

The PSI theory - understanding human collaboration

Jan L.G. Dietz

Czech Technical University in Prague, Delft University of Technology, Antwerp Management School

jan.dietz@fit.cvut.cz, j.l.g.dietz@tudelft.nl, jan.dietz@ams.ac.be

Abstract

The PSI theory is a theory about human collaboration in enterprises. PSI stands for Performance in Social Interaction. Based on the CIAO Paradigm (Communication, Information, Action, and Organisation), a communication-centric view is taken on the collaboration of people in enterprises, as manifested in business processes. The fundamental notion in understanding the operation of organisations, is the coordination act/fact. It consists of a performer, an addressee, an intention and a product. Coordination acts/facts are the key elements in business conversations, which are the constituting parts of business transactions. They are performed verbally, non-verbally or tacitly. A transaction involves two actors, one as the initiator and one as the executor. An actor is a subject (social individual) filling an actor role. The executor brings about the product of the transaction to the benefit of the initiator. The process of a transaction is a path, possibly including iterations, through a universal transaction pattern, which consists of one main pattern and four revocation patterns. The latter serve to revert the status in the main pattern to a previous status. Because of the inherent connection between an actor role and the transaction kind of which is the executor, the combination is named transactor role. Transactor roles are the universal building blocks of business processes. Two time aspects of coordination facts are distinguished: the event time and the settlement time. Regarding products (which consist of one independent production fact and a number of dependent facts), the notion of production time (i.e., the time at which the product becomes effective) is distinguished, next to event time. A fundamental principle in the PSI theory is that actors act autonomously, although they may be guided by business rules. Based on this principle, precise definitions are developed for the notions of authority, responsibility, accountability and competence.

Keywords: communicative act, coordination act, production act, business conversation, business transaction, business process, actor, transaction, transactor.

Every organised 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)

1 Introduction

The statement by Henry Mintzberg, quoted above [1], comprises the core problem of every enterprise¹: how to divide the total amount of work and to assign the resulting parts to (competent) workers and, subsequently, how to arrange the necessary coordination among them. The quote also hints already to the prospect that it could be possible to understand the notion of organisation in a very general way, independent of the particular kind of enterprise and independent of the particular workers. But how exactly should one conceive work and workers, so that a universal notion of organisation emerges that is at the same time uncommonly useful, particularly in current practice, where the pervasiveness of ICT applications blurs the sight on the ‘real’ organisation? Achieving this goal has been the ambition of a long-lasting research activity that started in the early nineties of the past century, and that resulted in the present *PSI theory* (or Ψ -theory; the Greek letter Ψ is pronounced as PSI, standing for Performance in Social Interaction), also called the *EE organisational operation theory*. It is a theory about human collaboration in enterprises. It is classified as an *ontological* theory in the framework of theories, as presented in [TEE-00], thus a theory that is about the essence of things.

In concordance with the TAO theory [TEE-02], the PSI theory exclusively takes the construction perspective on enterprises, so disregarding all functional aspects. In taking the construction perspective, we will consistently use the term “organisation” instead of “enterprise”. So, we will speak e.g., of the operation of organisations. In concordance with the DELTA theory [TEE-04], every organisation is conceived as a discrete event system, in the category of social systems [2]. By the construction of an organisation is understood the triple of its composition, its environment, and its structure, as defined in the DELTA theory. The elements in the composition and the environment are social individuals, commonly called subjects, and the structure of an organisation consists of several kinds of mutual influencing bonds between these subjects, as explained by the OMEGA theory [TEE-09]. The operation of an organisation is simply defined as the manifestation of its construction in the course of time.

Section 2 (foundations) starts with the presentation of the CIAO Paradigm. As a consequence, communication is the primal notion in understanding organisation, or, more poetically: communication is the thread of which organisation is woven [3]. Communicating is (also) acting, or as Austin [4] puts it: people do things with words. In the PSI theory, this is accentuated by distinguishing coordination acts and production acts, and by bringing them together in the concepts of business conversations and business transaction. In addition, the process of performing a coordination act is discussed. In section 3 (elaborations), several time aspects of transactions are high-

lighted, followed by the discussion of the operating cycle of actors. Next, attention is paid to human qualities in their collaboration, like authority, responsibility and competence. In section 4 (discussions), we discuss the evolution of the transaction pattern since its introduction in 2006.

2 Foundations

2.1 The CIAO Paradigm

2.1.1 Information-centric versus communication centric

Up to about 1975, the term “information systems” was not used, and there was no field of information systems engineering. The application of programmed computers in enterprises, for the sake of assisting both workers and managers, was called EDP (Electronic Data Processing). Around 1975, EDP was replaced by ISE (Information Systems Engineering) or by like names, referring to the field that concerns the application of ICT² in organisations. The primal and core notion became information, generally defined as the representation of knowledge. Subsequently, communication was defined as the exchange of information. The notion of action was something rather disconnected from information and communication, as was the notion of organisation, although there was the general recognition that organisation somehow implies action, communication, and information. Let us call this point of view the *information-centric* view on information systems (engineering).

One of the effects of the information-centric view is that developing (automated) information systems is understood as something that ICT professionals do ‘to the side’, after having elicited requirements from the people in the organisation, basically by interviewing these people. Once the system is built, it is ‘implanted’ in the organisation. A widely acknowledged drawback of this approach is that the delivered systems rarely meet the expectations of the users. In hindsight, the main reason for this failure is that requirements determination was ill-understood. Interviewing the members of an organisation for determining information requirements, is pretty much like asking the parts of a machine what information they need in order to properly operate and cooperate. Every embedded software engineer would rather start with getting an appropriate understanding of the machine for which he/she is going to build supporting software. Based on this understanding, he/she will develop the requirements. What will be elucidated by the PSI theory (and the OMEGA theory [TEE-09]), is that the parts can’t tell you, even if they were able to speak. As a consequence of this approach to requirements determination, relevant requirements are often missing, and irrelevant ones are included. For obscure reasons, the developers of ‘embedded’ software for organisations, have never felt the need to acquire an appropriate understanding of organisation. Instead, they rely, falsely, on what the ‘parts of the machine’, thus the people in the organisation, tell them.

In the nineties of the past century, an urgent need was felt in several research communities, that the information-centric view was not sustainable anymore. The

number and the size of failures in information systems engineering kept increasing, and the proclaimed benefits of standard packages, notably ERP systems, came along with the feeling of being armoured by the people that had to use these systems. Based on the achievements in language philosophy, notably Speech Act Theory [4], [5] and in (social) action theory, notably the Theory of Communicative Action [6], a community of researchers in information systems engineering, called LAP (Language/Action Perspective), proposed a paradigm shift [7]. By taking communication as the primal and core notion, the path was paved to a more appropriate and more integrated understanding of the other three: information, action, and organisation. Later, the name CIAO Paradigm has been coined for this *communication-centric* view on information systems (engineering). The acronym CIAO stands for Communication, Information, Action and Organisation. *Communication*³ is now defined as the sharing of thoughts between subjects (human beings), and *information* as the means for communication. So, the starting point is that people, in organisations and in society at large, have a need to communicate. Because it is impossible to do this directly, they have to use the vehicle of information. As explained and articulated by the ALPHA theory [TEE-08], every information system can appropriately be conceived as just some implementation (thereby possibly applying ICT) of a part of the organisation.

2.1.2 The notion of communicative act

In addition, communication became (also) understood as a form of *action*, by virtue of the intention that is present in every instance of communication. In Habermas' Theory of Communicative Action [6], which is a translation of [8], an instance of communication is called a *communicative act*. It consists of these four parts: performer, intention, addressee, and proposition, as exhibited in figure 1.

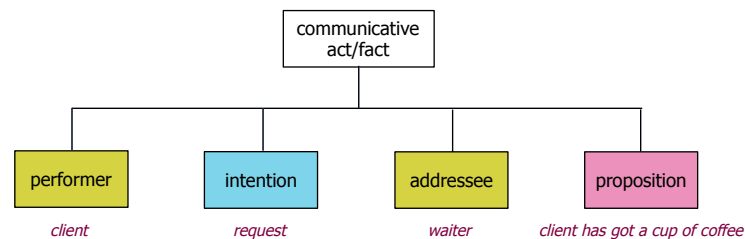


Figure 1 The structure of a communicative act/fact

The *performer* and the *addressee* are subjects, so human beings, particularly in their quality of social individual, which means being capable to engage in mutual commitments. The *proposition* is a state of affairs that is or can be the case. An example of a proposition in the context of a café is that a client has got a cup of coffee. The *intention* is the intent of the performer (the client in figure 1) towards the addressee (a waiter), with respect to the proposition. If the intention is 'request', the performer wants the addressee to make the proposition become true. In the café case, the client wants the waiter to bring her a cup of coffee. Habermas [6] tells us that, in

performing a communicative act, the performer raises three validity claims towards the addressee: the claim to rightness (German: *Richtigkeit*), the claim to sincerity or truthfulness (German: *Wahrhaftigkeit*), and the claim to truth (German: *Wahrheit*). These claims have to be assessed by the addressee, and the result of this assessment will guide him/her in the way he/she will respond. By accepting the *claim to rightness* in the above example, the waiter recognises the moral right or authority of the client to make the request. By accepting the *claim to sincerity*, the waiter expresses that he considers the client sincere in making the request. By accepting the *claim to truth*, the waiter expresses that the proposition can be made true. If all three claims are accepted, the communicative act is said to be successful. In the example, the waiter will then respond by a promise. In case of failure, the waiter will decline the request.

In every communicative act, one of the validity claims is dominant. The other two, however, are also present. Based on this dominance, Habermas [6] distinguishes three categories of communicative acts and three worlds in which these acts have effect. Figure 2 exhibits the distinctions. The dominance is indicated by the rectangles, coloured in correspondence with the related worlds: magenta for the objective world, cyan for the intersubjective world, and ochre for the subjective world.

	constativa	regulativa	expressiva	
objective world				claim to truth
intersubjective or social world				claim to rightness
subjective world				claim to sincerity
	question assertion ...	request promise ...	praise apology ...	

Figure 2 The dominant validity claims and the categories of communicative acts

In the category of *constativa*, the dominant claim is the claim to truth, and the world with which they are primarily concerned, is called the *objective world*. Examples of intentions in this category are question and assertion. If a railway passenger asks a railway officer for the departure time of the next train to Amsterdam, the dominant claim is the claim to truth, i.e. that the (objective) fact exists. This holds also for the answer by the officer (which would be the assertion of the fact). Facts like the departure time of trains are considered to exist in our common objective world, like the fact that the sun is shining, and that the current price of a glass of beer in your favourite pub is € 2,10. As said, the other two (non-dominant) validity claims must also be satisfied. In the train example, this means that the railway passenger respectively trusts the railway officer that he/she will provide the correct answer, and that this officer is authorised to provide the answer.

In the category of *expressiva*, the dominant claim is the claim to sincerity, and the world with which they are primarily concerned, is everyone's private *subjective*

world. Examples of intentions in this category are praise and apology. If the railway passenger starts his/her conversation with the railway officer by saying “I’m sorry to disturb you, madam, but ...”, then the dominant claim of this phrase is the claim to sincerity. If the officer feels that the passenger is insincere, she will most likely utter a sincerity checking sentence, and she may even ignore the question. Facts like feeling sorry are considered to exist in everyone’s subjective world. The claim to sincerity represents the most fundamental condition for human collaboration in the broadest sense of the word, which is mutual *trust*. At the same time, it is the hardest one to assess. Moreover, trust emerges from shared values and norms among people, and they constantly change, as we know. In language philosophy [9] and social action theory [6] it is assumed that people constantly check and adjust their values and norms when they are communicating. In [3] we have called it second order communication, and we have suggested that this is the lubricating oil of organisations and of society at large.

