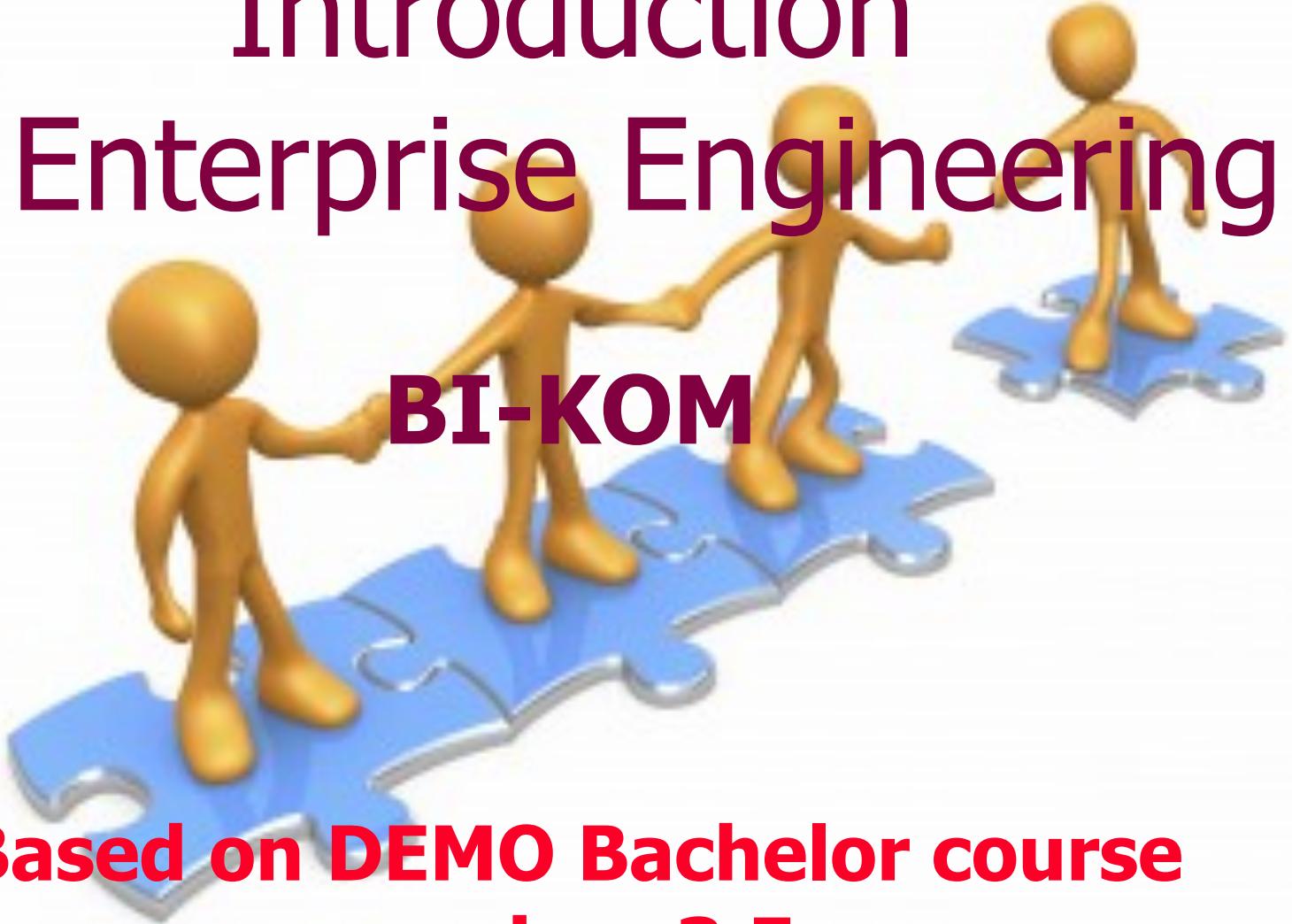


# Introduction to Enterprise Engineering



**BI-KOM**

**Based on DEMO Bachelor course  
version: 3.5**

**Last edition: 6 December 2018**

# Introduction



and

# overview

**Design & Engineering**

**DEMO®**

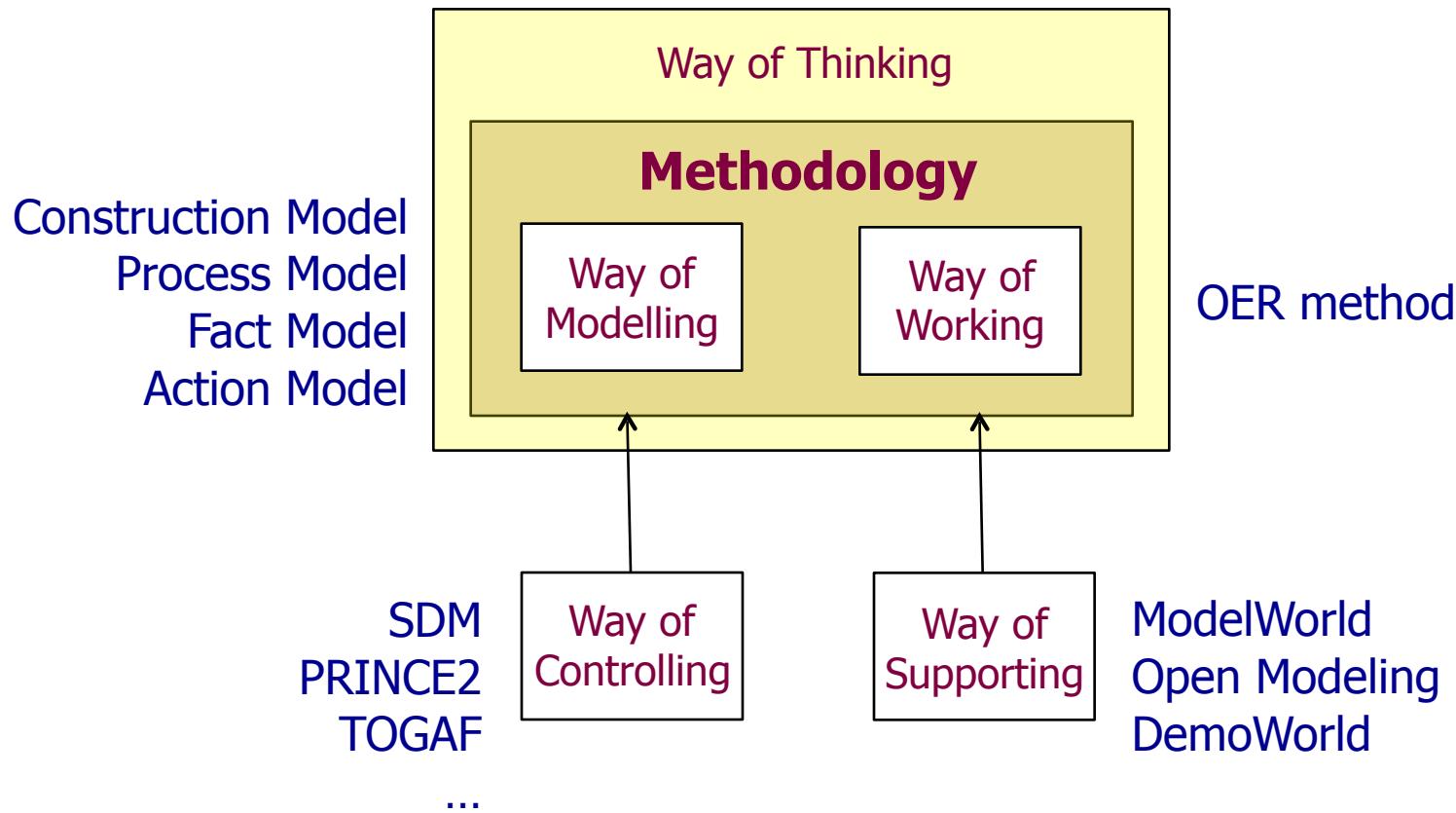
**Methodology for Organisations**

**essence and simplicity**



# DEMO in the Five Ways Framework

MU, TAO, FI; DELTA, PSI, OMEGA; ALPHA, BETA; SIGMA



adapted from P.S. Seligmann, G.M. Weijers, H.G. Sol: analyzing the structure of IS methodologies – an alternative approach, 1989



# The importance of a theoretical basis

There is nothing as practical as a good theory

(Kurt Lewin)

There is nothing as dangerous as a bad theory

(Sumantra Gosha)

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)

Bullshit is a greater enemy of truth than lies are

(Harry Frankfurt)

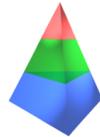


# The key notion of organisation

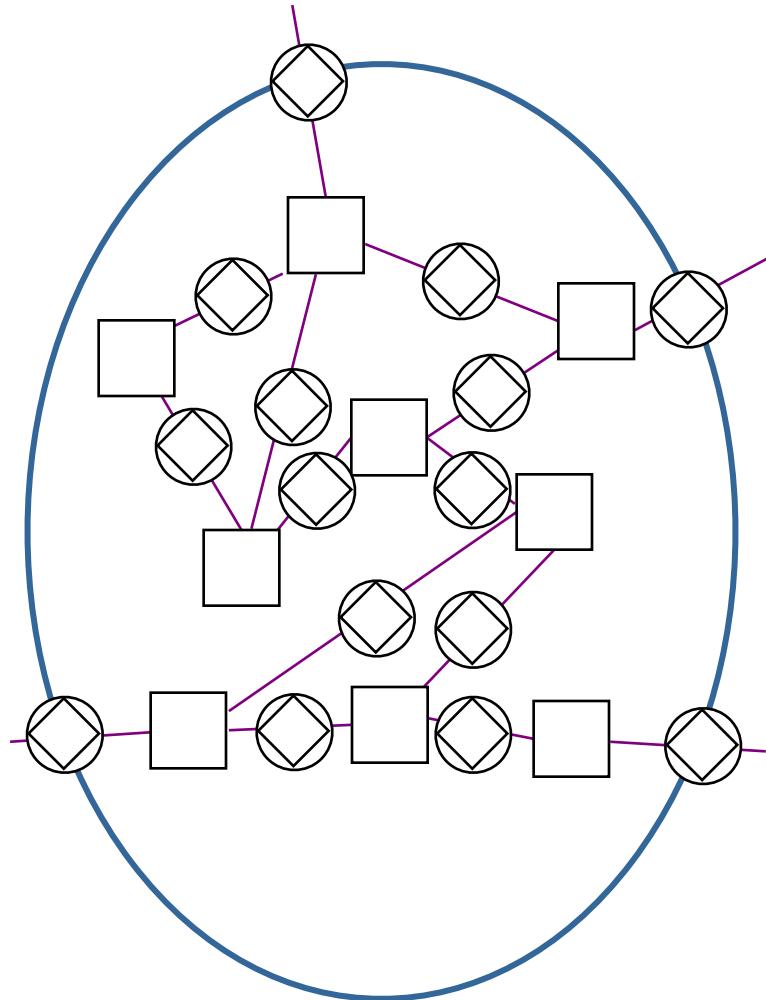


Every organized human activity - from the making of pots to the placing of a man on the moon - gives rise to two fundamental and opposing requirements: the division of labor into various **tasks** to be performed and the **coordination** of these tasks to accomplish the activity.

Henry Mintzberg, The Structuring of Organizations, 1979



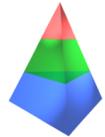
# It's all about production and coordination ...



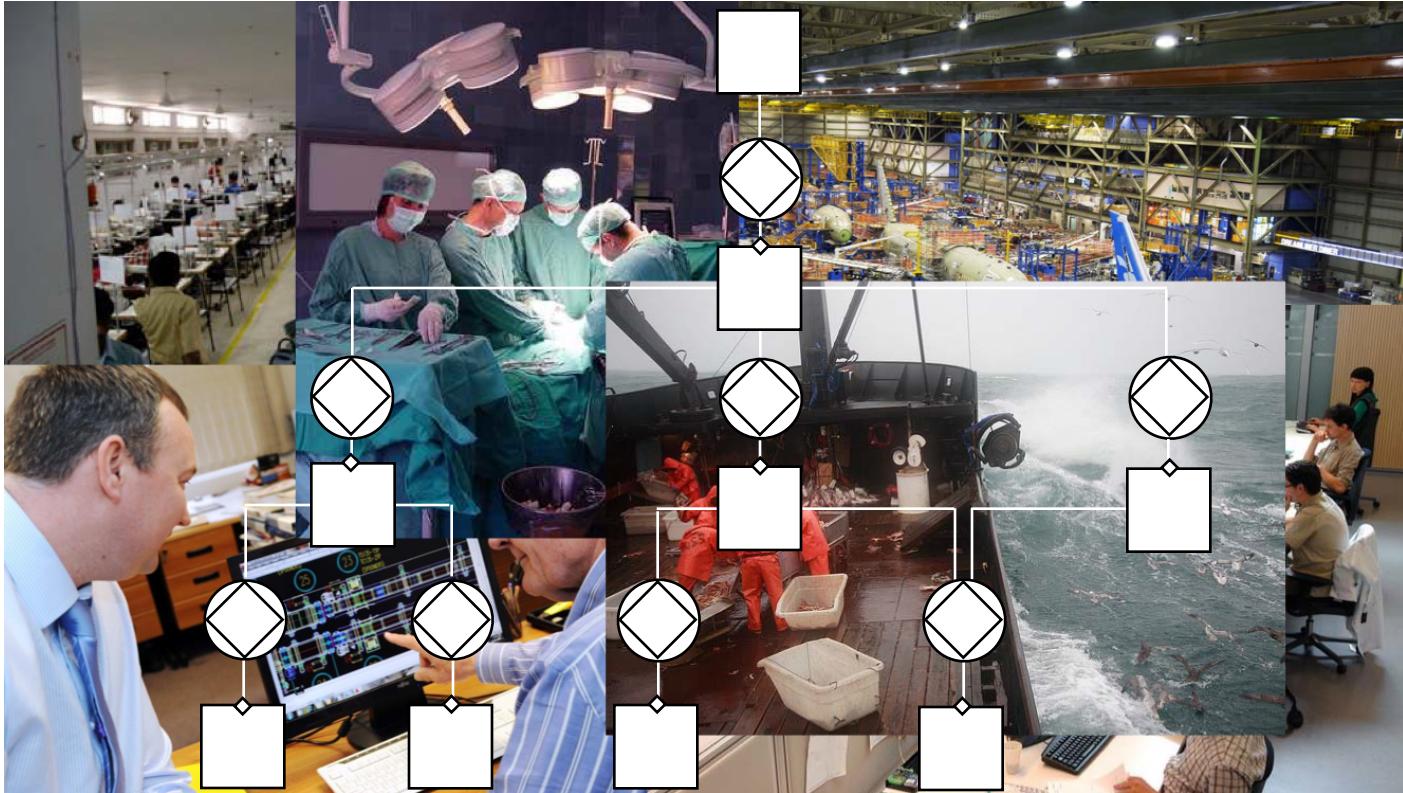
Mintzberg's division of labor is actually a division in **actor roles**: the 'production units' that bring about the (sub and end) **products** of the organisation.

Production and coordination occur in universal patterns, called **transactions**. A transaction comprises 4 to 20 generic **coordination steps** regarding 1 specific **production step**.

The **operating principle** of every organisation is that **actors** in performing **coordination steps**, enter into and comply with commitments regarding a **production step**.

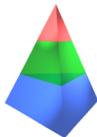


# ... and about organisational structure



Mintzberg states that the **structure** of an **organisation** is simply the sum total of the ways in which it divides its labor into distinct tasks and then achieves coordination among them.

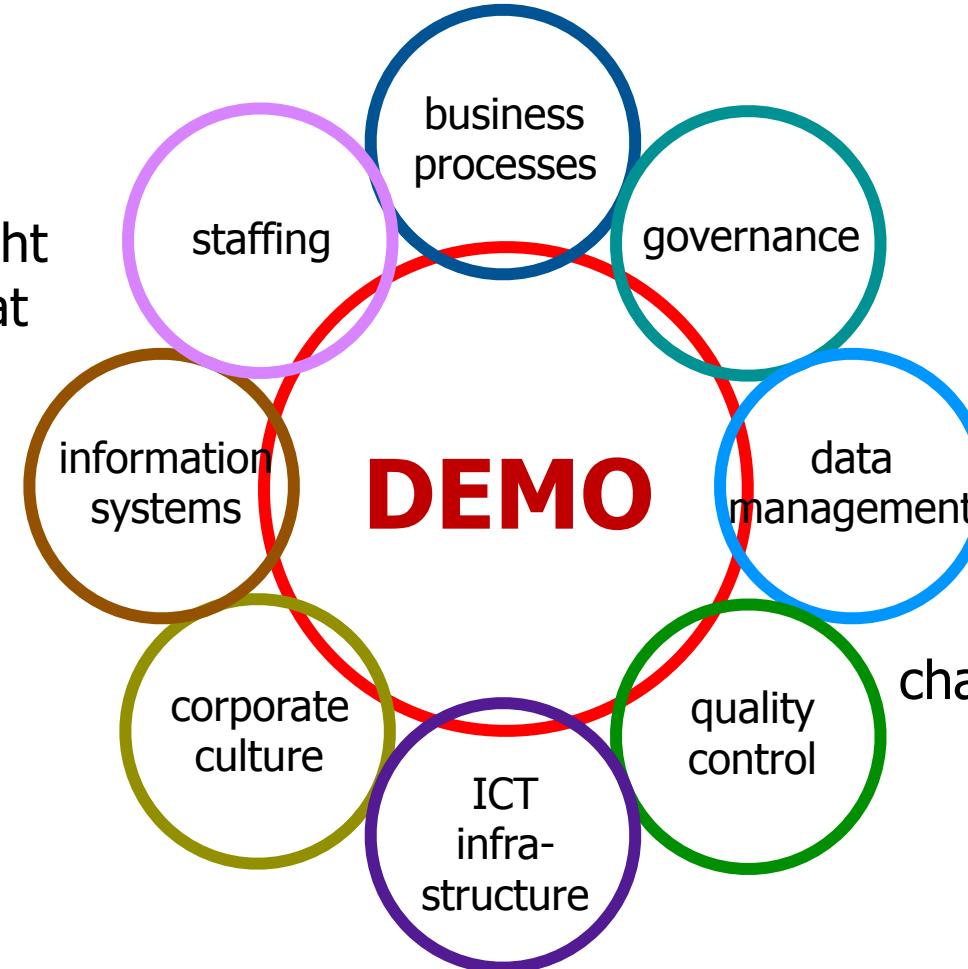
DEMO states more precisely that this structure consists of **trees** of **building blocks** that correspond with the **structures** of **products**.



# DEMO connects people and concerns ...

DEMO offers you an understanding of your organisation that is at the same time abstract and concise as well as accurate and precise

You get the insight  
and overview that  
you need ...



... to manage the  
changes you want to  
bring about

In DEMO, **human centred** and **system centred** thinking are balanced

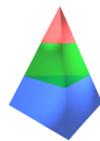
# The problem, and what DEMO offers ...

[http://youtu.be/jx7WH\\_ur\\_94](http://youtu.be/jx7WH_ur_94)

# Way of Thinking

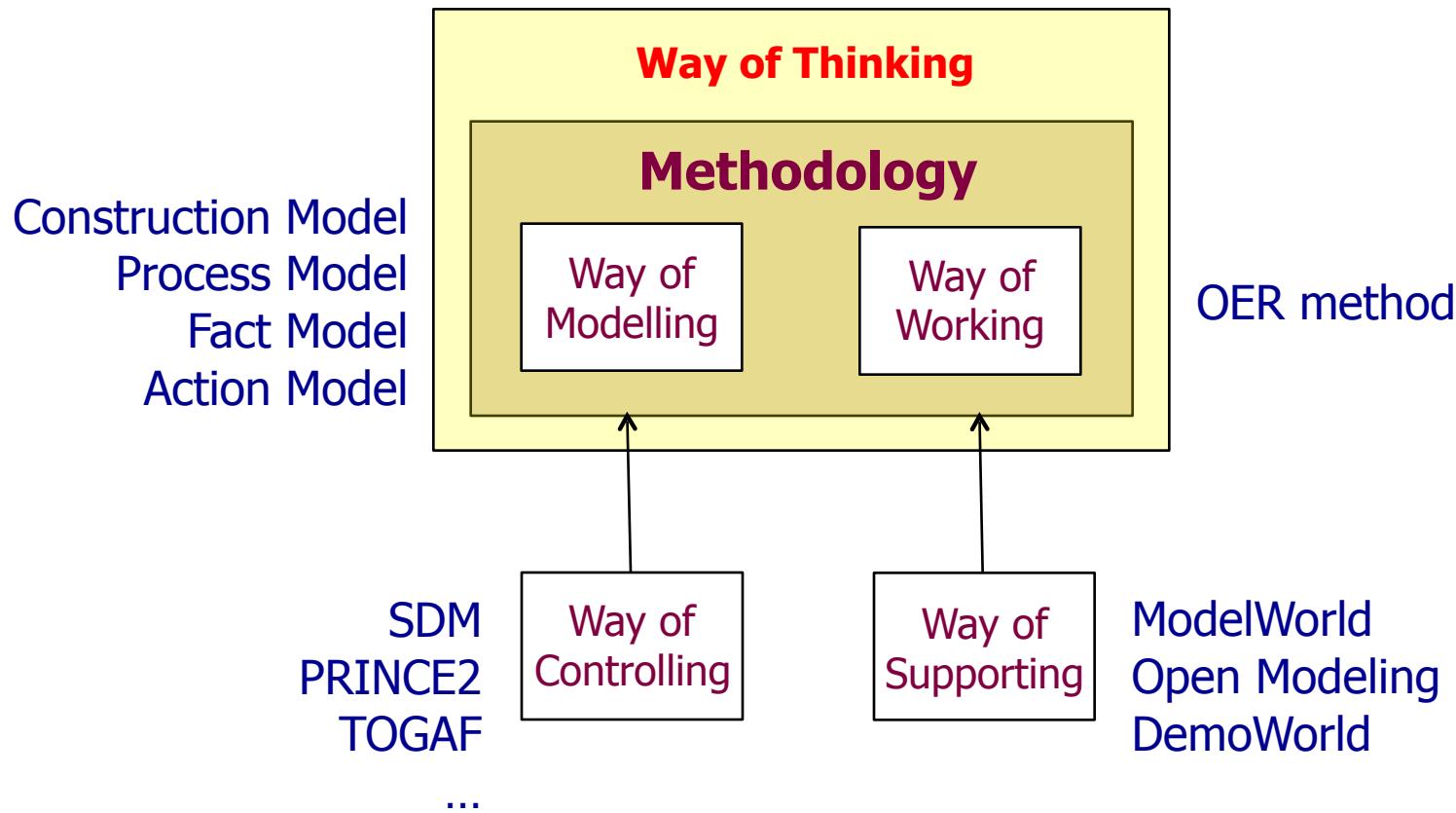


## The PSI theory

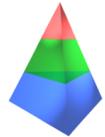


# DEMO in the Five Ways Framework

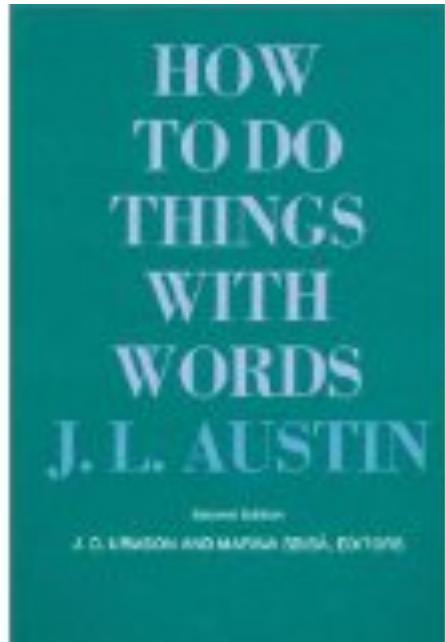
MU, TAO, FI; DELTA, **PSI**, OMEGA; ALPHA, BETA; SIGMA



adapted from P.S. Seligmann, G.M. Weijers, H.G. Sol: analyzing the structure of IS methodologies – an alternative approach, 1989

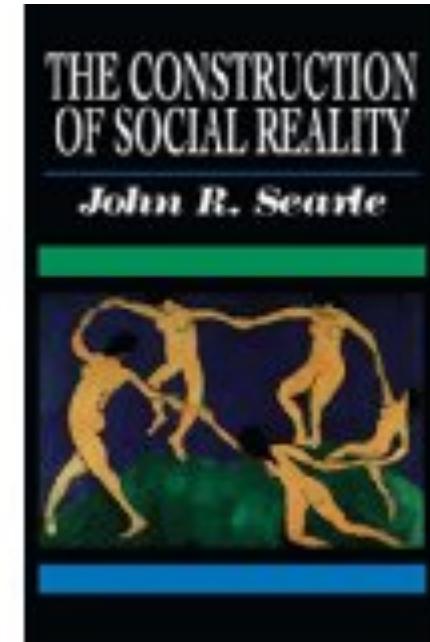
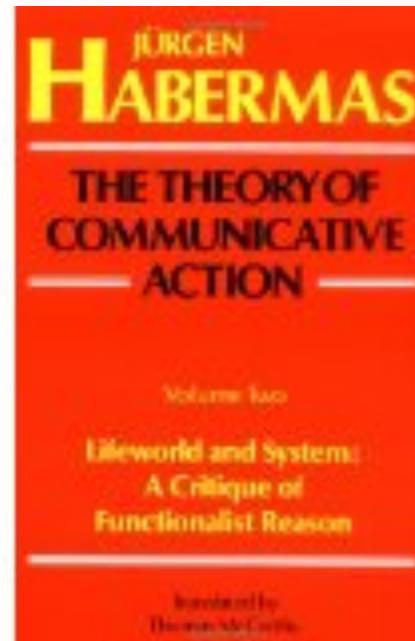


# Roots of the PSI theory

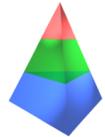


What are  
communicative acts?

What validity claims are  
raised in  
communicative acts?



How are social systems  
(societal institutions)  
constructed in  
communication?



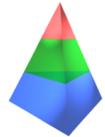
# PSI: Performing in Social Interaction (1)

The PSI theory (standing for “Performing in Social Interaction”) or organisational operation theory, is a theory about the **operation** of **organisations**. It explains how and why people – the pearls of every enterprise – collaborate to bring about the business of an enterprise.

The **operating principle** of organisations is that subjects (human beings) enter into and comply with commitments.

Commitments are raised in **coordination acts** (or C-acts), e.g. the placing of an order for a bouquet of red tulips by a customer in a flower shop. The result of a C-act is a **coordination fact** (or C-fact), e.g. the having been placed of the order.

Every C-act/fact regards a production act/fact (or P-act/fact). An example of a **production act** is the transfer of ownership of a bouquet of red tulips by the salesperson to the customer. The resulting P-fact or **product** is the being transferred of the ownership.



# PSI: Performing in Social Interaction (2)

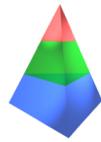
The creation of a C-fact is a C-event. C-events are state changes in the **coordination world** (or C-world) of an organisation. The state of a world is simply defined as a set of facts.

Similarly, the creation of a P-fact is a P-event. P-events are state changes in the **production world** (or P-world) of an organisation.

Every P-fact is the result of a transaction (or derived from original P-facts).

C-acts/facts and their corresponding P-act/fact occur in **transactions**, which are interaction patterns between two actors, one in the role of initiator, the other in the role of executor.

Every successfully carried out transaction results in the becoming existent of a product. A product consists of an independent P-fact and a number of dependent P-facts.

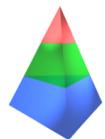


# PSI: Performing in Social Interaction (3)

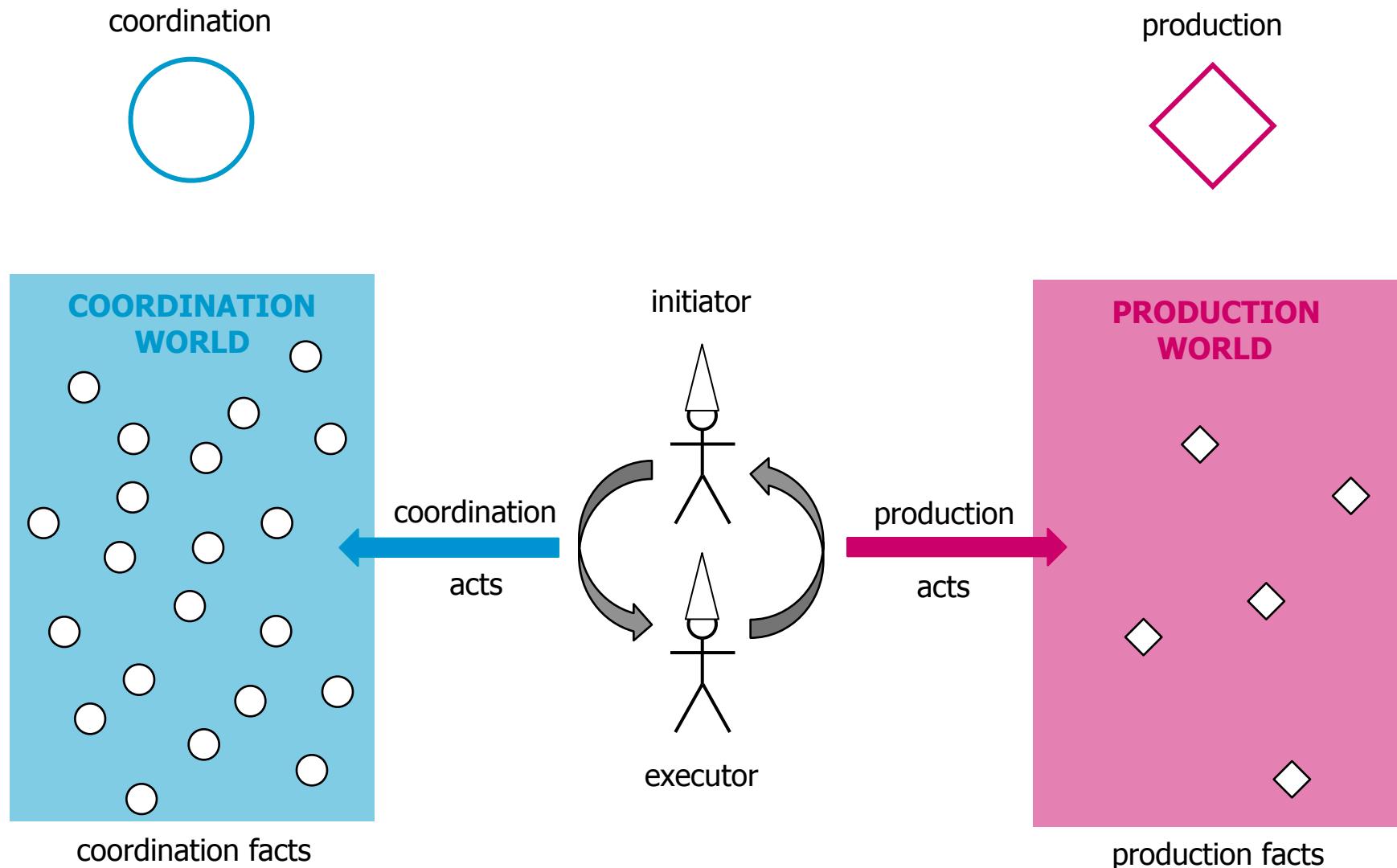
An **actor** is a subject to whom an actor role has been assigned.

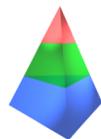
An **actor role** is the unit of authority, namely the authority to be executor in transactions of a specific **transaction kind**.

A subject may be assigned several actor roles (sequentially or simultaneously), and an actor role may be assigned to several subjects (sequentially or simultaneously or collectively).

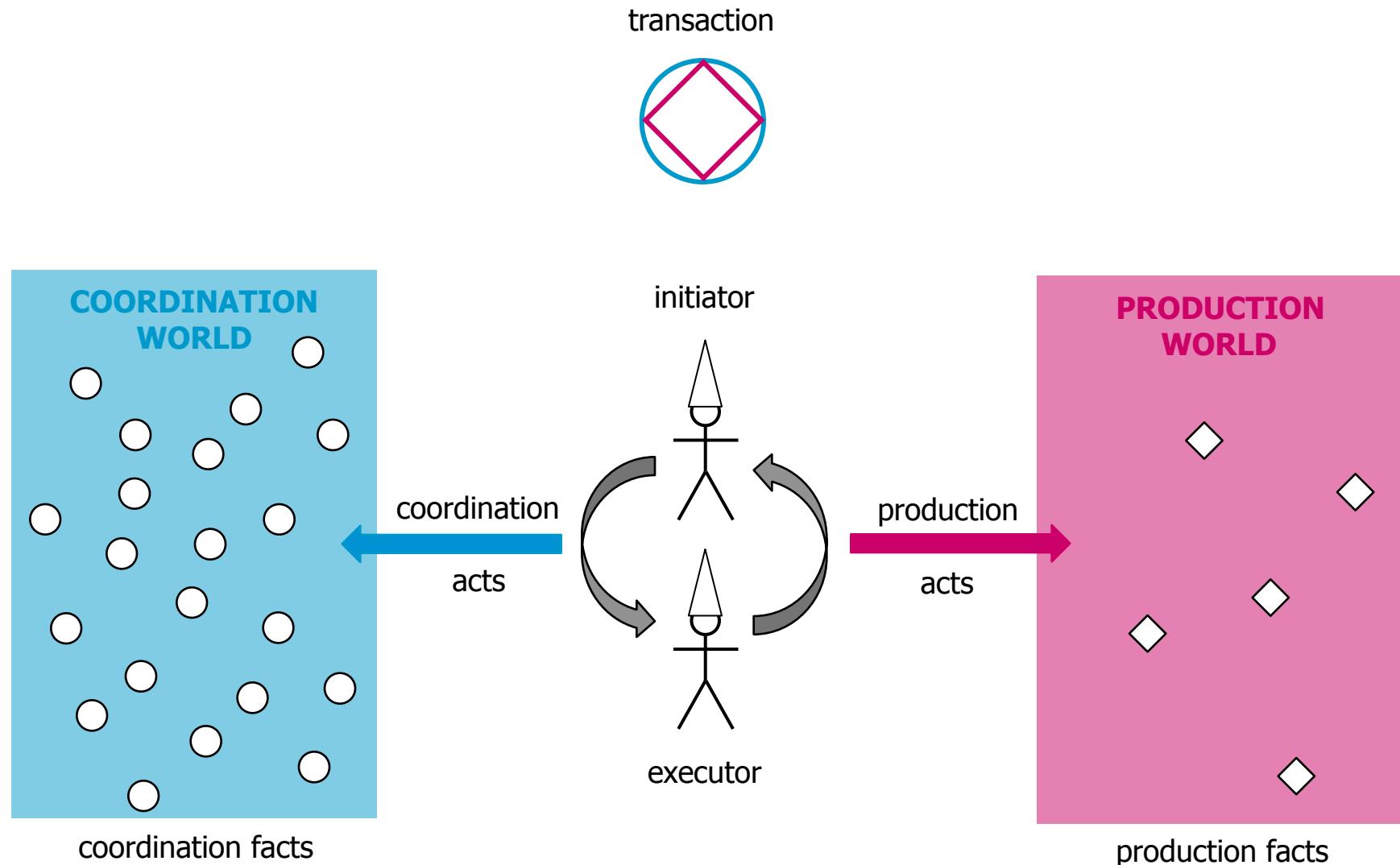


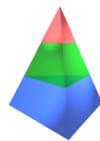
# Transaction = production + coordination





# Transaction = production + coordination

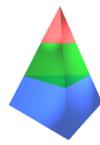




# Is it a P-act/fact or a C-act/fact ?

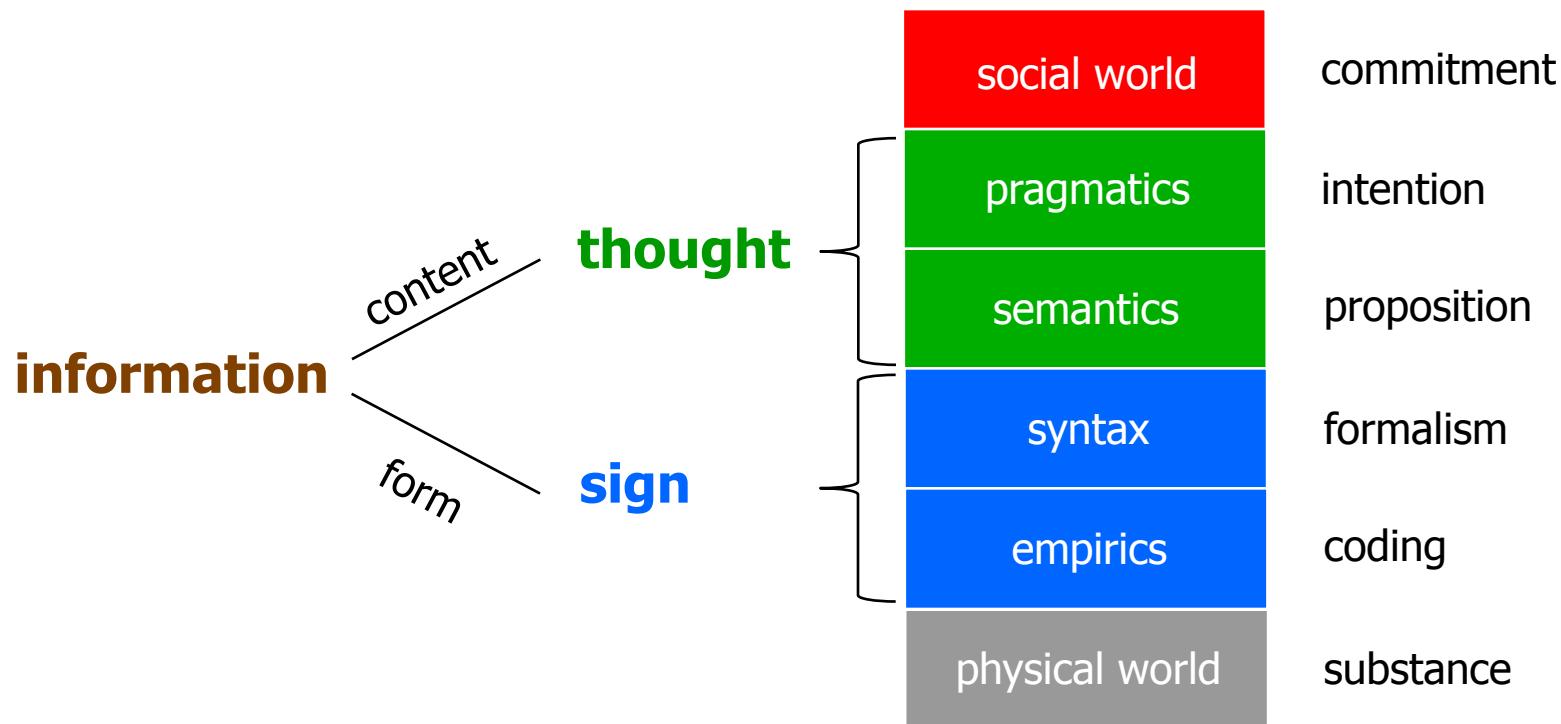
C  
C  
P  
C  
C  
C  
C  
C  
C  
C  
C  
C  
C  
P  
P

