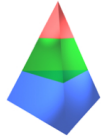
*the driver's perspective*

**function :**  
relationship between the car and  
the driver

**behavior :**  
the manifestation of the  
function in the course of time

***functional (de)composition***





## Function (3)

A **function** of a system is an affordance of the system for a subject. Thus, it is not an objective system property.

Because functional (conceptual) models of systems 'hide' their construction, they are called **black-box models**.

*Examples:*

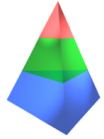
A Business Capability Map of an organisation

An IDEF0/SADT model of a work flow

A Data Flow Diagram of a software system

**There may be as many 'correct' functional models as there are modellers!**





# Assigning functions

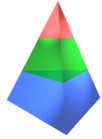
Next to *designing* artefacts with some function in mind, one can *assign* (new) functions to existing objects (whether they are artefacts or not).

Church? Refuge? Playground?



Marketplace? Parking lot? Skeeler area?





## The paradox of Theseus - solution

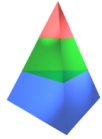
The key to solving the paradox of Theseus is to recognize that there are two ships: the constructional ship and the functional ship.

Is the **constructional ship** the same when all parts are replaced? Obviously not. It is already different when one part is replaced.

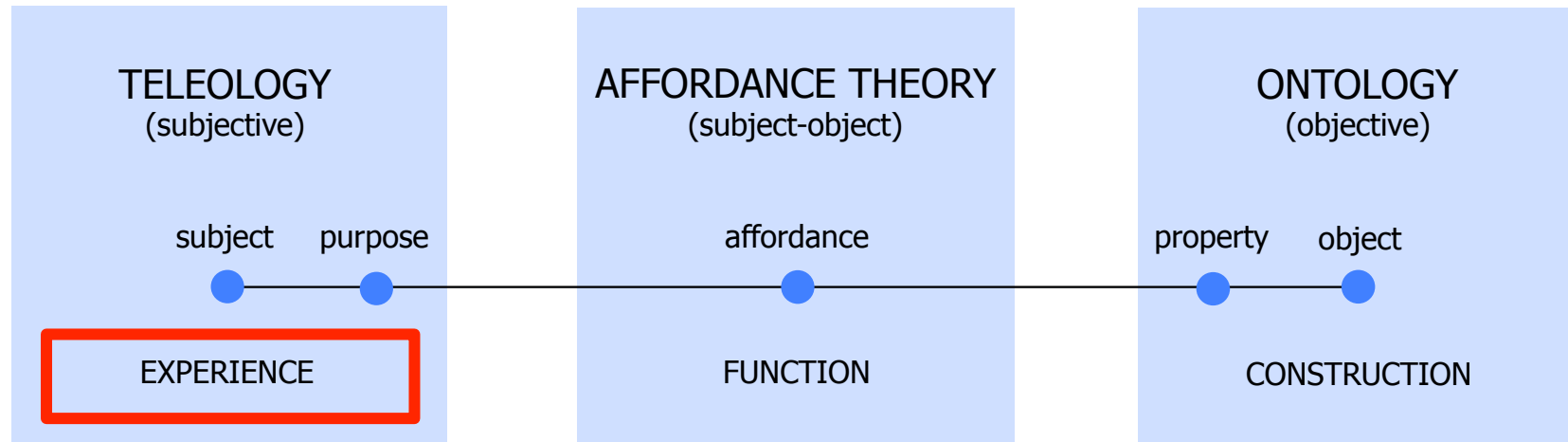
But, the ship remains an instance of the same type. That is why people tend to say that it is the *same* (constructional) ship.

The **functional ship** consists of the affordances that the constructional ship offers to Theseus. As long as this is the case, the functional ship remains the *same* to him.





# Experience

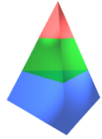


Affordances, whether intended (i.e. functions) or not, evoke *experiences* in the mind of subjects. Examples: feeling safe, looking good.

Experience is purely subjective, although subjects may share the 'same' experience.

Value is a determination of the degree in which an affordance satisfies a purpose of a subject.

