ALICIA P.C. PERINFORMA

THE ESSENCE OF ORGANISATION

AN INTRODUCTION TO ENTERPRISE ENGINEERING

Colophon

Perinforma, A.P.C.: The essence of organisation
3rd revised edition (2.1 +NL)
ISBN: 978-90-815449-4-8

1st edition: 2013
2nd edition: 2015

The front cover picture is taken from Lars Groth, Overview of theories on organizations and management.

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

(Henry Mintzberg, The Structuring of Organizations, 1979)

To all enterprise engineers

CONTENTS

PROLOG

The core of the organisational problem can hardly be better defined than in the way Henry Mintzberg did in 1979, as cited on the first page of this book. Still, the question remains how one should understand the division in tasks and their coordination. What exactly is a task? How big or small can it be? Is there a smallest task? And if so, how can bigger ones be built from them? Is there a 'best' division? And if so, how does one arrive at it? What exactly is coordination? Do tasks or people coordinate? And what exactly do they coordinate? These questions constitute the starting point of this introductory book in enterprise engineering, an emerging new discipline, that will be discussed in more detail in the first chapter.

In 1926, Albert Einstein crisply responded to Werner Heisenberg, when the latter plead for more experimental physics at the expense of theoretical physics: "Whether you can observe a thing or not depends on the theory that you use. It is the theory that decides what can be observed". The consequences of this statement, which I regard as a fundamental truth, are far reaching. Foremost, it means that there isn't such a thing as objective observation. One always looks through some mental glasses, may be unwittingly. Next, it entails that a particular theory may be more appropriate for a certain purpose than others. Consequently, it will always make sense to search for better fitting theories.

Einstein's statement, and its consequences, are the incentive for the Ciao! Network[1] to construct a solid and appropriate theoretical foundation for the discipline of enterprise engineering. One of these theories, and also the core one, is the PSI theory (Performing in Social Interaction). It serves as the main mental glasses through which I will look in this book to organisations, in order to answer the questions in the first paragraph. As you will experience while reading the book, the PSI theory allows one to deal with many organisational issues in a clear and integrated way. And to keep organisational changes or transformations intellectually manageable, which is one of the three generic goals of enterprise engineering, as envisioned by the Ciao! Network. The key to achieving this, is a huge but systematic reduction of complexity, culminating in the so-called essential model of an organisation.

---

[1] www.ciaonetwork.org

Together with "Red garden gnomes don't exist", this book constitutes the DEMO[2] Basis, the text book for the DEMO Bachelor course, which is taught in many universities and polytechnics, all over the world. In "Red garden gnomes don't exist"[3], Prof. Jan Dietz, at the time lecturer of Enterprise Ontology at the TU Delft, has written down my adventures in Gnomeland. In the epilog of the book, he has expressed the wish that I would write one day an introduction to enterprise engineering. Well, this is the book. It is meant for everyone who wants to acquire basic knowledge of enterprise engineering, in particular of DEMO, the principal methodology in enterprise engineering. The acquired knowledge is sufficient for understanding and validating DEMO models of organisations and for participating in all kinds of organisational studies and discussions that are based on these models.

In this book, I will explain what the essential model of an organisation is and what benefits it can offer. I will do this by using two small examples for illustration. The first one is a tennis club and the second one is a car rental company. I am fully aware that practical cases are much larger and much more complicated than these 'toy' examples. However, I am also quite confident that I can convey to the reader the gist of enterprise engineering. In particular, you will learn and hopefully appreciate how PSI and DEMO can help you in achieving intellectual manageability in dealing with your problems, so that you get the right insight and overview to make decisions about enterprise changes of any kind in a responsible way. Throughout the book I have applied the way of thinking, the way of modelling, and the way of working of DEMO-3, the most recent open standard, which is managed by the Enterprise Engineering Institute[4].

Lisboa, 28 April 2014,

Alicia P.C. Perinforma

---

[2] DEMO stands for Design and Engineering Methodology for Organisations.

[3] Dietz, J. L.G.: Red garden gnomes don't exist, 2010, ISBN: 978-90-815449-2-4

[4] www.ee-institute.org

1 INTRODUCTION

In the book "Red garden gnomes don't exist" (RGGDE)[5], Sapio summarises the PSI theory (Performing in Social Interaction) in this motto: *Communication is the thread of which Organisation is woven*. I like the analogy very much, but I also think Sapio doesn't know much about textile techniques. Otherwise he would have said "knitted", because in knitting you have really one thread, while in weaving you interlace threads passing in one direction with threads orthogonal to them. If one 'deconstructs' a knitted sweater, one ends up with only the thread, the sweater having vanished. Anyway, the basic lesson of Sapio's motto is that the key for understanding the operation of an organisation is in the communication between the people that constitute an enterprise's organisation[6]. As conveyed by the PSI theory, people are the pearls of every enterprise. Without people, there is no organisation, in a very literal sense. Although I am aware that human communication plays a much larger role in constituting organisations, I will limit myself to the communication that is directly related to the coordination of production acts. Based on their ability and their readiness to enter into and comply with commitments towards each other, people succeed in bringing about the products that are conceived as the business services by the customers of the enterprise (and by other stakeholders). A prerequisite is that the involved persons have the right authority, corresponding with the kind of product or service. In return, these persons are assumed to exert their authority in a responsible way. In my experience, most people like to work in this manner, they long for being empowered. They like to bear responsibilities in an organisation where authorisations are explicit and transparent. In short, people are the pearls in every organisation.

The people who actually constitute an organisation are also the main reason for the differences between enterprises in the same line of business, because the ontological essence of these enterprises is basically the same, as you will learn. Yet if one compares these similar enterprises from direct observations, the most apparent thing seems to be the many differences between them! I sometimes use the following

---

[5] Dietz, J.L.G.: Red garden gnomes don't exist, ISBN 978-90-815449-2-4. From now on, the book will be referred to by RGGDE.

[6] I will use the term "*enterprise*" to refer to any kind of collaborative activity by human beings: companies, governmental agencies, health care institutions, sports clubs, building projects, etc. With "*organisation*" I will refer to the construction of an enterprise, and with "*business*" to its function(s).

metaphor of the human body to explain this. Most people can very easily see the differences between human bodies and consequently are able to recognise them easily and quickly. But if they would only see each other's skeletons, it would become rather difficult. Apparently, the main differences are in the 'realisation and implementation' of the skeleton, i.e. the 'outside' that we observe with our senses. Well then, the essential model of an enterprise is comparable to the skeleton of a human body. It explains the fundamental construction and operation of the enterprise's organisation but it leaves a lot of freedom for its final shaping. If you want to change the realisation and implementation, thus if you want to re-design and re-engineer the enterprise, you clearly should start from this essence. In addition, the main reason why you become overwhelmed by all the details you observe when entering a large factory or office, why you often can't see the forest through the trees, is the lack of an appropriate theory, by which you can 'look through the skin and flesh into the skeleton', like an X-ray camera does.

Such an appropriate theory is the PSI theory. I will show you how you can apply it and discover the essence of every organisation. In current practice, one normally starts understanding an organisation by modelling its business processes. Examples of current modelling techniques are BPMN, ARIS/EPC, and Petri Nets. As it turns out, these techniques reduce business processes to work flows. Not only do they disregard the key role of people, they also disregard the deep structures that we will unveil and the realisation and implementation independent essence that we will extract. Therefore, such process models are not very suited for the purpose of (re-) designing and (re-) engineering business processes. Moreover, their semantics is often not well defined. Obviously, communicating in a language where the same sentence may have many meanings, is not the most appropriate way to discuss organisational problems. It reminds me of the dispute of Alice and Humpty Dumpty in 'Alice's Adventures in Wonderland'[7]:

"When I use a word," Humpty Dumpty said in rather a scornful tone, "it means just what I choose it to mean -- neither more nor less."
"The question is," said Alice, "whether you can make words mean so many different things."
"The question is," said Humpty Dumpty, "which is to be master - - that's all."

Nowadays, one would say that Humpty Dumpty speaks a 'barbapapa' language, where the meaning of a word can easily turn into almost any other one. In enterprise engineering, there is no place for barbapapa.

---

[7] Caroll, L.: Alice's Adventures in Wonderland, 1865

The PSI theory is one of the (currently) nine theories that constitute the theoretical foundations of the discipline of enterprise engineering (EE), as envisioned by the Ciao! Network[8]. Three generic goals are formulated. One of them is *intellectual manageability*, the other two are *organisational concinnity* and *social devotion*. Intellectual manageability is particularly achieved by applying the PSI theory (and a few supporting theories), in current practice also known as enterprise ontology, one of the conceptual pillars of EE. There are four more powerful notions or conceptual pillars in EE: enterprise architecture, enterprise design, enterprise management, and enterprise governance. They all support the basic premise that enterprises are purposefully developed systems, which consequently can be re-developed (i.e. re-designed and implemented) whenever there is a need for it. In this book, the focus is on enterprise ontology.

Enterprise ontology is the (PSI theory based) understanding of the essence of organisation, completely independent of the way in which this essence is realised and implemented. The notion of enterprise ontology allows one to make unequaled reductions of complexity, such that one has insight in and overview over the most complex enterprise transformations. *Enterprise architecture* is defined as the normative restriction of design freedom. It enables one to make sweeping strategic statements operational, in a coherent and consistent way, through formulating them as design principles, which guide a (re-) development process on top of the identified requirements. More precisely, design principles tell the designer how the requirements should be satisfied, thus filling in how the design freedom should be used. *Enterprise design* is the systematic accomplishment of enterprise changes. It incorporates the notions of ontology and architecture as mentioned before. I will briefly touch on enterprise design later. Lastly, whereas e*nterprise management* is about keeping things going as they were designed to go, e*nterprise governance* is about devising changes when and where needed, as well as about initiating and steering their effectuation. In order to fully adopt the idea of enterprise engineering, it is required that enterprise (re)design, enterprise management, and enterprise governance are established capabilities in an enterprise.

In my studies, I have learned five intellectual techniques for intellectually managing the complexity of enterprises, and thus for coping with the challenges that enterprises are facing, like being agile, competitive, innovative, and cost-effective. The five techniques are listed in Figure 1.1. As a mnemonic to remember them I use the word "Sapio", which for some reason I can easily remember. For the same reason, I

---

[8] Dietz, J.L.G., Hoogervorst, J.A.P., et al.: The Discipline of Enterprise Engineering. In: International Journal of Organisational Design and Engineering, 2013, Vol. 1 No. 1

call these intellectual techniques *sapiences*. As I will go along explaining and applying the PSI theory, I will show how they can be a very effective help in achieving what the theory eventually is all about, namely the deep understanding of the notion of enterprise ontology, and to apply this notion in mastering the complexity of enterprises.

Separation of concerns
Use of **a**bstraction
Devising **p**roper concepts
Verification by **i**nstantiation
Validation from **o**ntology

Figure 1.1 The five sapiences

To illustrate my argument, I will make use of example enterprises that I have been involved in. The first, and smallest one, is the tennis club Volley. When I became member of Volley, some years ago, a lot of things in their administrative processes went wrong. This was not due to the applied technology; the latest achievements in ICT[9] were applied. Therefore I suspected that there was something wrong in their understanding of the administrative processes. But I did not feel an urge to meddle in. This feeling changed when I got into conversation some day with two elder members, Adam and Eve. They had been administrator and secretary of the club since the 1970's, when everything was still manual and everyone was happy. Their story made me curious. So, I told them about my current work and my study of enterprise engineering. Among many other things, Adam and Eve told me that one day, somebody who worked at a big software company joined the club. He offered to automate the paper work of Volley. He would apply a new method, the person said (whom I will refer to as Mr. X, in order not to discredit him). The board agreed on letting Mr. X automate the paper work of Volley. Adam showed a yellowed piece of paper on which he had written down the process of becoming member of Volley at that time, and which Mr. X had used as the starting point for his work. This was written on the paper:

[9] ICT stands for Information and Communication Technology

One can become member of the tennis club Volley by sending a letter to the club by postal mail. In the letter one has to mention one's surname and first name, birth date, gender, telephone number, and postal mail address (street, house number, zip code, and town). Adam, the administrator of Volley, empties the mailbox daily and checks whether the information provided is complete. If not, he makes a telephone call to the sender in order to complete the data. Once a letter is complete, Adam writes an incoming mail number and the date on the letter, records the letter in the letter book, and puts it in a folder.
Every Wednesday evening, Adam takes the folder to Eve, the secretary of Volley. He also takes the member register with him. If Eve decides that an applicant can become member of Volley, she stamps 'new member' on the letter and writes the date below it. She then hands the letter to Adam in order to add the new member to the member register. This is a book with numbered lines. Each new member is entered on a new line. The line number is the number by which the new member is referenced in the administration. Next, Eve calculates the fee that the new member has to pay for the remaining part of the calendar year. She asks Adam for the annual fee, as decided at the general assembly, which Adam has recorded on a sheet of paper. Then, she asks Adam to write down the amount in the member register.
If Eve does not allow an applicant to become member (e.g. because he or she is too young or because the maximum number of members has been reached), Adam will send a letter in which he explains why the applicant cannot (yet) become member of Volley.
When all applications are processed, Adam takes the letters and the member register home and prepares an invoice to all new members for the payment of the first fee. He sends these invoices by postal mail. Payments have to be performed by bank transfers.
As soon as a bank statement is received, Adam prints a card on which the member number, the starting date, the name, the date of birth, the gender, and the residence are mentioned. The card is sent to the new member by postal mail.

In this text, Adam continued, Mr. X started to mark all verbs and all nouns. The nouns would become records in an indexed-sequential file, he explained, and the verbs would become procedures in Cobol. In the future, Adam would have to make punch cards of the application letters, and he would have to base the letters he had to send to the (aspirant) members, on computer printouts. Nobody was pleased with this 'improvement' of the administrative work, except Mr. X. After several years, the board allowed Adam and Eve to return to the way they used to work, after which Mr. X resigned.

It didn't take long before another enthusiast, let us call him Mr. Y, appeared on the stage, Eve continued. He could do a much better job, he said, by applying the method that was popular in his company. Basically, he said, one must first draw Flow Charts of the processes. Mr. Y's approach sounded better to Adam and Eve, and they said they would like to give it a chance. Adam handed me a copy of the Flow Charts that Mr. Y had produced. On the basis of the Flow Charts of Volley, Mr. Y and a few other persons started to automate again the administrative processes of Volley. The system was implemented and it has been used since then. However, Eve said, it is still not what we really would like to have. As an example, it is rather unclear who is responsible for actions that the system takes: me, Adam, or the system. Consequently, everybody is blaming the system if something goes wrong, while I think that is not the way to deal with it, is it? No, of course not, I replied, without wanting to continue the talk then, however.

The discussion with Adam and Eve stayed in my mind for several weeks, before I decided to interfere. After a talk with some members of the board, in which I explained how an enterprise engineer would address the problem, I was invited to present my approach during a board meeting. All members of the board were eager to know more about it, and therefore a first workshop was scheduled. Gradually, it became clear that more time was needed, which resulted in several more meetings. Hereafter, I will report in detail on the meetings with the board of Volley, according to the following structure. In Chapter 2 (Defining organisational essence), the notion of organisational essence will be defined in full detail. First, the notion of transaction is presented and discussed. It will appear to be a universal pattern of production and coordination acts. Next, coordination acts and facts are studied in more detail, as well as production acts an facts. In Chapter 3 (Revealing organisational essence), a method for revealing or 'uncovering' the essence of an organisation is presented, and applied to the case Volley. Two sources of information are used: the narrative description of Mr. X and the Flow Charts of Mr. Y. In Chapter 4 (Representing organisational essence), I will present and discuss the way in which organisational essence can be expressed in specific diagrams, tables, etc., and I will exemplify this with my model of the case Volley, based on the analysis in Chapter 3. Chapter 5 (Modelling organisational essence) is completely dedicated to the analysis and subsequent modelling in DEMO of another case, namely Rent-A-Car (RAC). In Chapter 6 (Exploring organisational essence), I will discuss various topics and ideas concerning the notion of organisational essence, of which some need to mature still.

Because this is not a scientific but a practical book, there are no references to literature sources, except a few that I couldn't do without.

2 DEFINING ORGANISATIONAL ESSENCE

2.1 THE UNIVERSAL TRANSACTION

In preparation for the first workshop, every board member had studied the RGGDE book. Therefore, I could limit myself to a summary of the PSI theory. For clarifying the transaction concept, I used the picture in Figure 2.1. It exhibits the *basic transaction pattern*. The left picture is an intuitive one, whereas the right picture shows the same things in a more formalised way. The process starts when the initiator performs the request regarding some product (also called proposition in this phase), e.g. becoming member of Volley. Performing the request results into the *transaction status* requested (rq). The executor responds by promising, which brings the transaction in the status promised (pm). The executor responds to this event by creating the product, after which he/she states to have done it, by which the status stated (st) is reached. The product is also called result now, e.g. being member of Volley. In response to this event, the initiator accepts the result, which brings the transaction in the final status accepted (ac).

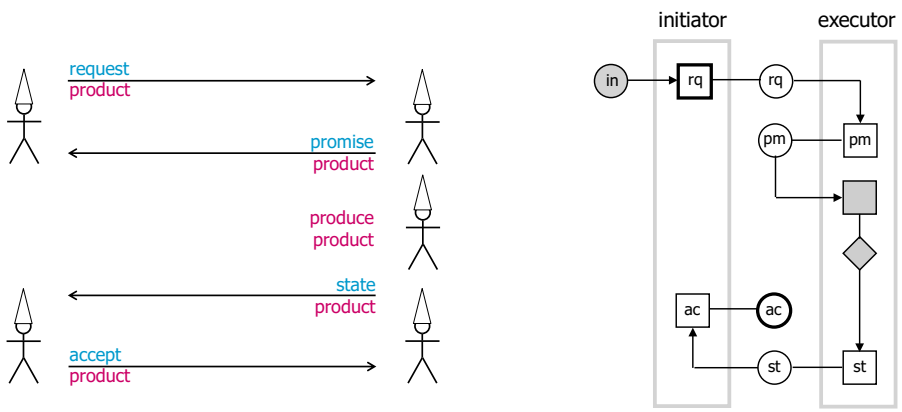


Figure 2.1 The basic transaction pattern

In the picture at the right side of Figure 2.1, the two *responsibility areas* of the initiator and the executor are indicated by means of light-grey coloured rectangles. The two distinct roles are authorised and responsible for the acts within their area. As shown, *coordination acts* are denoted by small white boxes, and *coordination facts* by small white disks. The reason for putting the disks between the responsibility areas, is that these facts are knowable to both actors. The production act is denoted by a small grey box, and the resulting production fact (product) by a small grey diamond. Grey means that this act/fact is 'private' to the executor, thus not knowable to the initiator. The small grey disk labeled "in", represents the initial status of the transaction process. It is, by nature, some status in some other transaction.

So, every coordination act is performed in response to a *coordination event*, i.e. to the creation of a coordination fact. In the figure, *response links* are denoted by an arrow from the coordination fact to the responding act. For the sake of convenience, I will use the notation "[rq]" to refer to the act of requesting, and "(rq)" to refer to the resulting fact of being requested. The same holds, mutatis mutandis, for the other coordination acts/facts. So, the arrow from (rq) to [pm] means that the executor performs the promise act in response to the event of the being requested of the proposition by the initiator. The line between [pm] and (pm) represents a *causal link*. It means that the act of promising causes the creation of the fact of being promised. The combination of a coordination act and its resulting fact, is also called a *process step*.

A more general understanding of a transaction[10], I continued, is to conceive it as an, on principle, undetermined number of interactions between two actors, one in the role of *initiator* and the other in the role of *executor*, proceeding in three phases: the order phase, the execution phase, and the result phase (Figure 2.2). The *order phase* is an *actagenic conversation*[11], in which the actors discuss and negotiate in order to come to agreement about the product to be brought about by the executor. The conversation may be as short as suggested in Figure 2.1, but it may take more steps and even last indefinitely, and it may end unsuccessfully. As said, the product is also called the proposition in this phase. In the *execution phase*, the executor brings about some product. The initiator is basically ignorant of what is going on in this phase. The *result phase* is a *factagenic conversation*, in which the actors discuss and negotiate in order to come to agreement about the result, i.e. the actually produced product. This conversation may also last indefinitely long, and it may also end unsuccessfully.

---

[10] The origin of the word "transaction" is the Latin verb "transagere", which means carrying through or out.

[11] The word "actagenic' means generating acts. The word "factagenic' means generating facts.

The coordination acts/facts in a transaction regard some product, which is represented by a predication (e.g. "is started") of an entity (e.g. "membership 1087"). A product consists of an *independent* production fact (like "membership 387 is started") and a number of *dependent* production facts, like the member of the membership. The independent production fact becomes existent as the result of a successfully completed transaction. The *dependent* production facts start to exist at the same time.

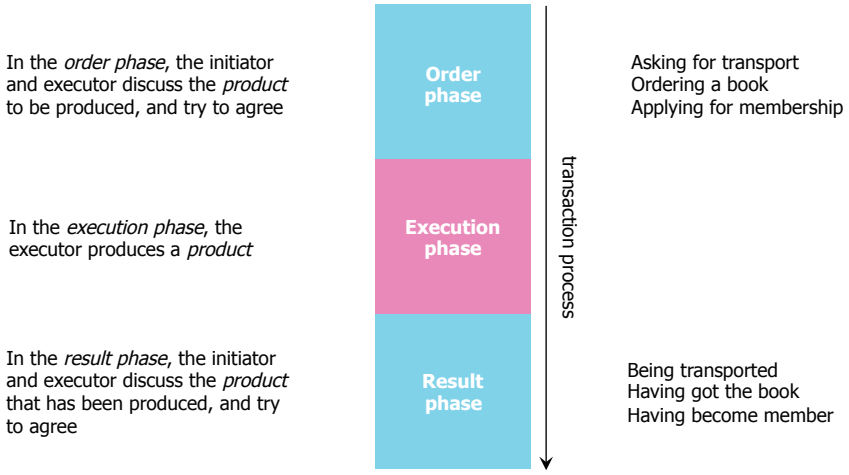


Figure 2.2 The transaction process

In Figure 2.3 (left side), the *standard transaction pattern* is projected on the three phases, for the purpose of studying the *transaction process* in detail. As the figure shows, instead of promising, the executor may decline the proposition; this is represented by the response link from (rq) to [dc]. The status declined (dc) is denoted by a double circle, to indicate that it is a *discussion status*. This is a situation in which the actors have to 'sit together' in order to elucidate the reason for deviating from the basic transaction pattern (also called the success path or happy flow) and to try to resolve the situation. If the actors come to an agreement, the outcome of the discussion will be a repeated request, presumably with a modified proposition. If they cannot, the transaction will be terminated unsuccessfully by the initiator, through a quit [qt], resulting in the unsuccessful terminal status quitted (qt), from where one returns to the initial status (in).

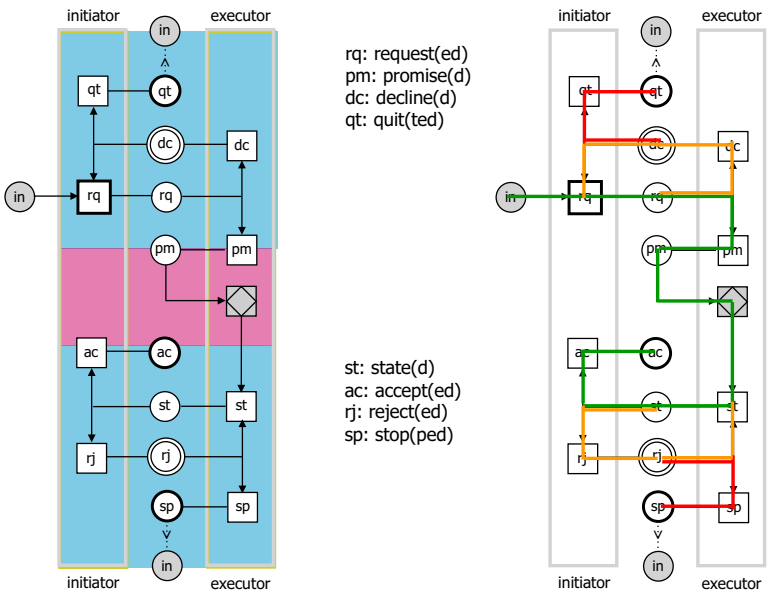


Figure 2.3 The standard transaction pattern

The result phase starts by stating the produced result by the executor. In response to this event, the initiator performs an accept [ac] or a reject [rj]. In case of an accept, the transaction ends successfully. By rejecting, the process comes in the discussion status rejected (rj). Again, the actors have to 'sit together' in order to elucidate the reason for this 'exception' and to resolve the situation. The outcome of the discussion is either a repeated state [st], meaning that the actors have resolved their divergent views, or the unsuccessful termination by the executor through performing a stop [sp], from where one returns to (in). Between the (pm) and the [st], the production act is performed, resulting in the creation of the product. Because the initiator is ignorant of the production step, this is normally not considered a separate step in the transaction process. At the right side of Figure 2.3, the possible paths through the standard pattern are indicated: the green path represents the 'happy flow', and the yellow paths the 'exceptions'. From an exception path, one may return to the happy flow. Otherwise, a red path is taken, which means failure.

During the process of a transaction, the properties of its product may change. Therefore, we speak of the *requested*, *promised*, *stated*, and *accepted product*, and of requested, promised, stated, and accepted product properties, like the production time. The order phase ends successfully if and when the most recently requested proposition (the proposed product) is promised. The result phase, and thus the complete transaction, ends successfully if and when the most recently stated result (the actually brought about product) is accepted.

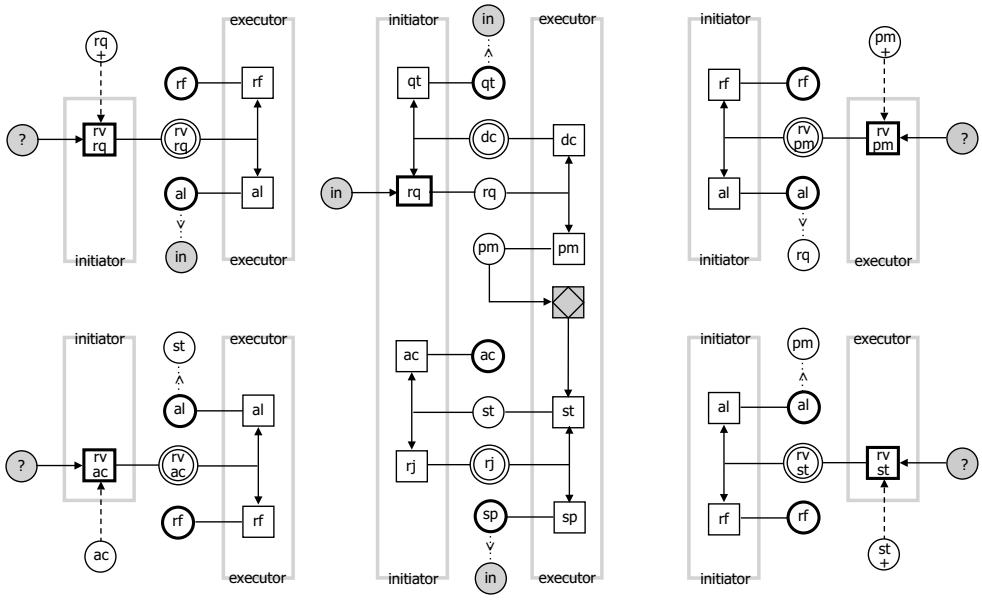


Figure 2.4 The complete transaction pattern

In addition to the standard transaction pattern, there are four *revocation patterns*, one for each of the basic process steps: request, promise, state, and accept. In Figure 2.4, they are drawn to the left and to the right of the standard pattern. Revocation means that one regrets having performed a step, and consequently

wants to make it 'undone'. The patterns are quite similar. By the act of revoking [rv] one ends up in the discussion state revoked (rv). Like we have seen for the statuses declined (dc) and rejected (rj), the two actors then have to sit together in order to discuss what to do. The result is either the allowance of the revocation [al],which means that the revocation is successful, or the refusal [rf], which means that it is unsuccessful. The effect of an unsuccessful revocation is that the main transaction process, so the progress in the stand pattern, remains unchanged. The effect of a successful revocation is that the transaction process switches to a specific status in the standard pattern. For example, by allowing the revocation of a promise [rv pm], the executor is able to perform a decline in the main process, resulting in the status declined (dc). I will not elaborate on the revocation processes, I said to the board members. Nevertheless, I commented, I had missed the option of revoking the request in the member registration procedure of Volley. It is unclear what has to be done if an aspirant member, after having submitted an application, wants to withdraw it. Based on my practical experience, the revocation processes should always be addressed explicitly. One cannot take them for granted or think they are trivial. In enterprise engineering, nothing is taken for granted or considered trivial. The standard pattern plus the revocation patterns constitute the *complete transaction pattern*.

To illustrate the complete transaction pattern, let us have a look at what the process steps in Figure 2.4 mean when I buy a bouquet of flowers at the florist's shop next to where I live, and be served by Celestine, the shop owner. So, I am the initiator and Celestine is the executor of the transaction we are going to carry out. I start the transaction by performing the request in response to some initial status, which is my evoked desire to have tulips in my house. I could say to Celestine: "I'd like to have such a bouquet of yellow tulips", pointing to a bucket of yellow tulips, which are priced € 4,95. Let us put this process step, for the sake of precision, in the so-called *normal form*:

<performer> : <intention> : <addressee> : <product>

Then we get the next formulation of my request [12]:

Alicia : request : Celestine : sale 1087 is completed

---

[12] One should not confuse the formulation of the product, "sale 1087 is completed", with the assertive sentence that this is the case.