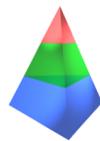
“I’d like to have such a bouquet of red tulips”  
“Ok, I will get one”  
Linda decides to give Jan a bouquet of red tulips  
“Here it is, your bouquet of red tulips”  
“Oh, I think I’d rather have those yellow roses”  
“You are quite capricious, aren’t you?”  
“Come on, I only made a mistake”  
“Alright, so a bouquet of yellow roses”  
Jan sends an invoice to Linda  
Linda receives the invoice  
But Linda says that the invoice is not correct  
Steven asks Hans to sweep the path  
But Hans says that he won’t do it  
Theo sweeps the path with a broom  
The broom is made by Edward



# The semiotic ladder

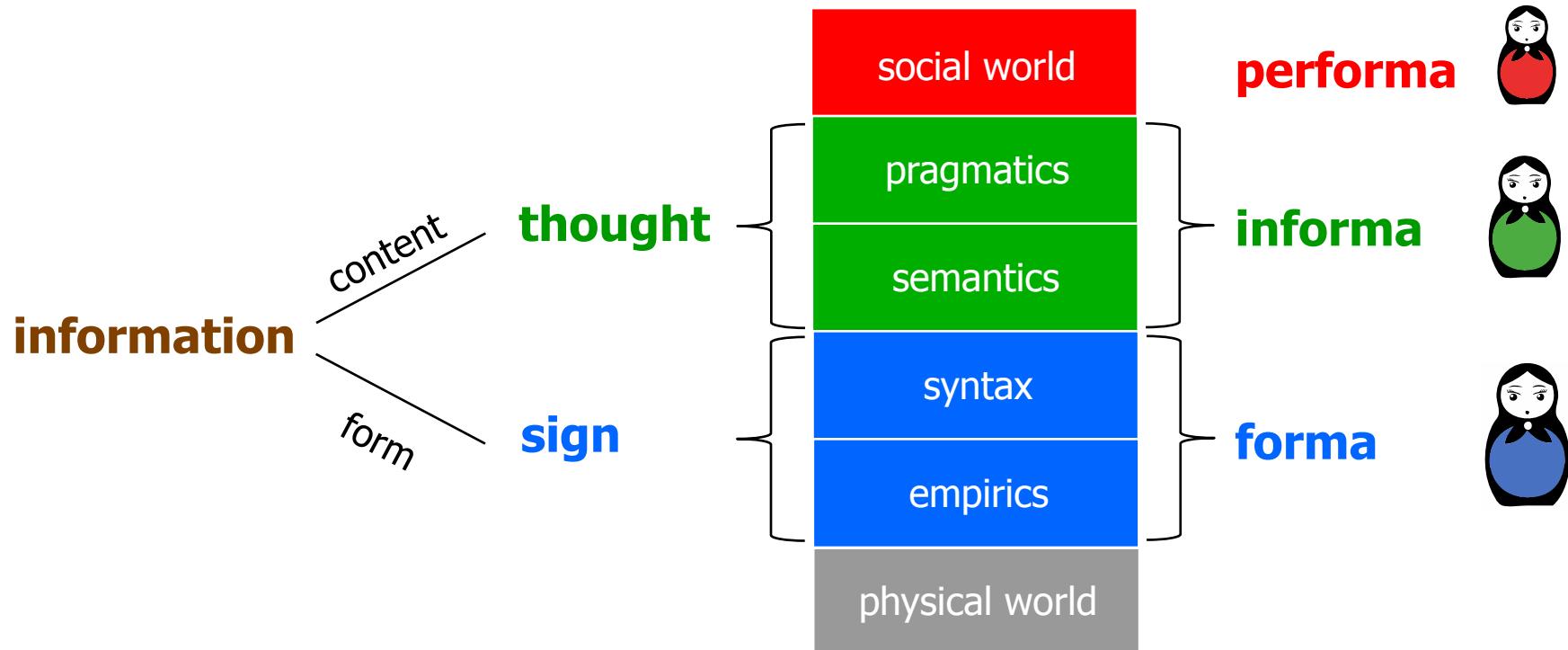
The study of communication and information by Semiotics has yielded the semiotic ladder.

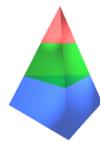




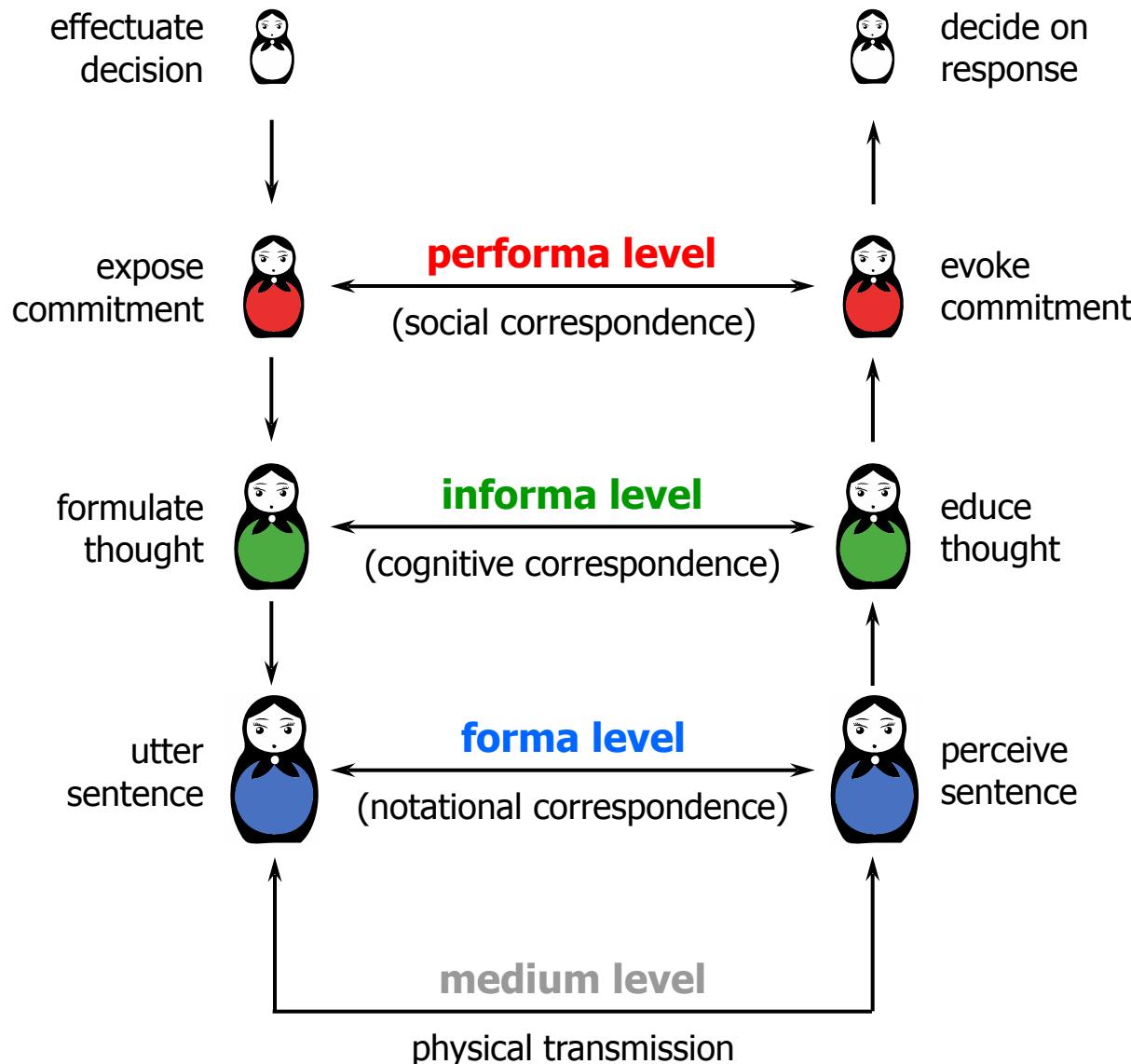
# Human abilities in coordination

Accordingly, three human abilities are distinguished in the PSI theory:  
**performa**, **informa**, and **forma**.





# The process of a coordination act/fact



Me



You



We



In my blue me  
I hear the words  
that you speak



**In my green me  
I grasp the thought  
from the sentence  
that you convey**



**In my red me  
I comprehend  
your intention:  
you want me  
to help you**



**In my blank me  
I decide by the  
reason and the heart:  
I believe you  
I trust you  
I will help you**



I promise my help  
in my red me



I form the thought  
in my green me



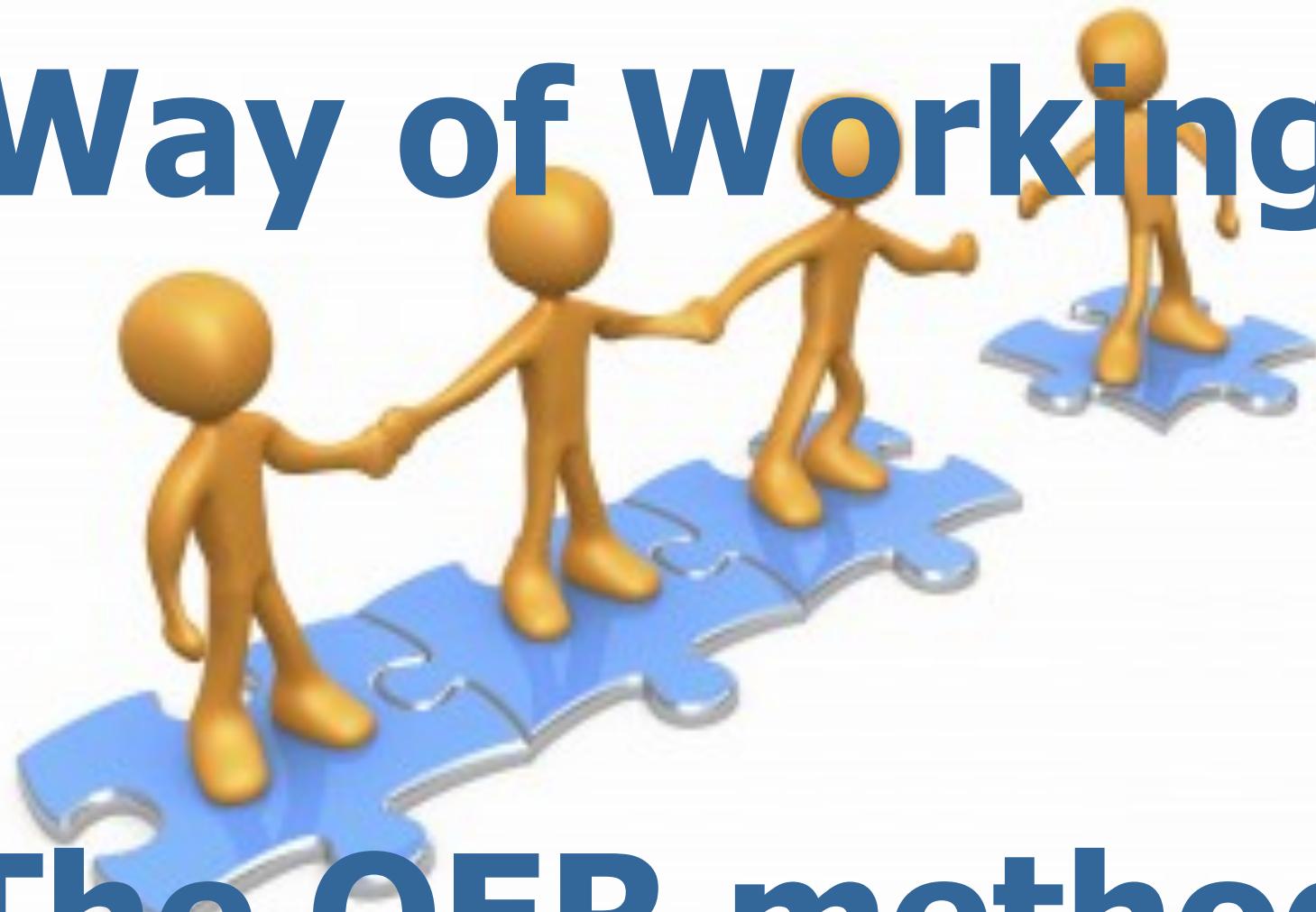
I utter the words  
in my blue me



**and hope that you will hear me**

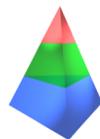


# Way of Working



The OER method

Part 1



# The OER method (1)

The **OER method** (Organisational Essence Revealing) is a method for producing (i.e. revealing) the **essential model** of an SoI.

The idea is to 'look through' the current way in which the enterprise is realised and implemented, and 'see' the '**oer' shape** (Dutch for primal) of the enterprise, by imagining that the employees can only communicate by talking and listening. Consequently, one gets rid of all implementation technology.

The preferred approach is to apply the OER method in a setting of intensive participation by the employees. The second best approach is to analyse existing documents.

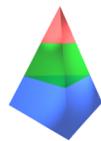
A convenient help in analysing documents, is to colour the text parts **red**, **green** and **blue**. At least colour the relevant '**red**' parts.

When doing this, be aware of the **blue trap**: there may be a hidden '**red**' meaning of apparently '**blue**' things.

# ILLUSTRATION



## Case Volley



# Volley: analysis (1)

One can become member of the tennis club Volley by sending a letter to the club by postal mail. In that letter one has to mention one's surname and first name, birth date, gender, telephone number, and postal mail address (street, house number, zip code, and town). Adam, the administrator of Volley, empties the mailbox daily and checks whether the information provided is complete. If not, he makes a telephone call to the sender in order to complete the data. Once a letter is complete, Adam writes an incoming mail number and the date on the letter, records the letter in the letter book, and puts it in a folder.

Every Wednesday evening, Adam takes the folder to Eve, the secretary of Volley. He also takes the member register with him. If Eve decides that an applicant can become member of Volley, she stamps 'new member' on the letter and writes the date below it. She then hands the letter to Adam in order to add the new member to the member register. This is a book with numbered lines. Each new member is entered on a new line. The line number is the number by which the new member is referenced in the administration.



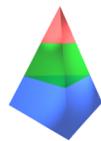
## Volley: analysis (2)

Next, Eve calculates the fee that the new member has to pay for the remaining part of the calendar year. She asks Adam for the annual fee, as decided at the general assembly, which Adam has recorded on a sheet of paper in his files. Then, she asks Adam to write down the amount in the member register.

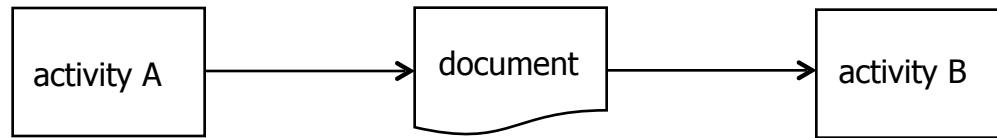
If Eve does not allow an applicant to become member (e.g., because he or she is too young or because the maximum number of members has been reached), Adam will send a letter in which he explains why the applicant cannot (yet) become member of Volley.

If all applications are processed, Adam takes the letters and the member register home and prepares an invoice to all new members for the payment of the first fee. He sends these invoices by postal mail. Payments have to be performed by bank transfers.

As soon as a bank statement is received, Adam prints a card on which the membership number, the starting date, the name, the date of birth, the gender, and the residence are mentioned. The card is sent to the new member by postal mail.



# Applying the ALPHA theory



Is passing the document from A to B:

A **documental** act?

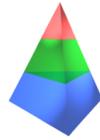
Example: A hands the document over to B for archiving it.

Or (also) an **informational** act?

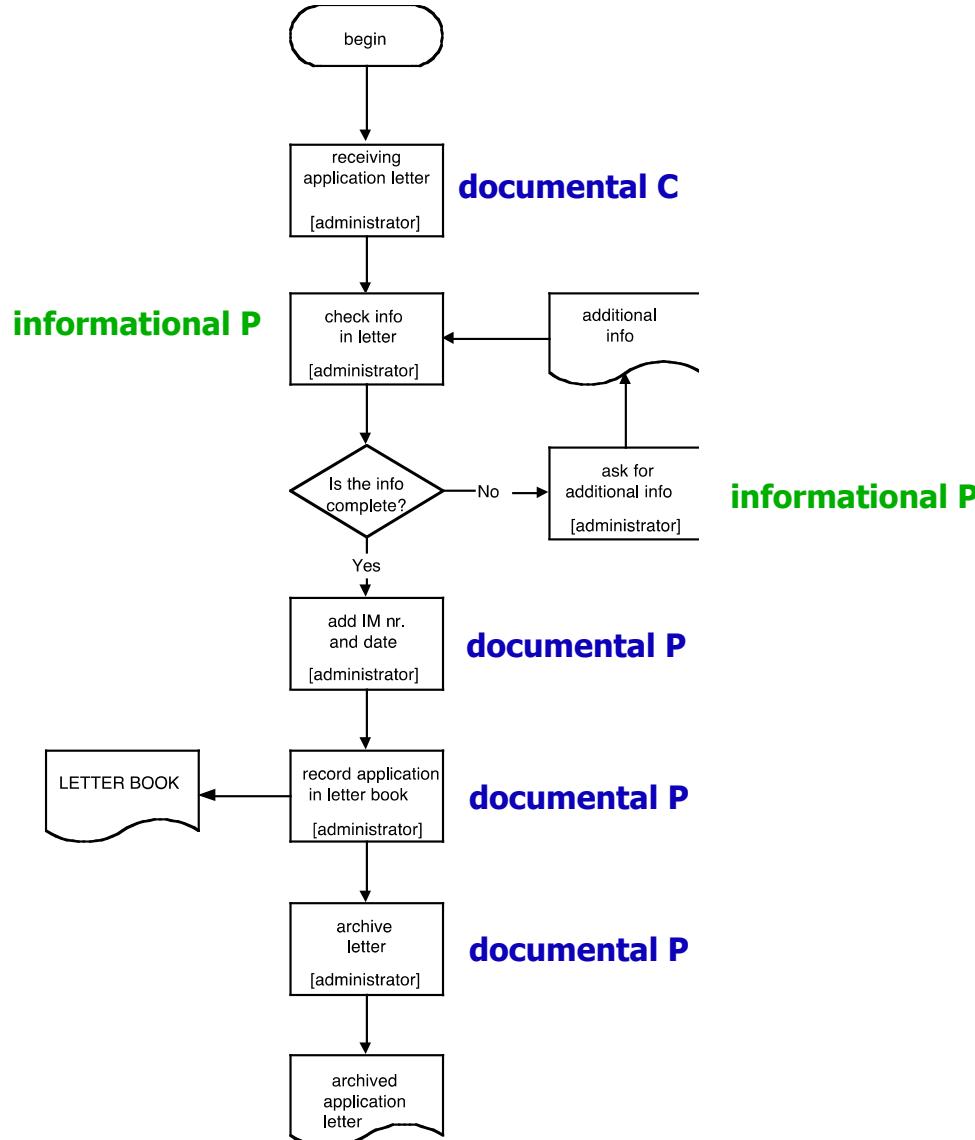
Example: A informs B about the content of the document.

Or (also) an **original** act?

Example: A requests B to do something.

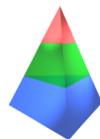


# PSI theory analysis of Volley Flow Chart 1

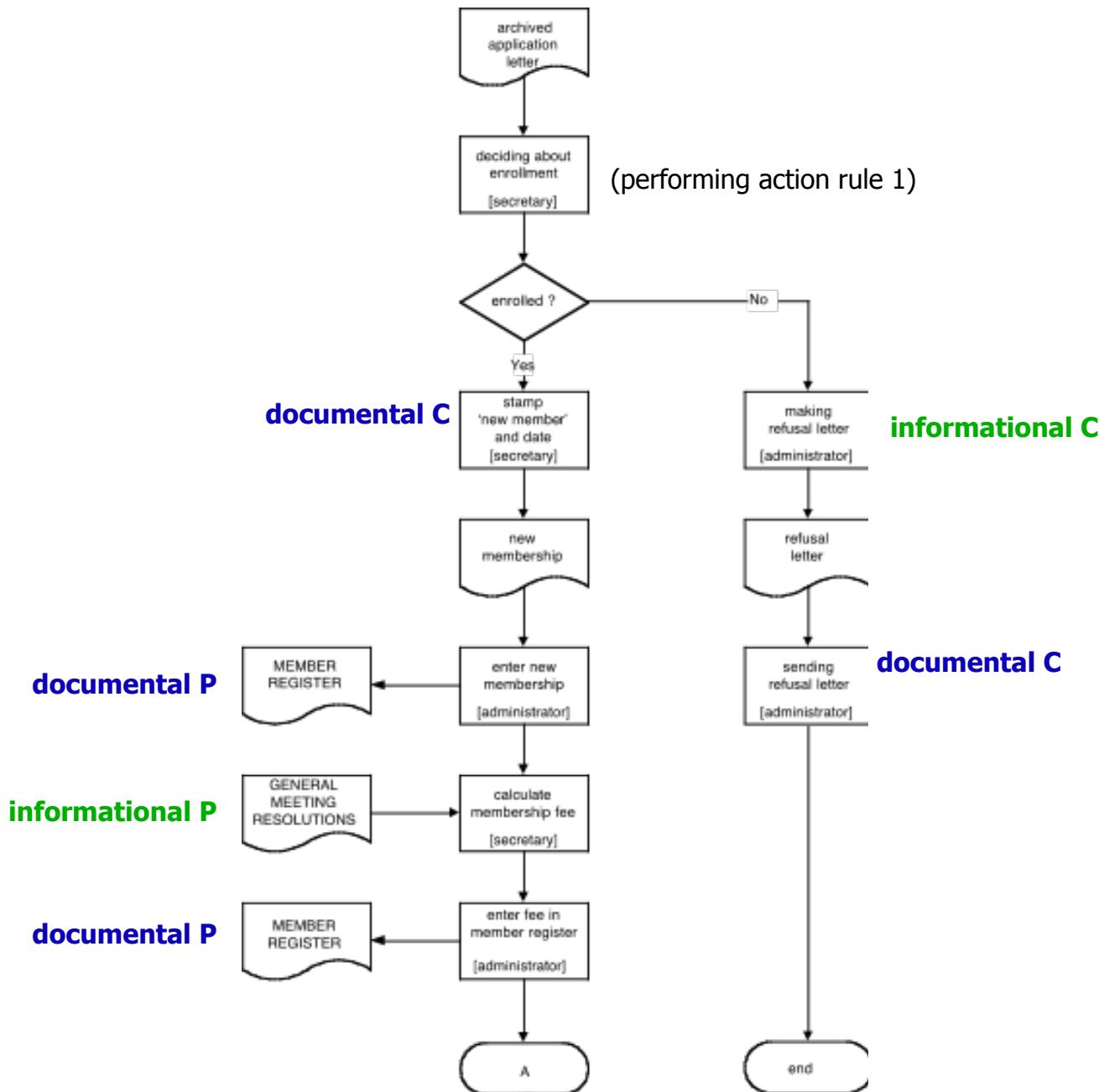


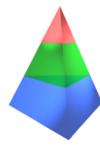
## NOTES

"[ ... ]" means tacitly performed  
"C" means coordination  
"P" means production

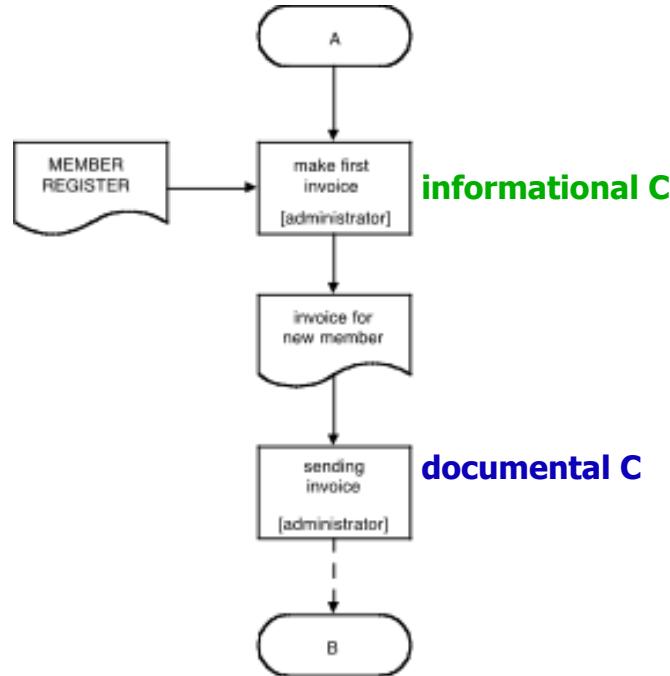


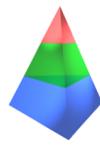
# PSI theory analysis of Volley Flow Chart 2



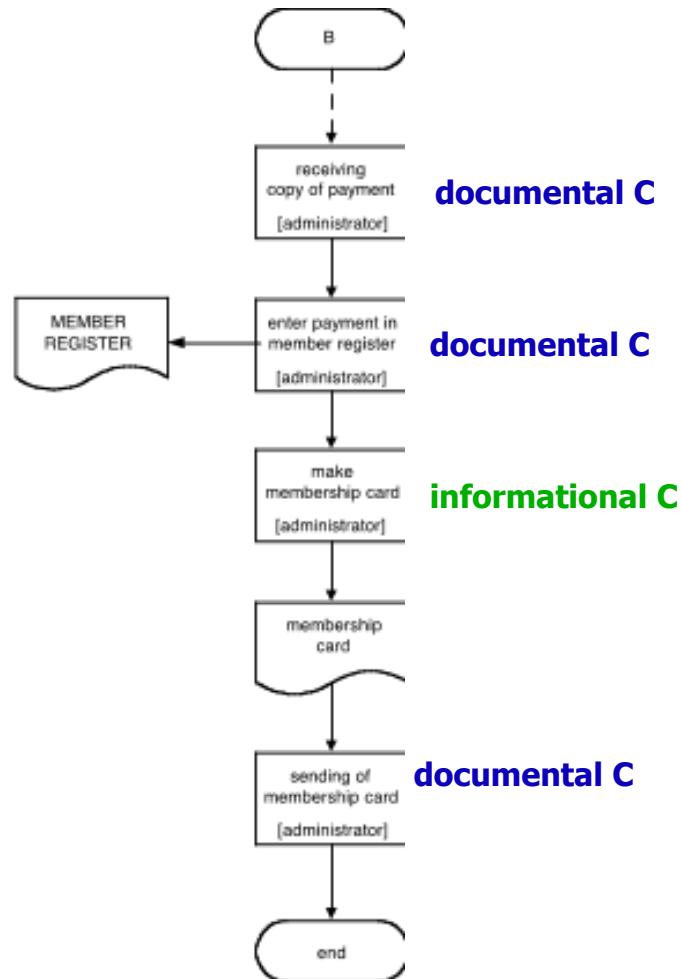


# PSI theory analysis of Volley Flow Chart 3





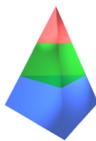
# PSI theory analysis of Volley Flow Chart 4



# ILLUSTRATION



## Case RAC

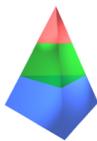


# RAC: analysis (1)

Rent-A-Car (or RAC for short) is a company that rents cars to persons, both private ones and representatives of legal bodies, like companies. It was founded by the twin brothers Janno and Ties back in the eighties. They started to hire out their own (two) cars, and they were among the first companies that allowed cars to be dropped off in a different location than where they were picked up. To this end, Janno and Ties had made agreements with students in several cities. For a small amount of money, a student would await the arrival of a rented car, e.g. at an airport, and drive it back to the office of RAC, after which the student would go home by public transport.

Currently, RAC operates from over fifty geographically dispersed branches in Europe. Many cities have a branch, some even several, and there are branches located near all airports. One of the branches is the original office where Janno and Ties started and where both are still around. Being mechanical engineer by education, they have kept loving to drive and maintain cars, even since they are the managing directors of a million euro company.

The head of the front office of the home branch is Chiara. There are two more desk officers working in this department. Customer orders are placed through several channels: walk-in, telephone, fax, and e-mail. Walk-in customers are usually people who want to rent a car immediately. Through the other channels one makes in general advance reservations. These can be made up to 200 days in advance. In all cases, an electronic rental form is filled out by one of the desk employees, as input to RACIS (RAC Information System). The next groups of data must be provided:



# RAC: analysis (2)

RENTAL: identification number (automatically generated), start date, end date, pick-up branch, drop-off branch, car group.

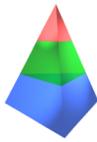
RENTER: identification (passport or driving license), first name, last name, address, date of birth, place of birth.

DRIVER: identification (driving license), first name, last name.

FINANCIAL: rental rate per day (basically determined by the car group).

Although it is the task of the desk officers to take the orders for renting a car, Janno or Ties may drop by and help a walk-in customer or pick up the telephone. Chiara does not really like these 'distortions' but she thinks she cannot do much about it. The problem with these spontaneous actions of Janno and Ties is that they often forget to record things properly, resulting in misunderstandings and even disputes with customers afterwards. Next, they sometimes act against the rules, for example by promising a car for a lower rate than the listed one.

The cars of RAC are divided in car groups. A car group may contain several types (brands and models). The common feature of the cars in a group is that they have the same rental rate per day. The board of directors, i.e. Janno and Ties, decide which brands and models belong to which group as well as what the rental rate is for every group. Normally they do this once a year.

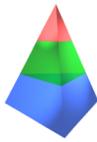


# RAC: analysis (3)

For a walk-in customer the starting day is usually the same day as on which the contract is established. Advance reservations have some future day as the starting day. RAC applies a maximum rental period (currently 10 days).

After the renter has signed the contract, the rental is concluded by the employee (Note: the signing by the renter counts as promising to pay the rental charge, which is the contracted duration times the daily rental rate. Because the rental may be an advance reservation, the payment may be delayed until the starting day).

On the starting day, the driver can pick up a car at the distribution department at the backside of the building, on presentation of a copy of the contract. There are three employees working in this department: Mik, Ferre, and Carlo, but not all of them are always present, as we will see. As soon as a driver shows up, one of them checks whether there is a car available of the contracted group. If there is one, he will allocate the car to the rental contract and sign the contract as being picked up. If there is no car available of the contracted group, he will 'upgrade' the contract and select a car from the next higher car group. The driver will get this 'upgraded' car, but for the price of the contracted group.



## RAC: analysis (4)

After the car of a rental has been dropped off at some branch, the possibly incurred fines have to be paid. There may be a penalty charge for returning the car after the contracted end date. It amounts to the number of extra days times the late return penalty rate. Next, the car may have been dropped off at another branch than the contracted return branch. In that case a location penalty charge has to be paid. This amounts to the distance between the actual and the contracted drop off branch times the penalty rate per kilometer.

The distribution department is also responsible for transporting cars between branches because cars may be dropped off at other locations, as we have seen. To this end, Mik schedules every morning the transportations that have to be performed that day. The transportations are carried out by all three of them, so also by Ferre and Carlo. That is why often some of them are away from the office.

# Way of Thinking

## The PSI theory



# Transactions



# The process of a transaction

A transaction proceeds in three phases:

the **order phase**, the **execution phase** and the **result phase**

The **order phase** is a conversation in which the initiator and executor discuss the product to be produced, and try to agree

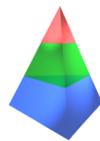
In the **execution phase**, the executor produces a product

The **result phase** is a conversation in which the initiator and executor discuss the product that has been produced, and try to agree

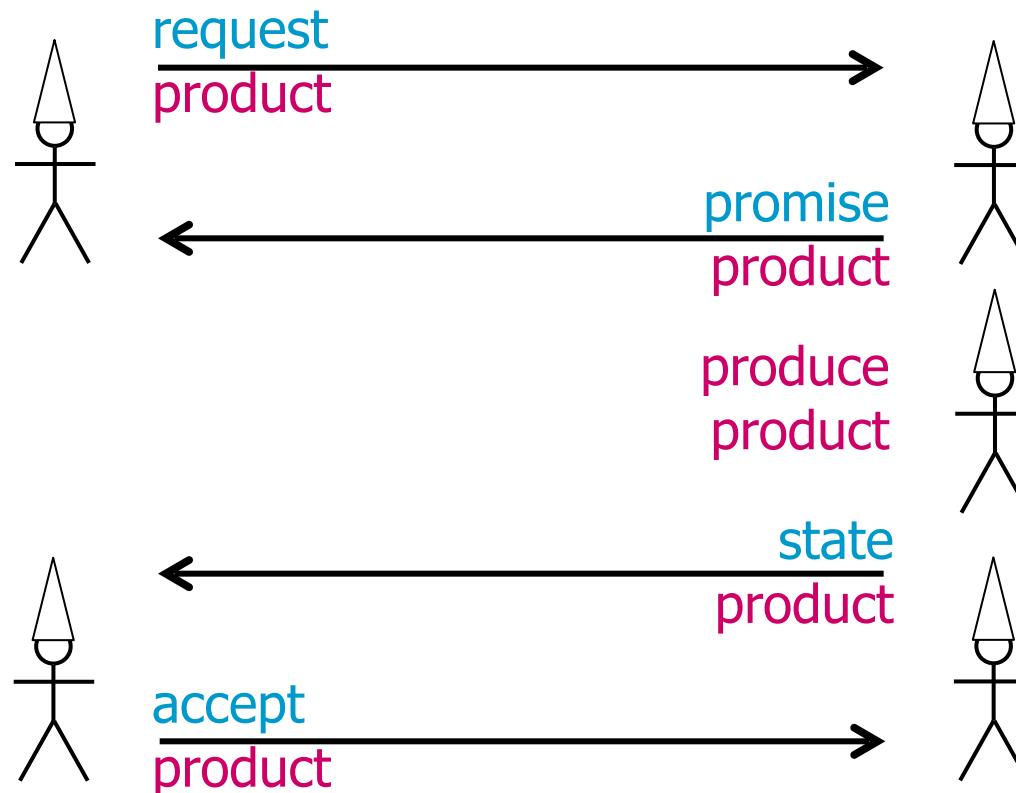


Asking for transport  
Ordering a book  
Applying for membership

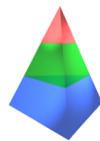
Being transported  
Having got the book  
Having become member



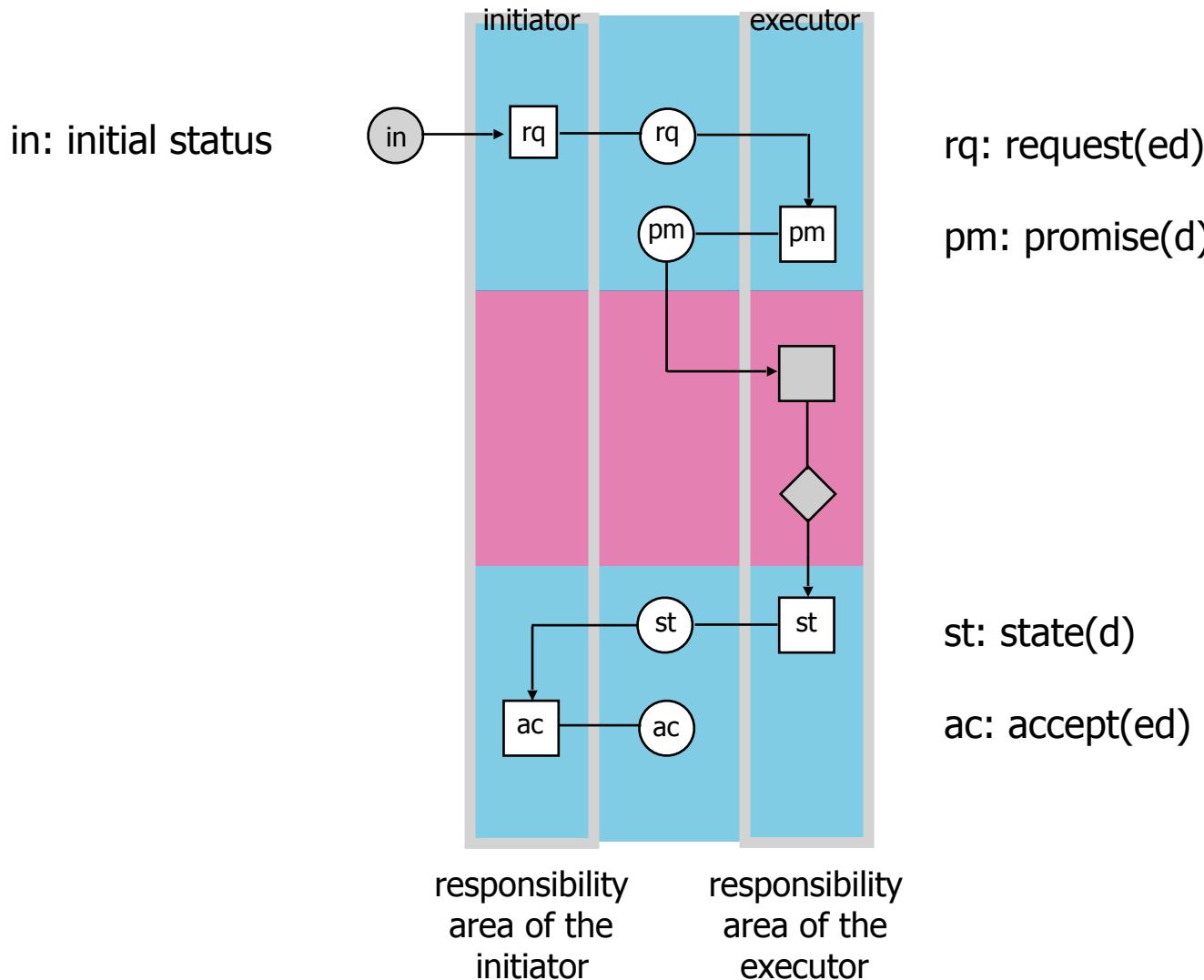
# The basic transaction pattern (1)



NOTE. In order to distinguish the three phases, the product is also called 'proposition' in the order phase and 'result' in the result phase.



