

Language Based Requirements Engineering Combined with Petri Nets

Joseph Barjis

Department of Information Systems & Algorithms, Delft University of Technology, The Netherlands.
Tel: +31-15-2785827, Fax: +31-15-2786632 e-mail: J.Barjis@is.twi.tudelft.nl

Jan L.G. Dietz

Department of Information Systems & Algorithms, Delft University of Technology, The Netherlands.
Tel: +31-15-2787822, Fax: +31-15-2786632 e-mail: j.l.g.dietz@its.tudelft.nl

Tychon Galatonov

Department of Information Systems & Algorithms, Delft University of Technology, The Netherlands.
Tel: +31-15-2783886, Fax: +31-15-2786632 e-mail: tygal@is.twi.tudelft.nl

ABSTRACT

There are numerous language-action based methodologies, approaches and theories introduced and adapted for requirements engineering and organizational study for the purpose of business process modeling and information systems design. The comparative analysis of these different approaches shows, however, that many modeling techniques, tools, methodologies and approaches were developed with a special interest in the authors' mind. Some of the approaches are good in one respect but poor in others.

By organizational/business systems one means social systems, components of which are actors (human or machine) exchanging information for fulfilling the mission of the organization. Social systems and requirements for their supporting information systems can be correctly studied using the Language/Action Perspective, recently getting more importance in requirements engineering.

This paper introduces a language based COmbined Requirements Engineering Approach (CORE) for business process modeling and subsequent Information Systems design. The approach takes advantage of semiotics and Petri nets; its concept core is based on the DEMO (Dynamic Essential Modeling of Organizations) transaction concept.

1 INTRODUCTION

When engineering requirements or designing and constructing social systems model, the difficulty lies in correct identification, capture, and elicitation of the essence of business within an organization. This difficulty is increased by the high level of human involvement in social systems. Process elicitation can be done either by reading the existing documents such as procedure manuals or by interviewing those who carry out the process, the process actors, or simply by recording the words they exchange while dealing. In all cases the most common communication medium is natural language. As a starting point for requirements engineering and social systems modeling and design, we propose to use natural language as an elicitation tool. For process actors it is more natural and easier to describe what they do in natural language. But it is inevitable that many of these descriptions will contain ambiguity. If the natural language description can be automatically converted into a model having graphical form (diagram), which is also executable then, these graphical executable models can be presented to the process actors and easily verified by demonstration.

There are numerous language-action based methodologies, approaches and theories introduced and adapted for requirements engineering and organizational study for the purpose of business process modeling and information systems design. However, the comparative analysis of different approaches shows that many modeling techniques, tools, methodologies and approaches are developed with a special interest in the authors' mind. Some of the approaches are good in one respect while being poor in others.

Among others, the authors have experience to use the following modeling techniques and approaches for requirements engineering and information systems design: *DEMO methodology* (Dietz, 1999), *Semiotic approach* (Stamper *et al.*, 2000) and *Petri nets* (Aalst, 1998). Namely these three techniques are used in this

paper to achieve a more complete, sound and comprehensible tool for better requirements engineering and information systems design.

For example, semiotic approach for information systems design is a sound and helpful tool while eliciting processes from a textual description – a description in natural language. However, it does not provide a sound and clear graphical notations to represent and visualize systems. The DEMO (Dynamic Essential Modeling of Organization) methodology has a series of sound and tested models for graphical representation of systems. It has a very sound concept for process elicitation using transaction concept, however, it is difficult and sophisticated for analysts who are not well familiar with the DEMO methodology. It takes quite a while to master this methodology. Petri nets are an ideal graphical language for representation of parallel, sequential, causal, and synchronous processes; but they don't provide any tools for process elicitation.

The author proposes to study the possibility of merging different approaches and techniques for better and adequate understanding, capturing, and representation of processes for the purpose of correct information system design (Barjis *et al.*, 2001). Yes, namely right and adequate information system that supports the needs of the business and business holders.

The following figures from (Liu, 2000) confirm the author's concern about the failure of most information systems that are designed not according to the business needs. Among others, the cause is the lack of profound modeling approach, techniques, and methods.

