Capturing Information Systems Requirements Through Enterprise and Speech Act Modelling

Christer Nellborn[†] and Peter Holm[†,‡]

[†]SISU - Swedish Institute for Systems Development, Electrum 212; S-164 40 Kista; Sweden; E-mail: chn@sisu.se, pholm@sisu.se
[‡]Department of Computer and Systems Science, Stockholm University, Electrum 230, S-164 40 Kista, Sweden

Enterprise modelling is a technique for capturing and validating information systems requirements. The validity depends on how well the requirements reflect the real needs of the enterprise and how well they are understood by both requirements holder and requirements engineer. In the F^3 project[1], enterprise models are designed for modelling goals, activities, concepts and actors and linking them to information system requirements.

Speech act modelling can improve traditional process and activity models, since it introduces a richer terminology in how people use information. The speech act modelling method, developed within the NATURE project[2], also introduces a classification of the organisational use of software.

In this paper we illustrate how these two methods developed within the F^3 and the NATURE project can be combined for improving the capture and validation of business process related information system requirements. We show this by applying the methods to a common example.[3]

1 Introduction

Understanding the enterprise is important for requirements engineering. An information system that is going to be well received and beneficial to the enterprise must be based on requirements that reflect the real needs of the enterprise [8, 1, 5]. Based on a field study, Curtis and Krasner [9] described the three worst problems of system development as being:

- the lack or scarcity of application domain knowledge
- fluctuating and conflicting requirements
- communication and co-ordination breakdowns between the participants in the projects.

This first was manifested by the fact that few people really understood the problem and the application domain well enough. On a project level this led to substantial design effort being spent on co-ordinating a common understanding among the staff of

[1]ESPRIT III project 6612. See [7] for details about project purpose and scope.

[2]ESPRIT III project 6353. See [14] for details about project purpose and scope.

[3]The F^3 Enterprise Model is described in more detail in [6,16], and the speech act modelling method is described formally in [11] and is presented in [12].

both the application domain and of how the system should perform within it. The conclusion indicates that the management of learning, especially of the application domain, is a major factor in productivity, quality and costs.

Fluctuating and conflicting requirements were usually the result of market factors such as differing needs among customers, changing needs of a single customer, changes in underlying technologies of competitors' products and misunderstanding the application domain. Other sources were company-internal, such as marketing and corporate politics. The requirements were not the stable reference for implementation that they were intended to be.

The communication and co-ordination processes within a project were crucial for managing the fluctuating and conflicting requirements. Organisational boundaries hindered understanding of the requirements and temporal boundaries buried the design rationale. Complex customer interface with many varying contacts hindered the establishment of stable requirements and increased the communication and negotiation costs.

In the report on the field study, three ways of improving software productivity and quality were proposed:

- Increase the amount of application domain knowledge across the entire software development staff.
- Software development tools and methods must accommodate change as an ordinary process and support the representation of uncertain design decisions. Change management and propagation is crucial throughout the design and development process.
- The software development environment must become a medium of communication to integrate people, tools and information.

In order to achieve these goals we need to develop methods that can be used for describing relevant aspects of an enterprise in such a way that they become useful for requirements engineering, models that can express and explain business objectives, processes, and organisational structure. It is also important to develop flexible and adaptive software development methods, where the set of models used is integrated. The F^3 and NATURE projects address these issues in various ways. In this article we will focus on describing and analysing the business processes. We will also show how this can improve the capture of information systems requirements.

2 Enterprise Modelling

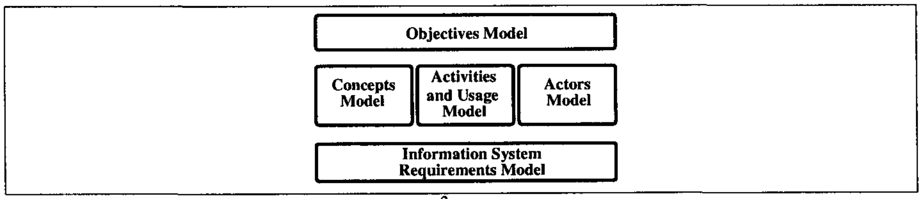
Objectives Model
Concepts Model
Activities and Usage Model
Actors Model
Information System Requirements Model

Figure 2-1. The F^3 Enterprise Model set

The purpose of Enterprise Modelling is to describe the application domain in such a way that is useful for the capture, analysis and validation of information system requirements. Willars [19] and others have described the importance of modelling the enterprises for the purpose of understanding its rationale. Within the F^3 project, a set of models is being developed for modelling the enterprise. The set contains five interlinked models as shown in figure 2-1.

2.1 The Objectives Model

This model contains components describing goals, problems, causes, business rules, opportunities, et cetera, interrelated through directed binary links. The components describe states of the application domain and the links describe how these states are related. The Objectives Model is used for analysing the rationale of the enterprise and the information system to be developed and to provide a framework where application domain processes described in the Activities and Usage Model, and information system requirements and goals described in the Information System Requirements Model can be motivated. An example of a simple Objectives Model is shown in figure 2-2.

