On the Nature of Business Rules

Jan L.G. Dietz

Delft University of Technology, the Netherlands
j.l.g.dietz@tudelft.nl

Abstract. Business rules are in the center of attention, both in the 'business world' and in the 'ICT applications world'. Recently, the OMG has completed a major study in defining the notion of business rule and its associated notions. On closer look, however, the definitions provided appear to be not as rigid and precise as one would hope and as deemed necessary. Based on the consistent and coherent theoretical framework of Enterprise Ontology, several clarifications of the core notions regarding business rules are presented. They are illustrated by means of a small example case.

Key words: Business Rule, Enterprise Ontology, DEMO, Modal Logic

1 Introduction

1.1 A survey of current business rule notions

Business rules constitute a subject of topical interest. They are presented and promoted as a means to achieve several highly valued properties of information systems (ICT applications), like flexibility, maintainability, transparency, and cost savings. Recently, the Object Management Group (OMG) adopted the SBVR standard (Semantics of Business Vocabulary and Business Rules) for specifying business objects, facts, and rules [13]. However, even in this impressive piece of work, the core notions appear not to be defined as crisply as one would whish.

One of the most well known documents regarding business rules is the authoritative work of Ronald Ross [14]. According to Ross, business rules build on terms and facts. A term is a basic noun or noun phrase in natural language. Examples of terms (taken from [14]) are:

Customer	(Basic? Atomic?)
Order	(Atomic?)
Quantity back-ordered	(Basic?)
Employee name	(Knowable?)

In order to keep the set of terms manageable, Ross proposes three fundamental tests that terms have to pass in order to be included. First, they should represent the

most *basic* things of an enterprise, i.e., they cannot be derived or computed. Second, they should be *atomic*, i.e., they should represent things that are indivisible. Third, they should be *knowable*, i.e., they should represent things that exist, rather than things hat happen. Unfortunately, no hard criteria are provided for determining whether a term succeeds or fails to pass the tests. Anticipating on the results of applying the Enterprise Ontology Theory, as presented in Section 3, we have put already some question marks next to the example list of terms above; they will be addressed later. Facts, according to Ross, are expressed by sentences that follow the subject-verb-object structure, where subjects and objects are referred to by terms. Examples of facts (taken from [14]) are:

Customer *places* order	(verbal predicate))
Order *is included in* shipment	(nominal predicate)
Employee *has a* gender	(nominal predicate)
Manager *is a category of* employee	(instance of meta fact type)

Again anticipating on the results of applying the Enterprise Ontology Theory in Section 3, the first observation to be made is that apparently fact types are represented: the sentences that can be instantiated. The fourth sentence is an exception. It cannot be instantiated since it is itself an instance, be it of a meta fact type. Second, no distinction is made between nominal and verbal predicates, i.e., between facts and (apparent) acts.

In accordance with many other authors (e.g., [10], [12]), Ross requires business rules to be declarative, instead of procedural, and to be expressed in well-formed (ideally: logical) formulas. Three fundamental categories are distinguished: rejectors, producers, and projectors. A *rejector* is a rule that constraints the behavior of a business. Rejectors can be violated. A *producer* is a rule for computing or logical derivation. A *projector* is a rule through which an action is evoked. Examples of each of these categories are:

A customer cannot rent more than one car at the same time	(rejector)
The amount to be paid is the list price plus VAT	(producer)
Reorder stock if the quantity on hand drops below some level	(projector)

Distinguishing between categories of rules undoubtedly makes sense but some questions immediately pop up: Why three? Why these three? Halpin [10], for example, proposes a subdivision of the rejectors in static and dynamic constraints, a distinction that is founded in database research. Next, the formulation of the projector example can hardly be called declarative. So, this contradicts the point of departure.

1.2 Research Questions and Research Approach

The short survey above should suffice to sketch the problem area we want to address and to formulate research questions that have both societal and scientific relevance, being motivated by the conviction that conceptual frameworks, like the SBVR, should be made much more rigid. The research questions to be addressed are:

1. Business rules appear to support and guide the operations of a business. But what is exactly their scope? In particular, how are they distinguished from design principles, as incorporated in the notion of architecture?
2. How can the notion of business rule be made crisper? Related to that, how important is the way of formulation (declarative-shape or imperative-shape)?
3. How is the notion of business rule related to the notions of business object, business fact, and business event?
4. What useful distinctions can be made in order to keep the total set of business rules manageable? Related to this: what makes a rule a business rule?