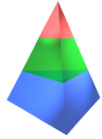




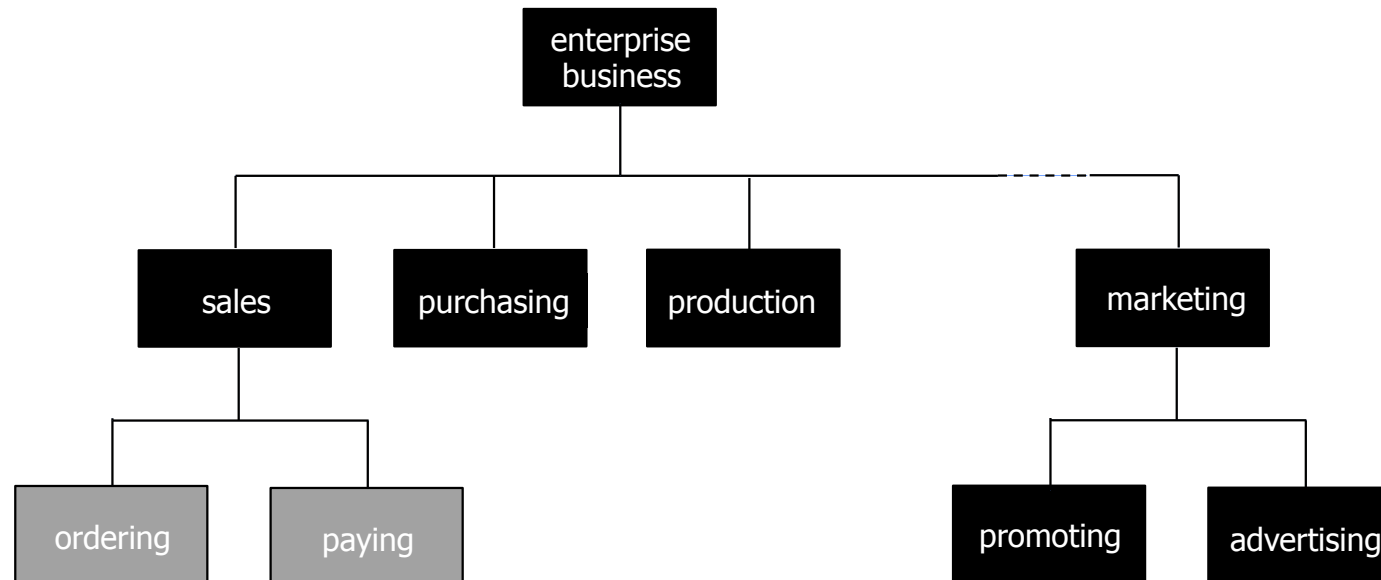
# Experience and implementation

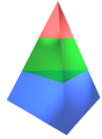
Different *implementations* of (ontologically) the same transaction may offer different *experiences* to the customer.



