Understanding and Modeling Business Processes with DEMO

Jan L.G. Dietz
Delft University of Technology, Department ISSE, P.O. Box 356
NL-2600 AJ Delft
j.l.g.dietz@its.tudelft.nl

Abstract. DEMO is a methodology for modeling, (re)designing and (re)engineering organizations. Its theoretical basis draws on three scientific sources of inspiration: Habermas' Communicative Action Theory, Stamper's Semiotic Ladder, and Bunge's Ontology. The core notion is the OER-transaction, which is a recurrent pattern of communication and action. This notion serves as the universal (atomic) building block of business processes. Some fifty projects have been carried out with DEMO, of very different kinds and in various organizations. The succes factor has been the same for all these projects, namely the practical relevance of the concepts of DEMO, as well as their clear and precise definitions. The application of the methodology is illustrated taking two well known cases.

1 Introduction

DEMO (Dynamic Essential Modeling of Organizations) incorporates a way of thinking about organization and technology that has originated from a deep dissatisfaction with current ways of thinking about information systems and business processes. These current ways of thinking fail to explain coherently and precisely how organization and ICT (Information and Communication Technology) are interrelated. They fail to provide assistance in articulating what is essential and invariant about the business processes and what are more or less incidental ways of doing. This is however what seems to be needed: separating 'essence' from 'technology'.

DEMO fits in a fairly new and promising perspective on business processes and information systems, called the Language/Action Perspective, or L/A Perspective for short. The theoretical foundation of this new perspective is constituted by Speech Acts Theory [1], [2], and the Theory of Communicative Action [3]. The pioneer of the L/A Perspective is undoubtedly Fernando Flores [4]. The L/A Perspective assumes that communication is a kind of action in that it creates commitments between the communicating parties. To communicate then is to perform language acts [2] or communicative acts [3], like requesting or promising. Other approaches in the same L/A Perspective can be found in e.g. [5], [6] and [7]. Three international workshops have been held up to now focussing on the L/A Perspective [8], [9], [10]. These proceedings contain several papers concerning DEMO. Some other relevant papers are [11], [12], and [13].

The major difference between DEMO and other L/A aproaches, like SAMPO [6], BAT [14], and Action WorkFlow [7], is that it is built on two additional theoretical pillars next to the Comunicatieve Action Paradigm, namely Stamper's Semiotic Ladder [15] and Bunge's Ontology [16], [17]. The outline of the paper is as follows. Sections 2 through 5 deal with the theory of DEMO. In section 2 the concept of communication is discussed. This constitutes the ground on which the three other core concepts, information, action and organization are founded. They are discussed in sections 3, 4 and 5 respectively. Sections 6 shows how organizations are modeled with the DEMO methodology, and section 7 contains some conclusions.

2 Communication

In line with the overall goal of DEMO to separate essence from technology, we define *communication* as the sharing of thoughts between social individuals or *subjects.* By thought we mean not only pieces of knowledge but also whishes, promises, feelings etc. The unit of communication consists of the sharing of one thought between two subjects, and is called the *communicative action.* The subject that is going to share one of his/her thoughts is called the locutor (L) of the action, and the subject with whom the thought is shared is called the addressee (A). The thought to be shared is formulated in some language that is common to L and A. The produced language expression is called (an elementary piece of) *information*. This information must somehow be made perceivable to A such that it can be interpreted. The effect of interpretation is the creation of a thought of A (of which both L and A usually hope that it is very similar to the thought of L).

Performing a communicative action successfully is not trivial or unproblematic. At least two requirements have to be met. The first one is that there exists a communication channel, i.e. a means by which the language utterances produced by L are transmitted to A. This requirement includes the verification of the identity of the participants. The second requirement is that A understands what L means. Otherwise said, the participants have to verify that the semantic interpretation of the information by L is correct. Assuming that these requirements are fulfilled, we will now focus on the social or intersubjective effect that the performance of a communicative action brings about. As the L/A Perspective tells us, a communicative action consists of an illocutionary action and a propositional action. In the propositional action, a fact in some world is referred to, as well as a time. In the illocutionary action the locutor expresses his/her ‘attitude’ towards the proposition. In DEMO, the next so-called OER-notation[1] is used to denote communicative actions:

<locutor> : <illocution> : <addressee> : <fact> : <time>

As an example of a communicative action, let us assume that someone, a hotel guest (G for short), addresses a reception employee (E) and utters the next sentence:

“Do you have suites?”

The illocution of the formulated thought is the question, the fact is ‘the hotel does have suites’, and the time is a not explicitly specified (default) period, most probably ‘currently’. The OER-notation of the example communicative action above is:

G : question : E : *the hotel does have suites* : currently

[1] The word “OER” is a Dutch word, meaning “primal” , “original”. It expresses that one seeks for the essence by abstracting from (current) realization.