We will seek answers to these questions on the basis of a scientifically sound foundation, namely Enterprise Ontology, in particular its underlying Ψ-theory [7]. The Ψ-theory offers a coherent and consistent understanding of the operation of an enterprise. Such a theory is the basis one needs to clearly and precisely define core notions like (business) rules, (business) objects, (business) facts, and (business) events. Any other basis will at best reduce the current confusion to some extent, but not sufficiently. The ambition of the research project on which this paper reports, is to remove the confusion definitively.

In Section 2, the theoretical basis of our research approach is summarized. Space limitations force us to keep it rather concise which means that a reader who is totally unfamiliar with the notion of Enterprise Ontology may need to read some references. On the basis of the presented theory, we will clarify the notion of business rule, as well as related notions, in Section 3. The analysis is illustrated by a small example case. Section 4 contains the conclusions that can be drawn.

2 An Introduction to Enterprise Ontology[1]

2.1 Theoretical Foundations

There exist two different system notions, each with its own value, its own purpose, and its own type of model: the function-oriented or teleological and the construction-oriented or ontological system notion [2]. The *teleological system* notion is about the function and the (external) behavior of a system. The corresponding type of model is the *black-box model*. Ideally, such a model is a (mathematical) relation between a set of input variables and a set of output variables, called the transfer function. The teleological system notion is adequate for the purpose of using or controlling a system. It is therefore the dominant system concept in e.g. the social sciences, including the organizational sciences. For the purpose of building and changing a system, one needs to adopt the ontological system notion. It is therefore the dominant system notion in all engineering sciences.

[1] The contents of this section is based on the Ψ-theory [7]. The Greek letter Ψ is pronounced as PSI, which stands for Performance in Social Interaction. It constitutes the basic paradigm of the theory and conveys the underlying philosophical stance of constructivism [15].

The *ontological system* notion is about the construction and operation of a system. The corresponding type of model is the *white-box model*, which is a direct conceptualization of the ontological system definition presented below. The relationship with function and behavior is that the behavior is brought forward, and consequently explained, by the construction and the operation of a system. These definitions are in accordance with the work of Gero et al. if one substitutes their use of "structure" by "construction and operation" [9]. The ontological definition of a system, based on the one that is provided in [2], is as follows. Something is a system if and only if it has the next properties:

- *Composition*: a set of elements of some category (physical, biological, social, chemical etc.).
- *Environment*: a set of elements of the same category. The composition and the environment are disjoint.
- *Structure*: a set of influencing bonds among the elements in the composition and between these and the elements in the environment.
- *Production*: the elements in the composition produce services that are delivered to the elements in the environment.

Associated with every system is the world in which the actions of the system get their effect. The *state* of a world is a set of facts. The *state space* of a world is the set of lawful states, and the *transition space* is the set of lawful sequences of transitions. The occurrence of a transition is called an *event*.

A *fact* is something that is the case [17]. The knowledge of a fact can be expressed in a predicate over one or more objects, where an object is conceived as a bare individual [1]. We will consider only elementary facts [6, 10]. Facts can be declared, like the declaration of the concept 'car', or defined, like the definition of the concept 'van' on the basis of the concept 'car'. This notion of fact is all one needs for modeling a world. It is only a matter of convenience to conceive of *entities* next to facts. An entity type is just a unary fact type, for example the type car. Including both types and classes in a conceptual model is also a matter of convenience. An entity *class* is just the extensional counterpart of the (intensional) entity type. As an example, the class CAR = {x | car(x)}. According to the distinction between function and construction, the collective services provided by an enterprise to its environment are called the *business* of the enterprise; it represents the function perspective. Likewise, the collective activities of an enterprise in which these services are brought about and delivered, including the human actors that perform these activities, are called the *organization* of the enterprise; it represents the construction perspective. An organization is a system in the category of social systems. This means that the elements are social individuals, i.e. human beings or subjects in their ability of entering into and complying with commitments about the things that are produced in cooperation. Subjects fulfill actor roles (to be explained later). A subject in its fulfillment of an actor role is called an *actor*.

2.2 The Universal Transaction Pattern

Actors perform two kinds of acts. By performing *production acts*, the actors contribute to bringing about and delivering services to the environment of the organization. A production act (P-act for short) may be material (manufacturing, transporting, etc.) or immaterial (deciding, judging, diagnosing, etc.). By performing *coordination acts* (C-acts for short), actors enter into and comply with commitments. In doing so, they initiate and coordinate the performance of production acts. Examples of C-acts are requesting and promising a P-fact. The result of successfully performing a C-act is a *coordination fact* or C-fact (e.g., the being requested of a P-fact).
The result of successfully performing a P-act is a *production fact* or P-fact. P-facts in the case Library (see Sect. 3) are "loan L has been started" and "the late return fine for loan L has been paid". The variable L denotes an instance of loan. An *actor role* is defined as a particular, atomic 'amount' of authority, viz. the authority needed to perform precisely one kind of production act.