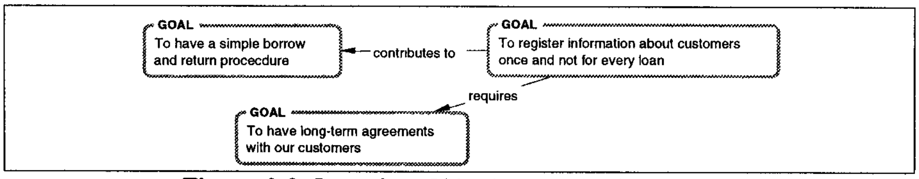
GOAL
To have a simple borrow and return procecdure
contributes to
GOAL
To register information about customers once and not for every loan
requires
GOAL
To have long-term agreements with our customers

Figure 2-2. Part of an Objectives Model for a library

2.2 The Concepts Model

The Concepts Model is used for defining the concepts, relationships, and concept properties of the application domain. As in the Objectives Model, the Concepts Model components are interrelated through directed binary links. In the Concepts Model, the important concepts of the application domain and how they are interrelated are defined. It has as a subset the concepts that will be used within the automated information system. Figure 2-3 shows an example of a Concepts Model. Concepts are defined by actors.

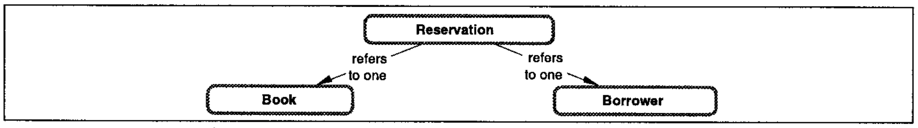

Figure 2-3. Part of a Concepts Model for a library

2.3 The Actors Model

This model defines the set of actors of the domain, (individuals, roles, organisational units, et cetera), and their interrelationships. The interrelationships are directed and binary as shown in the example in figure 2-4. The purpose of the Actors Model is to define the actors and how they are related. It can for instance be used in the information system development project to describe a complex customer organisation with many roles, groups and chains of command. Actors are responsible for requirements and goals and perform activities.

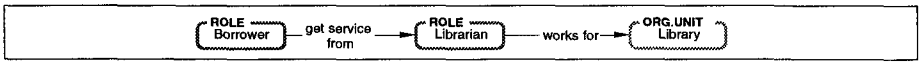

Figure 2-4. Example of part of an Actors Model for a library

2.4 The Activities and Usage Model

Activities and processes in the application domain are described in the Activities and Usage Model. The structure of this model is similar to that of traditional data-flow models. It contains activities, information sets and material sets. The information sets and the material sets go from and to activities. The purpose of modelling the activities of an enterprise is to describe the dynamic behaviour of the enterprise. In figure 2-5, the right hand side shows the traditional DFD type of modelling. The left hand side makes use of a graphical technique for describing how information can be made available without knowing in advance where that information is going to be used. This technique has been described by Janning and Sundblad in [13]. Processes are performed by actors and are motivated by the goals of the enterprise.

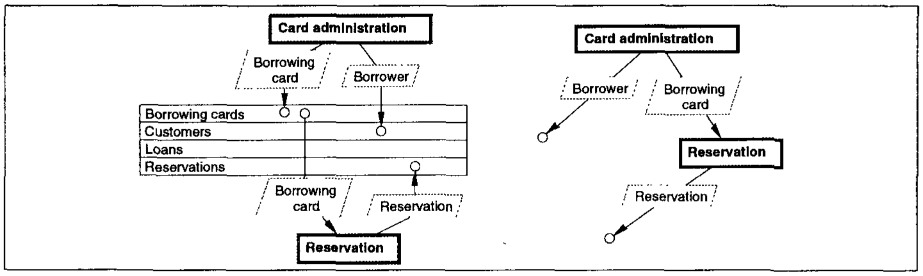

Figure 2-5. Example of an Activities and Usage Model for a library, two slightly different ways of modelling. Note: The horizontal layers in the middle of the left hand side of the figure indicate that information is made available for other activities.

2.5 The Information System Requirements Model

The Information System Requirements Model contains components such as information system goals, information system requirements et cetera. Requirements are semantically similar to goals. Both express states of affairs that should be achieved, although the word requirement is often regarded as a stronger word than goal. In the Information System Requirements Model, the word requirement is used to refer to details of the information system. Requirements should be measurable quantitatively or qualitatively. Requirements are motivated by information system goals which in turn are motivated by processes and activities or by the goals of the enterprise. Actors have the responsibility for defining requirements. Figure 2-6 illustrates a simple example.