The reply by the hotel employee to this question could be:

"Yes, we do."

The OER-notation of this communicative action is:

E : assertion : G : *the hotel does have suites* : currently

DEMO distinguishes six illocutionary kinds: question, assertion, request, promise, statement and acceptance.

A sequence of to and fro communicative actions between two subjects is called a *conversation*. DEMO distinguishes between informative and performative conversations. *Informative conversations* are conversations in which only questions and assertions occur. An example of an informative conversation between G and E is the combination of the two utterances, mentioned earlier:

G : Do you have suites?
E : Yes, we do

Performative conversations are conversations in which only requests, promises, statements and acceptances occur. Two subtypes are distinguished: actagenic and factagenic. An actagenic conversation is a conversation in which the request and the promise are the chief illocutionary kinds. An example in the hotel situation is:

G : I'd like to have a suite for 3 nights starting January the 3rd
E : Let me see ... yes, I will arrange that for you

The OER-notation of this conversation (in which 'asap' means 'as soon as possible') is:

G : request : E : *a suite is reserved for G from January 3 till January 6* : asap
E : promise : G : *a suite is reserved for G from January 3 till January 6* : asap

The result of this actagenic conversation is that E has committed him/her-self to make the agreed upon reservation. In a factagenic conversation, the two actors agree on the achieved result. An example of a corresponding factagenic conversation is:

E : Madame, I have reserved a suite for you for 3 nights starting January the 3rd
G : Thank you very much

The OER-notation of this conversation is:

E : statement : G : *a suite is reserved for G from January 3 till January 6* : asap
G : acceptance : E : *a suite is reserved for G from January 3 till January 6* : asap

The example conversations show that the real meaning of a sentence can often not be deduced from a grammatical analysis of the sentence, because it depends heavily on the context (of other sentences) in which it is uttered. Furthermore, the examples show clearly that every sentence possesses the components illocution and proposition (consisting of fact and time).

3 Information

As was already mentioned in the previous section, *information* is understood to be form given thought. Thus, the notion of information is closely related to the notion of communication. To be more specific: there is no information without communication, since information is produced only for the purpose of communicating. Information serves to bridge distances in both (physical) space and time. It allows therefore also communicating with 'oneself' at different points in time. There are three aspects of information to be distinguished, called the forma, the in-forma and the per-forma. These aspects are slight modifications of the semiotic distinctions as proposed in [15].

Every piece of information has a *forma*, meaning that it has some perceivable structure carried in some physical substance. This structure must be recognizable as being an expression in some 'language', which is the case if the forma con-forms to the syntactical rules of that language. The notion of language has to be taken broadly. Every phenomenon that counts as a forma according to the institutional rules of a society, is by definition a forma (cf. [18]). To every forma belongs at least one in-forma (if there are more in-forma's, these are called homonyms). The *in-forma* is the meaning of the forma, the reference to some Universe of Discourse, as defined by the semantics of the language. The aspect in-forma also includes the pragmatic rules of the language, like the choice of the right or best forma to express some in-forma in specific circumstances.

The *per-forma* of a piece of information is the effect on the relationship between the communicating subjects, caused by communicating the thought. It is determined by both the illocution and the proposition of the communicative action, and is further dependent on the current norms and values in the shared culture of the communicating subjects. For example, if one asks someone the time, one is supposed to need the answer (it is not considered just to ask such a question for fun), and the one who answers is expected to answer to the best of his knowledge. However, this 'norm' may be overwritten, e.g. if the one has just robbed the other one. In such a situation, it is e.g. acceptable to be dishonest.

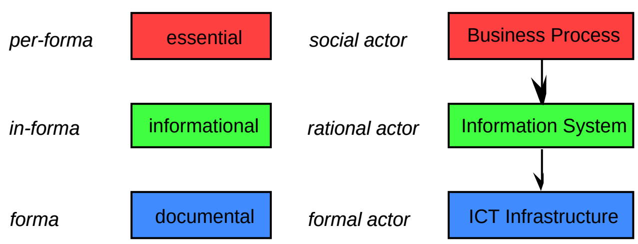

Figure 1 Information aspects, levels, actor roles, and system categories

The distinction between the three aspects per-forma, in-forma, and forma, gives rise to the distinction of three corresponding levels at which information in an organization has to be managed: the essential level, the informational level, and the documental level respectively (cf. figure 1). Likewise, these levels can be understood as 'glasses' through which one can look at an organization. Looking through the *essential* glasses, one observes social actors, i.e. elements that execute per-formative actions like requesting or promising objective actions, and like stating or accepting results of objective actions (Note. The notion of objective action will be explained in section 4). Looking through the *informational* glasses, one observes rational actors, i.e. elements that execute in-formative actions like collecting, providing, recalling and computing knowledge about per-formative actions and their results. Lastly, looking through the *documental* glasses, one observes formal actors, i.e. elements that execute formative actions like gathering, distributing, storing, copying, and destroying documents containing the aforementioned knowledge about per-formative actions and their results.

