EFFEKTIV IT

AFFÄRSKOMMUNIKATION

RAPPORT NR 4 – MARS 1994

CONCEPTS AND NOTATIONS FOR OPEN-EDI SCENARIOS

Matts Ahlsén

SVENSKA INSTITUTET FÖR SYSTEMUTVECKLING

SISU

SISU bedriver ett program för forskning och utveckling inom informationsteknologins tillämpningsområden – Effektiv IT. Grunden till programmet är en förstudie inom detta område som SISU genomfört på uppdrag av Näringsdepartementet och NUTEK. Forskningen koncentreras till områden som har stor ekonomisk relevans för svenskt näringsliv och förvaltning.

Målet med programmet är att svenskt näringsliv och förvaltning ska kunna använda resultaten för att:

- Effektivare styra och utveckla verksamheter
- Minska kostnaderna för informationsförsörjningen
- Bättre utnyttja befintliga informationssystem
- Använda bättre värderings- och kalkyleringsprinciper
- Minska ledtiderna vid införande av nya system
- Förbättra intern och extern kommunikation

Arbetet under första året drivs inom fem forskningsområden: *Systemutvecklingens ledtider och kvalitet*, *Systemarvet*, *Affärskommunikation*, *IT:s ekonomi och management* samt *Verktyg för verksamhetsutveckling*.

Concepts and Notations for Open-edi Scenarios

Matts Ahlsén
Swedish Institute for Systems Development[1]
Electrum 212, 164 40 Kista,
Sweden
Internet: matts@sisu.se

Hannu Pelkonen
Finnish Data Communication Association
Salomonkatu 17 A
FIN-00100 Helsinki
Finland
X.400: G=Hannu; S=Pelkonen; O=sty; A=Elisa; C=FI

Sverre Walseth
Norwegian Telecom
Research department
PB 83, N-2007 Kjeller
Norway
X.400: G=Sverre, S=Walseth; O=Tele; OU=tf; A=Telepost; C=NO

February 1994

Abstract

This report discusses the modelling and representation of Open-edi scenarios. A reference model for Open-edi is being proposed by ISO. A central concept in this model is that of a scenario, which is intended to model business processes based on public standards and involving multiple autonomous parties. This report introduces the Open-edi reference model and provides a classification of alternative description techniques to be used in the framework of Open-edi for scenario description. An example of a scenario description is given, based on some of these notations.

---

[1] This work is sponsored by the Effective IT programme at SISU

Contents

1. Introduction

Open-edi is the common concept for business systems interoperability, which is to be based on a new generation of public standards. These are intended to cover aspects of interoperability not supported by today's standards, such as the roles and the behaviour of a number of autonomous communicating parties. Today's message oriented standards (e.g. EDIFACT) will be subsumed by such Open-edi standards. A reference model for Open-edi is currently being defined by ISO and is intended to serve as the common conceptual framework for the development of such standards. This reference model will be proposed as an international standard in itself, and is in this sense similar in aim to the OSI reference model although different in scope. A central concept in the Open-edi reference model is the scenario, intended to capture aspects of roles, behaviour and information requirements.

The aim of this report is to introduce the reference model and to review description techniques, or notations, suitable for the specification of scenarios. We give a brief description of the reference model as it has been defined thus far, explaining its objectives and basic concepts. We then give a general characterisation of notations, followed by a classification and listing of some notations that can be considered candidates for use in scenario descriptions. This is followed by an example of a scenario description.

A design method for Open-edi scenarios should provide alternative description techniques, but should also recommend suitable notations for different parts of a scenario. We do not discuss in depth methodological issues involved in designing scenarios or in applying specific notations, and our selection of candidate notations is based on an intuitive judgement of their suitability of representing the central concepts that define scenarios. The report does not cover all existing notations, but lists representative examples of notations from different application domains. In the sequel of this document we will use the word scenario in the meaning of an Open-edi scenario.

It is relevant to consider how Open-edi relates to other aspects of inter-operability, both in terms of inter- and intra-organisational information systems. The scenario concept is intended to provide a model for the interactions between a set of autonomous business participants, and as such it should not impose any structure or otherwise constrain an organisation's internal business processes or their IT-support. In this sense the Open-edi scenario can be seen as a co-operative processing model on the inter-organisational level. The prevailing approach to inter-organisational communication is focused on exchange of computer stored documents. However, standards like EDIFACT fail to deal with the co-ordination of activities as well as the implied functional aspects of EDI messages, which is one of the motives for Open-edi.

Previous approaches to business process automation and office information systems have been dominated by task and flow oriented models, where the co-ordination of procedures and form flows have been in focus. These approaches primarily support intra-organisational processes, with an implicit assumption of global control (although processes and information can be distributed). The application domain of workflow management represents a further development of these approaches. A workflow implements a process by synchronising and relating information, activities and communication [Schäl and Zeller 1993]. A number of products for computer supported workflow are available today. If we regard Open-edi as a means for co-ordinating the business processes of autonomous organisations, then scenarios could compare to a form of inter-organisational workflow descriptions.

Another important issue in this context is the need for a consensus on the concept of architecture for information systems in support of interoperability and decentralisation. Demarcation of local systems and autonomy [Veijalainen 1992] are important aspects of architecture, as is the provision of a component oriented approach to information systems assembly. A problem is the lack of a coherent framework for architecture definitions (concepts, terminology, components, etc). As an example, the notion of "client-server architecture" may have many different meanings and is insufficient for capturing the different aspects of decentralised systems. Few development methods provide support for decentralised information systems based on a well defined concept of architecture, although most methods or frameworks for information systems design include an implicit assumption of architecture [Goldkuhl, Pettersson et al. 1993]. The reference model for Open-edi can be seen as a form architecture which addresses some of these issues.

Finally, we should note that Open-edi is not a method per se, but a conceptual and technical framework in which different design methods and standards will be employed.

2. The Open-edi Reference Model

2.1 Rationale for Open-edi

The economic advantages of EDI are widely recognised. However, the costs for setting up an EDI relationship are still very high because of the need for a detailed bilateral agreement between the involved business partners and for the necessary technical agreements. Some of the issues which the participants engaged in EDI have to agree upon are:

- Syntax of information to be exchanged
- Information content (messages including versions and subsets of messages)
- Communication protocols
- Security requirements
- Legal considerations
- Business commitments as a consequence of the message exchange.

This situation has led to islands of automation within industry sectors or smaller projects. Establishing EDI relationships between different closed user groups requires a considerable effort. Therefore, most of the successful EDI implementations have been realised in long-term partnerships. EDI links in short-term partnerships are rarely realised, as the costs of the formation of such an agreement are too high.

Open-edi is an initiative that will lower these barriers by introducing standard business scenarios that can be employed without prior agreement. This enables organisations to benefit from EDI in short term relationships as well. As these scenarios will be available to organisations within all industry sectors, Open-edi will provide the necessary means for implementing EDI on a cross-sectorial level.

The Open-edi standardisation work is performed by ISO/IEC JTC1. JTC1 established a working group called SWG-EDI (Special Working Group on EDI) in 1989. SWG-EDI was disbanded in 1981 after it had finalised its work on the Open-edi Conceptual Model [SWG-EDI 1991]. According to a recommendation of SWG-EDI a new working group was established in 1991. This group was called ISO/IEC JTC1/WG3 (Open-edi). WG3 got the status of a Subcommittee (SC 30) in February, 1994. The first task of WG3 is to create an Open-edi Reference Model which will be the basis for Open-edi standardisation [ISO/IEC/JTC1/WG3 1994]. The description in this chapter is based on the proposal from this group. However, the views and ideas introduced in the rest of this report represent the opinions of the authors of this report, and not necessarily those of WG3.