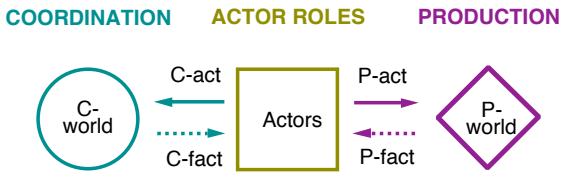

Fig. 1. The white-box model of an organization

Just as we distinguish between P-acts and C-acts, we also distinguish between two worlds in which these kinds of acts have effect: the *production world* or P-world and the *coordination world* or C-world respectively (see Fig. 1). At any moment, the C-world and the P-world are in a particular state, simply defined as a set of C-facts or P-facts respectively. When active, actors take the current state of the P-world and the C-world into account (indicated by the dotted arrows in Fig. 1). C-facts serve as agenda for actors, which they constantly try to deal with. Otherwise said, actors interact by means of creating and dealing with C-facts. The *operational principle* of organizations is that actors feel committed to deal adequately with their agenda.

P-acts and C-acts appear to occur in generic recurrent patterns, called *transactions* [4, 7]. Our notion of transaction is to a some extent similar to the notion of Conversation for Action in [16] and to the notion of Workflow Loop in [3]. A transaction goes off in three phases: the order phase (O-phase), the execution phase (E-phase), and the result phase (R-phase). It is carried through by two actors, who alternately perform acts. The actor who starts the transaction and eventually completes it, is called the initiator or *customer*. The other one, who actually performs the production act, is called the executor or *producer*. The O-phase is a conversation that starts with a request by the customer and ends (if successfully) with a promise by the producer. The R-phase is a conversation that starts with a statement by the producer and ends (if successfully) with an acceptance by the customer. In between these two conversations there is the E-phase in which the producer performs the P-act.

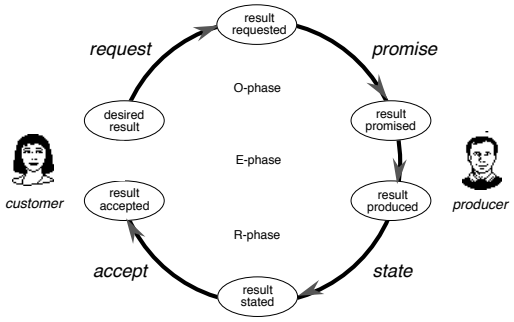

Fig. 2. The basic pattern of a transaction

In Fig. 2, we present the basic form of this transaction pattern. It shows that the bringing about of an original new, thus, ontological, production result (as an example: the delivery of a bouquet of flowers) starts with the requesting of this result by someone in the role of customer from someone in the role of producer. The original new thing that is created by this act, as is the case for every coordination act, is a *commitment*. Carrying through a transaction is a "game" of entering into and complying with commitments.

So, the process starts with the request for the bouquet by the customer, which brings the process to the state "result requested", the result being the ownership by the customer of the desired bouquet. The producer responds to the state "result requested" by promising to bring about the desired result, which brings the process to the state "result promised". This represents a to-do item for the producer: he has to comply with the promise by actually delivering the bouquet of flowers, i.e., executing the production act. In the act of handing over the bouquet to the customer, he states that he has complied with his promise. The process now comes to the state "result stated". The customer responds to this state by accepting the result. This act completes the transaction successfully.

The basic pattern must always be passed through for establishing a new P-fact. A few comments are in place however. First, performing a C-act does not necessarily mean that there is oral or written communication. Every (physical) act may count as a C-act. Second, C-acts may be performed *tacitly*, i.e. without any signs being produced. In particular the promise and the acceptance are often performed tacitly (according to the rule "no news is good news"). Third, next to the basic transaction pattern, as presented in Fig. 2, two dissent patterns and four cancellations patterns are identified [4, 7]. Together with the standard pattern they constitute the complete transaction pattern. It is exhibited in Fig. 3. Next to the basic transaction steps (a step is a combined C-act and C-fact) discussed before, there is the decline as the alternative of a promise, and the reject as the alternative of an accept. Both C-facts are discussion states, where the two actors have to 'sit together' and try to come to a (new) agreement. When unsuccessful, the transaction is stopped, either by the initiator or by

the executor. Four cancellation patterns, on the left and the right side, complete the transaction pattern, one for every basic step.