Put in other terms, one observes through the three distinct glasses, systems of three distinct categories, viz. business processes, information systems an ICT-infrastructure components (Note. We conform to the habit to speak of business processes instead of business systems). These three system categories and their relationships are also exhibited in figure 1.

The arrows from Business Process to Information System, and from the latter to ICT Infrastructure indicate one-to-many relationships, which should be understood as follows. Given a particular business process, one may conceive of a number of (collections of) information system(s) that support, and in doing so make operational, that business process. However, at any point of time, there is only one (collection of) information system(s). Likewise, given a particular information system, one can conceive of a number of ICT infrastructure configurations that are able to realize, and in doing so make operational, that information system.

4 Action

Action is a core notion in studying any dynamic system from an engineering point of view. According to the distinctions as proposed in the previous section, DEMO distinguishes between essential actions, informational actions and documental actions. The main interest however is in the essential actions, because these are the ‘authentic, genuine’ business actions, all other actions only serve to support them. The class of essential actions is further divided into objective actions and intersubjective actions.

By executing *objective actions*, the members of an organization fulfill the mission of the organization. The nature of an objective action can be material or immaterial. Examples of material actions are all manufacturing actions in the production of goods as well as all storage and transportation actions. Examples of immaterial actions are the judgement by a court to condemn someone, the decision to grant an insurance claim, and appointing someone president. By executing *intersubjective actions*, subjects enter into and comply with commitments. In doing so, they initiate and coordinate the execution of objective actions. Intersubjective actions are defined to be the performative communicative actions, as presented in section 2, thus requests, promises, statements and acceptances. In order to abstract from the particular subject that performs an action and to concentrate on the functional or organizational role of the subject in performing that action, the notion of *actor* is introduced (e.g. sales and purchasing). Thus, an actor is a particular ‘amount’ of authority and responsibility; it requires from the fulfilling subject a particular ‘amount’ of competence. An actor role can be fulfilled by a number of subjects (concurrently as well as collectively), and a subject may fulfill concurrently a number of actor roles.

In correspondence with the distinction between objective and intersubjective actions, DEMO distinguishes between two worlds in which each of these kinds of actions have effect: the *object world* and the *intersubject world* respectively. The effect of every action is a state transition. These transitions are considered to take place instantaneously. A particular transition (e.g. requesting a beer by someone from the barkeeper) at a particular point in time (e.g. the requesting of a beer by a particular person at a particular moment) is called an event. A time series of events is a process.

Objective actions and their related intersubjective actions appear to occur in a particular pattern, called the *OER-transaction*, as illustrated by figure 2. It consists of three phases: the order phase or O-phase, the execution phase or E-phase, and the result phase or R-phase (Note. The three letters O, E and R constitute the Dutch word “OER” which was explained earlier). A transaction is carried through by two actors, who alternately perform actions. The one who starts the transaction and eventually completes it, is called the *initiator* (I in figure 2), the other one, who actually performs the objective action, is called the *executor* (E in figure 2).

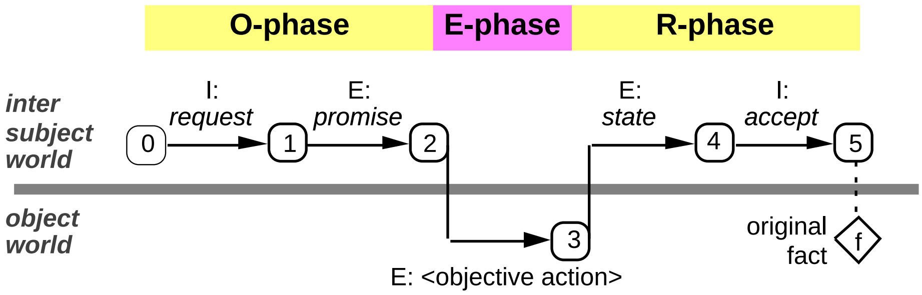

Figure 2 The OER-transaction

The order phase is an actagenic conversation, and the result phase is a factagenic conversation. Both conversations consist of intersubjective actions, having as effect a transition in the intersubject world. These actions are executed alternately by the initiator and the executor of the transaction. In between the two conversations, the objective action is executed, by the executor of the transaction. The effect of this action is a transition in the object world (Note. What figure 2 shows is in fact only the success layer of a transaction process. Next to this layer, a negotiation or discussion layer and a discourse layer are to be distinguished [19]).