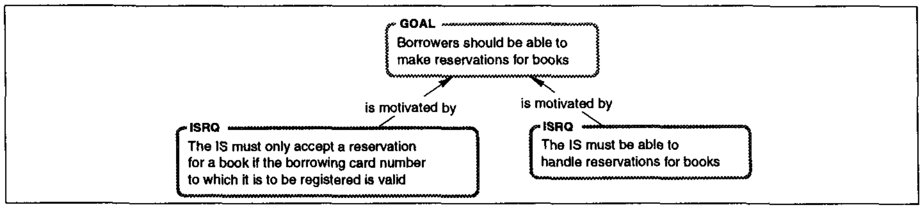
GOAL
Borrowers should be able to make reservations for books
is motivated by
is motivated by
ISRQ
The IS must only accept a reservation for a book if the borrowing card number to which it is to be registered is valid
ISRQ
The IS must be able to handle reservations for books

Figure 2-6. Example of part of an Information System Requirements Model for a library

2.6 The Requirements Engineering Responsibility

The requirements engineer and the requirements holder share the responsibility for the requirements. The requirements engineer for understanding and analysing the requirements well enough to understand how to design and develop the system, and the requirements holder for validating the requirements, i.e. that the requirements reflect the real need of the enterprise. In short, they share the responsibility for getting the *right requirements*, and for getting the *requirements right.*

3 Speech Act Modelling

The speech act modelling method proposed in the NATURE project is called the COMMODIOUS method. This is an acronym for *Com*munication *mod*elling as an aid to *i*llustrate the *o*rganisational *u*se of *s*oftware. In this paper we will show how it can be combined with the F^3 Enterprise Model. In order to do this, we will interpret the F^3 Concepts Model as a specification of statements that are made in the organisation. We will then classify some of the activities in the F^3 Activities and Usage model, as speech acts of various types, whose propositional content is described in the Concepts Model. Other activities, performed by computers, will be classified with respect to how they support the users.

The speech act theory formulated by Austin and later on developed by Searle [17, 18] has been very influential in the field of philosophy and linguistics. One of the major points for Austin, when he introduced the speech act concept, was to criticise what he called the *descriptive fallacy,* i.e. to suppose that people use language primarily to inform each other about certain states of affairs. An alternative approach is to view the use of language as consisting of different types of "speech acts". By saying things we act in different ways. Only one special kind of action is concerned with the assertions of facts. Searle has developed a taxonomy with five basic types of speech acts (illocutionary acts) [17]. Lyytinen et al. have developed a method for speech act modelling, based on Searle's theory, called the SAMPO method [2,3,4]. The method is intended to be used as a means for requirements capture in information systems development. It also aims at supporting business communication re-engineering in general. However, the SAMPO method has not focused on how we can analyse the role played by software systems. This is one of the objectives behind the COMMODIOUS method. In general the COMMODIOUS method tries to improve requirements engineering by using concepts from the speech act theory. People are viewed as performing different types of actions with the information in a database. Consider the act of making a work order. This act may be performed by storing information in a database. In this case the *information is the instrument of the action. It does not describe an action* performed somewhere else in the organisation.

In this paper we shall use a taxonomy of speech acts for so called contracting discourses. We will show how a development team may use this taxonomy to classify actions that are already described in a F^3 Activities and Usage Model. This is an attempt to improve the understanding and evaluation of the model. It will also help them to check both the Activities and Usage Model and the Concepts Model for completeness, and to identify new information needs for specific activities.

3.1 Contracting Discourses

A contracting discourse is, according to the COMMODIOUS method, a communication session between an organisation or a department and its customer. These discourses typically have two sub-discourses, one that administers long term agreements and one that administers the ordering procedure each time a service is provided. Figure 3–2 shows how this is represented graphically. Each arrow illustrates a customer-supplier relation, i.e. where one task or external agent is providing service to another task or external agent. Each symbol (or pair of symbols) of two shaking hands illustrates a contracting discourse. The large symbol represents an ordering procedure and the small symbol represents a sub-discourse for administrating long term agreements.

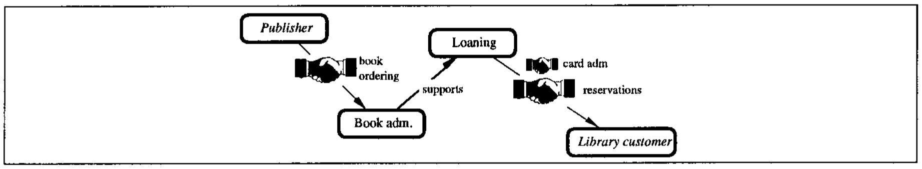

Figure 3–1. Identification of contracting discourses in a library

Each sub-discourse contains a set of speech acts of certain types, see figure 3-2. Some of them typically occur in a strictly defined sequence (group 1), others are loosely related to this sequence. The basic taxonomy for speech acts in contracting discourses as proposed in the NATURE project is based on Winograd and Flores' generic schema for "conversation for action" see, e.g. [10,15].