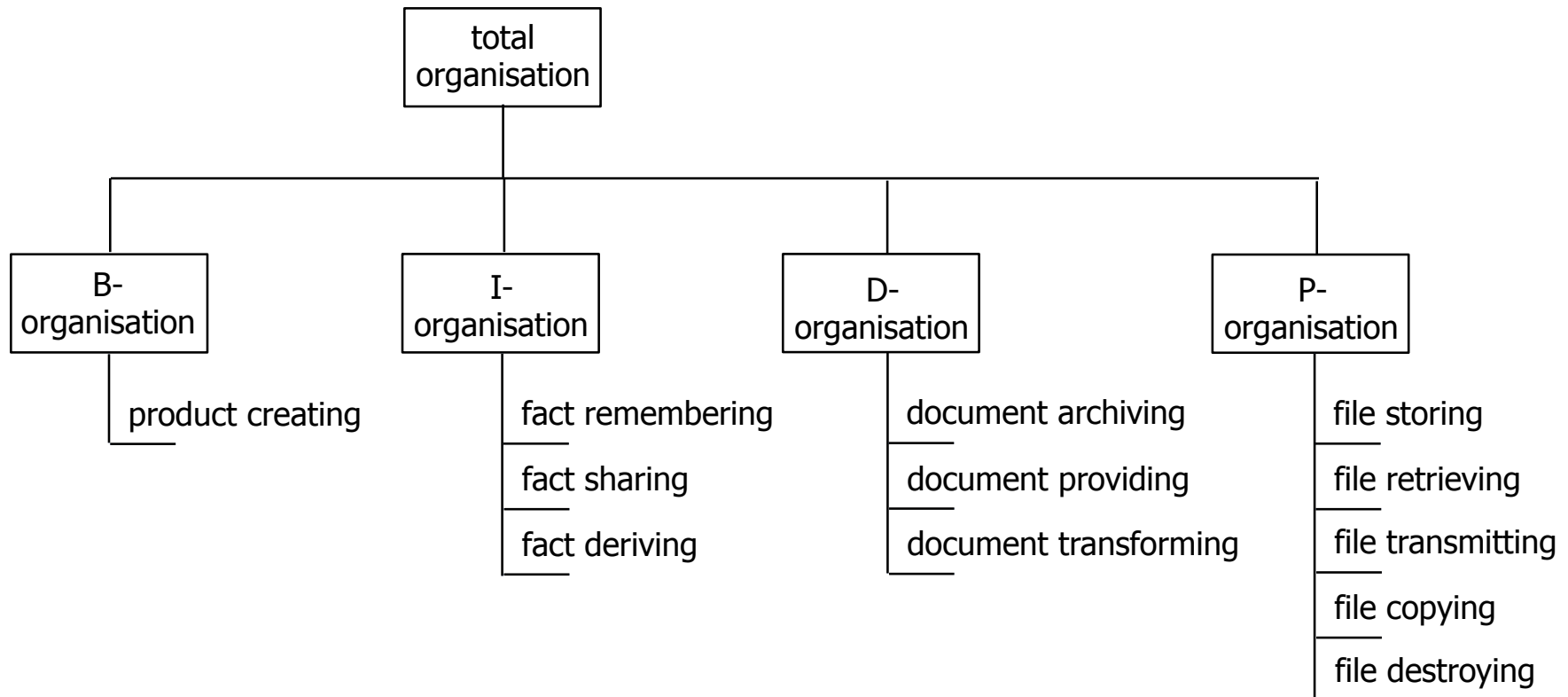


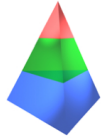
# Decomposition of an enterprise's business



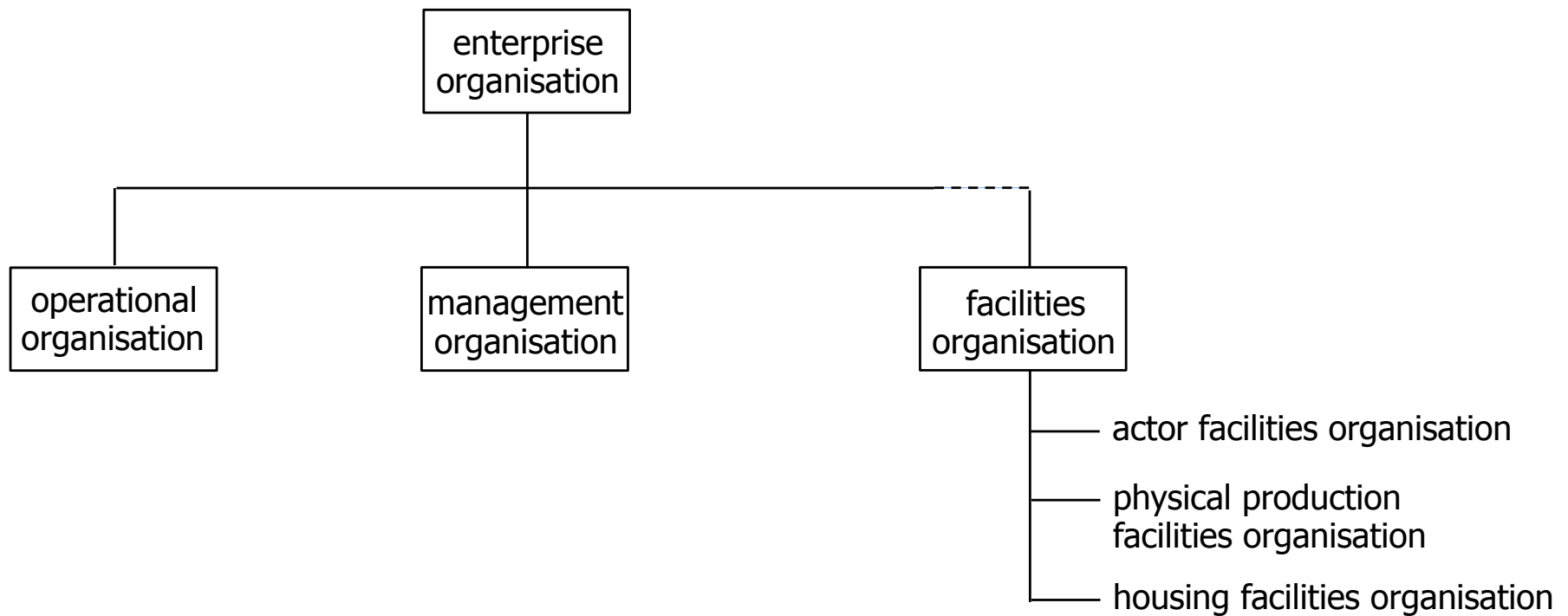


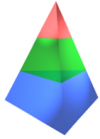
# Generic partitioning of an organisation