Principally, events in the object world are not knowable to the initiator (and to other actors) as long as they are not stated by the executor. This principal position is important. On the one hand, it stresses the supremacy of events in the intersubject world. On the other hand it allows material and immaterial objective actions (and resulting facts) to be dealt with in the same manner. For immaterial facts it is obvious that they cannot be said to exist unless they are stated and subsequently accepted, and thus that they come into existence at the moment of acceptance, i.e. when reaching transaction status 5. Although at first sight, and intuitively, material facts seem to come into existence in status 3, this appears not to be the case on closer observation. In every organization with material objective actions (like manufacturing or transporting firms), there appears always to be someone who is held responsible for a particular material fact being the case. Only he or she has the authority to declare that something is the case, which causes an event in the intersubject world.

The transaction concept as presented serves as the building block of business processes. Otherwise said, every business process is a (molecular) structure of (atomic) transactions. This structure may be arbitrarily complex.

5 Organization

There exists a variety of definitions for the term 'organization'. These definitions represent as many different notions, apparently depending on the point of view on takes towards the phenomenon organization. As indicated in the introduction, DEMO takes an engineering point of view. This is rather new among IS professionals, whose notions of organizations are dominated by the black-box model. Taking the black-box model when studying organizations (or any variant of it like the control model), one tries to find and understand the relevant input variables and output variables, and the relationship between them, called the transfer function. Knowing the transfer function means knowing how the system responds to variations in the values of the input variables. By manipulating the input variables, one is able to change the behavior of the system. If the transfer function is too complicated to understand, the technique of *functional decomposition* can be applied through which the system is replaced by a structure of subsystems of which the transfer functions are more readily understandable.

Like every other system (e.g. an alarm clock or a racing car), the functional behavior of an organization is brought about by the collective working of the constructional components. The construction and the operation of a system are most near to what a system really is, to its ontological description. A very exact and very general, ontological definition of a system is provided by Bunge [17]. Based on this definition, DEMO uses the next definition of an organization:

Something is an *organization* if and only if it fulfills the next properties:

- It has *composition*, i.e. it is composed of actors. These actors act on the basis of assigned authority and with corresponding responsibility.
- It has *effect*, i.e. the actors bring about changes in the object world.
- It has *structure*, i.e. the actors influence each other. Two kinds of mutual influencing are distinguished. *Interaction* consists of executing transactions. *Interstriction* consists of taking into account the results or the status of other transactions when carrying through a transaction.
- The composition is divided into two subsets, called the kernel and the environment, such that every actor in the environment influences, either through interaction or through interstriction, one or more actors in the kernel (and such that there are no 'isolated' parts in the kernel). The closed line that separates the kernel from the environment is called the *boundary.*

Based on this definition, a *white-box model* of a system can be made, as exhibited in figure 3. On the one hand, actors perform objective actions, thus changing the state of the object world. On the other hand, they perform intersubjective actions, thus changing the state of the intersubjective world. When being active the actors take into account (are restricted by) the state of the object world as well as the state of the intersubject world.

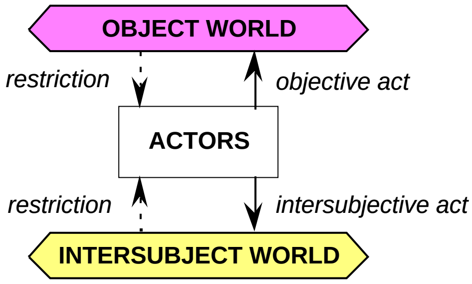

Figure 3 The white-box model

Analogous to the technique of functional (de)composition, there is a technique for composing and decomposing white-box models of a system. It is called *constructional (de)composition*. A central concept in the technique is the concept of subsystem. This concept is in DEMO defined as follows. S2 is a subsystem of S1 if and only if S1 and S2 are systems according to the definition above, the kernel of S2 is a subset of the kernel of S1, the structure of S2 is a subset of the structure of S1, and the environment of S2 is a subset of the composition of S1.

In conclusion, functional (de)composition and constructional (de)composition are similar techniques, but applied to very different system notions. An important consequence is that a functional component does not necessarily have a constructional counterpart. Attempting to relate the one kind of component to the other kind just does not make sense; their natures are inherently incompatible. Therefore, functional decomposition doesn't reveal a bit of the construction and operation of the system.