2.2 The Open-edi Reference Model

The Open-edi Reference Model provides a reference framework for the identification, development, and co-ordination of Open-edi standards. This framework addresses two perspectives of the Open-edi environment, a business perspective and a technical perspective. The application of each view allows the identification of generic modelling capabilities which provides the means for standardisation in each perspective.

Figure 2.1 gives an overview over the Open-edi Reference Model and its surroundings, while it also identifies the problem areas for the two views together with the main concepts.

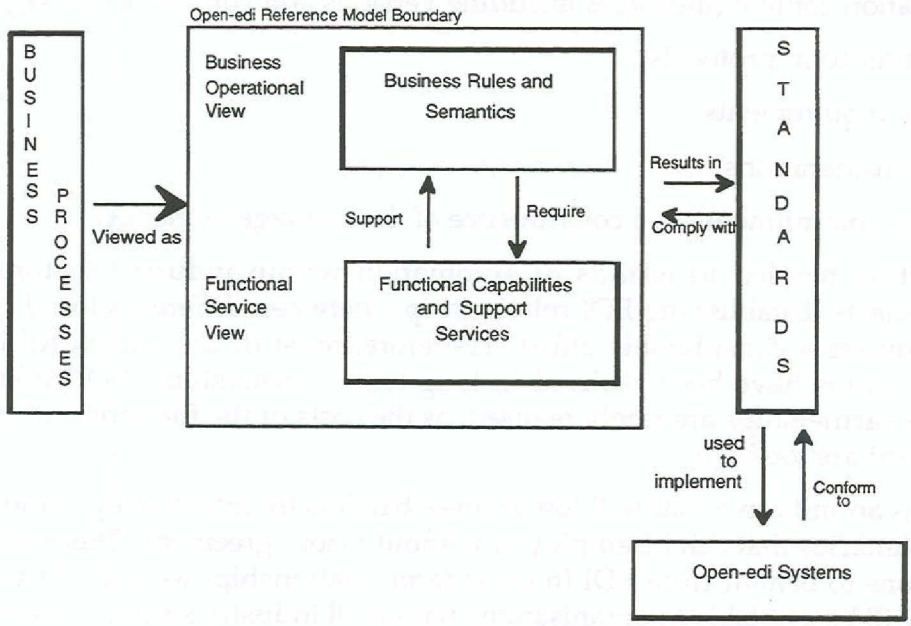

Figure 2.1: Open-edi environment

By separating the business aspects of Open-edi from the information technology aspects, the Open-edi reference model and the associated standards provide flexibility to accommodate changes in the Information Technology and the user demands without impacting the Open-edi standards related to business aspects of Open-edi.

Any information system which behaves in accordance with the Open-edi standards can be considered to be an Open-edi System.

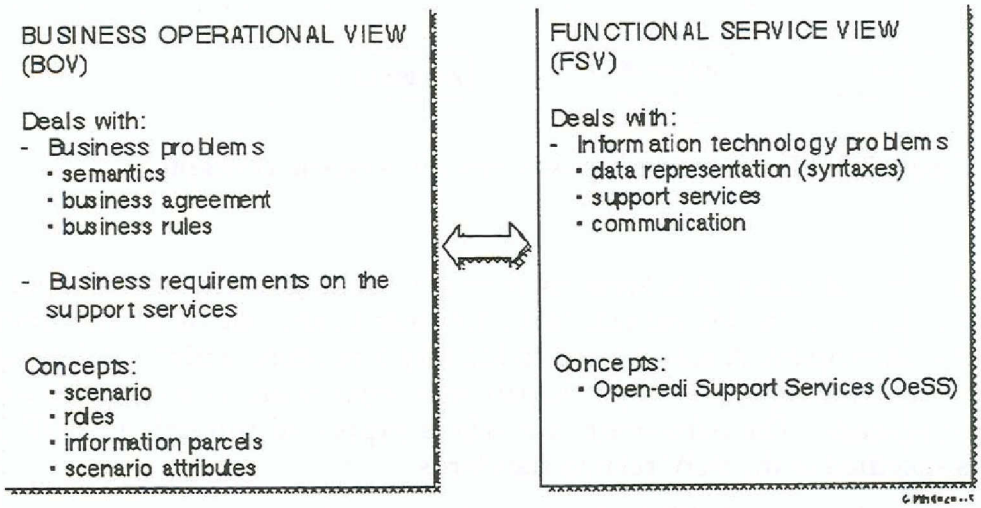

Figure 2.2: Problem areas and main concepts of the two views.

2.2.1 The Business Operational View (BOV)

The BOV addresses the business aspects of interoperability between the Open-edi systems, including business conventions, agreements and rules among business participants. The BOV related standards provide the tools for a formal business description of the external behaviour of business participants, as seen by other participants, in view of achieving a business goal. As such, the BOV related standards shall provide a means for capturing the static as well as the dynamic requirements.

The BOV related standards provide a specification of how to model an Open-edi scenario. Included in this specification is the modelling standard (including Formal Description Techniques – FDTs) to be used.

The BOV related standards shall provide for the possibility of defining Open-edi scenarios with different levels of granularity.

Open-edi scenarios cover the following aspects: entities, information units, and rules. These aspects are expressed using the modelling constructs: Scenario attributes, roles and information parcels (Figure 2.3).

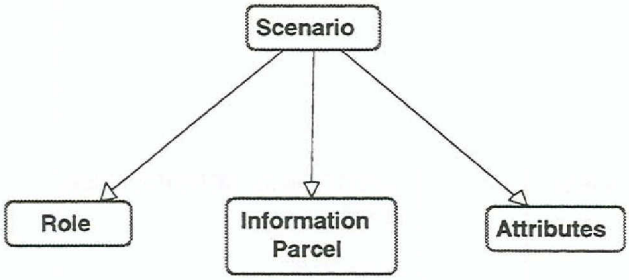

Figure 2.3: The relationship between the scenario concepts

A role is an abstract autonomous decision making entity in an Open-edi scenario, used to model the external allowable behaviour of an autonomous business participant. A role is used to identify a business activity independent of the actual participant, i.e. a participant may execute one or more roles within a given scenario. The behaviour of a role is expressed through an FDT. This FDT is specified in the BOV related standards.

An Information parcel (IP) is the formal description of the semantics of the information related to roles in an Open-edi scenario. It is used to model the business information that can be exchanged among roles.

Scenario attributes are the formal specification of information, relevant to an Open-edi scenario which is neither specific to individual roles nor information parcels.

2.2.2 The Functional Service View (FSV)

The Functional Service View addresses the information technology aspects of interoperability between the Open-edi Systems. Interoperability implies that two or more Open-edi Systems, conforming to the standards related to the FSV, are able to co-operate and support the execution of Business processes. Any private agreement between parties, other than the decision to engage in Open-edi transactions, is not necessary. The FSV identifies the entities which are generic functional capabilities of Open-EDI systems. These functional capabilities may be distributed among several organisations (by delegation from the concerned business participants).

An Open-edi Support Service (OeSS) is an abstract entity used in the FSV to model a set of generic functional capabilities, needed to support the execution of an Open-edi transaction.

An Information Management Domain (IMD) is an entity that groups at least one OeSS and may include role specification(s) and decision making. The purpose of IMDs is to instantiate a real operational configuration able to support Open-edi transactions between Open-edi systems.

2.3 Intended use and Users of the Standards

In the real world cost effective implementation of EDI requires co-operation from different types of experts, mainly business users aided by information analysts and information technology specialists including telecommunication experts. The Open-edi reference model facilitates this co-operation by providing two sets of standards, each set being used by a class of experts.

The Open-edi Reference Model will be used by the various standard bodies to develop standards which support both the Business Operational View and the Functional Service View. The potential exists for these standards to come from one or more different standards groups. In addition to using the Open-edi Reference Model, the standards bodies will also reference other models or standards as required.