# The basic transaction pattern (2)





# What kind of C-act/fact is it? (1)

**request**

“I’d like to have such a bouquet of red tulips”

**promise**

“Ok, I will get one”

**P**

Linda decides to give Jan a bouquet of red tulips

**state**

“Here it is, your bouquet of red tulips”

**?**

“Oh, I think I’d rather have those yellow roses”

**?**

“You are quite capricious, aren’t you?”

**?**

“Come on, I only made a mistake”

**?**

“Alright, so a bouquet of yellow roses”

**reause**

Jan sends an invoice to Linda

**request**

Linda receives the invoice

**?**

But Linda says that the invoice is not correct

**request**

Steven asks Hans to sweep the path

**?**

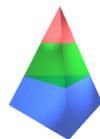
But Hans says that he won’t do it

**P**

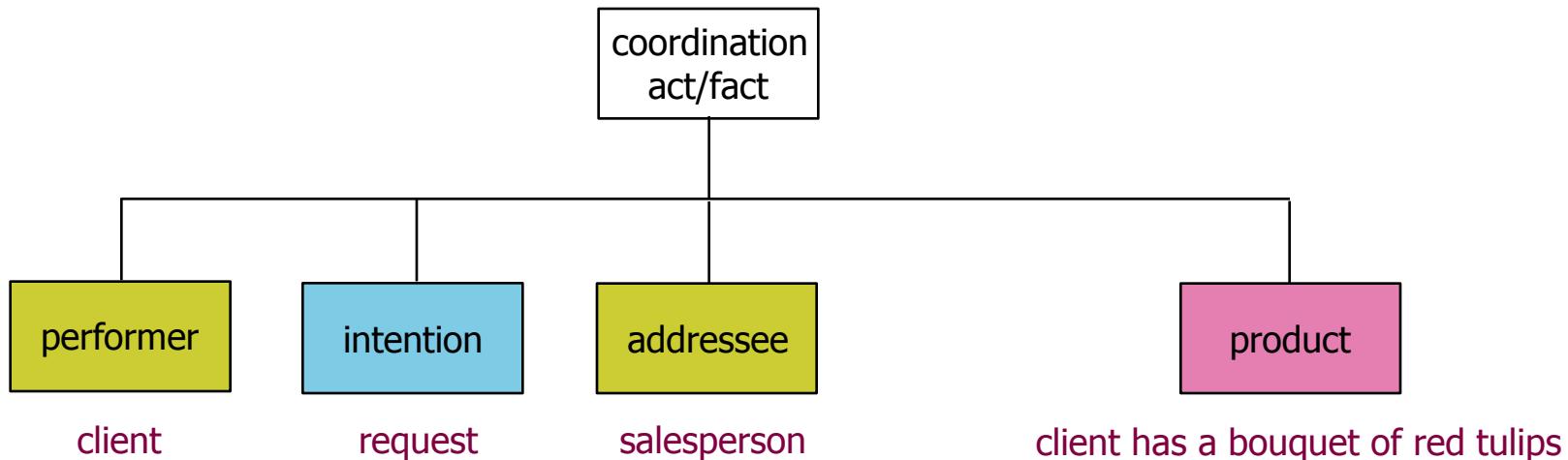
Theo sweeps the path with a broom

**P**

The broom is made by Edward



# The structure of a coordination act/fact



C-acts/facts are the **atomic** components of business processes. Only if and when an actor performs a **C-act**, something happens.

More specifically, a C-act causes a **C-event to** occur, as a consequence of which the corresponding **C-fact** starts to exist.



# Example of a transaction (basic pattern)

C: Hello, I'd like to have a bouquet of red tulips

**C : request :**    **S : client has a bouquet of red tulips**

S: With pleasure

proposition

**S : promise :**    **C : client has a bouquet of red tulips**

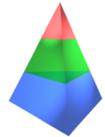
S: Here they are

**S : state :**    **C : client has a bouquet of red tulips**

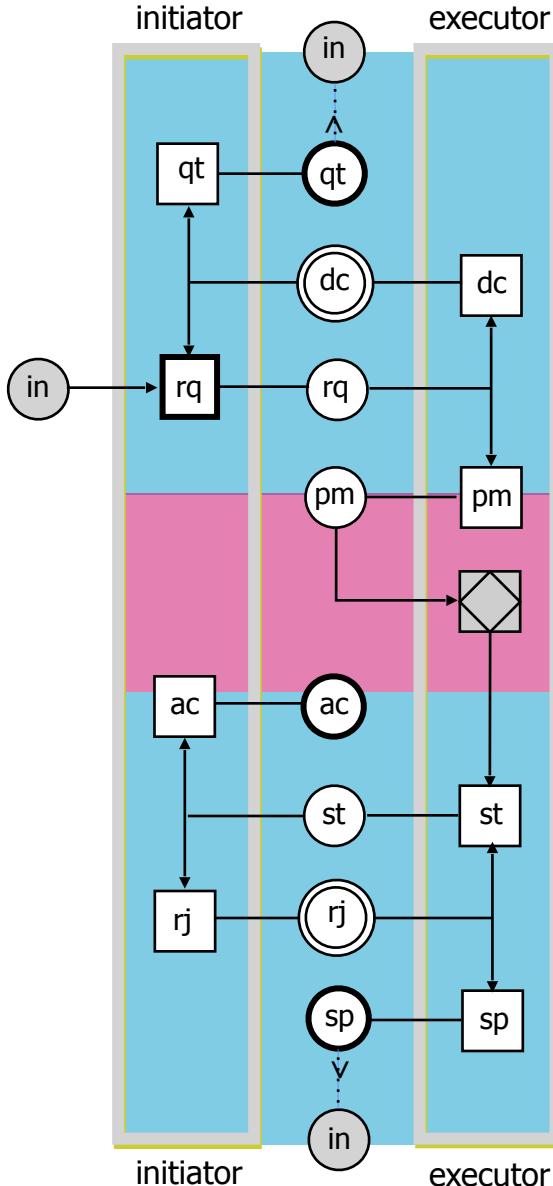
C: Thanks

result

**C : accept :**    **S : client has a bouquet of red tulips**



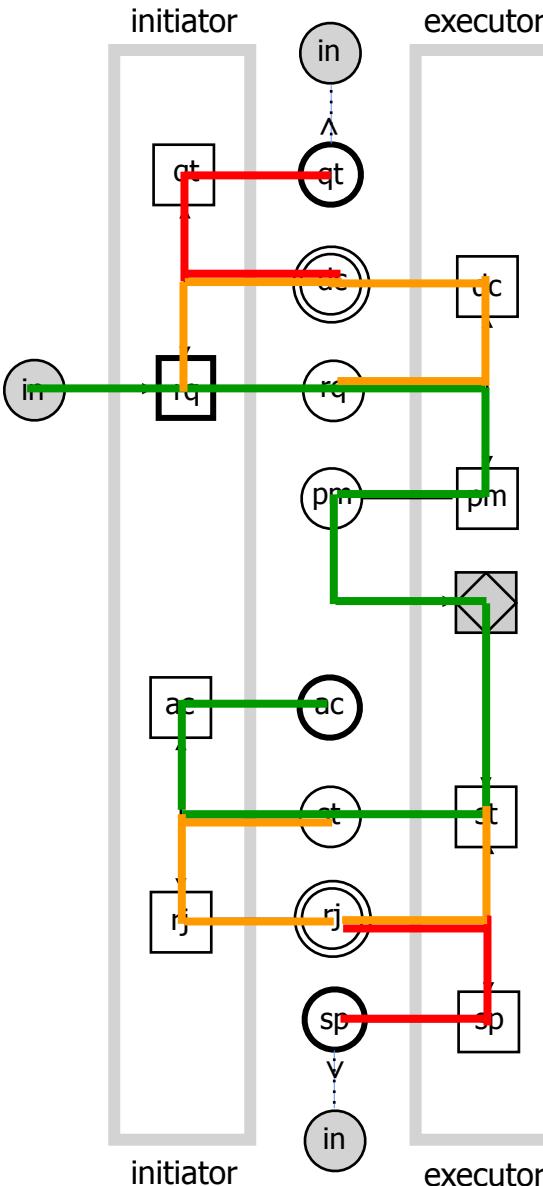
# The standard transaction pattern



**rq:** request(ed)  
**pm:** promise(d)  
**dc:** decline(d)  
**qt:** quit(ted)

in: initial status

**st:** state(d)  
**ac:** accept(ed)  
**rj:** reject(ed)  
**sp:** stop(ped)





# What kind of C-act/fact is it? (2)

**request**

“I’d like to have such a bouquet of red tulips”

**promise**

“Ok, I will get one”

**P**

Linda decides to give Jan a bouquet of red tulips

**state**

“Here it is, your bouquet of red tulips”

**?**

“Oh, I think I’d rather have those yellow roses”

**?**

“You are quite capricious, aren’t you?”

**?**

“Come on, I only made a mistake”

**?**

“Alright, so a bouquet of yellow roses”

**reause**

Jan sends an invoice to Linda

**reause**

Linda receives the invoice

**decline**

But Linda says that the invoice is not correct

**request**

Steven asks Hans to sweep the path

**decline**

But Hans says that he won’t do it

**P**

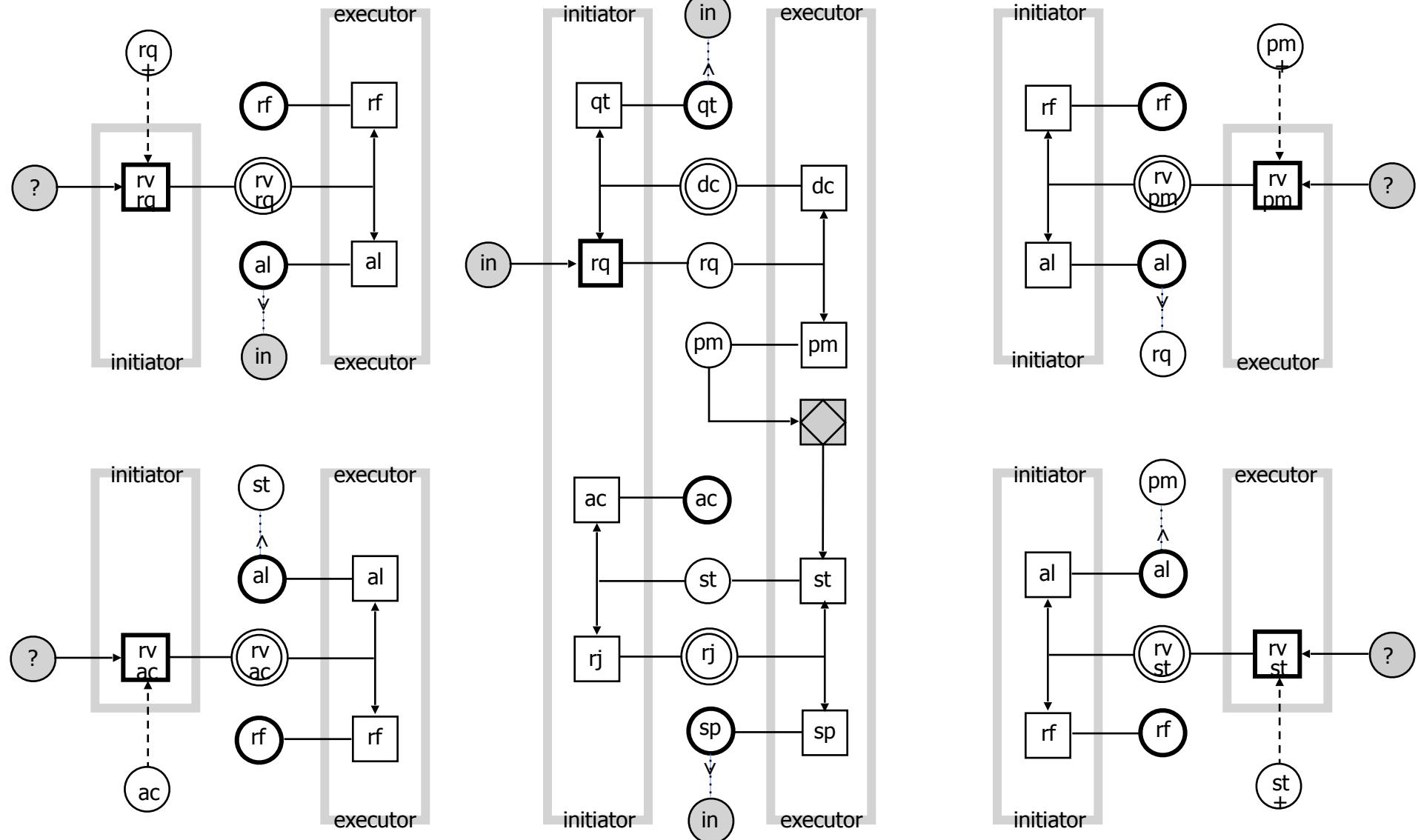
Theo sweeps the path with a broom

**P**

The broom is made by Edward



# The complete transaction pattern



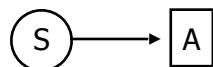


# Legend of the complete transaction pattern

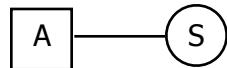
The pattern in the middle of the slide above is the Standard Transaction Pattern (STP). To save space, the symbols of the P-act and P-fact are put together.

To the left and the right of this pattern are the four revocation patterns (rv-rq, rv-pm, rv-st, and rv-ac). They may be initiated from any status in the STP, except qt and sp, because these are transitory statuses.

Bold lined boxes represent the starting acts of a process; bold lined disks represent the terminal statuses. The initial status (in) is colored grey because it is external.



response link: A is performed in response to the occurrence of S



causal link: performing A causes status (C-fact) S



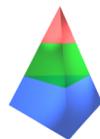
wait link: performing A has to wait until status S is reached



reversion link: reaching status S1 in some process entails the instantaneous reversion to status S2 in another process



S+ means: the (standard) process must be in status S or further

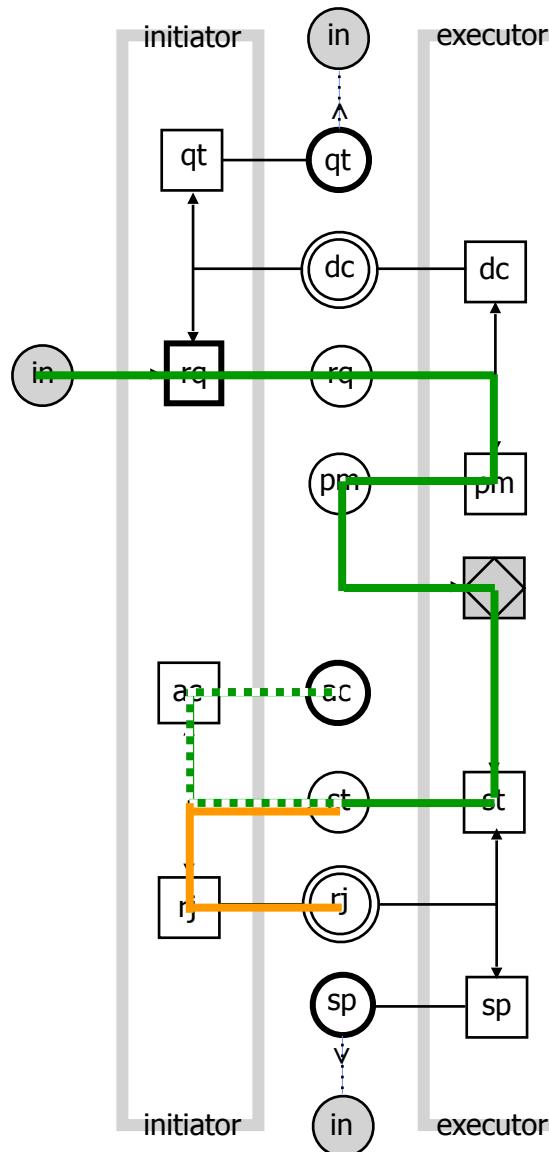
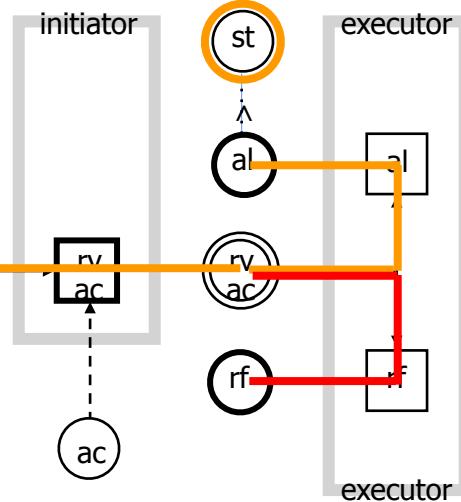


# Revocation of the accept

rq: request(ed)  
pm: promise(d)  
dc: decline(d)  
qt: quit(ted)

st: state(d)  
ac: accept(ed)  
rj: reject(ed)  
sp: stop(ped)

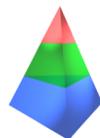
rv: revoke(d)  
al: allow(ed)  
rf: refuse(d)



After having left the flower shop, the customer notes that the bouquet she has got is not really fresh.

She re-enters the shop and explains that she would like to have another bouquet of red tulips.

After the main process has been reversed to the status (st), the client is able to reject this status.

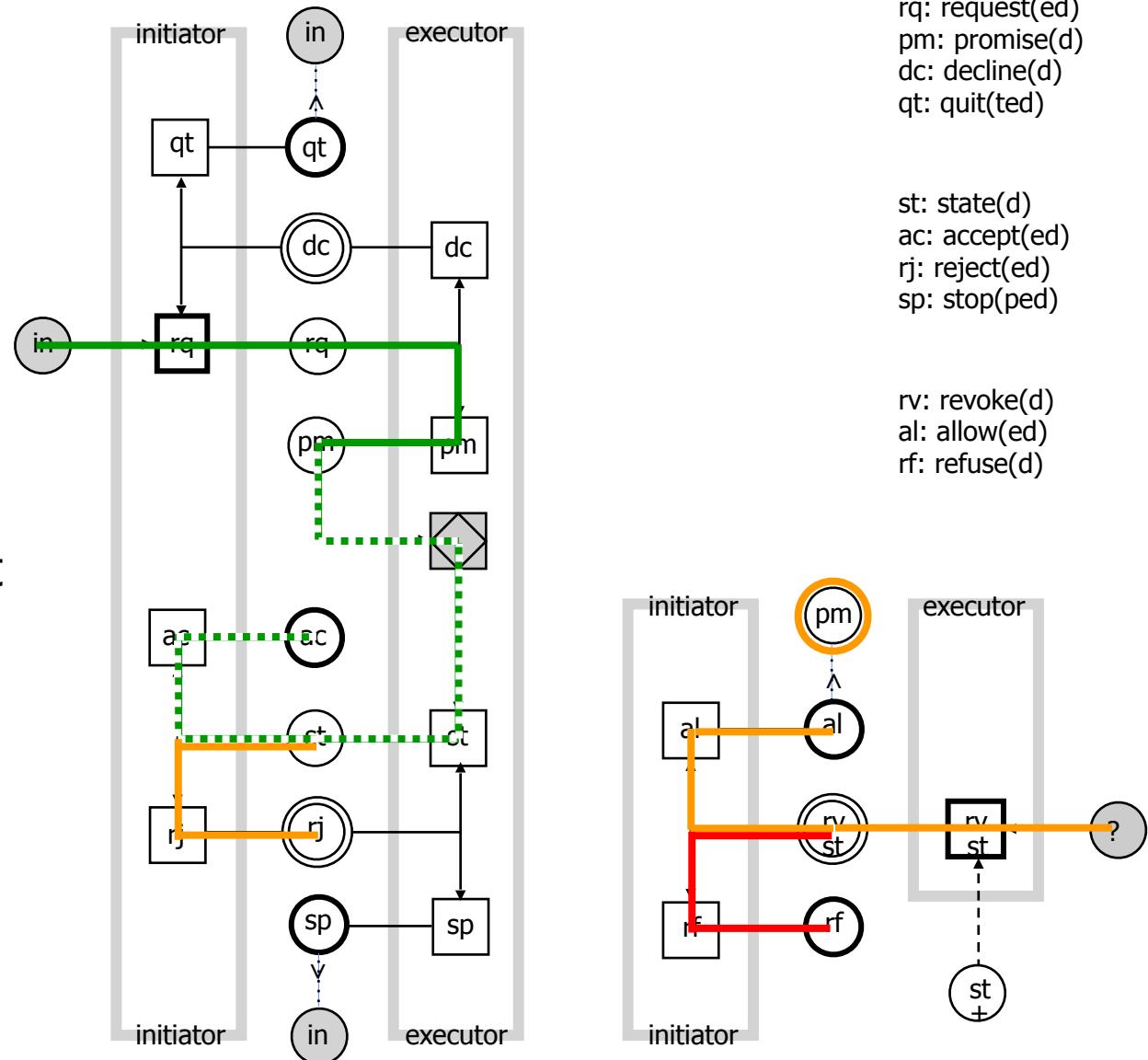


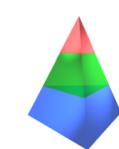
# Revocation of the state

After having agreed with the customer that the sold bouquet is indeed not really fresh, the salesperson takes it back and gets another bouquet.

He hands it over to the customer, who accepts it happily.

After the main process has been rolled back to the status (pm), a new happy flow is taking place from (pm) to (ac).



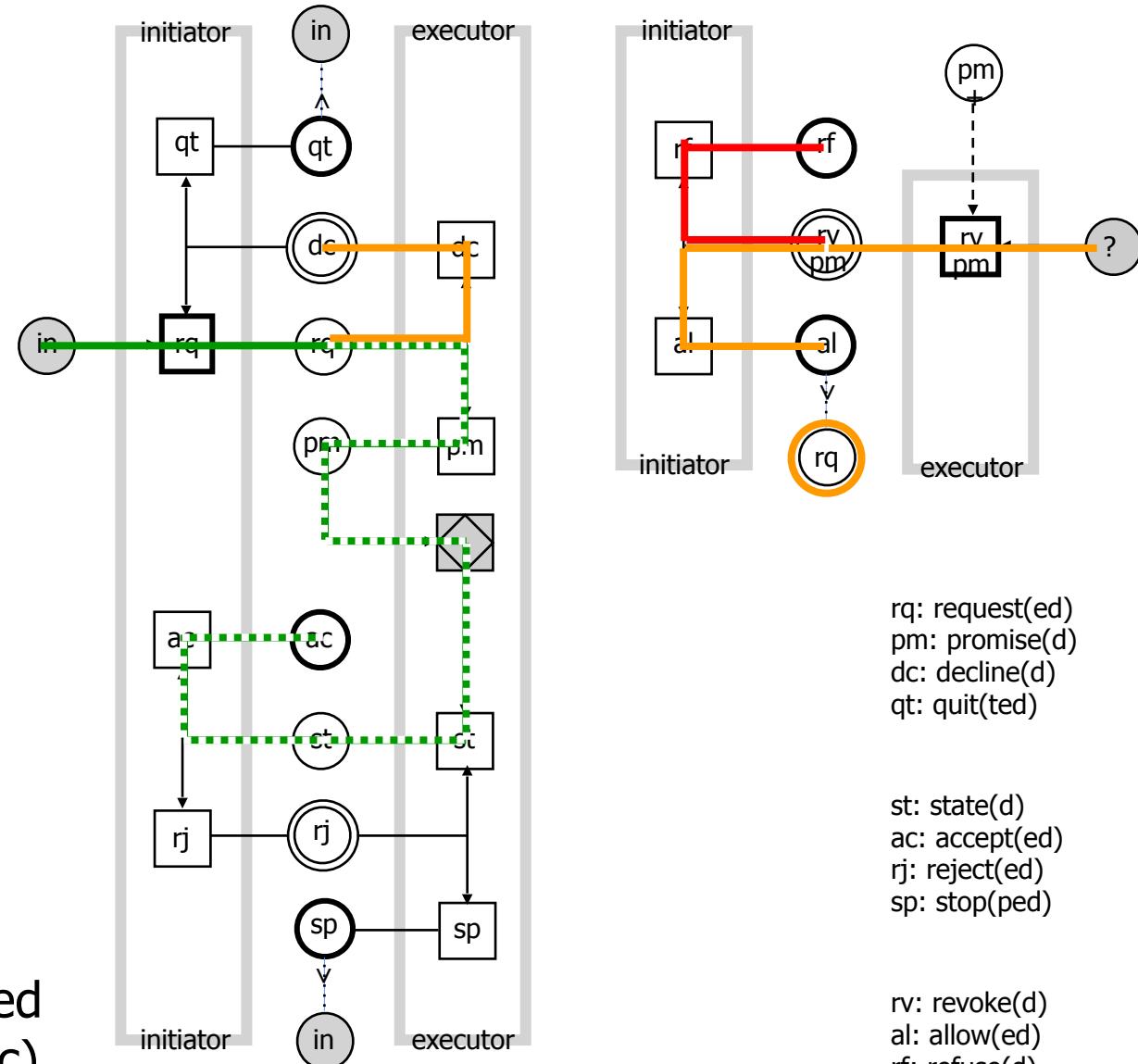


# **Revocation of the promise**

After having promised the customer to sell a bouquet of red tulips, the salesperson discovers that his colleague has sold the last one.

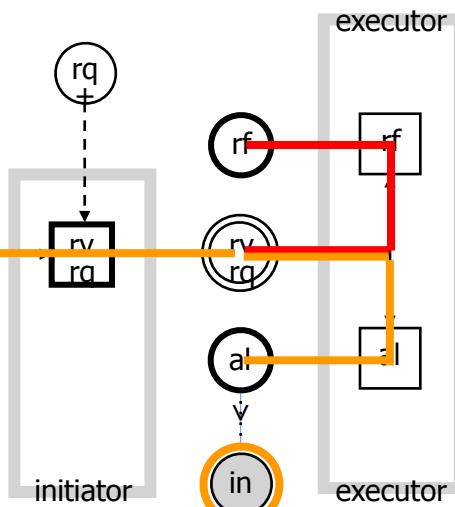
He makes his apologies to the customer, who cannot do otherwise than discussing an alternative solution in the status declined, in which they end up.

Note that a revoke promise can be performed even when the status (ac) is reached.





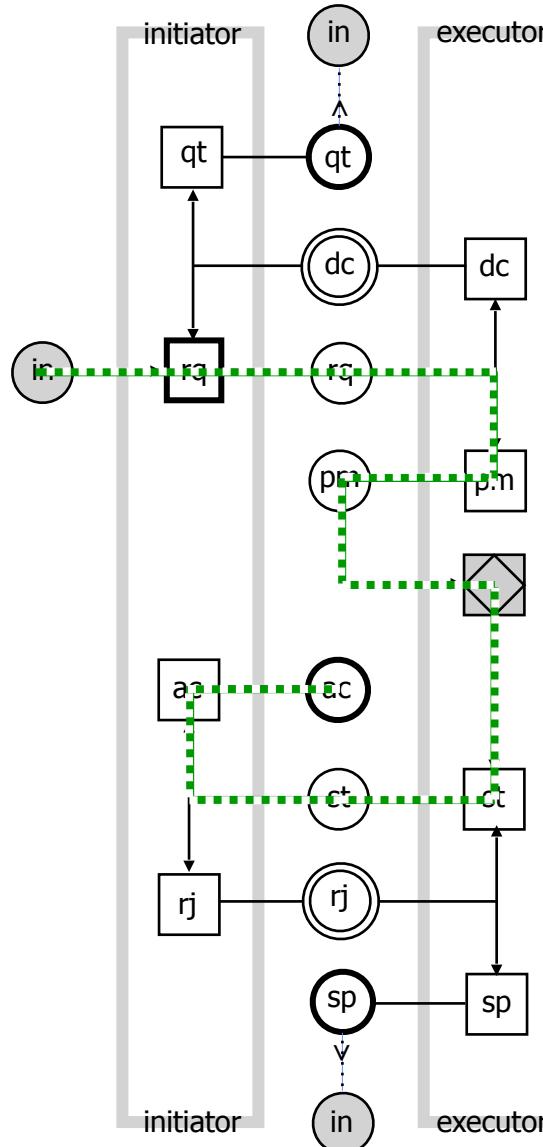
# Revocation of the request



rq: request(ed)  
pm: promise(d)  
dc: decline(d)  
qt: quit(ed)

st: state(d)  
ac: accept(ed)  
rj: reject(ed)  
sp: stop(ed)

rv: revoke(d)  
al: allow(ed)  
rf: refuse(d)



After having left the flower shop, the customer realises that she should have bought a bouquet of yellow roses.

She re-enters the shop and explains her mistake. The salesperson takes the sold bouquet back.

So, a revoke request can be performed from the status (ac). Can you give another example?



# What kind of C-act/fact is it? (3)

**request**

“I’d like to have such a bouquet of red tulips”

**promise**

“Ok, I will get one”

**P**

Linda decides to give Jan a bouquet of red tulips

**state**

“Here it is, your bouquet of red tulips”

**revoke rq, request**

“Oh, I think I’d rather have those yellow roses”

**(revoke rq)**

“You are quite capricious, aren’t you?”

**(revoke rq)**

“Come on, I only made a mistake”

**allow, promise**

“Alright, so a bouquet of yellow roses”

**reauest**

Jan sends an invoice to Linda

**reauest**

Linda receives the invoice

**decline**

But Linda says that the invoice is not correct

**request**

Steven asks Hans to sweep the path

**decline**

But Hans says that he won’t do it

**P**

Theo sweeps the path with a broom

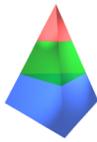
**P**

The broom is made by Edward

# EXAMPLE



## Case Fixit

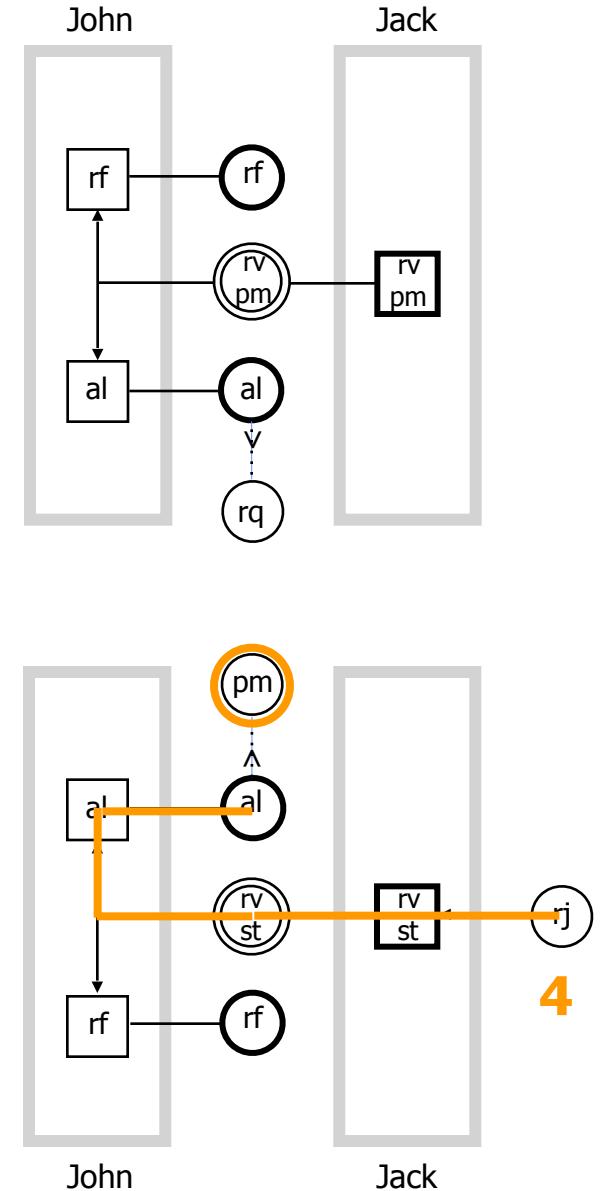
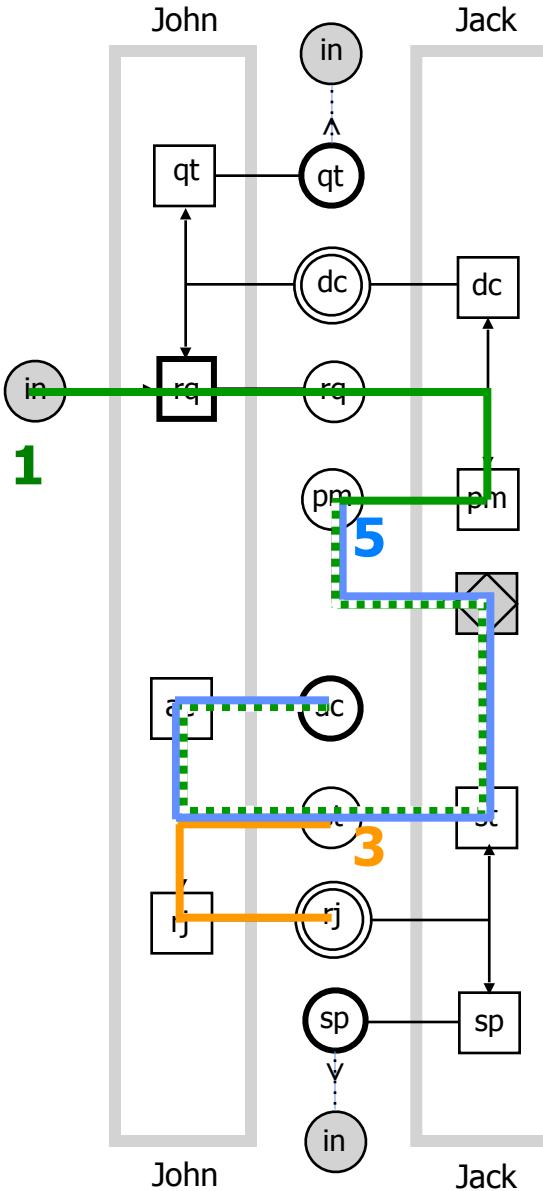
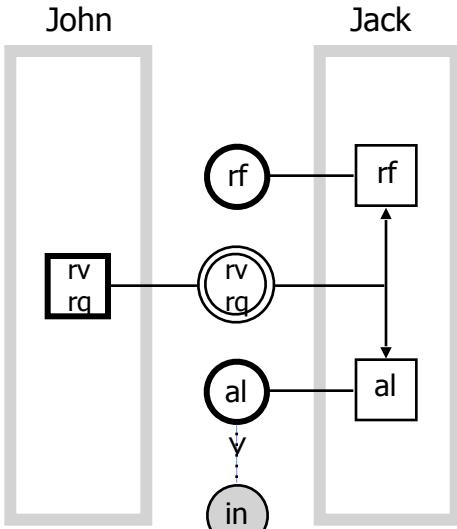


# RAC: analysis (4)

John Smith collects his car from the Fixit garage where he had brought the car the day before, because of a bumping noise he heard whenever he drove over a speed ramp. On the invoice he has to pay, he sees that the two rear shock absorbers are replaced. After having paid, John drives back home. At the first speed ramp, however, he hears the same bumping noise again! John returns to the garage and tells the boss (Jack) about what happened, and that he will not accept this outcome of the repair. Jack promises that he will have another look at the problem. At the end of the day, Jack calls John and tells him that they have found the real cause of the problem: two bolts being loose. They will fix it, and John can collect the car the next morning. When John presents himself again at the reception of Fixit the next morning, Jack says that there are no additional costs for the (second) repair. But that is not sufficient for John; he does not want to pay for the shock absorbers because that repair didn't solve the problem. After some discussion, they come to the agreement that John gets the shock absorbers for 60% of the price.



