By Jan L.G. Dietz

The Deep Structure of Business Processes

Delving beneath organizational surface structure to reveal the essential structure of business processes.

Whether you are an information system architect or designer, or a business process analyst or manager, the success of your work heavily depends on the degree of your understanding of organizations. Unfortunately, the kind of understanding you need in your profession cannot be drawn from the rich sources of the organizational and management sciences because of their predominant functional orientation. Functional knowledge is sufficient (and necessary) if you only need to use or control an organization, like you operate a car you are driving. But your task is to change the business processes or to improve their efficiency by applying information and communication technology. To do so, you need knowledge of the construction and the operation of organizations, like a car mechanic needs to know how a car is constructed and how it works. Moreover, the complexity of the problems you are faced with and

Illustration by Robert Neubecker

The fundamental question is how to reveal these structures, HOW TO EXTRACT THEM FROM THE OBSERVABLE SURFACE STRUCTURES that tend to obscure them.

the demands for timely delivery of high-quality solutions means that drawing unrelated and informal diagrams as a basis for redesign and reengineering will not be sufficient. Performing your work in a professional manner requires a solid scientific methodology. In this article, I will describe how the language-action perspective (LAP)-based DEMO[1] methodology reveals the essential structure of business processes, which is a theoretically sound and practically useful notion of deep structure. Business processes need not be mind-boggling railroad yards; they can be crystal structures of 'atoms' and 'molecules' [2, 3]. The fundamental question is how to reveal these structures, how to extract them from the observable surface structures that tend to obscure them.

Understanding the Construction and Operation of Organizations

The first thing to be recognized is that organizations are designed and engineered artifacts, like automobiles and information systems. The distinctive property of organizations is that the active elements are human beings, more specifically human beings in their role of social individual or subject. In the DEMO theory these subjects perform two kinds of acts: *production acts* (P-acts for short) and *coordination acts* (C-acts for short). By performing P-acts the subjects contribute to bringing about the goods or services that are delivered to the environment. A P-act is either material (like manufacturing and transporting goods) or immaterial (like granting insurance claims and selling goods). By performing C-acts, subjects enter into and comply with commitments toward each other regarding the performance of P-acts. Examples of C-acts are "request," "promise," and "decline." The effect of performing a C-act is that both the performer and the addressee of the act get involved in a commitment regarding the bringing about of the corresponding P-act.

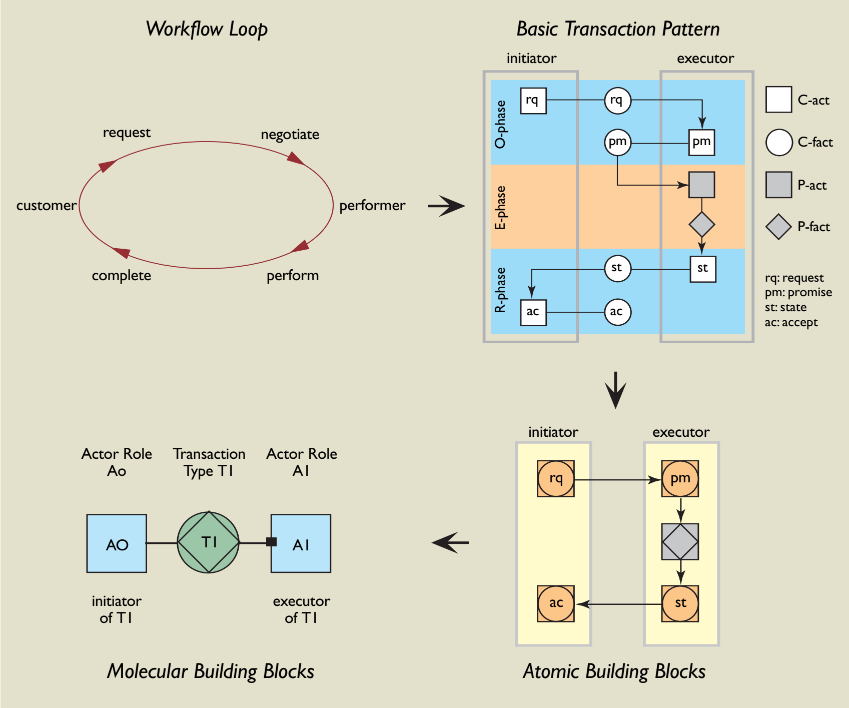

Figure 1. The atomic and molecular building blocks of organizations.