Generally the standards related to the Business Operational View will be developed and used by business representatives or users. The business representatives are those who understand the operating aspects of a business domain, business or enterprise. Once Business Operational View related standards are in place, user groups will use them to produce agreed upon models which represent their business processes. These agreed upon models will then be registered with an Open-edi registration authority.

The standards related to the Functional Service View will be developed by the information technology (IT) experts. The information technology experts are those within an enterprise who understand the information technology and use this technology to design and build systems which support the business needs. The standards developed to support the Functional Service View shall take into account the standards developed to support the Business Operational View.

Figure 2.4 shows the content and the users of the different Open-edi standards.

STAND-ARDS	Open-edi Referance Model	BOV related standards	FSV related standards
CONTENT	Framework for coordination of standards	- Modeling standarc for Open-edi scenario - Procedures for introducing/ updating Open-edi scenario in the repository - Procedures for Information Parcel repository - Catalogue of requirements on OeSE - Procedures for Role repository	- List of the OeSS - Description of their external behaviour - Identification of their mutal cooperation
USERS	Standardization bodies	Business Process Designers	IT-experts

Figure 2.4: Content and users of Open-edi standards

2.4 Requirements on FDTs used to model Open-edi Scenarios

FDTs are required to express an Open-edi Scenario in an unambiguous way. This is essential for the reusability and cross-sectorial use of Open-edi Scenarios.

Open-edi Scenarios are expressed through the constructs of roles, information parcels and scenario attributes. The constructs cover different aspects of a business process and different requirements for the FDTs used to express them. Together they form an unambiguous picture of a business process.

2.4.1 Requirements for FDTs used to express a Role

A role within an Open-edi scenario shall express the external behaviour of a participant in a business process. An FDT must be able to express the following properties of a role:

- The reception of information parcels (IP) from other roles
- The submission of IPs to other roles
- Identification of which roles IPs are exchanged between
- The sequencing of IP exchanges within a role
- The result of any internal event or decision which affects the external behaviour of a role
- The state of the role at any time

2.4.2 Requirements for FDTs used to express Information Parcels

An information parcel is used to model the business information which is exchanged between roles. An FDT must be able to express the following properties of an information parcel:

- The structure of the information parcel
- The content of constituent components of an information parcel
- An identity for the information parcel which at least is unique within a scenario
- Objective or purpose

2.4.3 Requirements for FDTs used to express Scenario Attributes

Scenario attributes are the formal specification of information relevant to an Open-edi scenario as a whole. An FDT must be able to express the following properties of scenario attributes:

- Information model of all IPs
- Overall role relationships
- Registration information
- Reference to laws and regulation that applies to the scenario

2.5 Introduction to an Open-edi Scenario

A relatively simple situation from the health care sector is chosen as an example. Let us consider the situation in which a business participant acts as a centre or agency of acquiring organs for those that need organ transplants. The roles to be modelled are:

- the organ requester
- the organ centre
- the organ donor

In this example only one organ requester, organ centre and organ donor is shown, while in general several requesters and donors could be connected to one centre. It would also be possible to extend this example to include transporters, finance departments, banks, etc. However, the goal with an example is an illustration, not an exhaustion.

The behaviour of the different roles of our example may briefly be described as:

The organ requester will

- Request the organ centre for organs.
- Remind the organ centre of earlier requests.
- Accept organs.
- Cancel requests (Patient died (or recovered!), or internal supply of organ).
- Receive refusal of request from organ centre.

The organ centre will:

- Receive and reply on requests for organs from the organ requester.
- Request organs from organ donor if no organ is locally available.
- Receive organs from organ donor .
- Refuse requests.

- Cancel request towards organ donor (request cancelled by organ requester or internal supply available).

The organ donor will:

- Receive requests for organs from the organ centre.
- Receive cancellations for earlier requests.
- Offer organs to the organ centre.

This Open-edi scenario is described in section 4, using some common description techniques.

3. Notations

3.1 Characterisation

This section gives a general characterisation of notations relevant for the description of scenarios. We are concerned with notations intended for representing different aspects of information systems. This report is not a survey of notations, and we do not pass judgement on individual notations. We use *notation* as synonymous to (formal) description technique (FDT), formal language, modelling language or formalism. We distinguish notations from the general concept of *model*; a notation is used to express a model of some domain[1], for Open-edi the domain is that part of a business process concerned with the interactions between business participants, and the resulting models are called scenarios. Further, a notation does not necessarily prescribe a specific *process* or *method* on how to arrive at a certain model. Some notations may however, be specifically intended to support a certain method.

The application of modelling notations to the analysis and specification of information systems falls within the area of conceptual modelling. Conceptual modelling covers the static and dynamic (behavioural) properties as well as the rules of the application domain. In this process a number of different notations must generally be employed. A conceptual model should ideally capture all these aspects, and in its most abstract sense, a scenario can be considered a conceptual model. Different classes of conceptual models (and corresponding notations) include: information models, process models and event models. Some research issues and a state of the art in conceptual modelling can be found in [Loucopoulos and Zicari 1992]. Semantic data models have been surveyed in [Hull and King 1987; Peckham and Maryanski 1988], and a comparative survey of diagramming techniques supported by CASE tools can be found in [Rock-Evans and Engelien 1989].

In general, a notation may emphasise a certain modelling perspective, by including specific symbols or language constructs for some model concept (e.g. processes and information objects). Further, a notation could possibly be executable; a notation can have multiple representations, such as graphs or lexical elements; a notation may include proof procedures, by which certain properties of a model can be verified or established. Another distinction can be made with respect to the intended use of a notation; some notations emphasise ease of description, e.g., in terms of notation syntax (e.g. graphs) and structuring capabilities, whereas others emphasise formal tractability, such as the verification of a specification.

---

[1] Often these two concepts are used as synonyms, e.g., notations based on the ER-approach are often collectively referred to as ER-models.

Notations which can be interpreted mechanically are called constructive. Most general examples of these kinds of notations are programming languages, but also several other formal descriptive techniques have this property. The opposite are notations which are non-constructive. This type of notations do not give an explicit model of the system being specified. The non-constructive notations specify systems in terms of properties and invariant conditions. The non-constructive notations usually have no mechanically derivable implementations.

In this context a (formal) notation is understood in a broad sense to include both simple graphical languages as well as more complex rigorously defined formalisms. In general, notations should have the properties of, (1); promoting abstraction, in order to be an instrument to handle complexity, and, (2); providing preciseness in notation, so as to avoid ambiguity in interpretations and to ease formal tractability. Given these properties, we can say that the primary purposes of notations are, to serve as an instrument for communication, thereby being usable in a specification and design process; and to be a basis for providing executable descriptions. Notations may of course be supported by computerised design tools, regardless of whether the models are executable or not.

A simple classification of some notations relevant to Open-edi can be based on; (1) the primary modelling perspective, and; (2) on the formal properties that make them suitable for a particular application. A classification based on syntactical and presentation properties (e.g. graphical, lexical) is possible but can be limited since some notations have multiple syntaxes and presentations. Here we consider the following modelling perspectives,

- Process/state synchronisation

- Information structuring

- Activity and data flow composition

- Activity co-ordination

There are a number of notations intended for synchronisation of processes and states, e.g. in distributed systems specification and protocol description. A process is here understood to be a computer process. Notations have been developed within ISO and CCITT. Languages like LOTOS, ESTELLE and SDL [ISO/IEC 1991] which also are standards today, dominate the specification of services in telecommunication systems. Object-oriented extensions to some of these languages have also been proposed, e.g. an object-oriented extension of SDL [Möller-Pederson, Belsnes et al. 1987].