6 The DEMO methodology

The DEMO methodology was originally intended for carrying out the requirements engineering phase of information systems development, and it is being applied very successfully for that purpose indeed. Next to that however, it appears to be very effective for a variety of other purposes, like Business Process Reengineering, WorkFlow Management, Electronic Commerce and Virtual Organizations. In short, the methodology has proven to be a coherent, integral, and comprehensive approach to the modeling, the (re)design and (re)engineering, as well as the managing, of business processes and information systems. For several years now it is being applied in practice by a growing number of consultancy enterprises in all kinds of organizations: commercial, industrial and governmental.

The main contribution of the methodology in an organization's change process, is the provision of the essential model of the organization. This is a description by means of several diagramming techniques of its business processes (as defined by the DEMO theory), thereby abstracting fully from the informational/documental as well as from the organizational (personnel) realization. In each of the about fifty projects undertaken up to now, it has turned out that this essential model provides an ideal starting point for rethinking firstly the business of the organization as such, and secondly the way in which it is realized now. Invariably, employees and managers are taken by the clarity, the conciseness, the relevance, and the effectiveness of the methodology and the produced results. Not only do DEMO projects cost a fraction of the costs of a ‘traditional’ project, it also succeeds where the ‘traditional’ methodologies keep failing. An extensive discussion of one of the projects is provided in [20].

22 The Ford Case

For the first illustration of the DEMO methodology, we take the well-known Ford Case [21]. Hammer provides the next description of it:

'When Ford's purchasing department wrote a purchase order, it sent a copy to accounts payable. Later, when material control received the goods, it sent a copy of the receiving document to accounts payable. Meanwhile, the vendor sent an invoice to accounts payable. It was up to accounts payable, then, to match the purchase order against the receiving document and the invoice. If they matched, the department issued payment.

The department spent most of its time on mismatches, instances where the purchase order, receiving document, and invoice disagreed...

One way to improve things might have been to help the accounts payable clerk investigate more efficiently, but a better choice was to prevent the mismatch in the first place. To this end, Ford instituted 'invoiceless processing'. Now when the purchasing department initiates an order, it enters the information into an on-line database. It doesn't send a copy of the purchase order to anyone. When the goods arrive at the receiving dock, the receiving clerk checks the database to see if they correspond to an outstanding purchase order. If so, he or she accepts them and enters the transaction into the computer system. (If receiving can't find a database entry for the received goods, it simply returns the order.)'

According to Hammer, Ford opted by the chosen solution for radical change, and achieved dramatic improvement. To illustrate this, it was mentioned that initially there were 500 people working at the accounts payable department, and that a 75% reduction of this figure was achieved after the solution had been implemented. Let us analyze this case briefly. To start with, figure 4 exhibits (part of) the essential model of the business activities concerned.

Transaction type	**Fact type**
T1: order_delivery	F1: *order O is delivered*
T2: order_payment	F2: *order O is paid*

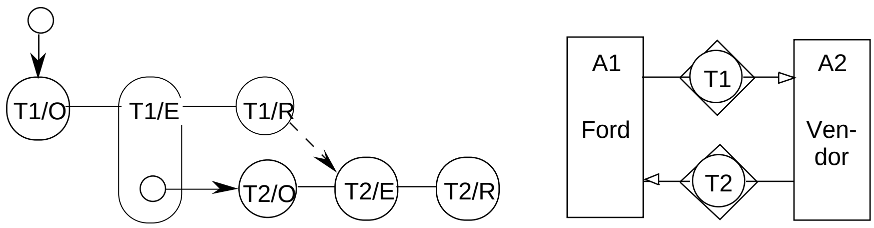

Figure 4 The essential model of the Ford Case

On top of figure 4, a so-called transaction table is shown. It lists the distinct transaction types, including the specification of the resulting fact types. In the Ford Case, there are two transaction types: order delivery, and order payment. In the lower right corner of the figure, the Interaction Model is shown. A box represents an actor. The symbol for a transaction type is a disk with a diamond ‘behind it’. The disk represents the intersubject world part and the diamond the object world part. Therefore, the disk may also be interpreted as a communication bank, i.e. the conceptual store of transaction states, and the diamond as a fact bank, i.e. the conceptual store of transaction results and associated facts. A solid line connects the transaction symbol to the initiator of the transaction type, and the solid line with open arrowhead connects it to its executor. For instance, actor A1 is initiator of transaction type T2 and executor of transaction type T1. The lower left corner contains the Process Model. In this diagram, each of the two transaction types is divided into its three constituent phases, the O-phase, the E-phase, and the R-phase. The solid arrows represent initiation relationships. Their meaning is that the process or activity at the point side is started from the process or activity at the shaft side. For every transaction, it holds that the E-phase is preceded by the O-phase, and that the R-phase is preceded by the E-phase. This is shown in figure 4. Also shown is that the O-phase of transaction type T2 is initiated from the E-phase of transaction type T1, and that the O-phase of transaction type T1 is initiated externally (this is indicated by the small circle). The dotted arrow from T1/R to T2/E represents a conditional relationship. It means that T2/E can only be completed after T1/R has been completed. The Process Model in figure 4 expresses that the request for payment is issued in parallel with the statement that the order is delivered, and that the payment is actually performed after the order delivery has been accepted.