In the category of *regulativa*, the dominant claim is the claim to rightness, and the world with which they are primarily concerned, is the *intersubjective* or *social world*. Examples of intentions in this category are the request and the promise. If the client in the café asks the waiter for a cup of coffee, the dominant claim is the claim to rightness, i.e. the client claims that she has the authority to make the request, and that she considers the waiter to be authorised for his role. This holds also for the response by the waiter, which is either a promise or a decline, as will be elaborated later. Facts like being authorised to do something are considered to exist in our common intersubjective or social world. Moreover, we have created them ourselves. Assigning each other authorities (and expecting that they will be exerted in a responsible way) is the way in which we build up our organisations, as well as our (highly institutionalised) societies. Of course, the other two validity claims must also be met. In the café example, this means that the client assumes that the waiter is able to provide her a cup of coffee, and that she trusts that the waiter is sincere in his promise to do so.

2.1.3 Communication - Information - Action - Organisation

Let us point out next what the consequences of the communication-centric view are for the other three concepts: information, action, and organisation, starting from the basic understanding that *communication* is the sharing of thoughts between human minds. Because people are not able to directly connect their minds, as said before, some vehicle for transmitting thoughts is needed, and this vehicle is information, or the sign, the preferred term in semiotics, which is the branch of philosophy that studies signs (cf FI theory [TEE-01]). A major outcome of this study is the semiotic ladder, exhibited in figure 3. It clarifies the role of signs in the communication of human beings. A unit of *information*, commonly called an information item, is the dyad of content and form, meaning that the two parts are distinguishable but not separable. The content of an information item is the *thought* that one wants to share, and the form is the agreed upon perceivable shapes (which are ‘inscribed’ in some substance), collectively called the *sign*. The content comprises both the intention (or pragmatics)

and the proposition (or semantics) of the thought, and the form comprises both the formalism (or syntax) and the coding (or empirics) of the sign.

In the café example, the client has, at some point in time, got the thought that she wants a cup of coffee. In order to share this thought, she has to express it in a sign that is intelligible for the waiter. The proposition of the thought is “client has got a cup of coffee” and the intention is the request. By performing the request, she enters into a commitment towards the waiter, like the waiter enters into a commitment if he performs a promise or a decline in response. The client may have expressed her thought in this English sentence: “I’d like to have a cup of coffee, please”, which constitutes the form part in figure 3. The applied formalism is the English grammar and the coding regards the construction of the words. The substance in which the sentence is inscribed consists of the air vibrations that are produced by the client and then perceived by the sense of hearing of the waiter.

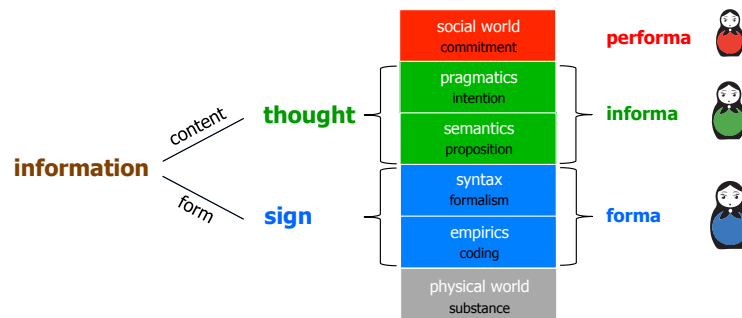


Figure 3 The semiotic ladder

For the concept of *action*, the communication-centric view means that communicating is also acting. As Austin [4] puts it: people do things with words. In the PSI theory, this is accentuated by distinguishing coordination acts and production acts (to be discussed in section 2.2), and by combining them in the concept of the business transaction (to be discussed in section 2.5). Consequently, the word “in” in the reading of the name “PSI” (Performance in Social Interaction), has a twofold meaning. First, it means that coordination acts, like requesting a cup of coffee, are brought about in communication, and thus in social interaction. Second, it means that production, like delivering a cup of coffee, is brought about in social interaction, namely in transactions, which are patterns of coordination acts, concerning one production act. The important consequence of it is that the two actors in a transaction may e.g. agree on the being delivered a cup of coffee, whereas this is not the case.

The (new) concept of *organisation* refers primarily to the network of actor roles and corresponding transaction kinds that emerges as the effect of identifying tasks, and the need for coordinating them [1]. The accentuation that the PSI theory adds to it, is that an actor role, together with the transaction kind of which it is the executor, constitutes a universal organisational building block (cf section 2.7).

2.2 Work is production plus coordination

2.2.1 Production acts and facts

All acts, in all systems, of all categories, can be divided in two sorts: production acts and coordination acts, as postulated by the DELTA theory [TEE-04]. By performing *production acts* (or P-acts for short), the subjects in an organisation create products. A production act can be material (like transporting goods) or immaterial (like becoming member of a library). The effect of performing a P-act is the creation of the corresponding P-fact. We distinguish between independent and dependent P-facts. Examples of independent P-facts are “membership 387 is started”, “rental 1087 is contracted”, “the car of rental 1087 is issued” and “sale 1618 is completed”. Every independent P-fact may have a number of dependent P-facts. They are *dependent* because they start to exist (come into being) as a consequence of, and together with, the related *independent* P-fact. Independent P-facts are mostly unary facts concerning some entity, whereas dependent P-facts are always binary facts (cf FI theory [TEE-01]). Dependent P-facts are conceived as mappings from one class (the domain) to another class (the range). These mappings constitute the *properties* of the entity. The combination of an independent P-fact and all of its related dependent P-facts (properties) is called a *product*. As an example, the independent P-fact ‘membership 387 is started’ is an instance of the P-fact type ‘[membership] is started’, which is a logical predication of the members of the class MEMBERSHIP. The term “[membership]” is a placeholder, or variable, that can be instantiated; membership 387 is one of the instances. Possible properties of membership 387 are that the concerned member is John, and that the starting date is 1 April 2016. Moreover, the *entity* membership 387 is unique. If the same person, after having resigned, wants to become member of the library again, another membership will be created, with the same person as member.

The formulation of the independent P-fact in a product as a fact that is uniquely identifiable in space and time, is crucial. By space is meant the state space of the production world (cf DELTA theory [TEE-04]). For example, if someone becomes several times member of a tennis club during his/her lifetime, there is only one way to deal with this adequately, which is to conceive the notion of membership whose instances are uniquely identifiable in space and time. As we have seen, membership 387 is a uniquely identified entity. The person who is the member of this membership, may in the course of time, and even at the same time, also be the member in other memberships. Note that a phrase like “membership 387 is started” is not the expression of the assertion that the proposition is true (cf figure 1). Instead, it is only the formulation of a proposition or, in terms of the PSI theory, of an independent P-fact. As will become clear later, the proposition may eventually become true, i.e. the P-fact may become existent, as the result of a successfully carried out transaction.

2.2.2 Coordination acts and facts

Coordination acts (or C-acts for short) are a special sort of communicative acts. With reference to figure 2, coordination acts belong to Habermas’ category of regulativa. The generic structure of a C-act is exhibited in figure 4, which is a specialisation and

an extension of figure 1. The illustrating example is taken from the case wheelbarrows, as discussed in [3]. The performer of the act is the specific Gnome 387, in his role of buyer on behalf of the company HORTUS, and the addressee is Gnome 691, in his role of seller on behalf of the company MALUM. The product is the purchase of a number of wheelbarrows. This is represented by the independent P-fact ‘purchase 31416 is fulfilled’ (split into the predication “is fulfilled” and the predicated entity “purchase 31416”), and the dependent P-facts or properties ‘article type is Quadra 75’, ‘number of items is 10’, ‘price is 165 (in some currency)’, and ‘delivery day is 731.513 (in Julian days)’. The intention of the C-act/fact is the request. It represents the intersubjective or social stance that Gnome 387, in his role of buyer, takes towards Gnome 691, in his role of seller. By conveying this intention to Gnome 691, Gnome 387 commits himself to his request, which means that he cannot simply say at some later point in time that he was just joking. Similarly, Gnome 691 will become committed to the response that he is going to perform. As discussed in section 2.1, we consider C-acts to be the atomic elements of business processes. So, although performing a C-act includes doing various things, as will be discussed in section 2.3, it is ontologically indivisible: one performs a ‘complete’ request or nothing.

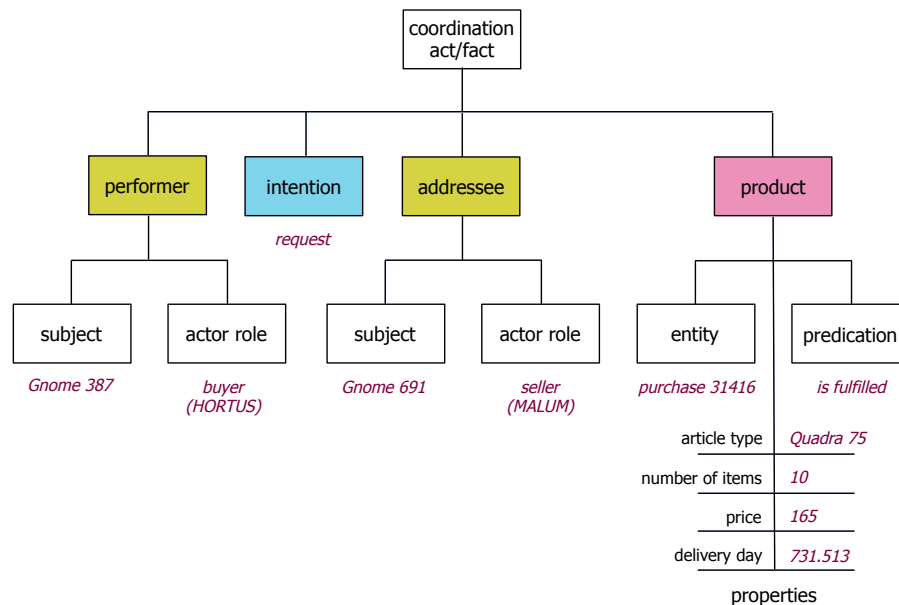


Figure 4 The structure of a coordination act/fact

2.2.3 States and events

The DELTA theory [TEE-04] states that with every system, a world is associated in which the acts of the system have effect. More precisely, every act (by the system) causes the creation of a fact (in its world). The state of the world of a system at some point in time consists of all facts that are created up to that point in time. The becom-

ing existent of a fact is called an *event*, which is a change of *state*, or *transition* of the world, and the point in time at which the event occurs, is called its *event time*.

The C-acts and P-acts in an organisation cause events in respectively the coordination world or C-world, and the production world or P-world. This is illustrated in figure 5. The two subjects in the middle interact through the creation of C-facts (represented by small disks), which are the results of performing C-acts (which concern a P-fact). The shapes of the P-acts and P-facts are coloured light-grey to indicate that they are not directly knowable, to be explained in section 2.6. In addition, the number of C-facts is much larger than the number of P-facts, to be explained also in section 2.6.

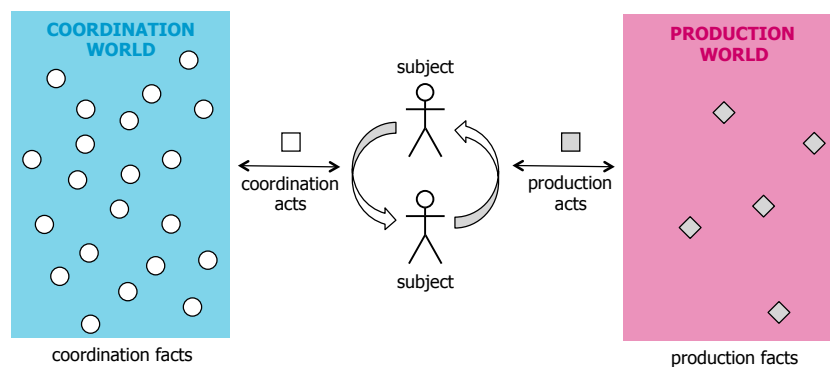


Figure 5 The coordination world and the production world of an organisation

2.3 The process of a coordination act

Corresponding with the semiotic ladder (cf figure 3), we distinguish three abilities that communicating subjects need to dispose of: the *forma* ability (in order to deal with codings and formalisms), the *informa* ability (in order to deal with propositions and intentions), and the *performa* ability (in order to deal with commitments). These abilities are shown, on the right side of figure 3, as three shapes that human beings can take on. Note that the physical world (dealing with substances) is not covered by the performa-informa-forma distinction. If needed, we will consider it to be included in the forma ability. For the sake of completeness, however, the physical (grey) shape is included in the elaborated explanation of the process of a coordination act in figure 6, next to the blank shape, which represents the pure and genuine self of every subject. There reside the wisdom and love that are considered to constitute the basis for deciding on how to respond to a conveyed commitment, and thus for all our social acts. The Matryoshka doll shapes in this figure are scaled, to illustrate that they should be understood as being enclosed in each other, from top to bottom.