Information structuring notations refer to languages designed primarily to classify and structure information objects or concepts, in order to describe the static structure of information systems. Typically, such languages are used both for analysis and in design, e.g. in conceptual modelling and (data base) schema design. There are a number of such languages based on the entity-attribute-relationship approach. Notations for information structuring are often referred

to as conceptual (data) models, information models or semantic data models. Whereas conceptual and information model notations are intended for analysis, a semantic data model is more focused on design (e.g. in terms of database structures). However, the emergence of object-oriented modelling notations gradually lessens this distinction.

Activity and data flow formalisms include a number of extensions and variations of the well-known data flow diagrams. Their primary application is in modelling information flows between related processes or functions. Some of these languages also provide synchronisation primitives and specification of constraints. Decomposition into process/function hierarchies allowing multiple levels of abstraction is another property of these notations.

Activity co-ordination refers to the modelling of processes involving both automated systems and human activity. The definition of roles of interacting agents and the co-ordination of activities are central aspects, but also the modelling of the commitments and obligations attributable to roles. Process modelling is receiving an increased attention involving such areas as concurrent engineering and enterprise modelling. Applications include the modelling of software development processes [Curtis, Kellner et al. 1992], or other co-operative design processes. Some approaches to workflow management include such co-ordination support, notably those based on speech-acts (se below).

Orthogonally to the above classification based on modelling perspectives we can consider the formal basis of the notations relevant to Open-edi. A possible distinction can be made between,

- Automata/Finite state machines
- Process Algebra
- Semantic net/Relational
- Linguistics / Speech Acts
- Rule-based/Logic-based

Automata formalisms, such as finite state machines (FSM), are prime candidates for distributed systems and protocol applications. Generally, in state-oriented notations the state of the modelled system is explicit, in terms of specific symbols and objects. A well-known problem with state based formalisms is the phenomenon of "state explosion" which seriously hampers the modelling of even moderately complex systems. The decomposition of states into sub states is sometimes desirable, in order to avoid this (in)famous phenomenon. However, the merits in formal tractability of these formalisms sometimes outweigh this problem. Extensions to the basic FSM formalism have been developed in order to alleviate the state explosion problem, e.g. the ESTELLE and SDL standards are based on such extensions. Languages based on finite automata or state machines exist in several variants, such as Extended FSMs, Communicating FSMs, and State Charts[Vanslembrouck and Verdonk

1991]. There are also a number of formalisms based on extensions of Petri Nets, e.g. adding time and structured objects to the basic formalism. State models are often combined with a simple form of graphical notation like Message Sequence Charts that depict messages and processes over time. The state of the modelled system is not explicit in these notations.

Whereas automata notations emphasise the notion of state, process algebras focus on the interaction of the components of a system with the state implicit. Process Algebra is the basis for formalisms that deal with the interaction and behaviour of processes in general, and stem from the work on CSP and CCS [Fekete 1993]. Thus languages for distributed systems and protocol design often include elements of process algebra (e.g. LOTOS) [Bolognesi and Brinksma 1987]. These languages typically include elements for process synchronisation in terms of signals, and sometimes also in combination with value transmission.

Semantic nets and relational formalisms dominate notations for information modelling and structuring, generally the static structures of information systems. We use the terms "semantic net" and "relational" here to emphasise the common basis for such notions. Semantic net refers to the use of graphs in the modelling of associations among concepts or objects in general, whereas relational refers to the modelling of mappings between sets of objects based on mathematical relations. Research and practice have led to the establishment of general abstraction mechanisms like classification, generalisation, grouping and aggregation, which to a varying degree are supported by current notations. The simplicity of the underlying formalism makes such languages good instruments for communication as well as for formal manipulation. Some information structuring notations also support representation of a temporal dimension as well as operations on the modelled information objects.

Recently we have seen a revival of the "linguistic" approach to model communication in organisations and between users of information systems. Some approaches to IS design are based on Speech Act theory [Flores, Graves et al. 1988], and approaches to workflow analysis and description have also been based on this theory. The speech-acts view adds a dimension of commitment and agreement to the modelling of co-ordination and communication e.g., in workflow. This is generally not emphasised in other approaches such as task or flow oriented approaches to workflow management. However, this will probably be required in Open-edi scenarios.

Rule-based or logic-based notations focus on the explicit representation of rules, e.g., business rules in the form of derivation rules or constraints[Wangler 1993]. Rule-based formalisms can promote declarative specifications and the ability for inferencing (derivation). Some notations are based on a subset of first order logic but other forms of modal logics have also been suggested for use in specifying business rules, such as deontic logics [R.J. Wieringa 1993].

The emerging object-oriented analysis and design methods generally include several notations for different modelling perspectives. Although there is no unified object-oriented notation, these methods combine notations to express

object structure, object state and events, and operations on objects. We include a brief discussion on object-oriented modelling at the end of this report.

3. 2 Example Notations

For the process of specification and design of scenarios, it is conceivable that several different notations could be used. In the review of candidate notations some general properties can be considered:

- Primary modelling perspective (e.g. process synchronisation)
- Emphasis and modelling constructs (e.g. signals)
- Theoretical or formal basis (e.g. state machines)
- Language syntaxes used in the notation (e.g. graph symbols)
- Support for automation (e.g. design tools and machine interpretation)
- Acceptance and use (e.g. de jure or de facto standard)
- Intended user (e.g. IT designer)

The following section gives an overview of notations. The overall classification of notations is based on their primary modelling perspective, and we briefly comment on the other properties where appropriate. Some notations may support several modelling perspectives. Thus an overlap between classes is possible.

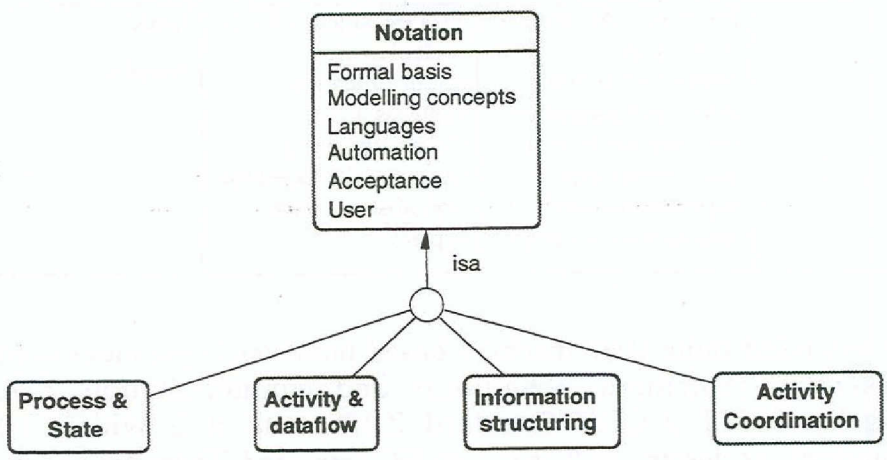

Figure 3.1: Possibly overlapping subclasses of notations

The listing of notations is obviously not complete. It is intended as a point of reference in the future development of the Open-edi model. The examplified notations come from a variety of sources, representing different aims, some are well established products, others are research prototypes.

3.2.1 Notations for Process and State Synchronisation

Of these notations [ISO/IEC 1991], SDL and ESTELLE emanate from the telecommunications field, whereas LOTOS is a more general specification language. They are generally not considered for the specification of information systems, although they in principle could be used for this.