# Fixit: repair completion transaction (1)





# Fixit: repair completion transaction (2)

**Path 1.** In first instance, the standard pattern of the transaction is fully passed through. In the status promised, Jack (the executor) makes a diagnosis and a repair plan.

After the complete carrying through of the transaction, John (the initiator) regrets his accept, because the problem is not fixed.

**Path 2.** John revokes the accept act. In the rv-ac discussion, Jack decides to allow the revocation, by which the status in the standard pattern is turned back to stated.

**Path 3.** This enables John to reject the stated product.

**Path 4.** The only feasible option for Jack in the status rejected is to revoke his state act. John allows this, which turns the transaction status back to promised.

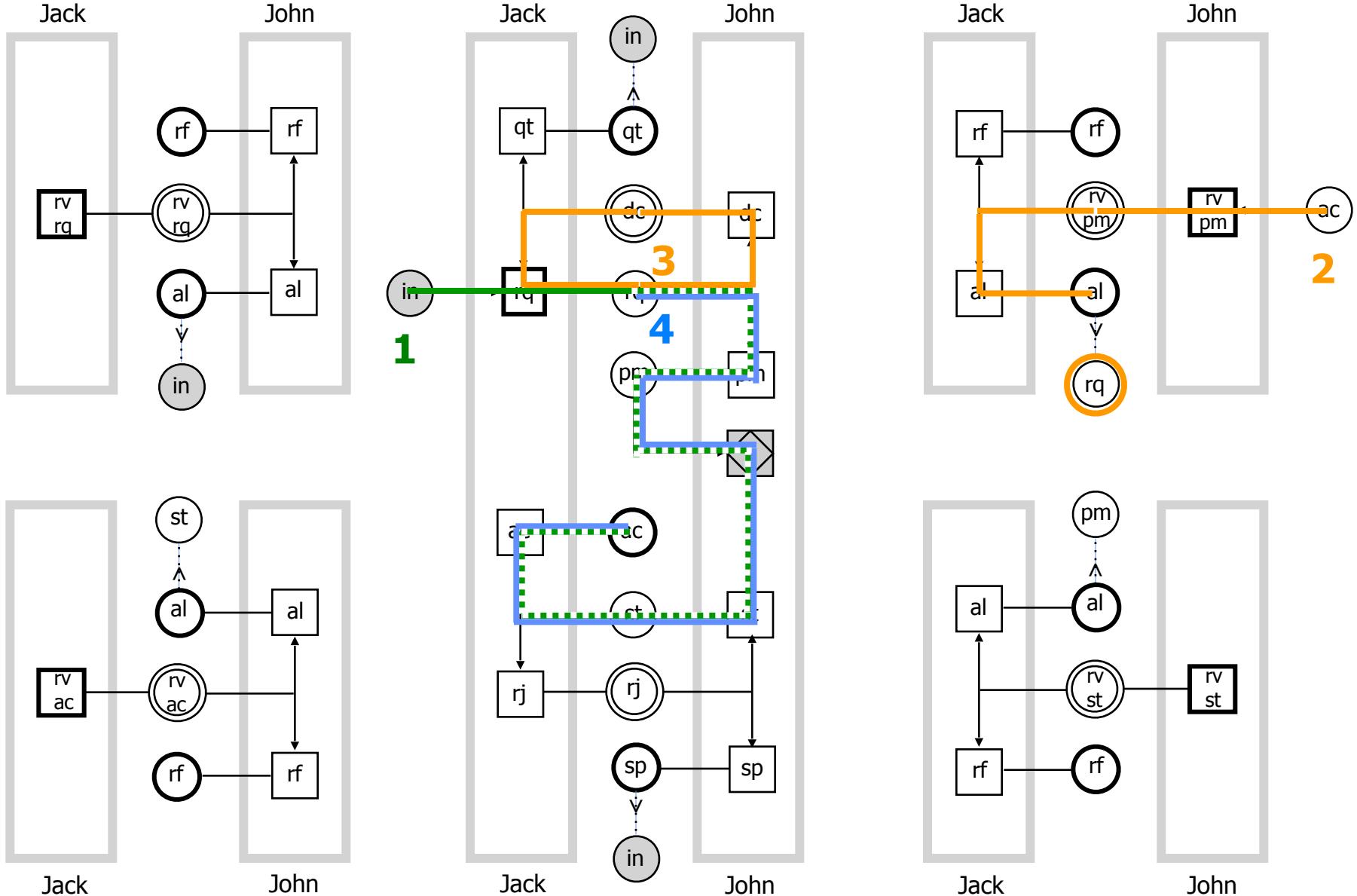
**Path 5.** From this status, Jack is able to make a new diagnosis and repair plan. Subsequently, he redoing the repair (P-act), and states the new product. Because the problem is now fixed, John accepts the new product.

NOTE. Redoing the P-act entails that the previous P-fact is made undone.

In this case, however, Jack does not take out the shock absorbers, but only fixes the two loose bolts.



# Fixit: repair payment transaction (1)





# Fixit: repair payment transaction (2)

**Path 1.** In first instance, the standard pattern of the transaction is fully passed through. After the second repair transaction, however, John (the executor) regrets his promise, because he thinks it is not fair that he has to pay the full amount of the shock absorbers.

**Path 2.** Consequently, John revokes his promise. In the rv-pm discussion, Jack decides to allow the revocation, by which the transaction status in the standard pattern is turned back to requested.

**Path 3.** This enables John to decline the (original) request by Jack for the full costs of the new shock absorbers. In the decline discussion they agree on a new amount to pay, which is then requested by Jack.

**Path 4.** John promises to pay the new amount, pays it (P-act), and subsequently states the new product. Because this is what they agreed upon, John accepts the new product.

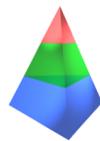
NOTE. Redoing the P-act entails that the previous P-fact is made undone.

In this case, the amount that John has paid in first instance must somehow be reimbursed. This is a matter of implementation, also depending on the way in which the money is transferred (by cash or by bank transfer, etc.)

# Way of Thinking

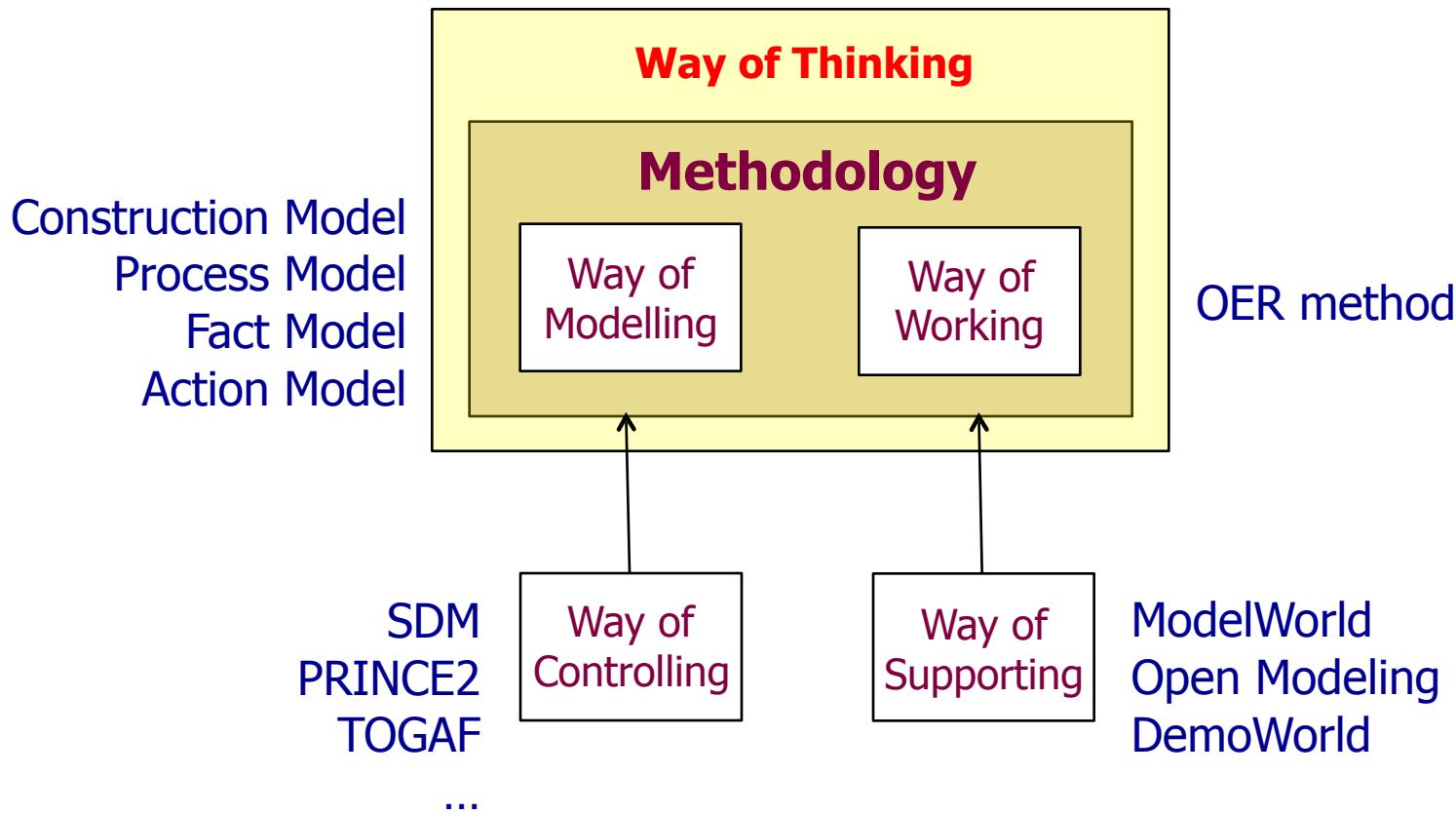


## The OMEGA theory



# DEMO in the Five Ways Framework

MU, TAO, FI; DELTA, PSI, **OMEGA**; ALPHA, BETA; SIGMA



adapted from P.S. Seligmann, G.M. Weijers, H.G. Sol: analyzing the structure of IS methodologies – an alternative approach, 1989



# The OMEGA theory

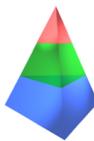
The OMEGA theory (standing for “Organisational Modules Emerging from General Arrangements”) or organisational construction theory, is a theory about the **construction** of **organisations**.

An organisation is a system in the category of social systems (of which the elements are social individuals, thus subjects).

The **construction** of an organisation consists of **tree structures** of actor roles and transaction kinds.

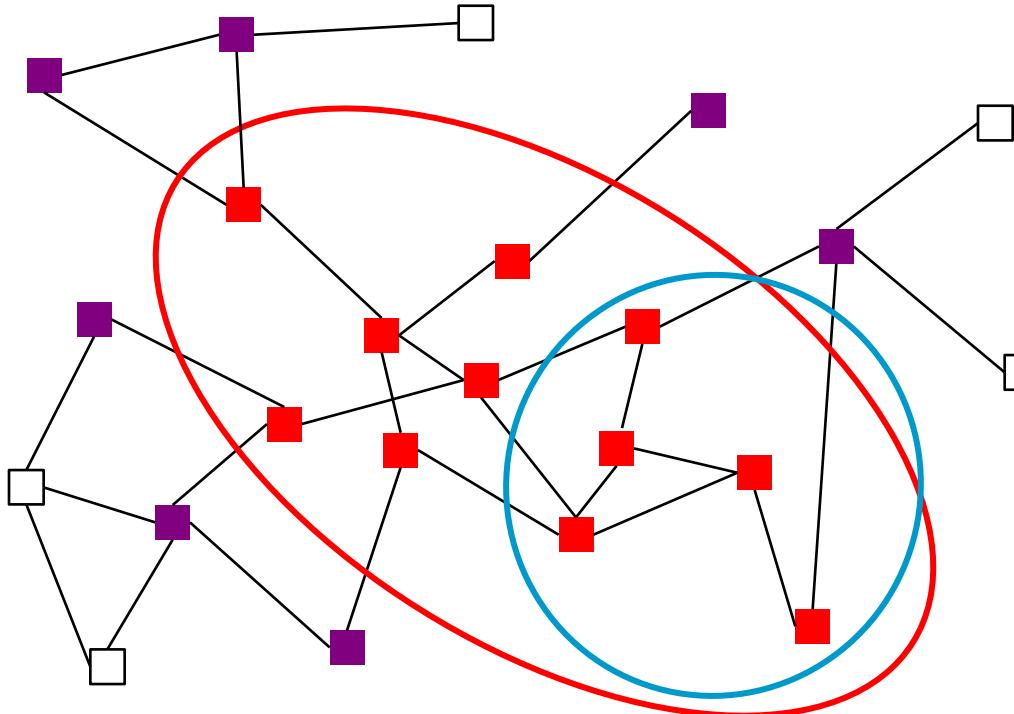
An actor role represents the authority to be executor in transactions of a certain kind. Actor roles can be assigned to subjects.

A transaction kind represents the existence of transactions of a certain kind, resulting in the creation of products of a certain kind.



# The construction of a system

**construction = composition + environment + structure**

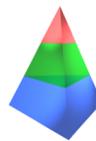


The red closed curve represents the border of the system.  
The blue closed curve represents the border of one of its subsystems.

■ composition element

■ environment element

□ external element



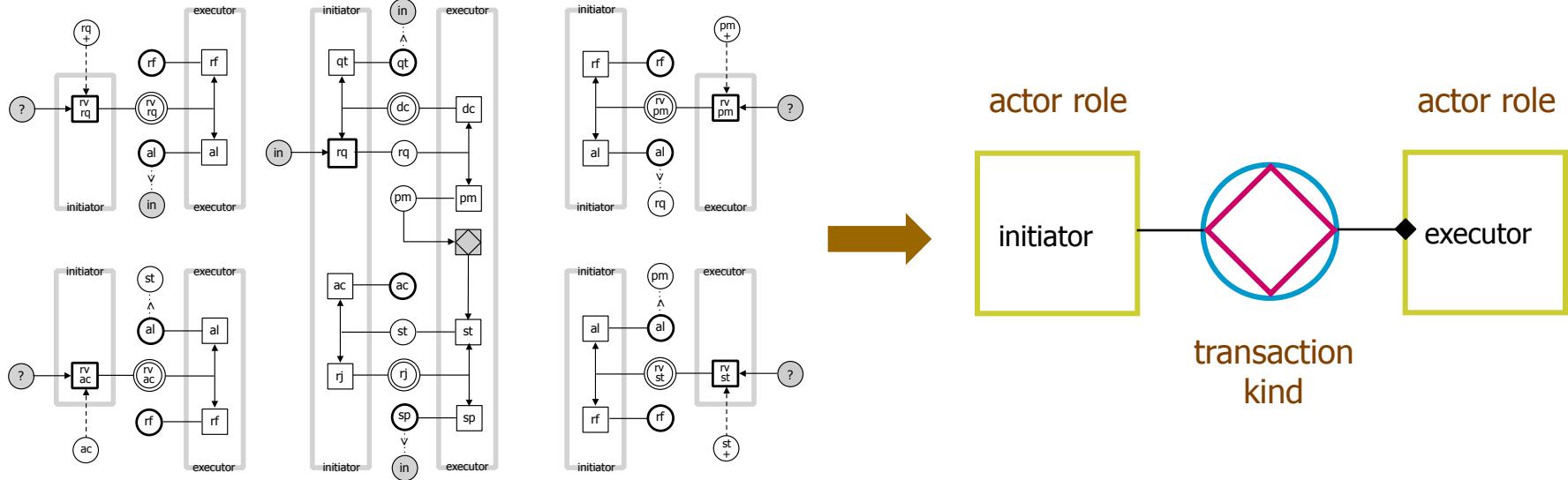
# The operation of a system

The operation of a system is the manifestation of its construction in the course of time: the sequences of events, caused by the system.





# The organisational building block (1)

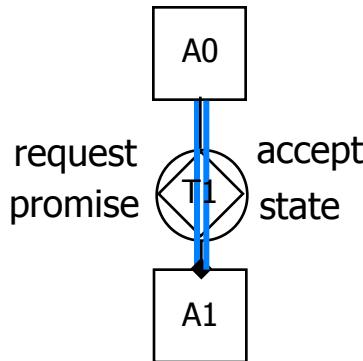


Transactions are the **molecular** components of an organisation. Only when a **transaction** is successfully completed, a **P-event** occurs, i.e. a new **P-fact** start to exist.

An **actor role** is defined as the **authority** to be the **executor** in transactions of exactly one transaction kind. It includes the authority to be the initiator in transactions of the enclosed transaction kinds.



# The organisational building block (2)

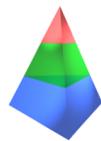


A **transaction kind** together with its **executor role** constitutes the **organisational building block**

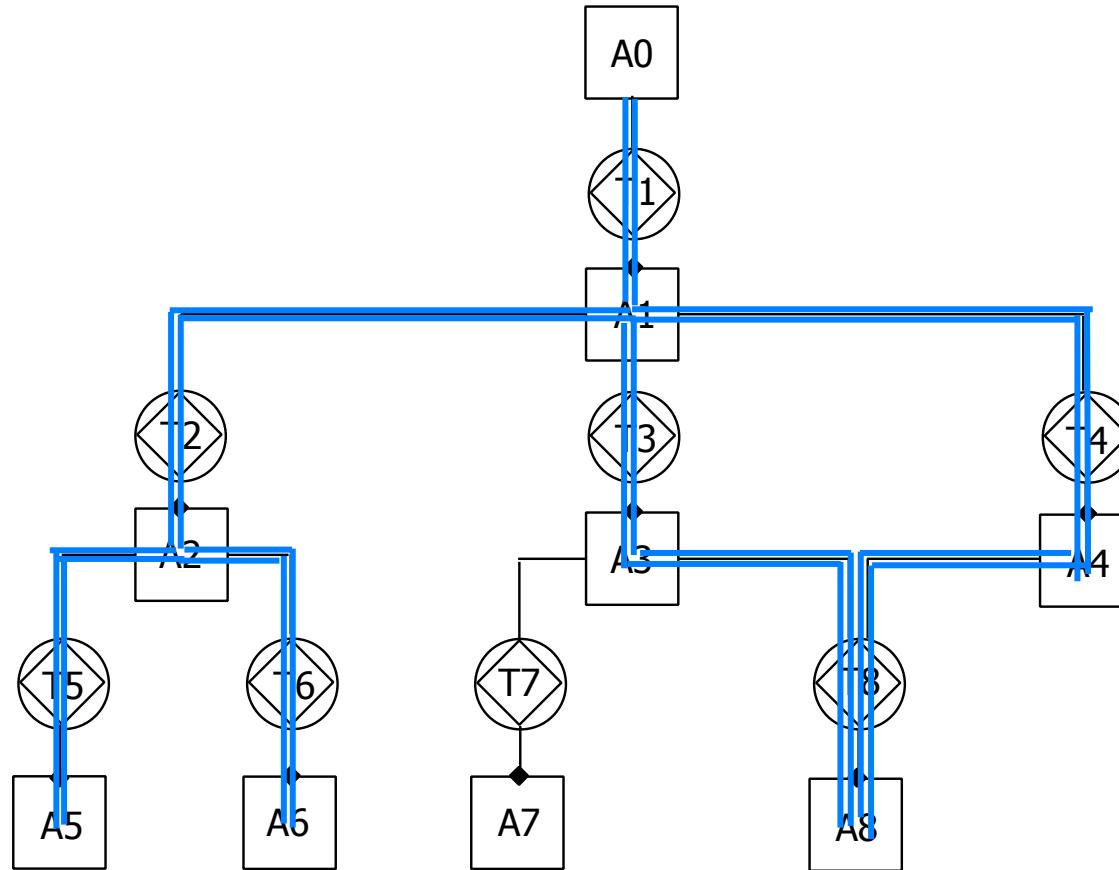
A0 (the initiator of T1) comes to agreement with A1 (the executor of T1) about the delivery of a product P1. By promising the P1, A1 takes the responsibility to produce it.

A0 does not care and need not care about what A1 has to do in order to bring about P1.

At some point in time, A1 addresses A0 and performs the state act in the transaction T1.

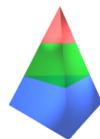


# Business process structures

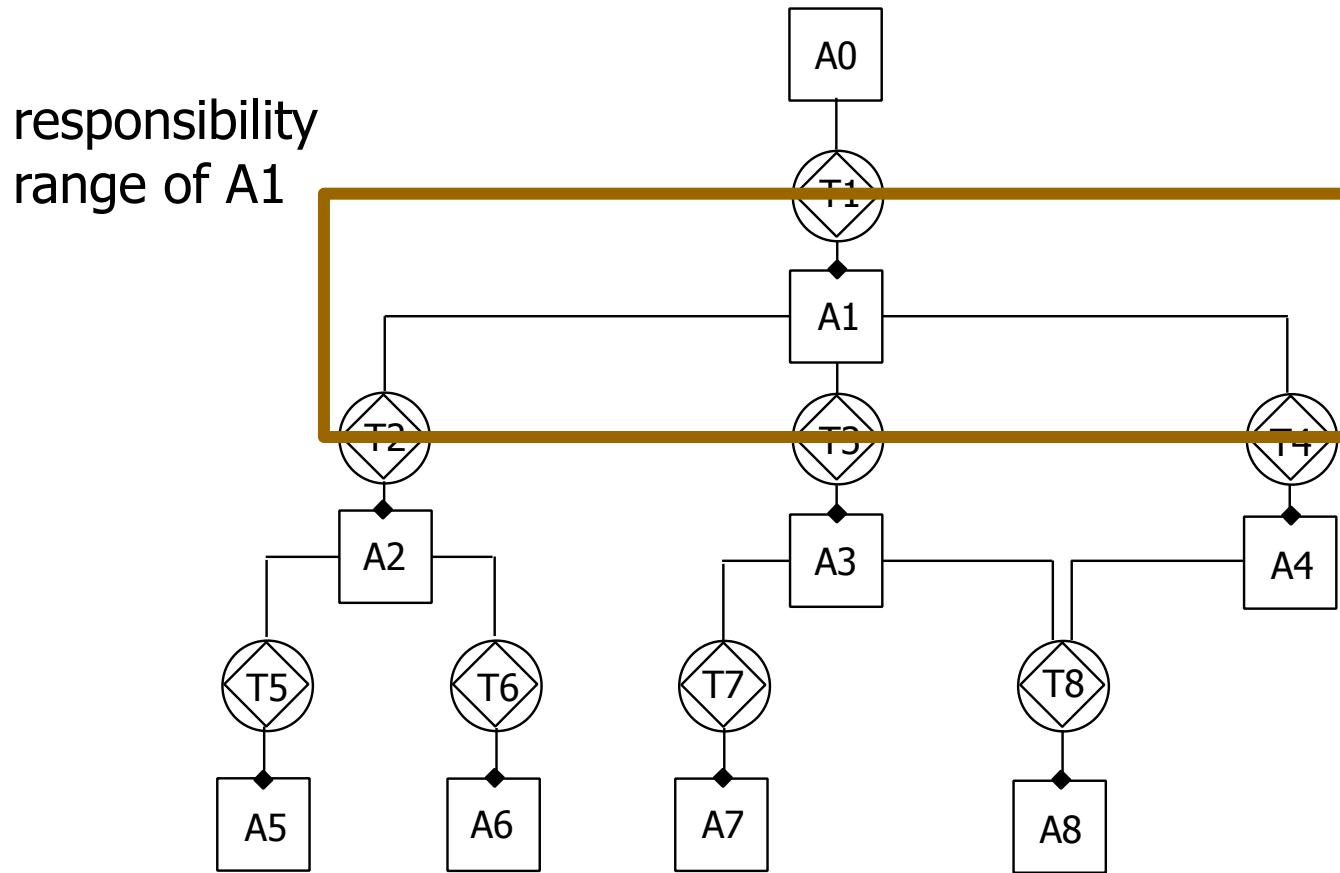


the **business process structure** corresponds to the **product structure**

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



# The responsibility range of actor roles

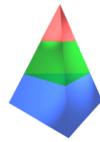


A1 is not only responsible for its executor role in T1 but also for its initiator role in T2, T3 and T4! Therefore, A1 could rightly be called the **process owner** of the trees of which T2, T3 and T4 are the top.

# OMEGA theory



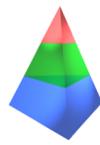
## reference models



# Reference models

There appear to be general arrangements in DEMO models that give rise to distinguish a number of **product categories** and to corresponding **reference models**.

	<b>tangible things</b>	<b>intangible things</b>
<b>creating things</b>	manufacturing (movable and immovable) goods	giving advices, passing judgments, etc.
<b>transporting and storing</b>	transporting and storing goods and data files	< not applicable >
<b>transferring ownership</b>	buying/selling goods	acquiring/lending rights, licences, etc.
<b>granting right of usufruct</b>	hiring/renting goods	acquiring/lending rights, licences, etc.



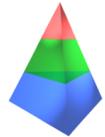
# Creating things

## Tangible

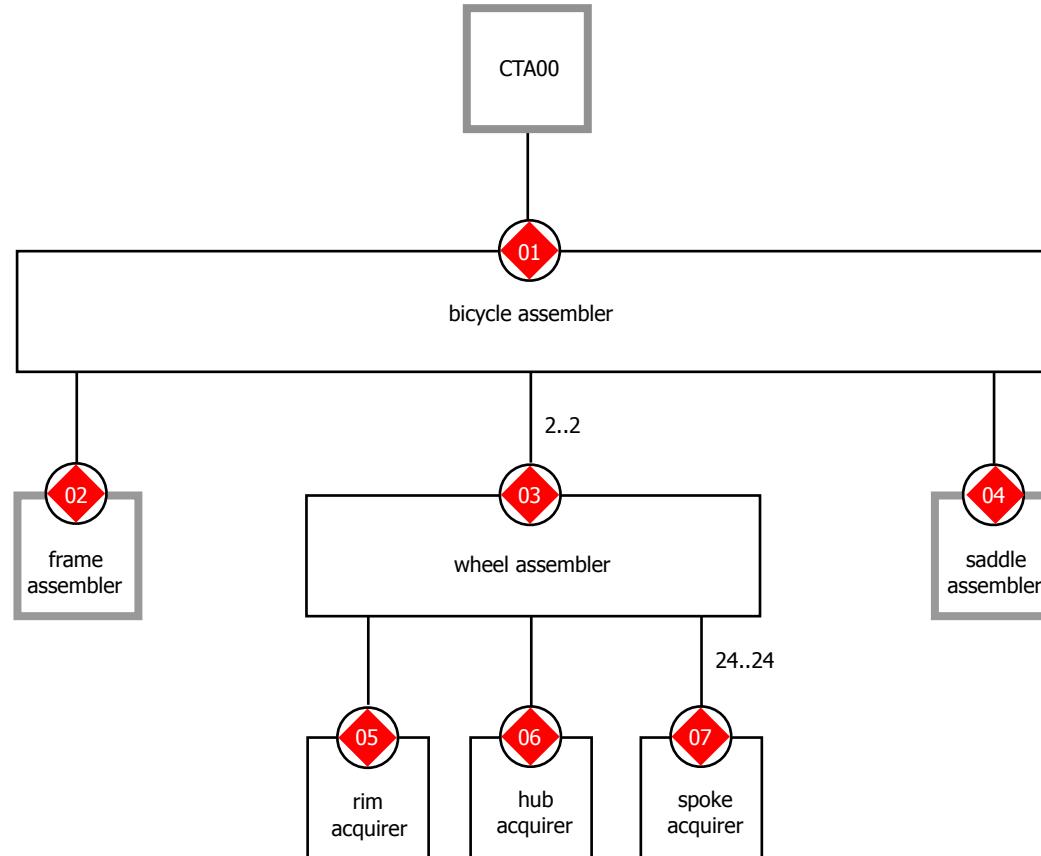
All kinds of goods, like bicycles and houses

## Intangible

Advices, judgments, contracts.



# Example of creating goods





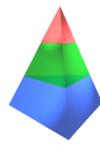
# Transporting and storing

## Tangible

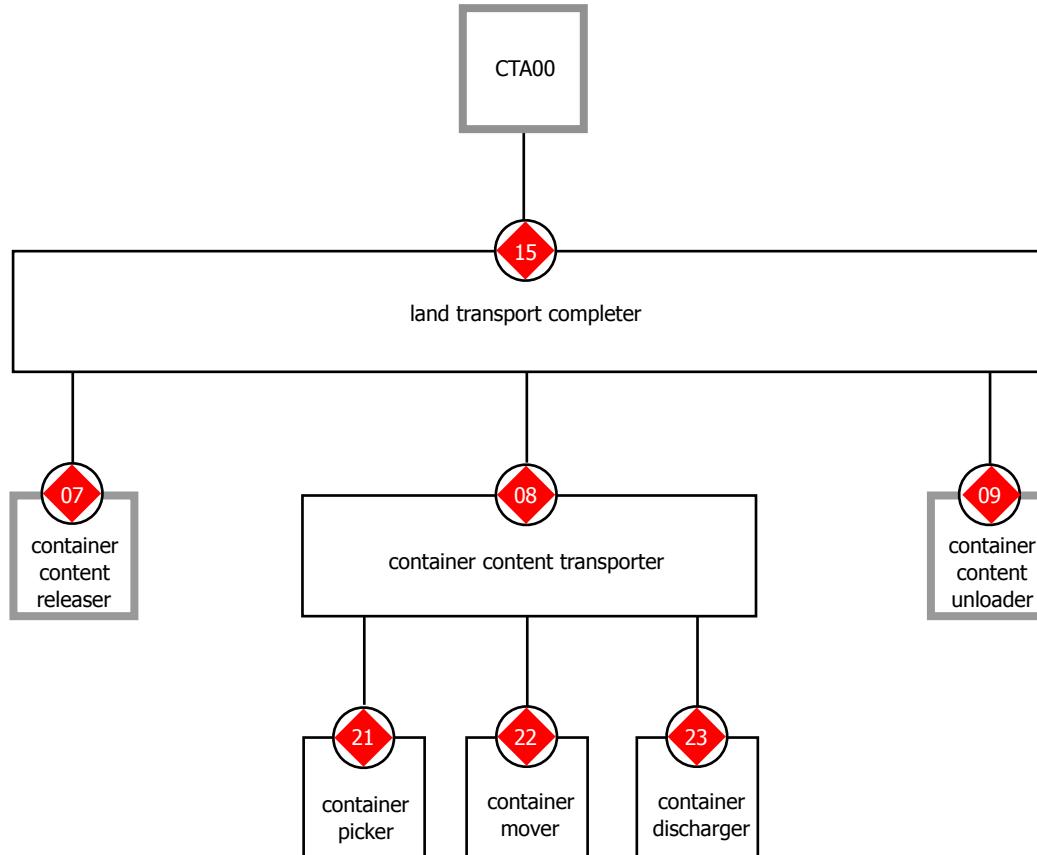
Goods, like bicycles and domestic appliances, but also files

## Intangible

< not applicable >



# Example of transporting goods





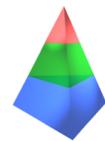
# Transferring ownership

## Tangible

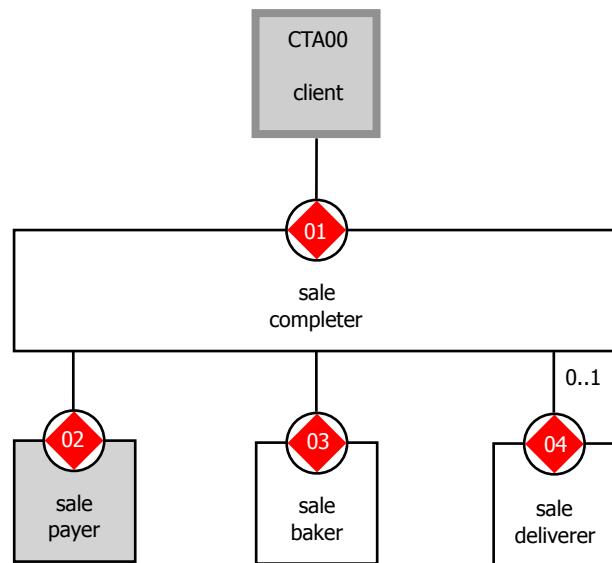
All kinds of goods, like bicycles, houses and pizzas

## Intangible

Several kinds of contracts, like insurance policies, ownership claims.



# Example of transferring ownership





# Granting right of usufruct

## Tangible

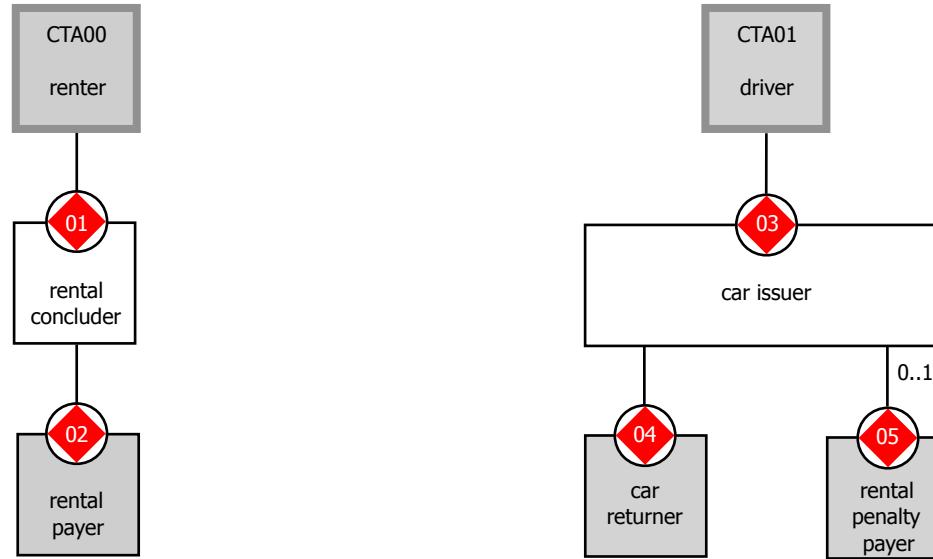
Cars, houses, hotel rooms, and theatre chairs

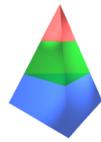
## Intangible

Insurances, memberships, subscriptions



# Example of granting right of usufruct



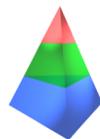


**For over your bed ...**

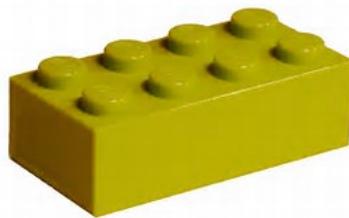
# Vergeet de stromen

# Denk in bomen

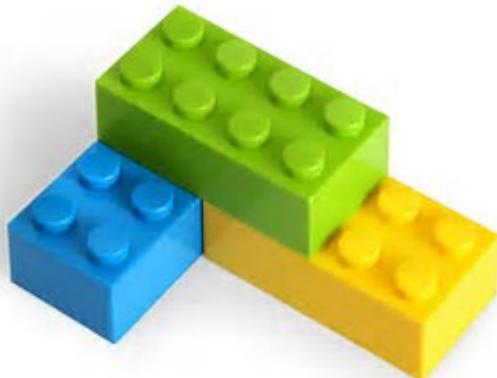
Forget the flows  
Think in trees



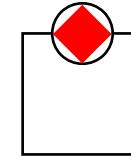
# In summary ...