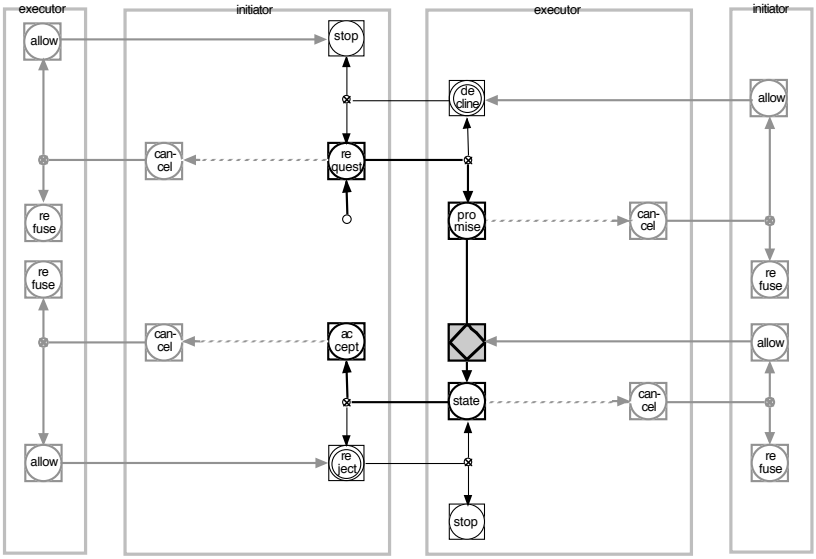

Fig. 3 The universal transaction pattern

Every *transaction process* is some path through this complete pattern, and every *business process* in every organization is a connected collection of such transaction processes. This holds also for processes across organizations, like in supply chains and networks. That is why the transaction pattern is universal and must be taken as a *socionomic law*: people always and everywhere conduct business (of whatever kind) along this pattern [7].

2.3 The Aspect Organizations

Three human abilities play a significant role in performing C-acts. They are called forma, informa and performa respectively [7]. The *forma* ability concerns being able to produce and perceive sentences. The *informa* ability concerns being able to formulate thoughts into sentences and to interpret sentences. The term 'thought' is used in the most general sense. It may be a fact, a wish, an emotion etc. The *performa* ability concerns being able to engage into commitments, either as performer or as addressee of a coordination act. This ability may be considered as the *essential* human ability for doing business (of any kind).

From the production side, the levels of ability may be understood as 'glasses' for viewing an organization (see Fig. 4). Looking through the *ontological* glasses, one observes the business actors (B-actors), who perform P-acts that result in original (i.e., non-derivable) facts. So, an ontological act is an act in which new original things

are brought about. Deciding and judging are typical ontological production acts. Ontological production acts and facts are collectively called B-things. Looking through the *infological*[2] glasses, one observes intellectual actors (I-actors), who perform infological acts like deriving, computing, and reasoning. As an example, calculating the late return fine in the case Library (Sect. 3) is an infological act. Infological production acts and facts are collectively called I-things.

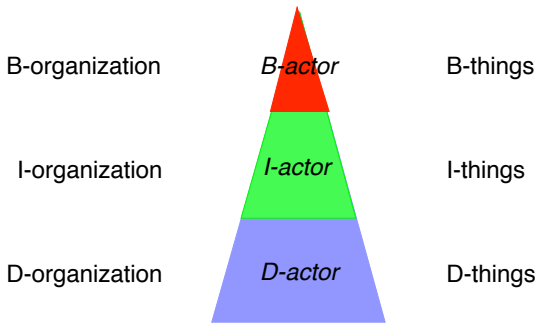

Fig. 4. Depiction of the aspect organizations

Looking through the *datalogical* glasses, one observes datalogical actors (D-actors), who execute datalogical acts like gathering, distributing, storing, and copying documents containing the facts mentioned above. So, a datalogical production act is an act in which one manipulates the form of information, commonly referred to as data, without being concerned about its content. For example, the act of recording a loan in the Library's database is a datalogical act. Datalogical production acts and facts are collectively called *D-things*.

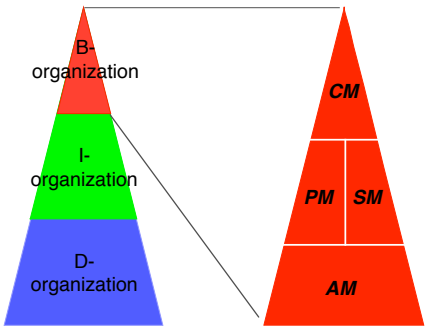

Fig. 5. The ontological aspect models

[2] The notions "infological" and "datalogical" are taken from Langefors [11].