Notation	Constructs	Basis	Language
SDL - Specification and Design Language CCITT Z.100	Processes, signals, pre-defined data types, synchronisation primitives, Interaction is supported by signal interchange, internal signals and shared data.	Extended FSM.	Graphic Lexical
ESTELLE ISO 9074	Processes, local or external signals, Pascal data types. Interaction correspond to the inputs and outputs of modules.	Extended FSM. Pascal-based	Lexical
LOTOS ISO 8807	Processes. Abstract data types. Interaction is built into synchronisation mechanism: two or more behaviour expressions may synchronise an event.	Process algebra (CCS -calculus of communicating systems, CSP - Communicating Sequential Processes) and abstract data types	Lexical Graphic

SDL is an extended finite state machine based notation. It provides constructs for representing structures, behaviours and communication. SDL was developed by CCITT (CCITT Standard Z.100) and it is widely used in telecommunications design. SDL is not an ISO standard. There are several tools based on SDL, as well as extensions, e.g. an object-oriented extension as mentioned above. SDL offers a relatively easy starting point for implementation. Maybe the weakest point of SDL seen from the Open-edi point of view is, that the contents of the messages between different processes can not be described. SDL is exemplified in the scenario description in section 4.

Estelle is a formally-defined specification language. It was designed to describe distributed or concurrent processing systems especially within the context of OSI services and protocols. An Estelle specification is a definition of a system of a hierarchically structured state machine. Communication between the modules (= state machines) defined is either asynchronous or synchronous. Estelle specifications can be prepared at different levels of abstraction. A specification can be very abstract or implementation-oriented. Implementation-oriented specifications can be derived from abstract specifications. There are several design tools based on Estelle, and compilers have been developed. Estelle has been standardised by ISO (ISO IS 9074). Because of its suitability to abstract and implementation-oriented specifications it can be used by IT designers as well as implementers.

LOTOS is a mathematically defined Formal Description Technique. Its basis is in theory based on CCS, CSP and abstract data types. Due to its mathematical basis LOTOS can be used for analysis and development of tools including simulation, compilation and testing. LOTOS permits modelling for both synchronous and asynchronous communication, and the basis of descriptions is the time sequence of processes. LOTOS can be used for producing specifications of the allowed behaviours of the system. Compared to Estelle and SDL, LOTOS is more abstract and gives more freedom for implementors. LOTOS is a design-oriented tool primarily for the use of IT designers. It is a well-established ISO Standard (ISO IS 8807).

There are obviously a number of other notations that can be used for process and state modelling. We limit ourselves to these three notations since they are well-known and standardised, and, should therefore be relevant to consider of EDI applications.

3.2.2 Notations for Information Structuring

The notations in this section are primarily used in information modelling and conceptual schema design. Most notations date back to the work of Bachman on database schema diagrams. Since then an increased understanding and consensus have been reached of the basic abstraction mechanisms for information modelling including classification, generalisation, grouping and aggregation. The common modelling constructs in most of these notations are based on entities and attributes, and relations between these. The notations can be intended for analysis or (database) design or both, where design implies the inclusion of data structuring constructs in the notation.

It is common to make a distinction between binary and n-ary notations with respect to the treatment of relations, where the latter are referred to as entity-attribute-relation (ER) notations. Chen's notation [Chen 1976] is often considered the prototype for ER-notations. Most notations are in fact binary, even some claiming kin with Chen's notation. Regardless of this, n-ary relations can be represented by entities in any of these notations. Another distinction is how strict the notations separate attributes from entities. Conceptually (e.g. in

an analysis situation) that distinction may not be necessary, but for design it may be practical. The distinction can be based on the assumption that entities have an identity (and hence an independent existence) whereas attributes have not. Or, if attributes can be considered as "values" (singles) or lexical objects, whereas entities are viewed as composite (or non-lexical) objects.

Most of the notations show small differences in graphical syntax, but they vary in their focus on analysis or design.

Notation	Constructs	Basis	Language
ER (Chen)	Entities, attributes and relations. Cardinality	Relational , semantic nets	Graphic
NIAM (RIDL) cf ISO TR9007	Entities and relations, Generalisation Distinguishes lexical from non-lexical entities.	Binary relations.	Graphic Lexical (RIDL)
STEP: EXPRESS ISO 10303 part 11	Schema, entity , relations (attributes), rules. Generalisation, Cardinality , functions, derivation rules.	Binary relations. Data structuring from various programming languages. Based on Conceptual Schema languages (ISO TR9007)	Lexical Graphic subset
IDEF-1X	Entities and relations Cardinality Subtypes	Binary relations.	Graphic
Information Engineering: IE			
OMT: Object Model	Classes (entities), relations, instances, attributes, operations/methods Generalisation, Cardinality	ER. Adapted with object-oriented concepts	
TEMPORA: ERT	Entities and relations, attributes. Derived objects. Generalisation. Time stamps	Binary relations, augmented with time properties.	

The ER-notation was originally proposed as an aid in database schema design, and has been extended in various directions since its introduction. The basic

ER notation support of n-ary relations and relationships may have attributes. Generalisation is supported by ISA relationships. Although well-known and widely used, the semantics of ISA relationships are still a source of confusion in many notations. There is extensive support for automation for ER notations.

NIAM [Nijssen and Halpin 1989] is a well-known representative of the binary approach to information modelling. This notation makes a distinction between lexical and non-lexical entities, informally corresponding to values and composite objects. RIDL is a lexical notation for NIAM in which additional rules and functions can be specified. Note that binary notations may have attributes, associated to entities as well as relationships.

EXPRESS [ISO 1992] is the information modelling notation in the STEP standard for product data representation and exchange. It is intended as a schema modelling and design language, and can in that sense be considered a semantic data modelling notation. EXPRESS is primarily defined as a lexical language, the graphic language is a subset of the full notation but includes the basic modelling constructs (entities, relations, attributes and cardinality). In addition, EXPRESS also defines a schema concept which can be used to relate a set of models as a number of independent schemata. Tool support exists, both products and research prototypes, for the design and interpretation of EXPRESS models. Figure 3.2 depicts a subset of the EXPRESS-G notation for a model in the context of our Organ-Request example.

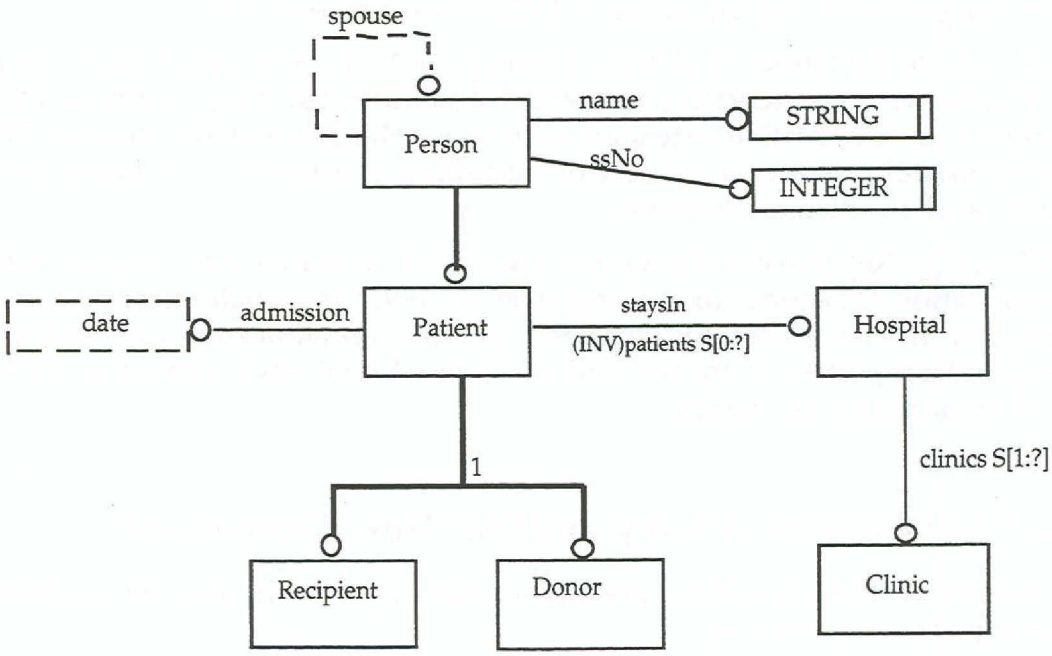

Figure 3.2: Express-G

The model examplifies generalisation in terms of subtype relations: Recipient and Donor are non-overlapping subtypes of Patient, i.e., a Patient is either Donor or Recipient of an organ. A given patient *staysIn* exactly one hospital, whereas a hospital may have zero or more patients (the *staysIn* inverse relationship). A hospital has one or many clinics. Person is a generalisation of Patient, persons have an optional *spouse* relationship. Both person and patient have some additional attributes of which *name* and *ssNo* are based on EXPRESS data types, and *admission* is based on a user defined type date. The graphical notation includes several other constructs, e.g. pertaining to the representation of data types. Some additional examples are given in section 4.