So, I am the performer of the coordination act, and Celestine is the addressee. The intention[13] of my act is 'request', by which I express that I want Celestine to deliver a bouquet of yellow tulips to me. The product is denoted by "sale 1087 is completed", where sale 1087 is some particular instance of sale that I use as an easy reference to the buying by me and the selling by Celestine of a specific kind of bouquet (yellow tulips), for a specific price (€ 4,95), at a specific time, etc. For shopping transactions, the default value of the production time is 'asap' (as soon as possible). It means that it would not be okay if Celestine takes an unreasonable long time for delivering a bouquet, for example because she starts doing other things, or because she is serving other clients first. However, I could attach an explicit production time to my request. For example, if I would specify as the production time "tomorrow at 11h00", it would mean that I like to have the bouquet tomorrow at eleven o'clock. In practice, this is commonly called making a *reservation*, like one can make reservations for hotel rooms and theatre seats. Clearly, making a reservation is not a separate transaction. It is just and only a matter of a future production time. Next, if the requested production time lies in the past, it means that the initiator wants something to start to exist in retroaction (backdating). For buying flowers this makes little sense, but starting a subscription to a magazine, for example, with a starting day in the past, does make sense and is generally possible. It would mean that one receives also older issues. Likewise, one can start a subscription at some future time, for example, at the beginning of the next quarter.

Let us assume that Celestine responds to my request by saying "Just a moment". Such an expression would count as promising the proposition. In the normal form this is formulated as follows:

Celestine: promise : Alicia: sale 1087 is completed

During the discussion and negotiation in the order phase, all of the properties of sale 1087 may change. However, for ending this phase successfully, the promised proposition must be equal to the most recent requested proposition, as we have seen. The subsequent production act then consists of her decision to actually give me one of the bouquets of yellow tulips. This decision becomes knowable to me as soon as she states it, presumably by saying "Here you are", while handing over the bouquet to me:

---

[13] The intention of a coordination act is the disposition of the performer with respect to both the product and the addressee. It corresponds with a particular kind of commitment that the performer engages in.

Celestine: state : Alicia: sale 1087 is completed

On principle, one or more properties of the product my differ from the values of the promised product. So, I have to decide that I find the stated result acceptable. If I am satisfied, I could for example say "Thanks", which would count as performing the accept act:

Alicia : accept : Celestine : sale 1087 is completed

By this act, the transaction will be completed successfully. Through the accept act, the accepted result, that is the produced product starts to exist, at the accepted production time[14]. If this is a future point in time, it will take until then before the product becomes existent. In the case Volley, an aspirant member could e.g. have asked to become member from the first day of some future month. If the accepted production time lies in the past, the fact will become existent in retroaction (backdating), as we have seen.

Instead of promising, Celestine could have declined my request during the order phase of the transaction (Cf. Figure 2.4), e.g. because she has run out of yellow tulips (and that the bouquets I see, are reserved). I then could decide to choose another kind of flowers, which would change the requested product, or wait for new yellow tulips being available, which would change the requested production time. In both cases the order phase could be completed successfully. But I could also need to quit the transaction unsuccessfully, for instance because I really need yellow tulips now. Similarly, instead of accepting the bouquet that Celestine hands over to me, I could reject it, for example because the tulips don't look fresh. Ultimately, this could lead (after a revocation of the state act by Celestine, followed by a renewed production act and subsequent state act) to getting another bouquet of yellow tulips, or to an unsuccessful termination of the transaction. I did not elaborate on these situations during my presentation for the board of Volley. I did also not discuss the revocation patterns since that would go beyond the purpose of an introductory lecture. It is good, however, to always have the complete transaction pattern in mind when dealing with transactions. In all enterprises, and all over the world, business processes are composed of transactions that conform to the complete transaction pattern. Therefore, it is considered to be *universal*. Every step in this pattern is ontologically atomic.

---

[14] Recall (from RGGDE) that all production facts are intersubjective, which means that they are the result of a successful transaction (or derived from other production facts).

Let me get a bit more precise, I then said to the board members. What exactly is the product in the florist's shop example? Is it handing over the bouquet? Or that, plus paying for it? Or something else? It took some discussion before we agreed that the correct answer is: the transfer of ownership of the particular bouquet that I have accepted from Celestine. It is not necessary for the successful completion of this transaction that I pay for the flowers, although I may have committed myself to respond properly to a request for payment during the ownership transfer transaction. Paying would be another transaction. Of course, in a normal sales situation, there will be a payment transaction, and the completion of this transaction will normally be a precondition for completing the ownership transfer transaction.

In transactions like purchasing flowers at the florist's shop, Celestine and I do our business in a quite easy and natural way. But how exactly do we go along with each other's coordination acts? What is the underlying mechanism of entering into and complying with *commitments*? Without needing to go deep into the foundations of the PSI theory, I said to the members of the board, a brief closer look may be helpful. The distinctive property of process step kinds is their *intention*, which is defined as the disposition of the performer with respect to the product concerned and the addressee, as we have seen. For example, the intention of a request is to get the proposition promised (and eventually to have the transaction completed successfully). The intention of a promise is that the executor will bring about the proposition in due time; promising implies that all necessary conditions have been checked. The intention of a state is that the executor declares that he/she has brought about the proposition to the best of his/her knowledge. Lastly, the intention of an accept is that the initiator accepts the produced result, thereby releasing the executor from his/her responsibility in the carried out transaction. He/she remains accountable, however, if hidden failures are discovered at some later point in time. These situations can be dealt with by revocations of one or more basic process steps.

The intention of a process step is set and checked through so-called *validity claims*[15]. In performing a coordination act, or any communicative act, the *performer* raises three validity claims towards the *addressee*: the claim to justice, 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 coordination act successful. The *claim to justice* (German: Richtigkeit) concerns the validity of a coordination act in the social context of the two actors. The claim

[15] The notion of communicative act as well as the validity claims originate from Jürgen Habermas' "Theorie des kommunikativen Handelns".

is satisfied if the addressee acknowledges the authority of the performer to play the role he/she plays, like the performer acknowledges the authority of the addressee, both in the social context in which the act is performed. In our case this means that Celestine acknowledges my authority to be the initiator of our transaction (as I acknowledge Celestine's authority to be the executor). Celestine's authority is settled by the operative business laws, whereas my authority is generally legitimated by the operative civil code. An example of unjust acting is that I ask Celestine for flowers while we are sitting next to each other in the theatre. The *claim to sincerity* (German: Wahrhaftigkeit) concerns the validity of a coordination act in the context of the personal relationship of the two subjects. It is a matter of trust. So, the question is: does Celestine trust that I am sincere in my request to order flowers? Clearly, satisfying the claim to sincerity must emerge from the particular situation in which Celestine and I find ourselves. Lastly, the *claim to truth* (German: Wahrheit) is satisfied if the product does exist or if creating it leads to a lawful new state of the production world. In the case of the florist shop, this is guaranteed as long as I request for flowers, and not for books or bread, because these don't exist in the 'florist world', and as long as Celestine is able to produce the agreed upon product.

To conclude our discussion of the complete transaction pattern, I said to the board members, let me summarise its universality. First, it applies to all transaction kinds, in all lines of business. Second, it allows for production times in the future (reservation) and in the past (retroaction). Third, it includes everything that is commonly called "exception" in practice: declining instead of promising, rejecting instead of accepting, as well as revoking the basic transaction steps. Calling these situations exceptions, is a severe misnomer: nothing is more natural and human than these 'exceptions'.

At this point in time, Adam asked whether one should record all steps in a transaction process. I replied that transaction processes like purchasing flowers will normally not be recorded fully. For other kinds of transactions, however, it may be the case. In purchasing a house, for example, the order phase is normally documented in the so-called deed of conveyance. In addition, the result phase will be documented in the deed of transfer. Moreover, this may need the formal approval by a legal entity, like a notary public. It is good to keep in mind, however, that there are no fundamental differences between purchasing a bouquet of flowers and purchasing a house. The underlying mechanism of the three validity claims, and the 'social value' of the commitments, is basically the same. So, in most cases it will be sufficient to record the accept fact, including all accepted product properties.

2.2 COORDINATION ACTS AND FACTS

Let us have another look at the structure of a coordination act/fact or process step, I suggested to the board members, and discuss some time aspects in more detail. Therefore I showed them Figure 2.5. As an example of a process step, it presents the request in the transaction we have been talking about. Therefore, the delivery day is the requested production time (expressed in Julian Time).

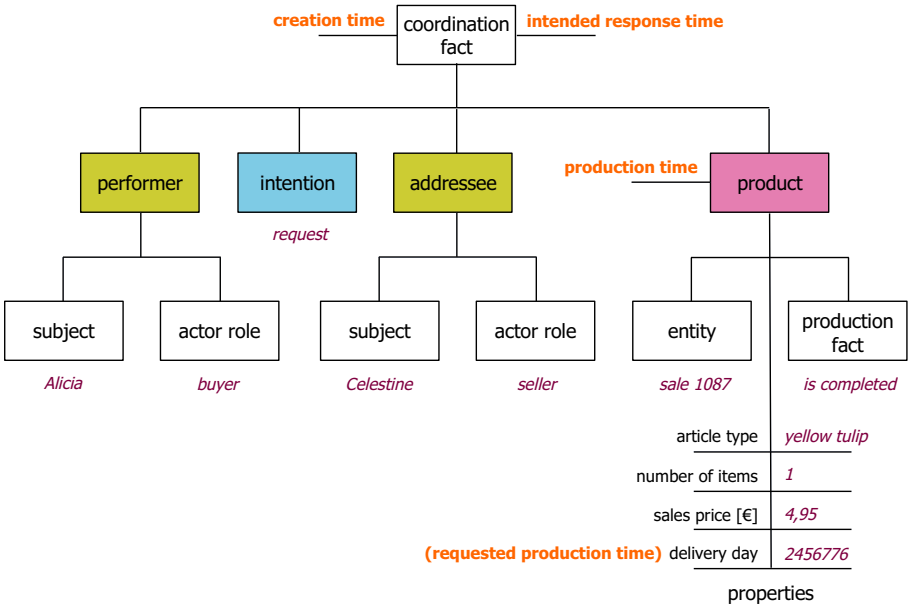


Figure 2.5 The structure of a coordination act

Next to the production time of the product, there is the creation time of the coordination fact and the intended response time. The *creation time* is the time at which the addressee of a coordination act has confirmed to have understood the act completely. I will elaborate this time aspect hereafter. The *intended response time* is the time at which the performer of the act wants the addressee to respond. In the ex-

ample, it is the time at which Alicia wants Celestine to either promise or decline. By the way, disagreeing with this time value may also be a reason for performing the decline or reject act, instead of respectively promising and accepting. So, next to specifying when one wants a product to come into existence, one also specifies, in every coordination act, when one wants the addressee to respond to the act. The difference between this point in time and the creation time of the coordination fact, is clarified by Figure 2.6, which exhibits the process of a coordination act.

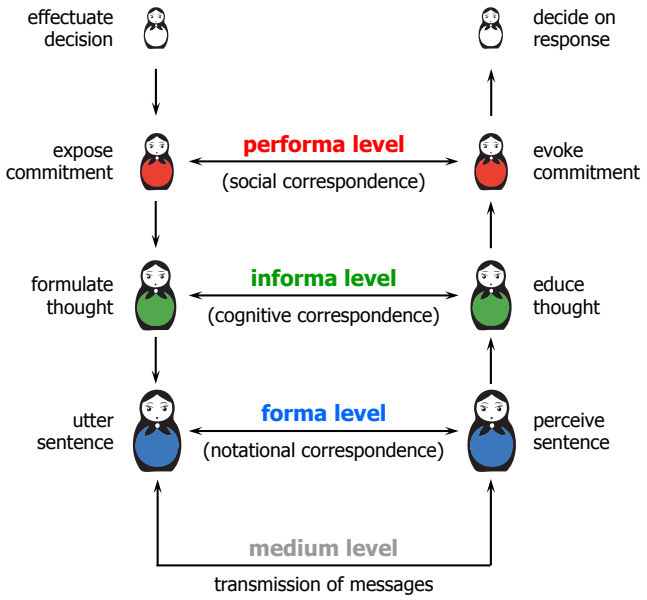


Figure 2.6 The process of a coordination act

In this process, three distinct human abilities play a role: the forma, the informa, and the performa ability. They relate to each other in a similar way as the pieces in a set of Russian Matryoshka's do. The figure shows what happens when the subject on the left communicates a commitment with the subject on the right. I will use the term "*message*" to refer in general to the unit of communication.

The lowest level in communication is the *medium level*. Here one is concerned with the physical substance in which the notational form of messages is imprinted. At this level, there is a difference between the word "gnome" written on a white board and the same word written on a piece of paper, or coded in bits and imprinted in electrical signals. The medium level also includes the transmission of messages between the physical locations of the two subjects. For example, in our flower sales transaction, Celestine and I used the air surrounding us as the medium for transmitting our spoken messages. As another example, for making a request to become member of Volley in the old days, one had to send a letter by postal mail. The medium at that time consisted of postmen carrying letters from one location to the other. The proper functioning of the medium level is a precondition for the proper functioning of the forma level. Let us assume that the medium level functions properly, so that we can disregard its intricacies.

At the *forma level* of communication, one considers only the form of messages, while abstracting from the medium. People apply their *forma* ability to utter sentences (using some medium) and to perceive sentences (from their imprints in some medium). There is no distinction anymore between the word "gnome" written on paper or projected on a screen, or spoken out etc. The prerequisites for understanding each other on the forma level, so to reach *notational correspondence*, are collectively called the *forma condition*. Satisfying this condition requires for instance that the communicating persons know the grammar of their common language. However, the content of the message is still disregarded. Satisfying the forma condition is a prerequisite for the proper functioning of the next level (informa).

At the *informa level* of communication, one considers only the content of messages, collectively constituting the cognitive meaning. People apply their *informa* ability to formulate thoughts (and subsequently express them in sentences) and to interpret thoughts (from perceived sentences). So, the focus is on the cognitive meaning of a message. For example, it doesn't matter whether I write or read "Chris owns Ketut" or "Ketut is owned by Chris" or even "Chris owns this cat" or "This cat is owned by Chris" (provided that "Ketut" and "this cat" signify the same thing). In all cases, the cognitive meaning is the same. The prerequisites for understanding each other at the informa level, that is reaching *cognitive correspondence*, are collectively called the *informa condition*. Satisfying this condition requires for instance that the communicating persons are proficient in the common language, and share the knowledge of a common production world. In the case of Volley it means that the members share notions like member, tennis, etc. Satisfying the informa condition is a prerequisite for the proper functioning of the next level (performa).

At the *performa level* of communication, one considers the intersubjective or social meaning of messages. The content of a message always comprises an intention and a product. Let me take as an example that I say to you: "Chris owns Ketut". The fact I refer to is that Chris owns Ketut, and the value of the production time is the default value 'now'. The intention of my message is to assert to you that this fact exists at this moment. If I had said: "Does Chris own Ketut?", I would refer to the same fact and time, but the intention would be to get your assertion or denial. People exert their *performa* ability to expose commitment (and subsequently formulate the thought) and to evoke commitment (from interpreted thoughts). By saying "Chris owns Ketut" I commit myself to the validity of the combination of the product, with all of its properties (like the default production time 'now'), and the formulated intention. The prerequisites for understanding each other on the performa level, so for reaching intersubjective or *social correspondence*, are collectively called the *performa condition*. Satisfying this condition requires that the communicating persons properly feel their responsibility regarding the commitments they expose, as well as the commitments they evoke in themselves.

By means of a *confirmation*, the addressee expresses towards the performer that they have reached social correspondence. By means of a *disconfirmation*, the addressee expresses towards the performer that they have not reached social correspondence (which may be caused by a lack of correspondence at one or more other levels in Figure 2.6). The performer can then try to compose and transmit another message. At the point in time that the addressee confirms, the coordination fact comes into being (starts to exist). Put differently, this point in time coincides with the creation time of the coordination fact (Cf. Figure 2.5).

Reaching social correspondence is crucial in transactions. For example, if I request Celestine to sell me a bouquet of flowers, I must feel responsible for having made the request. I must feel that I am committed to my request. It would be socially incorrect if, for example, I would say later on that I was only joking. To illustrate the importance of social correct behaviour: in most national civil codes (laws), having made a request counts as having entered into an agreement, which has legal consequences. To continue the example, Celestine must feel that she has to respond adequately to my request, which is either to promise or to decline, depending on the degree in which she accepts the validity claims I have raised. A similar reasoning holds for all other steps in the complete transaction pattern.

The ability and readiness of its employees to perform coordination acts, in response to coordination events, is the *operating principle* of every organisation, and, I dare say, of society as a whole. I see it in

this way. As Figure 2.6 exhibits, we take on three different shapes in performing a coordination act: the red shape, the green shape, and the blue shape. Inside my blue shape is my green shape, and inside my green shape is my red shape. To continue this analogy of the Russian dolls, inside my red shape is my blank shape, my most inner self. There reside my wisdom and my love, from which I decide how to respond to the commitments of others towards me. I have put this understanding of human interaction in the verse[16] below.

Me

You
We

In my blue me
I hear the words
that you speak

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

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

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

I promise my help
in my red me

I form the thought
in my green me

I utter the words
in my blue me

and hope that you will hear me

For me, this verse is the philosophical core of the PSI theory, the ultimate explanation of the operating principle of organisations. It includes the validity claims that must be satisfied in order to make coordina-

[16] Partly inspired by Blaise Pascal's quote "We know the truth not only by the reason, but also by the heart".

tion acts successful. The line “I believe you” validates the claim to truth. It expresses my conviction that my proper response to the coordination act will satisfy a true need of the other person, and also that it is feasible to do so. The line “I trust you” validates the claim to sincerity. It expresses that I consider the performer of the act to be sincere in asking me for help, that he/she really wants my help. The line “I will help you” validates the claim to justice. It expresses that I consider the one who addresses me, to be authorised to ask me for help. As a last comment, the line “and hope that you will hear me” does not only apply to the medium level of communication. The verb “hearing” should now be understood as the being receptive of the other person to my response in all shapes, including the blank one. Passing through all levels in Figure 2.6, is the only way human beings can communicate. From a ‘technical’ point of view, it is quite cumbersome, because so many things can go wrong. Sure, but we will have to live with it.

But, Adam said, it seems that in our process of becoming member of Volley, we skip process steps. For example, if I receive a letter from an applicant, I take this as performing the request, but I never promise! So, how is it possible that the process continues? Thanks, Adam, I replied, this is a clever remark. Well, according to the PSI theory, the transaction process cannot continue if the promise is not performed. So, what is going on here? Let me take the florist’s shop example to illustrate it. Instead of promising explicitly, as she did, Celestine could have said nothing, but subsequently walk to the bucket of tulips to fetch a bouquet, so acting as if she had promised. If this happens, then she has performed the promise, but tacitly. It would be socially not just (claim to justice) if Celestine, after having fetched a bouquet from the bucket, addresses me, only to say that she will not sell flowers to me, so performing the decline. Yes, indeed, coordination acts can be performed tacitly. In addition, they can be performed non-verbally. This means that there is an observable act, and this act must be considered to count as performing the coordination act. For example, after having decided that the fetched bouquet will be mine, Celestine could just hand over it to me, without saying anything. This handing over then would count as performing the state. Similarly, I could take the bouquet without saying anything. Then, this act would count as performing my acceptance. It took some time for the members of the board to grasp the idea of tacit acts; the non-verbal ones were easier.

It is interesting to notice, I continued my story, how the presence of transactions can be ‘blurred’ by the technology with which the medium level is implemented (Cf. Figure 2.6). To illustrate this, have a look at the next two different implementations of withdrawing money from a bank account. Nowadays, people would go through a process like this one:

ATM: Welcome to the ABC bank. Please insert your card
Client: < inserts card into the slot >
ATM: Enter your PIN please
Client: < keys in PIN >
ATM: Select the amount you want to withdraw
Client: < presses the button for € 100 >
ATM: Take your card please
Client: < takes out the card >
ATM: Your money is being counted
ATM: < pushes banknotes through a slot >
ATM: Take your money please
Client: < takes the banknotes and leaves >

At the time when there were no ATMs, one had to go to a bank office and present oneself to a bank employee at a counter. This is what could happen then:

Employee: How can I help you Mr. Bean?
Client: I want to withdraw money from my account
Employee: Your account current, I assume?
Client: Yes, of course
Employee: How much do you want?
Client: One hundred euros
Employee: < fills out a form >
Employee: Can you sign here please
Employee: One moment please
Employee: < hands over the bank notes to the client >
Employee: Here you are, Mr. Bean. Have a nice day.
Client: Thank you

Clearly, the last 'story' comprises a transaction. The request is built up in lines 1 thru 8. The promise is performed in line 9, the state in line 11, and the accept in line 12. Now, the important thing to understand, is that exactly the same transaction is carried out when one gets money from an ATM. Only the imple-

mentation of the steps differs (As an exercise, figure out where the request, promise, state, and accept are in the ATM case). This may surprise you, and I see it does, but it should not. Understanding the implementation independent meaning of transactions is key to understanding the essence of organisations. The additional blurring point, is that one doesn't seem to interact with a human actor. But actually, one does, although this person is unknown. This will turn out to be the case if something goes wrong, for example, if one gets less money from the ATM then was requested (and presumably debited). In such a case, one has to go to the bank office, to meet there the mister or miss 'mystery' who played the executor role.

2.3 PRODUCTION ACTS AND FACTS

After this explanation I resumed the relevance of the three human abilities (performa, informa, forma) for production acts and facts, as discussed in RGGDE. So, in our performa shape, we are able to perform *original production* acts, resulting in original new products. Regarding immaterial products, we normally speak of devising or deciding or judging. Regarding material products, we speak of manufacturing, transporting, observing, etc. In our informa shape we are able to perform *informational production* acts, such as remembering facts and recalling them, and deriving facts from existing ones. The important distinction between original and informational production is that only original production really creates facts that didn't exist before. Informational production, on the contrary, never creates new facts. This is quite obvious for remembering and recalling facts, but it also holds for computing. The outcome of a computation is a different view of the world, but it is still the same world. For example, if Celestine computes at the end of the day the total revenue of the day, she does not create a new fact. In a very literal sense, one can say that the fact was already there, because it was defined. It only had to be computed! In our forma shape we are able to perform *documental production* acts, i.e. to act on the form of information, while fully ignoring its content. Possible documental production acts are saving, providing, and transforming sentences (or clusters of sentences, called documents). Note that on the physical level, these acts would lead to storing, retrieving, transmitting, and copying messages or files.

The distinction between original, informational, and documental production leads to the distinction, in every organisation, between three *aspect organisations*. Like it was done in RGGDE, we distinguish between the O-organisation (O for original), the I-organisation (I for Informational), and the D-organisation (D for documental). The I-organisation does only and fully support the O-organisation through

providing informational services, as the D-organisation does only and fully support the I-organisation, through providing documental services. The three aspect organisations are exhibited in Figure 2.7. The meaning of the cone shape is twofold. First, it expresses that there is nothing 'above' the O-organisation. Second, the volumes of the red, the green, and the blue part, roughly correspond with the number of transaction kinds and actor roles in the respective aspect organisations. The ratio of the volumes of the three parts of the cone, so the O-organisation, the I-organisation, and the D-organisation, is 1:4:7.

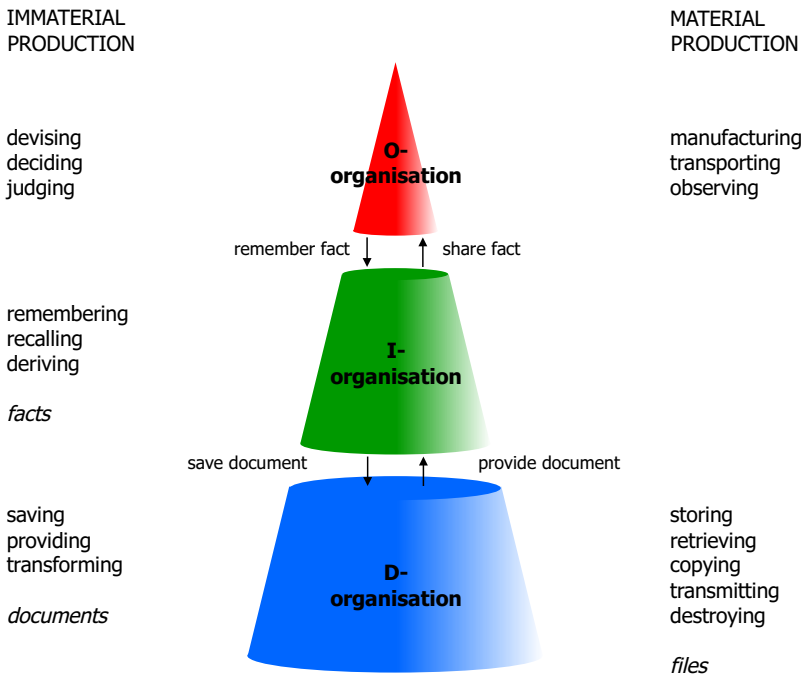


Figure 2.7 The three aspect organisations

The division of an organisation into its three aspect organisations provides us with another means to reduce the complexity of organisational models, in addition to applying the universal transaction pattern for

understanding the processes in an organisation. Because the essence of an enterprise is brought about in its O-organisation, one can focus on the O-organisation first. If and when needed, one can study the I-organisation, which supports the O-organisation by providing informational *services*[17], as one can subsequently study the D-organisation, which supports the I-organisation by providing documental services. The I-organisation and the D-organisation of an enterprise, are collectively called the *realisation* of its O-organisation. As Figure 2.7 exhibits, the average ratio in size between the O-organisation and the supporting I-organisation and D-organisation together, is 1:11. In practice, one may indeed confidently assume that the number of transactions and the number of actors in the O-organisation, is not more than 10% of the total number of transactions and actors, in every organisation. However, because the universal transaction pattern introduces (business) process steps that are normally ignored, like the decline and reject 'exceptions', and the four revocation patterns, it is safe, in my experience, to say that the reduction of complexity by abstracting from the realisation of a O-organisation, is about 80%. By abstracting from the implementation of an enterprise's O-organisation, that is from the informa and forma level (and the medium level) in coordination, and from the technological means in performing production acts, another 80% of reduction of complexity is achieved in most organisations. Combined, this mounts up to a reduction of complexity of well over 95%.

Next, I briefly discussed the following topic concerning actors. Every subject (employee) in an organisation may be allocated to quite a number of actor roles concurrently. For example, he or she may fulfil one or more roles in the O-organisation, one or more in the I-organisation, and one or more in the D-organisation. This is all a matter of differentiation and specialisation, and of the size of the organisation of course. As an example, Celestine will fulfil quite some actor roles, because she is the only employee in the florist's shop, except for a few temporary assistants. In addition, a subject, in holding an original actor role (that is, being the executor of an original transaction kind), may take on its green shape to initiate an informational transaction (with an informational actor role), or its blue shape to initiate a documental one (with a documental actor role). Similar comments hold for subjects that fulfil informational or documental actor roles. At the same time, many subjects may fulfil the same actor role simultaneously. Think for example, of all employees in a sales or a purchase department. Lastly, some actor roles need to be fulfilled by the collectivity of a number of subjects. An example is the members of the board of Volley, I told them. Only the collective members (or a majority) has the authority to make particular decisions. Of

[17] A service is the functional (or business) interpretation of a product.

course, the board members fully agreed. Another example is the general assembly: only the collective members of Volley (or a majority) has the authority to make particular decisions, e.g. to decide on next year's membership fee.

Next, as will be elaborated in Chapter 4, in general every transaction is enclosed in another one, and it may enclose a number of other transactions. In other words, transactions occur in tree-like structures, which we will call business processes, that reflect the product structure. Regarding material production (manufacturing), such a tree structure reflects the Bill of Materials (BoM) of an end product. However, similar structures appear to exist in non-material production.

To end this chapter, let me summarise the elements that constitute the essence of an organisation. Foremost, there are the people, the employees of an enterprise if you like, who, in mutual interaction and in interaction with the environment, constitute the organisation of the enterprise. Then *first*, we applied the distinction between the performa, the informa, and the forma ability of people (subjects) to coordination. By abstracting from the forma level (including the medium level) and the informa level (Cf. Figure 2.6), we concluded that the essence of coordination is in the commitments subjects enter into and comply with. In addition, we have seen that our performa shape includes our most inner shape, that is our 'blank me'. *Second*, we applied the performa-informa-forma distinction to production. By abstracting from the supporting services of the D-organisation, and subsequently by abstracting from the supporting services of the I-organisation, we concluded that the essence of production is in the O-organisation of an enterprise: the bringing about of original new things (Cf. Figure 2.7). *Third*, production and coordination appear to occur in processes that follow the universal pattern of the transaction (Cf. Figure 2.4). *Fourth*, transactions always occur in tree-like structures, commonly called business processes.

3 REVEALING ORGANISATIONAL ESSENCE

3.1 ANALYSING NARRATIVE DESCRIPTIONS

After a short break, in which the members of the board could ask their final questions regarding the PSI theory, I started to present my analysis of the processes in Volley, based on the narrative description by Mr. X (see Chapter 1). The goal of the analysis is to identify the O-organisation transaction kinds and actor roles. In order to achieve this goal, one must abstract from all realisation aspects of an organisation, i.e. from the services offered by the I-organisation and subsequently of the D-organisation, as well as from the informa and forma level of communication. In addition, one must abstract from all implementation aspects, i.e. the technologies with which production and coordination acts are performed, as well as the specific subjects that fulfil the actor roles. Let me illustrate what I mean to say, by using the narrative description of Mr. X, and indicate in the text with the colors red, green and blue, the original, informational, and documental production, as well as the performa, the informa, and the forma level in coordination. This is the result of my exercise:

One can become member of the tennis club Volley by sending a letter to the club by postal mail. In the letter one has to mention one's surname and first name, birth date, gender, telephone number, and postal mail address (street, house number, zip code, and town). Adam, the administrator of Volley, empties the mailbox daily and checks whether the information provided is complete. If not, he makes a telephone call to the sender in order to complete the data. Once a letter is complete, Adam writes an incoming mail number and the date on the letter, records the letter in the letter book, and puts it in a folder.
Every Wednesday evening, Adam takes the folder to Eve, the secretary of Volley. He also takes the member register with him. If Eve decides that an applicant can become member of Volley, she stamps 'new member' on the letter and writes the date below it. She then hands the letter to Adam in order to add the new member to the member register. This is a book with numbered lines. Each new member is entered on a new line. The line number is the number by which the new member is referenced in the administration.