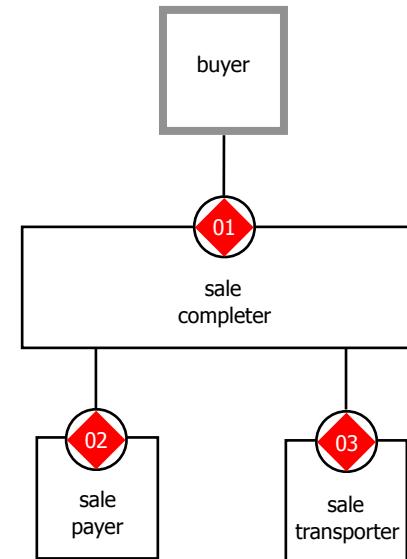
**LEGO brick:** studs at the top  
and holes at the bottom



**LEGO construction**



**DEMO brick:** a transaction kind (top)  
and its executing actor role (bottom)



**DEMO construction**

# Way of Thinking

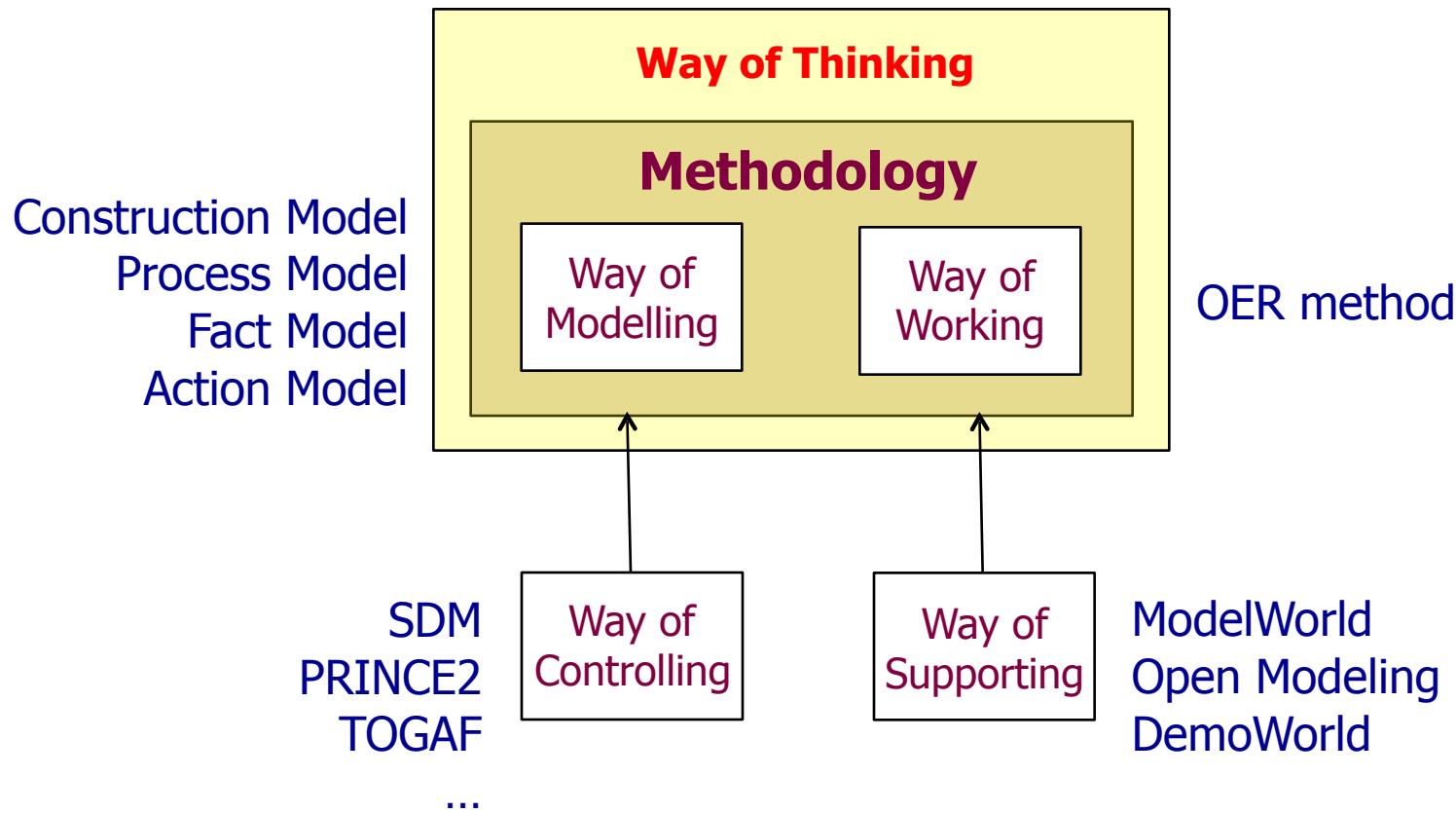


# The ALPHA theory



# DEMO in the Five Ways Framework

MU, TAO, FI; DELTA, PSI, OMEGA; **ALPHA**, BETA; SIGMA



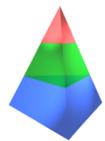
adapted from P.S. Seligmann, G.M. Weijers, H.G. Sol: analyzing the structure of IS methodologies – an alternative approach, 1989



# The ALPHA theory

The ALPHA theory (standing for “Abstraction Layers in Production for Holistic Analysis”) or organisational essence theory, is a theory about the **essence** of an **enterprise**, This essence is revealed in two steps:

1. By abstracting from the **realisation** of the enterprise, yielding its O-organisation
2. By abstracting from the **implementation** of the O-organisation



# ALPHA theory: realisation (1)

The three human abilities (performa, informa, and forma) that we distinguish in coordination, also apply to production:

## **Performa:**

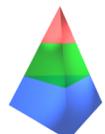
The ability to perform **original** production acts, such as to create (manufacture, transport, observe), devise, decide, judge.

## **Informa:**

The ability to perform **informational** production acts, such as to remember, recall, compute facts

## **Forma:**

The ability to perform **documental** production acts, such as to archive, provide, transform documents (or data) and to store, retrieve, transmit, copy, destroy files



# ALPHA theory: realisation (2)

Accordingly, three layers of actors or organisational layers can be distinguished in every enterprise:

the **O-organisation** (O from original)

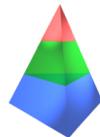
the **I-organisation** (I from informational)

the **D-organisation** (D from documental)

By the **realisation** of an enterprise is understood (the devising of)

its I-organisation and  
its D-organisation

starting from the O-organisation



# The organisational layers

IMMATERIAL  
PRODUCTION

devising  
deciding  
judging

remembering  
recalling  
computing

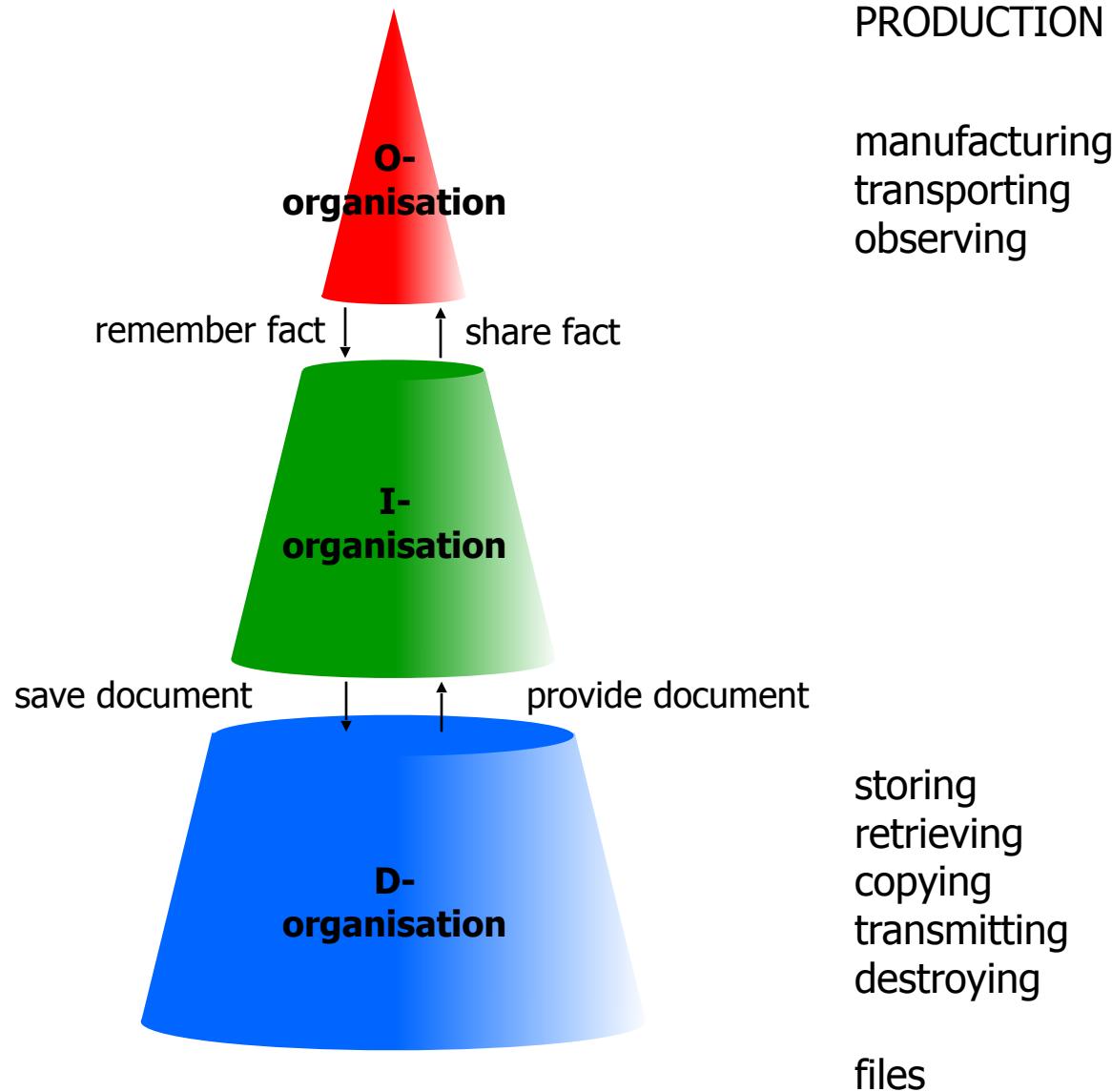
facts

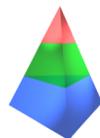
saving  
providing  
transforming  
documents

MATERIAL  
PRODUCTION

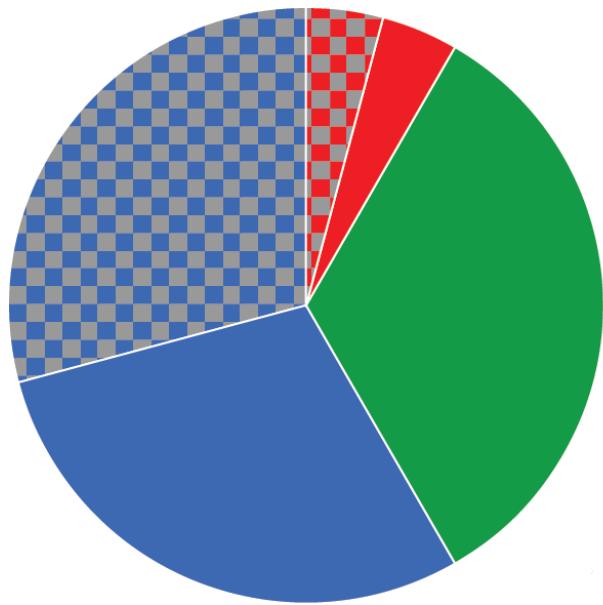
manufacturing  
transporting  
observing

storing  
retrieving  
copying  
transmitting  
destroying





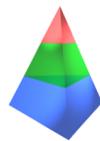
# Organisational layers and sorts of products



- O-organisation: material production
- O-organisation: immaterial production
- I-organisation
- D-organisation: file handling
- D-organisation: document handling

If the (sign) content of documents is disregarded, only the (physical) file aspect remains. But then, there is no distinction between file handling (in the D-organisation) and material production (in the O-organisation).

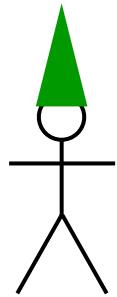
As an illustration, the postal office service in most countries doesn't care anymore whether the envelop or package contains information or not.



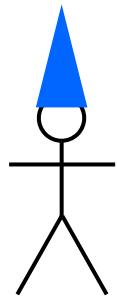
# Organisational layers and sorts of actors



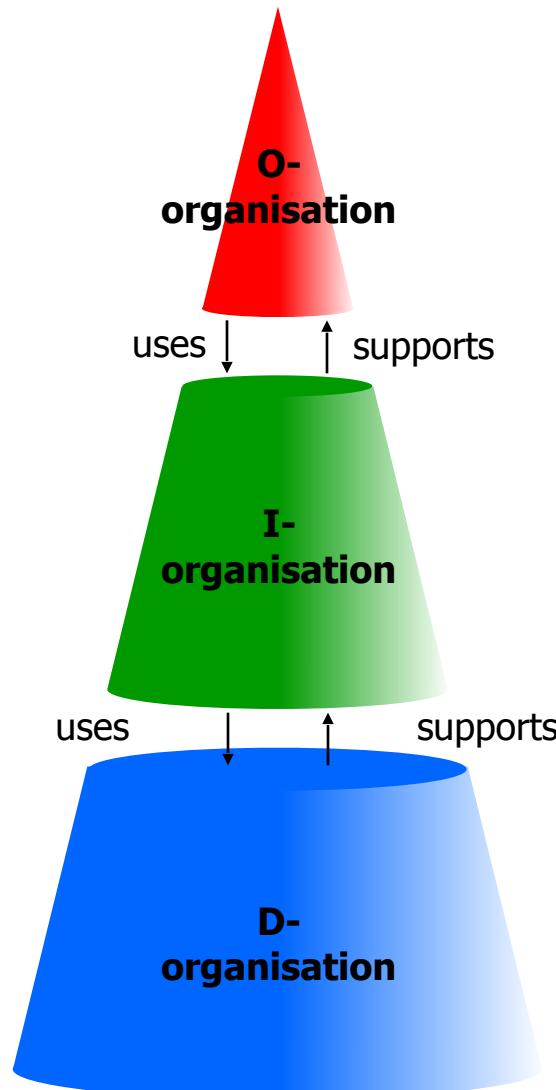
red actor



green actor



blue actor



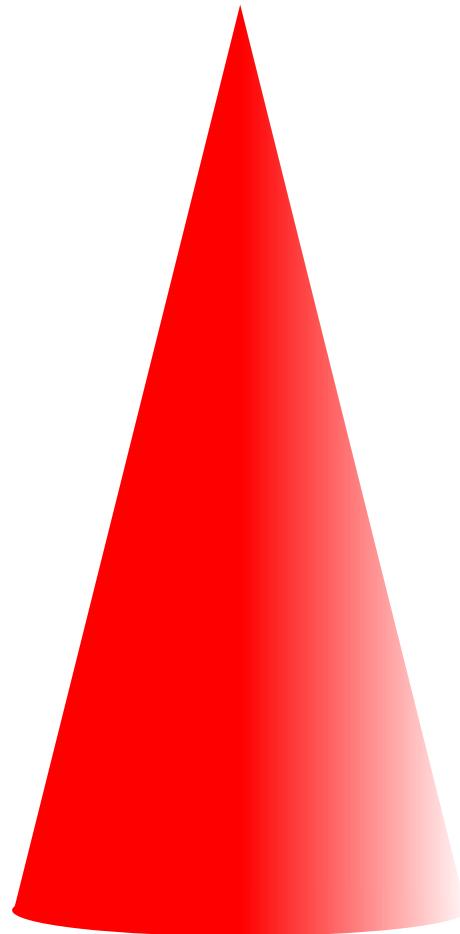
**Performa**

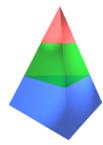
**Informa**

**Forma**

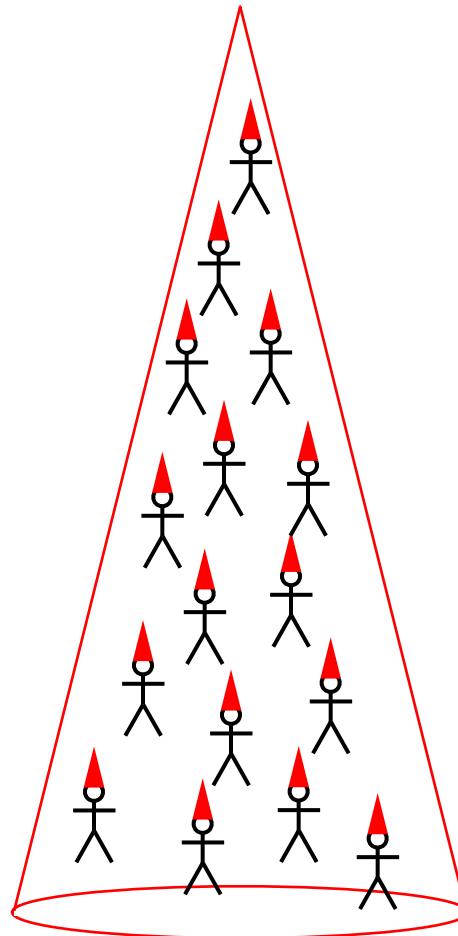


# Actors in the O-organisation



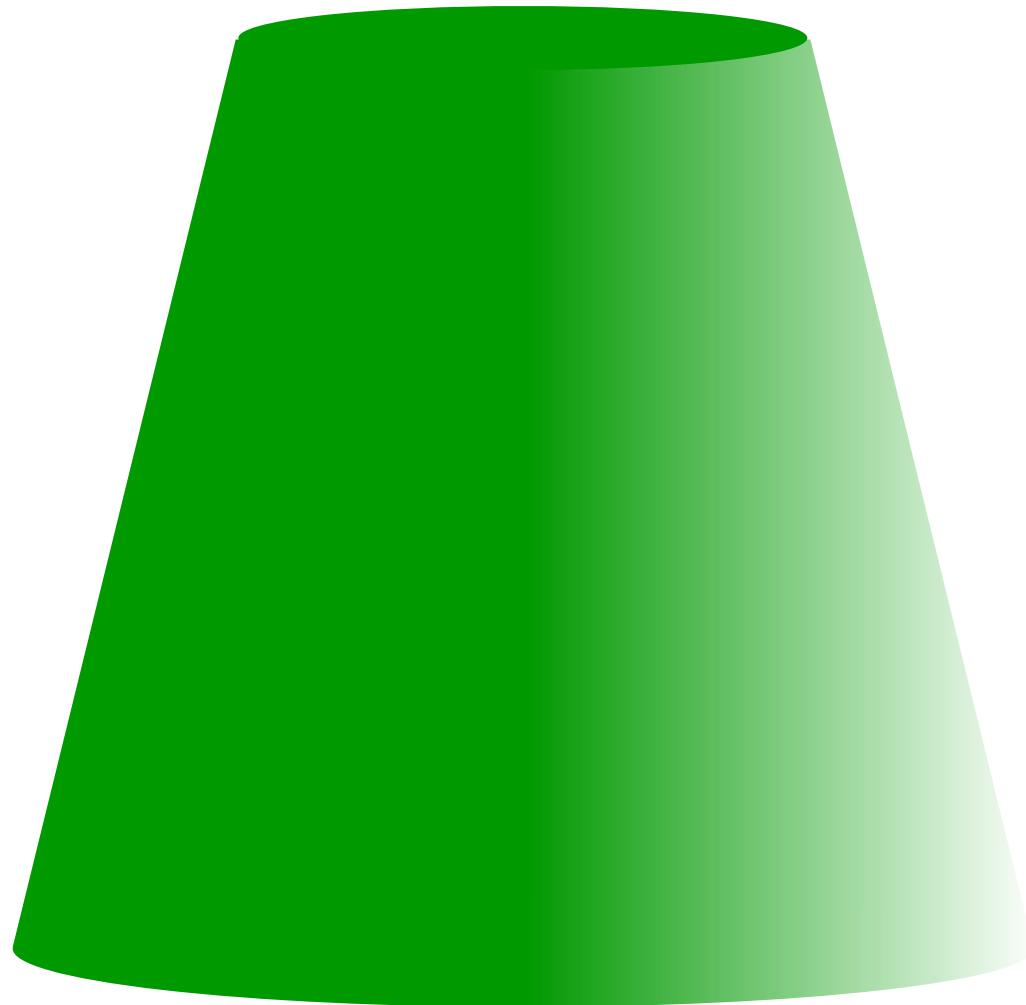


# Actors in the O-organisation



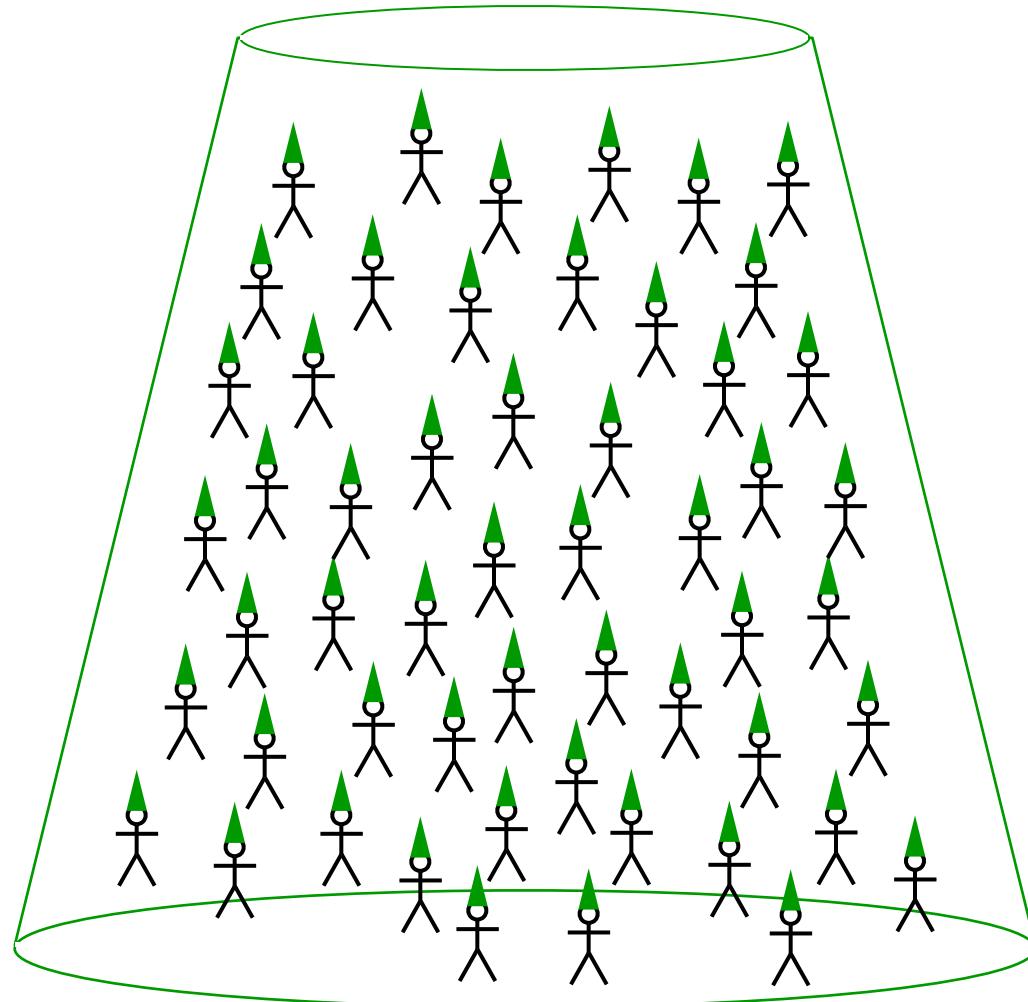


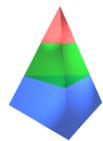
# Actors in the I-organisation



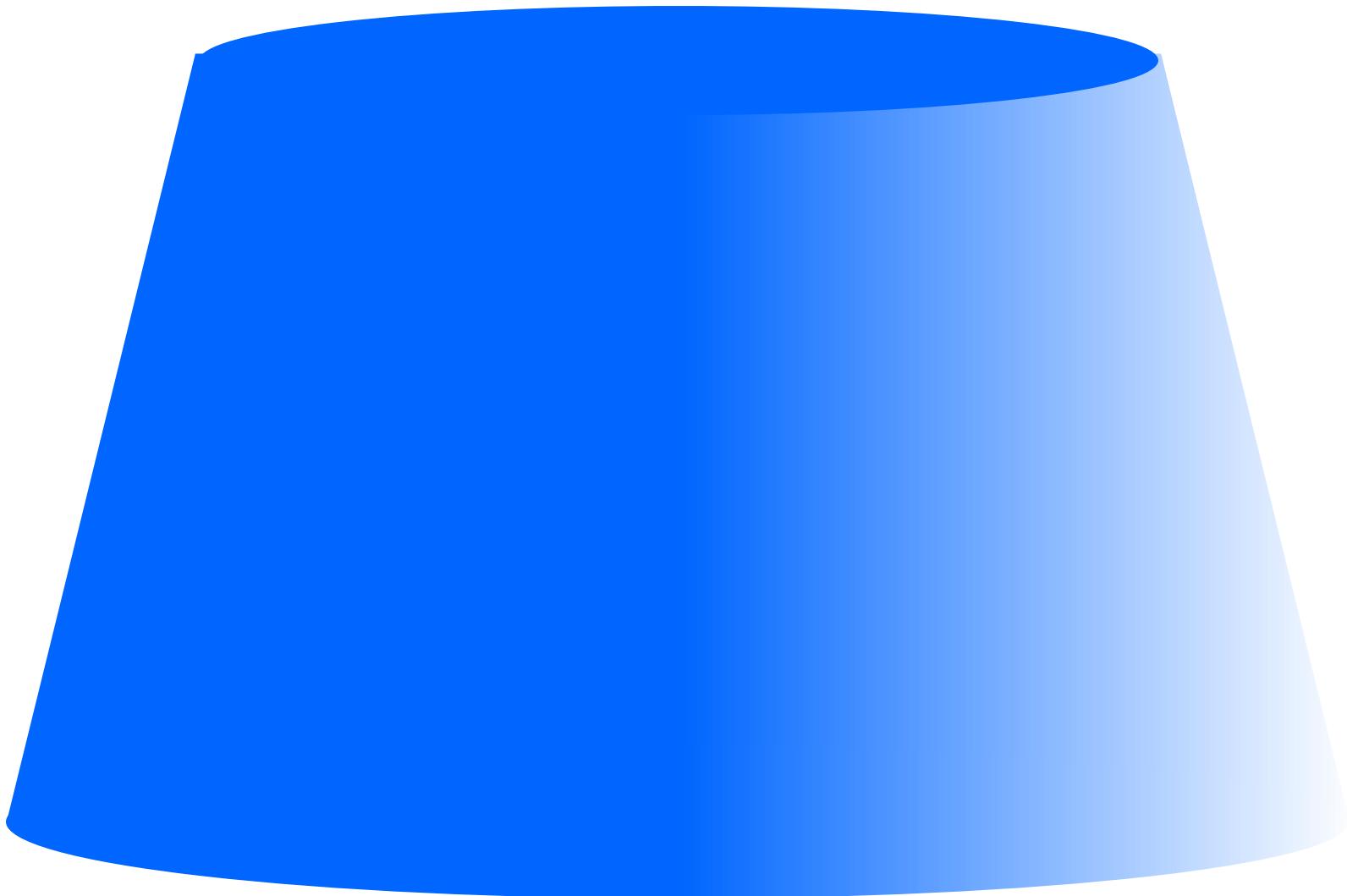


# Actors in the I-organisation



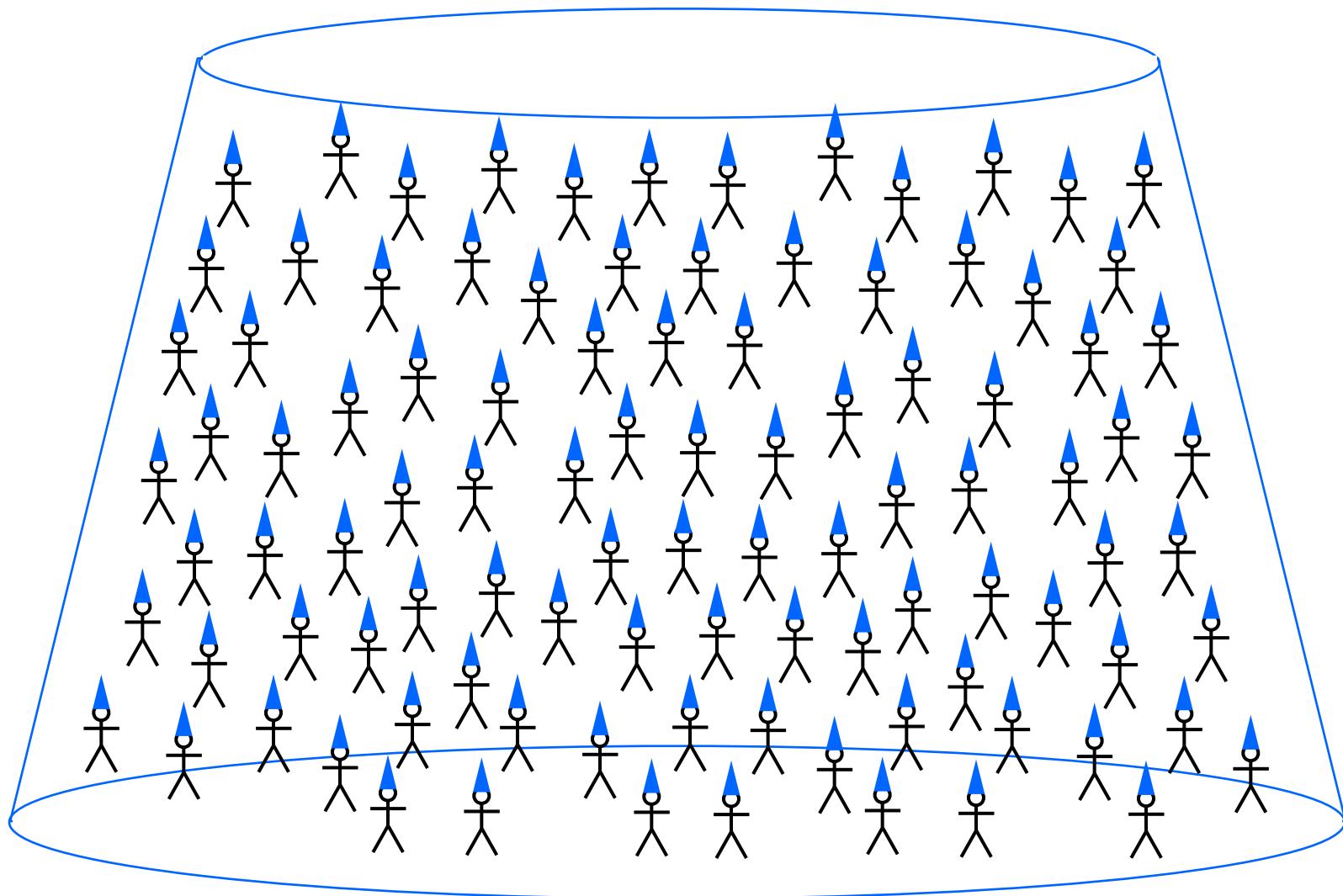


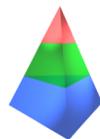
# Actors in the D-organisation





# Actors in the D-organisation





# ALPHA theory: implementation

By the **implementation** of an organisation, is understood the allocation of appropriate 'technological' means to its ontological parts.

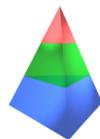
Regarding **actor roles**, it is their assignment to subjects.

Regarding **coordination** (C-acts), it is the allocation of appropriate means in order to perform them. Examples: face to face speaking and hearing, sending letters by postal mail, by telephone, by e-mailing, etc.

Regarding **production** (P-acts), a distinction must be made between material and immaterial production.

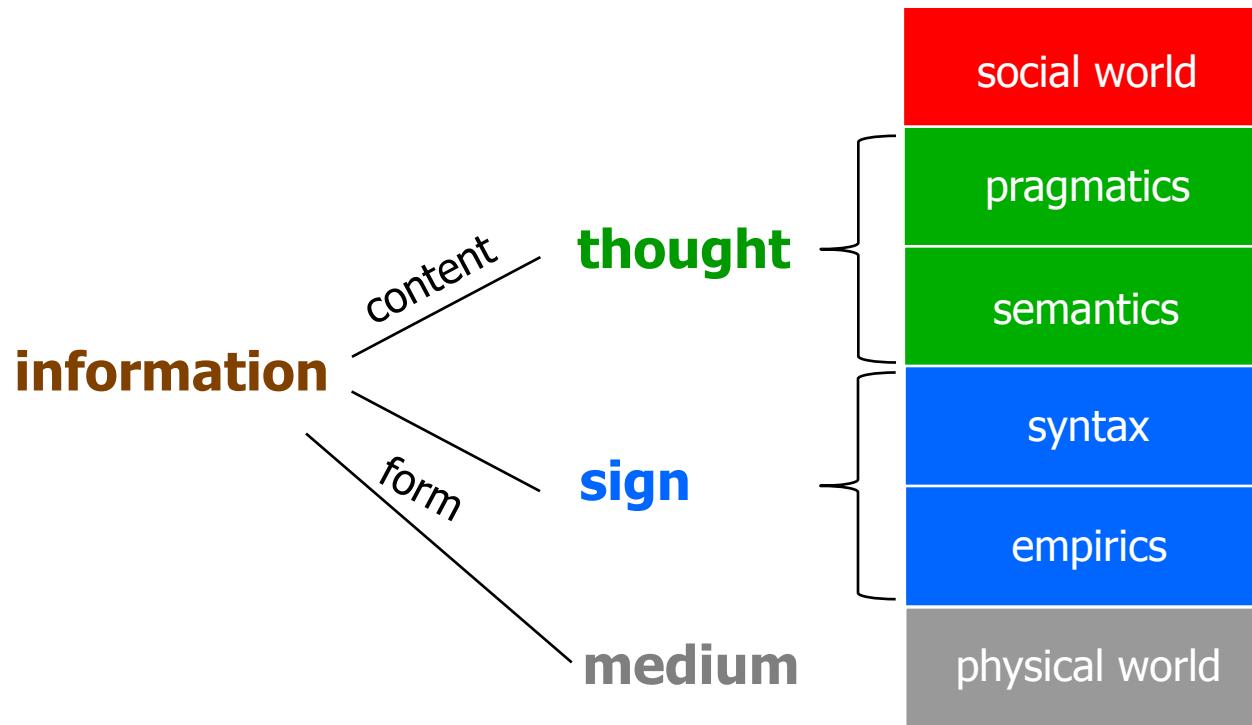
By the implementation of material P-acts (manufacturing, transporting, etc.), is understood the allocation of production means, like subjects, machines, tools, and work space.

Regarding immaterial P-acts (devising, deciding, judging), there are no implementation means, except for the subjects that perform them.



# The implementation of coordination

By the **implementation** of coordination acts is understood (the devising of) the **medium level**, thus the technological means by which the communication between actors takes place.





# Example of implementation (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

employee issues banknotes

Here you are, sir

Thank you

*promise*

*state  
accept*



# Example of implementation (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*

# PSI theory

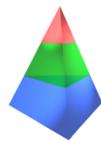


## Elaborations



# Verbal, non-verbal and tacit C-acts

- A coordination act is called **non-verbal** if some other act than a language act counts as performing the C-act. For example: handing over a bought item by a salesperson to you (as the customer) may count as performing the state act. Both verbal and non-verbal C-acts are explicit acts.
- A coordination act is called **tacit** if there is no act at all that counts as performing the C-act. Still the act is performed, but implicitly! For example: if you get, by postal mail, the membership card from a club you have become member of, you don't perform an explicit (verbal or non-verbal) accept act.



# Examples in the flower shop

P1: Hello, I'd like to have a bouquet of red tulips

**P1 : request :**    **P2 : sale 1618 is completed**

P2: **nodding**

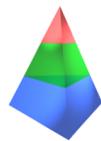
**P2 : promise :**    **P1 : sale 1618 is completed**

P2: **handing over the bouquet**

**P2 : state :**        **P1 : sale 1618 is completed**

P1: **doing nothing (tacitly accepting)**

**P1 : accept :**        **P2 : sale 1618 is completed**



# Time aspects in C- and P-acts/facts (1)

Every C-fact has a **creation time**, denoted by ct. It is the point in time at which the act is performed and thus the fact is created.

In addition, every C-fact has an **event time**, denoted by et. It is the time at which the fact starts to exist. It always holds that et > ct.

In (deterministic) technical systems, the difference between ct and et is a delay that is conveniently used for synchronisation.

In (non-deterministic) social systems, et can be used as the (intended) **settlement time** of the C-fact.

The addition “intended” is needed because social actors always act autonomously. Consequently, they may deal with the C-fact at an earlier or later point in time.

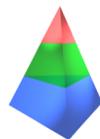


# Time aspects in C- and P-acts/facts (2)

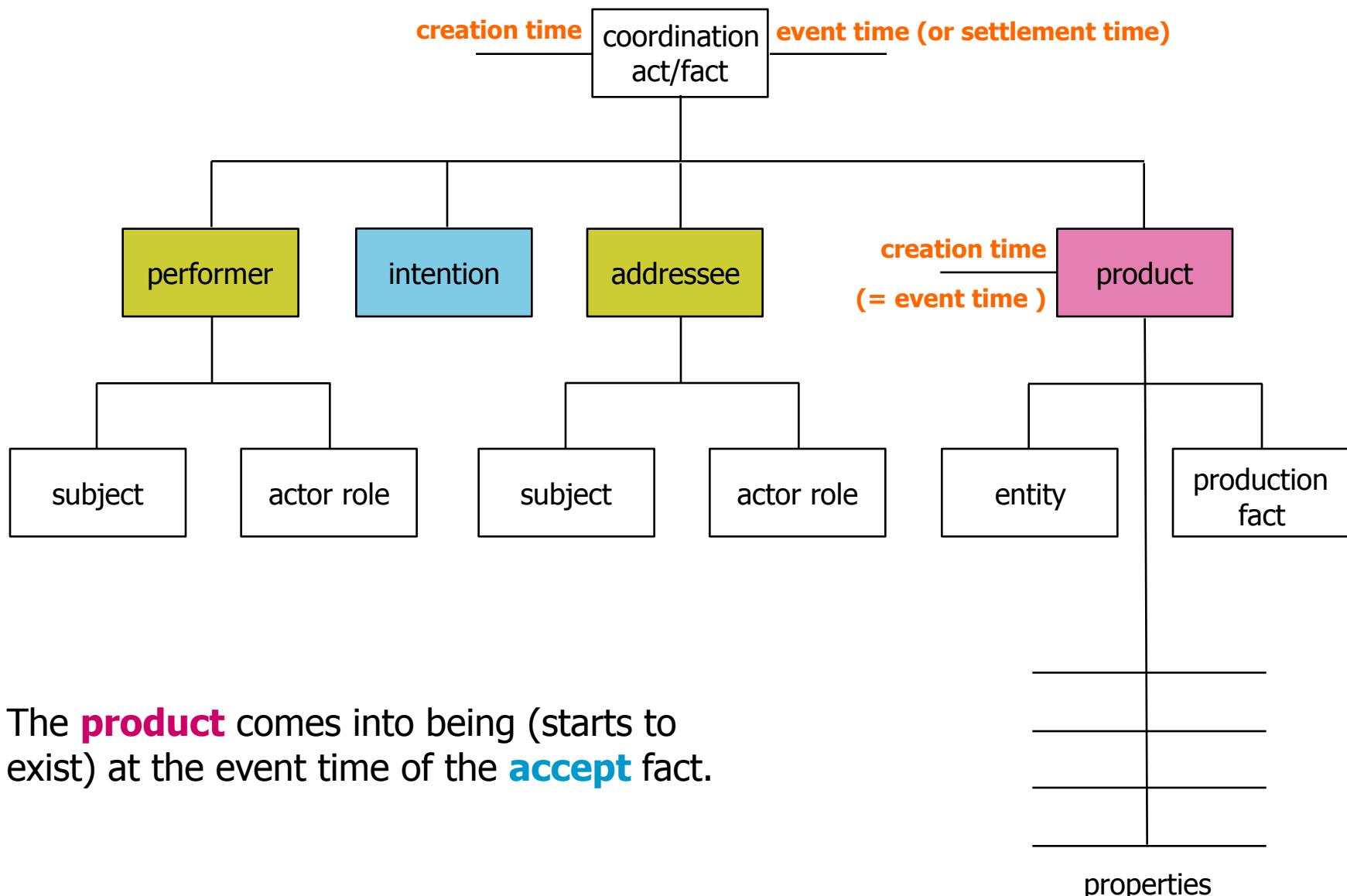
Every P-fact has an **event time**, denoted by et. It is the point in time at which the fact is created and comes into being (starts to exist). As we know from the PSI theory, this point in time is equal to the event time of the accept fact in the corresponding transaction. If the accept act is performed tacitly, then the event time of the state fact is taken.