The distinction levels as exhibited in 4 are an example of a *layered nesting* of systems [2]. Generally spoken, the system in some layer *supports* the system in the next higher layer. Conversely, the system in some layer *uses* the system in the next lower layer. So, the B-organization uses the I- organization and the I- organization uses the D- organization. Conversely, the D- organization supports the I- organization and the I- organization supports the B- organization.

In the Ψ-theory based DEMO methodology[3], four aspect models of the complete ontological model of an organization are distinguished, as exhibited in 5. The Construction Model (CM) specifies the construction of the organization: the actor roles in the composition and the environment as well as the transaction types in which they are involved. The Process Model (PM) specifies the state space and the transition space of the C-world. The State Model (SM) specifies the state space and the transition space of the P-world. The Action Model consists of the action rules that serve as guidelines for the actor roles in the composition of the organization.

Enterprise Ontology is one of the two pillars of the emerging field of Enterprise Engineering, Enterprise Architecture being the other one [8]. The paradigm of Enterprise Engineering is that an enterprise[4] is a designed artifact. Its implication is that any change of an enterprise, however small, means a redesign of the enterprise, mostly only a redesign of its construction, sometimes also a redesign of its function.

3 Assessing the Notion of Business Rule

3.1 Clarifications

As we have seen in Sec. 1, the core notion of business rule, common to all sources, is that it is a constraint on the behavior of an enterprise; it specifies what is allowable and what isn't. Within Enterprise Engineering, a clear distinction is made between the design phase and the operational phase of an enterprise [8]. In the design phase, one is concerned with the (re)design of both the function of the enterprise (its business) and the construction (its organization). The design process is guided by the applicable functional and constructional design principles. They are the operationalization of the notion of Enterprise Architecture, as explained in [8].

The *first clarification* we propose is to use the term "business rule" exclusively for the operational phase of an enterprise, thus to consider business rules as operational rules. Consequently, business rules are determined during the (re)design of an enterprise, and every change of a business rule, as well as the addition or deletion of a rule, implies a redesign of the enterprise. The relationship between business rules and enterprise policies is therefore indirect, namely via the design principles that are applied in the design phase. The knowledge sources we have referred to in Sec. 1 don't contain explicit statements regarding the distinction between design phase and opera-

[3] Design and Engineering Methodology of Organizations, see www.demo.nl

[4] We use the term "enterprise" in a most general way. It refers to companies, to governmental agencies, to unions, to not-for-profit institutions, etc.

tional phase. This prevents us from elaborating the issue, conjecturing at the same time that such a strict distinction is not made.

On the basis of the holistic enterprise ontology, as discussed in Sec. 2, the *second clarification* we propose is to consider business rules as specifications of the state space and the transition space (of both the production world and the coordination world) of an enterprise's B-organization.

Although business rules can very well be expressed in natural language [14] and in diagrammatic languages [10], the most precise and concise way is to express them by formulas in modal logic [7, 13]; conversely, one could say that it is the nature of a business rule to be a formula in modal logic. A modal logic formula is a (first-order, all-quantified) logical formula preceded by a modal operator. It appears that one can distinguish between two modal operators: *necessity* (with its negation: possibility) and *obligation* (with its negation: prohibition). Let us, based on the two modal operators, distinguish between declarative-shape and imperative-shape business rules respectively. A declarative-shape business rule expresses a constraint on the state space or the transition space of a world. Examples of a state space constraint and a transition space constraint are respectively:

A customer cannot rent more than one car at the same time
The start of a car rental has to be preceded by depositing a particular amount

An imperative-shape business rule expresses how to respond to a business event, like a procedure or protocol. It is important to notice that imperative-shape business rules do not come in addition to declarative-shape business rules but that they are operational transformations of declarative-shape business rules. Applying the imperative-shape business rules of an organization guarantees that one is compliant with the declarative-shape business rules. So, strictly spoken, one can do with only imperative-shape business rules. However, providing also the declarative-shape business rules gives much more insight in the state space and the transition space of both the production and the coordination world. Next, it is a helpful intermediate stage in formulating imperative-shape business rules. Of course, one has to take care that the imperative-shape rules and the declarative-shape rules are mutually consistent.

Projecting the modal operators on the four aspect models of DEMO (Fig. 5), it turns out that the declarative-shape business rules are contained in the State Model and the Process Model, and the imperative-shape business rules in the Action Model. Thus, the latter ones are action rules, prescribing how actors should respond to business events. As said before, declarative-shape business rules are constraints on the state space and the transition space of both the C-world and the P-world. Regarding the P-world, we propose to call its state elements business facts, being predications over business objects. Regarding the C-world, we already called C-facts business events. This is the *third clarification* we propose.