Information Engineering (IE) [Rock-Evans and Engelien 1989] represents a class of methods with a similar focus on information analysis in systems development. The information modelling notations use similar graphical language conventions, and currently they are possibly the most well-known. Originating in the CACI method, further developed and promoted by Martin & Finkelstein [Avison and Fitzgerald 1988][Connor 1985], IE as a method framework has been adapted and supported by several methods and case tools.

IDEF-1X [Rock-Evans and Engelien 1989] is the information modelling notation in the IDEF* suite of methods and techniques developed by the Integrated Computer-Aided Manufacturing (ICAM) program of the US Air force. IDEF-1X (a refinement of IDEF-1) has its origins in entity-relationship models and data models in general. IDEF-1 is supported by design tools, often in combination with the process modelling notation IDEF-0 (see next section).

We include the Object Modelling Technique (OMT) [Rumbaugh, Blaha et al. 1991] here as a representative of a recent object-oriented design method. OMT includes an ER-notation extended with some object-oriented concepts. This object model provides a rich set of modelling constructs, supporting most essential abstraction mechanisms.

ERT (Entity-Relationship-Time) is a product of research into requirements specification [Theodoulidis, Wangler et al. 1992] in which time has been included in an extended ER notation. The notation allows the association of temporal properties to entities and relations. A tool environment exists for analysis and database design.

3.2.3 Notations for Activity and Data Flow Composition

These notations focus activity precedence and dependence, and the functional decomposition of activities.

The primary use of basic dataflow notions is the structuring of processes/activities, or functional decomposition of systems. Generic constructs of dataflow notations include processes/activities , data stores ("sinks & sources") and precedence (or flow) relations. A dataflow model is (despite the term "flow") in most cases a static view of the system being modelled. It shows the

relationship between activities and data stores, in terms of directed arcs indicating a "flow" from one activity to another. Basic dataflow notations do not express sequencing or synchronisation of activities. Rather, they show the activity precedence structure in terms of what "flows" are required for a certain activity. The set of activities/processes are partially ordered. However, some notations do distinguish the flow of control from the flow of data or other physical flows. Other extensions include synchronisation operators, and explicit information flows between activities, by associating object types (or similar) with each flow relation.

The formal basis of dataflow notations depends on their extension; whereas basic dataflow notations cannot be said to have a formal basis, extensions using various synchronisation primitives can be related to state machines or Petri Nets.

Notation	Constructs	Basis	Language
SA/SD: Dataflow	Activities , data stores ("sinks & sources"), flow relations. Activity precedence and functional decomposition.	Precedence relations	Graphic
STRADIS: Dataflow			
SSADM: Dataflow			
SADT: Actigrams IDEF0	Activities, three types of flow relations. Activity decomposition.	Data & control flow	
Message Sequence Charts	Activities, messages, time.	(State Machines)	
Action Diagrams	Activities, explicit information & control flow	Precedence relations	
Design/CPN	States, transitions, typed flows (tokens) Activity decomposition	Coloured Petri Nets	

SA/SD [DeMarco 1978], STRADIS [Avison and Fitzgerald 1988] and SSADM [Downs, Clare et al. 1988] are representatives of traditional functional methods including the basic dataflow diagrams. They all share the same basic set of constructs with minor differences in graphic syntax. They do not provide a separation of data and control flow, or synchronisation primitives. Variations of

these DFDs are used in numerous CASE tools as well as in some recent object-oriented methods.

SADT was originally developed in the late 1960s as a general method for modelling complex systems. Since then, it has been widely used in information systems design for functional modelling[Floyd 1986], and lately also in more general applications of (business/production) process design[Azari 1993]. The SADT flow notation separates different types of control flows from the flow of artefacts.

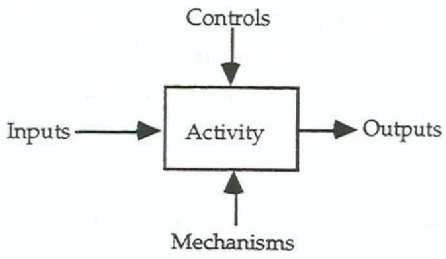

Figure 3.3: SADT activity

Inputs (e.g. information or other artefacts) are transformed into outputs in the activity, by some mechanism (e.g., a method or a human) under some form of control possibly including the outputs of other activities or any other form of control (e.g., regulations).

IDEF-0 is an adaptation of SADT included in the IDEF suite (mentioned above). Several tools exist that support SADT/IDEF [MetaSoftware 1993]. Figure 3.4 is a simple IDEF0 model of some activities of the Organ-request scenario, where the roles are depicted as mechanisms.

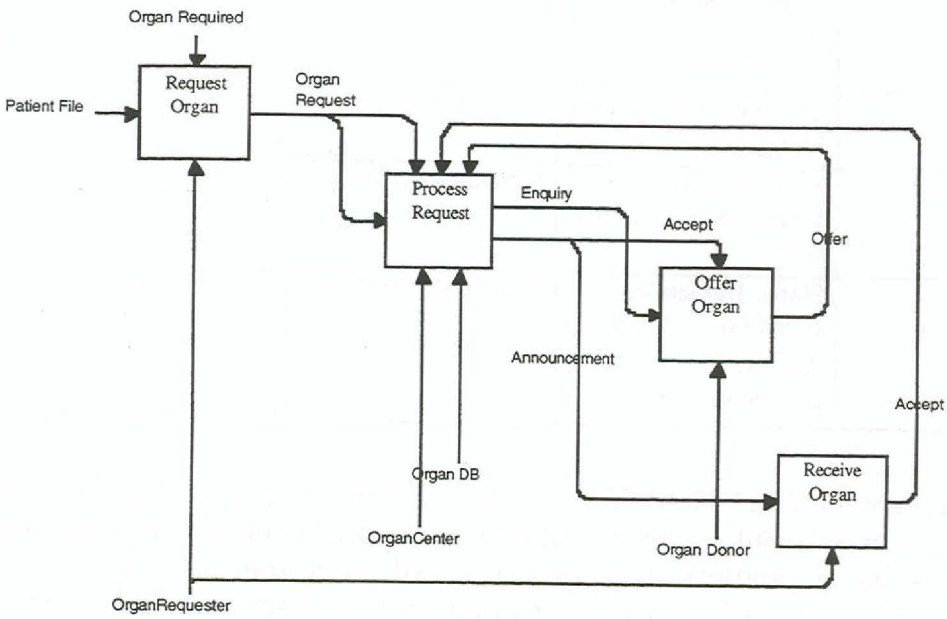

Figure 3.4: IDEF0 model

We include message sequence [Vanslembrouck and Verdonk 1991] charts here simply because they are a very simple and commonly used form to describe how messages (data, signals, etc) are sequenced between a set of activities (processes, agents, etc) over time.