Next to the Interaction Model and the Process Model, there are three other model types: the Interstriction Model (describing the inspection relationships between actors and information banks), the Action Model (describing in detail the business rules for carrying through the transactions of the distinct types) and the Fact Model (describing the state space of the object world and the intersubject world). Collectively, these partial models constitute what is called the *essential model* of an organization.

The interesting point in the Ford case is, that the essential model is the same before and after the radical (!) change. In the terminology of DEMO, it means that there has not been a redesign (i.e. a change at the essential level). Redesign would imply that (part of) the essential model, e.g. the Process Model, would have been changed. There has only been a reengineering, i.e. changes are made only at the informational and/or the documental level. In this respect, it sounds misleading to say that Ford has become invoiceless. This is only true at the documental level: there are just no paper invoices sent anymore. The essential meaning of an invoice however is that it is a request for payment, and this request is (of course) still performed, however by different means. The difference between the new situation and the old one is that in the new situation the delivery of goods counts as a request for payment, while in the old situation the paper invoice conveyed this request. All additional information that was contained on the paper invoice in the old situation, like the delivered items, the quantities, and the prices, must now be found elsewhere. In this respect, Ford did make an efficient agreement with its vendors, including that only completely delivered orders would be accepted and consequently paid; the order information was still available at Ford. One could argue that this agreement does affect slightly the essential level (there is a new business rule, stating that only completely delivered orders are accepted). Still, we would not call that a radical change.

6.2 The Elevator Control Case

Since DEMO focuses on the essential level of abstraction, the way in which the essence of a system is realized is hidden. This property offers the possibility to model ‘technical’ systems as organizations, thus as if they were ‘realized’ by human actors. Doing this not only makes it possible to reveal the essence of ‘technical’ systems; it also shows immediately that notions like authority and responsibility apply to these systems as well. On closer view, many ‘technical’ systems are ‘social’ systems; the difference lies in the ‘technology’ used to realize them.

The example taken is well known in the software engineering community, it was an almost classical example in the literature and on conferences in the eighties. A description as well as an analysis can be found in [22]. The description is formulated as the requirements for the design of a program that schedules and controls a set of elevators in a building. Because of the limited space, we do not repeat the complete description here, but confine ourselves to a summary:

The system consists of 4 elevators in a building with 40 floors. Next to the elevator doors at each floor, there is a summons button to go upward and a summons button to go downwards. In the interior of each elevator, there is a panel with 40 destination buttons, for every floor one. Above the door there is also a panel on which is shown at which floor the elevator is arriving (or is stopping). The software program receives summons and destination requests, schedules them, and controls the motors for moving the elevators upwards and downwards and for stopping them at a floor. This should be done in an efficient manner. The program also receives signals from the floor sensors in each elevator shaft, indicating that the elevator approaches the floor. After reception of a summons request, the program sends a signal to illuminate the button. As soon as an elevator visits the floor in the right direction, the light is switched off. A similar procedure holds for the destination buttons in each elevator. The program also sends signals to appropriately illuminate, in the interior of every elevator, the floor, at which it arrives or stops. Lastly, an elevator may be out of service, and a floor may be unreachable, for whatever reason.

A careful investigation at the essential level leads to the identification of four transaction types (cf. figure 5).

transaction type	**transaction result**
T1 summons_visit	F1 *an elevator visits floor F, with direction D*
T2 destination_visit	F2 *elevator E visits floor F*
T3 elevator_movement	F3 *elevator E moves in direction M*
T4 floor_visit	F4 *elevator E visits floor F, with direction D*

Figure 5 Transaction-result table of the elevator system

By way of illustration, we will explain which (seemingly non-communicative) actions do count as performing the request, promise, statement and acceptance of transaction T1 (cf. figure 2). Pressing a summons button counts as issuing a request for a T1. Having promised a T1, thus having reached state 2 in figure 2, is represented by the illumination of the button. The opening of the doors and the entering by the passenger of the car can be viewed as the statement and the acceptance respectively, but one might as well say that these actions are implicit. Such implicit actions are as important as the explicit ones, i.e. they should be identified as carefully as the other ones. In case of a breakdown of a transaction, there must be no doubt about which actor is failing to comply with its role, its commitments. Otherwise said, it must always be clear who is responsible.