In order to effectuate a decision to perform a C-act, a subject has to expose the corresponding commitment in its 'red' shape, i.e. by applying its performa ability. Because it is impossible, as we have seen in section 2.1, to convey the commitment directly to the addressee, he/she has to formulate, in his/her 'green' shape, a thought

that comprises the commitment. As we know from section 2.1, notably figure 3, the thought consists of a proposition and an intention. As it is impossible to share the thought directly with the addressee, the subject has to utter, in his/her 'blue' shape, a sentence that expresses the thought. In order to make the sentence perceivable to the addressee, he/she has to inscribe, in its 'grey' shape, the sentence in some substance, and have it transmitted through a proper channel to the addressee. This action succeeds if the message arrives undistorted. If so, we say that the two subjects have achieved *physical correspondence*, or that the *medium condition* is satisfied. The addressee is now able, in his/her 'blue' shape, to perceive the inscribed sentence from the transmitted substance. If he/she succeeds, we say that the two subjects have achieved *notational correspondence*, or that the *forma condition* is satisfied.

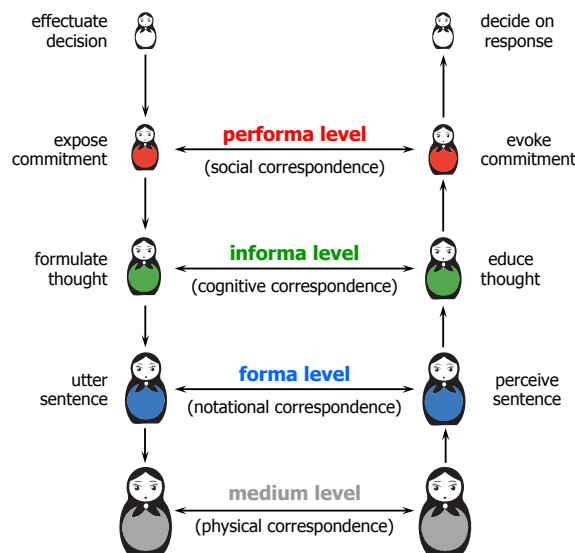


Figure 6 The process of a coordination act

Next, the addressee can educe, in his/her 'green' shape, the thought from the sentence, so the contained proposition and intention. If he/she succeeds, we say that the two subjects have achieved *cognitive correspondence*, or that the *informa condition* is satisfied. Then, in his/her 'red' shape, the addressee has to evoke in his/her mind the commitment that the performer of the C-act wanted to convey. If he/she succeeds, we say that the two subjects have achieved *social correspondence*, or that the *performa condition* is satisfied. Lastly, the addressee has to decide on how he/she will respond in his/her blank shape. Figure 6 clearly illustrates that communicating is a complex and laborious activity. Fortunately, people are experts at it, but it is good to be aware of the four conditions that must always be met in order to perform a C-act.

2.4 Business conversations

Communicative acts occur in sequences, called *conversations*, which are the objects of interest in conversation theory [10] [11] [12]. In the PSI theory, the focus is on *business conversations*, defined as conversations that takes place in an institutional setting and of which the participants aim at achieving some goal [13]. They are considered to satisfy the requirements of the ideal speech situation⁴. In the course of time, several patterns of conversation have been identified, like the conversation for information and the conversation for action [14]. Likewise, workflow loops are proposed, as extensions of conversations [15]. Based on these studies, we distinguish four kinds of conversations: actagenic conversations, factagenic conversations, reversiogenic conversations and cogitatiogenic conversations⁵. Henceforth, we assume that a conversation involves two participants, but we allow that a participant is the collectivity of a number of subjects (cf section 2.7).

An *actagenic conversation*, or A-conversation for short, is a conversation in which the participants strive to reach consensus about a product that one of them is going to bring about for the benefit of the other. The key C-acts in an A-conversation are the request and the promise. Both must be present for a successful conversation, and in this order, because the promise is a response to the request. However, apart from this order, they may occur at any place in the conversation. An example of an A-conversation in the café situation is given below (where C is the client and W the waiter). The request is expressed in the fourth line, and the corresponding promise in the fifth line.

W: Good morning madam, what a wonderful weather you bring with you.
 C: Yes indeed, it is beautiful outside. I think I will go for a walk later today.
 W: A very good idea, I would say, madam. What can I do for you?
 C: I think, I'd like to have a cup of coffee. [request]
 W: I will bring it right away, madam. Anything else? [promise]
 C: No thanks, that's all.

A *factagenic conversation*, or F-conversation for short, is a conversation in which the participants strive to reach consensus about a P-fact that one of them has produced for the other. The key C-acts in an F-conversation are the declare and the accept act. Both must be present for a successful conversation, and in this order, because the acceptance is a response to the declaration. Apart from this order, they may occur at any place in the conversation. F-conversations typically go together with A-conversations in the larger frame of transactions, to be discussed in section 2.5. An example of an F-conversation in the café situation that matches the A-conversation above, could be:

W: Here you are, madam, a fresh cup of coffee, the best in town! [declare]
 C: Ha ha, I hope so, sir. What makes you think it is the best in town?
 W: Just gut feeling, madam. Anyway, I know no better one!
 C: Well, let me see whether I can agree with you. [accept]
 W: I'll bet you will!

A *reversiogenic conversation*, or R-conversation for short, is a conversation in which the participants strive to agree on reverting (turning back) the current status in an A- or an F-conversation, in which they (also) participate. As will be shown in sections 2.6.4 thru 2.6.7, it is even possible to revert a whole transaction. The key C-acts in a successful R-conversation are the revoke, followed by the allow. An example of an R-conversation in the café situation, right after the A-conversation above, when the waiter is already on his way to get the coffee, could be:

C: Oh, waiter, please, on second thoughts ... do you have cappuccino? [revoke]
 W: Sure, madam, and it's no problem at all that you changed your mind. [allow]
 C: Oh, thanks a lot, you are very kind. So, a cappuccino please. [request]
 W: You're welcome, madam. [promise]

The revoke is expressed in the first line, and the corresponding allow in the second one. The third and fourth line contain the new request and the corresponding promise.

A *cogitatiogenic conversation*, or C-conversation for short, is a conversation in which the participants strive to reach consensus about an idea or plan for future action. C-conversations are typically held in preparation of a decision to perform a C-act. They include conversations that are commonly known as consultations and deliberations, in particular the conversations in the discussion statuses that we will see in section 2.6. Contrary to the other three kinds of conversations, the PSI theory does not contain specific patterns for C-conversations, because C-conversations are not composed of coordination acts but of other communicative acts. However, their existence and their relevance is recognised. An example of a C-conversation in the café situation, preceding the formal part of the A-conversation, could be:

W: Good morning madam, what a wonderful weather you bring with you.
 C: Yes indeed, it is beautiful outside. I think I will go for a walk later today.
 W: A very good idea, I would say, madam. What can I do for you?
 C: I don't know yet, I just came in because I wanted to have something stimulating. I feel a bit groggy, perhaps of the wine last night.
 W: Then I can recommend a cup of coffee or, if you don't feel for it, a glass of fresh mint tea.
 C: Hmm, well, sounds like a good idea.

As said, an A-conversation is successful if the status of being promised is reached. In case of no success, the status of the conversation can be reverted by a successful corresponding R-conversation, as will be discussed in section 2.6. Likewise, an F-conversation is successful if the status of being accepted is reached. In case of no success, the status of the conversation can be reverted by a successful corresponding R-conversation, as will be discussed in section 2.6. An R-conversation is said to be successful if the intended reversion is achieved, i.e. if the status of an A- or an F-conversation is reverted. In case of no success, the status in the other conversation will remain unchanged.

2.5 Business transactions

The conversations that we have discussed in section 2.4, commonly occur in larger units, which are called *transactions*⁶. Only C-conversations may occur separately, but commonly it will be in preparation of a transaction. A transaction involves two actors, one in the role of *initiator* (or consumer) and the other in the role of *executor* (or producer). An actor is a subject in the quality of filling an actor role. The notions of actor and actor role are elaborated in section 2.7.

The best general understanding of a transaction is that it proceeds in three phases: the order phase, the execution phase, and the result phase. The *order phase* is an A-conversation in which the two actors discuss and negotiate in order to come to agreement about a *product* that the executor can responsibly promise to bring about for the benefit of the initiator. The properties of the product include the terms of delivery (time, price, etc.) that are common in (business) transactions. In this phase, the product is also called the *proposition*. In the *execution phase*, the executor produces some product (which may differ from the promised one). The *result phase* is an F-conversation in which the two actors discuss and negotiate in order to come to agreement about the product as it can responsibly be accepted by the initiator. In this phase, the product is also called the *result*. While in the order phase, basically all properties of the product are negotiable, some are still also negotiable in the result phase, notably properties like price and delivery time. As an example in the café, the client may not be fully satisfied with the declared product if the coffee is not really warm. Instead of producing a fresh cup of coffee (something that will be discussed in section 2.6), the two actors could agree on a lower price (Note. The payment itself is a separate transaction, as discussed in the OMEGA theory [TEE-09]).

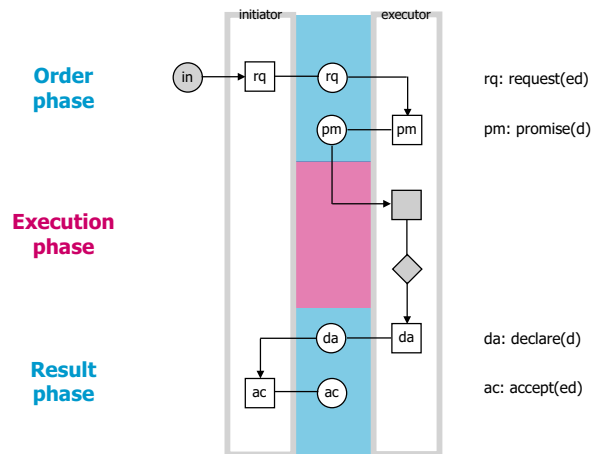


Figure 7 The basic transaction pattern

2.6 Transaction patterns

2.6.1 The basic transaction pattern

In figure 7, the *basic transaction pattern* is shown. It is called basic because it contains the five steps that must always be performed in order to let a transaction succeed: the request and the promise in the A-conversation (order phase), the declaration and the acceptance in the F-conversation (result phase), and the execution of the P-act in between them (execution phase). The symbols used are explained in table 1. The colours cyan and magenta indicate that the acts in the corresponding phase have effect respectively in the C-world and the P-world (cf figure 5).