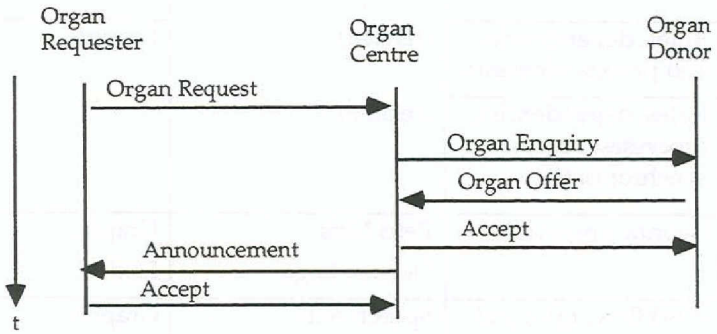

Figure 3.5: Message sequence chart

Design/CPN [MetaSoftware 1992]is a tool that provides a process design and simulation environment based on Coloured Petri Nets (see next section). The notation provides decomposition of activities and the modelling of states in terms of a flow of artefacts.

Action Diagrams [Goldkuhl 1993] are a further development of the ISAC method. It is basically a combined data and control flow notation, but it also introduces the concept of organisational context for processes and flows.

Most object-oriented methods include variations of dataflow notations, e.g. the OMT functional model. They typically relate a set of activities to an object type (entity). Message sequence charts are also used in some object-oriented methods showing message interactions between objects.

3.2.4 Notations for Activity Co-ordination

Activity co-ordination models should ideally integrate modelling constructs for role description, process synchronisation and interaction, as well as a representation of the commitments and obligations attributable to users.

We exemplify with some notations that tries to integrate some of these aspects. The difference between notations for activity co-ordination and those for activity and dataflow composition lies in their origin and intended use.

Notation	Constructs	Basis	Language
Coloured Petri Nets	States, transitions, tokens and typing (colouring)	Petri Nets	Graphic Lexical
Role Interaction Nets (RINs)	Roles, dependencies and process elements.	(related)	Graphic
Role Activity Diagrams (RADs)	Roles, dependencies, processes, process synchronisation	(related)	
Deontic Rule Language	Deontic operators	Petri Nets Deontic Logic	Graphic Lexical
ActionWorkflow: Business Design Language	Workflow, roles and acts. Customer-performer relations	Speech Acts	Graphic
ORDIT : Enterprise & Interaction Diagrams	Organisational structures, role relationships Role interactions in terms of functions and commitments.	ER-like Data/control flow Speech acts	

The Petri Net formalism [Jensen 1988] can be considered the formal basis for several notations where co-ordination is important. The basic Petri Net notation is very simple and powerful, for most practical applications it has been extended and complemented in various directions. We include Coloured Petri Nets (CPNs) here as it is a commonly used extension to Petri Nets. CPNs could very well be considered as a data and control flow formalism with high expressive power. These formalisms are well suited to model concurrency and to detect conflicts in systems that include interacting processes. CPNs (and Petri Nets in general) are therefore often used in simulation tools.

Role Interaction Nets (RINs) [Curtis, Kellner et al. 1992][Singh and Rein 1992][1] is a notation originally developed for software process co-ordination. It is similar to basic dataflow notations and message sequence charts and can be related to Petri Nets.

---

1 Reference not available at time of writing

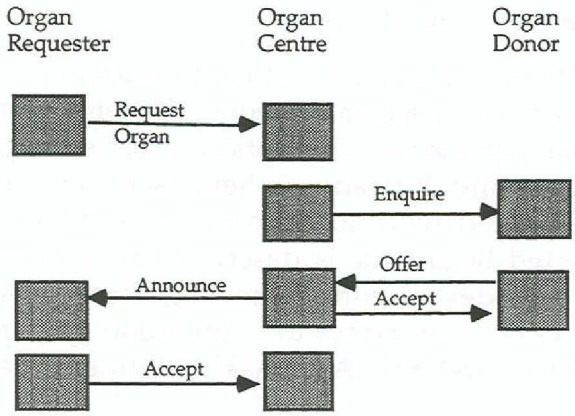

Figure 3.6: RINs

There are other graphical notations for role interaction based on RINs, such as Role Activity Diagrams (RADs) [Kontio 1994]. This notation adds process logic to role descriptions, providing synchronisation of activities within and in between roles.

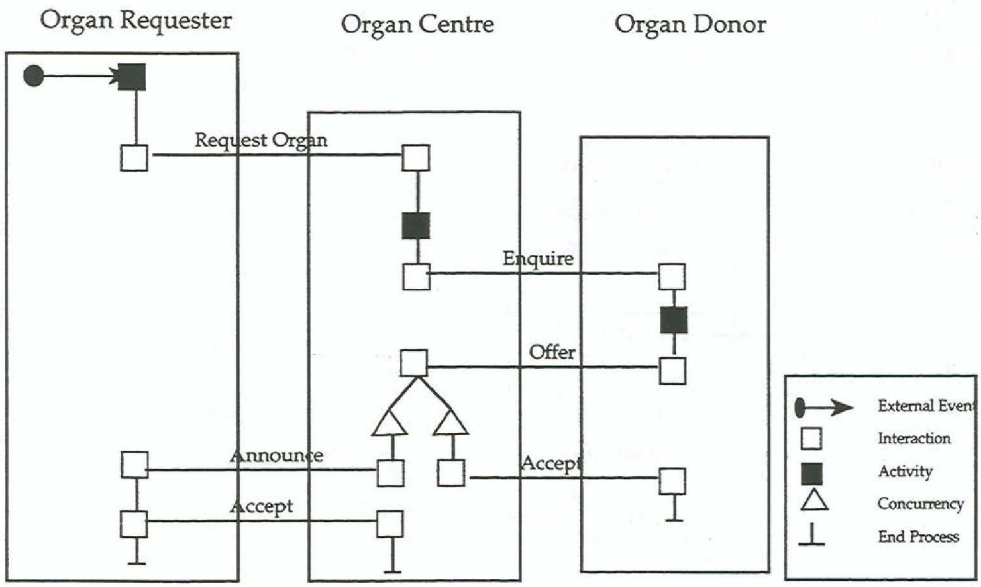

Figure 3.7: RADs

Deontic logic has been applied to information systems to model such aspects as contracting and authorisation. As such, it is not a co-ordination notation in the sense of, e.g. Petri Nets or similar. The reason for introducing this formalism in the Open-edi context is its potential use in specifying commitment and obligations, etc., on the basis of formal logic. This is an essential aspect of

scenarios, not covered by conventional notations. The notation referred to here [Lee 1988] combines deontic logic operators with Petri Nets.

Action Workflow [Medina-Mora, Winograd et al. 1992] is an example of a workflow product that is partly based on speech-act theory. It stems from previous work on the language/action approach to information systems design [Flores, Graves et al. 1988]. A simple graphical notation is here used to model business processes as a set of interrelated workflow loops. For each such workflow loop, the roles and expected behaviour is described by a client-performer relationship. The product provides support for creating client-server based applications from the models. Other approaches to information systems modelling based on speech-acts include SAMPO [Auramäki, Lehtinen et al. 1988].

ORDIT [Blyth, Chudge et al. 1993] is a method representing research in the area of enterprise modelling , intended for analysis of the goals, structures and roles in an organisation. Its graphical notation is intended to give a unified representation of control flow, information and of the obligations and commitments between roles. The method includes an enterprise modelling language and a role reference model. The Enterprise Diagrams (Figure 3.8) are notations in ORDIT which are intended to provide a clear separation of behavioural and structural properties of an organisational system in terms of roles. This represents another facet of role modelling compared to notations like RINs and RADs, focusing on obligations and expected behaviour of roles, and not primarily on process logic or interaction sequences.