- 31.1% of projects are cancelled before they ever get completed
- 52.7% of projects go over time and/or over budget, at an average cost 89% of their original estimates
- 16.2% of software projects are completed on time and on budget
 - In larger companies only 9% of their projects come in on time and on budget with approximately 42% of the originally proposed features and functions
 - In small companies 78.4% of their software projects get deployed, with at least 74.2% of their original features and functions

Studies show that among the leading reasons for these failures is the lack of sound and appropriate approach for requirements engineering that results in incorrect software applications and information systems design.

1.1 Research goals

So, taking into account the afore-mentioned arguments and reasons, the aim of this paper is to study and introduce a language-based COmbined Requirements Engineering Approach (CORE) helping analysts to correctly conduct all three phases of system study (for planned as well as existing systems). These three phases are Requirements Engineering, Modeling, and Simulation in order to validate the model. The first phase is to understand the needs of the users, the second – to build a model of the system, and the third – to watch/capture the dynamic behavior of the system before building a real system, and fix the system before it is built.

However, due to limited space and scope, the simulation phase is beyond the scope of this paper, and we will focus on the first two phases – requirements engineering and process modeling.

1.2 Research area

Requirements Engineering (RE) concerns the elicitation and articulation of the requirements that have to be met by an information system (software application). An information system serves the operation of some other system, usually called the object or target system. In this research project the attention is particularly focused on organizations as target systems. So, the users of an information system are human beings who use the system to perform the tasks that are assigned to them. Although new, innovative ideas are upcoming (Rolland, 1998; Siau, 1999), the common approach to RE is that the functional designers of the information system ask the future users to provide the requirements that must be met by the system. In the past two decades, substantial improvements of this approach have been achieved, ranging from structured analysis via prototyping to participatory design. The basic strategy however has remained the same: the designer asks the user what he/she wants. Jokingly, this strategy is often called the 'waiter

strategy'. Underlying the waiter strategy is the basic assumption that the user knows what he/she wants, that the user is able to formulate precisely and completely all the requirements. This, however, is rarely the case, and that makes RE according to the waiter strategy a hard and mostly unsatisfactory job. The only way out seems to be to make RE less subjective, to be less dependent on what the users say they need.

The three modeling techniques and approaches that comprise the CORE approach are Semiotic approach, DEMO methodology and Petri nets. The last one is used exclusively for its sound and comprehensible graphical notations and world-wide popularity among system analysts, modelers, designers and developers.

2 SEMIOTIC APPROACH: SEMANTIC ANALYSIS

The semiotic approach developed in the MEASUR research programme, which is now based on University of Twente, has offered a set of theories and methods for information systems development (Stamper, 1994). An information system is viewed in this paradigm as a part of human communication system. In studying the broad communication system and computerising a part of it, the important invariants are signs that are used by human agents with or without using computers. To construct an information system involves understanding and representing communication between people. The engineering of an information system is to produce a subsystem that can process signs as required by the agents. The functionalities of such a business system, as a part of a larger communication system, are determined by the social and business needs of the agents in order to fulfill their objectives in the most effective manner.

Semiotic approach comprises several methods. In this paper discussion is focused on Semantic Analysis of semiotic methods.

2.1 The Semantic Analysis Method

Semantic analysis is a process of finding the meanings of the problem at a business level and express them in a graphical form called the ontology chart. An objective of semantic analysis is to establish a requirements model in which basic patterns of behavior are represented and semantics are expressed (Stamper, 1997). The communications and norms are then described on the basis of the semantic information model (Liu *et al.*, 1999).

Semantic analysis is based on the view that information cannot be separated from those who use it. Each person (agent) can have a different view of the meaning of information through interpretation in their own context. Further, no knowledge is known without an agent being involved, and the agent cannot know without doing something. The information affects the behavior of the person interpreting it. Semantic analysis denies the existence of an objective reality.

The analysis focuses on agents-in-action by observing those agents as they undertake affordances in the context of roles. An agent is a person, group of people, or organization with responsibility for something. An affordance of an agent is some ability, or pattern of actions or behaviors. An affordance may also be a physical or conceptual thing (e.g. a table, a marriage). The roles of an agent segregate the affordances, giving more specific meaning, subject to situation and time. An agent must exist for its affordances to take place.

In contrast to many approaches to business process modeling, the semantic analysis method treats the use of language as a raw material for analysis but does not stop at the surface. It focuses analysts' attention on the meanings of linguistic expressions by examining the relationships between the use of language and behavior. In this way the analysis becomes an investigation of the patterns of business behavior and the specification relates signs to the actions if of substantive importance to the business, rather than dealing with actions of a bureaucratic nature.