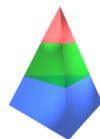
NOTE. As one of the properties of a product, a point in time may be added at which the product becomes effective. For example, the starting day of a membership, of a magazine subscription, and of a hotel stay.

This point in time may coincide with the event time of the product, but it may also be later or sooner. If it is later, we speak commonly of reservation or postponed begin. If it is sooner, we speak commonly of starting in retrospect.



# Time aspects in C- and P-acts/facts (3)





# Competence

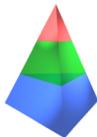
**Competence** is (primarily) defined as the collective knowledge, know-how and experience that is necessary and sufficient for a subject to perform (primarily) **production acts** as well as (secondarily) the corresponding coordination acts

Competence is mostly related to profession and education

Examples:

plumber  
physician  
judge





# Authority

A person acquires **authority** if an actor role is assigned to him/her. In practice, this is the result of a transaction, like hiring a person by an institution (company, society)

A normal prerequisite is that the person has the needed competence

Examples:



plumber at company X  
physician at hospital Y  
judge at court Z



# Responsibility

The original definition of **responsibility** the socially felt disposition to exercise assigned authority in a right way, thus as feeling responsible. Nowadays, it is mostly understood a being responsible, thus more as the obligation to exert the assigned authority, and in a proper way.

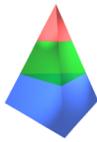
Responsibility shows itself particularly in performing **coordination acts**



Examples:

plumber of company X    ⇔  
physician in hospital Y    ⇔  
judge at court Z            ⇔

client of company X  
patient of hospital Y  
defendant at court Z



# How transparent is your organisation?

This is a story about four people, named Everybody, Somebody, Anybody and Nobody.

There was an important job to be done and Everybody was asked to do it.

Everybody was sure Somebody would do it.

Anybody could have done it, but Nobody did it.

Somebody got angry about that, because it was Everybody's job.

Everybody thought Anybody could do it but Nobody realized that Everybody wouldn't do it.

It ended up that Everybody blamed Somebody when Nobody did what Anybody could have done.

# Way of Modelling



## Construction Model

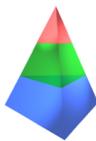


# The ontological model of an organisation

The **ontological model** of an organisation is the understanding of its **construction** and **operation** within the PSI, OMEGA and ALPHA theories; it is divided into four coherent sub models:

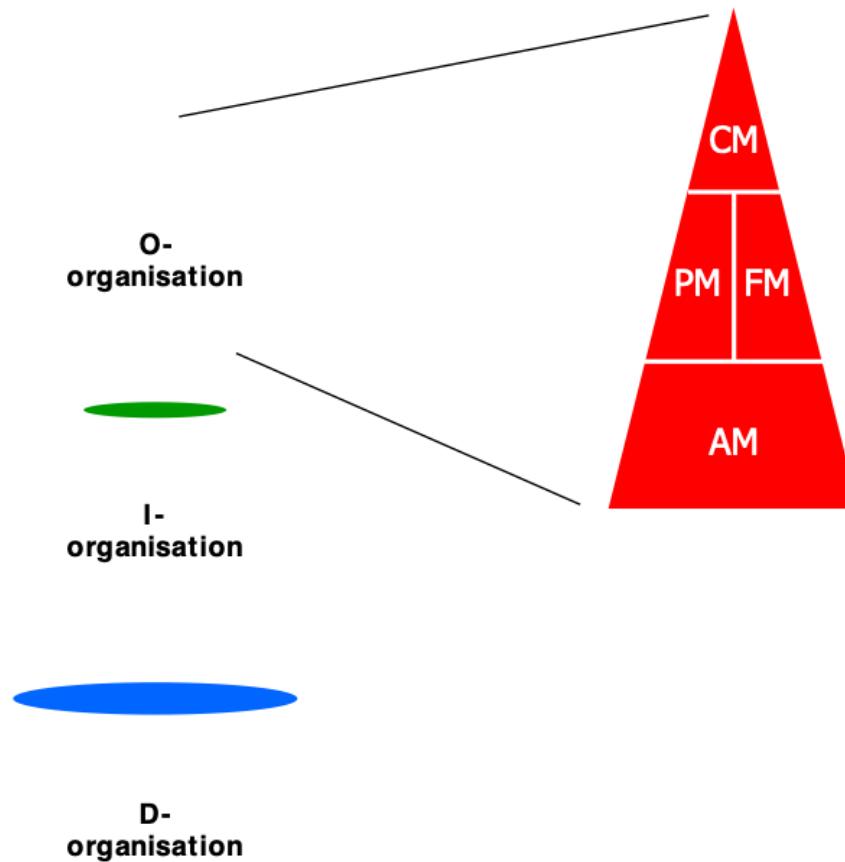
- **Construction Model (CM)**: the transaction kinds and actor roles, and the (initiator, executor, and information) links between them.
- **Action Model (AM)**: the action rules that guide actors in dealing with C-events, and the work instructions for producing products.
- **Process Model (PM)**: the transaction kinds, and the (response and wait) links between them.
- **Fact Model (FM)**: the entity kinds, the product kinds, and the property and attribute kinds in the production world.

Ideally, the four sub models are produced incrementally: as soon as a part of the CM is ready, one can produce the corresponding parts of the other three models. This way of working emphasises the inherent relationships between the four sub models.



# The essential model of an enterprise

The **essential model** of an enterprise is the ontological model of its O-organisation. It is expressed in a CM, AM, PM and FM.



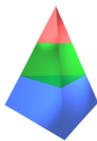


# The essential model of an enterprise (2)

The **essential model** of an enterprise is the understanding of the construction and operation of its **O-organisation** within the PSI, OMEGA and ALPHA theories; it is divided into four coherent sub models:

- **Construction Model (CM):** the transaction kinds and actor roles, and the (initiator, executor, and information) links between them.
- **Action Model (AM):** the action rules that guide actors in dealing with C-events, and the work instructions for producing products.
- **Process Model (PM):** the transaction kinds, and the (response and wait) links between them.
- **Fact Model (FM):** the entity kinds, the product kinds, and the property and attribute kinds in the production world.

Ideally, the four sub models are produced incrementally: as soon as a part of the CM is ready, one can produce the corresponding parts of the other three models. This way of working emphasises the inherent relationships between the four sub models.



# Quality aspects of the essential model

The essential model of an enterprise satisfies the **C<sub>4</sub>E** quality requirements. It is:

## Coherent

It constitutes the integrated whole of four sub models

## Consistent

The sub models fit perfectly together. There are no contradictions

## Comprehensive

It includes everything that is deemed necessary: process view, data view, business rules view, etc.

## Concise

Because it is rooted in very effective theories, it is very compact

## Essential

Because it is abstracted from realisation and implementation, it offers an unprecedented reduction of complexity

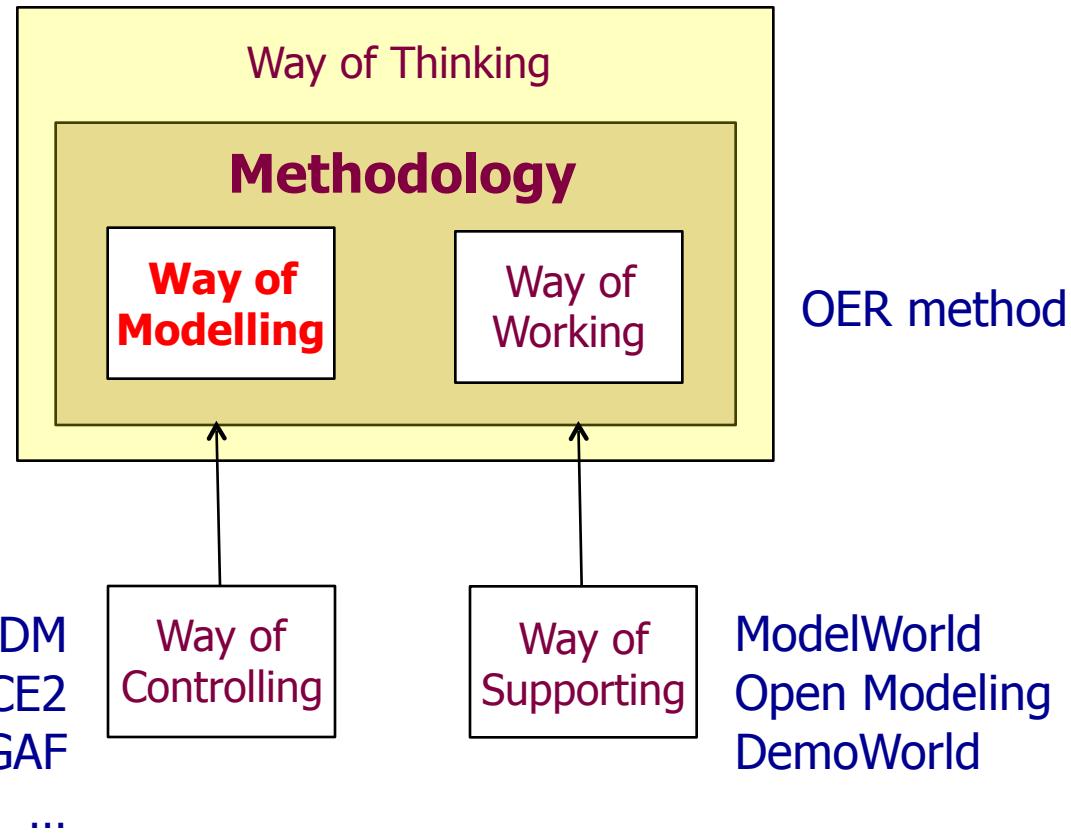


# DEMO in the Five Ways Framework

MU, TAO, FI; DELTA, PSI, OMEGA; ALPHA, BETA; SIGMA

## Construction Model

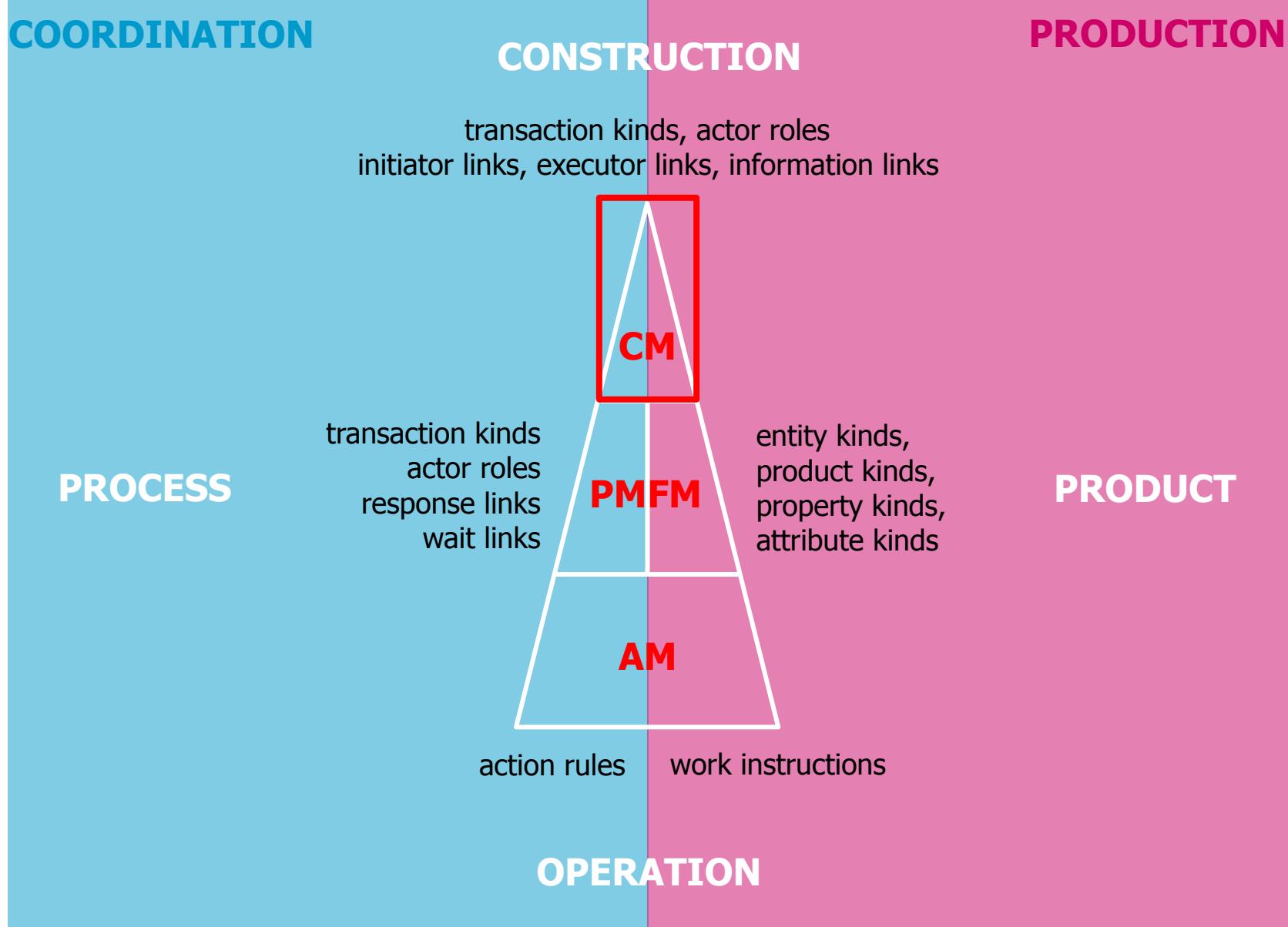
Process Model  
Fact Model  
Action Model



adapted from P.S. Seligmann, G.M. Weijers, H.G. Sol: analyzing the structure of IS methodologies – an alternative approach, 1989



# The Construction Model (1)





# Interaction and interstriction

In the ontological model of an organisation, actors influence each other in two ways:

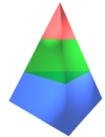
an active way, called interaction, and  
a passive way, called interstriction.

In performing coordination acts, actors create coordination events to which (other) actors respond. This way of mutual influencing is called **interaction**.

When responding to a coordination event, actors reckon with the current state of both the C-world and the P-world. By taking the state of the world into account, actors restrict each other's freedom of decision, since they change this state themselves.

This way of mutual influencing is called **interstriction**.

Interstriction stands for mutual (re)striction.



# Two interpretations of the transaction shape

Corresponding with the distinction between interaction and interstriction, there are two interpretations of the transaction shape.

In the **structure interpretation**, the transaction shape represents the complete **transaction pattern**, which contains all paths that may be followed while carrying out a transaction of the transaction kind. The structure interpretation corresponds with **interaction**.

In the **process interpretation**, the transaction shape represents a **transaction bank**: the conceptual container of all coordination facts in all transactions of the transaction kind, up to the current moment. The process interpretation corresponds with **interstriction**.

Interstriction is modelled by information links between actor roles and transaction banks. An **information link** represents the (reading) access right of an actor role to the contents of a transaction bank.



# The Construction Model (2)

The **Construction Model (CM)** of a Scope of Interest (SoI) is the ontological model of its **construction**. It contains:

- the identified internal actor roles and environmental actor roles
- the identified border transaction kinds and internal transaction kinds
- the initiator links between these actor roles and transaction kinds
- the executor links between these actor roles and transaction kinds
- the identified information links between the internal actor roles and (internal or external) transaction banks

A CM is expressed collectively in a:

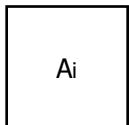
**Transaction Product Table (TPT)**

**Organisation Construction Diagram (OCD)**

**Bank Contents Table (BCT) (not covered here)**



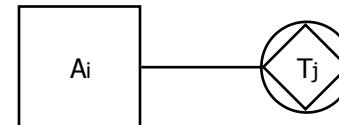
# Legend of the OCD (1)



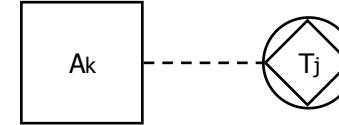
elementary  
actor role  $A_i$



composite  
actor role  $CA_k$



$A_i$  is an initiator role of  $T_j$



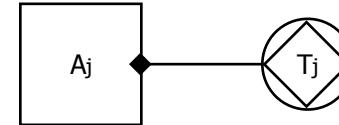
$A_k$  uses facts in the  
transaction bank of  $T_j$



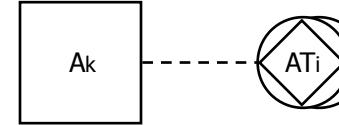
transaction kind  $T_j$



aggregate  
transaction kind  $AT_i$



$A_i$  is the executor role of  $T_j$



$A_k$  uses facts in the  
transaction banks of  $AT_i$

\_\_\_\_\_

initiator link

————◆————

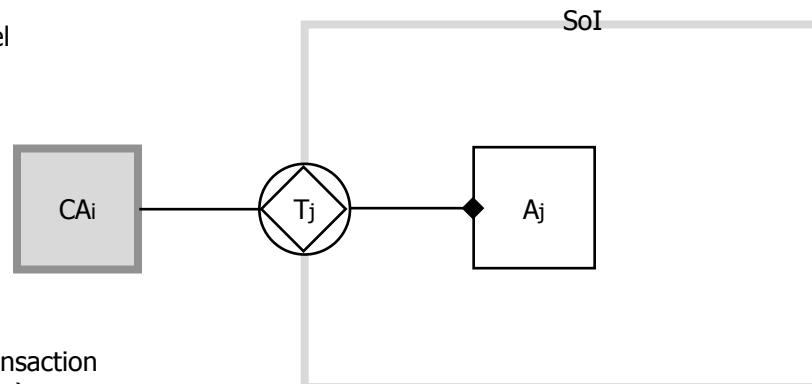
executor link

-----

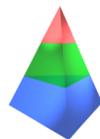
information link

The Scope of Interest (SoI) determines the kernel  
of the organisation that one wants to study

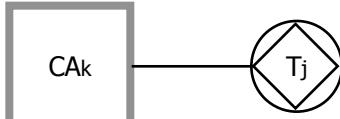
$CA_i$  is an environmental (composite) actor role  
 $A_j$  is an internal actor role  
 $T_j$  is a border transaction kind



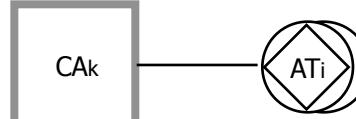
NOTE. Environmental actor roles and external transaction  
kinds are filled light-grey (like actor role  $CA_i$  above).



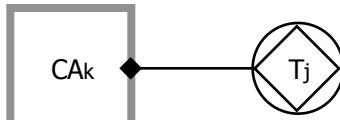
# Legend of the OCD (2)



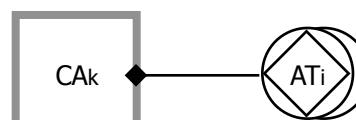
some actor role in CAk  
is an initiator role of Tj



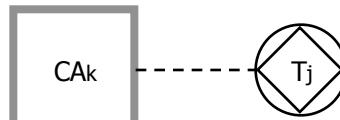
actor roles in Cak are initiator roles  
in transactions in ATi



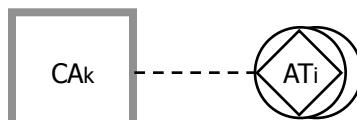
some actor role in CAk  
is the executor role of Tj



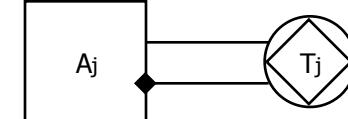
actor roles in Cak are executor roles  
in transactions in ATi



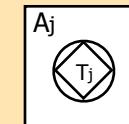
actor roles in CAk use facts in  
the transaction bank of Tj



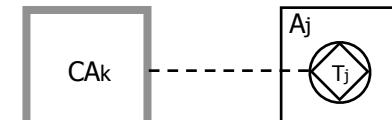
actor roles in CAk use facts in  
the transaction banks of ATi



self-activating actor role Aj (executor and  
initiator of transaction kind Tj )



shorthand notation of  
self-activating actor role Aj  
(executor and initiator of  
transaction kind Tj )



actor roles in CAk use facts in  
the transaction bank of Tj

# ILLUSTRATION



## Case Volley



# Volley: analysis (1)

T1/rq

One can become member of the tennis club Volley by sending a letter to the club by postal mail. In that letter one has to mention one's surname and first name, birth date, gender, telephone number, and postal mail address (street, house number, zip code, and town). Adam, the administrator of Volley, empties the mailbox daily and checks whether the information provided is complete. If not, he makes a telephone call to the sender in order to complete the data. Once a letter is complete, Adam writes an incoming mail number and the date on the letter, records the letter in the letter book, and puts it in a folder.

T1/pm

Every Wednesday evening, Adam takes the folder to Eve, the secretary of Volley. He also takes the member register with him. If Eve decides that an applicant can become member of Volley, she stamps 'new member' on the letter and writes the date below it. She then hands the letter to Adam in order to add the new member to the member register. This is a book with numbered lines. Each new member is entered on a new line. The line number is the number by which the new member is referenced in the administration.



## Volley: analysis (2)

Next, Eve calculates the fee that the new member has to pay for the remaining part of the calendar year. She asks Adam for the annual fee, as decided at the general assembly, which Adam has recorded on a sheet of paper in his files. Then, she asks Adam to write down the amount in the member register.

If Eve does not allow an applicant to become member (e.g., because he or she is too young or because the maximum number of members has been reached), Adam will send a letter in which he explains why the applicant cannot (yet) become member of Volley.

If all applications are processed, Adam takes the letters and the member register home and prepares an invoice to all new members for the payment of the first fee. He sends these invoices by postal mail. Payments have to be performed by bank transfers.

As soon as a bank statement is received, Adam prints a card on which the membership number, the starting date, the name, the date of birth, the gender, and the residence are mentioned. The card is sent to the new member by postal mail.

T1/dc

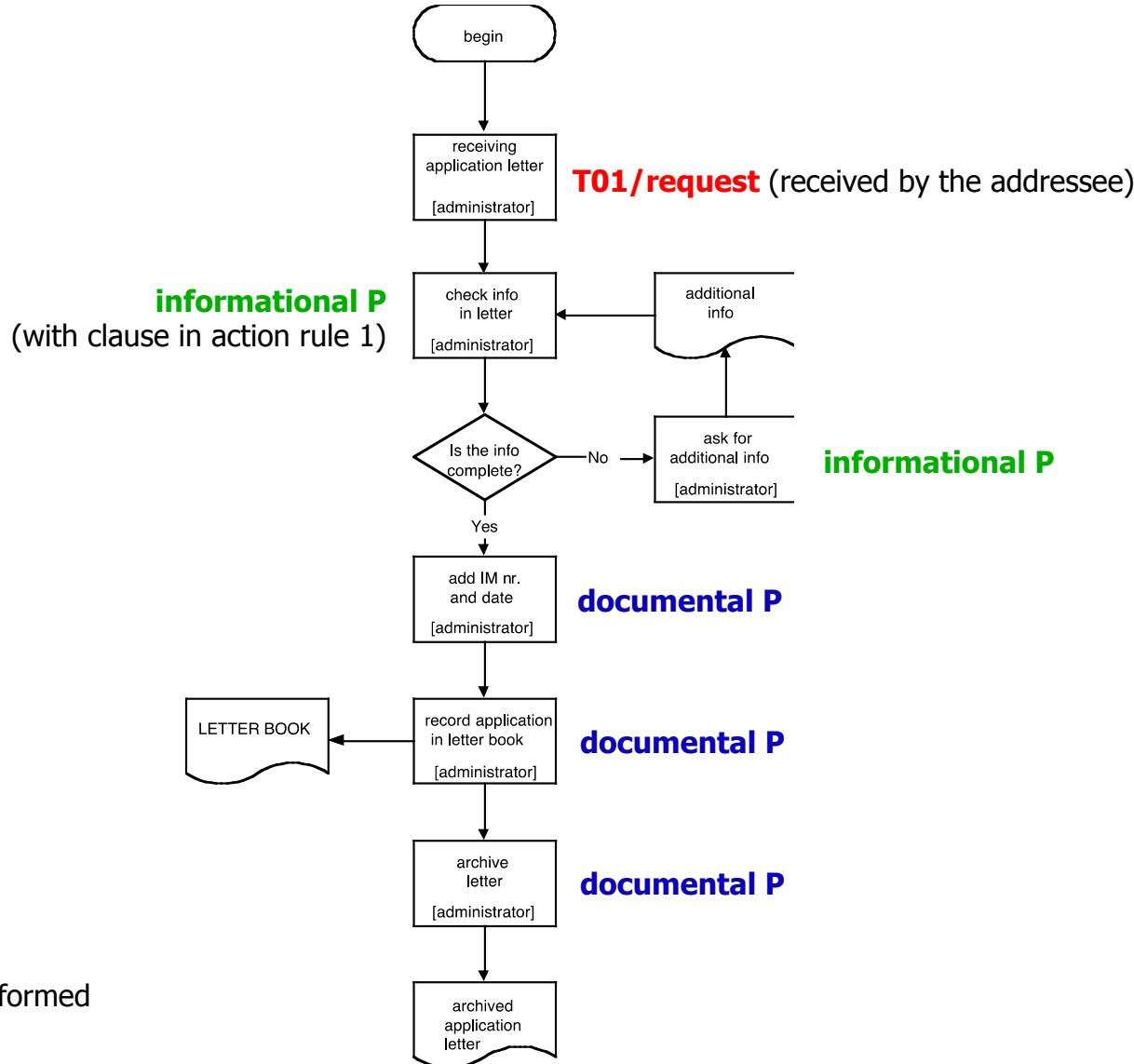
T2/rq

T2/st

T1/st

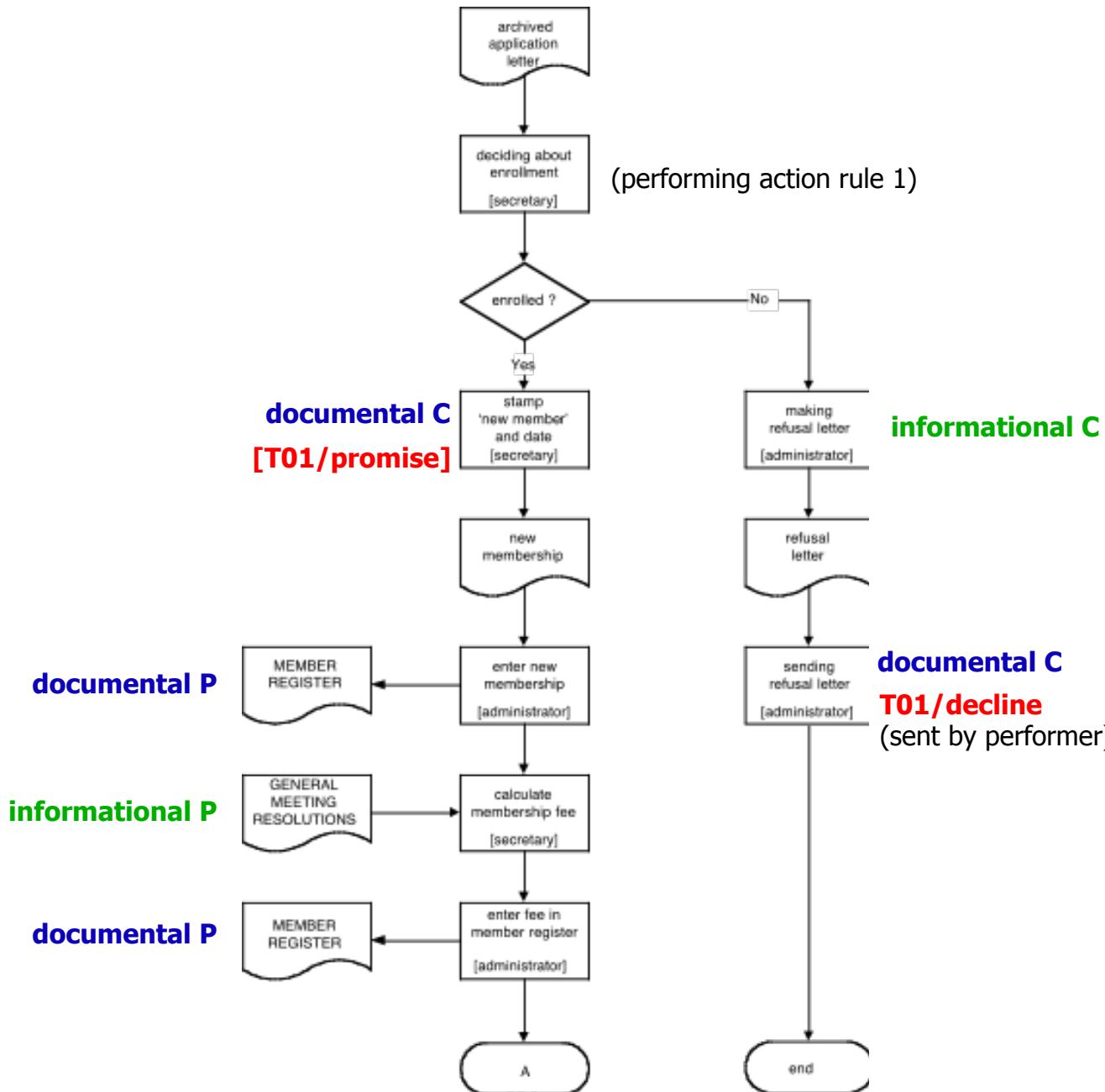


# PSI theory analysis of Volley Flow Chart 1



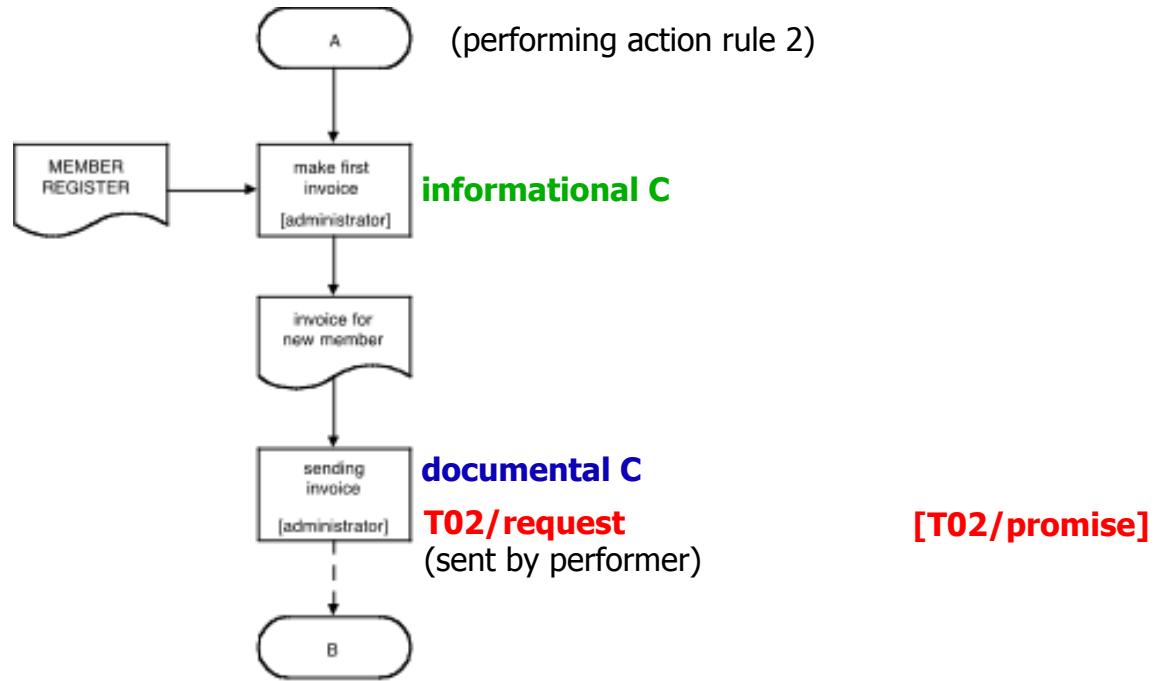


# PSI theory analysis of Volley Flow Chart 2



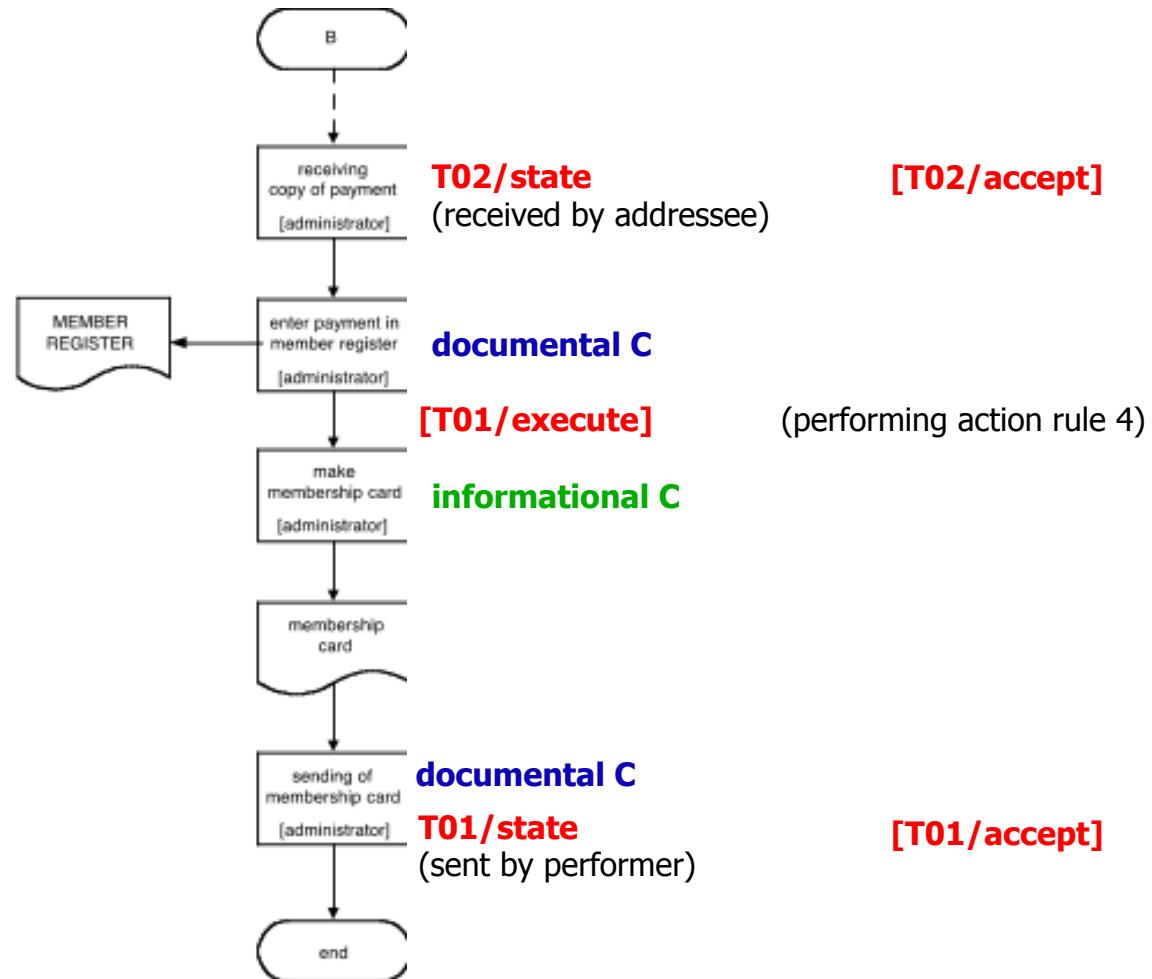


# PSI theory analysis of Volley Flow Chart 3

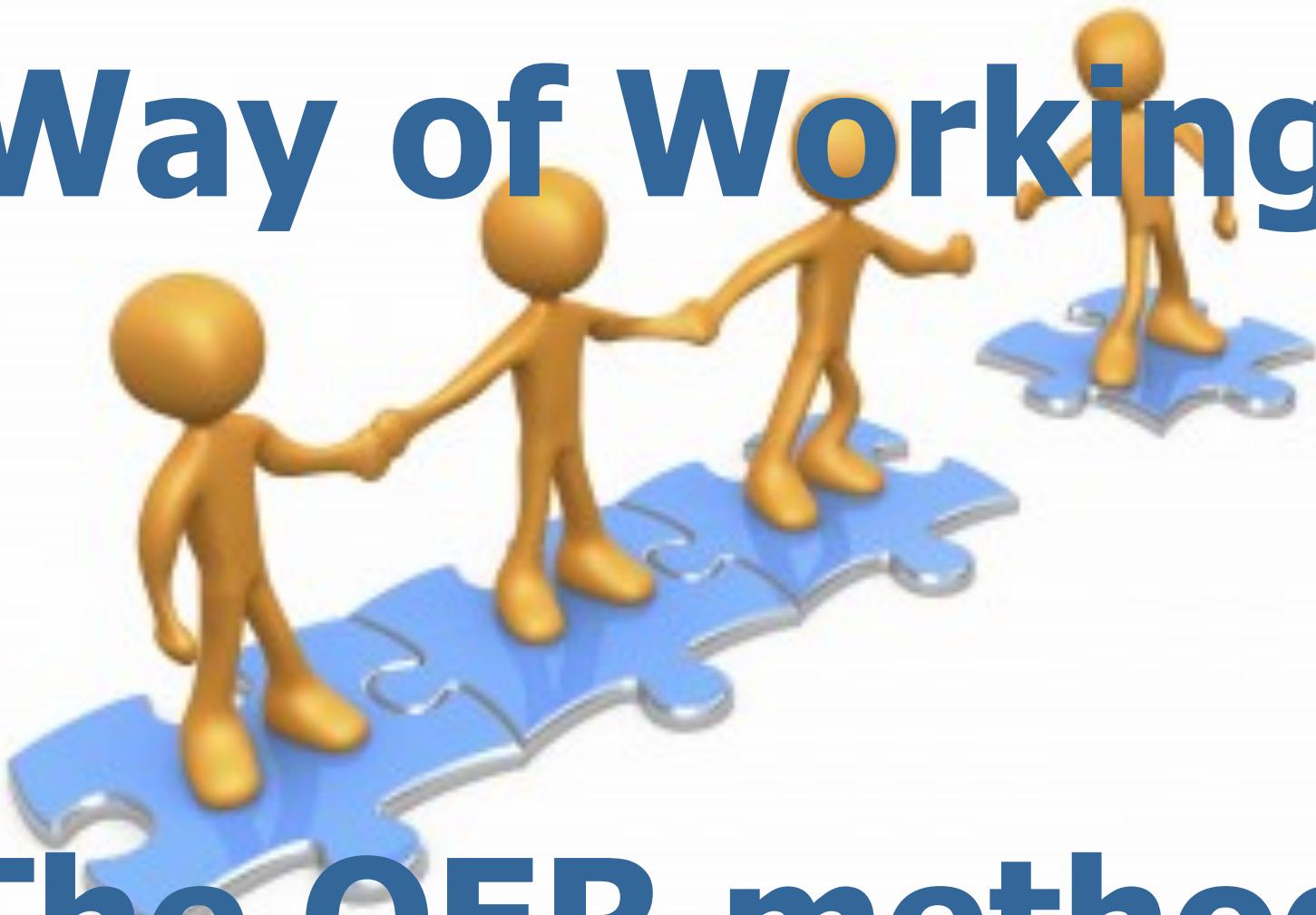




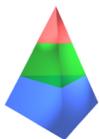
# PSI theory analysis of Volley Flow Chart 4



# Way of Working



The OER method  
Part 2



# The OER method applied to case Volley (3)

Applying the OER method to Volley yields two original product kinds  
(Note: **membership** is uniquely identifiable in time and space!):

**P1      Membership is started**

**P2      the first fee of Membership is paid**

The corresponding transaction kinds are:

**T1      membership starting**

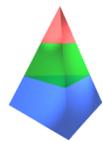
**T2      membership paying**

The corresponding executor roles are:

**A1      membership starter**

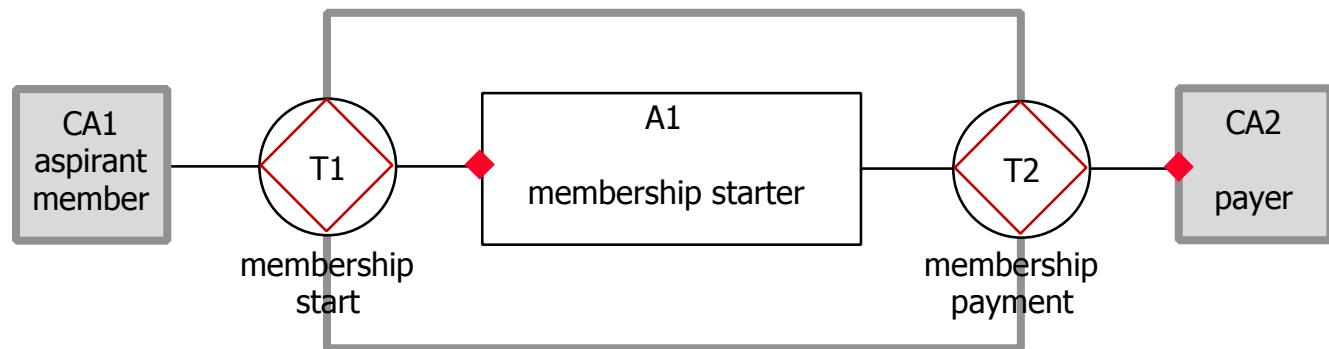
**A2      membership payer**

NOTE: the transaction kind T? is considered out of the SoI.