The last and *fourth clarification* we propose is to distinguish between the three aspect organizations. This separation of concerns is very useful in making the total set of 'business' rules manageable, next to the adoption of the universal transaction pattern that contains already a bunch of predefined rules. Both Halpin's categories of static constraints and dynamic constraints [10] and Ross' rejectors [14] presumably cover all three aspect organizations.

Next, we propose to reserve the term "business rule" exclusively for B-organization rules. This position is contrary to OMG's [13], where a business rule is defined as a rule under business jurisdiction. This doesn't seem to be a good criterion. First, it obfuscates that enterprises are also subject to rules from outside, e.g., from national legislation. Second, naming a concept by a term that does not reflect an inherent property of the concept is never a good idea, since the essence of the concept will not be captured.

3.2 Illustrations

To elaborate and illustrate our point of view, let us take an example case, namely the case Library [5, 7]. A general understanding of the operations of a library is sufficient to keep up with the discussion. We will focus on the processes concerning book loans. Among others, the next constraints apply:

A member cannot lend more than max_copies_in_loan at the same time	(1)
Lent books have to be returned within the standard_loan _period	(2)
A loan cannot be ended if the book copy has not been returned	(3)
Loans that are ended too late will be fined with the incurred_fine amount	(4)
A person may have more than one membership at the same time	(5)

Let us make some preliminary observations. Rule 1, 3, and 5 are state space constraints. Rules 2 and 4 are transition space constraints. Note that all rules can be violated except rule 5. Let us next project these rules on the theoretical foundations of Enterprise Ontology. Without further explanation we state that the next transactions, including their results, are involved in these rules, taken from [5, 7]:

T04 loan start	R04 loan L has been started
T05 book return	R05 the book copy of loan L has been returned
T06 loan end	R06 loan L has been ended
T07 fine payment	R07 the late return fine for loan L has been paid

The corresponding parts of the Process Model of the Library are exhibited in Fig. 6 and Fig. 7.

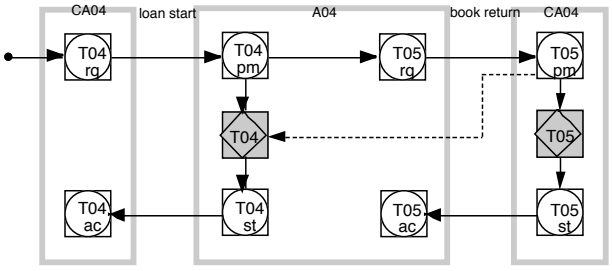

Fig. 6. Process Model of the loan start process

A box (symbol for an act) including a disk (symbol for coordination) represents a C-act (e.g., request T04) and its resulting C-fact (e.g., T04 requested), collectively called a transaction step, and indicated with e.g., T04/rq[5]. A box including a diamond (symbol for production) represents a P-act (e.g., the P-act of T04) and its resulting P-fact (e.g., the P-fact of T04), collectively called the execution step of the transaction, and indicated with e.g., T04. A solid arrow from a step S1 to a step S2 expresses the constraint that S1 precedes S2 as well as that S2 has to be performed after S1. So, for example, the promise of T04(L), where L denotes some loan, precedes the request of T05(L), and this request has to be performed once T04(L) is promised. A dotted arrow from S1 to S2 expresses only the constraint that S2 precedes S1. So, for example, the promise of T05(L) precedes the execution of T04(L), i.e., actor A04 has to wait for executing T04(L) until T05(L) is performed. The gray-lined rectangles represent the responsibility areas of the involved actor roles. For example, A04 is responsible for performing T04/pm, T05/rq, T04/ex, T04/st, and T05/ac. Note that, for the sake of simplicity, only the basic transaction pattern is shown in Fig. 6 and Fig. 7.

The corresponding, complete, part of the Action Model for actor A04 consists of the next action rules:

When T04(L) is requested, it must be declined if the total number of books in loan under the same membership as the one for L is equal to the current maximum number of copies in loan; otherwise it must be promised.
When T04(L) is promised, T05(L) must be requested.
When T05(L) is promised, T04(L) must be executed and stated.
When T05(L) is stated, it must be rejected if the book copy is damaged; otherwise it must be accepted.

Note that we have solved the having to wait until T05(L) is promised before being able to execute T04(L) by the 'trick' that T04(L) must be executed *when* T05(L) is promised. This is fully acceptable in practice, while preserving that we are dealing with an inter-transaction relationship.