Taxonomy for speech acts in a contracting discourse:

Group 1 (fully sequenced): make invitation, make request, decline request, inhibit request, create commitment, describe fulfilment of service, register claim, describe fulfilment of customer obligations, report completion of service, debit customer, send invoice, report completion of customer obligations, complaint against supplier, complaint against customer, accept completion of service, accept completion of customer obligation, cancel commitment

Group 2 (partially sequenced): declare regulations, describe customer, describe supplier, describe the services

Group 3: User defined type of speech acts (specific to the company or application)

Figure 3–2. Taxonomy for speech acts in a contracting discourse

Once a user of the COMMODIOUS method has identified which types of speech acts that exists in a contracting discourse, the sequence in which they occur is relatively predictable. In a sequence diagram, like the one in figure 3-3, each arrow specifies that one speech act may follow upon another. (In the Nature project we are developing a tool that can generate such diagrams as suggestions to the user.)

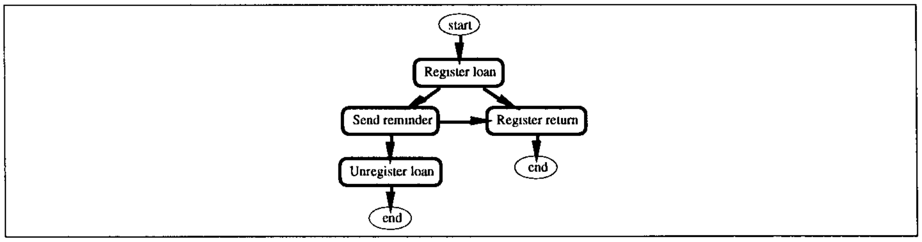

Figure 3–3. Part of a sequence diagram for a contracting discourse in a library

3.2 Classification of Software Support Functions

An essential part of the COMMODIOUS method is the classification of the organisational use of software. We propose a classification of software functions into six categories, with respect to how they are supposed to support the users.

Type of support function in the system	Type of activity supported
Resource supplying function	ordinary action
Product storage function	ordinary action
Performance function	ordinary action/ speech act
Instrumental function	ordinary action/ speech act
Action guidance function (active or passive)	ordinary action/ speech act

Table 3–1. A classification of software components, based on a characterisation of its relation to the action or task in the organisation that it is intended to support

Here follow some examples and comments to table 3-1. *Resource supplying function*: Consider the task of selling electronic books. The system stores the books and makes them available to the salesman whenever he wants to deliver them to a customer. *Product storage function*: Consider a research organisation that sells reports. Whenever the task of writing a report is finished, this report is stored in a database, together with all the other reports produced by the organisation. *Performance function* (for ordinary actions): The system performs a whole task. All robots can be considered to be agents performing tasks. Performance function (for speech acts): e.g. a system that automatically orders new material, when the company's store is below a certain level. *Instrumental function* (for ordinary actions): A support function (or subsystem) that is used as a tool, when performing a task, e.g. a word processor or a painting tool. (An alternative term might be "performance support function"). Instrumental function (for speech acts): All situations where the system mediates the communicative action, i.e. when the speech act is performed by using the system, e.g. an e-mail system or a system where a customer can make an order by directly inserting information in a database that is accessible by the supplier. (An alternative term might be "media function"). *Action guidance function*: The system instructs the user how to perform a task. In a passive action guidance function, the system simply presents information that is relevant to the task, so that the user can decide how to perform the task. We will focus on software functions that support speech acts.

The classification of support functions is meant to enhance the understanding and evaluation of a software requirements specification. It is also a way to illustrate the relevance of the enterprise model for the task of developing a software system. This can be illustrated with a simple example: Consider a library where there exists a task of lending books. There are many things we can say about this task. But, should we? How is our knowledge about the task relevant for requirements engineering and software design? By characterising the role of the future software system, the development team can get an initial idea of what knowledge is needed. How should the system guide the task of lending books? Should it passively monitor relevant information to the users, e.g. information about existing borrowing cards and reservations, or should it actively advice, instruct, or control the user? A design of an active action guidance function is (directly or indirectly) also a design of an organisational control mechanism. It must therefore be evaluated also with respect to the social institutions it creates or confirms. There is a need to discuss and analyse issues like: What are the rules for lending books? Can a book be lend to a customer, even if it is reserved for another customer? Who is allowed to change such rules?

3.3 The COMMODIOUS Method as a Complement to the F^3 Enterprise Model

In the next chapter, we shall show how we can use a part of the COMMODIOUS method as an auxiliary method in order to evaluate a partially specified F^3 Enterprise Model. We will also show how the development team will be guided in their further analysis, and how they obtain concrete guidelines for extending the models. We will apply the method according to the following schema:

- Identify the contracting discourses that exist in the enterprise
- Identify what speech acts there are in these discourses
- Create a sequence diagram (see figure 3-3) for each sub-discourse
- Identify which actions in the F^3 Activities and Usage Model corresponds to these speech acts
- Identify what parts of the F^3 Concepts Model constitute (a description of) the propositional content of these speech acts

4 An Example

In this chapter we will illustrate the basic ideas of the F^3 Enterprise Model and the COMMODIOUS method, and how they can complement each other. We will do this by using part of a library example.

4.1 Modelling a Library with the Enterprise Model Set

Populating the Objectives Model: The analysis may begin with any of the five submodels described in chapter 2. We choose the Objectives Model and focus on the goals and problems of the library example. In the example, we assume a dialogue between the requirements engineer (RE) and the requirements holder (RH) both being members of the development team.