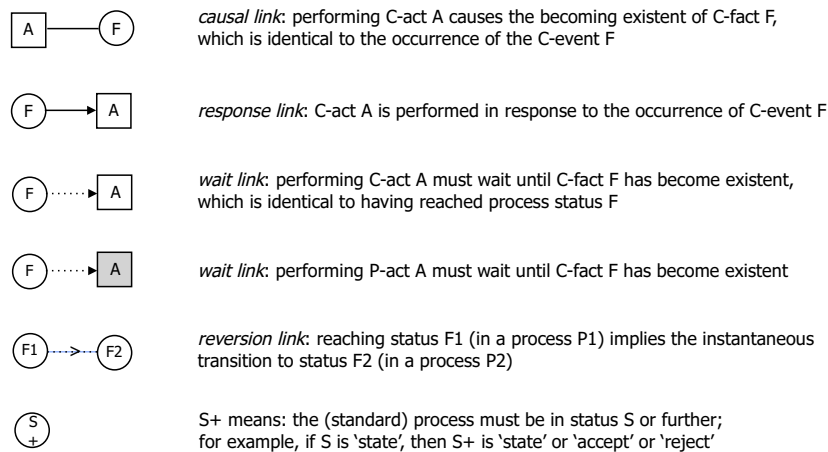


Table 1 Legend of the links in transaction patterns

In the café example, the first step is the request by the client (the initiator) for a cup of coffee, addressed to the waiter (the executor). We have entered the order phase of the transaction. The second step is the promise by the waiter to bring the cup of coffee. With this step, the order phase ends successfully, and the execution phase starts, in which the waiter produces the cup of coffee. The shapes of the P-act and the P-fact are coloured grey to indicate that they are purely private to the executor, and thus not knowable to the initiator (and anyone else). When this third step is done, the waiter addresses himself to the client again and declares the result of his work, which is the fourth step. We have entered now the result phase of the transaction. This phase ends successfully if the client accepts the declared result, which is the fifth transaction step.

The two light-grey lined rectangles in figure 7, indicate the *responsibility areas* of the two participants: the initiator is responsible for the request and the acceptance, and the executor for the promise, the P-act and the declaration. The notion of responsibility will be elaborated in section 3.3. The conversation statuses are put in between these rectangles to indicate that they are knowable to both actors. The initial status “in” is coloured grey, and is put to the left, because it is usually external to the transaction. The only exception is the case of self-activation (cf section 2.7).

There are two important notes worth making. The first one is that we abstract completely from the particular way in which the steps are performed. The abstraction from the implementation of C-acts is one of the key elements in considering the understanding of organisations that the PSI theory provides *ontological*. The other key element is the being rooted of this understanding in the atomicity of C-acts/facts, and in the molecularity of transactions.

C-acts may be performed *verbally*, e.g. by uttering sentences like “I’d like to have a cup of coffee, please” as a way to request, but they may also be performed *non-verbally*, which means that some other act counts as the C-act (like just putting the cup of coffee in front of the client as a way to perform the declaration). In both cases, the evidence of the act is *explicit*. C-acts may also be performed *tacitly*, which means that there are no acts that could count as performing them. Still they are performed, but the evidence is *implicit*, either from the presence or from the absence of other acts [13]. To illustrate this, if the waiter doesn’t perform an explicit promise, the client may deduce it from the absence of an explicit decline (cf. section 2.6.2). Next, when the waiter brings the cup of coffee and thus performs the declare act, the client may deduce the promise from the presence of this act. As one may expect, tacitly performed C-acts are a major cause of the class of business process failures where an actor is waiting for an explicit C-event that will never occur, however.

The second note regards the becoming existent of the product of a transaction and the possible difference between the requested product and the accepted one. The product of a transaction, i.e. the independent P-fact together with its dependent P-facts (cf FI theory [TEE-01]), can only become existent after a successful completion of the result phase. In this way, the concept of action is firmly connected to the concept of information: the resulting product is a fact, if and when it is accepted by the initiator of the transaction. Consequently, every P-fact is the result of a successfully completed transaction, or it is derived from such original facts.

Note that the initiator may accept a result that differs considerably from the requested product. Let us take the café example again to illustrate this. Suppose that the client has asked for a double espresso and the waiter delivers a cappuccino. Most people would reject the declare act by the waiter in such a case, but it is perfectly fine if the client accepts it. The example emphasises that the most important world for human beings, is Habermas’ intersubjective or social world (cf figure 2). To top that, the client may even accept the declaration if no coffee has been brought to him/her at all! This basic understanding of the ‘nature’ of P-facts reflects our basic understanding of the ‘nature’ of societal institutions, namely that they are primarily intersubjective or social constructs, in accordance with the core message in [9].

The carrying out of a transaction (of whatever kind) can be taken as a generic *business process* building block, and the duties of the executor of the transaction can be taken as a generic definition of the notion of *task*, which is a presumably more precise definition than the one that is implicitly applied in Mintzberg’s quote above the introduction (section 1).

2.6.2 The standard transaction pattern

Figure 8 exhibits an extension of the basic pattern. Note that the diamond of the P-fact is drawn in the box of the P-act (to save space) and that the phase colours are omitted. The basic transaction process, also called the happy flow, is represented by the green path. As discussed in section 2.1, the performer of a C-act raises three *validity claims* towards the addressee: the claim to rightness, the claim to sincerity, and the claim to truth. All three of them have to be accepted by the addressee in order to make the C-act successful. If so, the executor may respond by a promise. If not, the executor will decline the request, which brings the transaction process in the status declined. This is indicated in figure 8 by the yellow path from (rq)⁷, via [dc], to (dc).

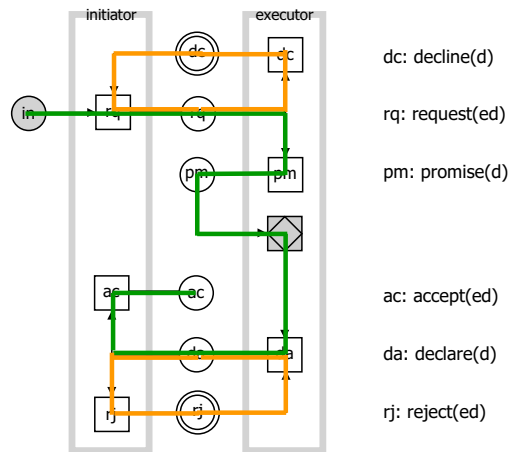


Figure 8 The standard transaction pattern (1)

The decline status is represented by a double disk to indicate that it is a discussion status. The executor gets the opportunity now to explain which validity claim(s) he/she could not accept, and the initiator gets the opportunity to refute the objectives of the executor and/or to discuss possible changes in the properties of the product. In the café example, the waiter could have declined the request for a cup of coffee because the coffee machine is broken down (claim to truth), or because the closing time has passed (claim to rightness), or because the client repeatedly has revoked his/her request without good reasons (claim to sincerity). The result of the discussion can be that the initiator stands by his/her request or that he/she performs an adapted request, like ordering tea instead of coffee, or waiting for the being fixed of the coffee machine (in which case only the delivery time of the coffee changes). This is indicated in figure 8 by the yellow path from (dc), via [rq], to (rq). If so, the executor can then perform the promise of the new request (green path).

Instead of accepting the declared product, the initiator may reject it, which brings the transaction process in the status rejected. This is indicated in figure 8 by the yellow path from (da), via [rj], to (rj). In this discussion status, the initiator gets the opportunity to explain which validity claim(s) he/she could not accept, and the executor

gets the opportunity to convince the initiator that the declared product is what was promised, so that it can still be accepted. In the café example, the client could have rejected the declaration (of a cup of coffee) by the waiter because she thinks that the coffee is not fresh or warm enough (claim to truth), or because an unidentified person brings the coffee (claim to rightness), or because she has waited for a long time for the coffee after having reminded the waiter several times (claim to sincerity). Likewise, as said earlier, the two may come to agree for example on a lower price for the client. Therefore, the result of the discussion can be that the executor stands by his/her declare act or that he/she performs an adapted declare. This is indicated in figure 8 by the yellow path from (rj), via [da], to (da). If so, the initiator can perform the acceptance (green path).

But what happens if the two participants do not succeed in getting out of the discussion statuses? The pattern in figure 8 offers no options to escape from such a deadlock situation. At the same time, staying for ever in a deadlock situation is not a satisfactory solution. In sections 2.6.3 thru 2.6.7, reversiogenic (or revocation) conversations (R-conversations) are presented, which allow to revert the main process to some previous status. Two of them, namely the revocation of the request, and the revocation of the declaration, can be used to solve the deadlock situations that may arise in the pattern of figure 8. Let us therefore add these R-conversations, such that the initiator gets the option to ‘undo’ his/her request from the status (dc), and that the executor gets the option to ‘redo’ the P-act and the subsequent declare act from the status (rj). The result is shown in figure 9. It contains two R-conversation patterns, one for revoking a request (left side) and one for revoking a declaration (right side).

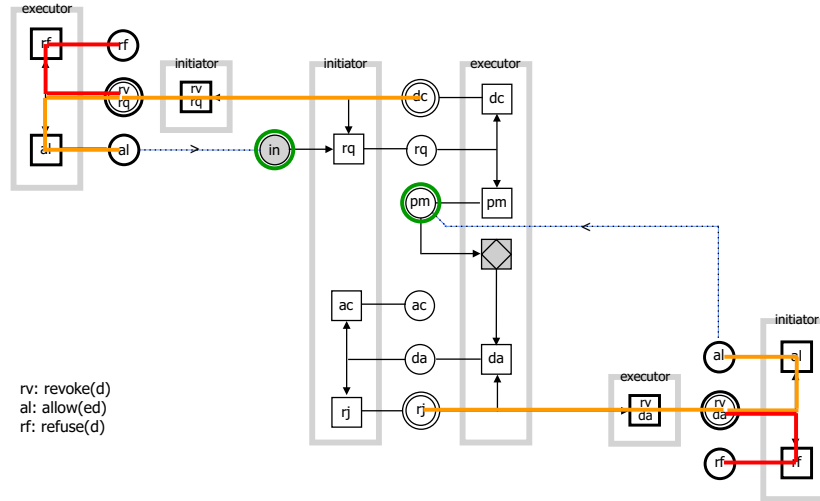


Figure 9 The standard transaction pattern (2)

Let us use the café example again to illustrate revocations. If the discussion between the client and the waiter in the status (dc) does not lead to keeping the current request or performing an adapted request, so that the waiter can promise to pro-

duce it, she has the option to start an R-conversation, in which she strives to turn the status in the main process back to the initial status (in), and thus to ‘undo’ her request. She does so by performing [rv rq], which brings the entered R-conversation in the status (rv rq), indicated in figure 9 by the yellow path from (dc), via [rv rq], to (rv rq). If the waiter allows the revoke, which he normally will do, he performs [al[rv rq]], which leads to the status (al[rv rq]), indicated by the yellow path from ((rv rq) to (al). The reversion link (cf table 1) from (al) to (in) expresses that the status of the main process will be reverted to the initial status (in). The social meaning of this return is that both actors are discharged from all commitments in the main process. However, if the waiter refuses the revocation, so if he performs [rf[rv rq]], by which the status (rf[rv rq]) is reached, indicated in figure 9 by the red path from (rv rq), via [rf], to (rf), the main process remains in the status (dc). The participants have now ended up in a situation of deadlock that one normally doesn’t want to be in. Apparently, the other party, in this case the waiter, does not strive to consensus, which is the fundamental assumption in Habermas’ theory [6]. Now the need has come to leave the *discussion layer* of the conversation and step over to the *discourse layer* [16]. On this layer, people investigate, challenge and discuss the values and norms that they apply in their social interaction. In our café example, the client can now challenge the waiter to provide an explanation of his ‘asocial’ behaviour.

Likewise, the executor may revoke his/her declaration, in order to escape from a deadlock situation in the status (rj). If successful, the main process will be reverted to the status (pm), from which the executor can redo the P-act, followed by the declaration of the new product. In case of failure, the main process remains in the status (rj).

2.6.3 The complete transaction pattern

In addition to the revocations of the request and the declare act, as discussed above, the two other basic C-acts, thus the promise and the acceptance, can also be revoked. Moreover, all four R-conversations can be initiated at any point in time, i.e. regardless the current status in the main transaction process, and also even repeatedly. In other words, both the initiator and the executor can revoke any basic step they have taken, from any status in the main transaction process. Figure 10 exhibits these extensions of the transaction pattern, resulting in the *complete transaction pattern*, in which all four R-conversation patterns are expressed in a similar way: one for the request, one for the promise, one for the declaration, and one for the acceptance. In an R-conversation pattern, the boxes and disks are bold-lined, in order to indicate that these conversations are at a *meta level* with regard to the main conversation or process of which they aim to change the status.