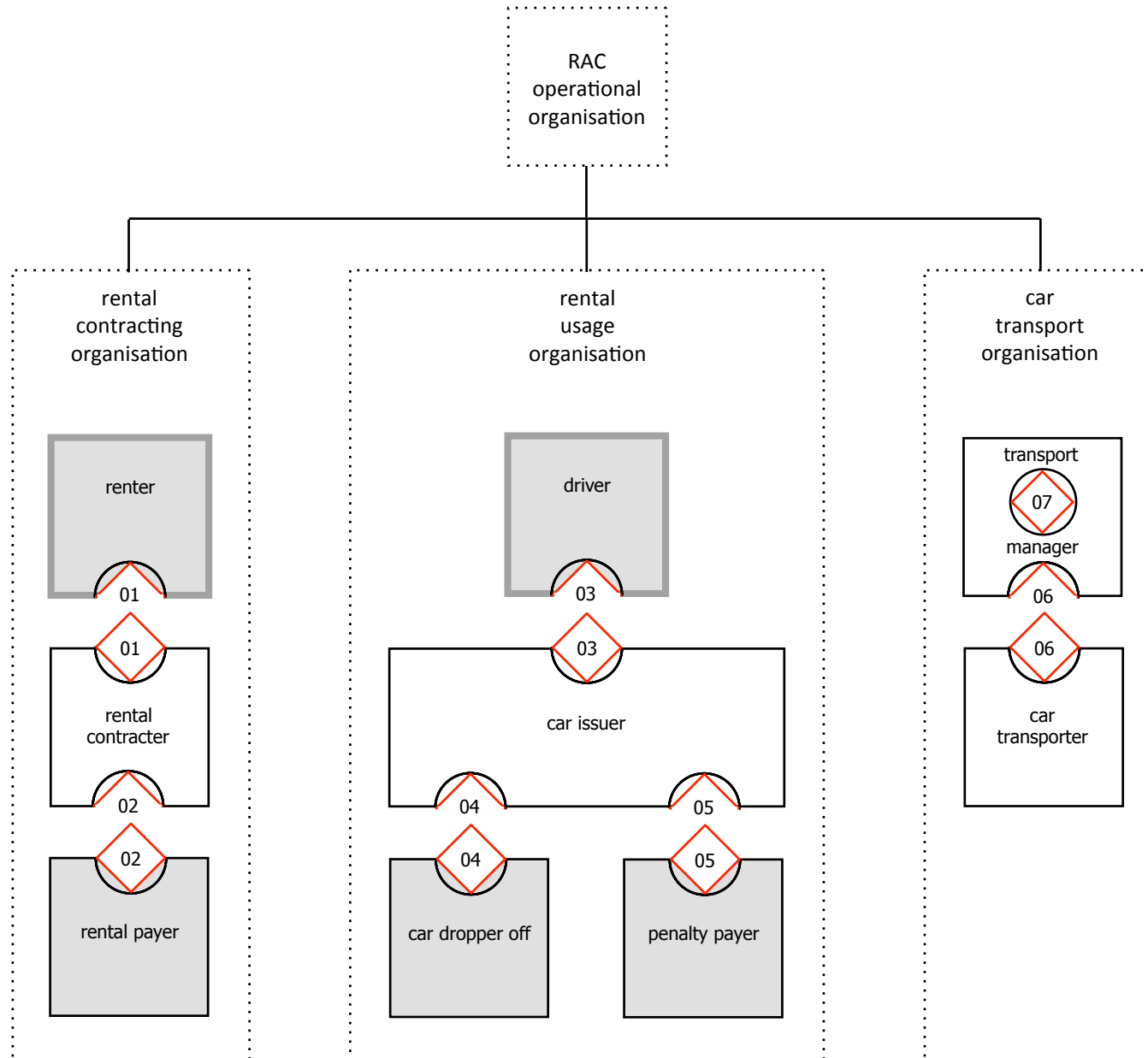


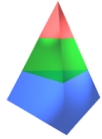
# Generic decomposition of an organisation





# Decomposition of the RAC B-organisation





# Engineering enterprises

## Construction

Assembly of organisational building blocks

## Operating Principle

Entering into and complying with commitments by actors (actor role fulfillers), powered by some power source

## Power source

People fuelled by Belgian beer and Flemish fries

## Operating Theory

The EE-theories, in particular the  $\psi$ -theory