All borrowers should be able to borrow the book they want is found to be an important goal for the library. The following question "why is it a goal?" is answered with "because it contributes to our goal to have satisfied borrowers". From this the RE concludes that there is a goal satisfied borrowers and that the first goal contributes to the second, see figure 4-1.

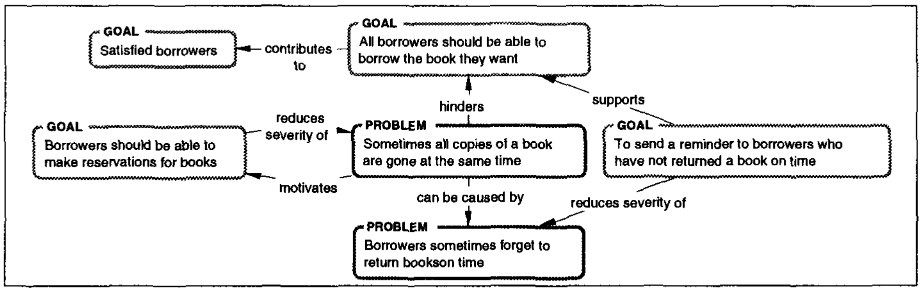
GOAL
Satisfied borrowers
contributes to
GOAL
All borrowers should be able to borrow the book they want
hinders
supports
GOAL
Borrowers should be able to make reservations for books
reduces severity of
motivates
PROBLEM
Sometimes all copies of a book are gone at the same time
GOAL
To send a reminder to borrowers who have not returned a book on time
can be caused by
reduces severity of
PROBLEM
Borrowers sometimes forget to return bookson time

Figure 4-1. Objectives Model for the library example

Next the RE analyses potential problems in the first goal. The RE finds that sometimes all copies of a book are gone at the same time. The RE notes that this hinders the achievement of the first goal. The next question is what the cause of this problem is.

The answer that borrowers sometimes forget to return books on time is given and the RE makes a note of this in the model. Asked the question "how can the problem that borrowers do not return the books on time be solved?" the RH answers that sending reminders to borrowers who have not returned a book on time reduces the problem, which is noted in the model. The rest of the model is developed in the same way.

Populating the Activities and Usage Model: The next focus is the library processes and activities and the submodel is the Activities and Usage Model. From the Objectives Model it is found that the core of the library business is lending books and a core activity is loaning. Further analysis of the library processes reveals that the library also issues and cancels borrowing cards. Information about loans, borrowers and borrowing cards is available for, and may be used by other activities. In the Activities and Usage Model we introduce two activities: Loaning and Card Administration. Through analysis of the objectives of the library, the need for two additional activities, reservation and send reminder, are detected and the Activities and Usage Model is updated accordingly, see figure 4–2. The layer in the centre of the picture is a graphical way of indicating that the information is commonly available.

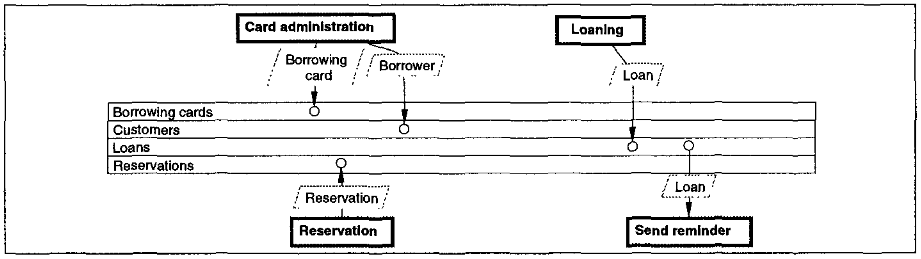

Figure 4–2. Activities and Usage Model for the library example

Populating the Concepts Model: Using the information we have gained so far, the Concepts Model may now be populated. From the Activities and Usage Model, the concepts Borrowing card, Borrower, Reservation and Loan are derived, and from the Objectives Model, the concepts Book and Copy of book, figure 4–3. The links between the concepts show how the concepts are semantically related.

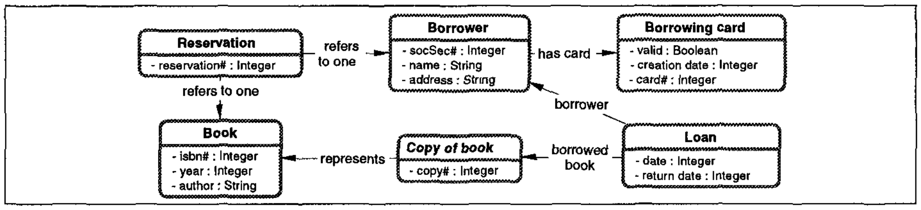

Figure 4–3. Refined Concept Model with attributes

Populating the Actors Model: The actors model in our library example is trivial: it consists of only one actor, the role of librarian.