An R-conversation can be initiated in response to any status in any (other) process. This initial status is shown in figure 10 as a C-fact named “?”. In addition, the R-conversations in figure 10 have a wait link (cf table 1) from a status in the main process to the revoke act, meaning that performing the revoke is impeded until this status has been reached. For the act [rv rq], the condition is named “rq+”, meaning that the status of the transaction process must be (rq) or further. To avoid confusion, the response links (dc) -> [rq] and (rj) -> [da] in figure 10, are made optional (indicated by the car-

dinality range 0..1). Likewise, the condition for performing [rv pm] is (pm+), for performing [rv da] it is (da+), and for performing [rv ac] it is (ac). Next, if an R-conversation is unsuccessful, the status of the transaction process remains unchanged, as explained above. This is indicated in figure 10 by reversion links (cf table 1) to the statuses (rq+), (pm+), (da+) and (ac) respectively; actually, nothing changes in that case. If an R-conversation is successful, the status of the main process will be reverted to (in) for a revoke of the request, to (rq) for a revoke of the promise, to (pm) for a revoke of the declaration, and to (da) for a revoke of the acceptance.

Let us have a closer look at the four R-conversations. In order to let the discussions be as general as possible, we will let all of them start from the status (ac) in the main process, so when the transaction is ended successfully. Note that revoking implies that one wants to undo a step that one has taken intentionally before. If a step is taken by mistake, it can be *cancelled*, but only before the addressee has responded. This is the way, for both participants, to correct plain mistakes.

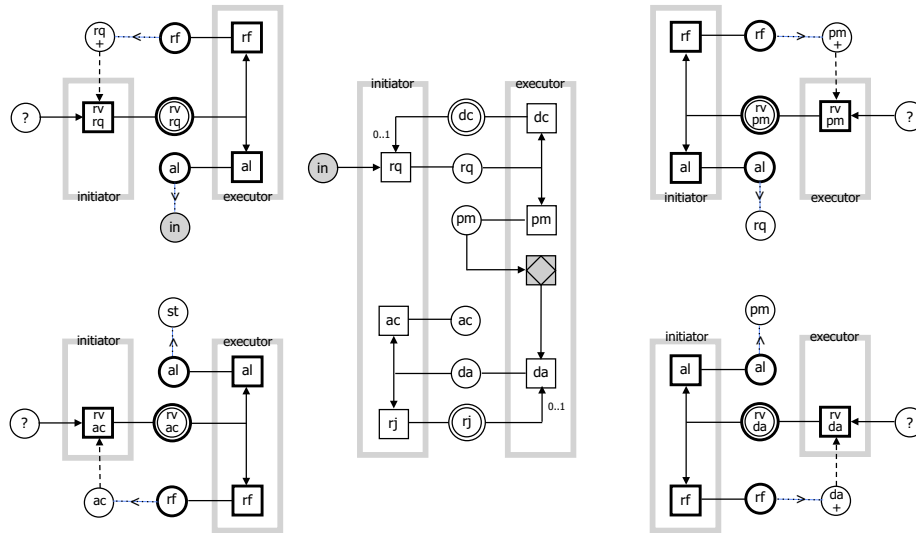


Figure 10 The complete transaction pattern

Concluding, every transaction process is some path (possibly including iterations) through the complete transaction pattern, starting from the status (in) and ending up either successfully in the status (ac) or unsuccessfully in the status (in). Although theoretically, every step in a transaction process should be revocable, the four R-conversations in figure 10 appear to be sufficient in practice. Therefore, the discussed complete transaction pattern is considered to be *universal*.

Revocations have a legal counterpart in the Civil Codes of many nations. In these Civil Codes, a (business) commitment cannot be made undone by a party without the explicit allowance by the other party. This legal requirement is fully satisfied by the revocation patterns as discussed above.

Figure 11 exhibits the process of revoking an accept act. It starts with performing the act [rv ac] by the initiator, resulting in the status (rv ac), indicated in the figure by the yellow path from (?), via [rv ac], to (rv ac). The dashed arrow from (ac) to [rv ac] represents a wait link (cf table 1). It means that the revocation can only be performed if the standard transaction process has reached the status (ac), which is the case. The status (rv ac) is a discussion status (indicated by the double disk), which means that the two actors have to sit together in order to discuss the proposed reversion of the main transaction process. The executor may allow the revoke, indicated by the continued yellow path to (al) or refuse it (the red path). If he/she allows, the main process will return to the status (da), as indicated by the green-white path on the right side of the figure. From there, the initiator is able to perform the reject (indicated by the blue path on the right side of figure 11). As an example in the café, the client may have accepted the delivered cup of coffee, but later on discovers that the coffee is not as warm as it should be. She then may call the waiter again and tell him that she finds this not acceptable, which counts as revoking the accept act. If the waiter allows the revocation, the client can subsequently reject the declaration.

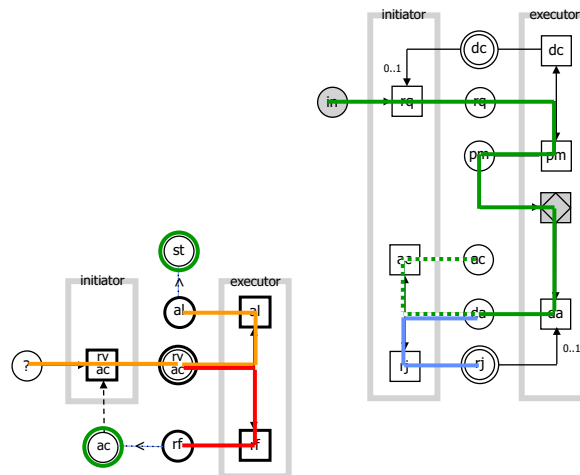


Figure 11 The process of revoking an accept act

The executor of a transaction may at any point in time want to undo the declare act. Logically, this entails that he/she wants to redo the P-act and subsequently perform a new declare act. Revoking the declare act is a usual response by the executor of a transaction in the event that the initiator has rejected the declare act, and the executor agrees, during the discussion in the status rejected, on the reason for the reject. As an example in the café, after the client has rejected the acceptance (because the coffee was cold), the waiter may want to redo the P-act and the subsequent declare act, i.e. he wants to bring a new, warm, cup of coffee.

The pattern for revoking the declare act is exhibited in figure 12. It starts with performing the act [rv da] by the executor, resulting in the status (rv da), indicated by the yellow path from (?) to (rv da). The wait link from (da+) to [rv da] represents the condition that the revocation can only be performed if the status in the main transaction process is at least (da). Thus, it can be performed from any status after, and including (da), even from the status (ac), from which we will start, as shown on the left side of the figure. The status (rv da) is a discussion status (indicated by the double disk), which means that the two actors have to sit together in order to discuss the proposed turning back of the main transaction process. The initiator may allow the revoke (continued yellow path) or refuse it (red path). If he/she allows, the status of the main process will be reverted to (pm), as indicated by the green-white path on the left side of figure 12. From there, the executor is able to redo the P-act, followed by a new declare act, as indicated by the blue line. If the initiator refuses the revocation, the status of the main process remains unchanged, which is the status (ac) in figure 12.

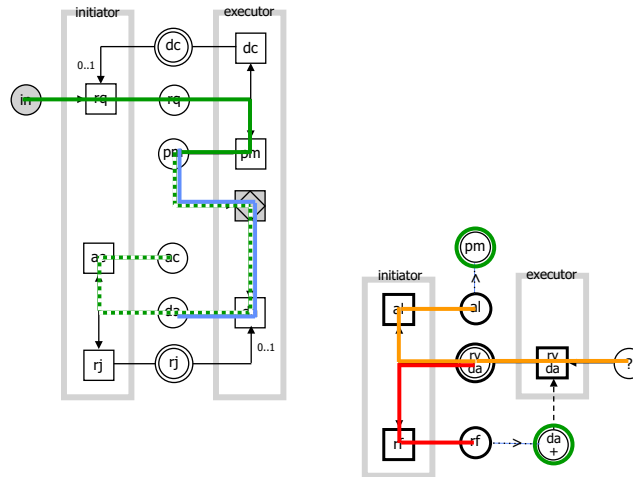


Figure 12 The process of revoking a declare act

2.6.6 The revocation of a promise act

The executor of a transaction may at some point in time want to undo the promise act. This will normally happen if he/she discovers that he/she cannot comply with the promise. As an example in the café, the waiter may have promised the client a cup of coffee, and then discovers that the coffee machine is broken. By revoking the promise, and by the subsequent allowance by the initiator, they can end up in the status declined, from which they can discuss other drinks, like a cup of tea.

The revocation pattern of a promise is exhibited in figure 13. It starts with the revoke promise [rv pm] by the executor, resulting in the status (rv pm), indicated by the yellow path from (?) to (rv pm). The wait link from (pm+) to [rv pm] represents the condition that the revocation can only be performed if the status in the main process is (pm) or further. The being revoked is a discussion status (indicated by the double

disk), which means that the two actors have to sit together in order to discuss the proposed reverting of the main transaction process. The initiator may allow the revoke (continued yellow path) or refuse it (red path). If he/she allows, the status of the main process will be reverted to (rq), as indicated by the green-white path on the left side of figure 13. From there, the executor can decline the request, and discuss changes to the proposition so that he/she will be able to promise it, as indicated by the blue path.

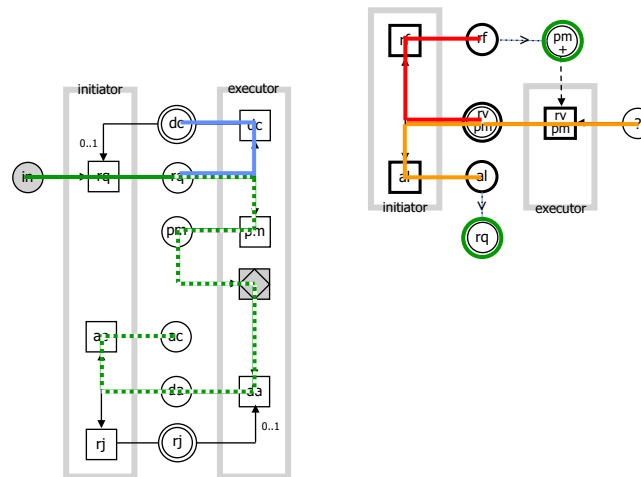


Figure 13 The process of revoking a promise act

If the initiator refuses the revocation, the status of the main process remains unchanged, which is in figure 13 the status (ac). A common example of the revocation of a promise from the status (ac), so when the transaction has been completed successfully, is in a shop situation, where the client has paid for buying something, and where the status (?) represents the initial status of the buying transaction, to which one has returned after a successful revoke request by the client. In order to get his/her money back, the client can revoke the promise (to pay), because she no longer agrees with the proposition that he/she has promised before. After the allow by the other party, the payment transaction is reverted to the status (rq), including the ‘undoing’ of the P-act. From the status (rq), the client then declines the original request (the blue path in figure 13). In the status (dc), the two participants can then discuss a new amount to pay, or the initiator can revoke the request, by which (after allowance by the client) the payment transaction is completely rolled back to the initial status.

2.6.7 The revocation of a request act

Lastly, the initiator of a transaction may at some point in time want to undo the request act. As an example in the café, the client may have asked for a cup of coffee, but then gets an urgent telephone call because of which she has to leave immediately. Or, after a while she changes her mind and asks for a cup of tea instead of coffee.