# Volley: Construction Model (interaction)

## Organisation Construction Diagram



## Transaction Product Table

transaction kind	product kind
T1 membership start T2 membership payment	P1 Membership is started P2 the first fee of Membership is paid



# Guidelines for producing a TPT

A **Transaction Product Table** (TPT) is a table of the **transaction kinds** in an SoI, and the corresponding **product kinds**. They are numbered by T<index> and P<index>, where <index> is a numeric code. So, for example T1, T2, T3, or T01, T02, T03, etc., and P1, P2, P3, or P01, P02, P03, etc.

A product kind is formulated as a predication (e.g. “is completed”) of an entity type (e.g. “sale”). It is expressed in a perfective sentence, in which the entity type is represented by a noun starting with a capital, e.g. “Sale is completed”.

The instances of the entity type must be **uniquely identifiable**. In the example above, every sale must have a unique identifier, like sale1, sale2, … , sale387, …

Give every transaction kind an appropriate name, corresponding with the product kind. For example: “sale completing”.



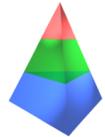
# Guidelines for producing an OCD (1)

1. Represent all transaction kinds and actor roles within the chosen SoI by a **composite actor role**. This is the **kernel** of the SoI.
2. Draw the **border transaction kinds** on the border of the SoI. A border transaction kind has an internal actor role as executor role and an environmental actor role as initiator role, or vice versa.
  - 2.1. If the executor role of a border transaction kind is an environmental actor role, draw this actor role and connect it with the transaction kind by means of an **executor link** (Note: one may also draw a composite actor role in which the executor role is contained). Then connect the border transaction kind with the kernel of the SoI by means of an **initiator link**.
  - 2.2. If the executor role of a border transaction kind is internal, connect the transaction kind with the kernel of the SoI by means of an **executor link**. Use or draw a **composite actor role** in the environment and connect it with the transaction kind by means of an **initiator link**.



# Guidelines for producing an OCD (2)

3. It is now time to 'unveil' the kernel composite actor role (which by definition covers a network of transaction kinds and actor roles).
  - 3.1. For every border transaction kind with an internal executor role, draw an **internal actor role**, and number it A<index> where <index> is equal to the index of the transaction kind of which it has the executor role. For example, the executor of T7 is A7. Connect the transaction kind with its executor role by an **executor link**.
  - 3.2. Next, draw all **internal transaction kinds** (from the TPT). Draw the corresponding (new) **actor role** that has the executor role and connect the two with an **executor link**.



# Guidelines for producing an OCD (3)

3.3. For every border transaction kind with an external executor role and for every internal transaction kind: try to find an internal actor role that has the initiator role in the transaction kind (Note that there may be more than one). Connect the transaction kind with the initiator role by an **initiator link**.

3.4. For every border transaction kind with an external executor role and for every internal transaction kind for which you cannot find an internal initiator role: introduce a **self-activating actor role** as the initiator role. Draw these self-activating actor roles and connect them with the 'loose' transaction kinds by an **initiator link**. Extend the TPT with the new transaction kind.

3.5 Because it is sometimes hard to produce the complete TPT from the beginning, one can always extend the TPT while modelling.



# Guidelines for producing an OCD (4)

4. It is now time to add information links. The ideal basis for determining them is the Action Model. If this model is missing, you have to rely on other documents, like a case description.

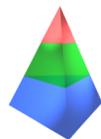
4.1. For every border transaction kind and every internal transaction kind: determine which internal actor roles must have (reading) access to the contents of the transaction banks. Draw an **information link** between the transaction bank and the actor roles (Note: there is an information link 'under' every initiator link and every executor link).

4.2. Determine whether there are external transaction kinds to which internal actor roles must have (reading) access. Draw them as **aggregate transaction banks**. Draw an **information link** between the transaction bank and the actor roles.

# ILLUSTRATION



Case  
**Wheelbarrows**



# Case Wheelbarrows: the Hortus view (1)

Hortus has always hundreds of wheelbarrows in use for the landscaping and maintenance of gardens. It is the responsibility of the purchase department of Hortus that there is always a sufficient number of wheelbarrows for the landscaping and maintenance gnomes to do their work.

We identify a self-activating actor role and transaction kind. Let us label the actor role A03 and call it “purchase manager”, and let us label the transaction kind T03 and call it “purchase management”. The product kind is P03 “purchase management for Period is done”.

Therefore, the purchasers of Hortus have access to an information source where they can look up the (estimated) number of wheelbarrows that must be available in a specific time period, as well as the minimum number for this period.

This paragraph mentions the existence of an information source that is inspected by A03. However, because we only model the interaction relationships, we ignore this knowledge.



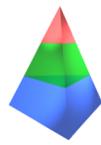
# Case Wheelbarrows: the Hortus view (2)

When there is a need to purchase a number of wheelbarrows, one of the purchasers of Hortus addresses himself to one of the salespersons of Malum, in order to carry out a transaction. This transaction is a purchase transaction for Hortus and a sale transaction for Malum.

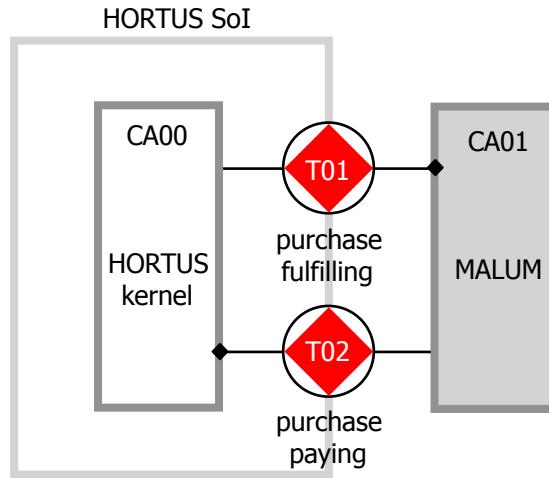
In this paragraph, we identify a transaction kind T01 “purchase fulfilling” with product kind P02 “Purchase is fulfilled”. Every transaction of this kind regards a particular purchase (with its own article kind, delivery date, number of items, etc.).

Clearly, A03 has the initiator role of T01, whereas the executor role (A01) is within the Malum enterprise. Because our SoI is Hortus, Malum is represented by a composite actor role CA01 “MALUM”.

NOTE. The existence of T02 in the OCD of Hortus is a consequence of the analysis of the Malum view on the case Wheelbarrows.



# Case Wheelbarrows – assignment 1 (1)

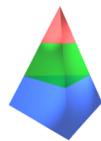


Transaction kind

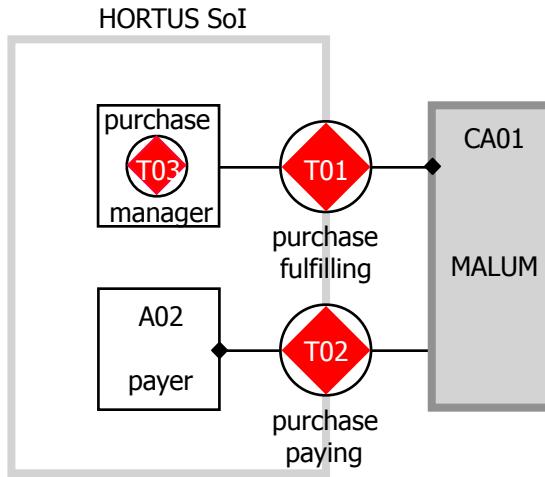
T01 purchase fulfilling  
T02 purchase paying

Product kind

P01 Purchase is fulfilled  
P02 Purchase is paid



# Case Wheelbarrows – assignment 1 (2)



Transaction kind	Product kind
T01 purchase fulfilling	P01 Purchase is fulfilled
T02 purchase paying	P02 Purchase is paid
T03 purchase management	P03 purchase management for Period is done

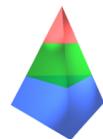


# Case Wheelbarrows: the Malum view (1)

As said before, the same transaction is a purchase transaction for Hortus and a sale transaction for Malum. Because of their long lasting business relationship, the goods of a sale order are normally delivered to Hortus before the corresponding invoice has been paid. Yet, the completion of the payment is a condition for completing the sale transaction.

It seems convenient to label the sale transaction kind also T01 but to call it now “sale completing”. The product kind P01 is “Sale is completed”. Every transaction of this kind regards a particular salen (with its own article kind, delivery date, number of items, etc.). The initiator role lies within Hortus (modelled as the composite actor role CA02 “HORTUS”), whereas actors A01 (seller) are the executor of transactions T01.

We also identify transaction kind T02 “sale paying” with product kind P02 “Sale is paid”. A01 has the initiator role and the (environmental) actor role A02 (within Hortus) has the executor role.



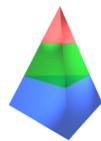
# Case Wheelbarrows: the Malum view (2)

Delivering implies the physical transportation of the ordered wheelbarrows from the warehouse of Malum to the goods receiving department of Hortus. This is done by employees of Malum. The transportation costs are included in the price of the wheelbarrow. To emphasise this, Malum calls delivering a free service.

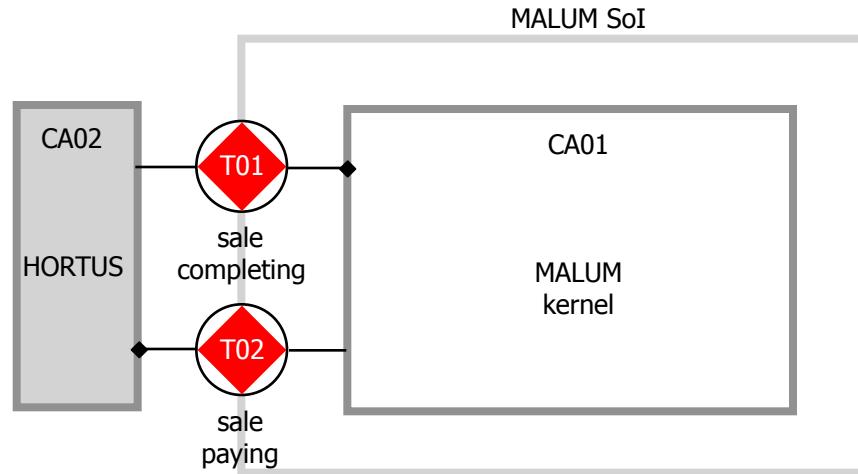
We identify transaction kind T03 “sale delivering” with product kind “Sale is delivered”, meaning that the goods of a particular sale are delivered to the customer (Hortus).

Note that the executor role (A03 “deliverer”), and thus also the transaction kind T03, is internal to Malum. It means (only) that the responsibility for delivering the goods is with Malum. Otherwise said, a CM is not a logistic flow model!

Note that transactions T02 and T03 are enclosed in a transaction T01.



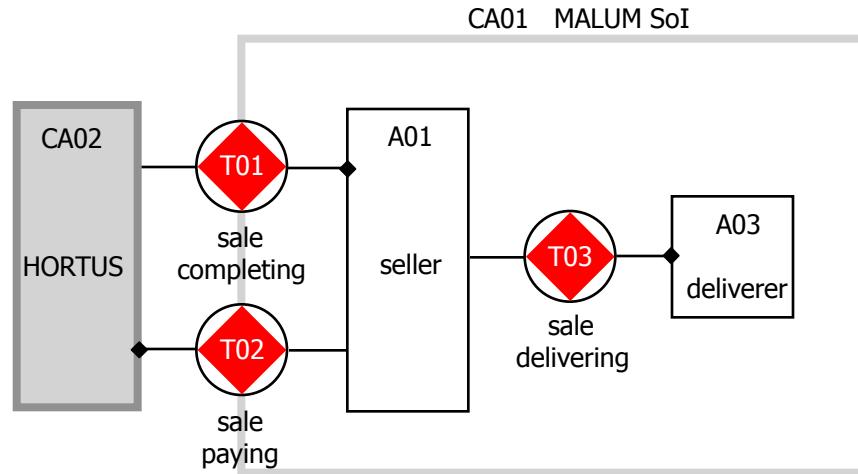
# Case Wheelbarrows – assignment 2 (1)



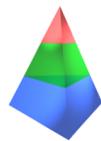
Transaction kind	Product kind
T01 sale completing	P01 Sale is completed
T02 sale paying	P02 Sale is paid



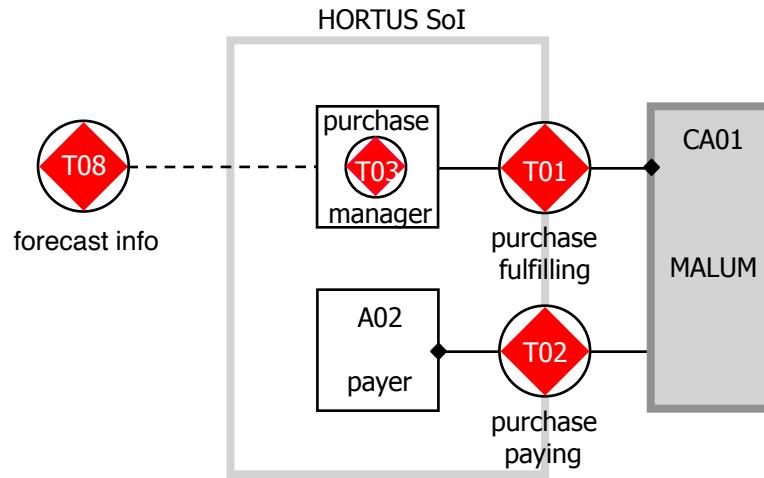
# Case Wheelbarrows – assignment 2 (2)



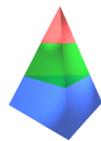
Transaction kind	Product kind
T01 sale completing	P01 Sale is completed
T02 sale paying	P02 Sale is paid
T03 sale delivering	P03 Sale is delivered



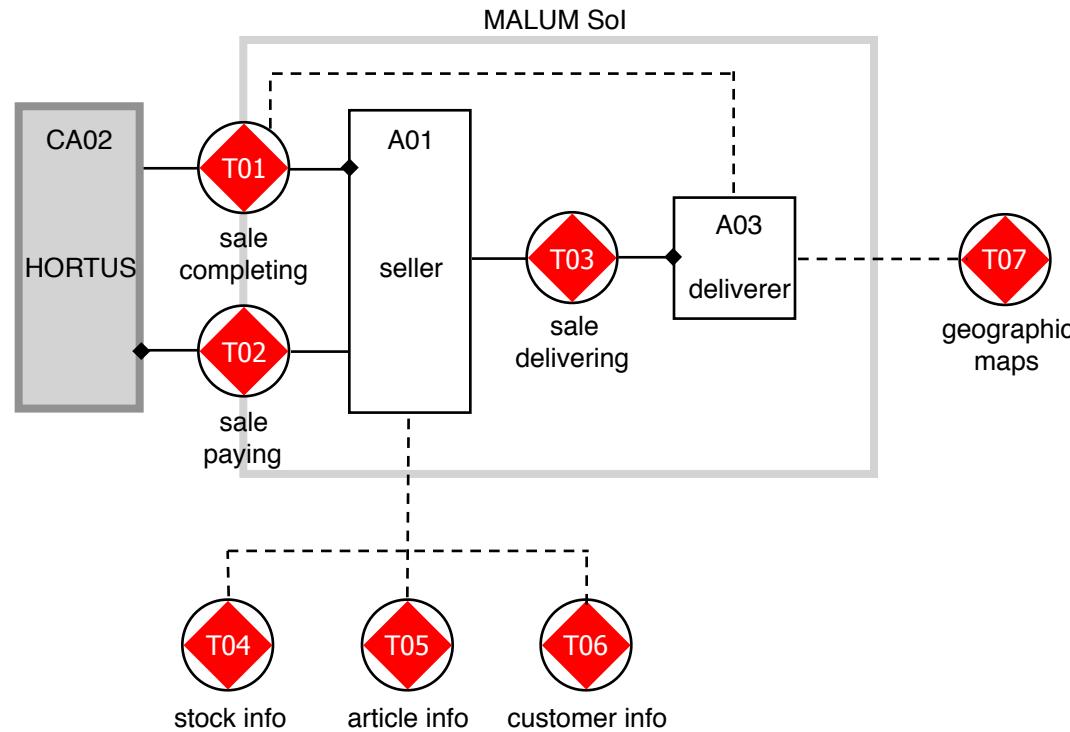
# Case Wheelbarrows – assignment 3 (1)



As mentioned in the case description, the purchase people at Hortus have access to an information source (T08 “forecast info”) where they can find the expected need for wheelbarrows.



# Case Wheelbarrows – assignment 3 (2)



From the case description, one can deduce that the actors A01 need to have knowledge about the article assortment (T05) and the customers (T06), as well as the current stock of wheelbarrows (T04). In addition, one can deduce that actors A03 need to know the details of sales orders (T01) as well as route information (T07).

# ILLUSTRATION



## Case Rent-A-Car



# RAC: analysis (1)

Rent-A-Car (or RAC for short) is a company that rents cars to persons, both private ones and representatives of legal bodies, like companies. It was founded by the twin brothers Janno and Ties back in the eighties. They started to hire out their own (two) cars, and they were among the first companies that allowed cars to be dropped off in a different location than where they were picked up. To this end, Janno and Ties had made agreements with students in several cities. For a small amount of money, a student would await the arrival of a rented car, e.g. at an airport, and drive it back to the office of RAC, after which the student would go home by public transport.

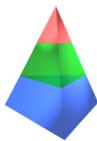
Currently, RAC operates from over fifty geographically dispersed branches in Europe. Many cities have a branch, some even several, and there are branches located near all airports. One of the branches is the original office where Janno and Ties started and where both are still around. Being mechanical engineer by education, they have kept loving to drive and maintain cars, even since they are the managing directors of a million euro company.

The head of the front office of the home branch is Chiara. There are two more desk officers working in this department. Customer orders are placed through several channels: walk-in, telephone, fax, and e-mail. **Walk-in customers** are usually people who want to rent a car immediately. Through the other channels one makes in general **advance reservations**. These can be made up to 200 days in advance. In all cases, an **electronic rental form** is filled out by one of the desk employees, as input to RACIS (RAC Information System). The next groups of data must be provided:

**T1/rq**

**T1/ex**

**T1/rq**



# RAC: analysis (2)

RENTAL: identification number (automatically generated), start date, end date, pick-up branch, drop-off branch, car group.

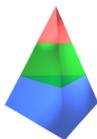
RENTER: identification (passport or driving license), first name, last name, address, date of birth, place of birth.

DRIVER: identification (driving license), first name, last name.

FINANCIAL: rental rate per day (basically determined by the car group).

Although it is the task of the desk officers to take the orders for renting a car, Janno or Ties may drop by and help a walk-in customer or pick up the telephone. Chiara does not really like these 'distortions' but she thinks she cannot do much about it. The problem with these spontaneous actions of Janno and Ties is that they often forget to record things properly, resulting in misunderstandings and even disputes with customers afterwards. Next, they sometimes act against the rules, for example by promising a car for a lower rate than the listed one.

The cars of RAC are divided in car groups. A car group may contain several types (brands and models). The common feature of the cars in a group is that they have the same rental rate per day. The board of directors, i.e. Janno and Ties, decide which brands and models belong to which group as well as what the rental rate is for every group. Normally they do this once a year.



# RAC: analysis (3)

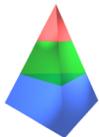
For a walk-in customer the starting day is usually the same day as on which the contract is established. Advance reservations have some future day as the starting day. RAC applies a maximum rental period (currently 10 days).

**T1/ac**

After the renter has signed the contract, the rental is concluded by the employee (Note: the signing by the renter counts as promising to pay the rental charge, which is the contracted duration times the daily rental rate. Because the rental may be an advance reservation, the payment may be delayed until the starting day).

**T3/ex**

On the starting day, the driver can pick up a car at the distribution department at the backside of the building, on presentation of a copy of the contract. There are three employees working in this department: Mik, Ferre, and Carlo, but not all of them are always present, as we will see. As soon as a driver shows up, one of them checks whether there is a car available of the contracted group. If there is one, he will allocate the car to the rental contract and sign the contract as being picked up. If there is no car available of the contracted group, he will 'upgrade' the contract and select a car from the next higher car group. The driver will get this 'upgraded' car, but for the price of the contracted group.



# RAC: analysis (4)

**T4/ex**

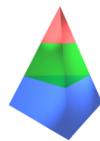
After the car of a rental has been dropped off at some branch, the possibly incurred fines have to be paid. There may be a penalty charge for returning the car after the contracted end date. It amounts to the number of extra days times the late return penalty rate. Next, the car may have been dropped off at another branch than the contracted return branch. In that case a location penalty charge has to be paid. This amounts to the distance between the actual and the contracted drop off branch times the penalty rate per kilometer.

**T5/ex**

The distribution department is also responsible for transporting cars between branches because cars may be dropped off at other locations, as we have seen. To this end, Mik schedules every morning the transportations that have to be performed that day. The transportations are carried out by all three of them, so also by Ferre and Carlo. That is why often some of them are away from the office.

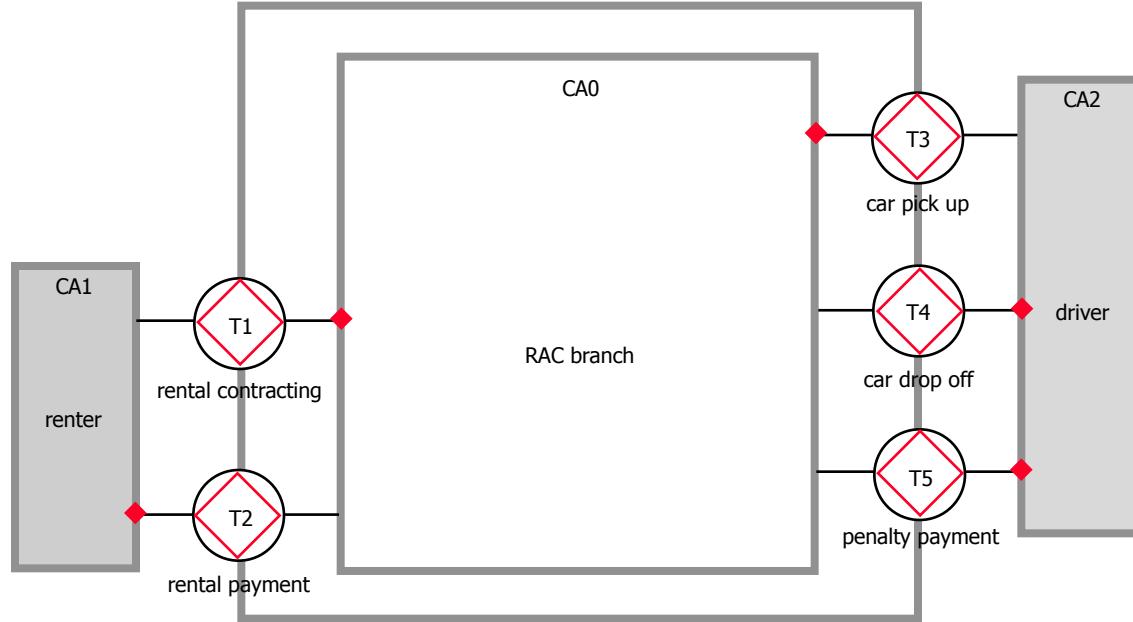
**T7/ex**

**T6/ex**



# RAC: Global OCD and TPT

## Organization Construction Diagram

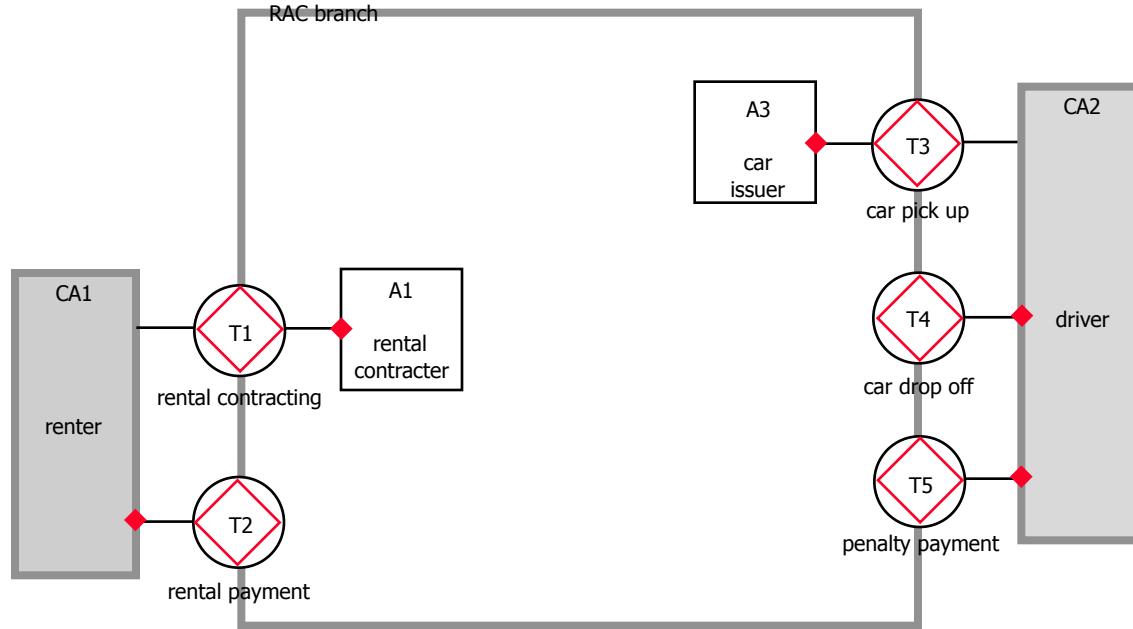


## Transaction Product Table

transaction kind	product kind
T1 rental concluding	P1 Rental <b>is</b> concluded
T2 rental paying	P2 <b>the rent of</b> Rental <b>is</b> paid
T3 car issuing	P3 <b>the car of</b> Rental <b>is</b> issued
T4 car returning	P4 <b>the car of</b> Rental <b>is</b> returned
T5 penalty paying	P5 <b>the penalty of</b> Rental <b>is</b> paid

# RAC: Detailed OCD and TPT (1)

## Organization Construction Diagram

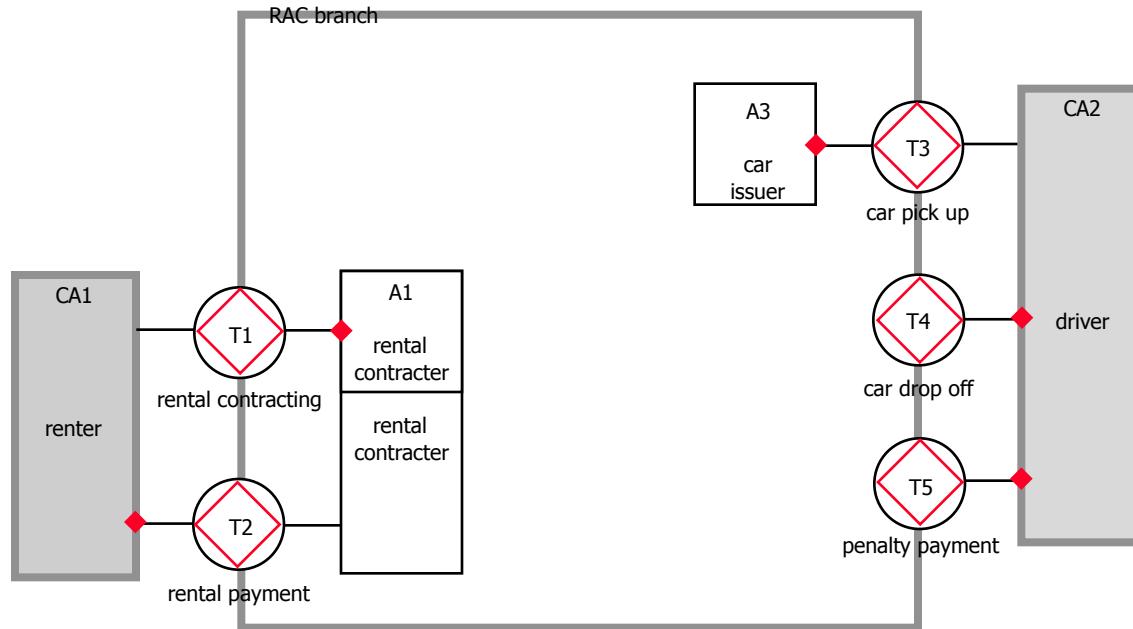


## Transaction Product Table

transaction kind	product kind
T1 rental concluding	P1 Rental <b>is</b> concluded
T2 rental paying	P2 <b>the rent of</b> Rental <b>is</b> paid
T3 car issuing	P3 <b>the car of</b> Rental <b>is</b> issued
T4 car returning	P4 <b>the car of</b> Rental <b>is</b> returned
T5 penalty paying	P5 <b>the penalty of</b> Rental <b>is</b> paid

# RAC: Detailed OCD and TPT (2)

## Organization Construction Diagram

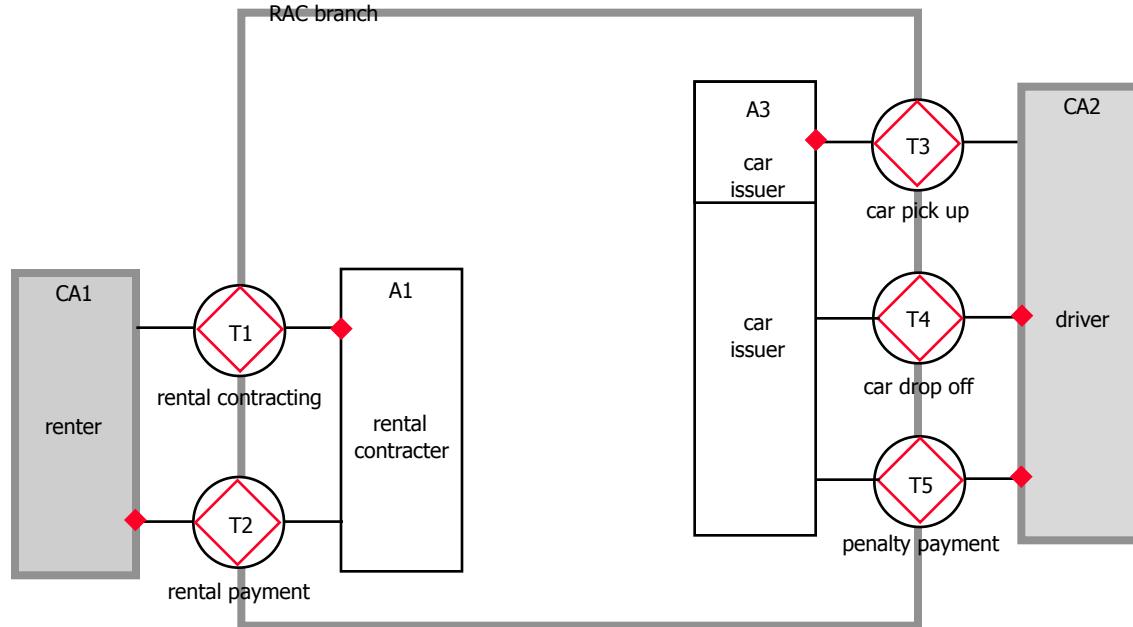


## Transaction Product Table

transaction kind	product kind
T1 rental concluding	P1 Rental <b>is</b> concluded
T2 rental paying	P2 <b>the rent of</b> Rental <b>is</b> paid
T3 car issuing	P3 <b>the car of</b> Rental <b>is</b> issued
T4 car returning	P4 <b>the car of</b> Rental <b>is</b> returned
T5 penalty paying	P5 <b>the penalty of</b> Rental <b>is</b> paid