C-acts and P-acts appear to occur as steps in a generic coordination pattern, called *transaction*. It is based on the conversation-for-action [9] and the workflow loop [7]. Figure 1 depicts this workflow loop as well as the basic transaction pattern of DEMO. The main difference between both is that the transaction pattern is fully based on Habermas's Theory of Communicative Action [4]. In my experience, this theory provides the best explanation of communicative (or

[1] DEMO is an acronym for Design and Engineering Methodology for Organizations; see www.demo.nl for more information.

speech) acts. Instead of customer and performer, as used in [1], the more technical terms *initiator* and *executor* are used for the roles of the partaking subjects. The gray-lined rectangles in Figure 1 enclose the acts for which they are responsible.

A transaction evolves in three phases: the *order* phase (O-phase for short), the *execution* phase (E-phase for short), and the *result* phase (R-phase for short). In the order phase, the initiator and the executor negotiate for achieving consensus about the P-fact that the executor is going to bring about. The main C-acts in the O-phase are the request and the promise. In the execution phase, the P-fact is brought about by the executor. In the result phase, the initiator and the executor negotiate for achieving consensus about the P-fact that is actually produced (which may differ from the requested one). The main C-acts in the R-phase are the state and the corresponding accept.

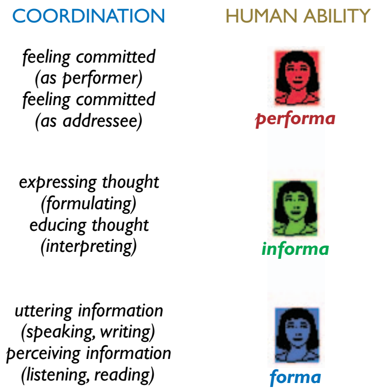

Figure 2. The three human capabilities.

The actual course of a transaction may be much more extensive than the basic pattern in Figure 1. This is accommodated in DEMO by two formal extensions of the basic pattern. The first extension covers the cases that the two actor roles dissent. For example, instead of promising one may respond to a request by declining it, and instead of accepting one may respond to a statement by rejecting it. It brings the process in the state "declined" or "rejected" respectively, which are so-called discussion states. The second extension consists of adding four cancellation patterns, one for each main transaction step (request, promise, state, accept). The basic pattern plus the extensions is called the complete transaction pattern [2, 3]. This pattern is considered to be a *socionomic law*: every transaction in every kind of organization is some path through this pattern.

Performing a C-act does not necessarily mean there is oral or written communication. In particular the promise and the accept are often performed by a nod or some other non-verbal act. For example, in a bakery, putting the bread on the counter in front of the customer counts as performing the state act. Moreover, C-acts may be performed *tacitly*. This means there is no act at all that counts as performing the C-act. Tacit C-acts must be understood as being agreed upon in an (implicit or explicit) contract that governs the transaction. An example in the same bakery is the promise; it may very well be performed without any observable sign. Tacitly performing a C-act, however, does not count less than performing it explicitly. This comes to the front in cases of breakdown (see [9]). For the sake of avoiding misunderstandings as much as possible, it is advisable to perform all C-acts explicitly. One of the major benefits of modern communication technology is that it costs very little. That is why you will find that in e-business, like your shopping on the Internet, mostly all C-acts are performed explicitly.

In the lower-right corner of Figure 1 a comprised notation is shown of the basic transaction pattern; the (C- or P-) act symbol and the corresponding (C- or P-) fact symbol are taken together. Such a combined act and fact is called an *atomic building block* of business processes. A further compression is shown in the lower-left part of Figure 1. The transaction symbol, a diamond (representing production) embedded in a disk (representing coordination), stands for the complete transaction pattern as explained previously. Next, instead of subjects, actor roles are shown.

An *actor role* is defined as the authority and responsibility to be the executor of a transaction type. Actor roles are fulfilled by subjects, such that an actor role may be fulfilled by several subjects and a subject may fulfill several actor roles. In general, actor roles do not map straightforwardly to common organizational units or functions. Ideally, an actor role is fulfilled by the same subject for all acts: the request and the accept as initiator, and the promise and the state as executor. Transaction types and actor roles are the molecular building blocks of business processes and organizations. In DEMO, a *business process* is defined as a tree structure of enclosed transactions. A transaction T2 is enclosed in transaction T1 if T2 is initiated by the executor of T1.

Extracting the Essential Map

For understanding the essential operation of organizations, the distinction between three human abilities, which are exerted both in C-acts and in P-acts (see Figure 2) are considered here. Regarding coordination, the forma ability concerns the form aspects, the informa ability concerns the content aspects, and the performa ability concerns engagement in commitments. The *first abstraction* DEMO makes in arriving at the essential map of an organization consists of taking only into account the performa ability in *coordination*, thus leaving out how C-acts are actually performed on the informa and the forma level. It results in an enormous reduction of complexity, on the average estimated to be 70% in terms of the amount of documentation [3]. This abstraction was already made in Figure 1, and thus is also achieved by the approach as discussed in [1].