Next, Eve calculates the fee that the new member has to pay for the remaining part of the calendar year. She asks Adam for the annual fee, as decided at the general assembly, which Adam has recorded on a sheet of paper. Then, she asks Adam to write down the amount in the member register.
If Eve does not allow an applicant to become member (e.g., because he or she is too young or because the maximum number of members has been reached), Adam will send a letter in which he explains why the applicant cannot (yet) become member of Volley.
When all applications are processed, Adam takes the letters and the member register home and prepares an invoice to all new members for the payment of the first fee. He sends these invoices by postal mail. Payments have to be performed by bank transfers.
As soon as a bank statement is received, Adam prints a card on which the member number, the starting date, the name, the date of birth, the gender, and the residence are mentioned. The card is sent to the new member by postal mail.

After having presented this 'colorful' analysis, I started to clarify and interpret it, as follows. The part '*become member of the tennis club Volley*' obviously represents an original product. Although '*sending a letter*' certainly refers also to a documental action, the most important meaning is that it is a coordination act at the performa level: it is the request to become member. As we know from the standard transaction pattern (Figure 2.3), the response to a request is either a promise or a decline. The promise is apparently represented by '*decides that an applicant can become member of Volley*', and the decline by '*will send a letter in which he explains why the applicant cannot (yet) become member of Volley*'. The part '*the payment of the first fee*' obviously represents also an original product. Although '*sends these invoices*' certainly refers also to a documental action, the most important meaning is that it is a coordination act at the performa level: invoices are requests to pay. The part '*Payments have to be performed*' additionally indicates the second original product kind we have found. The part '*a bank statement is received*' represents the statement of the payment transaction, and the part '*is sent to*' represents the statement of the transaction in which one becomes member. For both parts it also holds that they exist at the informa and forma level, but the performa level is the most important one. It should not be a big problem for you to agree with the parts of the text that are coloured green and blue. They are clearly about informational production or the informa level of coordination, and about documental production or the forma level of coordination respectively.

From the analysis so far, two original product kinds can be identified: “Membership is started” and “the first fee of Membership is paid”. In these specifications, a word starting with a capital (in this case: Membership) is a placeholder for individual instances. An example of an instance of Membership is “membership 1087”. Let me refer to the two product kinds, for the sake of convenience, by “P1” and P2” respectively. Note that the word “membership” does not appear in the narrative description. Coming up with the notion of membership is the result of applying the sapience ‘devising proper concepts’ (Cf. Figure 1.1). One can do without the notion of membership, but one would not be able then to easily distinguish between two or more disjoint periods in which the same person is member of Volley. Therefore, the general rule for formulating product kinds is to find a fact kind (in our case membership) of which the instances are uniquely identifiable in space and time, where space is the state space of the production world.

The instances of P1 and P2 are the products of transactions of the kinds “membership start” and “membership payment” respectively. I will refer to these transaction kinds by “T1” and “T2” respectively. By convention, a transaction kind and its executor role are referred to by the same number, like transaction kinds and corresponding product kinds are. So, A1 is the executor role of transaction kind T1, and A2 of T2. During the process of carrying out a T1, the corresponding T2 is initiated. Therefore, the T2 is said to be enclosed in the T1. It implies that A1 is an initiator role of T2. Let us refer to the initiator role of T1 by A0. It is an actor role of which we do not know more than that it must exist.

Let us verify our conclusions, and make them more precise, by considering the process of becoming member of Volley of a particular person, Anna, and consider the possible interactions between her and Eve[18]. Apparently, Eve fulfils actor role A1, and Anna fulfils A0 as well as A2 (Note: for the sake of simplicity, we disregard the roles of Adam, and we allow payments to be made in cash). Below, after every uttered coordination act, its normal form is presented between brackets. The progress of the transaction process is shown in a self-explaining formulation (written in italics). For the sake of clarity, I will add the actor role to the performer and the addressee of the process steps, and I will refer to the particular membership by “1087”.

Anna : I would like to become member of Volley, as soon as possible
(A0/Anna : request : A1/Eve : membership 1087 is started, with production time is asap)

---

[18] This an example of applying the sapience ‘verification by instantiation’ (Cf. page 11).

Eve : I am happy that you have chosen for Volley, the best tennis club there is!
(By this expression, Eve confirms that she has understood Anna; it does not count as the promise!)
membership start for membership 1087 is requested *(T1/rq)*

From the facts in the current state of the production world and the coordination world of Volley, Eve finds out that on the first day of the next month (which is the default starting day of memberships), Anna will be 12 years old (which is the current minimum age), and that the maximum number will not be exceeded if she allows her to become member.

Eve : I will see to it that you become member as per the first day of the next month
(A1/Eve : promise : A0/Anna : membership 1087 is started, with production time is the first day of the next month)

Anna : Great
(By this expression, Anna confirms that she has understood Eve's promise)

membership start for membership 1087 is promised *(T1/pm)*

Next, Eve computes that the first membership fee to be paid is € 75 (Note that this is an informational transaction). Then she addresses Anna in Anna's role A2, the executor of T2.

Eve : You have to pay the fee for the remainder of this calendar year, which is € 75.
(A1/Eve : request : A2/Anna : the first fee of membership 1087 is paid, with the amount paid is € 75, and the production time is asap)

membership payment for membership 1087 is requested *(T2/rq)*

Anna : I will do it right now
(A2/Anna : promise : A1/Eve : the first fee of membership 1087 is paid, with the amount paid is € 75, and the production time is now)

membership payment for membership 1087 is promised *(T2/pm)*

Anna takes € 75 out of her wallet and hands it over to Eve. Performing this act presupposes that the (not directly knowable) production act is performed: T2(1087) is executed.

Anna : Here you are
(A2/Anna : state : A1/Eve : the first fee of membership 1087 is paid, with the amount paid is € 75, and the production time is now)

membership payment for membership 1087 is stated *(T2/st)*

Eve : Thanks
(A1/Eve : accept : A2/Anna : the first fee of membership 1087 is paid, with the amount paid is € 75, and the production time is now)

membership payment for membership 1087 is accepted *(T2/ac)*

Eve can now decide to create membership 1087, because the condition of the being paid of the first fee has been satisfied. Put differently, she proceeds to perform the production act in T1, which, again, is not directly knowable to other persons: T1(1087) is executed.

Eve : Well then, welcome as member of Volley, as per the first day of the next month
(A1/Eve : state : A0/Anna : membership 1087 is started, with production time is the first day of the next month)

membership start for membership 1087 is stated *(T1/st)*

Anna : Thanks
(A0/Anna : accept : A1/Eve : membership 1087 is started, with production time is the first day of the next month)

membership start for membership 1087 is accepted *(T1/ac)*

From this exercise we learn that if one focuses on the O-organisation of an enterprise, one can straightforwardly identify the core business processes in an enterprise. They are composed of transaction kinds and actor roles in the O-organisation (Recall that we abstract from all enclosed informational and documental transaction kinds and actor roles that realise these business processes).

3.2 ANALYSING STRUCTURED DESCRIPTIONS

Next, I proposed to discuss the Flow Charts as produced by Mr. Y, for clarifying the analysis we had made. All members of the board agreed. So, I started to use the Flow Charts to elaborate on the distinction between original, informational, and documental production. One can easily do this by going through all actions (represented by boxes) in the Flow Chart and asking oneself the next questions. Is it a production act? And if so, is it original, informational, or documental? Or is it a coordination act? And if so, does it represent only the forma level, or also the informa level, or also the performa level? In order to involve the attendants actively, I let them, in rotation, answer these questions while going through all actions in the Flow Charts. The results are shown in Figures 3.1, 3.2, and 3.3. The following legend applies. The boxes are actions, and the shapes with an undulated underside are documents. An arrow indicates that the action on the shaft side precedes the one on the point side. A diamond represents a choice between two or more flows and the 'sausages' are just connectors between separate parts of the Flow Chart.

At first glance, one may be inclined to classify the first action in Figure 3.1 as a documental one. The point is that it is indeed a documental action. Transporting the application letter from the applicant to Volley is even a complete documental transaction, executed by a postal mail company, but this aspect of the action is not the most important one. With reference to Figure 2.6, we are dealing with the medium and forma level of a coordination act in the O-organisation, although it is labeled by its appearance at medium and forma level (receiving application letter). As has become clear from the analysis of the narrative description, the sending of an application letter by somebody and the subsequent receiving of the letter by Adam, counts as performing the request in a transaction of the kind T1. Next, checking the completeness of a set of information items and asking for additional information (in the second and the third box) are purely informational actions, namely the sharing of facts that do exist already. Clearly, the other three boxes represent documental actions. They are not about the contents of the documents but about their documental processing (recording, archiving, etc.).

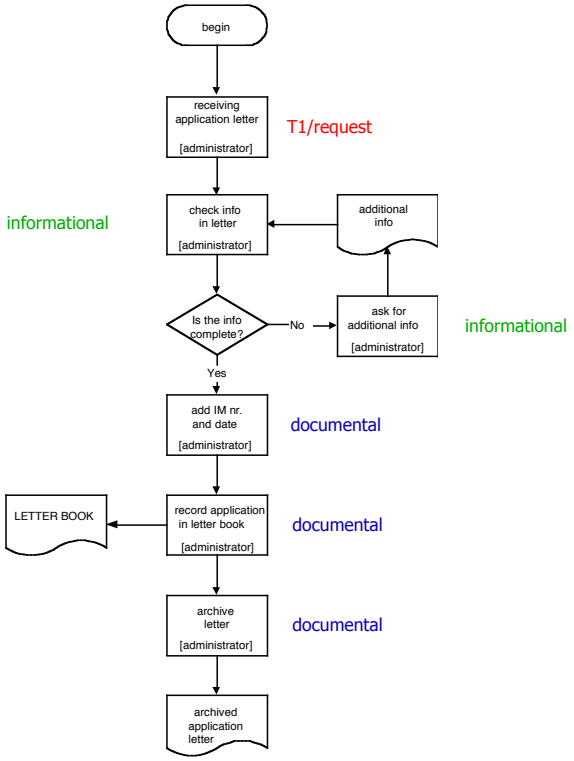


Figure 3.1 Analysis of the Flow Charts of Volley (1)

Regarding the analysis as presented in Figure 3.2, some boxes may need more explanation. The first action may look like performing the production act, also called the *execute* act [T1/ex], but this cannot be the case yet because there has not been a promise. Therefore, we must conclude that it is performing either [T1/pm] or [T1/dc]. Next, making a refusal letter, in order to perform the [T1/dc], is primarily informational. More precisely, it is the formulation of the thought in Figure 2.6. Sending the letter counts as actually performing the [T1/dc], although it is also a documental transaction. Lastly, calculating the

membership fee is purely informational. Because membership fee is a defined fact type, its value is ontologically always present; it only has to be computed, when needed.

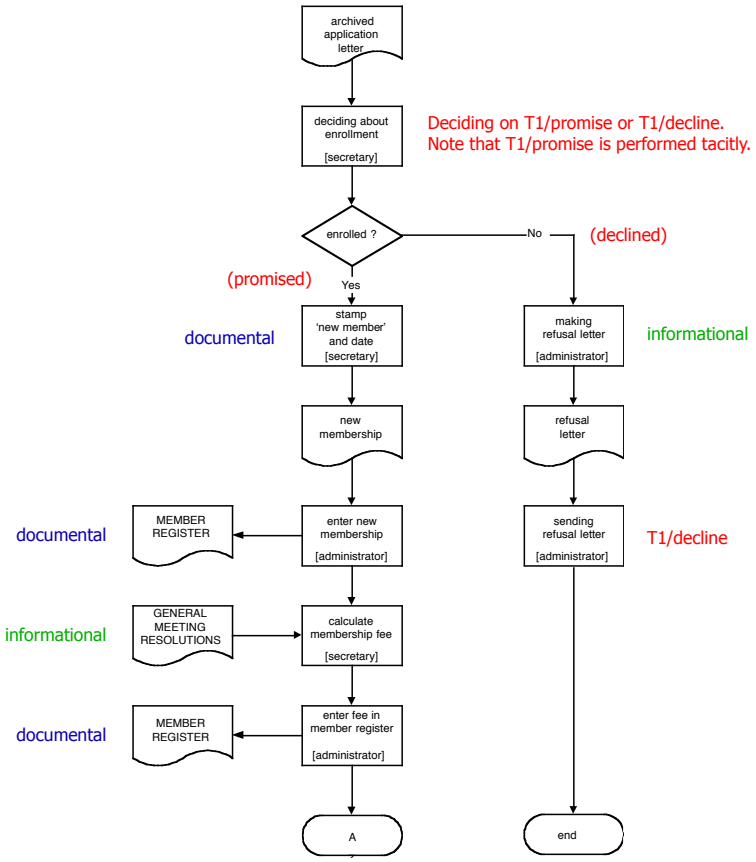


Figure 3.2 Analysis of the Flow Charts of Volley (2)

Regarding the third part (Figure 3.3), the next explanation holds. Like I said with regard to the action of making a refusal letter (Figure 3.2), making the first invoice is also primarily informational. Sending the invoice, together with the subsequent receiving, however, counts as performing [T2/rq]. A similar reasoning holds for making the membership card. Sending the card counts as performing [T1/st]. Lastly, the performa level meaning of receiving a copy of the payment, together with the preceding sending of the copy, counts as performing the [T2/st].

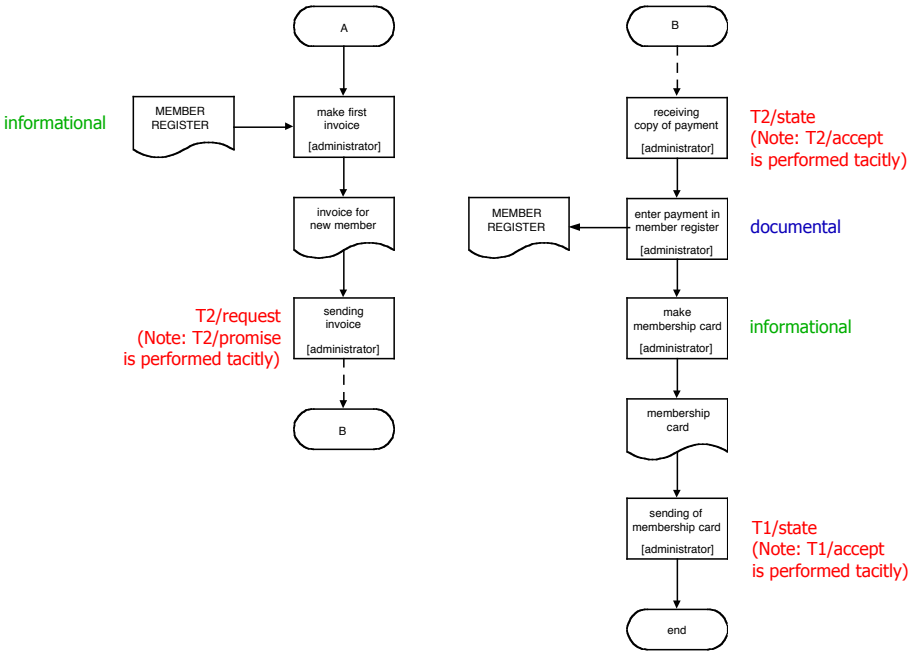


Figure 3.3 Analysis of the Flow Charts of Volley (3)

Eve then raised the question where and how the production act [T1/ex] of the Volley case must be modelled. I replied to her that by definition this act is 'invisible' (Cf. Figure 2.3). The execution act and the resulting fact are fundamentally not knowable to persons other than the executor. We only know that it must take place after the [T2/ac]. As soon as the state act [T1/st] is performed, it is certain that the execute act has been performed, but when exactly, one will never know.

There was no real problem for the members of the board to understand and agree with my analysis of the Flow Charts, except for the classification "tacitly performed", which occurs no less than four times: [T1/pm], [T1/ac], [T2/pm], and [T2/ac]. Most of them considered these acts irrelevant; one could do without them. The point is, I said, that these acts must have been performed for every successful membership application, since they are part of the basic transaction pattern. Therefore, tacitly performed coordination acts must be conceived as being performed, however according to the general practical rule "no news is good news". This rule is applied on a large scale, in many enterprises and all over the world. At the same time, it is also quite harmful. It happens to be the cause of many failures and process hiccups, and of many misunderstandings. The original reason for applying the rule, and the only acceptable justification in practice, is that the costs of the available communication technology are too high. For example, in response to the application letter of an aspirant member, Adam does not send back a letter for telling that the application has been promised. That would take too much time and would cost too much money. He only does so for the decline (by sending the refusal letter). The "no news is good news" rule implies that the aspirant member may infer that the promise has been made if he/she does not receive the refusal letter within a reasonable time, let's say within 10 days. The members of the board could fully agree with my explication; they were happy to have learnt that one should not take everything for granted. The promise and the accept act, I continued, are typical examples of acts that are often performed tacitly. Fortunately, modern ICT allows for performing all coordination acts explicitly, leaving no doubt about the actual status of a transaction for both participants. If nowadays one orders something from a web shop, the whole transaction process is explicit and transparent. Why? Because there are virtually no communication costs.

The second analysis reinforced our findings of the first one. All members of the board agreed on this observation, and some of them liked the second one better than the first one. I can understand this; structured business process descriptions, like Flow Charts, BPMN and UML diagrams, ARIS/EPCs and Petri Nets, are generally more welcomed by analysts than narrative descriptions. Still, a serious warning is in

place, again regarding the tacitly performed coordination acts. I was able to find them, I clarified, because I had the template of the complete transaction pattern in my mind. Without it, I would have overlooked them, as all modellers and analysts using current business process modelling techniques do. However, as one of the attendants rightly pointed out, this defect in current practice can easily be put right by applying the universal transaction pattern when modelling in one of those current approaches. I could only agree with her, and I mentioned several DEMO Masters, whom I personally know, who have to conform to some industry standard for modelling business processes, but apply the PSI theory as their way of thinking. This comment inspired the chairman to saying "Think PSI, and speak the language in force". I can very well live with this point of view. Although DEMO can replace many current approaches, it can also be viewed as complementary and as an enrichment, since DEMO is the only approach that produces truly ontological models, and that is built on a sound theoretical basis.

3.3 THE OER METHOD

The method I have applied, is called the OER method. The word "OER" is an acronym for Organisational Essence Revealing, but it is also a Dutch word, meaning primal and primeval[19]. This meaning conveys the core idea that one has to go to the 'original shape' of a Scope of Interest in order to discover its essence. By this original shape is understood the image of the Scope of Interest in which all applied communication and information technology is removed. Put differently, there is only face-to-face communication between actors, and the remembering and computing of facts is done only mentally. This helps enormously in applying the OER method, which consists of these three stages:

1. Identify the O-organisation coordination acts/facts and production acts/facts.
2. Identify the transaction kinds in which these acts/facts occur.
3. Identify the tree structures in which these transaction kinds occur.

As an analogy, in the first stage, one discovers the *atoms* of the 'matter' under investigation. In the second stage, the *molecules* are discovered that are constituted by the atoms. In the third stage, the *polymers* or *fibers* are discovered that are constituted by the molecules.

---

[19] In English the, originally German, combining form "ur-" exists, like in urtext. It has the same meaning as "oer".

The preferred starting point for applying the OER method, is that one has interviews with all people within the Scope of Interest, structured by these questions: What kind(s) of original facts do you produce? Who is requesting you to do this? and Whom do you request to produce original facts for you? Note that these questions are a guidance, you should not ask them literally, because one wouldn't understand you. Instead, you must talk 'their language' and then translate your findings to the DEMO language. Based on the answer(s) to the first question, you are able to identify the transaction(s) of which the interviewed person is the executor, as well as to formulate the product kind(s) in such a way that it is uniquely identifiable in time and space. Based on the answer(s) to the second question, you find the initiator(s) of the identified transaction kind(s). If you have interviewed them already, you can check whether his/her information matches what you have found up to this moment, and discuss mismatches. If not, you will be able to do this later. Based on the answer(s) to the third question, you can check whether his/her information matches what you have found up to this moment, and discuss mismatches. If not, you will be able to do this later. Your investigation has a natural ending in the border transaction kinds, thus the transaction kinds of which one of the roles (initiator or executor) is in the environment of your Scope of Interest.

If interviewing is not possible (for whatever reason), a good alternative is the analysis of documents in which the business processes are described, preferably as detailed as possible[20], like the narrative description and the Flow Charts we have about Volley. Regardless the starting point, one must always also apply the sapiences 'verification by instantiation' and 'validation from ontology' in order to make sure that all findings are checked, and that nothing is overlooked, like I have done for the case Volley. Validation entails in particular that one checks how all transaction steps, in the complete transaction pattern, are performed. This is still no guarantee that one has found all relevant transaction kinds, but the chance that one has completely overlooked a transaction kind is very small, as practical experience shows.

Lastly, although the three stages of the OER method logically follow the order in which they are listed, it is quite common among experienced enterprise engineers that one reiterates the sequence, and even that one jumps from one instantiation of a stage to an arbitrary other one. By the way, the word OER is also, and not incidentally, an acronym for the three phases in a transaction process: Order phase, Execution phase, and Result phase. Adam liked this non-coincidence.

---

[20] A very valuable kind of documents are the ISO 9000 files.

4 REPRESENTING ORGANISATIONAL ESSENCE

4.1 THE NOTION OF ESSENTIAL MODEL

When I was near the end of my analysis of the case Volley, I noticed that several members of the board had a hard time to digest the new ideas I was feeding them. Therefore, I proposed to schedule another meeting for presenting the ontological model of Volley. They all welcomed the suggestion. It strengthened my conviction that the members of the board were really interested in my story, and maybe not only because they saw the problems at Volley being addressed in the right way. I am pretty sure that their thoughts also dwelt towards the organisations they are working in, where some of them hold influential positions. I was pleased by this observation. After all, Volley is a toy example, quite appropriate for explaining the PSI theory and for illustrating the models of DEMO, but the real excitement and conviction of the power of enterprise ontology can only arise from large scale applications in practice, as I knew very well.

I started the next meeting with asking the members of the board to join me in addressing the question of how the idea of the essence of an organisation, as discussed in the previous chapters, can be represented in a practically useful way. To this end, I want to introduce the notion of ontological model to you, I said. The *ontological model* of an (aspect) organisation is the understanding of the organisation as a network of transaction kinds and actor roles, while abstracting from their implementation. Conversely, *implementing* an organisation consists of assigning appropriate technological means to the elements of its ontological model. Regarding actor roles, the 'implementation technology' is basically the allocation of subjects to them. Regarding coordination acts, one has to select particular formulations of thoughts (informa level), particular expressions of these thoughts in sentences (forma level), as well as particular ways of imprinting sentences in physical substances, and particular media for transmitting messages (postal mail, e-mail, etc.). Regarding production acts, various technologies may be applied, depending on the character of the production. For all three aspect organisations, it holds that their ontological model is the right starting point for validating its current or proposed implementation (Cf. Figure 1.1). The ontological model of an (aspect) organisation is divided into four sub models: the Construction Model (CM), the Action Model (AM), the Process Model (PM), and the Fact Model (FM).

Let me start with presenting and discussing the Construction Model, since it is the 'top' model. The complete transaction process, as represented on the left side of Figure 4.1, has been 'squeezed' into one symbol on the right side: a diamond, standing for production, embedded in a disk, standing for coordination. The box to the left represents the initiator role of transactions of the pictured kind. The line between the actor role shape and the transaction kind shape denotes an *initiator link*. The box to the right represents the executor role of transactions of the pictured kind. The line between the actor symbol and the transaction symbol denotes the *executor link*; it is marked by the small solid diamond on the edge of the executor role box. The picture represents the basic construct in construction models of organisations. So, one must conceive the transaction symbol as representing the complete transaction pattern, concerning a particular product kind. Although every transaction kind has exactly one actor role as its executor role, it may have several actor roles as initiator role. Conversely, every actor role can be initiator role of a number of transaction kinds, and their executor roles can be initiator role of yet other transaction kinds, etc.

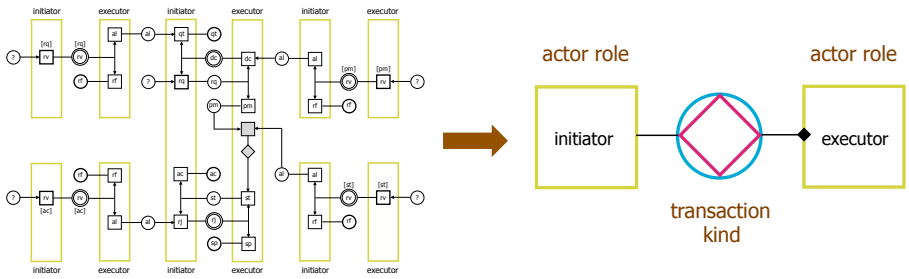


Figure 4.1 The basic construct in the Construction Model

A basic construct is connected to another basic construct if the executor role of a transaction kind is an initiator role of another transaction kind, etc. In this way, arbitrarily large, tree-like, networks of transaction kinds and actor roles can be constructed, as exhibited in Figure 4.2. Actor role A0 is the 'root' of the tree and A5, A6, A7, and A8 are the 'leafs'. The 'root' and the 'leafs' are actually determined by the *Scope of Interest* one chooses: there is some transaction kind of which A0 is the executor role, and every 'leaf' actor role may be initiator of other transaction kinds. As discussed earlier, the executor role of a transaction kind gets the same number as the transaction kind. For illustration, consider the transaction in which

I got a bouquet of yellow tulips from Celestine. Before promising, she could have initiated another transaction (with another executor) of which the result is the current number of bouquets of yellow tulips in stock. She then could base her decision to promise on this information. One can think of several other transactions (of several other kinds) that Celestine could have been engaged in, before promising. She also could do such things for every other process step in our transaction that she has to perform. The same holds for me. I could find it necessary or convenient to have other transactions completed before deciding to take a step in the transaction with Celestine. Therefore, one must conceive the construction model of an organisation as a network of connected organisational building blocks. To go one step further, one must conceive that the construction models of all organisations, all over the world, constitute one huge network. The construction model of a chosen Scope of Interest is a subnetwork, delineated in such a way that the 'border line' is 'under' those transaction kinds, of which the initiator role and the executor role are on different sides of the 'border line'. This network may exactly be the construction model of some organisation, but it may also be some part of it, or it may also comprise a part of the environment of the organisation. Examples are shown in Figures 4.4 and 5.2.

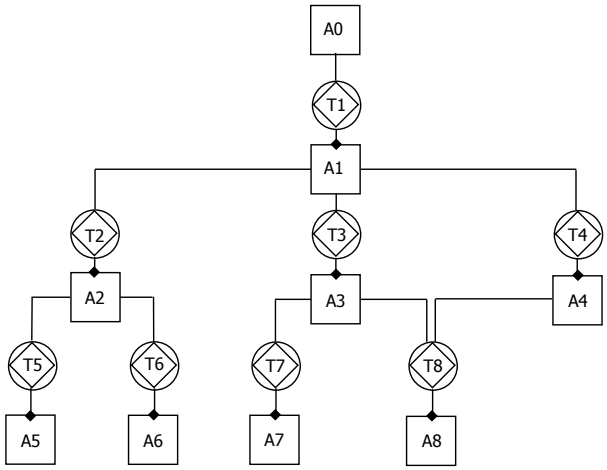


Figure 4.2 A business process as a tree of organisational building blocks

A tree-like network, as exhibited in Figure 4.2, may rightly be called a *construction model* of a Scope of Interest, because it shows the constituting elements (the actor roles) and the interaction structure between them (the transaction kinds). By allocating subjects to the actor roles, it can become operational, i.e. (business) processes can start to take place. Every process instance is some path through the network, starting at the 'root', with the initiation of a transaction of the kind T1. During the process of carrying out this instance of T1, A1 may start a number of instances of T2, T3 and T4, either consecutively, or in parallel, or in any mixed order. On their turn, A2, A3 and A4 may start a number of instances of the transaction kinds of which they are initiator roles. Note that both A3 and A4 are an initiator role of transaction kind T8. So, in the total business process instance, there may be instances of T8 that are enclosed in a transaction T3, and there may be instances of T8 enclosed in a transaction T4. Figure 4.2 clarifies immediately that current business process modelling techniques (BPMN, ARIS/EPC, Petri Nets, etc.) disregard the underlying tree structure of basic constructs. They only show the sequence of steps, thereby also disregarding the recurrent universal transaction pattern. At the same time, the notion of transaction provides us with a means to reduce the complexity of business process models enormously. One can easily see that a representation of a business process model in one of the current techniques comprises at least about 10 times as many elements as a corresponding construction model does.