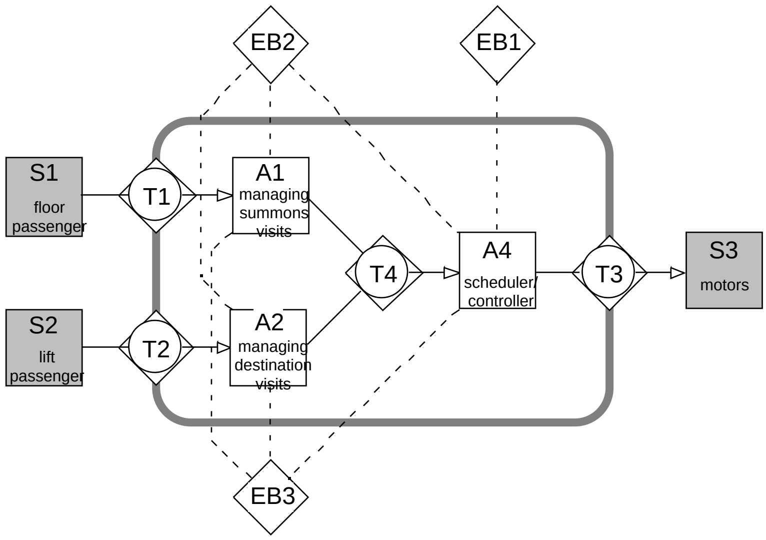

Figure 6 Interaction and Interstriction Model of the elevator system

Based on the table of figure 5, the diagram in figure 6 can be drawn. This diagram exhibits both the Interaction Model and the Interstriction Model: it shows the transaction types with their initiator(s) and executor, as well as the informative conversations between actors via information banks. The gray 'roundangle' represents the system boundary. Since one usually does not have sufficient knowledge about the actors in the environment (or does not care), we draw so-called complex actors or system kernels in the environment: S1, S2 and S3 (In fact, what is inside the boundary, is a revelation of the system kernel S0). Both actor A1 (the executor of T1) and A2 (the executor of T2) are initiator of the internal transaction type T4.

The dotted lines represent inspection links between actors and information banks. The three external banks EB1, EB2 and EB3 contain the elevator position information (from the sensors), the available elevators and the reachable floors respectively. For the analysis of the system, we don't care who is providing this information, even less how it is stored or transmitted. It suffices to know, and to indicate in the models, that a particular kind of information is needed, and that it is available for specific actors.

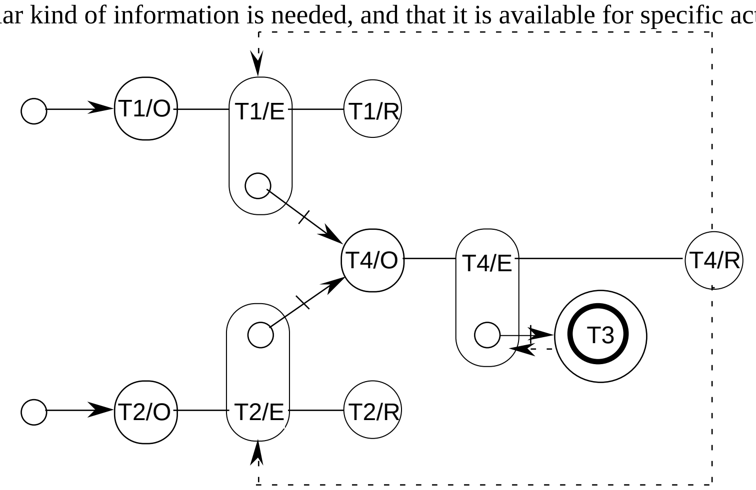

Figure 7 Process Diagram of the elevator system

Lastly, we present in figure 7 the (business) process diagram. It exhibits that T1 and T2 are initiated externally (thus by an actor in the environment), and that a T4 is iniated either during the execution of a T1 or during the execution of a T2. The bars on the initiation links indicate optionality. It means e.g. that not in every instance of T2/E a request for executing a T4 is issued. Practically spoken, this covers the situation that two lift passengers want to go to the same floor. Figure 7 also shows that during the execution of a T4 a set of transactions of type T3 can be initiated. This way of modeling reflects that sometimes there may be no need to issue a motor control command, and sometimes it may be necessary to issue several commands (e.g. to move upwards first and then to stop). Furthermore, the diagram exhibits that the successful completion of a T4 is a condition for the completion of the corresponding T1 and/or T2.