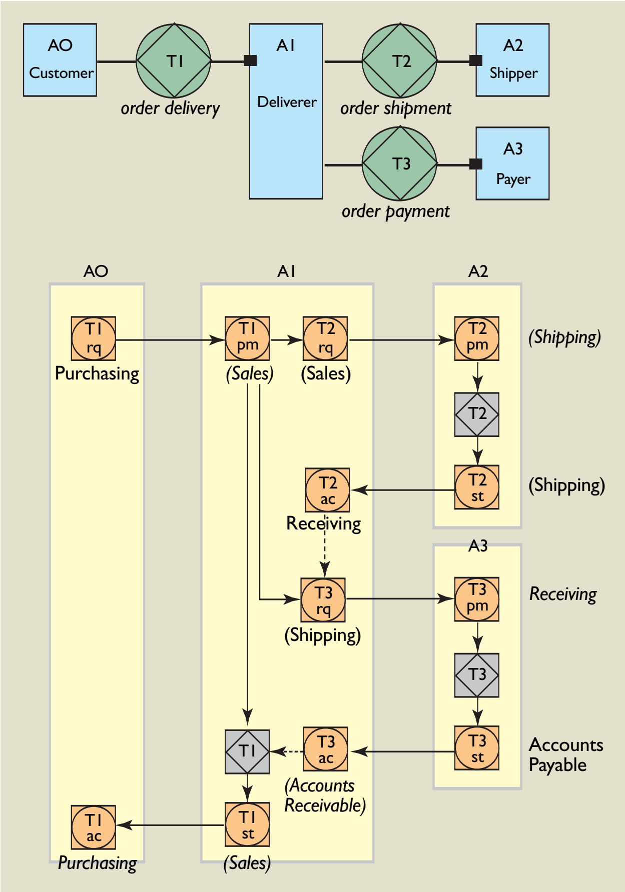

Figure 3. The essential model of the Ford case.

Regarding production, the forma ability concerns the documental production, the informa ability concerns the informational production, and the performa ability concerns the essential production. The performa ability is the *essential* human ability for doing business of any kind. The *second abstraction* DEMO makes in arriving at the essential map of an organization consists of taking only into account the performa ability in *production*, thus leaving out the P-acts on the informa and the forma level. This results in a second enormous reduction of complexity, on the average estimated also to be 70% in terms of the amount of documentation [3]. This second abstraction is a unique property of DEMO; no other contemporary methodology includes it. Certainly, there are other approaches that aim at revealing a deep structure, of which the one presented in [6] is very notable. However, none of them is built on such a solid theoretical foundation, and is therefore able to perform the second abstraction step.

Application to the Ford Case

In order to demonstrate how applying DEMO reveals the deep, essential structure of business processes, we take the well-known Ford Case [5] as an example:

> *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 'invoice-less processing'. Now when the purchasing department initiates an order, it enters the information into an online 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, the receiving clerk 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.)*

Here, the Ford case is analyzed using DEMO as the analysis instrument. To begin, the upper part of Figure 3 exhibits the essential map of the business process concerned. Only three essential transaction types are identified. The first one (T1) concerns the transfer of property of the goods delivered, referred to as "order delivery." This transaction must be distinguished from the physical shipping of the goods to the Receiving department of Ford, as represented by transaction type T2, called "order shipment." The third transaction type (T3) concerns the payment of orders and is therefore called "order payment." Because we abstract at the essential level also from

One of the benefits of having the essential model of an organization at your disposal is that it shows in a VERY CONCISE FORM some interesting things about the construction and operation of the organization.

organizational functions and departments and even from company borders, the actor roles get just 'logical' names. In this case, the initiator of T1 (A0) is called "Customer" and the executor of T1 (A1) is called "Deliverer." Actor role A1 appears to be the initiator of both T2 and T3: every transaction T1 (delivery) encloses a transaction T2 (transport) and a transaction T3 (payment). The executors of these transaction types are called "Shipper" and "Payer" respectively.

The atomic level of the business transactions is shown in the lower part of Figure 3. From the coordination state "T1 is promised" (T1/pm), actor A1 performs three acts. First, it requests a transaction T2, in order to get the goods of the order shipped. Second, it requests a transaction T3, in order to get paid for the delivery. Performing this request has to wait for T2 to be accepted (this is indicated by the dashed arrow in Figure 3 from T2/ac to T3/rq). This waiting condition reflects the agreement between Ford and the vendors that deliveries are paid after the goods have been received. Third, A1 performs the P-act of T1, thus the actual decision to transfer the ownership of the transported goods to the customer. This act must wait until T3 has been accepted (indicated by the dashed arrow from T3/ac). The actual transfer of ownership is achieved through its acceptance by the customer (T1/ac).