Linking the Enterprise Submodels together: Using inter-model links between the enterprise submodels we are able to express the motivation for activities, the responsibilities for activities and for the achievement of goals, see figure 4-4.

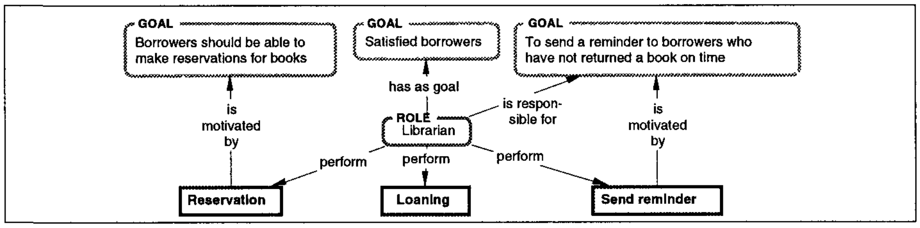

Figure 4–4. Responsibilities and motivations in the library example

The activities are motivated by goals. The role of librarian is needed to describe the responsibilities for the goals and to describe who actually performs the activities. In a more complex enterprise this becomes more complicated and requires a more thorough analysis of the actors.

4.2 Further Guidance Through Speech Act Modelling

So far we have produced a partial F^3 Enterprise Model for the library. We will now focus on how we can make the model more detailed, by using the COMMODIOUS method as a vehicle to detect incompleteness, e.g. with respect to the specification of sub-activities, information needs, and rules for activities.

The first step when applying the COMMODIOUS method is to identify customer-supplier relations in the library and to detect contracting discourses. We have already illustrated, in figure 3-1, the discourses that exist in this library. The core business task, Loaning, is providing service to the library customers (i.e. to let them borrow books). The Activities and Usage Model produced so far is obviously concerned with parts of the contracting discourse for loaning, i.e. to administer long term agreements with the customers (borrowing cards) and to administer the ordering procedure for a specific service occasion (reservations and registrations of loans). We may use the generic taxonomy of speech acts in contracting discourses to identify what speech acts need to be performed in this particular library. We can then map these to the partially specified Activities and Usage Model. This is illustrated in table 4-1. (The speech act Report to customer about available book is viewed as a speech act specific for libraries. It is therefore not an instance of a speech act type in the predefined taxonomy.)

Sub-disc.	Speech act	Speech act type	Perf. in activity
Card administration	Register new customer	Describe customer	Card administration
	Give borrowing card	Make agreement	Card administration
	Invalidate card	Cancel commitment	Card administration
Reservation	Make reservation	Create commitment	Reservation
	Cancel reservation	Cancel commitment	Reservation
	Report to customer about available book	User defined type of speech act	Reservation
	Register loan	Describe fulfilment of service	Loaning
	Send reminder	*Complaint* against customer	Send reminder
	Register return	Describe fulfilment of customer obl.	Loaning
		Cancel commitment	
	Unregister loan		Send reminder

Table 4-1. The identification of speech acts performed in the contracting discourse regarding loaning

We can also classify the concepts, attributes, and concept relations in the Concepts Model directly, by linking each one of them to the speech act that is producing the information. This is illustrated in table 4-2. (The notation Loan.{date, borrower,-} denotes the set of attributes and concepts relations related to the concept Loan. The symbol "–" denotes the concept itself.)

Sub-disc.	Predicate	Type
Card administration	Borrowing card.{ -, creation date, card#}	Information about commitment
	Borrower.{has card}	Information about commitment
	Borrower.{ -, name, address, socSec#}	Information about customer
Reservation	Loan.{ -, borrower, date, borrowed book}	Information about fulfilment if service
	Loan.{return date}	Information about fulfilment of customer obl.
	Reservation.{all attributes}	Commitment information

Table 4-2. Classification of the concepts, concept relations, and attributes in the Concepts Model

This classification of speech acts can be used to guide the interpretation and evaluation of the activities in the Activities and Usage Model. There is a big difference between making an agreement, making a promise, and describing a state of affairs. The different types of speech acts should be understood and validated in different ways, e.g. for promises and agreements we may ask if it is clear to all partners how to interpret the commitments that are created, i.e. check commitment ambiguity. For cancellations of commitments, we may ask who is allowed to do it and under what conditions, i.e. check role ambiguity, confer Auramäki et al. [1]. As the information flows and activities were described earlier, they were all treated in the same way.

The COMMODIOUS method can also help us to give the Activities and Usage Model and the Concepts Model a rationale. If there is a customer-supplier relation, there is often a need for a contracting discourse. There are general social conventions in our society as to how such communications should be performed. This can motivate the existence of specific activities, information flows, and rules. In this way we can use the notion of a contracting discourse and the speech act taxonomy as a complement to the Objectives Model.

The next step is to specify sequence conditions for the speech acts in the discourses. Figure 4-5 describes what speech acts may follow upon each other in each sub-discourse. Such a diagram should include all speech acts in group 1 (see figure 3-2). Figure 4-6 illustrates additional sequence conditions, i.e. between the two sub-discourses and between speech acts in group 1 (the fully sequenced speech acts) and others.