After a short break, I continued with discussing the *essential model* of a Scope of Interest. This is the ontological model of its O-organisation, extended with elements that represent in an abstract way the services of the supporting I-organisation. Only then do we have a self-contained model of the enterprise's O-organisation. Well then, there are two main sorts of informational transactions, of which the initiator is an O-actor (in its green shape), and the executor is an I-actor in the I-organisation. One sort is the remembering of coordination (and production) facts, as extensively discussed in RGGDE. To accommodate this service sort, we assume that all coordination facts are remembered. For the ontological model of the O-organisation, this means that there are two interpretations of the transaction kind. One is the process interpretation, and the other one is the state interpretation. The *process interpretation* is the one that we have applied up to now. It entails that we interpret the transaction kind as the complete transaction pattern, and that we understand every transaction instance as some path through this pattern. In the *state interpretation*, the transaction kind is conceived as a container or bank that contains all coordination facts in all instances of the transaction kind, throughout history and up to the current point in time. The other main sort of informational transactions is the sharing of facts. In a sharing transaction, an O-actor asks for a coordination fact or a production fact. The needed fact may be a remembered (original) one, or a de-

rived fact. In the latter case, there is a derivation rule that specifies the fact type. Next, to properly accommodate the sharing of facts, information links are added to the ontological model. An *information link* connects an actor role with a transaction kind, which is called *transaction bank* now. The link represents the access right of actors in the actor role to the contents of the transaction bank, whenever needed.

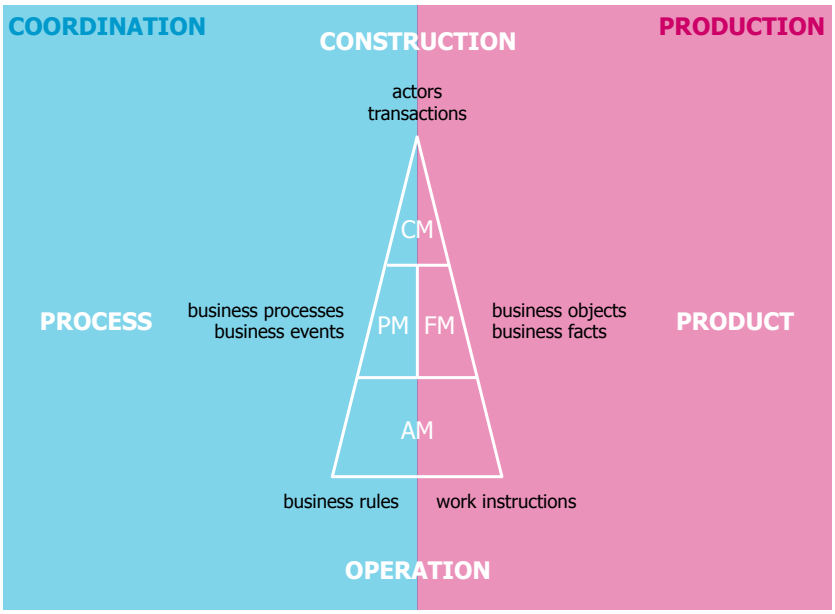


Figure 4.3 The sub models of an ontological model

Figure 4.3 exhibits the way in which the four sub models of a complete ontological model are related. They are four different views on the (complete) ontological model. The *Construction Model* (CM) is the most concise sub-model. Therefore it is put at the top of the triangle. An additional meaning of this position is that there is nothing 'above' the CM. The Construction Model shows the identified transaction kinds, the corresponding actor roles, and the border of the Scope of Interest. The *Action Model* (AM) is

the most comprehensive one, in the sense that the other three may be derived from it. The AM of an organisation consists of the *action rule specifications* for every internal actor role. *Action rules* are guidelines for dealing with the events that actors have to respond to. In current practice, they are often referred to as *business rules*. In addition, the AM may contain *work instructions*. These regard the execution of production acts. Work instructions may be useful if the production acts are material, like baking pizzas. However, also for immaterial products, like judgments and decisions, there can be 'work instructions'. In practice, quite complicated judgment or decision rules or protocols may apply. They are often supported by decision support systems or expert systems. In the case Volley, there are only immaterial products.

The *Process Model* shows precisely how the identified transactions are interrelated in tree structures. As we have seen, these tree structures are what people commonly refer to as *business process models*[21]. In addition, the coming into being of coordination facts is commonly referred to as *business events*. The *Fact Model* shows the fact kinds in the production world of the organisation and their interrelationships. In current practice, they are called *business objects* and *business facts*. In addition, the FM contains the laws that must be obeyed in order to keep every state and every state *transition* of the production world lawful. As these laws are the declarative version of the (imperative) business rules, as specified in the AM, they are called *business laws*. A similar reasoning holds for the coordination world. However, as the lawfulness of the states and the state transitions in the coordination world is fully determined by the complete transaction pattern, one usually does not pay separate attention to it.

The Process Model (PM) is put in between the CM and the AM, which means that it is more detailed than the CM but less than the AM. A similar reasoning holds for the Fact Model (FM). The juxtaposition of the PM and the FM expresses that they have a dual nature, they are like the two sides of the same coin. The PM takes the state view as well as the process view on the coordination world, and the FM does the same for the production world. The two worlds are brought together in the CM and the AM. In the *state view* one considers the states of a world, in the *process view* one considers the state transitions.

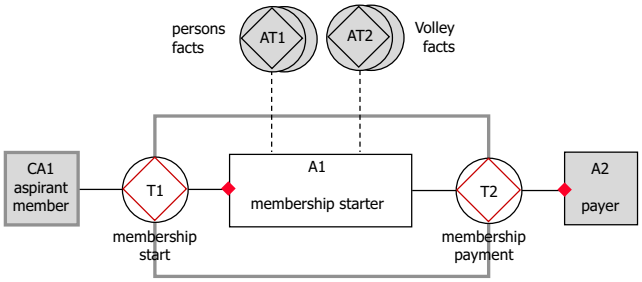
---

[21] The use of the word "business" here, and hereafter, is not consistent with my comment in footnote 6 (page 7). To remain consistent, I should speak of "organisational process", "organisational event" etc. However, the usage of the term "business" is so widespread that I have given in to the 'pressure of practice'.

Concluding, the essential model of a Scope of Interest, is the ontological model of its O-organisation, extended with information links. It implies that one takes both the process and the state interpretation of transaction kinds. Consequently, one can say that in the essential model of a Scope of Interest, there are two kinds of mutual influencing among actors, an active one, called interaction, and a passive one, called interstriction. *Interaction* consists of activating each other by causing coordination events to which (other) actors have to respond. *Interstriction* means literally that actors restrict each other's 'playing field'. It happens when an actor, in responding to a coordination event, takes particular facts (either original or derived) into account. These facts are usually created by other actors. Interstriction is a very powerful notion in understanding the construction of organisations 'at the essential level'. Because of it, the essential model of an enterprise satisfies the so-called C4E quality criterion: it is Comprehensive, Consistent, Coherent, and yet Concise, because it is Essential.

4.2 THE ESSENTIAL MODEL OF VOLLEY

Let me show you now the essential model of Volley that I have produced, based on the analysis that we have discussed in Chapter 3. First the Construction Model, represented in an *Organisation Construction Diagram* (OCD) and a *Transaction Product Table* (TPT). They are exhibited in Figure 4.4. It shows that actor role A1 is the executor role of transaction kind T1 and the initiator role of T2; the red color of the diamond in the transaction symbol indicates that T1 and T2 are original transaction kinds, so belonging to the O-organisation. The red color of the small diamond on the executor link emphasises that the holders of the actor role take on their red shape in bringing about the products. For legibility reasons, names are added to the identifiers. It is highly recommended to name actor roles differently from organisational functions, like secretary or administrator in the case of Volley. A practical guideline is to use a name that reflects the kind of production act that the actor role performs. The dark-grey thick-lined rectangle 'behind' the transaction symbols, represents the Scope of Interest. It means that we will only investigate actor role A1 in detail; the other two actor roles are outside our scope. One of them is included in the composite actor role CA1 and the other one is A2. A *composite actor role* is a network of transactions kinds and (elementary) actor roles, of which one does not know (or want to know) the details. A composite actor role is represented by a box with a thick borderline that is coloured dark-grey. Because CA1 and A2 are outside the Scope of Interest, the inner area is coloured light-grey. It is common practice in DEMO to model environmental actor roles as composite actor roles.



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

Figure 4.4 Organisation Construction Diagram and Transaction Product Table of Volley

The actor role A1 is connected by means of dashed lines with so-called aggregate transaction kinds outside the Scope of Interest. An *aggregate transaction kind* consists of one or more transaction kinds (which usually are not exactly known). It suffices that one knows that they exist. The dashed lines represent information links. As an indication of the kind of information, the label "persons facts" is attached to AT1. A similar reasoning holds for AT2, which is indicated by "Volley facts". Because AT1 and AT2 are outside the Scope of Interest, they are coloured light-grey. The initiator and the executor role holders of a transaction kind have always access to the history of their transaction instances. Therefore, one must imagine that there is an information link between a transaction kind and its initiator and executor roles. Put differently, the solid lines between actor roles and transaction kinds 'hide' a dashed line.

The next sub-model to be discussed is the Process Model, represented in the *Process Structure Diagram* (PSD) and the *Transaction Process Diagrams* (TPD). It shows the tree structures of the identified business processes and the exact way in which a transaction kind is enclosed in another one. The PSD of Volley is exhibited in Figure 4.5. The disk of the transaction symbol is stretched horizontally, such that it looks like a sausage. One must imagine that there is an invisible and non-proportional time line from left to right. For example, the promise of T1 is performed after its request. Coordination acts and facts are

represented respectively by small boxes and disks on the border of the transaction symbol (the sausage). The production act, also denoted as [T1/ex] from "execute", is represented by a small grey box on the edge of the production symbol (the 'big' diamond), which represents the execution phase of the transaction. To the left of it is the order phase, and to the right the result phase.

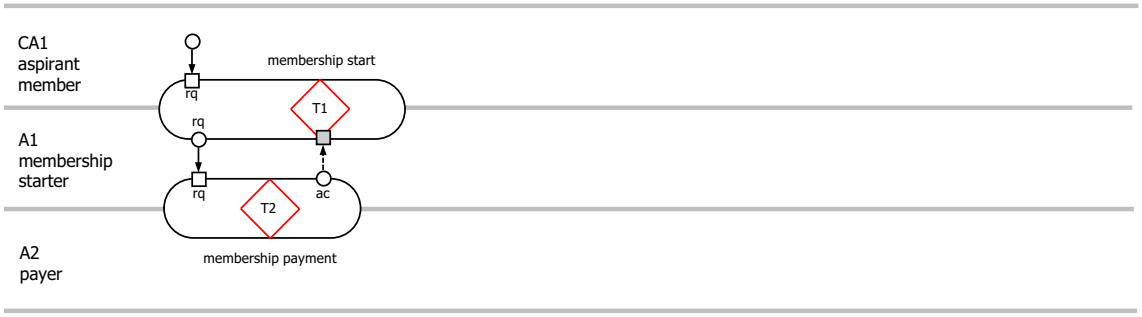


Figure 4.5 Process Structure Diagram of Volley

As a shorthand notation for process steps, we will use the same one as we did before. For example, the request of a transaction T1 is denoted as [T1/rq], and the being requested as (T1/rq). The two connected transaction kinds in Figure 4.5 constitute the (sole) business process of Volley. It proceeds as follows. A transaction of the kind T1 is initiated by CA1 from some transaction status outside of the Scope of Interest. Generally one does not know which one, and one often also does not care. Therefore we say that transactions T1 are initiated externally. This is represented by the small disk at the shaft of the arrow that points to [T1/rq]. Next, on the occurrence of the event (T1/pm), actor A1 performs [T2/rq]. This is represented by the arrow from (T1/pm) to [T2/rq]. Solid arrows represent *response links*. Their meaning is that the act at the arrow point is performed in response to the event at the shaft. Dashed arrows represent *waiting links*. Their meaning is that performing the act at the arrow point must wait for the event at the shaft having occurred. Through 'swim lanes' the responsibility areas of the actor roles are indicated.

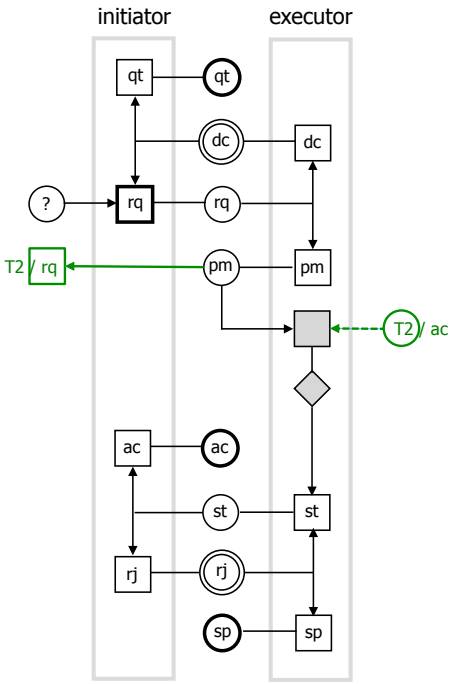


Figure 4.6 Transaction Process Diagram of T1 in Volley

So, the waiting link from (T2/ac) to [T1/ex] means that A1 has to wait until the status (T2/ac) is created before performing the production act [T1/ex], in response to (T1/pm). The way in which T2 is enclosed in T1 could also have been otherwise. Presumably, this is the current policy of Volley. One must also keep in mind that the actual course of every instance of a transaction can be any path through the complete transaction pattern. Therefore, the complete Process Model (PM) of Volley consists of the PSD in Figure 4.5 and the Transaction Process Diagrams (TPD) of T1 and T2. These are shown in Figures 4.6 and 4.7.

Next to presenting the complete transaction pattern, Figure 4.6 also shows the links between T1 and T2. From the status (T1/pm), a transaction of the kind T2 is initiated, by performing the process step [T2/rq]. Subsequently, the process of T1 has to wait until the process of T2 is successfully completed, by having

performed the process step [T2/ac]. Similar links are shown in Figure 4.7, but now from the perspective of T2. Here we see that transactions of the kind T2 are initiated from the status (T1/pm) in transactions of the kind T1. Next, as soon as the status (T2/ac) is reached, the transaction process of the corresponding T1 is resumed with performing [T1/ex]. The relevance of adding the complete transaction patterns for all transaction kinds (within the Scope of Interest of course) is that one is forced to investigate if and how each step is performed in the current or the future situation.

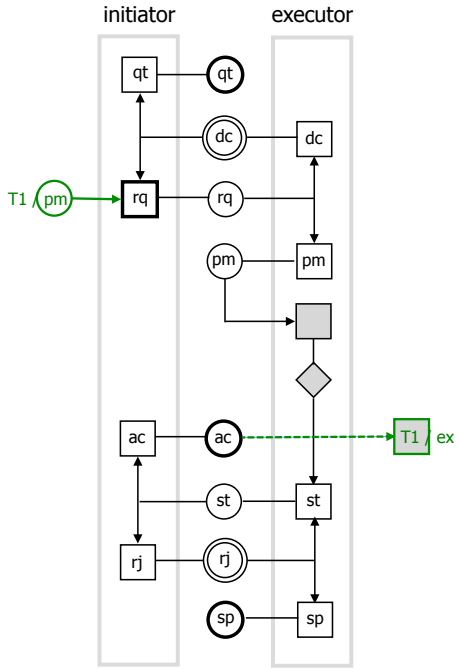


Figure 4.7 Transaction Process Diagram of T2 in Volley

In addition to the Organisation Construction Diagram, Figure 4.4 shows the Transaction Product Table (TPT) of Volley. One can read from it that the product kind of T1 is “Membership **is** started” and that it is labeled “P1”. Likewise, the product kind of T2 is “**the** first fee **of** Membership **is** paid”, labeled “P2”. The

TPT bridges the Construction Model with the Fact Model, which is represented in an *Object Fact Diagram* (OFD). In a Fact Model one specifies which business facts are relevant in a production world, and which business laws apply. Three sorts of business fact types are distinguished: object classes, property types, and attribute types. An *object class* is the extension of a unary *fact type* (conversely, a unary fact type is the intension of an object class). An example of an object class in Volley is MEMBERSHIP. A *property type* is a special binary fact type, namely a mapping from an object class, called the domain, to an object class, called the range. An example of a property type in Volley is the member of a membership. The instances of a property type are existentially dependent on the existence of the objects in the domain (which is drawn at the 'open' side of the ">" symbol). A property type is called an *attribute type* if the range is a *value scale*. An example of an attribute type is the day of birth of a person. Attribute types are listed within the shape of the object class. The dimension of the value scale is mentioned between brackets. Examples of dimensions are TIME and MONEY. They can be specialised into value scales, for example into YEAR, MONTH, DAY, and EURO or DOLLAR respectively.

Figure 4.8 exhibits the Object Fact Diagram (OFD) of Volley. The 'roundangles' represent object classes. The object classes MEMBERSHIP and PERSON are *core* classes; they are the extensions of the *core* entity types membership and person. Being core means that they cannot be defined on the basis of other fact types; instead they are declared to exist. The color light-grey of the category PERSON indicates that it is external. It means that persons are not created within our Scope of Interest, whereas memberships are. The line between MEMBERSHIP and PERSON, labeled "**the** member **of** Membership **is** Person", represents a property type. The symbol ">" expresses that every membership has exactly one person as its member, whereas every person may be member of 0, 1 or more memberships (Note: although it would not make much sense to have simultaneous memberships with the same person as member, a person can very well have multiple memberships (consecutively) in the course of time). As said before, the symbol ">" also expresses that this fact type is (existentially) *dependent* on the object class MEMBERSHIP. This means that it comes into being together with the corresponding instance of MEMBERSHIP. The product kinds P1 and P2 are (existentially) *independent* unary fact types: they are the results of transactions of the kind T1 and T2 respectively; that is why they have a diamond shape. Figure 4.8 exhibits that any membership can become a paid membership, but only paid memberships can become started memberships. Put differently, the class PAID MEMBERSHIP is a subclass of MEMBERSHIP, and the class STARTED MEMBERSHIP is a subclass of PAID MEMBERSHIP. Or, memberships first enter the phase of being paid and then of being started (and alive).

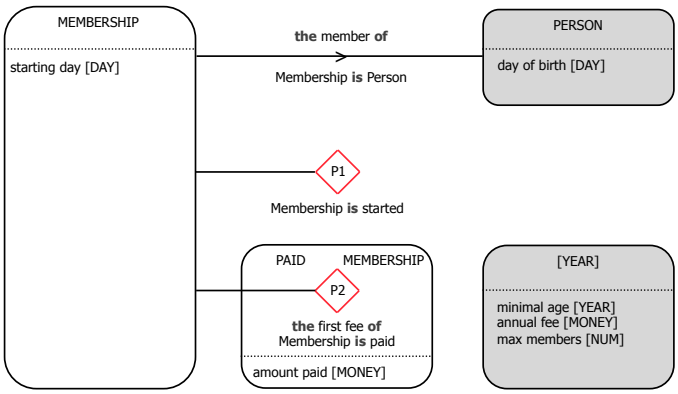


Figure 4.8 Object Fact Diagram of Volley

The object class [YEAR] is actually not an object class but a value scale; therefore the name is put between square brackets. In this way, one can elegantly model that minimal age, max members, and annual fee, are time dependent properties. To be precise, they are set every year. As an example, and as mentioned in the narrative description of Mr. X, fees are annually decided by the general assembly of Volley. So, one can speak of the annual fee in 2012, which may differ from the annual fee in 2011. These decisions are however outside our Scope of Interest. Therefore, the shape of [YEAR] is coloured grey.

Fact types may be *declared*, like MEMBERSHIP and date of birth, but they may also be *derived*. This means that they are not originally created but derivable from original fact types. The derived fact types in the case Volley are shown in the *Derived Fact Specifications* in Figure 4.9[22], expressed in a specific 'structured English' language. This figure also exhibits some business laws that come in addition to the ones that are expressed graphically in Figure 4.8. Examples of graphically expressed laws are that paid membership is a specialisation of membership, and that the binary fact kind '**the** member **of** Membership **is** Person' is a property type of paid membership.

[22] Regarding 'date' values in the TIME dimension, we adopt the Julian Time as the basic value scale.

Derived Fact Specifications

the age **of** Person **on** Day **=** Day **minus the** day of birth **of** Person
the number of members **on** Day **= the cardinality of** STARTED MEMBERSHIP **on** Day
the first fee **of** Membership **=** (12 **minus the** month **of the** starting day **of** Membership) **times the** annual fee **in the** year **of the** starting day **of** Membership

Business Laws

Membership **is** started **on** Day **implies that** Day **is the** first day **of some** Month **and** Month **is equal to or greater than** Current Month
Membership **is** started **on** Day **implies that the** age **of the** member **of** Membership **is equal to or greater than the** minimal age **in the** year **of** Day
Membership **is** started **on** Day **implies that the** number of members **on** Day **is less than or equal to the** max members **in the** year **of** Day

Figure 4.9 Derived Fact Specifications of Volley and Business Laws

We have discussed up to now three sub-models of Volley: the Construction Model, the Process Model, and the Fact Model. The fourth and last model to be discussed is the Action Model (AM). It consists of a set of *Action Rule Specifications* (ARS), according to the specific 'structured English' grammar that has already been used in Figure 4.9. As said, *action rules* are guidelines for actors when dealing with the events they have to respond to. The general form of an action rule is:

<event part> <assess part> <response part>.

The *event part* specifies which event (or set of events) will be settled. The *when clause* in Figure 4.10 shows that the event is the being requested of a transaction T1. The term "new" indicates that a new instance of membership is created. The *with clause* specifies the dependent facts of the membership that need to be known. The *assess part* in an action rule is divided in three sections, corresponding with the three validity claims: the claim to justice, the claim to sincerity, and the claim to truth. (Note: the justice conditions are facts that are contained in transaction banks of so-called allocation transactions, which we will not elaborate in this introductory lecture. For example, the initiator role in T1 is allocated to the (aspirant) member of a membership, in our specific case, to Anna). After assessing the conditions, the *response part* is entered. Its *if clause* specifies what action has to be taken if the actor considers complying with the conditions to be justifiable, and possibly what action must be taken if this is not the case. By this way of formulating action rules allows the performer to deviate from the 'rule', if he/she thinks this is justifiable (and for which he/she will be held accountable).

Regarding the grammar of the specification language, the next explanation holds. Bold printed terms are standard grammatical elements in the specification language; they serve to improve the readability of the rules. Underscored terms are specific expressions in action rule specifications, comparable to reserved words in a programming language. All other terms are specific for the modelled organisation. In order to show the relationships between an action rule and the corresponding part of the Process Model, coordination acts and facts are indicated at the far right side (Cf. Figure 4.10). Informally specified conditions and comments are written in italics. The three parts of the rule are marked by greenish boxes.

when	membership start **for** new Membership is requested		(T1/rq)
	with	**the** member **of** Membership **is some** person	
		the payer **of** Membership **is some** person	
		the starting day **of** Membership **is some** day	
assess	*justice:*	**the** performer **of the** request **is the** member **of** Membership	
		the addressee **of the** request **is a** membership starter	
	sincerity:	< no specific condition >	
	truth:	**the** starting day **of** Membership **is the** first day **of some** month;	
		the age **of the** member **of** Membership **on the** starting day **of** Membership **is equal to or greater than the** minimal age **in the** year **of the** starting day **of** Membership;	
		the number of members **on the** starting day **of** Membership	
		is less than the max members **in the** year **of the** starting day **of** Membership	
if	*complying with the assessment is considered justifiable*		
then	promise	membership start **for** Membership	[T1/pm]
	with	**the** addressee **of the** promise **is the** member **of** Membership;	
else	decline	membership start **for** Membership	[T1/dc]
	with	**the** addressee **of the** decline **is the** member **of** Membership;	

Figure 4.10 Action Rule Specification for A1 (1)

Let us take the example from the previous Chapter to explain the action rule in Figure 4.10 in more detail. In that instance of the rule, Anna is the performer of the request and Eve is the addressee. On principle, Eve is autonomous in dealing with the event. This implies two things. First, she is autonomous in determining the time at which she will deal with it. Second, she may deviate from the rule, by violating one or more conditions, as we have seen above. However, being authorised to fulfil actor role A1, Eve is supposed to act responsibly; she can always be held accountable for the consequences of her deeds. In the

event part, more particularly in the with clause, the applicable constraints are mentioned: the member must be some person, and the starting day must be some day. Assessing justice comes down to validating the authorisation of the performer. In our case it means that Eve acknowledges the authority of Anna to be an initiator of T1. Being a Dutch citizen makes her eligible for being an initiator of a transaction T1. Assessing sincerity means validating the sincerity of the performer of the coordination act. The question is: can Eve trust that Anna is sincere in her request to become member? Lastly, assessing truth means validating the truth of the production fact. It is valid if the fact exists, or if creating the fact leads to a lawful new state of the production world. For Volley this is guaranteed as long as the starting day of a membership is the first day of a month, the new member is at least as old as the minimal age on the starting day, and the maximum number of members is not exceeded.

I think the grammar of the presented 'structured English' specification language is quite straightforward, but you may need some time to get used to this way of formulating business rules, I said to the members of the board. Indeed, some of them looked rather puzzled when I discussed the first action rule. The point is, I explained, that the formulation must be analysable by formal means, like automated analysers. Therefore, it is necessary that every sentence can be unambiguously transformed into a logical formula. This is the case for the presented specification language. Recall that names starting with a capital are variables: they are placeholders for individual objects. For example, Membership is a placeholder of some instance of the type membership. During the execution of an action rule, the actual value of a variable does not change. So, in the action rule in Figure 4.10, the particular membership that is referred to by Membership, is the same in all of its occurrences in this instance of the action rule.

when	membership start **for** Membership <u>is promised</u>		(T1/pm)
assess	*justice:*	**the** <u>performer</u> **of the** <u>promise</u> **is the** membership starter **of** Membership **the** <u>addressee</u> **of the** <u>promise</u> **is the** member **of** Membership;	
	sincerity:	< no specific condition >	
	truth:	< no specific condition >	
if	*complying with the assessment is considered justifiable*		
then	<u>request</u>	membership payment **for** Membership	[T2/rq]
	with	**the** <u>addressee</u> **of the** <u>request</u> **is the** payer **of** Membership; **the** amount to pay **of** Membership **is equal to the** first fee **of** Membership	

Figure 4.12 Action Rule Specification for A1 (2)

In Figures 4.11, 4.12, and 4.13, the other action rules for actor role A1 are specified. Note that the assessment conditions are sometimes not specific. The reason is that the case description is also not specific about it. The actor roles A1 and A2 are referred to by their names: “membership starter” and “payer”.

```
when     membership payment for Membership is stated                          (T2/st)
         with     the amount paid of Membership is some money

assess   justice:   the performer of the state is the payer of Membership;
                    the addressee of the state is the membership starter of Membership;
         sincerity: < no specific condition >
         truth:     the amount paid of Membership is equal to the first fee of Membership

if       complying with the assessment is considered justifiable
then     accept   membership payment for Membership                            [T2/ac]
         with     the addressee of the accept is the payer of Membership;
else     reject   membership payment for Membership                            [T2/rj]
         with     the addressee of the reject is the payer of Membership;
```

Figure 4.12 Action Rule Specification for A1 (3)

```
when     membership start for Membership is promised                          (T1/pm)
         while    membership payment for Membership is accepted               (T2/ac)

assess   justice:   the performer of the promise is the membership starter of Membership
                    the addressee of the promise is the member of Membership;
         sincerity: < no specific condition >
         truth:     < no specific condition >

if       complying with the assessment is considered justifiable
then     execute  membership start for Membership                              [T1/ex]
         state    membership start for Membership                              [T1/st]
         with     the addressee of the state is the member of Membership;
```

Figure 4.13 Action Rule Specification for A1 (4)

The action rule specification in Figure 4.13 shows how waiting links are dealt with in the Action Model. The event to respond to in the when clause is (T1/pm), but the event(s) in the while clause must also have occurred. As long as this is not the case, the item remains on the agenda of the actor. The event(s) after "while" represent the waiting links in the Process Model. They may be intra-process links (within the same business process), but they can also be inter-process links (between different business processes).

It is the right time now to present the so-called *Bank Contents Table*, which completes the Construction Model. It takes the state interpretation on transaction kinds, which we have discussed before. The Bank Contents Table of Volley is exhibited in Figure 4.14. It tells us that in transaction bank T1 one can find every membership that has been created, the fact that it has been started (P1), the starting day, and the person who is the member. The instances of the property "amount paid" can be found in transaction bank T2, next to the product kind P2. The (external) aggregate transaction bank AT2 contains for every year the values of the listed parameters of Volley, while AT1 contains all persons and their days of birth.

bank	**independent/dependent facts**
T1 membership start	MEMBERSHIP Membership **is** started **the** starting day **of** Membership **the** member **of** Membership **the** amount to pay **of** Membership
T2 membership payment	**the** first fee **of** Membership **is** paid **the** amount paid **of** Membership
AT1 persons facts	PERSON **the** day of birth **of** Person
AT2 Volley facts	[YEAR] **the** minimal age **in** Year **the** annual fee **in** Year **the** max members **in** Year

Figure 4.14 Bank Contents Table of Volley

Right at the moment, one of the board members exclaimed: “I am lost! What is this all about? Why do we need all this? I am business architect at the National Tax Authorities, and I have to deal with many and very complicated projects, I can tell you! But I have never heard about transaction trees or coordination acts, or you name it! Our IT people make BPMN diagrams and we are fully satisfied with them. So, why would I need to know and do what you talk about?” I felt a bit upset by his comment, but at the same time I realised that something like this would have happened anyhow at some point in time. I should not be upset; enterprise engineering is just too much new stuff for most people. So, I asked the business architect whether the BPMN diagrams in his organisation were really so useful as he says they are. I asked him how exactly these diagrams were used for the purpose of redesigning and re-engineering business processes. After several more exchanges of thought, in which also other members of the board participated, I could convince him that the complete transaction pattern is of utmost importance for developing or selecting supporting IT applications. He had to admit that in his organisation such IT projects often failed because the ‘exceptions’ in Figure 2.4 were forgotten. I was happy with this turn in the meeting. It had become so natural for me to think in transactions and to have always the complete transaction pattern in mind that I had forgotten how unnatural it is for most people. So, I elaborated the topic and provided the next example from my own experience. It happened at a furniture factory, where I had to wait in the lobby for my appointment. During that time I overheard a telephone call between a customer and the receptionist. Apparently, the customer had bought a sofa the day before, but being back home discovered that the selected color definitely didn’t match the rest of the furniture (although yesterday she thought it did). So, she wanted to change the color. In terms of the complete transaction pattern (Cf. Figure 2.4), the process was in the status ‘promised’ and the customer wanted to revoke the request and make a new one, with a slightly different proposition. The receptionist told her that she very well understood the problem but that her order application didn’t have the option of cancelling a request. But don’t worry, she said, we will fix it. Just wait until the ‘wrong’ sofa is delivered at your home. Then let it be returned immediately. For we do have a returned goods procedure, and you will be notified soon after your sofa has been returned, when you can submit your new order. Every time I tell this story I feel vicarious shame; I can’t help it ...