The method of semantic analysis can be summarized in a few major phases. The first phase is for people involved to receive a document, which is sometimes called a *Problem Statement or Definition*, and for them to understand the problem. The people involved may be a single analyst or, for a large project, often a group of analysts with users' participation. The next phase, *the Candidate Affordance Generation*, is to produce a list of vocabulary of semantic units that may possibly used in the semantic model to describe agents and their patterns of behavior. The phase of *Candidate Grouping*, will allow further analysis of the semantic units. Following that, it is *Ontology Charting*, where a complete semantic model is produced.

Finally, the method of *Norm Analysis* will allow to identify norms and to link the norms with each part of the semantic model. Below, each of these phases will be elaborated.

It must be noted that the authors use only the first two phases of the semantic method. They consider these two phases to be a helpful starting point for requirements engineering and process modeling. As for the two following phases, Candidate Grouping and Ontology Charting, they suffer less readability and formal way of representation. Actually complementing this part of the semantic analysis, the following two methods are used – the DEMO methodology and Petri net.

2.2 Understanding the Problem Domain

A semantic analysis exercise always begins with the phase of Problem Definition. In this phase, the assignment of system development is given in the form of a written document where the problem is defined. But the written official definition would normally not be enough to capture a complete picture of the problem. Therefore, thorough investigations have to be done by studying the relevant documents. In addition, interviews with potential users of the envisaged system can be carried out for collecting supplementary information.

The following is the problem statement of the QuickQuote (QQ) case originated from (Barjis & Chong, 2000). The QQ is an internet-based insurance company pioneering business service attempting to establish a foothold in the cyberindustry. It provides a quotation comparison and purchasing service with about 35 insurance companies from all over the US (with the exception of a few states) registered with it. QQ's system on the Internet is the first real-time insurance comparison system.

In summary, QQ is like an independent insurance agent who represents several companies. The only difference is that it does not owe alliance or loyalty to any one in particular. It does not try to influence the user's choice. The influence however is not completely absent. The clients after all have a choice of only those companies that are registered with the service, which at the moment is limited to 35 insurance companies.

Looking at the process "buying a policy through the QuickQuote" the following order is observed.

A customer, in order to buy a policy, begins his search with the QQ that offers an appropriate policy (insurance company). After finding through the QQ an appropriate insurance company matching customer's requirements and wishes, the customer applies to that insurance company to buy a policy. In order for the insurance company to issue a new policy to the customer, the customer has to pay for it to the insurance company. Once a policy is paid by the customer and issued by the insurance company, according to arrangements between the insurance company and the QQ, the insurance company has to pay commissions for each issued policy arranged through the QQ.

The problem is basically introduced in the problem statement, so by reading it the analyst can gain an overview of the problem. However, sometimes the problem statement may be too vague and the requirements fuzzy. In this case, gathering more information about the problem is needed, which is normally done by collecting the relevant documents and conducting interviews with the problem-owners. Those documents and interviews, as important supplements to the formal definition, can help the analyst to understand better the words and expressions in the problem statement.

2.3 Generating Candidate Affordances

The second phase, *Candidate Affordance Generation*, is to study the collected papers and notes, and to single out the semantic units, for example, nouns, noun-phrases, verbs, verb-phrases, and prepositions, which may indicate possible agents/actors, affordances and other relationships.

In general, affordances are processes, but an affordance can appear as the subject of a process. The interpretation varies according to viewpoint. Requirements engineers must be aware of such dual interpretations.

Having in mind elicitation of the main (business) processes, in the following table, verbs and related nouns are grouped.

Verbs	Related nouns
Buy	<i>Customer, policy, QQ</i>
Begin	<i>Customer, search, QQ</i>
Offer	<i>QQ, policy, customer</i>
Find	<i>QQ, insurance company, customer</i>
Apply	<i>Customer, policy, insurance company</i>
Buy	<i>Customer, policy, insurance company</i>
Issue	<i>Insurance company, policy, customer</i>
Pay	<i>Customer, policy, insurance company</i>
Pay	<i>Insurance company, commission, QQ</i>

Table 1. Candidate processes.