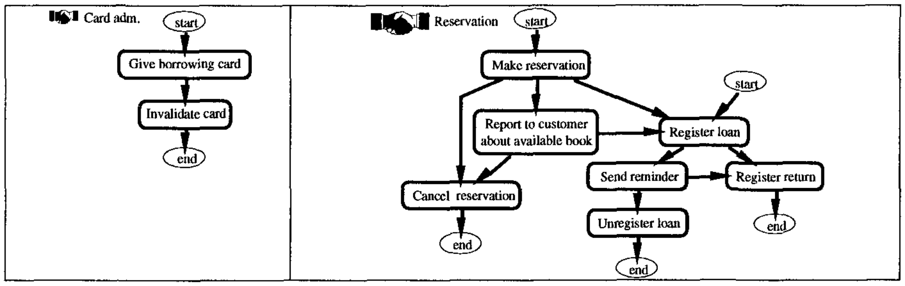

Figure 4-5. Sequence diagram for the two sub-discourses

After producing the sequence diagram, we can test the discourse for completeness by checking that each branch ends in an appropriate way and that the discourse model covers all possible moves after each speech act. It is easy to forget the speech act Unregister loan. This speech act has to be performed if a book is not returned. It is, however, easily discovered when checking that the diagram covers all things that may happen after sending a reminder. By examining the information produced by each speech act, we may also check the Concepts Model for completeness. For instance, the current Concepts Model does not contain any reminders.

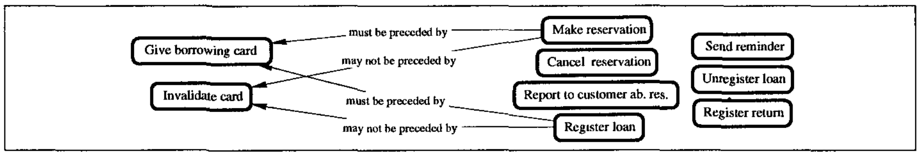

Figure 4-6. Graphical illustration of sequence conditions between the two sub-discourses

When each speech act in a sub-discourse is performed, the actors need information that is produced earlier in the discourse. There are two reasons for this. They must identify the chain of information produced in the discourse of concern and they must check that the precedence conditions are fulfilled, i.e. that a speech act that must precede the current speech act actually has occurred. They do not need to access all parts of the information produced in a specific sub-discourse, but a good heuristic rule is that they need to access the information produced by the latest speech act in the discourse, that must have been performed. For example, when Register return is performed, people need to access the information produced by the speech act Register loan, in order to identify the discourse (about a particular loan) that Register return is a part of. From this simple rule we can detect a set of missing information flows in the Activities and Usage Model. Here are some examples, which are illustrated in figure 4-7: Loaning needs the information Loan (when register returns). Send reminder needs to access the information Loan. Reservation needs to access the (public) information Reservation. Loaning needs to access the information Reservation. Some of the information flows will not be made public, though, but are local within one activity. In the activity Send Reminder, for example, they need to store information about the reminders sent to customers. This information is, however, only needed locally in this activity.

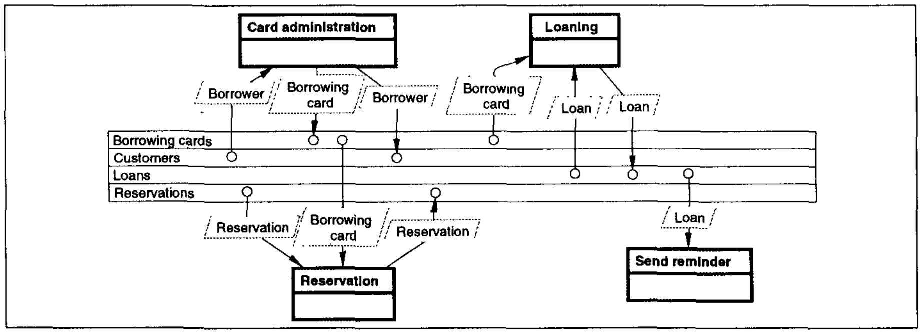

Figure 4-7. Information flows, motivated by sequence conditions

Another rule is that all speech acts that must (or may not) be preceded by a specific speech act, as specified in figure 4-6, need to access the information produced by the latter speech act. Register loan must be preceded by a Give borrowing card. It may not be preceded by the speech act Invalidate card. Hence the activity of loaning needs information about the borrowing cards. More precisely it needs the information Borrowing card.{all attributes} and Borrower.{has card}. The same holds for the speech act reservation. Analogously, Card administration needs to access (the public information) Borrower (when giving borrowing cards).