Before saying goodbye to the members of the board, I emphasised that I did not expect them to be able to produce the DEMO models that I had presented, but that it would suffice if they did understand them. As an example, I elaborated on the Fact Model a bit by comparing it with data base models and data dictionaries, which some of them are familiar with. I said that one can look at the Fact Model as the core of a

data dictionary. One only would need to add explanatory texts to each of the elements. Likewise, one may view the Fact Model as the core of a data base model. Everything in it is needed, because it is ontological, but one may keep record of many more facts in practice, e.g. facts that are needed in informational or documental transactions. Lastly I emphasised that normally data base models cover both the coordination world and the production world of an organisation, although usually also being incomplete concerning the coordination world, because of disregarding the complete transaction pattern. The universality and comprehensiveness of the complete transaction pattern warrants that all relevant information will be recorded and will be available if the Process Model is completely implemented in a database. This final illumination did help, as several members of the board assured me.
Then Eve brought in the question how Volley should deal with the complete transaction pattern, when a new supporting information system would be considered. Is it necessary, she asked, to accommodate all 20 process steps? I put in the counter-question why one should not do it. The point is, I once more stressed, that the universal pattern is the natural way in which human beings interact. Consequently, they will expect from an information system that they can take any step in the complete transaction pattern if and when deemed appropriate. If the information system does not support it, you have to deal with it in some ad-hoc manual way. So, what do you want? Moreover, if the human-system interface of the new information system is fully web-based, the once-only extra development costs will certainly outweigh the costs of processing 'exceptions' manually, even if they occur rarely. My answer reassured Eve; it was what she had wanted to hear.

The last question came from Adam. He asked why there was no transaction kind in the model for resigning memberships. I could only tell him that this was a very accurate question, which had popped up in my mind too when I produced the model, but that there was no mentioning at all of such a transaction kind, neither in the narrative description of Mr. X nor in the Flow Charts of mr. Y. Clearly, this must be an omission, I said, which Adam could only heartily agree with. He told the board members that he just had forgotten to include it in the narrative description, and that the presented essential model of Volley had made him aware of the omission.

Then I asked Eve and Adam whom of them is holding actor role A1. As I expected, both thought they did, but then they realised that the answer is not as trivial as it may look. Well, I said, let us have a look again at Figure 9, Figure 10, and Figure 11. Clearly, the secretary is the one who responds to a (T1/request) by performing either [T1/promise] or [T1/decline]. In this part of the process, the administrator only carries

out informational and documental tasks. Performing the promise act (and the subsequent production act), I told them, normally makes someone the responsible executor of a transaction. It implies that this person is the principal responsible for the full completion of the transaction, although part of his or her tasks may be delegated to other persons. So, this would be Eve, the secretary. As the third part of the process (Figure 11) shows, the secretary has delegated her authority to the administrator for some process steps. To be precise, it is the administrator who performs [T1/state], as well as [T2/request] and [T2/accept]. So, he is responsible for correctly performing these steps, but the secretary remains the principal responsible for the whole process, because she has been allocated the mandated authority. Only she can be held accountable by the general assembly for any failure, not the administrator. After some discussion, everyone agreed that things are as I had supposed they would be.

Finally, after discussing the whole situation at Volley, the board accepted my offer to complete the essential model, and to join the meetings that Adam and Eve would organise with candidate software solution suppliers, in order to arrive at a fully satisfying automated support for the work of Adam and Eve. I was really happy with my success! Then I offered them to present and discuss the application of DEMO to another, larger, case. Surprisingly, all members wanted to attend that session, so we planned it. For sure, the motivation to spend another half day on getting to know better the practice of enterprise engineering, must come from their primary work; this thought made me even happier.

5 MODELLING ORGANISATIONAL ESSENCE

As I have illustrated in the analysis and the discussion of the case Volley, I said at the start of the extra session, the essential model of an enterprise's organisation can provide unexpected new insights in the operation of an enterprise. This is due to the radically different view one gets when looking at an enterprise through the glasses of enterprise ontology. This holds even for the case Volley, although it is indeed a very simple one. Therefore let me present and discuss another case, one from my own practice as enterprise engineer. It is a car rental company that I happen to know well. I handed out the narrative description, which I had produced after some days of observing and interviewing in the company. Here it is:

Rent-A-Car (or RAC for short) is a company that rents cars to persons, both private ones and representatives of legal bodies, like companies. It was founded by the twin brothers Janno and Ties back in the eighties. They started to hire out their own (two) cars, and they were among the first companies that allowed cars to be dropped off in a different location than where they were picked up. To this end, Janno and Ties had made agreements with students in several cities. For a small amount of money, a student would await the arrival of a rented car, e.g. at an airport, and drive it back to the office of RAC, after which the student would go home by public transport.
Currently, RAC operates from over fifty geographically dispersed branches in Europe. Many cities have a branch, some even several, and there are branches located near all airports. One of the branches is the original office where Janno and Ties started and where both are still around. Being mechanical engineer by education, they have kept loving to drive and maintain cars, even since they are the managing directors of a million euro company.

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

RENTAL: identification number (automatically generated), start date, end date, issuing branch, return branch, car group.
RENTER: identification (passport or driving license), first name, last name, address, date of birth, place of birth.
DRIVER: driving license (also for identification), first name, last name.
FINANCIAL: rental rate per day (basically determined by the car group).

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

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

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

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

On the starting day, the driver can pick up a car at the distribution department, located at the backside of the building, on presentation of a copy of the contract. There are three employees working in this department: Mik, Ferre, and Carlo, but not all of them are always present, as we will see. As soon as a driver shows up, one of them checks whether there is a car available of the contracted group. If there is one, he

will allocate the car to the rental contract and sign the contract as being picked up. If there is no car available of the contracted group, he will 'upgrade' the contract and select a car from the next higher car group. The driver will get this 'upgraded' car, for the price of the contracted group.

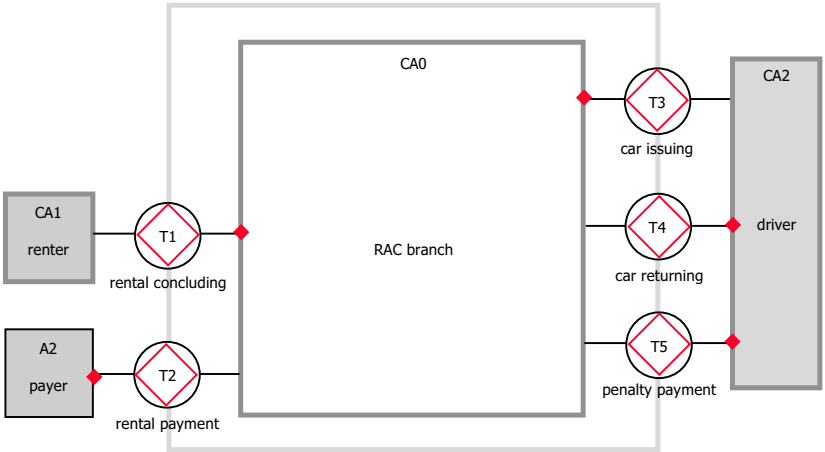
After the car of a rental has been dropped off at some branch, the possibly incurred fines have to be paid. There may be a penalty charge for returning the car after the contracted end date. It amounts to the number of extra days times the late return penalty rate. In addition, the car may have been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

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

Like I have shown for the case Volley, let us apply the OER method, thus abstracting from all realisation and all implementation aspects. This means in particular that we abstract from the way in which the actors communicate (like making telephone calls and filling out electronic forms), and the way in which information is stored and retrieved (like using RACIS). Instead, we focus on the commitments that the actors enter into and comply with, and on the content of information, regardless their form and the substance in which the form is inscribed. In addition, we put the template of the universal transaction pattern on our acquired understanding of the case, and try to identify the original transactions.

A practical advise for medium to large size organisations is to start with 'covering' the internal actor roles and transaction kinds, and to identify the 'border' transaction kinds first. The Organisation Construction Diagram (OCD) in Figure 5.1, shows how this is done, namely by putting a composite actor role in the centre; it represents all internal activities. The exhibited OCD and Transaction Product Table (TPT) are called global because they only show the interactions of the home branch of RAC (our Scope of Interest) with the environment. The main (composite) actor roles in the environment are the renter (CA1) and the driver (CA2). These are the initiators or executors of transaction kinds of which the other actor role is internal to RAC.

From the narrative description one can identify five original transaction kinds; they are listed in the TPT in the lower part of Figure 5.1. All of them can quite straightforwardly be deduced from the narrative description, although the formulation of the product kinds may require some effort. What may not be straightforward, is the modelling of the initiator and the executor roles of T3 (car pick up) and T4 (car drop off), as well as the separation of the rental payment and the penalty payment. I will elaborate these topics in due time.



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

Figure 5.1 Global OCD and TPT of Rent-A-Car

By applying the sapience 'verification by instantiation' (Cf. Figure 1.1), one can easily assure oneself that the variable in the TPT in Figure 5.1 is a placeholder for uniquely identifiable objects: a rental is a specific instance of the renting of a specific car by a specific person during a specific period of time. The fact that all transaction kinds regard the same entity type, namely rental (like membership in the case for Volley) is a bit incidental. Later we will see that there are internal transaction kinds that concern different entity types. In addition to the environmental actor roles (and the border transaction kinds), two external aggregate transaction banks are shown in Figure 5.1. The need for these banks follows from the narrative description. AT1 is the conceptual container of all kinds of facts regarding persons (renters and drivers). AT2 contains the facts that are created by RAC, but outside our Scope of Interest, and that are needed for the operational tasks that we are considering.

I have validated the global construction model with the staff of the home branch of RAC, through applying the sapience 'validation from ontology' (Cf. Figure 1.1). More specifically, I went through the complete transaction pattern for each of the identified transaction kinds, and discussed with the employees how each step was implemented currently, if it was implemented at all. As I expected, there were several tacitly performed coordination acts, as well as many disregarded revocation patterns.

5.1 MODELLING THE PRIMARY PROCESSES OF RAC

Let us try now, I said to the board members, to 'unveil' the internals of the kernel CA0. After some discussion, the result as exhibited in Figure 5.2 was produced, for which the following explanation holds. By definition, the executor of every border transaction kind of which the initiator is an environmental actor role, is a 'new' internal actor role. By convention, it gets the same number as the transaction kind number. We can identify right away two internal actor roles: A1, which can properly be named "rental concluder" and A3, which we name "car issuer". There are three border transaction kinds of which the initiator is an internal actor role, namely T2 (rental payment), T4 (car drop off), and T5 (penalty payment). The question then is: who is the initiator of these transactions? Both actor roles that we have identified up to now (A1 and A3) are candidates, but it could be necessary to introduce new actor roles. By logical reasoning we deduce that A1 is the initiator of T2, and that A3 is the initiator of T4 and T5. The arguments are quite straightforward: A1 deals with the contracting of a rental, and A3 with everything related to the handling of the rented car. As a consequence we can deduce that transactions of the kind T2 are enclosed in trans-

actions of the kind T1. Likewise, transactions of the kind T4 and T5 are enclosed in transactions of the kind T3. How exactly we will see later, in the PM. The information links to the external aggregate transaction banks AT1 and AT2 are not connected with the internal actor roles but with the border of our Scope of Interest. This is the way to indicate that all internal actor roles have access to an external transaction bank. This avoids cluttering the diagram by drawing information links from every external transaction bank to every internal actor role.

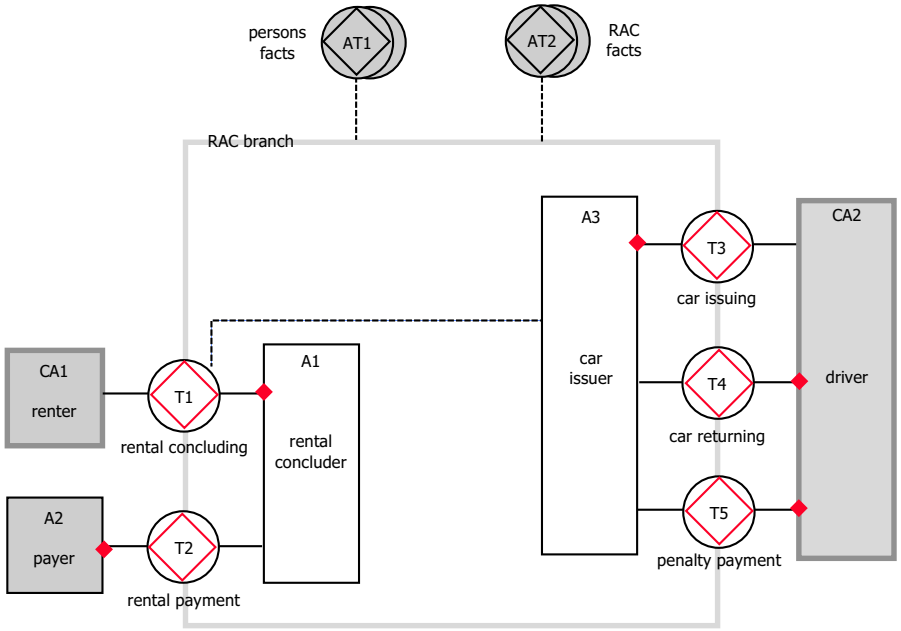


Figure 5.2 Detailed OCD of Rent-A-Car (1)

At first glance, the OCD in Figure 5.2 may look strange: there are two business processes (trees of transaction kinds) that seem not to be related. They are, though, but in a 'loosely coupled' way, namely through interstriction, represented in the diagram by the information link between actor role A3 and transaction bank T1. The meaning of this information link is that actors in role A3 have access to the contents

of the transaction bank T1. As will become fully clear from the Action Model, these actors need to check whether the rental contract, under which somebody wants to pick up a car, does exist.

Let us now try to model the first part of the Fact Model, by examining only the information that is used by or created by the part of the CM that consists of T1 and T2. Although we know that all Action Rule Specifications are needed to be fully sure about it, we can already identify a lot of fact types (object classes), by reading carefully the narrative description, and by additional logical reasoning. The result of this analysis is the Object Fact Diagram (OFD) in Figure 5.3.

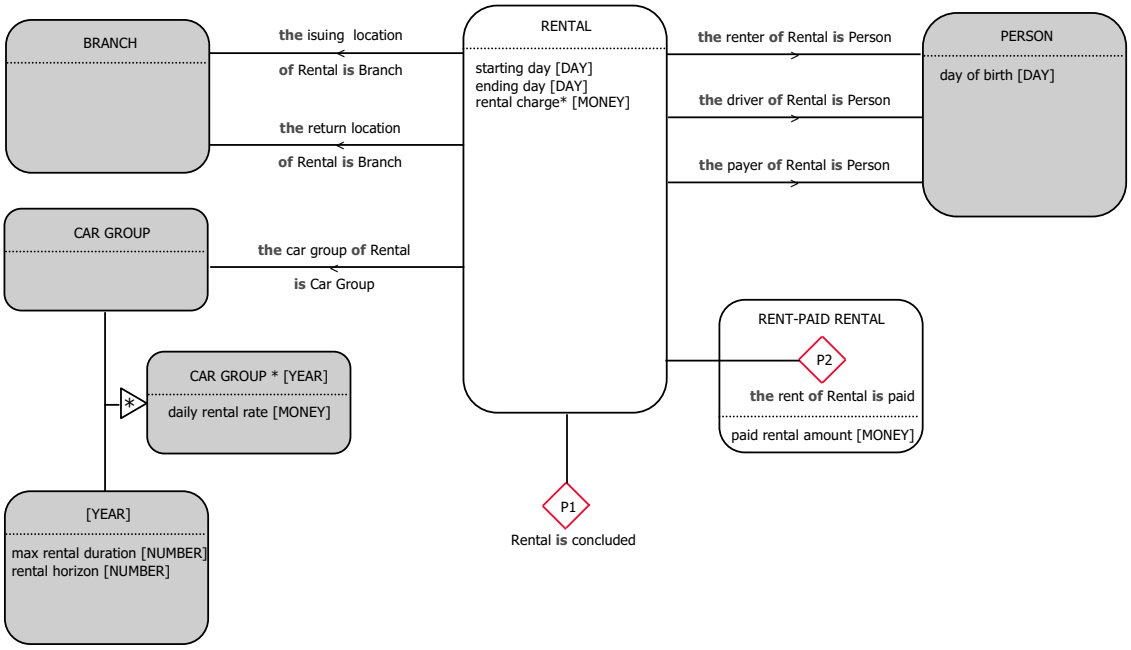


Figure 5.3 Object Fact Diagram of Rent-A-Car (1)

It shows first of all that the object class RENTAL is the core concept in the case RAC. RENTAL is the domain of two product kinds, P1 and P2 (Cf. the TPT in Figure 5.1). The legend of the OFD allows for defining subclasses graphically, like it is done in Figure 5.3 for PAID RENTAL. In this way, one can precisely indicate that < rental amount > is an attribute of a paid rental, not of any rental. The grey coloured object classes contain objects that are created outside our Scope of Interest, but about which we need information. This information is contained in the external aggregate transaction banks AT1 and AT2. The exact contents of every transaction bank is specified in the Bank Contents Table (BCT), in Figure 5.4.

transaction bank	**fact kind**
T1 rental contracting	RENTAL
	Rental **is** concluded
	the starting day **of** Rental
	the ending day **of** Rental
	the renter **of** Rental
	the driver **of** Rental
	the car group **of** Rental
	the pick-up location **of** Rental
	the drop-off location **of** Rental
T2 rental payment	**the** rent **of** Rental **is** paid
	the rental amount **of** Rental
AT1 persons facts	PERSON
	the day of birth **of** Person
	the driving license **of** Person
AT2 RAC facts	BRANCH
	the location **of** Branch
	CAR GROUP x [YEAR]
	the daily rental rate **of** Car Group **in** Year
	[YEAR]
	the max rental duration **in** Year
	the location penalty rate **in** Year
	the late return penalty rate **in** Year
	the rental horizon **in** Year

Figure 5.4 Bank Contents Table of Rent-A-Car (1)

The daily rental rate of the cars in a car group, is not only dependent on the car group but also on the (current) year. This is indicated in an OFD in the way as exhibited in Figure 5.3. The symbol "*" in the small triangle represents the so-called cartesian product of the classes CAR GROUP and [YEAR] (the last one is put between brackets to indicate that it is not a 'real' object class but a value scale). By default, the

cardinality range “1..1” applies to the ‘open’ side of the “>” and “0..*” to the other side. For example, every rental has exactly one branch as its pick up branch, and every branch has zero to an arbitrary number (*) of rentals for which it is the pick up branch. In such cases, the cardinality ranges are not indicated explicitly in the diagram, as we know from the case Volley. Moreover, these fact types are called properties of the objects on the ‘open’ side of the “>”.

Next, in the symbol of the object class RENTAL, three attribute types are listed: starting day (with value scale DAY), ending day (also with value scale DAY), and rental charge (with dimension MONEY). The asterisk next to “rental charge” indicates that this is a defined fact kind, for which the definition has to be provided. Figure 5.4 exhibits the Bank Contents Table (BCT) for the part of the essential model of RAC that we have produced up to now. Note that the car group properties “location penalty” and “late return penalty” are not needed in the part of RAC that we are studying now. However, they will be needed later.

Let us now try to produce the action rules for actor role A1, and check whether we have indeed identified all needed fact kinds. Because A1 is the executor role of T1 and an initiator role of T2, we have to provide at least the action rules for dealing with the events in the basic transaction pattern, while disregarding the rules for dealing with the ‘exception’ statuses (decline, reject, and revocations of the basic process steps). On the basis of the models we have produced so far, and the narrative description, we arrive at the Action Rule Specifications that are exhibited in Figures 5.5 thru 5.8. For the largest part, the action rule in Figure 5.5 will be self-explaining. However, some parts may need explanation. First, in response to a request that can be complied with, no promise of the T1 is yet performed. Instead, a request for T2 is issued. This is apparently the business rule, as can be found in the narrative description. Moreover, the promise of T1 has to wait for the promise of the corresponding T2. This is formulated in the action rule in Figure 5.6. In this action rule, we see once more how a wait condition is modelled in the Action Model. Promising T1 is actually a response to a request of T1. However, since we have to wait in this case for the being promised of the corresponding T2, it has to be modelled as is done in Figure 5.6 (when ... while).

In action rules, variables like “This Year”, “Next Year”, “Today” etc. have standard defined values, whose meaning is straightforward. Next, the value of the variable “Rental Horizon” is 200 days from now (see case description). It means that one cannot make reservations for days beyond that period. In the case Volley I did not mention the addressees of acts explicitly, to keep things simple. In the case RAC, however, they are mentioned explicitly.

```
when     rental concluding for new Rental is requested                                  (T1/rq)
         with     the starting day of Rental is some day
                  the ending day of Rental is some day
                  the renter of Rental is some person
                  the payer of Rental is some person
                  the driver of Rental is some person
                  the payer of Rental is some person
                  the car group of Rental is some car group
                  the issuing location of Rental is some branch
                  the return location of Rental is some branch

assess   justice:     the performer of the request is the renter of Rental;
                      the addressee of the request is a rental concluder;
         sincerity:   < no specific condition >
         truth:       the starting day of Rental is in the Rental Horizon of the year of
                      the starting day of Rental;
                      the ending day of Rental is in the Rental Horizon of the year of
                      the ending day of Rental;
                      the ending day of Rental is equal to or greater than the starting day of Rental;
                      the duration of Rental is less than or equal to
                      the max rental duration in the year of the starting day of Rental;
                      the number of cars in the car group of Rental on every day in the rental period
                      of Rental is greater than zero

if       complying with the assessment is considered justifiable
then     request  rental payment for Rental                                               [T2/rq]
         with     the addressee of the request is the payer of Rental;
                  the requested production day of rental payment for Rental
                  is less than or equal to the starting day of Rental;
                  the requested paid rental amount of Rental is equal to the rental charge of Rental;
else     decline  rental concluding for Rental                                            [T1/dc]
         with     the addressee of the decline is the renter of Rental
```

Figure 5.5. Action Rule Specification for A1 (1)

The rental charge in the action rule in Figure 5.5 is a derived fact kind. It is defined as the duration of the rental times the daily rental rate for the car group of the rental in the year of the starting day. Next, the duration of a rental is defined as the difference between the ending day and the starting day of the rental.

A request by a renter for a rental contract will generally not be addressed to a specific actor. Therefore, in Figure 5.5, the second justice condition speaks of *a* rental concluder (somebody who is authorised to fulfil actor role A1). In Figure 5.6, however, the particular rental has been allocated a specific actor as the rental concluder. Therefore, it speaks of *the* rental concluder.

when rental concluding **for** Rental is requested			(T1/rq)
	while	rental payment **for** Rental is promised	(T2/pm)
assess	*justice:*	**the** performer **of the** request **is the** renter **of** Rental;	
		the addressee **of the** request **is the** rental concluder **of** Rental;	
	sincerity:	< no specific condition >	
	truth:	**the** promised paid rental amount **of** Rental **is equal to**	
		the requested paid rental amount **of** Rental;	
		the promised production day **of** rental payment **for** Rental	
		is less than or equal to the starting day **of** Rental	
if	*complying with the assessment is considered justifiable*		
then	promise	rental concluding **for** Rental	[T1/pm]
	with	**the** addressee **of the** promise **is the** rental concluder **of** Rental	
else	decline	rental concluding **for** Rental	[T1/dc]
	with	**the** addressee **of the** decline **is the** renter **of** Rental	

Figure 5.6 Action Rule Specification for A1 (2)

Note that, although the justice condition in Figure 5.6 says that the specific rental contractor of the rental must be addressed, it may also be someone who has the delegated authority to do so. This is a general rule; every actor is allowed to delegate his/her *principal authority* to someone else. However, the principal actor will remain accountable for the deeds of the delegate. I will not go into the details of delegation, I mean not further then we did for the case Volley.

On the basis of the identified action rules in Figures 5.5 thru 5.8, it is not so difficult to produce the corresponding process model, represented in a Process Structure Diagram (PSD). It is shown in Figure 5.9. The corresponding transaction pattern diagrams are not presented (try to make them yourself).

when rental payment **for** Rental is stated (T2/st)

assess *justice:* **the** performer **of the** statement **is the** renter **of** Rental;
the addressee **of the** statement **is the** rental concluder **of** Rental;
sincerity: < no specific condition >
truth: **the** stated paid rental amount **of** Rental **is equal to**
the promised paid rental amount **of** Rental;
the stated production day **of** rental payment **for** Rental
is equal to the promised production day **of** rental payment **for** Rental

if *complying with the assessment is considered justifiable*
then accept rental payment **for** Rental [T2/ac]
with **the** addressee **of the** acceptance **is the** renter **of** Rental
else reject rental payment **for** Rental [T2/rj]
with **the** addressee **of the** rejection **is the** renter **of** Rental

Figure 5.7 Action Rule Specification for A1 (3)

when rental concluding **for** Rental is promised (T1/pm)
while rental payment **for** Rental is accepted (T2/ac)

assess *justice:* **the** performer **of the** promise **is the** rental concluder **of** Rental;
the addressee **of the** promise **is the** renter **of** Rental;
sincerity: < no specific condition >
truth: < no specific condition >

if *complying with the assessment is considered justifiable*
then execute rental concluding **for** Rental [T1/ex]
state rental concluding **for** Rental [T1/st]
with **the** addressee **of the** statement **is the** renter **of** Rental

Figure 5.8 Action Rule Specification for A1 (4)

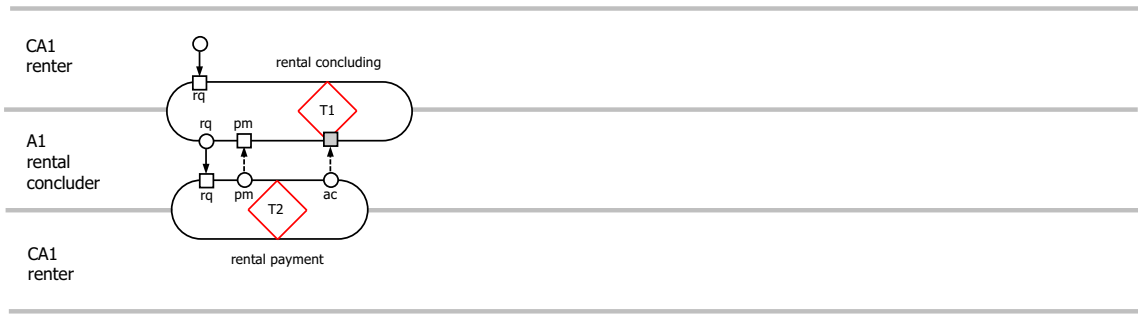


Figure 5.9 Process Structure Diagram of Rent-A-Car (1)

The PSD clearly shows how transactions of the kind T1 and T2 are related. In response to some external event, the act [T1/rq] is performed. Then, in response to the event (T1/rq), A1 performs the act [T2/rq]. This is in conformity with action rule 1 (Figure 5.5). Next, A1 waits until the event (T2/pm) has occurred before performing the act [T1/pm], in conformity with action rule 2 (Figure 5.6). Lastly, the execution of T1 has to wait for the occurrence of the event (T2/ac), in conformity with action rule 4 (Figure 5.8).

Let us put our attention now to the other part of the OCD in Figure 5.2, consisting of the transaction kinds T3, T4, and T5, and let us start this time with the process model, again based on the narrative description. Then we arrive at the PSD that is exhibited in Figure 5.10. The model shows how transactions of the kind T3, T4, and T5 are related. The process starts with the external initiation of a transaction T3. Then, in response to the event (T3/pm), A3 performs the act [T4/rq]. Next, A3 waits until the event (T4/pm) has occurred before performing the act [T3/ex]. Then we see for the first time that the occurrence of an 'exception' event is responded to, namely (T4/rj). Its practical meaning is that the statement by CA2 (the driver) of having dropped off the rented car is rejected by A3, because the car is dropped off at another branch and/or too late. Next, A3 performs the act [T5/rq] in response to the event (T4/rj). The cardinality range "0..1" indicates that this act is optional (it will not always take place). For keeping the diagram unambiguous, the cardinality range "0..1" is repeated next to the waiting link from the event (T5/ac) to the act [T4/ac]. The practical meaning of this construct is that A3 waits with accepting T5 until the incurred penalty has been paid (if there is a T5 at all).