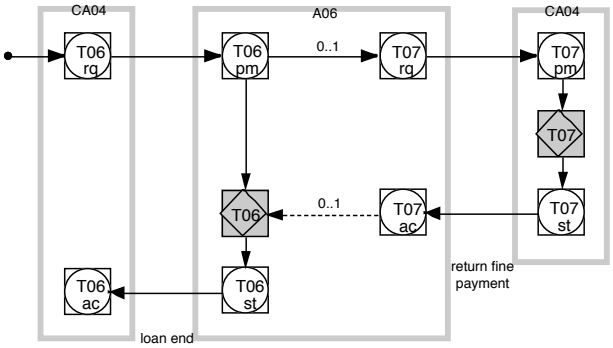

Fig. 7. Process Model of the loan end process

[5] rq stands for request, pm for promise, st for state, ac for accept, and ex for executing the P-act.

The corresponding, complete, part of the Action Model for actor A06 consists of the next action rules:

When T06(L) is requested, it must be declined if T05(L) is not accepted; otherwise it must be promised.
When T06(L) is promised, T07(L) must be requested if the acceptance of T05(L) was too late; otherwise T06(L) is executed and stated.
When T07(L) is stated, it must be rejected if the amount paid is not correct; otherwise it must be accepted.
When T07(L) is accepted, T06(L) is executed and stated.

Note that T07 is an optional enclosed transaction in T06. It will only be performed if applicable. Only in that case the last two rules are applied.

Fig. 8 exhibits the part of the State Model that corresponds with the two loan processes, as presented and discussed above, according to the diagrammatic language of WOSL [6], which is based on ORM [10]. It shows, among other things, the rule that a person may have more than one membership at the same time, and that a loan must have an associated membership and an associated book copy. The diamond shaped unary fact types are transaction results. They are the only fact types that can be created by actors, together with the objects to which they belong. All other fact types are existentially dependent on them. The object class PERSON is colored gray to express that its members are created outside the scope of the Library, contrary to MEMBERSHIP, LOAN, and BOOK COPY.

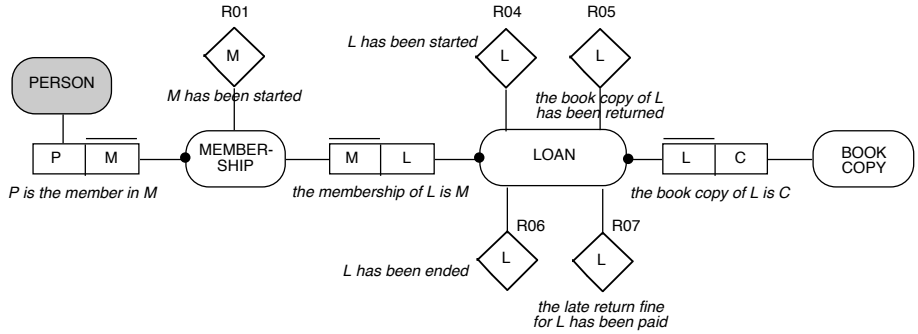

Fig. 8. State Model of (a part of) the Library

Lastly, let us address the comments we have put to citations of the knowledge sources in Sec. 1. Regarding the concept customer, two questions were raised: is it basic and is it atomic? We don't consider it basic because it is a role of a person or an institute. In the latter case it would even also be not atomic. The atomicity of the concept of order is questionable because it normally is an aggregation of things. The concept of quantity back-ordered is not considered basic because it normally would be computed.

4 Conclusions

Based on the Ψ-theory (see Section 2), a clear distinction can be made between the function and the construction of an enterprise, respectively called its business and its organization. Another useful separation of concerns that can be made subsequently regards the distinction between three aspect organizations: the B-organization, the I-organization, and the D-organization. The ontological model of an enterprise is the (Ψ-theory-based) model of its B-organization. Among other things, it contains a complete set of business object classes, business fact types, business event types, and business rules.

A business object class is the extensional counterpart of an (intensional) unary fact type, and thus not a separate concept [17]. As an example, the class CAR = {x | car(x)}. Consequently, there is only one concept needed to describe the state of a (business) world, which is the concept of business fact.

A business event is the occurrence of a transition in the coordination world of an enterprise's B-organization. It is the effect of an act by an actor in the B-organization. Business events are agenda for actors, i.e., things to which they respond by taking appropriate actions.