In this phase, semantic units are identified from the problem definition as listed in the above table. Every word in the problem definition is in principle useful in analysis; therefore, one has to be careful when a word is going to be ignored. Even an article, a preposition and an auxiliary verb may suggest some additional information to a concept, a relationship, or an intention, though nouns and verbs are more likely to suggest concepts of actor/agent and transaction/process. The words and phrases that form semantic units are singled out as candidate transactions, and they contain complete concepts. It can be noted that some words are left out but that does not mean they will not be included in the model. Some words are taken in different forms, e.g. from the plural form of nouns to the singular form.

2.4 Candidate Grouping

The phase of *Candidate Grouping* consists of a few small steps. First of all, the semantic candidates can be categorized as agents, or affordances. Among affordances, they can be further classified into universal action-types, determiners, role-names, and so on.

Semantic items	Related actors
Buy_policy	<i>Candidate transaction</i>
Begin_search	<i>Candidate transaction</i>
Offer_policy	<i>Candidate transaction</i>
Find_company	<i>Candidate transaction</i>
Apply_policy	<i>Candidate transaction</i>
Buy_policy	<i>Candidate transaction</i>
Issue_policy	<i>Candidate transaction</i>
Pay_policy	<i>Candidate transaction</i>
Pay_commission	<i>Candidate transaction</i>
Customer	<i>Candidate actor</i>
QQ	<i>Candidate actor</i>
Insurance company	<i>Candidate actor</i>

Table 2. Candidate processes/transactions and agents/actors

In this phase, the functions of the listed semantic units are firstly studied by labelling them as actors and transactions.

It should be noted that in this phase one must take rid of duplicated and auxiliary verbs like e.g. “buy” that is repeated twice, while in both cases it indicates “the customer buys a policy”. However, the verb “pay” is not a duplication, because it refers to two different processes. In the first case the customer pays for the policy, and in the second case the insurance company pays commission to the QQ. Therefore, using the related nouns, one has to further specify them, e.g., like: “pay_policy” and “pay_commission”.

From the above description and semantic analysis of the QQ case the following conclusion can be drawn.

This analysis of table 1 and table 2 has shown how we can identify the candidate transactions of the QQ system. It also forces analysts to note that every verb in the table has two related actors. These actors are the initiator and executor of the transaction. Though not all, most of the verbs indicate transactions to be carried out by two actors (initiator and executor or, in other words, customer and supplier). Thus the semantic analysis can help us to make the analysis thorough.

However, this analysis doesn't provide sufficient and explicit information to gather all the requirements correctly and design a proper model of the system under study, especially in part of its graphical representation. The issues are: first, how to distinguish between the verbs that represent a business transaction and those being either duplications or auxiliary, or verbs that have simply descriptive or informative nature, etc.; how to visualize the processes by constructing models using sound graphical notations. For this purpose we go on to another sound and tested concept called DEMO methodology.

3 THE TRANSACTION CONCEPT OF THE DEMO METHODOLOGY

The DEMO methodology is built on a new and effective way of thinking about organizations. This new way of thinking has got the name 'OER-paradigm' (OER stands for Order, Execution, Result). It fits in the so-called Language/Action Perspective (Dietz, 1999; Reijswoud *et al.*, 1999; Goldkuhl *et al.*, 1998). Contrary to the prevailing notion that communication is exchanging sentences, expressing some proposition with regard to some world, the Language/Action 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 or communicative acts. The OER-paradigm, and accordingly DEMO, differs from other approaches in that it offers two useful additions. The first is the distinction between the essence of a business process and its realization. Otherwise said, while modeling business processes, we abstract fully from all information aspects (including calculations) and from all personnel aspects (including organizational/functional hierarchy). The second addition is that we conceive every business process to be some specific molecular structure of the 'atoms' having the same generic form. These atoms are called OER-transactions or business transactions. Some relevant papers/books are (Dietz, 1999; Reijswoud *et al.*, 1999).

An organization is conceived as a (discrete dynamic) system, of which the elements are social individuals or *subjects*, each of them having the *authority* to perform particular (*objective*) *actions* and a corresponding *responsibility* to do that in an appropriate and accountable way. For the coordination of their actions, the subjects enter into and comply with *commitments* towards each other. The changes that are brought about as the result of the actions of the subjects are discrete. This means that they are considered to take place instantaneously, and that there is a finite number of changes within a certain period. We thus classify organizations, in their essence, as social systems, where human beings play the role of social actor, distinct from their role as rational actor (i.e. knowing, communicating thoughts, and reasoning). Commitments are the effect of intersubjective actions, like requesting, promising, stating and accepting. The collective word for the *intersubjective* actions that actors perform in order to coordinate their (objective) actions is '*interaction*'. Objective actions and the related intersubjective actions appear to occur in a particular pattern, called the business transaction or *transaction* for short (cf. figure 1).