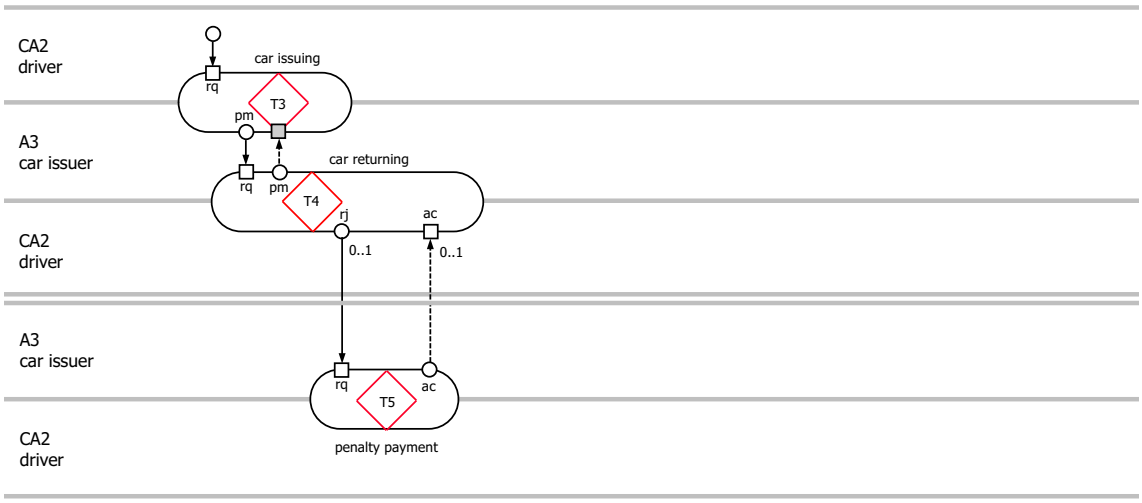


Figure 5.10 Process Structure Diagram of Rent-A-Car (2)

The penalty is paid by the driver, whereas the rental charge is paid by the renter. The separation of these two payments makes practical sense in a situation that, for example, a company rents cars for its employees, but does not want to be responsible for paying (unnecessarily) incurred fines caused by 'abnormal' behaviour of the driver. Commonly, however, the renter and the driver will be the same person.

The next interesting question is how exactly the action rules look like that guide the business process that is exhibited in Figure 5.10. They are presented in Figures 5.11 thru 5.15. I think most of them are self-explaining. In the narrative description of the case, it says that a rental can be 'upgraded' if there is no car available in the contracted car group at the day of the pick up. I have disregarded this option in the presented action rules. As can be read from the action rule in Figure 5.12, the pick up of a car is successfully completed if there is a suitable car, and if the driver has promised to return the car at the right time and at the right location (see also the action rule in Figure 5.11). The action rule in Figure 5.13 specifies what happens if the driver does not live up with his/her promise. Then the statement of the drop off, (T4/st), will be rejected instead of accepted, as we have seen above. In addition, the request for the penalty payment is issued.

```
when     car issuing for Rental is requested                                    (T3/rq)

assess   justice:    the performer of the request is the driver of Rental;
                     the addressee of the promise is a car issuer;
         sincerity:  < no specific condition >
         truth:      Rental is concluded;                                       (T1/ac)
                     the driving license of the driver of Rental is valid;
                     the starting day of Rental is less than or equal to Today;
                     the number of cars in the car group of Rental on Today is greater than zero

if       complying with the assessment is considered justifiable
then     promise     car issuing for Rental                                     [T3/pm]
         with        the addressee of the promise is the driver of Rental;
                     the promised production time of car issuing for Rental is within Today;
         request     car returning for Rental                                   [T4/rq]
         with        the addressee of the request is the driver of Rental;
                     the requested production time of car returning for Rental is within
                                 the ending day of Rental
else     decline     pick up for Rental                                         [T3/dc]
         with        the addressee of the decline is the driver of Rental;
```

Figure 5.11 Action Rule Specification for A3 (1)

```
when     car issuing for Rental is promised                                     (T3/pm)
         while       car returning for Rental is promised                       (T4/pm)

assess   justice:    the performer of the promise is the car issuer of Rental;
                     the addressee of the promise is the driver of Rental;
         sincerity:  < no specific condition >
         truth:      there is a Car in the car group of Rental on Today;

if       complying with the assessment is considered justifiable
then     execute     car issuing for Rental                                     [T3/ex]
         state       car issuing for Rental                                     [T3/st]
         with        the addressee of the statement is the driver of Rental;
                     the car of Rental is Car
```

Figure 5.12 Action Rule Specification for A3 (2)

The action rule in Figure 5.13 contains two derived fact kinds, namely the location penalty charge and the late return penalty charge. The first one is defined as the geographical distance between the contracted drop off location and the actual drop off location times the current location penalty rate. The late return penalty charge is defined as the time distance (or difference) in days, between the actual drop off day and the contracted drop off day times the current daily penalty rate.

```
when     car returning for Rental is stated                                    (T4/st)
         with     the actual return location of Rental is some BRANCH

assess   justice:  the performer of the statement is the driver of Rental;
                   the addressee of the statement is the car issuer of Rental;
         sincerity: < no specific condition >
         truth:    the actual return location of Rental is the return location of Rental;
                   Today is less than or equal to the ending day of Rental

if       complying with the assessment is considered justifiable
then     accept   car returning for Rental                                     [T4/ac]
         with     the addressee of the acceptance is the driver of Rental;
else     reject   car returning for Rental                                     [T4/rj]
         with     the addressee of the reject is the driver of Rental;
         request  penalty payment for Rental                                   [T5/rq]
         with     the addressee of the request is the driver of Rental;
                  the requested production time of penalty payment for Rental is Now
                  the requested penalty amount of Rental is equal to
                  the location penalty charge of Rental plus the late return penalty charge of Rental
```

Figure 5.13 Action Rule Specification for A3 (3)

The carrying out of a transaction T4 may need some clarification. First, a when clause refers to the most recent occurrence of the mentioned process status. So, the status (T4/st), as mentioned in the when clause in action rule 5 (Figure 5.15) may have occurred before, but the most recent instance is meant here. Next, if the conditions for 'normal' processing of T4 (car drop off) are not met, the act [T4/st] by the driver (CA2) will be rejected by A3 (see action rule 3, Figure 5.13). Figure 5.16 exhibits the TPD of transaction kind T4. The orange coloured path from (T4/st) to [T4/rj] represents what happens if the car drop off is rejected. In the discussion status (T4/rj), the employee of RAC (who fulfills actor role A3) will probably explain to the driver that the car has been dropped off at the wrong branch and/or that it has been dropped off too late. Anyhow, A3 will initiate a transaction T5 (penalty payment). As soon as this transaction has

reached the status (T5/ac), CA2 can perform [T4/st] once more. This is represented by the orange path from (T4/rj) to [T4/st], and the continuing green path from [T4/st] to (T4/st) in Figure 5.16. As soon as T5 is in the status accepted, A3 is able to accept the T4, as shown by the green path from (T4/st) to [T4/ac].

```
when     penalty payment for Rental is stated                                  (T5/st)

assess   justice:   the performer of the statement is the driver of Rental;
                    the addressee of the statement is the car issuer of Rental;
         sincerity: < no specific condition >
         truth:     the stated penalty amount of Rental is equal to
                    the requested penalty amount of Rental;
                    the stated production time of penalty payment for Rental is
                        within the promised production time of penalty payment for Rental

if       complying with the assessment is considered justifiable
then     accept     penalty payment for Rental                                 [T5/ac]
         with       the addressee of the acceptance is the driver of Rental
else     reject     penalty payment for Rental                                 [T5/rj]
         with       the addressee of the acceptance is the driver of Rental
```

Figure 5.14 Action Rule Specification for A3 (4)

```
when     car returning for Rental is stated                                    (T4/st)
         while      penalty payment for Rental is accepted                     (T5/ac)

assess   justice:   the performer of the statement is the driver of Rental;
                    the addressee of the statement is the car issuer of Rental;
         sincerity: < no specific condition >
         truth:     < no specific condition >

if       complying with the assessment is considered justifiable
then     accept     car returning for Rental                                   [T4/ac]
         with       the addressee of the acceptance is the driver of Rental
else     reject     car returning for Rental                                   [T4/rj]
         with       the addressee of the rejection is the driver of Rental
```

Figure 5.15 Action Rule Specification for A3 (5)

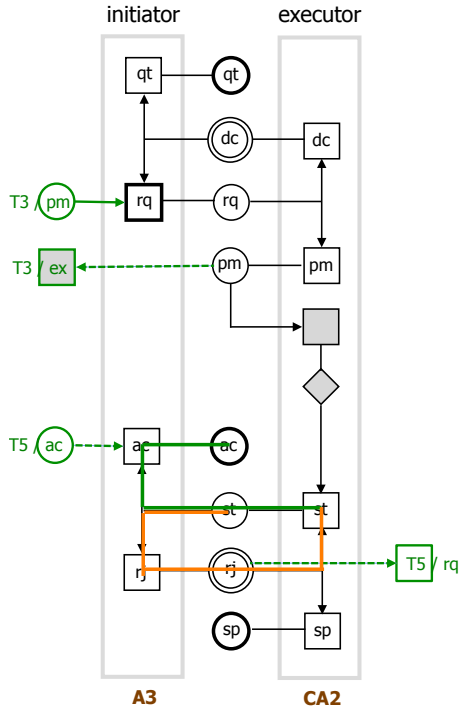


Figure 5.16 Transaction Process Diagram for T4

After I had finished the extensive explanation of the transaction process of T4, I saw that the business architect had become angry again. “Why must this be so complicated?” he exclaimed. “As I have told you, I never make or see such process models, and this is even still a ‘toy’ case!” There we go again! So, I asked him whether he thinks that one does not need to go into the process details we have got into, if the purpose of the model is to understand deeply what the process is about, for example for the purpose of redesigning the process, or for developing or selecting supporting software, as we had extensively discussed earlier. After a long time, he admitted that indeed at some point in the whole project this would be necessary. But, he added: “I don’t make such models and I never will, because it’s not my job. My job is

to tell the IT people, like information architects and process architects, what we need. So, I assume that it is their job to make the kind of models you are talking about." Ok, fine, I said, but then you let these IT people determine what your business processes are. Is that what you really want? Obviously, he was not prepared for this question. In the subsequent discussion, I tried to clarify to him that on the basis of what he told the IT people he would need, chances are great that the solution they come up with is not what he really needs. After some more discussions, in which other board members also took part, he could only fully and even emphatically agree, he said; there were always misunderstandings. Well, I replied, the conclusion is that these IT people make the wrong organisational models, or they don't make them at all. But that is not the way you want things to go, do you? Besides, it is not the task of the IT people to make organisational models, it is your task, the task of business architects to do it, because you have all the knowledge that is needed to make proper and correct models, don't you? So? After several more exchanges of thought, the business architect could not deny that it is in his interest that the communication between him and the IT people is complete and as perfect as possible, that it is his responsibility to produce the essential model, and that the essential model of (a part of) an enterprise provides all the information that is relevant, without constraining the design freedom of the IT people unnecessarily. He would give it a thought, he promised me, and discuss this subject with his colleagues at the National Tax Authorities.

Well, I continued, we are not yet done. First, we have to complete the set of models concerning the second business process (as shown in Figure 5.10). What is missing, is the corresponding Object Fact Diagram, as well as the Bank Contents Table. The last one, we will not discuss, in order to save time, and also because this kind of table can always straightforwardly be derived. The Fact Model, however, needs to be produced. Figure 5.17 exhibits the OFD for the two business processes we have discussed. It comes in addition to the OFD in Figure 5.3. Together, the diagrams illustrates quite well that commonly entities, like rentals, have a life-time, during which they pass several phases: concluded, paid, issued, returned, etc.

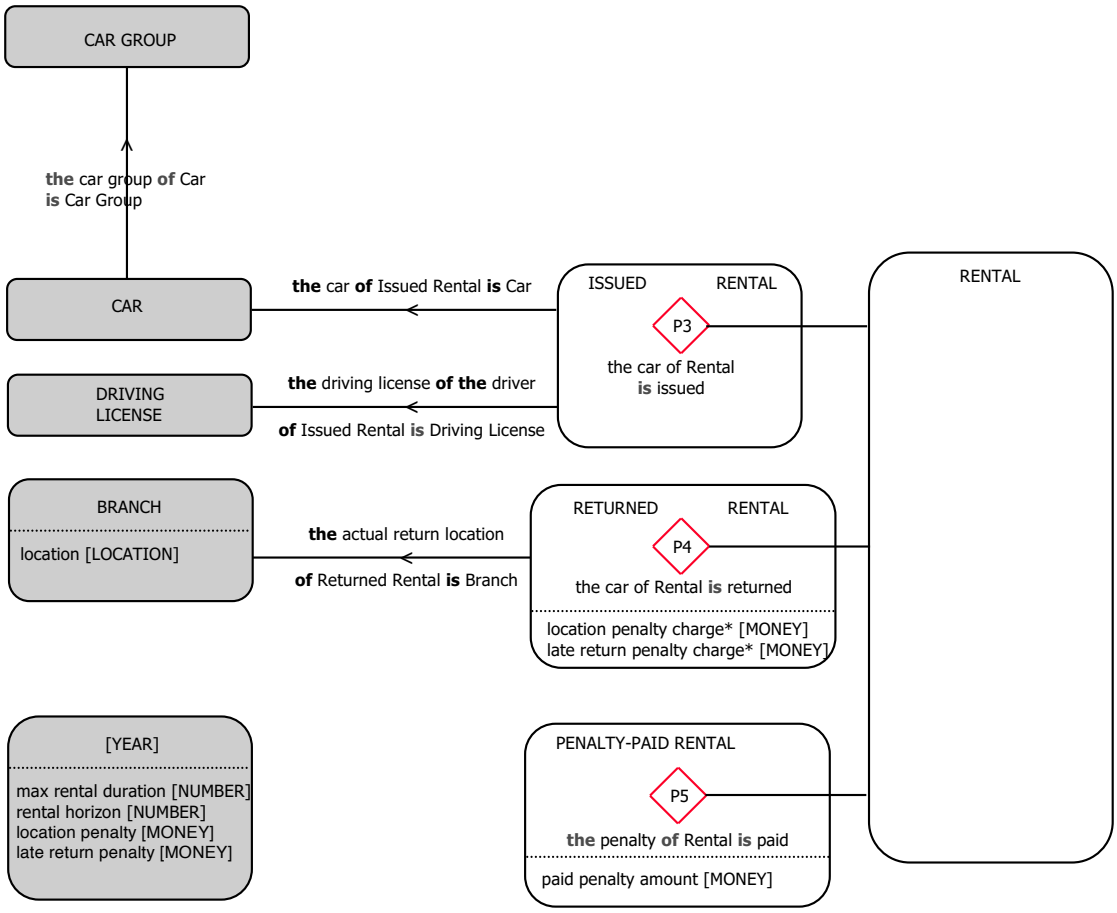
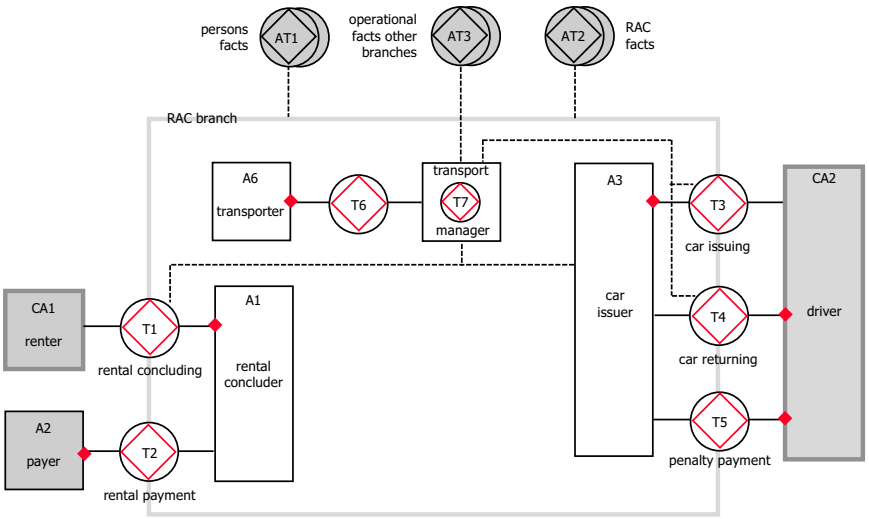


Figure 5.17 Fact Model of Rent-A-Car (2)

5.2 MODELLING THE SECONDARY PROCESSES OF RAC

The last part of the case description has to be dealt with still, namely, the part about the transport of cars between the distinct branches of RAC. Figure 5.18 exhibits the OCD and the TPT of RAC once more, but now extended with the part that concerns the transport of cars. The first thing to notice is a new symbol: the box of an actor role with the downsized symbol of a transaction kind in it. Such a construct is called a *self-activating actor role*. It is the standard way to model periodic activities. The actor role is A7, named transport manager, and the transaction kind is T7, named transport management (see the corresponding part of the TPT). A7 is both initiator and executor of T7, and it is also initiator of transactions T6. Transport management is executed daily, as one can read from the TPT (P7).



transaction kind	product kind
T6 transport completion T7 transport management	P6 Transport **is** completed P7 transport management **for** Day is done

Figure 5.18 Detailed OCD and TPT of Rent-A-Car (2)

Figure 5.19 exhibits how periodic activities are modelled in the Process Model. In response to the status (T7/rq) A7 performs two acts: the 'normal' one [T7/pm] and the 'extra' one [T7/rq] for the next period. Next, in response to the status (T7/pm) A7 initiates a number of transactions T6, possibly zero. This expresses that the transport manager determines daily which cars have to be transported and issues accordingly transport completion requests. As soon as all of them are completed, T7 can be executed, which means that the transport manager is discharged now from the transport management responsibility of the completed period. Note that earlier instances of T7 may still be running, which means in practice that the transport manager is not yet discharged of his responsibility for the corresponding days.

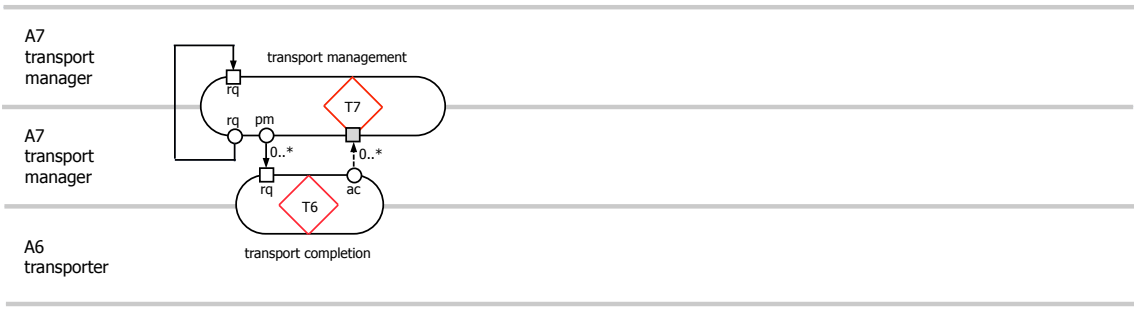


Figure 5.19. PSD of Rent-A-Car (3)

Let me present and discuss now the corresponding action rules, I said to the board members, of which for some reason the business architect had become very attentive. Probably, he is involved in management and/or periodic activities, I figured. The first action rule, exhibited in Figure 5.20, contains the standard way in which periodic activities are modelled in the Action Model. In responding to a (T7/rq) the thing to do always, is performing the [T7/rq] for the next day. Therefore, it is put unconditionally at the end of the action rule. Next, since the initiator and the executor of T7 is the same actor role (A7), dealing with the status (T7/dc) is quite simple: the executor decides to quit the transaction. This will always happen if there are no cars to be transported. In order to keep things simple, the truth condition in Figure 5.20 is specified in an informal way.

when	transport management **for** Day is requested		(T7/rq)
assess	*justice:*	**the** performer **of the** request **is the** transport manager	
		the addressee **of the** request **is the** transport manager	
	sincerity:	< no specific condition >	
	truth:	*the number of cars to be transported on Day is greater than zero*	
if	*complying with request is considered justifiable*		
then	promise	transport management **for** Day	[T7/pm]
	with	**the** addressee **of the** promise **is the** transport manager	
else	decline	transport management **for** Day	[T7/dc]
	with	**the** addressee **of the** decline **is the** transport manager;	
	request	transport management **for** Next Day	(T7/rq)

Figure 5.20 Action Rule Specification for A7 (1)

If there is at least one car to be transported, A7 will initiate a transaction T6 for every car to be transported, in response to the being promised of a T7. The corresponding action rule is shown in Figure 5.21. Although most members of the board seemed to have understood the generic patterns for dealing with periodic activities in the two action rules, apart from their being (also) specific for the case RAC, I spent some more time to emphasise it, with special attention to the action rules in Figures 5.21 and 5.22. The waiting event to take into account is the 'last' transport completion for a given day. What the rule does is checking for every completed transport on a given day whether it is the last one, that is whether all cars for that day have been transported. If so, A7 can safely conclude that his/her job is done, which means that T7 can be executed and stated, and consequently accepted. The action rule in Figure 5.22 is added to make the model complete, but in general the acceptance of the statement of a T7 cannot go wrong.

Note that I have also used an informal specification in Figure 5.21, namely "{cars to be transported on Day}". Its meaning is quite straightforward, however its formal definition requires some mathematics, which I leave to the interested reader.

when transport management **for** Day is promised (T7/pm)

assess *justice:* **the** performer **of the** promise **is the** transport manager
the addressee **of the** promise **is the** transport manager
sincerity: < no specific condition >
truth: < no specific condition >

if *complying with promise is considered justifiable*
then **for each** Car **in** {*cars to be transported on Day*}
request transport completion **for** Transport (T06/rq)
with **the** addressee **of the** request **is a** transporter;
the car of Transport **is** Car

when transport management **for** Day is promised (T7/pm)
while **for each** Car **in** {*cars to be transported on Day*}
transport completion **for** Transport **of** Car **is** accepted (T06/ac)

assess *justice:* **the** performer **of the** promise **is the** transport manager
the addressee **of the** promise **is the** transport manager
sincerity: < no specific condition >
truth: < no specific condition >

if *complying with promise is considered justifiable*
then execute transport management **for** Day [T7/ex]
state transport management **for** Day [T7/st]
with **the** addressee **of the** statement **is the** transport manager

Figure 5.21 Action Rule Specification for A7 (2 and 3)

when	transport management **for** Day is stated		(T7/st)
assess	*justice:*	**the** performer **of the** statement **is the** transport manager	
		the addressee **of the** statement **is the** transport manager	
	sincerity:	< no specific condition >	
	truth:	< no specific condition >	
if	*complying with the assessment is considered justifiable*		
then	accept	transport management **for** Day	[T7/ac]
	with	**the** addressee **of the** statement **is the** transport manager	
else	reject	transport management **for** Day	[T7/rj]
	with	**the** addressee **of the** reject **is the** transport manager	

Figure 5.22 Action Rule Specification for A7 (4)

when	transport completion **for** Transport is requested		(T6/rq)
	with	**the** day of transport **of** Transport **is some** day	
		the car of Transport **is some** car	
		the from-branch **of** Transport **is some** branch	
		the to-branch **of** Transport **is some** beanch	
assess	*justice:*	**the** performer **of the** request **is the** transport manager;	
		the addressee **of the** statement **is the** transporter **for** Transport	
	sincerity:	< no specific condition >	
	truth:	*transport is doable*	
if	*complying with the assessment is considered justifiable*		
then	promise	transport completion **for** Transport	[T6/pm]
	with	**the** addressee **of the** promise **is the** transport manager;	
		the promised production time **of** Transport **is**	
		within the requested production time **of** Transport	
else	decline	transport completion **for** Transport	[T6/dc]
	with	**the** addressee **of the** decline **is the** transport manager	

Figure 5.23 Action Rule Specification for A6 (1)

Then I presented and discussed the action rules pertaining to the actual transportations. They are exhibited in Figures 5.23 and 5.24. One of the conditions for the claim to truth in the action rule in Figure 5.23, is that it must be doable to execute the transport. This informally specified condition is supposed to cover all kinds of logistic conditions (is it possible to drive from A to B at the requested day? How long will it take?), as well as the availability of a transporter (Mik, Ferre, or Carlo) at that day. A similar condition is included in the action rules in Figure 5.24.

when	transport completion **for** Transport is promised		(T6/pm)
assess	*justice:*	**the** performer **of the** promise **is the** transporter **for** Transport;	
		the addressee **of the** promise **is the** transport manager;	
	sincerity:	< no specific condition >	
	truth:	*it is possible to execute the task on day of transport of Transport*	
if	*complying with promise is considered justifiable*		
then	execute	transport completion **for** Transport	[T6/ex]
	state	transport completion **for** Transport	[T6/st]
	with	**the** addressee **of the** statement **is the** transport manager	

when	transport completion **for** Transport is stated		(T6/st)
assess	*justice:*	**the** performer **of the** statement **is the** transporter **for** Transport;	
		the addressee **of the** statement **is the** transport manager;	
	sincerity:	< no specific condition >	
	truth:	*the car of Transport has been delivered at the to-branch of Transport*	
if	*complying with statement is considered justifiable*		
then	accept	transport completion **for** Transport	[T6/ac]
	with	**the** addressee **of the** acceptance **is the** transporter **for** Transport	
else	reject	transport completion **for** Transport	[T6/rj]
	with	**the** addressee **of the** rejection **is the** transporter **for** Transport	

Figure 5.24 Action Rule Specification for A6 (2 and 3)

The actual execution of T6, the car transport, includes a lot of things, of course. In the way it is modelled in the first action rule in Figure 5.24, we assume that the executor has the proper competence to perform

the task (for which additional work instructions may apply, Cf. Figure 4.3). Of course, it is possible to decompose the product P6, and to get enclosed transactions in transaction T6, but there is need for it.
In order to complete the essential model of the transportation part of the case RAC, I add the corresponding part of the Fact Model in Figure 5.25. The way in which P7 is modelled is also typical and generic for periodic activities. Of course, instead of the period of a day, another time period may be chosen.

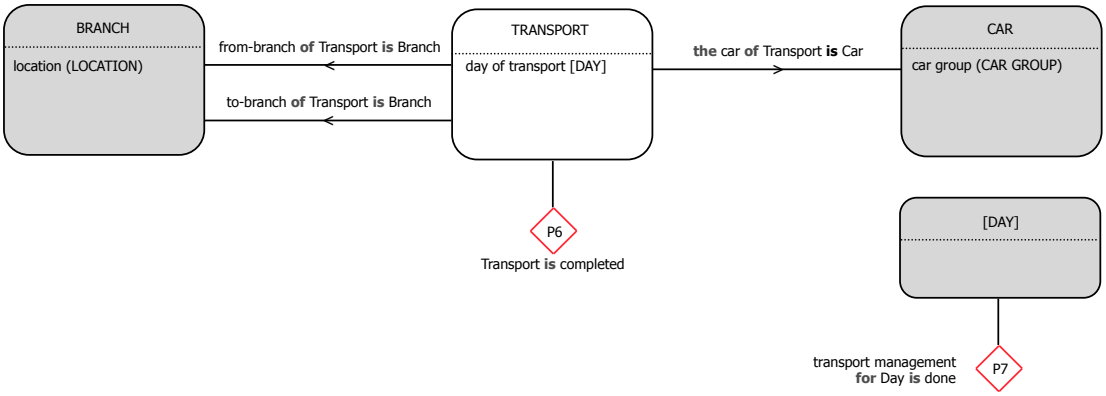


Figure 5.25. Object Fact Diagram of Rent-A-Car (3)

While modelling the case RAC, we have seen at several points that applying the PSI theory, helps effectively in clarifying issues, I said. But let us elaborate this a bit further. What we have done is producing the essential model of the home branch of RAC. Now, suppose that Janno and Ties decide that the whole contracting process must become the same for all branches, and web-based. So, all potential renters visit the RAC website and specify what they want, including the branch where they want to pick up the car. The question then is what this decision would mean for the organisation of RAC, and more specifically, what would it mean for the essential model of the home branch. This was a challenging question for the board members, and it therefore took them quite some time before they agreed on the answer. Of course, the answer was that it wouldn't affect the essential model at all, because this model is completely abstracted from all implementation issues. But it would affect the implementation, obviously. At the same time, they all recognised that the essential model of the home branch is also the essential model of all other branches.

This concludes my presentation and discussion of the case Rent-A-Car, I said to the members of the board, who had had another long session. But they were still quite attentive. Maybe because of the light collision with the business architect, I thought, but I was not sure. No matter, they have been attentive. That is what I had hoped for and what I had got. After some final small questions and clarifications, everybody was satisfied. I am sure that Eve represented the opinion of all of them when she expressed her gratitude to me about all the efforts I had taken, and all the patience I had brought in while they were trying hard to unlearn their old ways of thinking about business processes and information systems, and to assimilate a new, and proper, one. Does it also hold for this business architect?, was the question that crossed my mind. Did he also try hard? At that moment, the business architect raised his hand to ask a question. I would find it useful, he said, if the current functionary types, like salesmen and purchasers, and/or the specific employees could be mentioned in the actor role symbols in the OCDs. Now, I have to check the case description every time in order to know how things are currently going. That is a really good question, I replied, thank you. Let us spend some time on discussing it. To start with, I do know this need from my own practice too, and I know some DEMO Masters who write function and/or employee names in the OCD. However, I am always reluctant to do this, because I like the ontological models to be truly ontological. What I do instead often, is to produce an Actor-Function-Matrix, in which functionary types may also be replaced by employees. Let me present the matrix for the case RAC, as an example. It is shown in Figure 5.26. It says that the person Chiara, and the other desk employees, fulfil actor role A1 (rental concluder), and that Mik, Ferre, and Carlo fulfil the actor roles A3 (car issuer) as well as A6 (transporter). In addition, Mik fulfils actor role A7 (transport manager). This matrix is what I meant, said the business architect, it is very useful.