A business rule is a statement that constrains either the state space or the transition space of either the production world or the coordination world of an enterprise's B-organization. Defined in this way, we speak of a declarative-shape rule. It appears very practical to transform declarative-shape rules into imperative-shape rules, i.e., action rules. Ross' projectors [14] seem to be action rules.

Although there is no fundamental difference between the declarative way of formulating rules and the imperative way, one could argue that imperative-shape rules offer less freedom to act than declarative-shape rules. It is good to realize, however, that this is only a matter of appearance. An interesting topic for future research would be the relationship between the kind of an organization and the preference for one of the two shapes of rules. Our hypothesis is that it is likely to find a preference for declarative-shape rules in organizations where people have a high level of education and professionalism. Conversely, one may expect to find a preference for imperative-shape rules in organizations where this level is low. Compliance with the rules is in the first kind of organizations a matter of trust in the competence and the responsibility of people. In the second kind, it is more likely that compliance is enforced by (automated) workflow systems, in which the rules are 'hard-wired'.

Next to business rules, so the rules applicable to the B-organization, there are similar operational rules concerning the I-organization and the D-organization. Examples of an I-rule and a D-rule are respectively:

> *Customers must be informed about penalties of late return before the car rental starts (i.e., before they accept the rental by signing the contract).*
> *A copy of the driver license must be made before the customer fills out his data.*

These I-rules and D-rules are certainly not unimportant. At the same time, it is obvious that the impact of violating them on the business is far less than the impact of violating B-rules. Therefore, it seems to be a good idea to deal with them separately.

Derivation rules (Halpin [10]) or producers (Ross [14]) are infological rules; they belong to the I-organization. A subtle but important distinction can be made between the ontological definition of a fact and the infological rule by which it is computed or derived [6].

The analysis and discussion in this paper is performed in the context of Enterprise Engineering, where enterprises are considered to be designed artifacts. Business rules are part of the design and engineering of an enterprise, starting from its Enterprise Ontology. This design has been guided by the design principles of the applied Enterprise Architecture [8]. Business rules guide the operation of an enterprise; design principles guide its design.

References

1. Bunge, M.A.: *Treatise on Basic Philosophy, vol. 3, The Furniture of the World*, D. Reidel Publishing Company, Dordrecht, The Netherlands (1977)
2. Bunge, M.A.: *Treatise on Basic Philosophy, vol.4, A World of Systems* (D. Reidel Publishing Company, Dordrecht, The Netherlands 1979
3. Denning, P., Medina-Mora, R., Completing the loops. In: *ORSA/TIMS Interfaces* (1995) 25: 42-55
4. Dietz, J.L.G., Generic recurrent patterns in business processes. In: Aalst, W. van der, Hofstede, A. ter, Weske, M. (eds.) *Business Process Management*, Lecture Notes in Computer Science 2678, Springer-Verlag (2003)
5. Dietz, J.L.G., T.A. Halpin, Using DEMO and ORM in concert – A Case Study, in: Siau, K. (ed), *Advanced Topics in Database Research*, vol. 3, IDEA Publishing, London, 2004
6. Dietz, J.L.G.: A World Ontology Specification Language, in: Meersman, R., Z. Tari (eds), Proceedings OTM 2005 Workshops, Springer LNCS 3762, 2005, pp. 688-699
7. Dietz, J.L.G.: *Enterprise Ontology – Theory and Methodology*, Springer-Verlag (2006)
8. Dietz, J.L.G., Hoogervorst, J.A.P.: Enterprise Ontology and Enterprise Architecture – how to let them evolve into effective complementary notions, GEAO Journal of Enterprise Architecture, vol. 2, nr. 1, March 2007
9. Gero, J.S., Kannengiesser, U.: The situated function-behaviour-structure framework. In: *Design Studies* 25 (2004) 4: 373-391
10. Halpin, T.A., *Information Modeling and Relational Databases.* San Francisco: Morgan Kaufmann, 2001
11. Langefors, B.: Information System Theory. Information Systems **2**, 207–219 (1977)
12. Morgan, T.: Business Rules and Information Systems, Addison-Wesley, 2002
13. OMG SVBR, http://www.omg.org/docs/bei/05-08-01.pdf
14. Ross, R.G.: Principles of the Business Rules Approach, Addison-Wesley, 2003
15. Searle, J.R., *The Construction of Social Reality*, Allen Lane, The Penguin Press, London, 1995
16. Winograd, T, F. Flores, 1986. Understanding Computers and Cognition: A New Foundation for Design. Ablex, Norwood NJ.
17. Wittgenstein, L., Tractatus logico-philosophicus (German text with an English translation by C.K. Ogden), Routledge & Kegan Paul Ltd, London, 1922