# RAC: Detailed OCD and TPT (3)

## Organization Construction Diagram

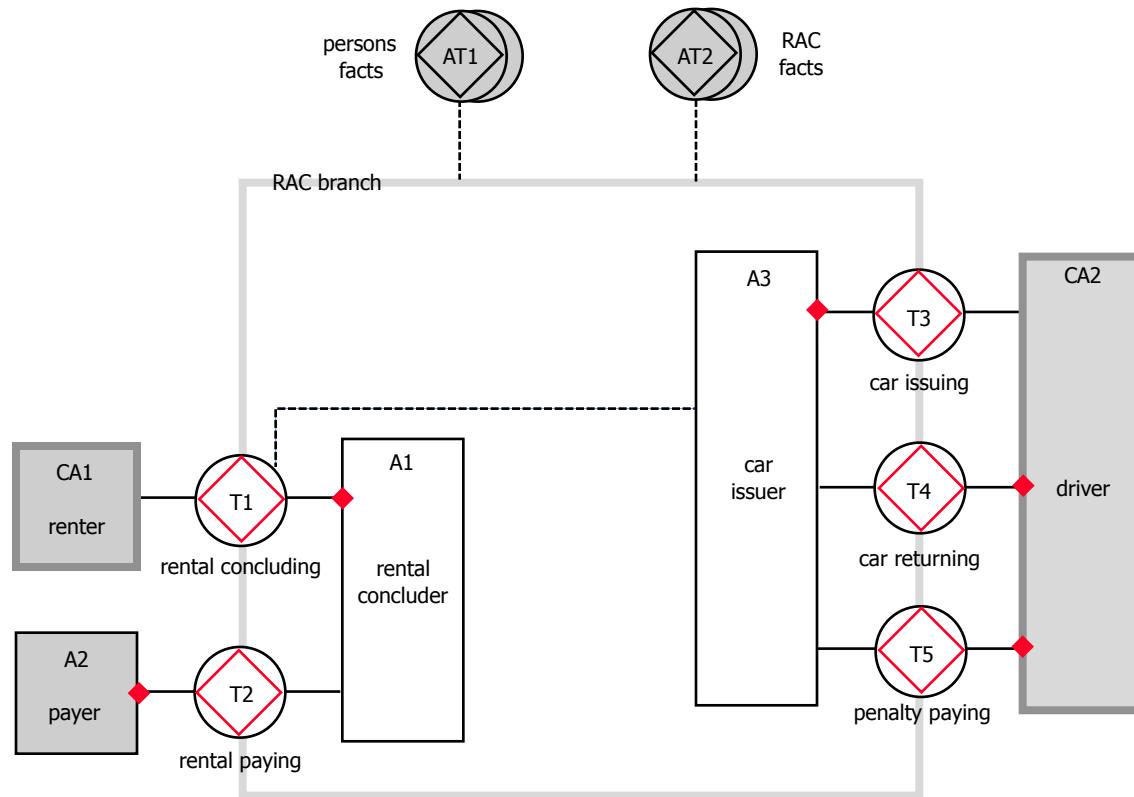


## Transaction Product Table

transaction kind	product kind
T1 rental concluding	P1 Rental <b>is</b> concluded
T2 rental paying	P2 <b>the rent of</b> Rental <b>is</b> paid
T3 car issuing	P3 <b>the car of</b> Rental <b>is</b> issued
T4 car returning	P4 <b>the car of</b> Rental <b>is</b> returned
T5 penalty paying	P5 <b>the penalty of</b> Rental <b>is</b> paid

# RAC: Detailed OCD and TPT (4)

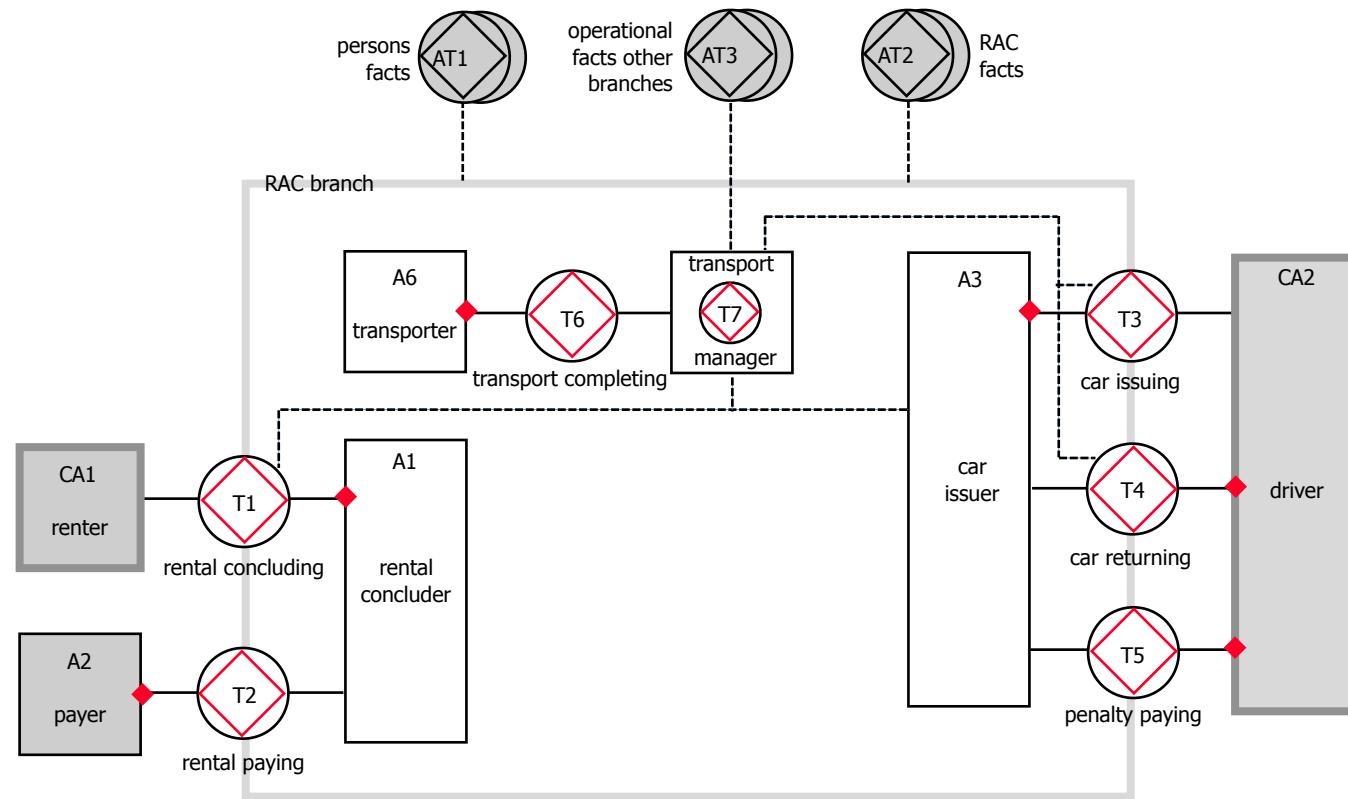
## Organization Construction Diagram



## Transaction Product Table

transaction kind	product kind
T1 rental concluding	P1 Rental <b>is</b> concluded
T2 rental paying	P2 <b>the rent of</b> Rental <b>is</b> paid
T3 car issuing	P3 <b>the car of</b> Rental <b>is</b> issued
T4 car returning	P4 <b>the car of</b> Rental <b>is</b> returned
T5 penalty paying	P5 <b>the penalty of</b> Rental <b>is</b> paid

# RAC: Detailed OCD and TPT (5)



transaction kind	product kind
T1 rental concluding	P1 Rental <b>is</b> concluded
T2 rental paying	P2 <b>the rent of</b> Rental <b>is</b> paid
T3 car issuing	P3 <b>the car of</b> Rental <b>is</b> issued
T4 car returning	P4 <b>the car of</b> Rental <b>is</b> returned
T5 penalty paying	P5 <b>the penalty of</b> Rental <b>is</b> paid
T6 transport completing	P6 Transport <b>is</b> completed
T7 transport management	P7 transport management <b>for</b> Day <b>is</b> done



# RAC: Actor Function Matrix

An Actor Function Matrix (or Actor Person Matrix) gives insight in the current or planned **implementation of actor roles**.

Rows and columns with several crosses need special consideration.  
Several crosses in a row means that a person fulfills several actor roles simultaneously; this may be a problem.

Several crosses in a column means that several persons fulfill the same actor role; this may be a problem.

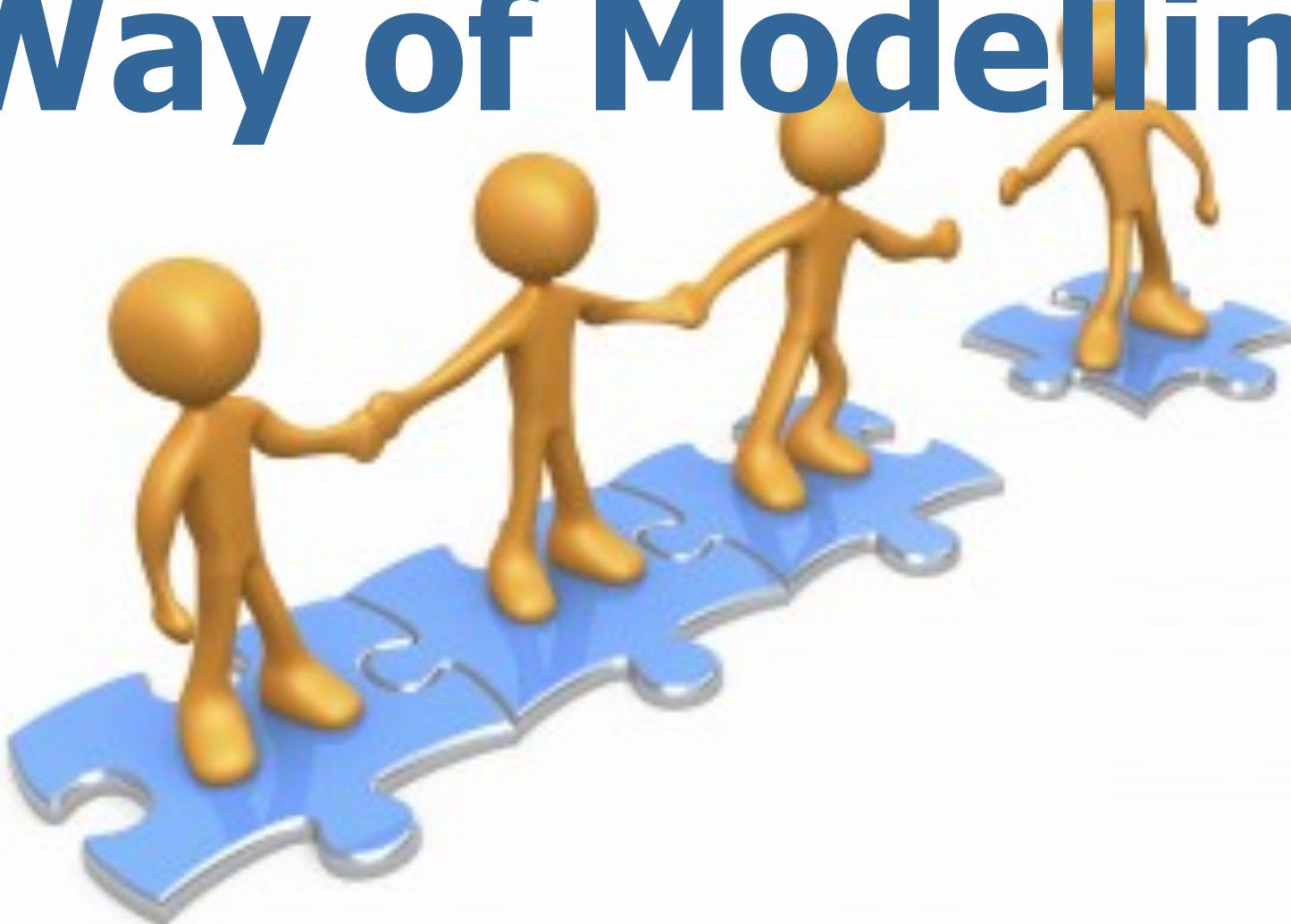
	A1	A3	A6	A7
Chiara	x			
Desk employee	x			
Mik		x	x	x
Ferre		x	x	
Carlo		x	x	

More detail:

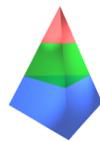
R = responsible

D = delegated

# Way of Modelling

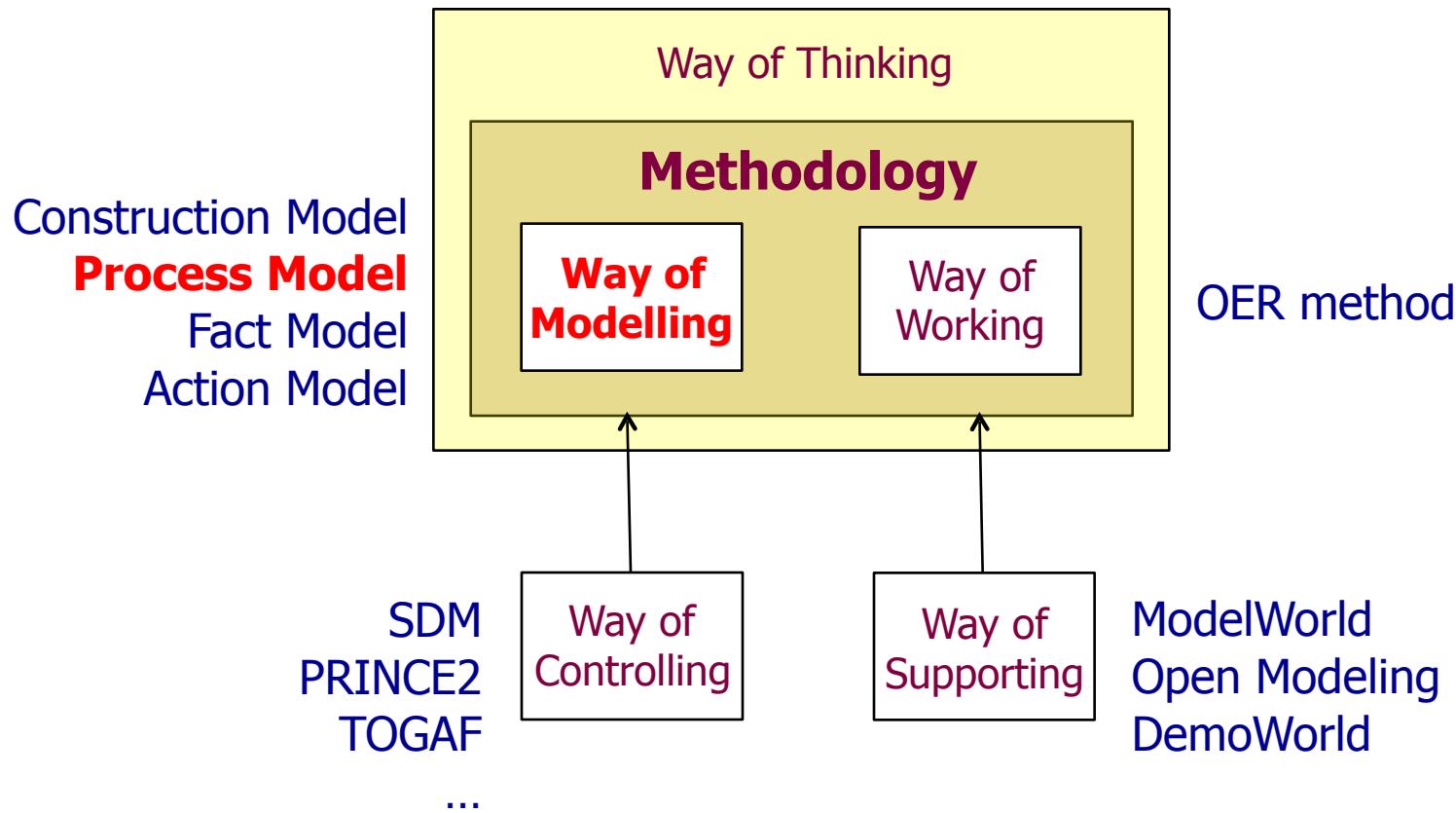


## Process Model



# DEMO in the Five Ways Framework

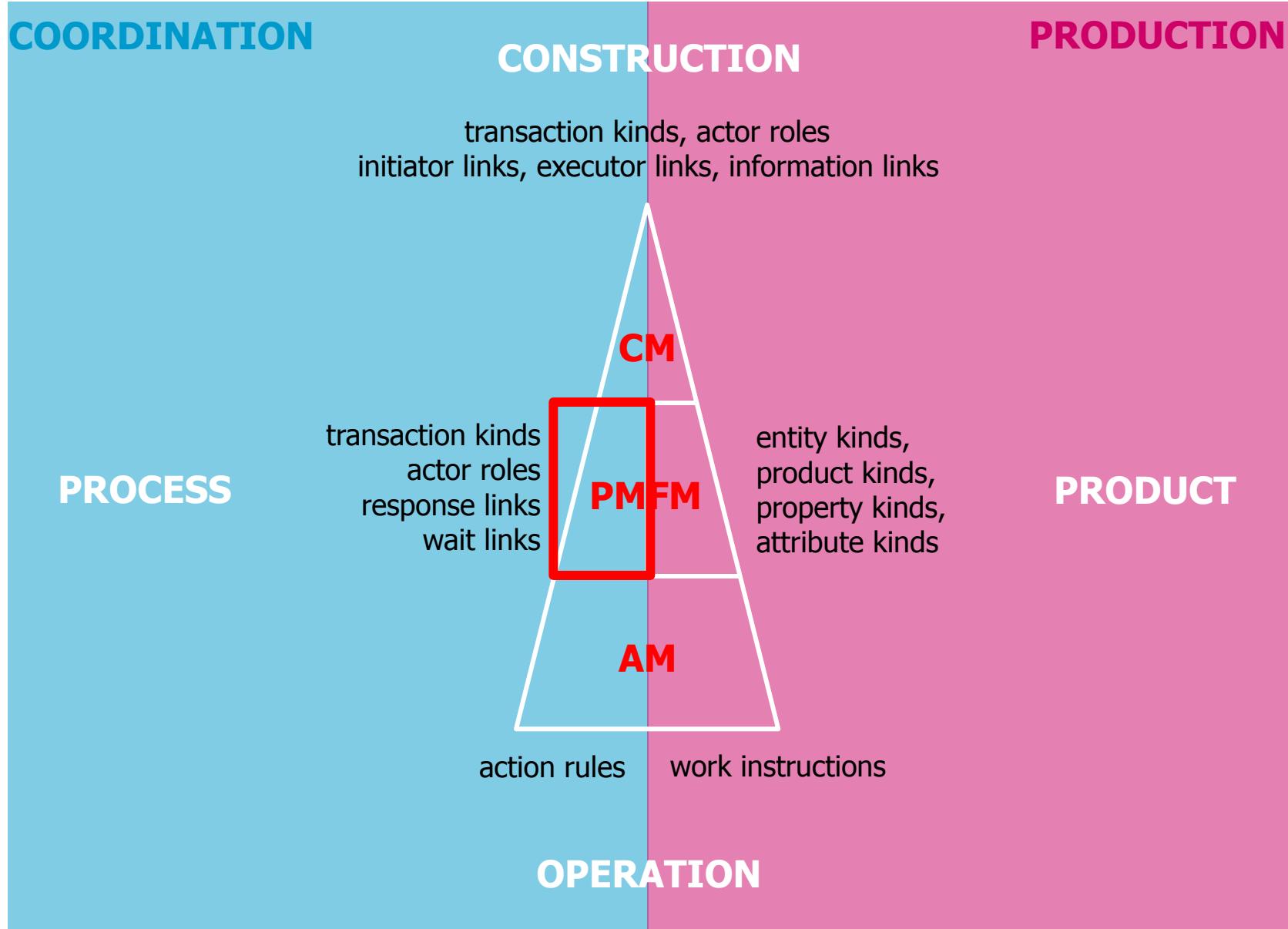
MU, TAO, FI; DELTA, PSI, OMEGA; ALPHA, BETA; SIGMA

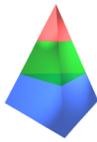


adapted from P.S. Seligmann, G.M. Weijers, H.G. Sol: analyzing the structure of IS methodologies – an alternative approach, 1989



# The Process Model (1)





# The Process Model (2)

The **Process Model (PM)** of a Scope of Interest is the ontological model of the effects of its operation on the **coordination world** (CW)

A PM contains:

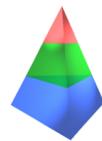
- The **transaction processes** of all internal and border transaction kinds.
- The **response links** and **wait links** between them.

A PM is expressed in:

**Process Structure Diagram (PSD)**

**Transaction Pattern Diagram (TPD)** for every transaction kind. In these (optional) diagrams, the 'exceptions' to be dealt with are indicated.

Ideally, the PM of an SoI is based on its AM. If the AM is missing, one has to rely on the case description.



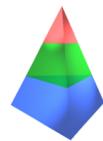
# Guidelines for producing the PSD

From the OCD, one can already derive the **tree structures** of transactions.

For every transaction tree, a PSD is produced. Every transaction kind is represented by a '**sausage**' (extracted disk) with the production diamond in it. Every 'sausage' is drawn over a grey bold line, which separates the **initiator role** (above the line) and the **executor role** (below the line).

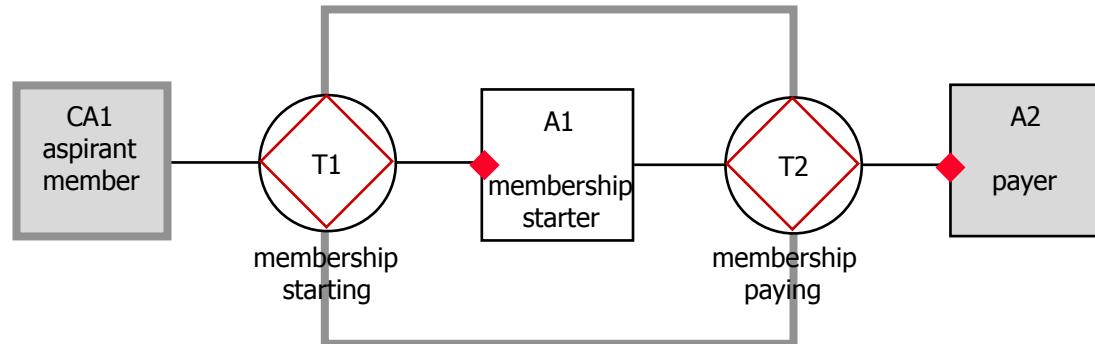
Its **enclosing** transaction kind is drawn **above** it, and the **enclosed** transaction kinds are drawn **below** it.

Based on the AM (or other information if the AM is lacking), one draws the **response links** and the **wait links** between the transaction kinds.



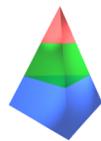
# Volley: Construction Model (only interaction)

## Organisation Construction Diagram

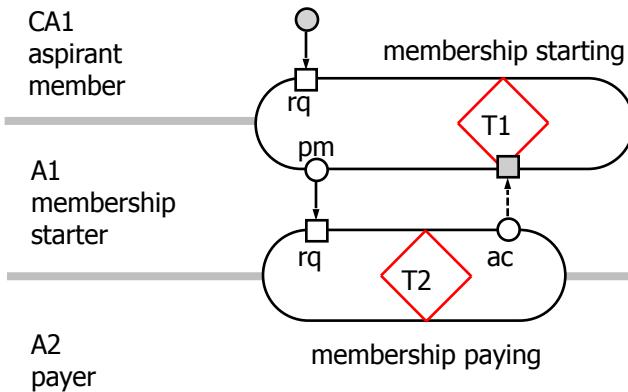


## Transaction Product Table

transaction kind	product kind
T1 membership starting	P1 Membership <b>is</b> started
T2 membership paying	P2 <b>the first fee of</b> Membership <b>is</b> paid



# Volley: Process Structure Diagram



The case description is not very clear about the status in T1 from which the T2/rq is performed. Therefore, we assume that the 'default' holds, which means that the T2/rq is performed from the T1/pm.

In the case description, one can read that the T1/st is performed after the T2/ac. Because the state act always follows the execute act, it is common practice to include a wait link from T2/ac to T12/ex.

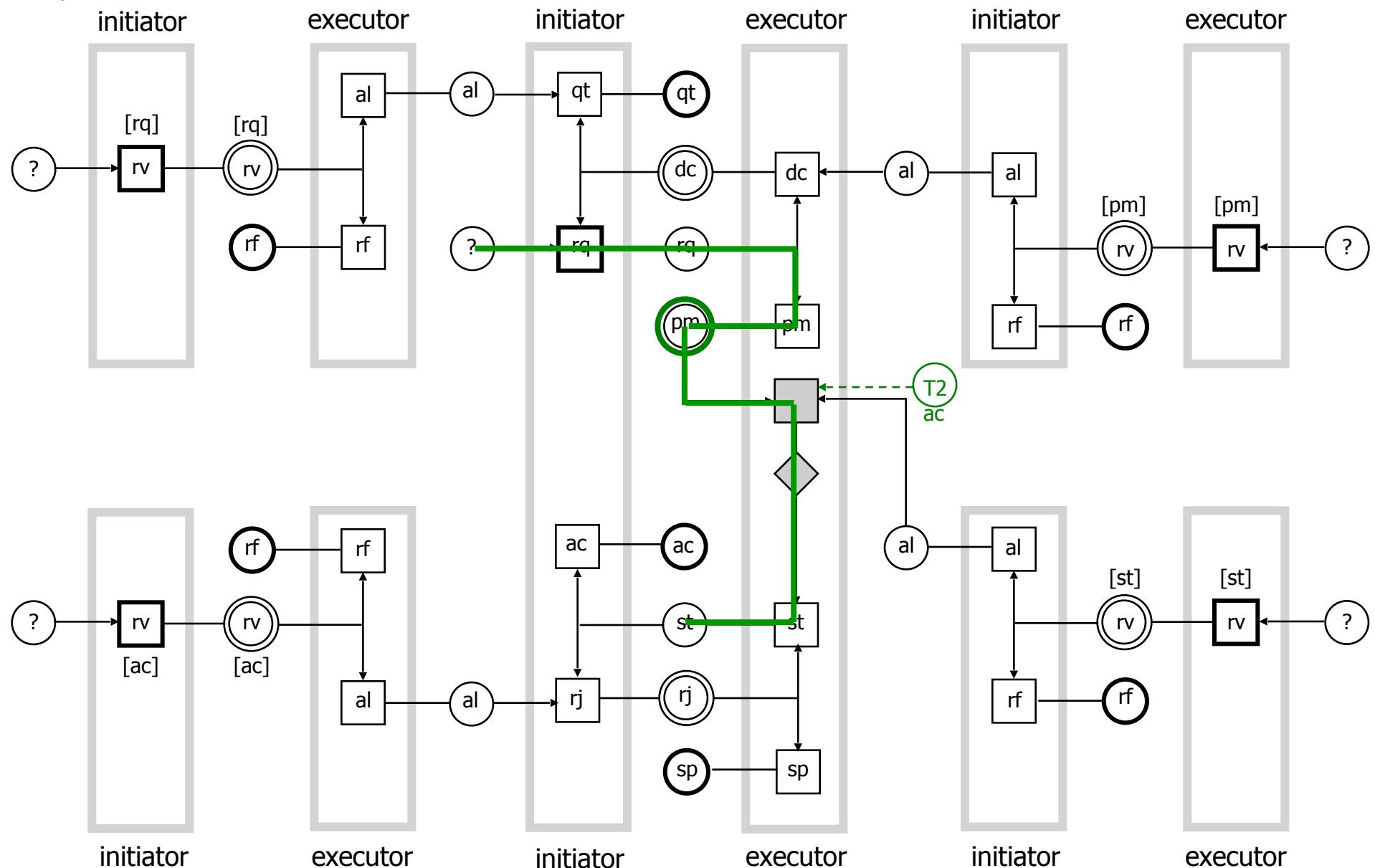
# ILLUSTRATION



## Case Volley



# Volley: Transaction kind T1



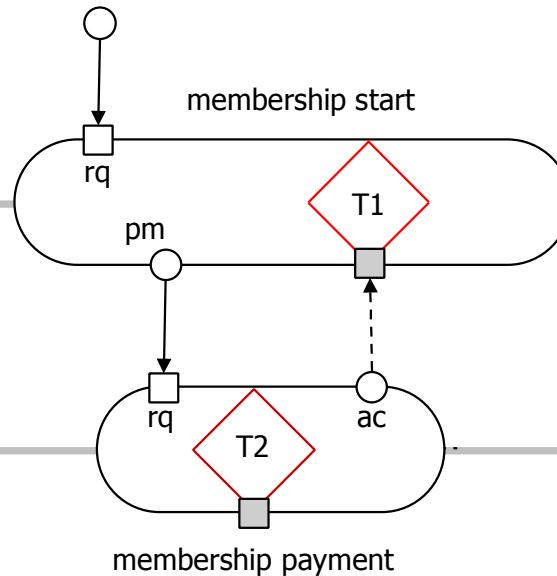


# Volley: Process Structure Diagram

CA1  
aspirant  
member

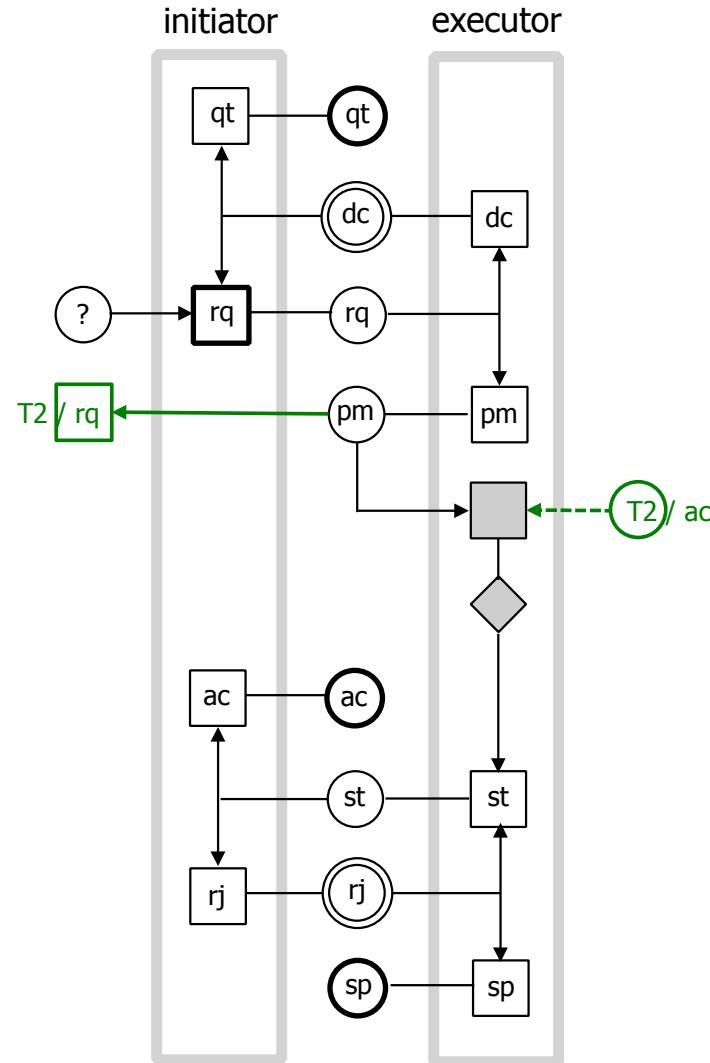
A1  
membership  
starter

CA2  
payer





# Volley: delegations in T1

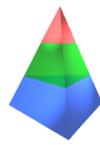


Eve/Adam

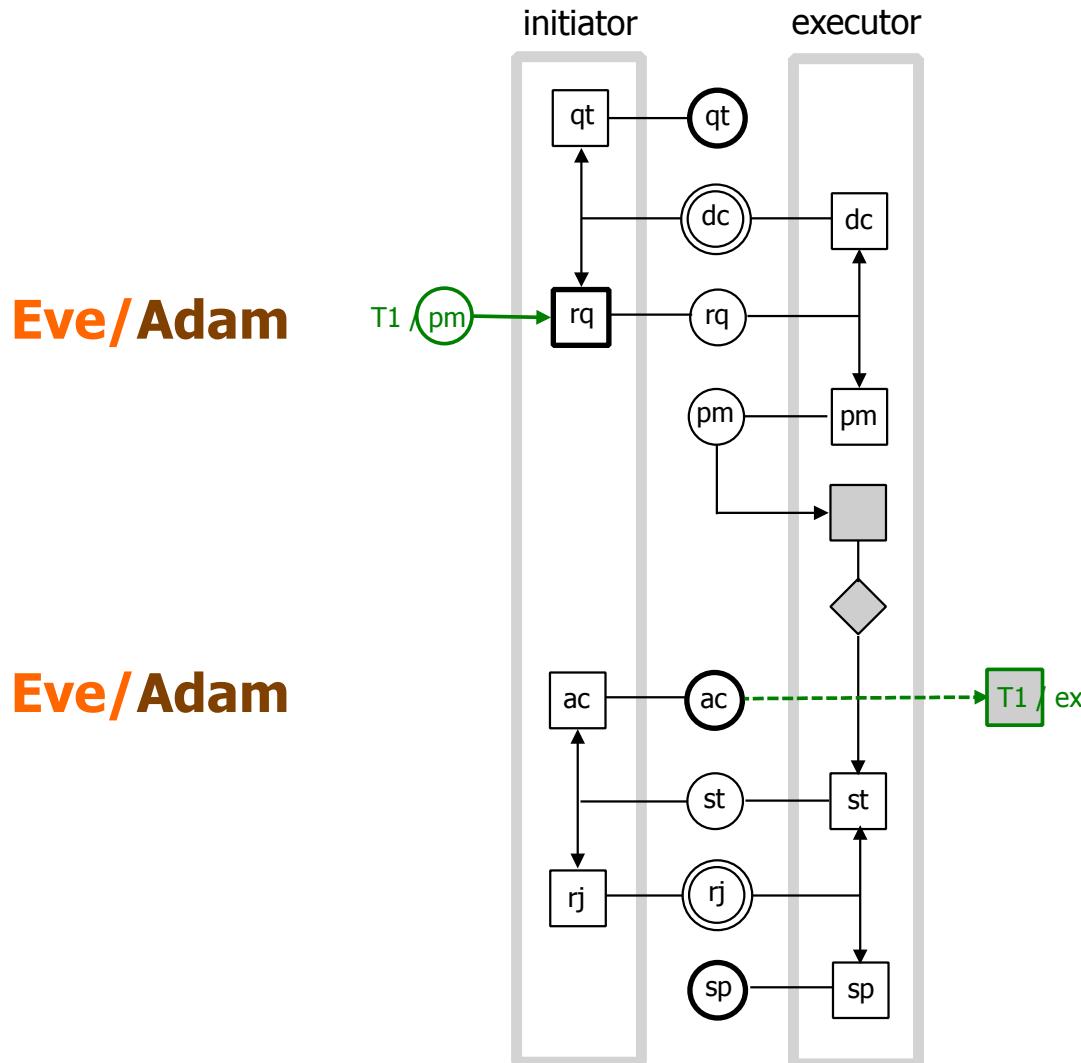
Eve (tacitly)

Eve

Eve/Adam



# Volley: delegations in T2



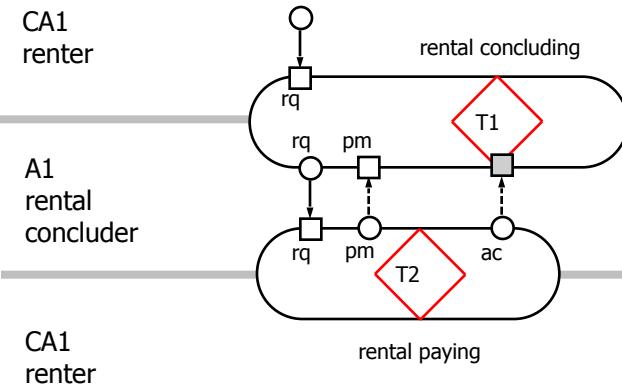
# ILLUSTRATION

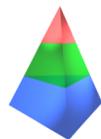


## Case Rent-A-Car

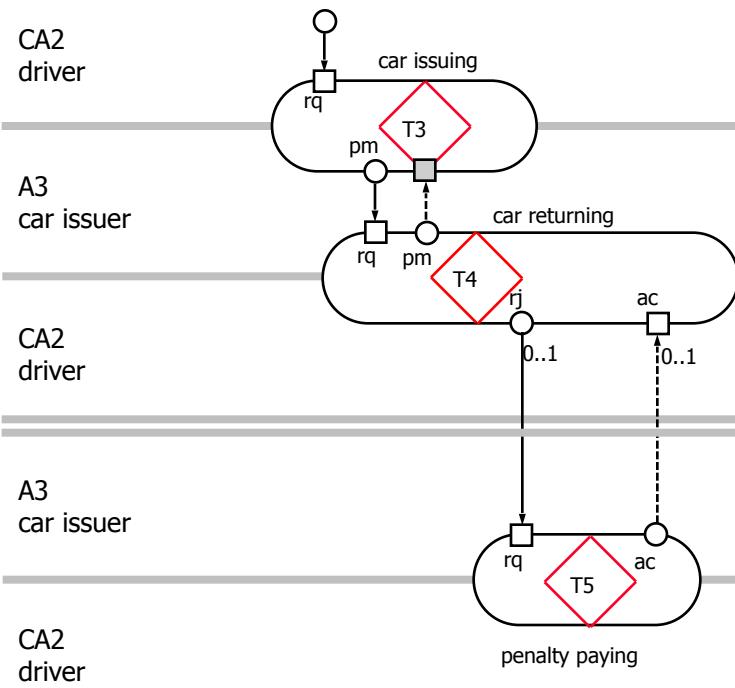


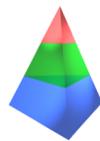
# RAC: PSD (1)





# RAC: PSD (2)





# RAC: TPD for T4