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

Figure 5.26 Actor Function Matrix of Rent-A-Car

6 EXPLORING ORGANISATIONAL ESSENCE

In the previous chapters, we have touched upon the notion of organisational essence in several ways. I have provided you with the precise definition of the essential model of an enterprise's organisation, which is the ontological model of its O-organisation, extended with information links, so comprising both interaction and interstriction. This definition enables you to understand essential models, and maybe even to produce them, but it does not yet answer fully and in depth the question of what organisational essence precisely is. Frankly spoken, I don't know whether there is one final and definite answer to this question, so let me just tell you what organisational essence means to me, and how I apply the notion in my work as enterprise engineer.

6.1 FUNCTION AND CONSTRUCTION

As I have said in Chapter 1, the term 'organisation' refers to the construction perspective on enterprises, while the term 'business' is reserved for the function perspective. The distinction between the function and the construction of an enterprise, and in fact of any kind of system, is really fundamental and crucial in the analysis and design of enterprises. It is an example of applying the sapience 'separation of concerns' (Cf. Figure 1.1) to the enormous diversity and complexity of enterprises. Let me start to discuss the difference by explaining the construction perspective first.

Taking up the *construction* perspective, is considering something as it is, without thinking at all about what you could do with it, how you could use it for whatever purpose you have in mind. I agree, it is not easy to think purely constructional. Look for example at the coffee cup on your desk, or at your cell phone lying next to it, or your car, and try to think only of how these things are constructed, how they could have been fabricated or assembled, without at the same time seeing the 'functional' cup, cell phone, or car. If you succeed in doing such exercises, then you are studying things truly and purely from the construction perspective. Obviously, this perspective is crucial for engineers. Equally crucial is the insight that every system, and thus every enterprise, has (at any moment) only one construction. That is why I said that taking the construction perspective means considering a system as it is.

Taking up the *function* perspective on a system, is thinking only of what you could do with it, how you could use it, while ignoring for the time being its construction. Immediately, the fundamental difference between function and construction becomes apparent: construction is an inherent property of a system (a system *is* its construction), while function is a relationship between the system and a stakeholder. Actually, one cannot speak of *the* function of a system; every system may 'have' at least as many functions as there are stakeholders. At the same time, it apparently makes sense to ask for the function of a cup, or a cell phone, or a car, in a seemingly non-subjective way. So there is more to be said. Let me elaborate on this issue for artefacts. Artefacts are designed systems, and designers always have a purpose (or a few purposes) in mind when conceiving a construction that would enable stakeholders to achieve their purpose(s). Normally, the purpose(s) in the mind of the designer is/are not arbitrary and private, but something that a community of stakeholders share. Consequently, there is a community of cup users, of cell phone users, of car users, etc. So, artefacts have an intended function for a class of stakeholders (usually the intended users of the artefact), and this intended function is commonly called *the function* of the artefact. But, as we have seen, function is a relationship between the artefact and a stakeholder. Therefore, next to this 'primary' function, as intended by the designer, people 'invent' many other functions. E.g., one can use a cell phone as a paperweight, and one can do many things with a car, next to driving in it.

Let us have a brief look at the design process of an artefact, because the distinction between function and construction is particularly relevant in this process. It is exhibited in Figure 6.1. A design process always consists of two sub processes: function design and construction design. The result of *function design* is the determination of the intended function(s) of the artefact. Since there may be many, and different, stakeholders, the resulting function(s) is/are generally a compromise. This holds for cups, for cell phones, for cars, etc. Nevertheless, in the end there is agreement about what the function(s) will be, documented in the *functional specifications*. The result of *construction design* is the conception of some construction that is able to bring about the intended function(s), once it is implemented. It is documented in the *constructional specifications*. As said before, next to the intended function(s), a construction may serve all additional purposes that any stakeholder may come up with. The division of a design process into these two sub processes does not mean that they take place fully consecutively. They may be intertwined to a large degree, or at least there will be some iteration, because, in general, the resulting construction is a balanced compromise between (reasonable) functional specifications and (feasible) constructional specifications. Yet, the ability of a designer to make a clear separation between the concern for function and the concern for construction, is crucial for good design.

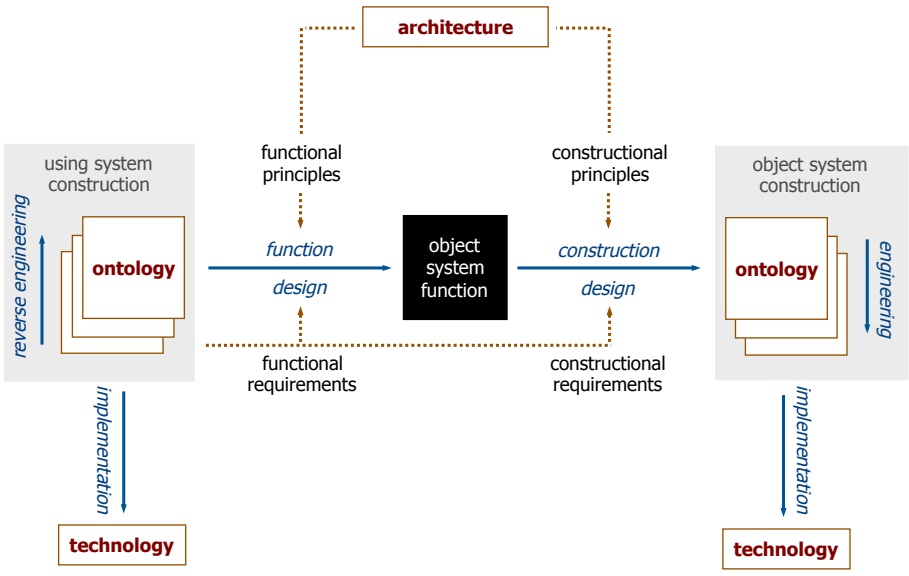


Figure 6.1 The Generic System Development Process (GSDP)

The GSDP in Figure 6.1 explains the complete process of developing a system, called object system, for the future benefit of another system, called using system. The first major phase in the development process is function design. It starts from the ontological model of the using system and ends with the functional specifications of the object system (the object system function in Figure 6.1). This design phase is fed by the functional requirements, provided by the using system, and the functional principles of the applicable architecture (see for its definition Chapter 1). The second major phase is construction design. It is the most creative one, because one has to bridge two universes of discourse: the one of the using system and the one of the object system. This phase is fed by the constructional requirements, provided by the using system, and the constructional principles of the applicable architecture. Ideally, construction design proceeds in two consecutive phases. In the first phase, the ontological model of the artefact is produced. Ontological means that it does not contain any clue to a possible implementation. In the second phase, the ontological model is detailed, to such a degree that the final model can be implemented with

the available technologies. This second phase is called engineering[23] or implementation design. In current practice, within the area of business processes and information systems development I mean, the ideal of ontological design is rarely met. Mostly, the highest level constructional models (like UML models) are already implementation dependent to quite some extent.

Enterprises are just as much artefacts as cups are, or cell phones, or cars. They are purposefully designed and implemented systems, that consequently can be re-designed and re-implemented. This is the starting position of enterprise engineering. Every change, even the slightest one, encompasses at least some re-implementation, often also some implementation re-design, and sometimes also some ontological re-design. The really challenging changes, like enterprise transformations, encompass quite some (ontological and implementation) re-design, and subsequent re-implementation, and often also functional re-design. To accomplish this, is the profession of enterprise engineers.

Both the function (business) perspective and the construction (organisation) perspective are necessary to think up enterprise changes and to implement them. The one is not more important than the other; both the function (business) design and the construction (organisation) design need to be conducted properly. The traditional organisational sciences provide all knowledge for proper business design. The missing part up to now is the knowledge that is needed for proper organisation design; this is what enterprise engineering contributes. It does so in particular through the notions of enterprise ontology, enterprise architecture, and enterprise governance (as defined in Chapter 1).

6.2 ORGANISATION PATTERNS

As I said before, the notion of enterprise ontology comprises more than its 'technical' definition in the PSI theory. It stimulates me to keep deepening my understanding of organisations. As an example, I have come to find the notion of authorisation very powerful. I have learned to apply it effectively in producing ontological models. It is the most elegant mechanism to organise things. The starting point is that a subject must be authorised to fulfil an actor role. This authorisation, however, is itself a product, that is, the

[23] The word "engineering" is used here in the narrow sense, as opposed to its use in the broad sense, like in "enterprise engineering".

result of some transaction. Let me give an example for clarification. The tennis club Volley wants to limit access to the tennis courts to a selected group of people. It does so by applying the notion of membership. However, membership is nothing else than authorisation: only members are allowed to play the role of initiator in 'tennis transactions'. It is not fundamentally different from the authorisation of Eve to fulfil actor role A1, including the right to delegate some tasks to Adam. As an other example, buying a ticket for the theatre or the stadium is nothing else than getting an authorisation: only ticket holders are allowed to fulfil the role of initiator in 'entrance transactions'.

An even more enlightening application of this authorisation mechanism, is the outsourcing of transactions, in the O-, the I-, or the D-organisation. As a concrete example, suppose that Volley decides to outsource the calculation of the first fee for new memberships. Clearly, this calculation is an informational transaction in the I-organisation of Volley. Eve, as holder of actor role A1 starts this transaction in her green shape; the executor role is some internal informational actor role of Volley. Outsourcing means that one or more subjects in another enterprise will fill this role; let us call this enterprise Calcul. Eve remains the authorised initiator of the transaction, but in the new situation this authorisation has to be ratified by Calcul. To accommodate this, a new original transaction must be put into place, of which the initiator is (an actor role in) Volley, and the executor is (an actor role in) Calcul. The transaction product is the ratification or approval of Eve's authorisation for being the initiator of the calculation transaction. The calculation transaction is by definition an informational transaction in the I-organisation of Volley, but the executing subject is an employee of Calcul. So, it is the sourcing of the holders of actor roles that determines which enterprise (as a societal institution) is responsible for executing certain transactions. Note, however, that this is an implementation issue, not an ontological one. The crucial point in the example is that the executor role of the calculation transaction does not belong to the I-organisation of Calcul, since it does not support the O-organisation of Calcul.

Summarising, the authorisation of a subject to fulfil the initiator role in a 'border' transaction kind is, on principal, twofold. On the one side, it has to be granted by the 'customer' organisation. On the other side, the authorisation has to be granted by the 'supplier' organisation. Both instances of granting are the result of an original transaction, regardless how these transactions are named: authorisation, or granting, or ratification, or just contract, or whatever. The case RAC also provides an example of this mechanism. By having concluded a rental contract, the driver has the authority to initiate a transaction T3 (car issuing). This is the justice check in the corresponding action rule (Figure 5.11).

6.3 REALISATION AND IMPLEMENTATION

Somewhat related to the previous discussion is the question how information links are realised. Let us have a closer look at it. As examples I use the two information links between actor role A1 and the external aggregate transactions AT1 and AT2 in the case Volley (see Figure 4.4). But first a more general question: if some (authorised) person asks you for your day of birth, what would you say? Probably you will tell the person your day of birth, or you will write it down on a form. But how do you know it? How truthful is your answer? What I am pointing at is that every fact in our (intersubjective) world is the result of an original transaction (or of a computation based on original facts). This holds also for your birth fact. Therefore, the next interesting question is: who is the executor of this transaction, and for that reason the 'owner' of your birth fact? In our modern, western society, the correct answer is: some authorised civil servant in the municipality where you are born. And the initiator? Probably your father, who told the civil servant about the biological event. And at some point in time, you have been informed by your father or mother about both the biological and the official birth fact. So, actually, everyone's knowledge of his/her birth fact is 'second hand' knowledge.

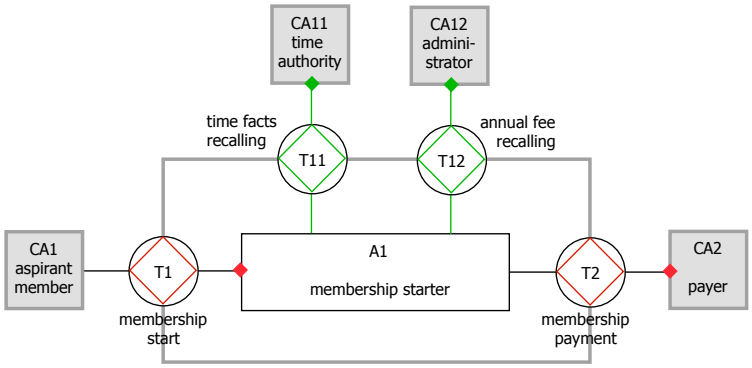


Figure 6.2 Extending the OCD of Volley with informational transactions

So, if someone asks you for your day of birth, he/she is mostly satisfied with the 'second hand' answer you can provide; there is no need to be perfectly sure. This also happens in the case Volley. The aspirant member provides his/her day of birth to Adam, who records the fact (and some other facts) in the member register. This recording is a documental transaction 'below' the informational transaction of remembering the facts that the aspirant member has shared with him. Let us abstract from the documental level, so from the D-organisation of Volley, and only consider a few I-organisation things, in particular during an execution of action rule 2 of A1, as exhibited in Figure 4.11. Two facts have to be recalled in order to compute the first fee: the annual fee, and the current month of the year, as specified in Figure 4.9. This is done in the informational transactions T11 and T12 respectively, as shown in Figure 6.2 (Note: for the sake of simplicity, I leave out the computation itself, as this would require another informational transaction). Similarly, if Adam wants to get time facts (like the current month of the year), he initiates a transaction T11, and if Eve wants to get the annual fee, she initiates a transaction T12. This may seem a bit overdone, but in enterprise engineering, nothing is taken for granted; one must be specific about everything that has to be done. So also in this case, although we know that actor role CA11 is fulfilled by Adam himself. The point is that this allocation of actor roles to subjects may change, and through the way I have modelled it now, I am prepared for every future change in the implementation. To indicate that T11 and T12 are informational transaction kinds, the diamond in the transaction symbols is coloured green. Next, the small green diamonds on the edges of the actor symbols CA11 and CA12 indicate that the role holders of these (composite) actor roles take on their green shape in producing the transaction products.

There is one fundamental assumption underlying the whole PSI theory, which also holds for its root theory, namely Habermas' theory of communicative action. It is that actors strive for consensus. For example, a business process, as the one shown in Figure 4.2, can only be completed successfully if the participating actors are striving for making it a success. If one or more are obstructing or cheating, we are lost. In order to check the willingness of actors to strive for consensus, the claim to sincerity is included in every action rule. But there is more to be said about ensuring that actors behave as the are expected to do. Next to being responsible for the correct execution of a transaction, every actor is also responsible for seeing to it that the executors of transactions that he/she has initiated, do what they have promised to do. These two responsibilities are actually the same. Let me take as an example the carrying out of a transaction T3, initiated by an actor A1 (i.e., some subject holding actor role A1) and executed by an actor A3 (Figure 4.2). This actor A3 is also initiator of a transaction T7 and of a transaction T8, and the results of these transactions are needed for completing the T3. Therefore, in order to live up to his/her promise to-

wards A1, A3 must also see to it that the actors A7 and A8 do what they have promised to do. This implies that A3 may need to point out explicitly to A7 and/or A8 that they must keep their promises. It is not sufficient for A3 to just sit and wait; that would be irresponsible.

So, generally spoken, every actor may need to 'chase' the executors of the transactions that he/she has initiated, in order to deliver their products, on time and with the agreed quality. This initiator responsibility of every actor may even imply that he/she has to revoke his/her request, and to carry out a similar transaction with another actor. Such problems typically happen in supply chains, which are actually tree structures, as we know by now. What has been said about the initiator responsibility, does not affect, of course, the prime executor responsibility of every actor, namely to deliver the agreed upon product, and on time. In other words, actors can never plea in excuse for not having done their duties as executor, that they were not reminded of it by the initiator of the transaction.

The good news is that, in my experience, most people love to get authority and to bear the corresponding responsibility. Because it allows them to exercise their competence, they find sense in their work. People in such a working environment, will consider "striving for consensus" a natural and obvious condition.

6.3 THE GENOTYPE AND PHENOTYPE OF ORGANISATIONS

Let me finish this introductory book in enterprise engineering with the presentation of a metaphor that helps me a lot in understanding the actual operation of an organisation, in connection with its ontological model. Here we go. An organisation, as defined by the ontological models of its three aspect organisations, is implemented by allocating subjects to the actor roles and by assigning appropriate technological support to the coordination acts/facts and production acts/facts. Once implemented, an organisation can become operational. The metaphor I use is to envision an organisation as a biological organism. As you may know, a biological organism, for example a human body, has a genotype and a phenotype. The genotype of an organism is its genetic make-up; it is recorded in the DNA of the organism. The genotype of an organism determines to a large extent its phenotype, which is the totality of observable properties. But also environmental influences play a role; sometimes they are even dominant. As everyone knows, two persons with the same genotype, so an identical twin, may get quite different phenotypes in the course of time. Conversely, two persons with remarkably similar phenotypes, may have quite different genotypes.

Next, I make a distinction between the generic and the specific genotype of an (aspect) organisation. The *generic genotype* is 'recorded' in the Constructional Model of the (aspect) organisation. The *specific genotype* is 'recorded' in its Action Model. The Process Model and the Fact Model serve to clarify the connections between the Constructional Model and the Action Model, as we have seen. To illustrate this, two rental car companies are very similar in the generic genotype (the CM) of their O-organisation, but they may differ substantially in the specific genotype (the AM). Similar remarks hold for the other aspect organisations. Next to that, the two car companies will differ in the way in which their organisations are implemented. I call the implementation of an (aspect) organisation its *phenotype*. The phenotype of an organisation is what one usually represents in a Flow Chart or some other kind of process diagram. The genotype then is the 'deep structure' that is 'hidden' the phenotype. Ideally, the genotype has served as a template for the implementation. So, in addition to the difference in genotype, the two car companies may differ a lot in their phenotypes.

Let us call the collective action rules of an actor role the specific genotype of the actor role. So, actors with the same actor role are organisational cells with the same genotype. Figure 6.3 exhibits what I am trying to convey. It is based on the OCD that is shown in Figure 6.2. There is an actor role A1, an actor role A2 (contained in CA2), and a transaction kind T2; A1 is the initiator role of transactions of the kind T2, and A2 is the executor role of these transactions. In Figure 6.3, actors are identified by <actor role>:<subject> or <actor role>:<subject 1>/<subject 2>. In the latter case, subject 1 has the mandated authority, and subject 2 the delegated authority to be executor of the transaction kind that is drawn as a full diamond. So, we have actors A0:Anna, A1:Eve/Adam, A2:Anna, A11:Adam (which is included in CA11 in Figure 6.2), and A12:Adam (which is included in CA12 in Figure 6.2). Note that many more subjects may be authorised simultaneously for holding the mentioned actor roles, but I will only consider the ones in Figure 6.3.

The genotype of actor A1:Eve/Adam is A1, and the genotype of actor A2:Anna is A2. It means that A1:Eve/Adam has the ability to be initiator of transactions of the kind T2, and A2:Anna has the ability to be executor of these transactions. As soon as A1:Eve/Adam addresses herself/himself to actor A2:Anna, the two cells 'couple', which means that they are now able to carry out a transaction of the kind T2. This is shown on the right side of Figure 6.3. The actors are now in the action mode, one could say. Biological cells interact through the exchange of molecules. Organisational cells interact through the exchange of commitments, represented by coordination facts. As long as the transaction goes on, the two cells stay

coupled, but as soon as the transaction ends (either successfully or unsuccessfully), the cells 'decouple'. Then they return to the idle mode, one could say, ready to couple again with any suitable partner. Suitable means that the new partner has the right genotype.

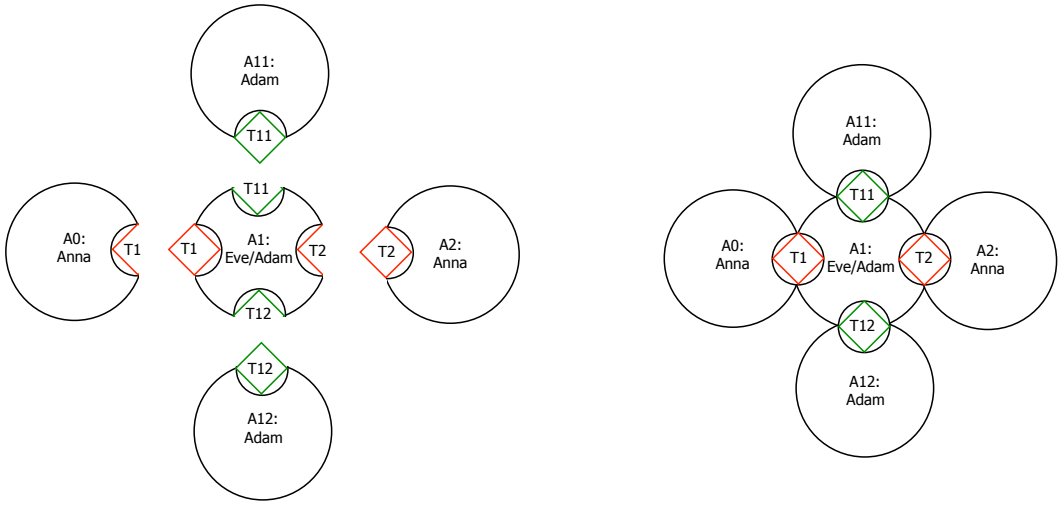


Figure 6.3 Illustration of the organism metaphor

Let us look particularly to the execution of action rule 2 of A1, as shown in Figure 4.11. In the response part of this action rule, the annual fee has to be calculated, in accordance with the Fact Definition List in Figure 4.9. To this end, actor A1:Eve/Adam has to get to know the annual fee and the year of the starting day. The first fact is provided in a transaction T12 by actor A12/Adam, the second one in a transaction T11 by A11/Adam. As we know from the Flow Charts in Figures 3.2 and 3.3, Eve calculates the first fee, and Adam requests for the payment. So, there is also a switch between Eve and Adam in holding actor role A1, but I will not elaborate this, nor will I elaborate the calculation itself, as said earlier.

Figure 6.3 also shows that actor A1:Eve/Adam has the ability to be executor of transactions T1. Actually, actor A1:Eve/Adam can only 'couple' with actor A2:Anna if it is already 'coupled' with some subject that is able to be initiator of transactions T1, for example A0:Anna. Only if all connections are made can actor A1:Eve/Adam get in the action mode. As soon as this is the case, the other actors may enter their action mode. What one must imagine, is that a concrete business process instance is built up, as a tree structure of transactions and actors, according to the generic genotype of the organisation, and the subsequent specific genotypes of the involved actor roles. However, the corresponding phenotypes must allow the needed 'coupling'. For example, if A1 can only accept written messages from A0, while A0 is only able to communicate in speaking, the connection cannot be made. That's why contemporary information systems need multi-channel interfaces.

Next to clarifying the concrete operation of organisations, the biological organism metaphor makes the need for operational management evident. By this I mean that at any point in time, there must be a sufficient number of organisational cells (actors) available of the diverse genotypes, in order to accommodate for the present need for their services. So, *operational management* is about creating, as well as destroying, organisational cells (actors). These activities are of course just the production acts in 'management' transactions, which I referred to as allocation transactions before. Moreover, to make operational management possible, there must be a sufficient number of competent subjects present at any point in time. So, there is also a need for *tactical management*. I will not elaborate this issue, but one can expect that it goes largely along the same line of reasoning as I applied to operational management.

EPILOG

After my discussion of the basics of enterprise ontology with the board of Volley, I went on summer holiday with a friend. We have been hiking in the Picos de Europa, a magnificent mountain area on the border of the Spanish provinces Asturias and Cantabria. It was a wonderful holiday, during which I also had memories, pleasant ones at that, of the meetings with the board of Volley. They reassured me once more that the only way out of the current world-wide mess of applying ICT in enterprises, is the way of enterprise engineering.

A week after our return from the holiday trip, I played tennis with Celestine. After the match we drank tea and talked about all kinds of things. Celestine happens to be the daughter of Eve, something I had no clue of before. She told me that Eve was quite upset about a recent turmoil in Volley. A new member, let me call him Mr. Z, had told her that he is the new CEO of Rent-A-Car. Apparently, Janno and Ties have stepped back a bit. In this function he had purchased licenses on an ERP system for all departments, in all branches. Everyone had to work with it. Celestine also told me that Mr. Z happens to be Adam's son, which exacerbated the situation to Eve.

Of course, I was quite upset by this news. I think it is close to criminal to sell standard software implementations, like ERP systems, of essential models that have never been produced! What makes it worse is that neither the ERP-system seller nor the naive customer, are aware of this. But that doesn't make it less true: an ERP application, like any other enterprise information system, is by definition some (most likely incomplete) implementation of some essential model (most likely unknown). Every organisation has its own genotype, and as long as one does not know this genotype fully, 'transplantations of extraneous parts' are quite risky. The weird thing in current practice is that this genotype is mostly unknown, and that consequently the seller is just bullshitting when he/she assures the customer that the offered ERP application fits the customer's organisation.

The news made me happy too. It proved that Eve had understood the message of enterprise engineering!

In this book, I have emphasised enterprise ontology, because this notion is most appropriate and effective, compared to current approaches to business process management, to organisational change, and to the application of ICT. It is also the most difficult one to grasp; it requires nothing less than a paradigm shift. One has to learn a new way of thinking, and to unlearn existing ones, because they are inappropriate and ineffective. Indeed, ontological thinking is new, unknown, and therefore not immediately loved, as I have experienced in my professional career up to now. But things are changing. I sense a growing interest in ontological thinking, and a growing awareness of its indispensability. Unmistakably, the paradigm shift from an application-centric view to an organisation-centric view is taking place.

I like to end as a fairy at the cradle of a new-born princess, with these three prophecies:

Some day, all enterprises will have made their genotypes explicit, recorded in the ontological model of the O-organisation, of the I-organisation, and of the D-organisation. They will use these models for every organisational change, like a neural surgeon will use his/her essential knowledge of the human body for 'changing' the neural system, or like a building architect will start from the existing constructional drawings to make changes and analyse the consequences before having them implemented.

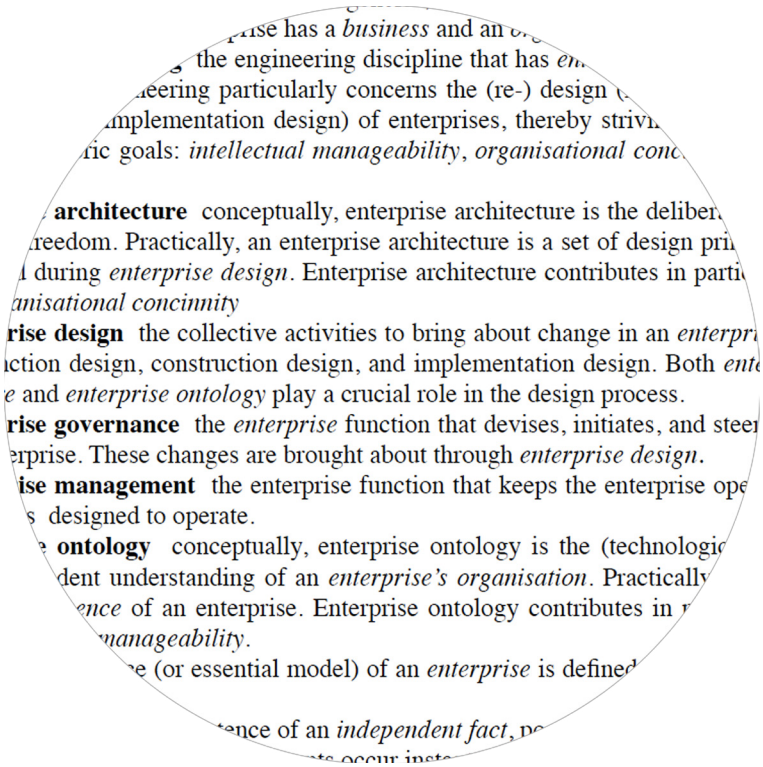
Some day there will be software engines that are fed with the complete essential model of an enterprise, and that consequently become the truthful software mimic of the enterprise's organisation. This generated system would be the ultimate enterprise information system, free of all errors that are made in the current way of designing, engineering, and implementing application software. In addition, all actors use the same 'transaction app' to carry out their transactions.

Some day, people in all enterprises will be empowered with the authority and responsibility they need and deserve, and organisations will be transparent. Because of this, people can give full meaning to their professional lives, while acting from their wisdom and love. Consequently, very little supervising management will be needed.

These prophecies can come true if you help me make it happen.