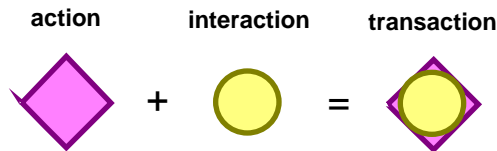


Figure 1. The business transaction concept

A transaction is carried through by two actors, in close cooperation. The generic pattern of a transaction is depicted in figure 2. The one who starts the transaction and eventually completes it, is called the customer or *initiator*, the other one, who actually performs the objective action, is called the supplier or *executor*. A transaction consists of three phases: the order phase or O-phase, the execution phase or E-phase, and the result phase or R-phase.

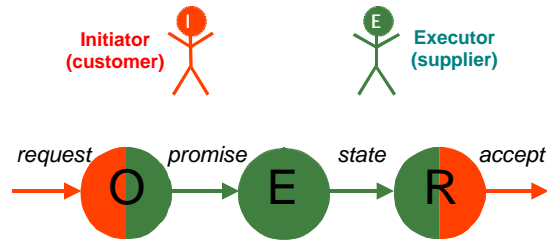


Figure 2. The transaction pattern

The order phase (O) is an interaction between the initiator and the executor that starts with the request of the initiator and ends with the promise by the executor (therefore the left half is colored orange and the right half green). It is aimed at reaching an agreement about some result in the object world to be achieved. For example, the assembler of a bicycle asks an assistant to mount a saddle and the assistant agrees to do that. The execution phase (E) starts with the promise and ends with the statement by the executor that the objective action, leading to the agreed upon result, is executed. This phase is performed by the executor solely (therefore it is all green), indicating the objective action. Example: the assistant mounts the saddle. The result phase (R) is again an interaction between the two actors, that starts with the statement and ends with the acceptance of the result by the initiator (therefore the left half is colored green and the right half orange). It is aimed at reaching mutual agreement about the established result. The result is the establishment of a fact in the object world. Example: the assistant states towards the assembler that the saddle is mounted, and the assembler accepts this.

At the moment of acceptance by the initiator, the result fact in the object world is said to come into existence, not at the moment of finishing the objective action. 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 acts (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. Although at first sight, and intuitively, material facts seem to come into existence after having finished the objective action, 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 then causes an event in the intersubject world.

During each of the phases of a transaction, new transactions may be initiated, the result of which is needed to proceed with the original one. In this way transactions are chained into arbitrarily large structures, called *business processes*.

Now returning to the analysis of the QQ case, the DEMO concept is applied as follows.

First of all, one needs to identify business/essential transactions and their related actors. For this purpose we recall the content of table 2. Applying the DEMO concept the following general rule can be formulated while identifying business transactions from the list of candidate transactions of table 2.

Each verb, in the semiotic terms referred to as affordance, represents a business transaction that triggers an action or a set of actions that creates a single fact in the objective world; such a verb can be interpreted as a core activity and can be represented as a business transaction according to DEMO.

So, if we take a look at table 2 and begin from the first semantic item then we see that the first item “Buy_policy” is just a duplication of the 6th row in the table. The second “Begin_search” doesn’t establish any fact in the real world, consequently, it is not a transaction. “Offer_policy” is based on the results of a search according to the customer’s requirements and wishes. The fact that is established is “A proper policy or company is found and offered to the customer”. So, policy_offering is treated as the first transaction (T1).

Going on, “Find_company” is also part of the first transaction. This is not a separate transaction. “Apply_policy” is the same as “Buy_policy”, so, the second transaction is policy_buying (T2). “Issue_policy” is engaged in the second transaction, so, this item is skipped. “Pay_policy” is the next transaction – policy_payment (T3). Here, it should be noted that T3, the policy_payment, must be accomplished before the T2 is complete. In other words, for the completion of T2 the insurance company needs the result of T3 and therefore T3 has to be completed during execution of T2. This situation will be

graphically represented when we come to constructing a Petri net model of the process. Finally, “Pay_commission” is still another transaction – commission_payment (T4).

All the identified business transactions are represented in table 3. This table also contains information about the facts (F#) that create these transactions, and the related initiator and executor of each transaction.