8 Conclusions

The essence of an organization lies in the entering into and the complying with commitments by authorized and responsible subjects. This constitutes the working principle of any organization. The DEMO transaction is the elementary building block of every business process, irrespective of the nature of the business, i.e. of the kind of the objective actions (material or immaterial). At the same time it becomes clear that a business process differs fundamentally from a production or a logistic process, and that so-called information intensive organizations (banks, insurance companies etc.) do have business processes like all other organizations, they only don’t have production and logistic processes (at least at the essential level, they do have them at the documental level, like every organization has).

Modeling business processes is a prerequisite for (re)designing and (re)engineering them. Current approaches to modeling business processes however do not embody an appropriate understanding of the notion of business process, and consequently do not provide an effective help. The presented DEMO methodology does offer the appropriate, engineering oriented, understanding. It has been illustrated that the methodology can be an effective help in various situations concerning the analysis and optimization of business processes. Because one abstracts completely from the way in which the essential model of an organization is realized, one has the right design freedom for thinking anew about that realization. Otherwise said, one is in the right position to think of applying modern information and communication technology in an optimal way.

References

1 Austin,J.L., *How to do things with words*, Harvard University Press, Cambridge MA, 1962

2 Searle, J.R., *Speech Acts, an Essay in the Philosophy of Language*, Cambridge University Press, Cambridge MA, 1969

3 Habermas, J., *Theorie des Kommunikatives Handelns*, Erster Band, Suhrkamp Verlag, Frankfurt am Main, 1981

4 Flores, F.M., J.J. Ludlow, Doing and Speaking in the Office, in: *Decision Support Systems: Issues and Challenges*, Pergamon, Oxford, 1980

5 Taylor, J.R., *Rethinking the Theory of Organizational Communication: how to read an organization*, Ablex, Norwood, N.J., 1993

6 Auramäki, E. E. Lehtinen, K. Lyytinen, A Speech Act based Office Modeling Approach, *ACM TOIS*, vol. 6, no. 2, 1988

7 Medina-Mora, R., T. Winograd, R. Flores, F. Flores, The Action Workflow Approach to Workflow Management Technology, *Proc. 4th Int. Conf. on CSCW*, ACM, New York, 1992

8 Dignum, F., J.L.G. Dietz, E. Verharen, H. Weigand, *Communication Modeling - the Language/Action Perspective*, Electronic Workshops in Computing, Springer, 1996, http://www.springer.co.nk/ewic/workshops/CM96/

9 Dignum, F., J.L.G. Dietz, *Proceedings of the Second International Workshop on Communication Modeling*, Computing Science Reports, Eindhoven University of Technology, 1997

10 Goldkuhl, G., M. Lind, U. Seigerroth, *Proceedings of the Third International Workshop on Communication Modeling*, Department of Informatics, Jönköping International Business School, 1998

11 Dietz, J.L.G., Modeling Business Processes for the Purpose of Redesign, in: *Proc. IFIP TC8 Open Conference on BPR*, North-Holland, Amsterdam, 1994

12 Van der Rijst, N.B.J., V.E. van Reijswoud, Comparing Speech Act Based Modeling Approaches for the Purpose of Information Systems Development, in: *Proc. of the Third European Conference on Information Systems*, Athens, Greece, 1995

13 Dietz, J.L.G., J.B.F. Mulder, Realizing Strategic Reengineering Objectives with DEMO, in: *Proc. International Symposium on Business Process Modeling*, Springer-Verlag 1996

14 Lind, M., G. Goldkuhl, Reconstruction of Different Business Prosesses: A Theory and Method Driven Analysis, in: [Dignum, Dietz, 1997]

15 Stamper, R.K., Applied Semiotics, in: *Proc. of the ICL/University of New Castle Seminar 'Information'*, New castle, 1993

16 Bunge, M.A., *Treatise on Basic Philosophy*, vol.3, D. Reidel Publishing Company, Dordrecht, The Netherlands, 1977

17 Bunge, M.A., *Treatise on Basic Philosophy*, vol.4, D. Reidel Publishing Company, Dordrecht, The Netherlands, 1979

18 Searle, J.R., *The Construction of Social Reality*,
Allen Lane, The Penguin Press, London, 1995
19 Van Reijswoud, V.E., *The Structure of Business Communication - Theory, Model and Application*, Doctoral Thesis, Delft University of Technology, 199611
20 Van Reijswoud, V.E., J.B.F. Mulder, J.L.G. Dietz, Speech Act Based Business Process and Information Modeling with DEMO, *Information Systems Journal*, 1999
21 Hammer, M., Reengineering Work: Don't Automate, Obliterate',
Harvard Business Review, 1990
22 Yourdon, E., *Modern Structured Analysis*, Prentice-Hall, Inc., 1989

For more information the reader is referred to the website www.demo.tudelft.nl