In order to see by whom the atomic acts in Figure 3 are performed in the new situation at Ford, the departments in Ford and in the vendor company who fulfill the identified actor roles are mentioned next to the corresponding steps. To avoid confusion, the names of the departments within the vendor company are put in parentheses. Next, a department's name is printed in italics if the C-act is performed tacitly. Some particular aspects of the process model will be explored in greater detail next.

Discussion and Evaluation

One of the benefits of having the essential model of an organization at your disposal is that it shows in a very concise form some interesting things about the construction and operation of the organization. It answers questions like "which competences do we need" (actor roles), "how complicated are our business processes" (enclosing tree structure). Also, because it has been abstracted from informational and documental issues, it is the ideal starting point for discussing current operational problems and suggesting solutions. In many projects this diagnostic application of DEMO was already sufficient for an organization; subsequently, the internal discussions were supported by a new and deep insight into its operations. Particularly in organizations with immaterial production processes (such as service industries and governmental agencies) DEMO has a high score. The explanation for this situation is that it is LAP-based: only if one recognizes that the core of organization is in the coordination processes (the entering into and complying with commitments) and not in the production processes, can one understand the operation of these kinds of organizations. Other modeling approaches easily fall in the trap that there are no 'real' business processes, only information processes.

Another benefit of the essential model is related to the redesign and the reengineering of the organization. One of the early reengineering projects is reported in [8]. Because of the abstraction in the essential model from every kind of implementation it was possible to discover the similarity of about 20 seemingly different business processes, and accordingly to propose only one supporting information system with just slight variations.

A summation of the presented benefits can be considered in respect to the Ford case, which was shown in [5] as an example of radically reengineering of business processes in conformity with the article's title "don't automate, obliterate." From the analysis of the case with DEMO, the following observations are appropriate. First, the essential model of the situation is the same before and after the radical change. This demonstrates the stability of the essential model. It makes having such a model an important commodity for every organization. Second, the 'invoice-less' process in the new situation at Ford is only invoice-less to the extent that no 'real' invoices are sent by the vendor to Ford. However, there still exists the essence of an invoice, being the request for payment. In the new situation, this request is just performed implicitly, which means there is some other act that counts as performing the request for payment. The analysis tells us that the shipment transaction (T2/st) counts as the request for the payment transaction and that the accept act (T2/ac) counts as the promise (see Figure 3). Therefore, instead of calling the new situation 'invoice-less' it would have been more appropriate to say that the received shipment document is (also) the invoice. Third, only the deep insight in the business process as provided by Figure 3 can explain the conflicts that occur as a consequence of delegation of authority, both by Purchasing to Receiving in T2/ac and by Accounts Payable to Receiving in T3/pm. The practical decision to do this and thus to deviate from the 'ideal' implementation is fully justified. At the same time it means these departments need regular meetings for discussing and tuning in the backgrounds from which they enter into and comply with commitments. This need may be well known to you already from your practical experience; now you have a better understanding of the reasons why. C

References

1. Denning, P. and Medina-Mora, R. Completing the loops. In *ORSA/TIMS Interfaces 25*, 3 (May–June 1995), 42–57.
2. Dietz, J.L.G. and Albani, A. Basic notions regarding business processes and supporting information systems. *Requirements Engineering 10*, 4 (2005).
3. Dietz, J.L.G. *Enterprise Ontology—Theory and Methodology*. Springer-Verlag Heidelberg, Berlin, New York, 2006.
4. Habermas, J. *The Theory of Communicative Action: Reason and Rationalization of Society*. Polity Press, Cambridge, 1984.
5. Hammer, M. Reengineering work: Don't automate, obliterate. *Harvard Business Review* (July–Aug. 1990), 104–112.
6. Malone, T.W., Crowston, K., and Herman, G.A. *Organizing Business Knowledge: The MIT Process Handbook* (chapter 12), MIT Press, 2003.
7. Medina-Mora, R., Winograd, T., Flores, R., Flores, F. The Action Workflow approach to workflow management technology. In J. Turner and R. Kraut, Eds., *Proceedings of the 4th Conference on Computer Supported Cooperative Work*. ACM, New York, 1992.
8. Reijswoud, V.E., Mulder, J.B.F., and Dietz, J.L.G. Speech act-based business process and information modeling with DEMO. *Information Systems Journal* (1999).
9. Winograd, T. and Flores, F. *Understanding Computers and Cognition: A New Foundation for Design*. Ablex, Norwood NJ, 1986.

Jan Dietz (j.l.g.dietz@tudelft.nl) is a professor in Information Systems Design at Delft University of Technology, The Netherlands.

© 2006 ACM 0001-0782/06/0500 $5.00