Transaction type	Result Fact Type	Initiator (Actor)	Executer (Actor)
T1 policy_offering	F1 policy P is offered	Customer	QQ
T2 policy_buying	F2 policy P is bought	Customer	Insurance Company
T3 policy_payment	F3 policy P is paid	Insurance Company	Customer
T4 commission_payment	F4 the commission for policy P is paid	QQ	Insurance Company

Table 3. Transaction types

The next step in DEMO methodology is to identify the relevant actors and to determine their roles as initiator and executor of the transaction types. Once this is done, all interaction relationships are determined. These are drawn as exhibited in figure 3. Transaction types are represented by a symbol consisting of a disk with a diamond behind it. The disk represents the ‘intersubject world part’ of the transaction, the diamond represents the ‘object world part’. Therefore, the disk can also be considered as a bank that contains the intersubject world states, and a diamond can be considered as a bank that contains the object world facts that are created as the result of successfully completed transactions. Boxes represent actors. A white box represents an elementary actor, i.e. an actor that is executor of precisely one transaction type, and a gray box represents a composite actor, also called system kernel. A transaction type is connected to its initiator by a plain line and to its executor by a line with a ‘open’ arrow point. The system boundary is represented by a gray roundangle. The diagram exhibits e.g. that actor A1 is the executor of transaction type T1 and that (an elementary actor in) the system kernel S1 (customer) is its initiator.

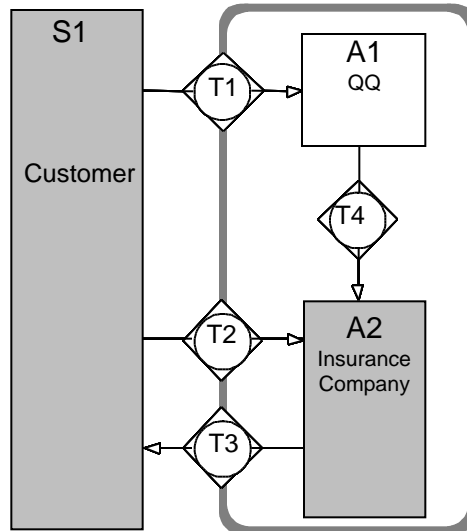


Figure 3. Interaction model of the QQ case

The following section introduces the concept behind the Petri net used in this paper for constructing a process model with respect to time order.

Definition:

The Business Oriented Petri Net or BOPN is a regular Petri net (PN) of place-transition class (PT net) that is based on the DEMO transaction concept. Consequently, the BOPN has two types of transitions: transitions representing intersubjective actions and transitions representing objective actions. The class of BOPNs adheres to the same rules as ordinary PNs. The BOPN is especially designed for modeling and analyzing business processes. At the basis of its construction lies the transaction concept, as introduced in section 3 and depicted in figure 2. BOPN models can be analyzed by the tools of standard PNs (Dietz and Barjis, 1999).

Figure 4 represents the legend of the BOPN. In the legend, the system boundary is represented by a round-angled rectangle with a thick gray line. Introduction of the system boundary helps when one needs to model or demonstrate interaction of the system with the environment (outside world). This interaction can be modeled with a special set of places on the system boundary. However, in this paper we will not elaborate the problem of drawing boundaries. There are four types of rectangles used in the BOPN. The rectangle with a plain line represents an intersubjective action, and a gray colored rectangle with a plain line represents an objective action. The double lined rectangle represents a nested (complex) transaction. The rectangle with a bold line is a compact notation for a complete transaction (comprising intersubjective and objective actions). The advantage of the compact notation is that the model becomes smaller and easier to communicate to the analysts and users. It can be applied to simple transactions, i.e. when there are no transactions initiated during its execution.

In addition to the regular Petri net arc, a conditional link is also used, shown as dotted arc. The conditional link represents a situation, when for the completion of a transaction it is necessary to complete another transaction (or transactions). Transactions may be initiated optionally, meaning that they do not necessarily take place always, but only if some condition is fulfilled.