The revocation pattern is exhibited in figure 14. It starts with the revoke request [rv rq] by the initiator, resulting in the status (rv rq), indicated by the yellow path from (?)

to (rv rq). The wait link from (rq+) to [rv rq] represents the condition that the revocation can only be performed if the status of the main process (rq) or further. The status (rv rq) is a discussion status (indicated by the double disk), which means that the two actors have to sit together in order to discuss the proposed turning back of the main transaction process. The executor may allow the revoke, indicated by the continued yellow path to (al) or refuse it (the red path). If he/she allows, the status of the main process will be reverted (in), as indicated by the green-white path on the right side of the figure. If he/she refuses, the status of the main process remains what it is.

Note that a revoke request can even be performed from the status (ac), as shown in the figure, so when the transaction has been completed successfully. A common example is that one has bought something from a shop and brings it back because one sees no need for having it anymore. The being revoked is a discussion status (indicated by the double disk), which means that the two actors have to sit together in order to discuss the proposed reversion of the main transaction process. The executor may allow the revoke (continued yellow path) or refuse it (red path). If he/she allows, the process in the standard pattern will revert to the status (in), as indicated by the green-white path on the right side of figure 14.

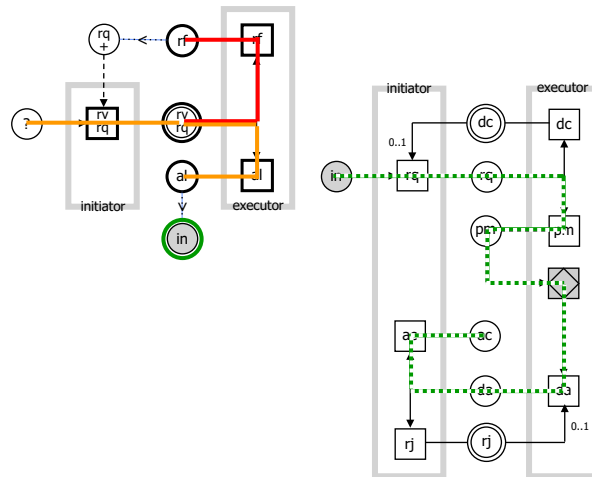


Figure 14 The process of revoking a request act

A modern illustration of the process sketched above is the returning of goods that one has acquired from an online shop. In terms of the PSI theory, it means that one revokes the request, by which the transaction is completely ‘rolled back’. As said before, for (business) transactions like acquiring goods, there is mostly a favour in return. Normally this is the payment for the delivered goods or the rendered service. It is a separate transaction, as discussed in the OMEGA theory [TEE-09]. Rolling back the acquiring transaction then implies both rolling back the physical delivery transaction and rolling back the payment transaction, as discussed in section 2.6.6.

As becomes evident from the discussion of the complete transaction pattern above, the path of every single transaction through this pattern, may comprise an arbitrary

number of loops. But there are some definite landmarks, however. Both the order phase and the result phase have a clear successful completion. The order phase of a transaction is completed successfully if the (latest) promised product is equal to the (latest) requested one. Likewise, the result phase of a transaction is completed successfully if the (latest) accepted product is equal to the (latest) declared one. If so, the accepted product starts to exist at the event time of the acceptance.

Still, even a successfully completed transaction is not concluded forever. It is always possible to ‘break it open’ by revoking one of the basic steps. The permanent option to (try to) return to a previous status in the transaction process, long after the transaction is completed, has important practical relevance. All warranty clauses are actually about revoking the accept act.

2.6.8 The operating principle of organisations

Every (dynamic) system has some internal mechanism that makes it ‘tick’. For inanimate systems, this mechanism is usually called the operating principle. Regarding animate systems, one usually speaks of the vital force [17]. But, although the subjects in organisations (and other social systems) are animate systems, the organisations themselves are considered inanimate. Therefore, we speak of the operating principle of organisations, in much the same way as one speaks of the operating principles of cars, aircrafts, etc.

By performing coordination acts, subjects enter into and comply with commitments towards each other regarding the product to be brought about. *Commitments* are the social agencies through which subjects collaborate. In other words, subjects comply with commitments because they feel the social/cultural obligation to do so. We consider this ‘mechanism’ to be the *operating principle* of every organisation. The underlying premise, as articulated already in Habermas’ Theory of Communicative Action [6], is that the two actors in a transaction strive to reach consensus. This is only possible if the actors *trust* each other, and this trust is verified in particular when the claim to rightness is assessed (cf figure 2).

2.7 Transaction plus actor is transactor

In the previous sections, we have talked about transactions and about the initiator role and the executor role that subjects play in carrying them out. In all exhibited transaction patterns, there is a clear separation between the acts that the initiator of a transaction can perform and the acts that fall in the responsibility area of the executor. In the diagrams, these areas are indicated by grey coloured, bold lined rectangles. We have also discussed the necessity of precisely formulating the product of a transaction, in such a way that it is uniquely identified in space and time (cf section 2.2.1). An example of an independent P-fact type description that we have used, is “Membership is started”, where “Membership” intentionally starts with a capital, expressing that it is a placeholder or variable in the logical predicate type <Membership is started>⁸. An example of an instance of this predicate type (in mathematical logic called proposition) is <membership 387 is started>. Henceforth, we will take a logical expression

like <Membership is started> as the expression of a *product kind*, while being aware that it is only the expression of its independent P-fact type (cf section 2.2.2).

Related to the notion of product kind, we will use the notion of *transaction kind*, as a basic property of every transaction. There is a one-to-one relationship between transaction kinds and product kinds. A proper naming of the transaction kind, in whose instances products of the kind <Membership is started> are brought about, could be “membership starting”. Next, we introduce the notion of *actor role*, defined as the authority to fill the executor role in transactions of a particular transaction kind. This authority can be assigned to subjects, through which these subjects are eligible to act accordingly. This topic will be elaborated in section 3.3.

The combination of a subject and an assigned actor role is called *actor*. Only actors can act, i.e. be active in transactions. A subject may fill several actor roles and an actor role may be assigned to several subjects, both sequentially and simultaneously. As examples in the café situation, the waiter (subject plus actor role) can at the same time be cashier (same subject, different actor role), and there may be several waiters in the café (one actor role, assigned to several subjects). Next, an actor role can be assigned to a collectivity of subjects, which means that these subjects can fill the role only together. Well-known examples are the board of directors of a company and the general assembly of an association.

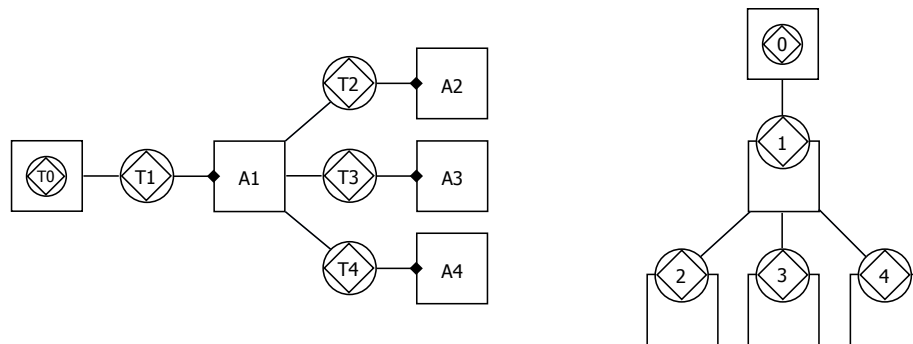


Figure 15 Graphical notations of transaction kinds, actor roles, and transactor roles

As a consequence of the definition of actor role, every transaction kind has exactly one actor role as its executor role, and vice versa. An actor role may however have an initiator role in several transaction kinds. The left side of figure 15 shows the graphical notation of these relationships. The shape of a transaction kind is a diamond (the symbol of production) in a disk (the symbol of coordination), and the shape of an actor role is a box (possibly stretched to a rectangle). A transaction kind may be connected to an actor role through an initiation link (represented by a line between the shapes) or through an execution link (represented by a line between the shapes and a small black diamond at the junction of the line and the actor role shape). The figure exhibits that actors A0 are the initiator in transactions T1 and actors A1 are the executor of transactions T1⁹. Actors A0 are self-activating, which means that they are

both initiator and executor of transactions T0 (cf DELTA theory [TEE-04]). Next, actors A1 are also initiator of transactions T2, T3, and T4, of which respectively actors A2, A3 and A4 are the executors.

The *transaction kind shape* has two interpretations. In the *constructional interpretation*, it represents the complete transaction pattern, as discussed in section 2.6.3. Every instance of a transaction of the transaction kind is some path, possibly including iterations, through the complete transaction pattern. In the *operational interpretation*, the transaction kind shape represents the history of all transaction processes of the transaction kind. In other words, it is the conceptual store of all created C-facts in the course of time. The shape is therefore also called *transaction bank*. Basically, the subjects who fill or have filled the related actor roles in one or more transactions of some kind, have access to the contents of the corresponding transaction bank, possibly restricted to the transactions in which they have participated personally. In the OMEGA theory [TEE-09], it will be clarified that also other actors may have (reading) access to transaction banks.

The *actor role shape* has also two interpretations. The *constructional interpretation* is that it represents the authority to fill the executor role in transactions of the transaction kind to which it is linked by an execution link. The *operational interpretation* is that it represents the guidelines, ranging from norms and values (culture) to specific imperative rules (business rules), that actor in the actor role apply when they carry out transactions. These guidelines regard the role of executor of transactions of the kind to which it is linked by an execution link as well as the role of initiator in transactions of the kind to which it is linked by an initiation link. So, for example, actor role A1 contains guidelines both for dealing with C-events in transactions T1 (of which it is the executor) and for dealing with C-events in transactions T2, T3 and T4 (of which it is an initiator).

Because of the one-to-one relationship between a transaction kind and its executor role, it makes sense to use one shape for the combination of the two. This shape, called *transactor role*, is shown at the right side of figure 15. The diagram represents the same structure as the one on the left, but now from top to bottom instead of left to right (which is semantically equivalent). Another simplification is that we just number transactor roles, without using prefixes (like the “T” and the “A” that are used on the left side of figure 15). Henceforth we will preferably talk of transactors and transactor roles, thereby emphasising that transactors in the PSI theory, and in Enterprise Engineering at large, are transaction processing actors. The links between the transactor roles in figure 15 are *initiation* links. So, for example, transactor role TA1 is initiator in transactions of the kinds T2, T3 and T4. Put differently, carrying out a transaction T1 comprises carrying out a number of transactions T2, T3 and T4 (cf OMEGA theory [TEE-09]).

Because transactor roles can have an initiator role in several transaction kinds, next to having the executor role in their ‘own’ transaction kind, they constitute tree structures (which are elaborated in the OMEGA theory [TEE-09]). To illustrate the meaning of the tree structure in figure 15, we use the Pizzeria case (as used in DEMO courses¹⁰). Table 2 represents the so-called Transactor Product Table of the Pizzeria.

The whole process of completing a client order (which is conceived as a sale by the Pizzeria) starts with the request of a transaction of the kind {sale completing}¹¹ by someone who fills transactor role *client* (TA0), to someone who fills transactor role *sale completer* (TA1). During the carrying out of this transaction, three other transactions are initiated: one of the kind {sale paying}, one of the kind {sale baking} and one of the kind {sale delivering}. As clarified by the OMEGA theory, all three must be carried out before the transaction of the kind {sale completing} can be finished. In other words, every product of the kind <Sale is completed> comprises three other products: one of the kind <Sale is paid>, one of the kind <Sale is baked> and one of the kind <Sale is delivered>.

#	transactor role	transaction kind	product kind
1	sale completer	sale completing	Sale is completed
2	sale payer	sale paying	Sale
3	sale baker	sale baking	Sale is baked
4	sale deliverer	sale delivering	Sale is delivered

Table 2 Transactor Product Table of Pizzeria

3 Elaborations

3.1 Time aspects of transactions

As we have discussed in section 2, the process of a transaction is a time series of C-events, starting with the request (in response to the initial status) and ending with either the acceptance (in case of success) or a reversion to the initial status (in case of failure). We will ignore for the moment the third possibility of an eternal deadlock, which was discussed in section 2.6.2. Let us have a closer look now at the time aspects of C-acts, which may lead also to a deeper understanding of the concept of transaction.