The COMMODIOUS method also supports the capturing of rules regarding certain speech acts, e.g. by looking at the sequence conditions. A set of reasonable checking mechanisms and rules for active action guidance functions can even be derived automatically, if there exists a specification of unique identifiers for each concept in the Concepts Model. Consider, for instance, this example:

Software function:	active action guidance function for registration of loans
Input:	aCardNr
Preconditions:	"The information produced by the speech act 'Give borrowing card' must exist"
	∃x,y(borrowing_card(x) & card#(x, y) & y = aCardNr)

Finally, we can use the classification of software support proposed by the COMMODIOUS method, and characterise the type of software support that each speech act should have. This classification will guide us in the further analysis. Consider the speech act Send reminder. Should it be supported by an active or a passive action guidance function? If we decide to design an active action guidance function, then the enterprise model must be made more detailed regarding this task. The system will probably contain rules that implement the customer policy on this point. Should, for instance, all customers be treated the same? If the team, on the other hand, should decide to have a passive action guidance function, such aspects may still be important to describe. However, in this case the rules are probably not as important and essential for the task of designing a software system.

By classifying support functions in this way we gain important information on how functions in the system should behave towards, and be perceived by, the user. This information is valuable input when analysing non-functional-, functional- and human-computer interface requirements, and when designing system functions. But perhaps more important as a tool for validation of the behavioural consistency of the system and as a trigger for further requirements analysis.

5 Summary

We have in this paper described two techniques, enterprise modelling and speech act modelling, for capturing, modelling and validating business process related information system requirements. We have also shown how positive synergy effects can be achieved by combining the two techniques.

6 References

1. AD/Cycle Information Model Overview, IBM, 1992

2. Auramäki E, Hirschheim R, and Lyytinen K (1992) "Modelling Offices Through Discourse Analysis: A Comparison and Evaluation of SAMPO and OSSAD and ICN", In *The Computer Journal*, 35, No 5 1992, (pp. 492 -500)

3. Auramäki E, Hirschheim R, and Lyytinen K (1992) "Modelling Offices Through Discourse Analysis: The SAMPO Approach", In: *The Computer Journal*, 35, No 4 1992, (pp. 342 - 352)

4. Auramäki E, Lehtinen E, and Lyytinen K (1988) "A Speech-Act-Based Office Modeling Approach", In: *ACM Transactions on Office Information Systems*, 6(2), (pp. 126-152)

5. Avison, D E, Fitzgerald, G, *Information Systems Development, Methodologies, Techniques and Tools*, Blackwell Scientific Publications, 1988

6. Bubenko J A jr, Gustafsson M R, Nellborn C, and Song W *Computer Support for Enterprise Modelling and Requirements Acquisition,* E6612/SISU/3-1-3-R1.B, 1992, SISU, Electrum 212, S-164 40, Kista, Sweden

7. Bubenko J A jr, Rolland C, Loucopoulos P, and DeAntonellis V (1994) *Facilitating "Fuzzy to Formal" Requirements Modelling,* In conference proceedings: IEEE International Conference on Requirements Engineering, Colorado Springs, Colorado, USA and Taipei, Taiwan, ROC, IEEE

8. Bubenko, J A, jr, *Next Generation Information Systems: an Organisational Perspective* SYSLAB Report, DSV, University of Stockholm, Sweden, ISSN 1101-8526, 1991

9. Curtis, B, Krasner, H, *A Field Study of the Software Design Process for Large Systems*, Communications of the ACM 1988 31(11): 1268 ff.

10. Flores F, Graves M, Hartfield B, and Winograd T (1988) "Computer Systems and the Design of Organizational Interaction", In *ACM Transactions on Office Information Systems,* 6 (2), (pp. 87-108)

11. Holm P (1994) *A Formal Description of the COMMODIOUS Method,* Technical Report, Manuscript

12. Holm P (1994) *The COMMODIOUS Method - Communication Modelling as an Aid to Illustrate the Organisational Use of Software*, In conference proceedings: Sixth International Conference on Software Engineering and Knowledge Engineering, Jurmala, Latvia

13. Janning, M, Sundblad, C, *Key Charts - a visualisation technique for business analysis*, Technical note, Swedish Institute for Systems Development, 1991

14. Jarke M, Bubenko J A jr, Rolland C, Sutcliffe A, and Vassiliou Y (1993) *Theories Underlying Requirements Engineering: An Overview of NATURE at Genesis* In conference proceedings: IEEE Symposium on Requirements Engineering, RE'93, San Diego, CA, Jan. 4-6, 1993

15. Medina-Mora R, Winograd T, Flores R, and Flores F (1993) *The Action Workflow Approach to Workflow Management Technology*, In conference proceedings: Third European Conference on Computer Supported Cooperative Work, (pp. 281-288), Milano, Italy

16. Nellborn C, Gustafsson M R, and Bubenko J A jr *Enterprise Modelling - an Approach to Capture Requirements,* E6612/SISU/3-1-3-R1A, 1992, SISU, Electrum 212, S-164 40 Kista, Sweden

17. Searle J R (1979) *Expression and Meaning*, Cambridge, 1979, Cambridge U P

18. Searle J R and Vanderveken D (1985) *Foundations of Illocutionary Logic*, Cambridge, 1985, Cambridge University Press

19. Willars, H, *Amplification of Business Cognition Through Modelling Techniques*, 11th IEA congress in Paris 1991, In congress proceedings