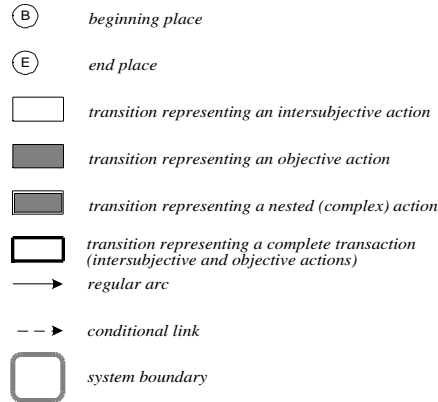


Figure 4. Legend of the BOPN

Figure 5 represents a business transaction in the BOPN notation. It shows how a single business transaction can be translated from the DEMO notation into the BOPN notation. Figure 5a represents a single transaction as a chain of three phases - order phase, execution phase and result phase, whereas figure 5b represents the compact notation of a transaction, compressing the three constituent phases of a business transaction into one rectangle. The gray colored rectangle in figure 5a represents the objective action (execution phase).

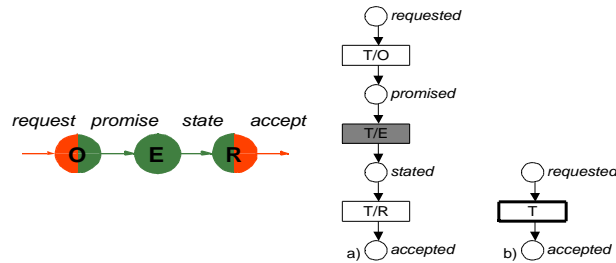


Figure 5. A business transaction in the BOPN notation: a) full notation; b) compact notation

Thus the appropriate translation from the business transaction to the BOPN replaces the phases (O-phase, E-phase and R-phase) of the business transaction with transitions and the arcs between phases with places. The places correspond to the transaction states: requested, promised, stated and accepted. They are the result of the respective intersubjective actions: request, promise, state and accept.

The core notion in the OER-paradigm is the OER-transaction, as introduced above. It serves as the ‘atomic’ building block for the ‘molecular’ structures in the operation of organizations, called (business) processes. These processes may be arbitrarily complex. Because of the high level of abstraction from the actual realization that is taken, the model of an organization from the essential perspective is often quite simple. To illustrate this, and meanwhile to demonstrate the combination of the DEMO methodology and the BOPN, the QuickQuote case is worked out in the next section.

4.1 Petri net model of QuickQuote

The diagram of figure 3 does not show how the identified transaction types are put together to form the business process(es). To this end, the BOPN is used, as shown in figure 6. It contains only the ‘intersubject world part’ of the transaction types, which are mostly split up into the three transitions O, E and R, interconnected by arcs. For example, figure 6 exhibits the structure of the business process, which consists of transaction types T1, T2, T3 and T4. T1 and T2 are initiated externally (identified as B1 and B2). T3 is initiated during the execution phase of T2. To represent this, a double lined rectangle is used. Two dotted lines indicate that T3 has to be completed before T2/E can be completed. This reflects the fact that one receives insurance policy after having paid the policy fee.

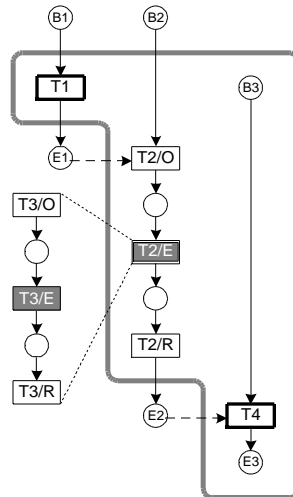


Figure 6. Process model of the QQ case

Once having a model of the system in the form of figure 6, the results can be communicated to the business owners to see if the requirements are correctly engineered. For validity and adequacy purposes, such a model can be easily simulated and animated to demonstrate how correctly it functions. Finally, this model demonstrates a strict order of the processes with respect to time.

5 CONCLUSION

The language-based combined requirements engineering approach introduced in this paper provides a good analytical tool. It is shown in the paper that social systems can be properly understood starting from human communication analysis.

The results of this paper provide a sound methodology and basis for requirements engineering, business process analysis and modeling, and information systems design.

The paper has shown that semantic analysis provides a good tool in the sense that any study should start from a problem statement and then must be followed by the study of signs (words) exchanged between actors. This is a clue to proper elicitation of processes and related actors. After this phase of study, the DEMO methodology provides a sound concept and helpful tool to elicit processes in terms of business transactions, and actors and their roles as initiator and/or executor of certain transactions.