The structure of a C-act/fact that was presented in figure 4, is reproduced in figure 16, now without the specific example, but including the time aspects that are common to all C-acts/facts. The first one is the point in time at which the C-act is performed, and thus at which the C-fact is created. This is the *event time* of the C-fact. The second time aspect of C-facts is their (intended) *settlement time*. By this is meant the point in time at which the performer of the C-fact wants the addressee to settle it. Although often, like in the café example, the settlement time is asap (as soon as possible), it may be set at a specific point in time, thus introducing a specific time delay. In technical systems, time delays are purposefully applied to synchronise processes, as discussed in the DELTA theory [TEE-04]. In social systems, notably organisations, the settlement time tells the addressee that the performer wants him/her to respond at that time, not later and not sooner. Note that a point in time is actually a time interval

of some length (cf DELTA theory [TEE-04]). If no specific settlement time is set, the performer of the act expects the addressee to respond within a reasonable amount of time after the event time. For example, if the client in the café asks the waiter for a cup of coffee, she assumes that the waiter responds to her request within a couple of seconds. She will certainly be amazed if this would take a minute.

Regarding P-acts/facts, we distinguish between the event time and the production time. The *event time* of a P-fact is by definition equal to the event time of the accept fact in the corresponding transaction. If the acceptance is performed tacitly, it is equal to the event time of the declared fact. In addition, every P-fact has a *production time*. It is a property of the product, more precisely of its independent P-fact. In the course of the transaction process, four values of the production time are distinguished: the requested, the promised, the declared and the accepted production time. A product becomes effective at the accepted production time.

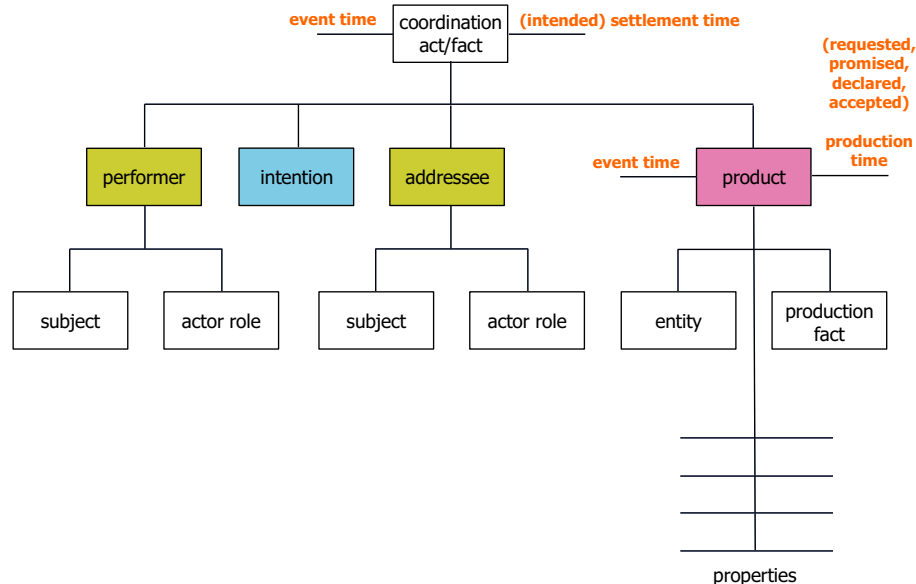


Figure 16 Time aspects of coordination acts/facts

Mostly, like e.g. in the café example, the initiator wants the product to be brought about asap. However, in a business-to-business situation, one often wants to set a specific future point in time. This holds for all product categories, as discussed in the OMEGA theory [TEE-09]: creation of things, transport and storage, transferring ownership, transferring right of usufruct. As an example in the category of transferring ownership, you may give a money transfer order to your bank today but have the transfer effectuated on the 28th of this month. Other quite common examples are making appointments with professionals, like medical doctors, for getting diagnosis, advice or treatment (all of them falling in the category of creating things), and making reservations for hotel stays, theatre performances, car rentals, flights, etc (all of them

falling in the category of transferring right of usufruct). So, for example, if you conclude today a reservation for a hotel stay of three nights, starting on the 13th of this month, the event time of the product is today, but the production time is let's say 15:00 hrs. on the 13th. From that time on you have the right to use a room of a particular kind. In order to effectuate this right, you have to check in. When doing this, you will be requested to check-out on let's say 11:00 hrs. on the 16th.

For the product category transferring right of usufruct, a future production time is also common, and it may even make sense to set the production time in the *past*. This may sound strange at first sight, but it is quite usual. An example is that you subscribe to a monthly magazine, let's say during the month of May, but that the subscription starts in retroaction from January on, so that you receive all issues of the current year. Similar advantages may hold for becoming member of an association in retroaction.

3.2 The operating cycle of actors

In the DELTA theory [TEE-04], we have briefly discussed the operating cycle of processors, in the next way. At every point in time, a processor loops through its *operating cycle*. It then 'selects' a C-event from the current C-events to be dealt with.

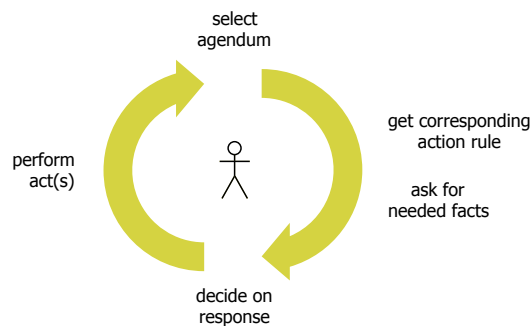


Figure 17 The operating cycle of actors

A similar definition holds for the operating cycle of actors, which is shown in figure 17. At every moment, an actor disposes of his/her agenda, which is a set of C-events to deal with. The cycle begins when the actor selects the agendum to be settled. Although the actor is basically autonomous in this step, the selection may be influenced by managers through adding priority attributes, in addition to the intended settlement time, set by the performer of the C-act. This passive way of influencing by management is crucial. Managers are not part of the operational processes.

Then, the actor fetches the applicable action rule, which is similar to an imperative business rule [18]. If there is no specific action rule, he/she is lead by the general guidelines (culture, policies) of the enterprise [18]. Next, the actor gathers the facts he/she needs, and decides how he/she will respond to the selected C-event. Then, the actor performs the act(s) that are implied by the response that he/she has decided to perform. As said, action rules are guidelines, because actors are autonomous in deciding how to act. However, they are responsible and possibly also accountable for their

acts (cf section 3.3). To illustrate the point, suppose that a small tennis club in the countryside of The Netherlands has reached the maximum number of members, but then someone named Rafael Nadal asks to become member. What do you do? A good chance that the general meeting will endorse your decision to let Nadal in.

3.3 Human qualities in transactions

In section 2, the notions of responsibility and authority were briefly touched at. We have talked about the responsibility areas of the initiator and the executor in a transaction, and we have defined the notion of actor role as the authority to be executor in transactions of a specific transaction kind. Let us have a closer look at these human qualities, and at two other ones, namely accountability and competence. The definitions we will provide, are not (fully) compatible with the RACI framework¹², which is currently quite popular in management sciences. The reason for the incompatibility is that the corresponding notions in the RACI framework lack a solid foundation. What will be defined hereafter, has a solid foundation in the PSI theory.

3.3.1 Authority

The notion of *authority* is defined as the right of a subject to perform particular C-acts in response to particular C-events. Taking the café situation again, the subject who is referred to as waiter, apparently has the authority to be executor in ‘café sale’ transactions, and the subject who is referred to as client, apparently has the authority to be initiator in these transactions.

There are two ways in which authority can be assigned to subjects: through authorisation and through delegation. *Authorisation* is the common way in which authorities are assigned to people in organisations, and it is commonly the authority of human resource functionaries to authorise employees. In current practice, the unit of authorisation is mostly the organisational function: salesperson, accountant, assistant accountant, secretary, trainee, etc. In the PSI theory, the unit in which authorisation takes place is the (trans)actor role.

An authorisation may include the right to *delegate* a part of the authority to other subjects. In the PSI theory, this part is a subset of the transaction steps in the transaction kind(s) for which the delegator is authorised to respond to.

Delegation is sometimes inevitable, for physical/logistic reasons. The case Pizzeria, well-known from the DEMO courses¹³, is a good example to illustrate this. Table 2 shows the applicable transaction kinds and actor roles. The subject who fills the actor role **sale completer**, called Mama Mia, is executor of transactions {sale completing} and initiator of transactions {sale paying}. This implies that she has to perform the accept act in transactions {sale paying}. However, if the pizza’s are delivered at the home address of the customer, she has no other choice than to delegate this process step to the delivery boy, who fills the actor role **sale deliverer**. In addition, she has to delegate the authority to perform the declare act in transactions {sale completing} to the delivery boy, because he is the only one who has direct contact with the customer at the customer’s home address.

3.3.2 Responsibility

The notion of *responsibility* seems to have two meanings, which we will refer to as feeling responsible and being responsible. By nature, most people *feel responsible* for using the resources offered by our planet in such a way that they will not get exhausted, or for causing irreversible damages to the environment. Similarly, most people feel responsible for behaving properly vis-à-vis fellow human beings. In the context of enterprises, most people feel the social obligation to exert their authority in the best way possible, so by acting professionally, by behaving properly towards other actors, etc. It corresponds with the fundamental ‘social’ principle in the PSI theory to base one’s decisions on one’s wisdom and love (cf section 2.3).

In our view, the meaning “*being responsible*” is the institutionalised version of “feeling responsible”. If someone has been assigned a particular authority (either by authorisation or by delegation), he or she is said to be responsible for exerting this authority in the best possible way. As a consequence, authority and responsibility may be considered as the two sides of the same coin; none of them can exist without the other. In section 2.6, we applied this notion already when we spoke of the responsibility areas of the initiator and the executor in carrying out transactions.

A related crucial notion in the PSI theory is *autonomy*, by which is meant that actors may deviate from the existing rules or norms. To illustrate this important point, let us take a purchaser in a manufacturing company as an example. Suppose that he/she gets the advice by the ‘intelligent’ purchase system to place a purchase order from a specific supplier. Normally, the purchaser would follow the advice, but suppose that he/she has heard the evening before in the local bar that this supplier is going bankrupt soon. Wouldn’t it be irresponsible to place the order? And, doesn’t he/she have a good story to justify his/her disobeying the rule?

In order to clarify this issue, one often distinguishes in current practice between advisory and compulsory business rules. Advisory rules can be disobeyed but compulsory business rules cannot. In the light of the foregoing discussion, the question is whether this distinction is tenable. Also with respect to compulsory rules, there can always be a situation that requires one to deviate from the rule, just and only for acting responsibly.

3.3.3 Accountability

Responsibility seems often to be confused with *accountability*, which we define as the obligation to provide justifications of one’s acts, whenever there is a need for it. Because of the basic autonomy that actors have in the PSI theory, they are accountable for all of their acts. This is what autonomy is all about. So, if a particular subject S1 is authorised to perform certain acts, he/she is responsible and accountable for the way he/she performs the acts, also if the authority is acquired through delegation. This becomes particularly apparent when an actor has violated the applicable business rules (for which there may be very good reasons, as we have seen). The actor will be held accountable for having deviated from the rule or the norm. In such a case, sanctions may be applied, depending on the severity of the deviation.

Related to accountability is the notion of *liability*, which means that an actor may be persecuted legally on the basis of the applicable Civil Code. With respect to liability, there is an important difference between authorisation and delegation: delegates may be held accountable for their deeds, but they are never liable. Liable is the actor who has delegated the corresponding part of his/her authority. Therefore, distinguishing between authorisation (often also called mandating) and delegation is an interesting issue in every enterprise. Ultimately, the ‘big boss’ is liable.