GLOSSARY

(MET NEDERLANDSE VERTALING VAN DE TERMEN)



rise has a *business* and an

the engineering discipline that has

eering particularly concerns the (re-) design

mplementation design) of enterprises, thereby striv

ric goals: *intellectual manageability*, *organisational con*

architecture conceptually, enterprise architecture is the deliber

reedom. Practically, an enterprise architecture is a set of design pri

during *enterprise design*. Enterprise architecture contributes in parti

anisational concinnity

rise design the collective activities to bring about change in an *enterpr*

ction design, construction design, and implementation design. Both *ent*

e and *enterprise ontology* play a crucial role in the design process.

rise governance the *enterprise* function that devises, initiates, and stee

rprise. These changes are brought about through *enterprise design*.

ise management the enterprise function that keeps the enterprise ope

s designed to operate.

ontology conceptually, enterprise ontology is the (technologi

dent understanding of an *enterprise's organisation*. Practically

ence of an enterprise. Enterprise ontology contributes in

manageability.

e (or essential model) of an *enterprise* is define

ence of an *independent fact*, p

act (handeling) the atomic unit of action in *organisations*. Two kinds of acts are distinguished: *coordination acts* (C-acts) and *production acts* (P-acts). Acts are performed by *actors*. [see also *transaction*]

action rule (actieregel) [synonym for *business rule*] a guideline for an actor for dealing with an *agendum* kind. It consists of three consecutive parts: the *event part*, the *assess part*, and the *response part*. Action rules are the procedural equivalent of *business laws*.

action rule specification (actieregelspecificatie) [representation form of the *action model*]

action model (actiemodel) one of the four sub models of the *ontological model* of an *aspect organisation*. It consists of a set of *action rules*.

actor (actor) a *subject* in his or her fulfilment of an *actor role*.

actor role (actorrol) the unit of *authority* in *organisations*. [see also *elementary actor role* and *composite actor role*]

addressee (geadresseerde) [*property* of *process step*] the *actor* to whom the *coordination act* is addressed.

agenda (agenda) a set of *agendums*. Every *actor* has his/her own agenda.

agendum (agendum) a *coordination event* to which an *actor* has to respond.

agent (agent) an artefact that is used by an *actor* as its digital proxy. Unlike actors, agents cannot be held responsible for their deeds. Therefore, agents can only support actors.

aggregate transaction kind (geaggregeerd transactiesoort) a collection of *transaction kinds*. Aggregate transaction kinds may be useful if one does not (need to) know exactly the constituent transaction kinds (e.g., because they are outside the *Scope of Interest*).

assess part (vaststellingsdeel) the part of an *action rule* that contains the conditions to be evaluated. These conditions are divided according to the three *validity claims*.

aspect organisation (aspectorganisatie) the *organisation* of an *enterprise* consists of three disjoint aspect organisations: the *O-organisation*, the *I-organisation*, and the *D-organisation*.

attribute (attribuut) instance of an *attribute type*.

attribute type (attribuuttype) a *property type* that is a purely mathematical mapping from an *object class* or a *value scale* to a value scale.

authority (bevoegdheid) the entitlement (to be) assigned to a *subject* to fulfil an *actor role*. [see also *responsibility* and *competence*]

bank contents table (bankinhoudstabel) table of *production fact types* and the *transaction banks* in which their instances are contained.

basic transaction pattern (basistransactiepatroon) [see *transaction pattern*]

business (business) term to refer to the *function perspective* on an *enterprise*, in particular the function as perceived by the consumers of its *services*. [see also *organisation*]

business event (bedrijfsgebeurtenis) [synonym for *coordination event*]

business fact (bedrijfsfeit) [synonym for a *fact* in the *production world* of an *organisation*]

business law (bedrijfswet) a declaratively formulated rule that determines the lawfulness of *states* and *transitions* in the *coordination* and the *production world* of an *organisation*.

business object (bedrijfsobject) [synonym for an *object* in a *production world*]

business process (bedrijfsproces) [synonym for a *process* in the *O-organisation* of an *enterprise*]

business process model (bedrijfsprocesmodel) a tree structure of *transaction kinds* (possibly one) in the *O-organisation* of an *enterprise*. Every transaction kind in the tree is enclosed in a transaction kind on the next higher level (except the 'highest' transaction kind), and encloses one or more transaction kinds on the next lower level (except the 'lowest' transaction kinds).

business rule (bedrijfsregel) [synonym for *action rule*]

category (categorie) synonym for "extension of an original unary fact kind".

causal link (causale verbinding) link between a *coordination act* and its resulting *coordination fact*, indicating that the fact is the result of the act. [see also *process step*]

claim to justice (juistheidsaanspraak) one of the three *validity claims* in a *coordination act*. The claim to justice regards the *authority* of the *performer* to perform the act.

claim to sincerity (waarachtigheidsaanspraak) one of the three *validity claims* in a *coordination act*. The claim to sincerity regards the sincerity of the *performer* in performing the act.

claim to truth (waarheidsaanspraak) one of the three *validity claims* in a *coordination act*. The claim to truth regards the (potential) existence of the *product* concerned.

cognitive correspondence (cognitieve overeenstemming) the mutual understanding of a *coordination act* at the *informa level* of *communication*. Cognitive correspondence is reached if the *addressee* thinks he/she has understood the content of the message correctly.

commitment (verplichting) the being dedicated of the *performer* of a *process step* to its *intention*. It entails that the performer feels the moral obligation to let his/her future actions be in agreement with this intention.

communication (communicatie) the sharing of thoughts between *subjects*. Communication is brought about by exchanging *information*.

competence (competentie) the collective capabilities of a *subject* that makes him/her eligible to perform specific *acts*, in particular to be assigned the *authority* to fulfil the *executor role* in *transaction kinds*. [see also *responsibility*]

complete transaction pattern (compleet transactiepatroon) [see *transaction pattern*]

composite actor role (samengestelde actorrol) a composite actor role consists of two or more *elementary actor roles* and the *transaction kinds* between them.

confirmation (bevestiging) the expression by the *addressee* of a *coordination act* towards the *performer* that he/she thinks they have reached *notational*, *cognitive*, or *social correspondence*.

construction model (constructiemodel) one of the four sub models of the *ontological model* of an *aspect organisation*. It contains the identified *transaction kinds* and *actor roles*, as well as the *initiator links*, the *executor links*, and the *information links* between them.

construction (constructie) there are two fundamentally different perspectives on any system: the construction perspective and the *function* perspective. In the construction perspective on an *enterprise* one considers the (objective) *organisation* of the enterprise, fully ignoring its (subjective) function(s).

coordination act (coördinatiehandeling) the atomic act in *transactions*. The result of a successfully performed coordination act is the creation of the corresponding *coordination fact*. [see also *process step*]

coordination event (coördinatiegebeurtenis) an *event* in the *coordination world*. The occurrence of a coordination event is identical to the coming into being of a *coordination fact*.

coordination fact (coördinatiefeit) a *state* element in the *coordination world*. A coordination fact is the result of a successfully performed *coordination act*. [see also *process step*]

coordination world (coördinatiewereld) one of the two *worlds* in which the *actors* in an *organisation* cause *transitions*.

creation time (creatietijdstip) [*attribute* of *fact*] the *point in time* at which a *fact* has or will come into being.

date (datum) indication of a day on a *time scale*. The Julian Date is taken to do this, which means indicating days by consecutive integers. Julian dates can be mapped to any calendar. *Time units* like year, month, and week, are dependent on the chosen calendar.

declared fact type (verklaard feittype) a *fact type* that is declared to be part of the *fact model* of an *aspect organisation*. [see also *derived fact type*]

dependent fact (afhankelijk feit) a fact that comes into existence dependent on, and together with, an *independent fact*. [see also *properties* and *attributes*]

derived fact specification (afgeleidfeittypespecificatie) [representation form of the *fact model*] the formal specification of *derived fact types*.

derived fact type (afgeleid feittype) a fact type that is defined on the basis of other (*declared* or derived) fact types in the *fact model* of an *aspect organisation*. [see also *declared fact type*]

disconfirmation (weerlegging) the expression by the *addressee* of a *coordination act* towards the *performer* that he/she thinks they have not reached *notational*, *cognitive*, or *social correspondence*.

discussion status (discussiestatus) *transaction status* in which the *initiator* and the *executor* have to 'sit together' and discuss what the cause of the 'exception' is and how to proceed. The *standard transaction pattern* has two discussion statuses: declined and rejected. The revocation patterns have one discussion status each: revoked.

documental (documenteel) there are three sorts of *production acts/facts*, and consequently of *actors/actor roles*, and *transactions/transaction kinds*: *original*, *informational*, and documental. Documental services comprise storing, retrieving, copying, and transmitting documents.

D-actor (role) (D-actor(rol)) [shorthand for *actor (role)* in the *D-organisation* of an *enterprise*]

D-organisation (D-organisatie) the *aspect organisation* of an *enterprise* that supports its *I-organisation* by providing *documental services*. It consists of *documental actor roles* and *transaction kinds*.

D-transaction (kind) (D-transactie(soort)) [shorthand for *transaction (kind)* in the *D-organisation* of an *enterprise*]

elementary actor role (elementaire actorrol) the *authority* to fulfil the *executor role* of a *transaction kind*. A subject holding an elementary actor role is assumed to be selected on the basis of his/her *competence*, and to exhibit *responsibility* in practising the assigned authority.

enterprise the general term to refer to any kind of collaborative activity by human beings. Examples: companies, governmental agencies, health care institutions, sports clubs, and building projects. Every enterprise has a *business* and an *organisation*.

entity (entiteit) an *independent* unary *production fact*.

enterprise engineering the engineering discipline that has *enterprises* as its objects of interest. Enterprise engineering pursues three generic goals: *intellectual manageability*, *organisational concinnity*, and *social devotion*.

enterprise architecture conceptually, enterprise architecture is the deliberate restriction of design freedom. Practically, an enterprise architecture is a set of design principles that are applied during *enterprise design*. Enterprise architecture contributes in particular to achieving *organisational concinnity*

enterprise design the collective activities to change an *enterprise*, consisting of *function* design, *construction* design, and implementation design. Both *enterprise architecture* and *enterprise ontology* play a crucial role in the design process.

enterprise governance the *enterprise function* that devises, initiates, and steers changes in the enterprise. These changes are brought about through *enterprise* (re-) *design*.

enterprise management the *enterprise function* that keeps the enterprise operating in the way it was designed to operate.

enterprise ontology conceptually, enterprise ontology is the implementation independent understanding of an *enterprise's organisation*. Practically, it is the PSI theory based model of the organisation. Enterprise ontology contributes in particular to achieving *intellectual manageability*.

essential model (essentieel model) the essential model of an *organisation* is defined as the *ontological model* of its *O-organisation*, to which *information links* are added. These links represent the informational services of the supporting *I-organisation*.

event (gebeurtenis) the coming into being of a *fact*. Events occur instantly, which means that their duration falls within one *time unit*. Events that occur in the same time unit, are said to occur simultaneously.

event part (gebeurtenisdeel) the part of an *action rule* that contains the *agendum* that the action rule is dealing with.

execution phase (uitvoeringsfase) the *transaction* phase in which the *executor* performs the *production act*. It starts from the *transaction status* 'promised' and it ends at the transaction status 'stated'.

executor link (executorverbinding) a link in a *construction model* between an *actor role* and a *transaction kind*, indicating that the actor role is the *executor role* of the transaction kind.

executor (role) (executor(rol)) a role of an *actor (role)* in a *transaction (kind)*. An actor (role) who is *authorised* to be the executor (role) of a transaction (kind), is *responsible* for performing the *production act* and the corresponding *coordination acts*, according to the *transaction pattern*.

fact (feit) elementary state of affairs in a *world*. A fact can best be conceived as a predicate over one or more *objects* or *values*. Every fact is an instance of a *fact type*. [see also *property* and *attribute*]

fact model (feitenmodel) one of the four sub models of the *ontological model* of an *aspect organisation*. It contains the identified *fact types* and *business laws* in the *production world* of the organisation.

fact type (feittype) the generic concept for modelling the *state* of a *world*. The extension of a fact type is an *object class*. Conversely, a fact type is the intension of an object class. Examples of fact types in a

production world: membership, rental, member_of_membership, renter_of_rental. Examples of fact types in a coordination world: membership_is_requested, rental_is_promised.

file (bestand) a physical embodiment of a *document*. A document may, for example, be inscribed by marks on paper or by magnetic bits in a computer. A document may be inscribed in many files; these files are called copies of the document.

forma (forma) term to refer to the *competence* of a *subject* to perform *coordination acts* at the *forma level* as well as *documental production acts*.

forma condition (formavoorwaarde) the collective prerequisites for reaching *notational correspondence* in performing a *coordination act*.

forma level (formaniveau) the level of *communication* in performing a *coordination act* at which the *performer* and the *addressee* strive for *notational correspondence*.

function (functie) there are two fundamentally different perspectives on any system: the *construction* perspective and the function perspective. Function is not a system property but a relationship between a system and a stakeholder; therefore, a system may 'have' many functions. In the function perspective on an *enterprise* one considers the functions of the enterprise as perceived by its various stakeholders. [see also *business*]

independent fact (onafhankelijk feit) a *fact* that comes into existence as the direct result of performing an *act*. [see also *dependent fact*]

informa (informa) term to refer to the *competence* of a *subject* to perform *coordination acts* at the *informa level* as well as *informational production acts*.

informa condition (informavoorwaarde) the collective prerequisites for reaching *cognitive correspondence* in performing a *coordination act*.

informa level (informaniveau) the level of *communication* in performing a *coordination act* at which the *performer* and the *addressee* strive for *cognitive correspondence*.

information (informatie) the expression by a *subject* of thought(s) in a form that is perceivable for other subjects. Briefly: information is from given thought.

information link (informatieverbinding) a link in a *construction model* between an *actor role* and a *transaction kind*, indicating that the actor role has access to the corresponding *transaction bank*. Through these links, the *interstriction* between actors is realised.

informational (informationeel) there are three sorts of *production acts/facts*, and consequently of *actors/actor roles*, and *transactions/transaction kinds*: *original*, informational, and *documental*. Informational services comprise remembering, recalling, and computing.

informational service (informationele dienst) there are two kinds of services that an I-organisation offers to its corresponding *O-organisation*: remembering *facts* and sharing facts.

initiator link (initiatorverbinding) a link in a *construction model* between an *actor role* and a *transaction kind*, indicating that the actor role is an *initiator role* of the transaction kind.

initiator (role) (initiator(rol)) a role of an *actor (role)* in a *transaction (kind)*. An actor (role) who is *authorised* to be the initiator (role) of a transaction (kind), is *responsible* for performing the corresponding *coordination acts*, according to the *transaction pattern*.

intellectual manageability (verstandelijke beheersbaarheid) one of the three generic goals of *enterprise engineering*. Intellectual manageability means that one has insight and overview in enterprise changes. It is achieved through appropriate and well-controlled reduction of complexity, as e.g. the one that is offered by the *essential model* of an organisation.

intention (intentie) [*property* of *process step*] the disposition of the *performer* of a *process step* regarding the *product*. Examples of intentions: 'request', 'promise', 'decline', 'state', 'accept', and 'reject' [see also *commitment*].

interaction (interactie) the active mutual influencing of *actors*, by performing *coordination acts*.

interstriction (interstrictie) the passive mutual influencing of *actors*, by taking into account the state of the *coordination world* and/or the *production world* when dealing with *agenda*.

I-actor (role) (I-actor(rol)) [shorthand for *actor (role)* in the *I-organisation* of an *enterprise*]

I-organisation (I-organisatie) the *aspect organisation* of an *enterprise* that supports its *O-organisation*, by providing *informational services*. It consists of *informational actor roles* and *transaction kinds*.

I-transaction (kind) (I-transactie(soort)) [shorthand for *transaction (kind)* in the *I-organisation* of an *enterprise*]

medium level (mediumniveau) the level of communication in performing a *coordination act* at which the *performer* and the *addressee* exchange the substances in which their messages are imprinted.

notational correspondence (notationele overeenstemming) the degree of mutual understanding of a *coordination act* at the *forma level* of *communication*. Notational correspondence is reached if the *addressee* thinks he/she has understood the form of the message correctly.

object (object) constituent element of a world. [see also *fact*]

object class (objectklasse) set of *objects* that conform to the same *fact type*.

object fact diagram (objectfeitdiagram) [representation form of the *fact model*]

ontological model (ontologisch model) a conceptual model of the *construction* and the operation of a system that is fully independent of its implementation.

operation (werking) the operating status of the *construction* of an *organisation*. It is guided by b*usiness rules* and *work instructions*.

operating principle (werkingsprincipe) the operating principle of an *organisation* is the ability and readiness of the participating *subjects* to enter into and comply with *commitments* regarding the bringing about of *products*.

order phase (orderfase) the *transaction* phase in which the *initiator* and the *executor* strive to reach consensus about the *product* that the executor has to bring about. It starts from the transaction status 'requested' and ends at the status 'promised' or 'quitted'.

organisation (organisatie) term to refer to the *construction perspective* on an *enterprise*. The organisation of an enterprise consists of a network of *actor roles* and *transaction kinds*. [see also *business*]

organisation construction diagram (organisatieconstructiediagram) [representation form of the *construction model*].

organisational concinnity (organisatorische harmonie) One of the three generic goals of *enterprise engineering*. Organisational concinnity means that the organisation of an enterprise is a harmoniously integrated whole. This is achieved through coherent and consistent *enterprise design*, guided by the applicable *enterprise architecture*.

original (origineel) there are three sorts of *production acts/facts*: original, *informational*, and *documental*. Original production acts comprise manufacturing, transporting, observing, deciding, and judging. Consequently, there are three similar sorts of *transactions/transaction kinds* (in which the production acts are executed) and *actors/actor roles* (the initiators and executors of these transactions).

O-actor (role) (O-actor(rol)) [shorthand for *actor (role)* in the *O-organisation* of an *enterprise*]

O-organisation (O-organisatie) (O from original) the *aspect organisation* of an *enterprise* in which *original* facts are brought about. It consists of *original actor roles* and *transaction kinds*.

O-transaction (kind) (O-transactie(soort)) [shorthand for *transaction (kind)* in the *O-organisation* of an *enterprise*]

performa (performa) term to refer to the competence of a subject to perform *coordination acts* at the *performa level* as well as *original production acts*.

performa condition (performavoorwaarde) the collective prerequisites for reaching *social correspondence* in performing a *coordination act*.

performa level (performaniveau) the level of communication in performing a *coordination act* at which the *performer* and the *addressee* strive for *social correspondence*.

performer (verrichter) [*property* of *process step*] the performing *actor* of a *process step*.

point in time (tijdstip) a particular value on a *time scale*, expressed in the corresponding *time unit*. Examples: week 36 (week), today (day), tomorrow at 11:25 hrs (minute).

process (proces) a sequence of *acts* in a (discrete event) system and their resulting *transitions* in its corresponding *world*.

process interpretation (procesinterpretatie) there are two interpretations of a *transaction kind* in the *construction model* of an organisation: the process interpretation and the *state interpretation*. In the process interpretation, a transaction kind is conceived as the *complete transaction pattern,* which will be passed through by all of the transaction instances.

process model (procesmodel) one of the four sub models of the *ontological model* of an *aspect organisation*. It contains the identified *transaction kinds* and the specific ways in which they are interrelated through *response links* and *waiting links*.

process step (processtap) the atomic building block of a *transaction process*. It consists of a *coordination act* and its resulting *coordination fact*. A process step has four core properties: the *performer*, the *addressee*, the *intention*, and the *product*.

process step kind (processtapsoort) [*property* of *process step*] every *process step* is of some kind. This process kind is referred to by two distinctive *properties: transaction kind* (which corresponds with *product kind*) and *intention*.

process structure diagram (processtructuurdiagram) [representation form of *process model*].

process view (procesaanblik) there are two possible views on a *world*: the *state view* and the process view. In the process view, one is concerned with the possible *transitions* in the world under consideration, and the causing *acts* in the corresponding system.

product (product) [*property* of *transaction*] *independent production fact*. A product is represented by a predication of an *entity*, e.g. the predication "is delivered" of the entity "sale 1618". A product has a number of *properties*, which are *dependent facts*.

product kind (productsoort) [*property* of *product*] *products* of the same product kind are brought about in *transactions* of the same *transaction kind*. Examples of product kinds: "Membership is started", "Rental is contracted", "the fee for Membership in Year is paid". The variables in a product kind are placeholders for *entities*.

production act (productiehandeling) the *act* in a *transaction* by which the *executor* creates the *product* as well as its *dependent production facts*. [see also *transaction pattern*]

production fact (productiefeit) a *state* element in the *production world*. A distinction is made between *independent* and *dependent* production facts. [see also *product*]

production time (productietijdstip) [*attribute* of *product*] the *point in time* at which the *product* of a *transaction* has started to exist or will start to exist.

production world (productiewereld) one of the two *worlds* in which the *actors* in an organisation cause *transitions*. A *state* of a production world is a set of *production facts*.

property (eigenschap) instance of a *property type*.

property type (eigenschaptype) a binary *fact type* that is a purely mathematical mapping from an *object class*, called the domain, to an object class, called the range.

proposition (propositie) term to refer to the *product* of a *transaction* during the *proposition phase*.

response link (responsverbinding) a link in a *Process Model* between a *coordination act* and a *coordination fact*, indicating that the *coordination act* is performed in response to the corresponding *coordination event*.

response part (responsdeel) the part of an *action rule* that contains the *act(s)* to be performed in response to the *agendum* that is dealt with.

response time (responsetijdstip) [*attribute* of *coordination fact*] the *point in time* at which the *performer* of the *coordination act* that creates the coordination fact wants the addressee to respond.

responsibility (verantwoordelijkheid) disposition of a *subject* to be committed to the *coordination acts* he or she has performed, as well as the *agenda* he or she has to respond to [see also *authority* and *competence*]

result (resultaat) term to refer to the *product* of a *transaction* during the *result phase*.

result phase (resultaatfase) the *transaction* phase in which the *initiator* and the *executor* strive to reach consensus about the *product* that the executor has brought about. It starts from the transaction status 'stated' and ends at the status 'accepted' or 'stopped'.

revocation pattern (herroepingspatroon) a pattern of *process steps* in which one of the steps in the *basic transaction pattern* is revoked.

self-activating actor (role) (zelfactiverende actor(rol)) an *actor* (*role*) is called self-activating if it is *initiator (role)* and *executor (role)* of the same *transaction (kind)*.

sentence (zin) the syntactic form in which thoughts, especially *facts*, are expressed.

service (dienst) the functional appearance of a *product* to its consumers.

Scope of Interest (interessegebied) the delineation of the (part of the) *organisation* that one has chosen to focus on.

social devotion (sociale toewijding) one of the three generic goals of *enterprise engineering*. *Enterprises* should be designed in such a way that its employees get all the opportunities and facilities they need

to develop and exert their *competences*. Social devotion implies that the employees are empowered with proper *authorities*.

social correspondence (sociale overeenstemming) the degree of mutual understanding of a *coordination act* at the *performa level* of *communication*. Social correspondence is reached if the *addressee* thinks he/she has understood the *intention* of the message correctly. Success or failure is conveyed by respectively a *confirmation* or a *disconfirmation*.

standard transaction pattern (standaard transactiepatroon) [see *transaction pattern*]

state (toestand) at any *point in time*, a *world* is in some state, defined as the set of *facts* that have been created up to (and including) the point in time.

state interpretation (toestandsinterpretatie) there are two possible interpretations of a *transaction kind* in the *construction model* of an organisation: the process interpretation and the *state interpretation*. In the state interpretation, a transaction kind is conceived as a *transaction bank*.

state view (toestandsaanblik) there are two possible views on a *world*: the state view and the *process view*. In the state view, one is concerned with the possible *states* of the world under consideration.

subject (subject) a human being in his or her quality of social individual, in particular its being able to enter into and comply with *commitments*. Only subjects can fulfil *actor roles*.

time scale (tijdschaal) a division of a (discrete and linear) time dimension in consecutive pieces of equal length (duration), called *time units*.

time unit (tijdeenheid) the unit for indicating or measuring pieces of *time* on a *time scale*. It may be indefinitely large or small, dependent on the needed precision. Examples: day, hour, minute, second [see also *point in time* and *date*].

transaction (transactie) the unit of *production* in an *organisation*. The result of a successfully carried out transaction is the coming into being of a *product*. [see also *transaction process*]

transaction bank (transactiebank) the conceptual store associated with a *transaction kind* in which all *coordination facts* in all of its carried out *transactions* are stored.

transaction kind (transactiesoort) [*property* of *transaction*] *transactions* of the same transaction kind regard *products* of the same *product kind*. Examples of transaction kinds: membership_start, rental_-contracting

transaction pattern (transactiepatroon) a pattern of *process steps*, alternately taken by two *actor roles*. The *basic transaction pattern* consists of the process steps 'request', 'promise', 'state', and 'accept' (Note: in between the promise and the state act, the *production act* is performed; it precedes immediately and unconditionally the state act, but it is not considered a process step itself). The *stan-*

dard transaction pattern contains in addition the process steps 'decline', 'quit', 'reject', and 'stop'. The *complete transaction pattern* consists of the standard pattern and the four *revocation patterns*.

transaction phase (transactiefase) the *standard transaction pattern* consists of three consecutive phases: the *proposition phase*, the *execution phase*, and the *result phase*.

transaction process (transactieproces) a sequence of *process steps*; it is some route through the *complete transaction pattern*, during which process steps of the same process step kind may occur multiple times. [see also *transaction pattern*]

transaction process diagram (transactieprocesdiagram) [representation form of the *transaction pattern*]

transaction product table (transactieproducttabel) table of *transaction kinds* and their corresponding *product kinds*.

transaction status (transactiestatus) most recently performed *process step* in a *transaction process*.

transition (transitie) a change of *state* of a *world*. A transition consists of the creation of an *independent fact*, usually together with a number of *dependent facts*.

validity claim (geldigheidsaanspraak) in performing a *coordination act*, three validity claims are raised by the *performer* and validated by the *addressee*: the *claim to truth*, the *claim to justice*, and the *claim to sincerity*. Which *act(s)* will be taken in response to a *coordination event* depends on the degree in which the three *validity claims* are accepted by the addressee. [see also *action rule*]

value (waarde) abstract *object*, notably a number or a boolean (true or false).

value scale (waardenschaal) an *object class* whose elements are *values*. A value scale has a dimension (like time, length, mass, temperature, or money) and a scale unit (like minute, meter, gram, degree Celsius, or euro).

waiting link (wachtverbinding) a link in a *process model* between a *coordination act* and a *coordination fact*, indicating that the performance of the *coordination act* has to wait until the *coordination fact* has been created.

work instructions (werkinstructies) next to *action rules*, the *action model* of an *aspect organisation* may contain work instructions, i.e. guidelines for performing *production acts*.

world (wereld) with every (discrete event) system, a world is associated in which the effects of the *acts* in the system take place. More specifically, the effect of an act in the system is the creation of a *fact* in the system's world.

ABBREVIATIONS

AM Action Model
ARS Action Rule Specification
BCT Bank Contents Table
CM Construction Model
DEMO Design and Engineering Methodology for Organisations
DFS Derived Fact Specification
FM Fact Model
OCD Organisation Construction Diagram
OFD Object Fact Diagram
PM Process Model
PSD Process Structure Diagram
TPT Transaction Product Table
TPD Transaction Process Diagram
WIS Work Instruction Specification

NOTE: the formal definitions of the representation forms of the models (diagrams, tables, etc.) are contained in the document "DEMO Models & Representations". This document can be downloaded freely from www.ee-institute.org/publications.

ABOUT THE AUTHOR

Alicia P.C. Perinforma is a pseudonym of Jan L.G. Dietz. He uses it as an expression of gratitude and homage to the many students who have been valuable whetstones to him when he was developing DEMO and its underlying theories. Jan Dietz is emeritus professor in information systems design at Delft University of Technology, and professor in enterprise engineering at Antwerp Management School. After graduating in electrical engineering (in 1970) at Eindhoven University of Technology, he has worked in business automation for ten years. Also during his academic career, from 1980 on, he has kept close contact with practice. He holds a doctoral degree in computer science and is author of over 250 scientific and practice-oriented articles and books.

Jan Dietz is the spiritual father of DEMO (Design & Engineering Methodology for Organisations), the main methodology in enterprise engineering. He is co-founder and honorary chairman of the Enterprise Engineering Institute (www.ee-institute.org) and he has participated in numerous practical DEMO projects regarding (re)designing and (re)engineering organisations. For advancing the discipline of enterprise engineering, he chairs the international CIAO! Network (www.ciaonetwork.org). He is also editor-in-chief of a book series on enterprise engineering, published by Springer Verlag, Germany.

The teaching courses about DEMO and Enterprise Engineering that he delivers comprise both academic and commercial courses, in all kinds of educational institutes and enterprises. In addition, he has given numerous tutorials and keynote lectures at international conferences.

Jan Dietz is also director of Sapio Enterprise Engineering (www.sapio.nl), through which he performs his business activities (courses, consultancy, and expert judgments). Sapio is member of the Enterprise Engineering Consortium (www.ee-consortium.com).

Jan Dietz is present in the wikipedia (http://en.wikipedia.org/wiki/Jan_Dietz). You can contact him at these e-mail addresses: j.l.g.dietz@tudelft.nl, jan.dietz@ua.ac.be, jan.dietz@sapio.nl

ABOUT THE BOOK

This book is one of the two books that constitute the DEMO basis. The other one is titled "Red garden gnomes don't exist", authored by Jan L.G. Dietz. Both books can be obtained from www.sapio.nl. The DEMO basis elucidates the role of ICT (Information and Communication Technology) in organisations and offers an easy introduction in the emerging discipline of enterprise engineering. It is meant for everyone who feels the need for a coherent, consistent, comprehensive, and yet concise understanding of organisation and ICT. The DEMO basis aims in particular at professionals in organisation and ICT, like enterprise architects, business process managers, and information system designers. It consitutes the course material for the DEMO Bachelor course, which is taught at several universities and polytechnics, all over the world.

DEMO (Design and Engineering Methodology for Organisations) is the leading methodology in the discipline of enterprise engineering, whose founding article has been published in 2013. Three generic goals are identified. The first one, *intellectual manageability*, is the basis for mastering complexity; current approaches fall short in assisting professionals to master the complexity of enterprises and enterprise changes. The second goal, *organizational concinnity*, is conditional for making strategic initiatives operational; current approaches do not, or inadequately, address this objective. The third goal, *social devotion*, is the basis for achieving employee empowerment as well as knowledgeable management and governance; modern employees are highly educated knowledge workers; yet, the mindset of managers has not evolved accordingly. Enterprise engineering is developed by a growing international network of universities and research institutes, in close connection with practice. This so-called CIAO! network currently comprises universities and research institutes from Belgium, Brazil, Czech Republic, Germany, Japan, Luxembourg, Netherlands, Portugal, Russia, Switzerland, and USA.