However, a plain identification of transaction is not an ultimate goal for business processes study. It is important to put them together and to see how the entire process will look with respect to time aspects. Towards this end, the paper introduced a type of Petri net called BOPN, which is based on the DEMO transaction concept.

The proposed BOPN is easily readable, simple and adapted to capture the dynamics of intersubjective and objective actions. Another advantage of this type of Petri net is its place-transition structure. There are many available tools supporting Petri net models of place-transition type.

Finally, the results of this work are helpful for modeling and simulating business systems that are mostly social systems, where human actors exchange information in order to carry out the mission of their organization and the only starting point for analysis, modeling and design is natural language. In other words, the introduced approach is a sound way of requirements engineering to process natural language and construct models of an organization.

These are just the preliminary results of a large project. The future work will be focused towards further theoretical enrichment of the CORE approach and application of the approach to larger systems of the real world.

6 REFERENCES

- Aalst, W.M.P. van der (1998). Finding Errors in the Design of a Workflow Process: A Petri-net-based Approach. In the proceedings of the Workshop on Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98) within the 19th International Conference on Applications and Theory of Petri Nets, Lisbon, Portugal, June 22.
- Agostini, A.; Michelis, G. De. (1998). Simple Workflow Models. In the proceedings of the Workshop on Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98) within the 19th International Conference on Applications and Theory of Petri Nets, Lisbon, Portugal, June 22.
- Barjis J.; Dietz J.L.G.; Liu K. "Combining the DEMO Methodology with Semiotic Methods in Business Process Modeling". In Kecheng Liu et al. (Ed.) Information, Organization and Technology: Studies in Organizational Semiotics 1. Kluwer Academic Publishers, 2001, Boston, USA, pp.213-246.
- Barjis, J.; Chong, S. Agent-Based E-Commerce Systems Based on the DEMO Transaction Concept and Semiotic Approach. In the proceedings of the International ICSC Congress on Intelligent Systems and Applications (ISA'2000), December 11-15, 2000, Wollongong, Australia.
- Dietz, J.L.G. (1999). Understanding and modeling business processes with DEMO, in: Proc. ER'99, Annual International Conference on Conceptual Modeling, Paris, November.
- Dietz, J.L.G.; Barjis, J. (1999). Supporting the DEMO Methodology with a Business Oriented Petri Net. 12 p. In the proceedings of the Fourth CAiSE/EMMSAD'99, Heidelberg, Germany, June 14-15, 1999.
- Goldkuhl, G.; Lind, M.; Seigerroth, U. (1998). The Language Action Perspective on Communication Modeling, Proceedings of the Third International Workshop, Dept. Informatics, Jönköping International Business School, Sweden.

- Hammer, M. and Champy, J. (1993). *Reengineering the Corporation*, Harper Collins Books, New York, NY.
- Liu, K., 2000, *Semiotics in Information Systems Development*, Cambridge University Press, Cambridge, UK.
- Liu, K.; Alderson, A.; Shah, H.; Sharp, B. Applying Semiotic Methods to Requirements Recovery, In A. T. Wood-Hamper, N. Jayaratna, J.Wood (eds), Springer-Verlag, pp. 142-152, (1999), *Methodologies for Developing and Managing Emerging Technology Based Information Systems*.
- Reijswoud, V. E. van, J.B.F. Mulder, J.L.G. Dietz (1999). *Speech Act Based Business Process and Information Modelling with DEMO*, *Information Systems Journal*.
- Rolland, C., 1998. *A Comprehensive View of Process Engineering*. Proceedings of the 10th International Conference CAiSE'98, B. Lecture Notes in Computer Science 1413, Pernici, C. Thanos (Eds), Springer. Pisa, Italy, June 1998.
- Siau, K., 1999. *Method engineering: An empirical approach*. In the proceedings of the Fourth CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'99), Heidelberg, Germany, June 14-15, 1999.
- Stamper, R. K., Liu, K.; Hafkamp, M., & Ades, Y, 2000, *Understanding the roles of signs and norms in organizations - a semiotic approach to information systems design*, *Journal of Behaviour and Information Technology* 19(1), pp. 15 - 27.
- Stamper, R.K. *Organizational Semiotics*. In Mingers J. & Stowell F. (eds.) *Information Systems: An Emerging Discipline*. Mc Graw Hill, London, (1997) 267-283.
- Stamper, Ronald K. (1994) *Social Norms in Requirements Analysis*, in *Jirotko, et al.*