3.3.4 Competence

Although one may assign any authority to any subject, one normally would do so on the basis of his or her competence. By someone’s *competence* is understood the totality of knowledge, skills and experience that the person possesses. Competence is thus a capability. It can be divided in production competence and coordination competence. Production competence is quite specific; it regards the being able to bring about products of one or more kinds. On the other hand, coordination competence is quite generic; it regards basically the performing of all coordination acts in the complete transaction pattern. It belongs to what are commonly called communication skills or soft skills.

To exemplify this, let us assume that someone has acquired, e.g. through education, the (production) competence of a plumber. In order to practice this competence, he/she has to get the corresponding authority from some (institutional) actor, for example the boss of a plumbing company. Once this is done, he/she is expected to exert the authority in a responsible way, and he/she is accountable for all acts, as discussed in the previous sections. This will of course apply to the ‘real’ plumbing work in the houses of the clients of the company. But it also holds for the corresponding coordination acts. More specifically, the plumber is supposed to behave properly in the contacts with the clients.

4 Discussions

4.1 Comments on the previous transaction pattern

During the EEWK (Enterprise Engineering Working Conference) in 2016, I had a discussion about the transaction pattern with Duarte Gouveia, a PhD student at the University of Madeira, supervised by Dr David Aveiro. He pointed me at the ‘strange’ roles of the C-acts quit and stop in the complete transaction pattern that is in use since its publication in 2006 [19], of which figure 18 exhibits a slightly adapted version, almost similar to figure 10. In the course of our talk I came to realise that I had made an error in reasoning at the time. This error is corrected in the second version of the complete transaction pattern, as presented in section 2.

The problem I faced when devising the first pattern was the possible deadlock situations, which are discussed in section 2.6.2. They may occur, as we have seen, in the status declined and in the status rejected. The solutions I came up with in 2006, are the quit and the stop respectively. These solutions were ‘borrowed’ from authoritative

sources, notably Winograd and Flores [14]. The flaw that I have taken over is, as I see it now, to not make a distinction between what I have called the main process level and the meta level (cf. section 2.6.2). The insight I have got, after the discussion with Gouveia, is that these two levels should be clearly distinguished. Both the quit act [qt], which is an escape from the possible deadlock in (dc) and the stop act [sp], which is an escape from the possible deadlock in (rj), are solutions that cannot be part of the main process. Instead they must be sought for at the meta level because they aim at changing the status at the main level. As said, the correct solutions are provided by the second complete transaction pattern, as discussed in section 2. Concluding, transactions are basically carried out at two levels: the main level and the meta level (although the latter need not be there).

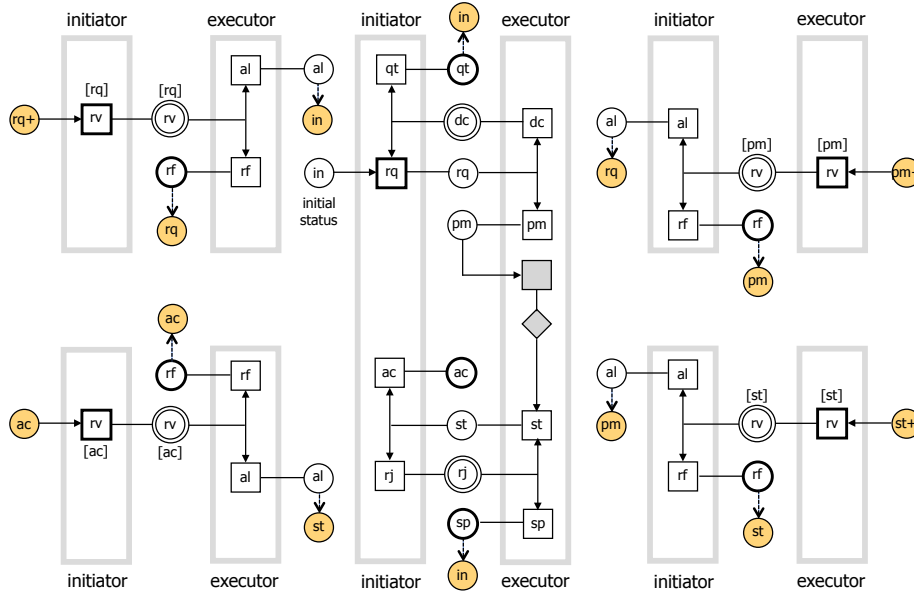


Figure 18 Adapted version of the first complete transaction pattern

4.2 Striving for consensus and culture

As discussed in sections 2.4 and 2.6, the fundamental assumption in Habermas' theory of communicative action, and consequently of the PSI theory, is that the participants in (business) transactions strive for consensus, thus for mutual agreement. In other words, they attempt to make the transactions that they are involved in, successful. This is the core of the notion of communicative action. Habermas recognises however that this precondition is not always satisfied. Therefore he includes and discusses a fourth category of communicative acts, namely the imperativa, where the claim to power is the dominant one. Habermas explicitly presents this category outside the region of 'proper' coordination. This is clarified in a comparison of Searle's speech act theory [5] and Habermas' theory of communicative action [6], which I

produced in collaboration with Guy Widdershoven [20], during the time that I worked at Maastricht University. Below, I insert a summary.

Because Searle overlooks the orientation towards mutual agreement, he is incapable to distinguish between power claims and validity claims. He considers communication primarily as an interaction between persons who try to let one another perform actions. A speech act thus succeeds if the course of action aimed at, is taken. In this ontology, it is impossible to distinguish a situation in which the addressee acts because he/she wants to evade sanctions, from one in which he/she responds to the demand of the performer because he accepts the validity of his/her claims in a rational way. Otherwise said, Searle's theory is incapable to distinguish between empirical and rational coordination of action. The central point of Habermas' critique however is that Searle fails to reveal what really makes a speech act work. This mechanism is that validity claims are criticisable, stemming from the orientation of the participants towards mutual agreement, and giving rise to negotiations about the claims made. It is particularly because of this weakness in Searle's theory that his taxonomy misses several important distinctions. One of these is the distinction between speech acts that are based on power claims and speech acts that are based on validity claims (or speech acts proper).

Whether people in organisations are engaged in proper coordination, i.e. communicative acts that succeed because they satisfy validity claims, or in imperative coordination, i.e. exerting the power to force other to do things, is largely a matter of culture, as explained in the SIGMA theory [TEE-07]. Culture is defined as the whole of values, norms, convictions, and beliefs (rational or irrational, implicit or explicit) that the members of an organisation (and of societies at large) have learned through social interaction, and apply in their collaboration. Cases about successful enterprise transformation, like the famous NUMMI case¹⁴, show that management behaviour in the form of leadership and culture (meaning, purpose, norms, and values) are the crucial determinants of enterprise success.

Bibliography

1. Mintzberg, H., *The structuring of organizations : a synthesis of the research*. Theory of management policy. 1979, Englewood Cliffs, N. J. ; London: Prentice-Hall International. xvi, 512 p.
2. Bunge, M., *Treatise on Basic Philosophy Ontology II: A World of Systems*, in *Treatise on Basic Philosophy* 4. 1979, Reidel: Dordrecht, Netherlands.
3. Dietz, J.L.G., *Red garden gnomes don't exist*. 2012: Sapio Enterprise Engineering, The Netherlands.
4. Austin, J.L., *How to do things with words*. 1962, Cambridge: Harvard University Press.
5. Searle, J.R., *Speech acts: an essay in the philosophy of language*. 1969, London,: Cambridge U.P. vii, 203 p.
6. Habermas, J., *The theory of communicative action*. 1986, Cambridge: Polity Press.

7. Weigand, H., Two decades of the language-action perspective. *Communications of the ACM*, 2006. 49(4).
8. Habermas, J., *Theorie des kommunikativen Handelns*. 1981, Frankfurt a. M.: Suhrkamp. 2 Bde.
9. Searle, J.R., *The construction of social reality*. 1995, New York: Free Press. xiii, 241 p.
10. Edmondson, W.J., *Spoken discourse : a model for analysis*. Longman linguistics library. 1981, London ; New York: Longman. 217 p.
11. Grice, H.P., *Logic and Conversation*. Syntax and Semantics; Academic Press, New York, 1975. Vol 3: Speech Acts.
12. Grice, H.P., *Further Notes on Logic and Conversation*. Syntax and Semantics; Academic Press, New York, 1978. Vol 9: Pragmatics.
13. Steuten, A., *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*. 1998, Delft: Delft University of Technology.
14. Winograd, T. and F. Flores, Understanding computers and cognition : a new foundation for design. 1986, Norwood, N.J.: Ablex Pub. Corp. xiv, 207 p.
15. Denning, P. and R. Medina-Mora, *Completing the loops*. *Interfaces*, 1995. 25(3): p. 15.
16. Reijswoud, V.E. van, *The Structure of Business Communication: Theory, Model and Application*. 1996, Delft: Delft University of Technology.
17. Bergson, H., *Creative evolution*. 1913, London: Macmillan. XV, 425 S.
18. Dietz, J.L.G., On the nature of business rules, in *Advances in Enterprise Engineering I*. 2009, Springer: Berlin-Heidelberg.
19. Dietz, J.L.G., *Enterprise ontology - theory and methodology*. 2006, Berlin ; New York: Springer. xiii, 243 p.
20. Dietz J.L.G., Widdershoven G.A.M.: Speech Acts or Communicative Action? in *Second European Conference on Computer Supported Cooperative Work*. 1991. Kluwer, Dordrecht.

Referenced Theories in Enterprise Engineering

TEE-00: Foundations of EE

TEE-01: FI theory (Factual Information)

TEE-02: TAO theory (Teleology - Affordance - Ontology)

TEE-04: DELTA theory (Discrete Event in Linear Time Automaton)

TEE-07: SIGMA theory (Socially Inspired Governance and Management Approach)

TEE-08: ALPHA theory (Actor Layers and PHysical Abstraction)

TEE-09: OMEGA theory (Organisational Modules Emerging from General Arrangements)

Endnotes

¹ In enterprise engineering, the term “enterprise” is used to denote any instance of human collaboration, ranging from organising a birthday party to running a multinational company.

² ICT stands for Information and Communication Technology. It refers specially to the modern practice of applying electronic digital means to process, store and transmit data (understood as the form part of information, as discussed in [TEE-01]).

³ The English word “communicate” comes from the Latin word “communicare”, which means ‘making something common’. In a more specific sense, it means ‘sharing thoughts’.

⁴ The notion of “ideal speech situation” is introduced in the early philosophy of Jürgen Habermas. Based on various criticisms in the course of time, Habermas elaborated the notion into “discourse ethics” in his book “Moral Consciousness and Communicative Action” (written in 1983; the English translation is published in 1990).

⁵ The words actagenic, factagenic, reversiogenic and cogitatiogenic mean respectively: act creating, fact creating, reversion creating, and idea or plan creating.

⁶ The noun “transaction” is related to the verb “to transact”, which originates from the Latin verb “transigere”, meaning carrying out, bringing through.

⁷ In order to refer simply to acts and facts (or statuses), we use “(“ and “)” for a C-fact or conversation status, and “[“ and “]” for a C-act or P-act. For example, [rq] denotes the act of requesting and (rq) denotes the fact of being requested.

⁸ Next to the convention to use “[“ and “]” to indicate C-acts as well as P-acts, and “(“ and “)” to indicate C-facts or conversation statuses, we will use “<“ and “>” to indicate P-facts.

⁹ We will write “actors Ax” as a shorthand for “actors in actor role Ax”, and “transactions Tx” as a shorthand for “transactions of the kind T1”

¹⁰ For more information about DEMO courses, visit www.ee-institute.org

¹¹ In addition to the conventions that have been introduced, we will use “{“ and “}” for transaction kinds, “*” and “*” for indicating transactor roles, and “<“ and “>” for product kinds.

¹² See <https://www.raci.com>

¹³ For more information about DEMO courses, visit www.ee-institute.org

¹⁴ <https://en.wikipedia.org/wiki/NUMMI>