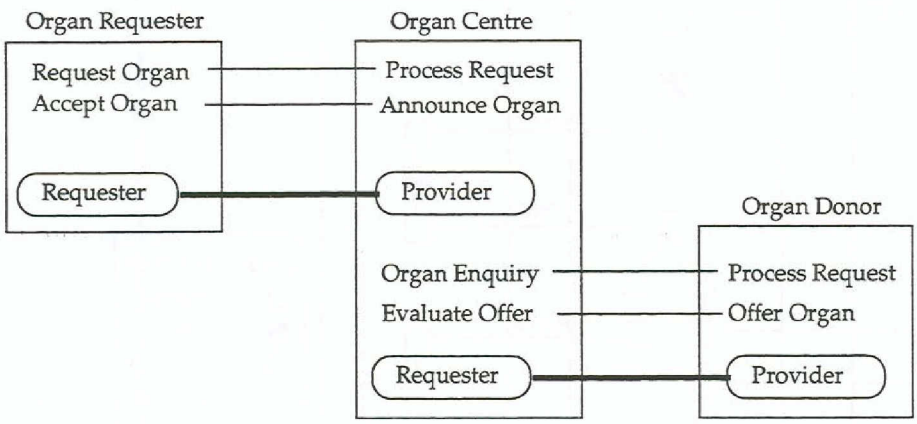

Figure 3.8: Ordit Enterprise Diagram

For role dependencies this notation distinguishes between structural relationships (Requester-provider), representing obligations, and functional relationships (Request Organ - Process Request), representing behaviour.

3.2.5 Notations for Interchange Formats

There is a wide range of notations intended for interchange formats. These should not primarily be regarded as modelling formalisms as discussed above, and they are not intended as such. We include a brief discussion on some interchange formats in order to relate them to the Open-edi reference model. Most of these notations also contain abstraction mechanisms similar to other notations, and could in principle be used for (limited) information analysis and modelling. However, the primary objectives are to define the structure and format of data to be exchanged between applications. This often implies inclusion of syntax elements or structures with a specific meaning to the transmission process (such as message sequencing, coding and addressing). We give example of three classes of inter-change formats which represent notations quite different in aims and scope.

Message structure and format:

ASN.1 is a language syntax suitable for defining message structures. It is primarily a data structuring language with an emphasis on data type definition. In addition to the syntax, ASN.1 also includes basic encoding rules (BER) for generating a transmission format, although other transmission formats are possible to use with ASN.1.

Document interchange:

SGML (ISO 8879) is used to define document mark-up languages. There are in principle two approaches to document coding; procedural coding focusing the layout and descriptive coding focusing the structural contents. Mark-up languages are based on the latter approach. Such a language structures documents into different elements (title, authors, section, paragraph, etc) using pre-defined mark-up identifiers, irrespective of document contents. SGML is used to define the grammar of languages to describe classes of documents (or types), in this sense it is a grammar representation standard. A mark-up language like SGML is of general applicability, and is also a tool for structuring document data in archiving applications. Various tools and products include SGML as a component, e.g. the hyper media standard HyTime (ISO 10744), based on SGML, and also a component in frameworks like CALS.

Another example in this class is the ODA (Open Document Architecture) (ISO 8613) for document structuring and interchange. The objective is to support the interchange of documents including logical structure, the layout and presentation as well as contents. ODA specifies an interchange format called ODIF (Open Document Interchange Format), which is an ASN.1 encoding for the transfer of documents between applications. ODA also includes the ODL (Open Document Language) which is an application of SGML defining a mark up language for ODA.

Structured data interchange:

Another class of interchange formats can be referred to as structured data interchange[1], which emphasises the grouping and sequencing of pre-defined (or standardised) data elements. An example here is the EDIFACT standard and its continuously increasing number of generic message types (UNSM - UN Standard Messages). Subsets of these general and complex types are formed for use by specific business sectors. The syntax of EDIFACT in combination with these principles for message design, makes it unsuitable for modelling and representation in general, and in particular for information parcels in scenarios. Efforts are, however, being made to promote the use of conceptual (information) modelling techniques to design EDI message types. The EDIFACT Business and Information Modelling approach (BIM) advocates the use of a shared message repository accessible by different design tools but independent from specific methods and models. A meta model for such a repository has been proposed [Johansen 1993] using an IE based entity-relation notation.

Numerous other notations and syntaxes exist that can qualify as interchange formats. A comprehensive evaluation of interchange formats for health-care applications has been carried out by CEN [CEN/TC251 1992]. Interchange formats are needed in Open-edi standards but should be considered part of the support services. In this sense they belong to the Functional Service View of the Open-edi reference model and are not part of scenario descriptions in the Business Operational View. Information parcels should be mapped to the appropriate interchange formats, and it is conceivable that alternative interchange formats can be used in the same scenario description.

---

[1] Intuitively "EDI" would be a more appropriate label. However, EDI should be considered as an application domain including both Open EDI and interchange formats such as the EDIFACT standard and its subsets.

4. A Scenario Description

In this section we give an example of an Open-edi scenario description. The Organ-Request example was briefly introduced in chapter 2.5, and is further elaborated here. The proposed structure of a scenario description is the following:

1. Scenario objectives
2. Scenario attributes
2.1. Rules and Regulations
2.2. Role Relationship Model
2.2. Scenario Information Model
3. Information Parcel Model
4. Role Model
5. Usage Case

We provide descriptions of different aspects of the scenario using some of the previously mentioned notations, such as SDL, EXPRESS and message sequence charts, in combination with natural language explanations. The choice of these notations is quite arbitrary, other notations may provide a higher expressive power. We do not consider the description to be complete, nor do we explain all details. The scenario description defines certain relationships between the various models in the different sections. However, it should not be seen as a method for scenario definition, merely as a way of structuring the models. The description includes both the models of the actual scenario example as well as general comments on suggested contents of each section.

4.1 Scenario Objectives

The application domain of this scenario is the health-care sector. It is intended to support the management of information related to organ request and supply. See section 2.5 for the introductory description.

Candidate notations: Depending on the importance of the objectives, e.g. their explanatory value, a more formal description could be used, e.g. an information modelling notation where entities represent objectives or goals.

4.2 Scenario Attributes

4.2.1 Rules and Regulations

This section should include references to any national/international laws or regulations pertaining to the scenario. Any commitments or obligations attributable to the participants of the scenario could also be specified here.

Candidate notations: Co-ordination notations ,e.g. based on speech acts or deontic logic, could be interesting to investigate for this aspect asa scenario.

4.2.2 Role Relationship Model

The Role Relationship Model provides an overview of role relationships, without considering internal behaviour of individual roles. It should also define any restrictions or constraints imposed on the roles. When defining a scenario, we provide a description, i.e. a generic and reusable definition that can be instantiated by business participants assuming the roles of the scenario. Thus, at any one time a number of scenario instances may be active or executing, with each role bound to some business participant. The participants of this scenario could be different hospitals. Three roles have been identified.

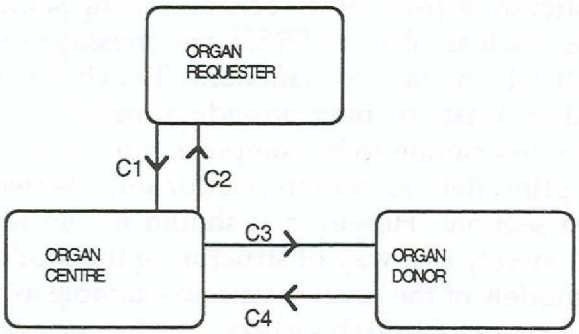

Figure 4.1: Role Relationships

The description in Fig. 4.1 is based on the block diagram of SDL. Each such block will subsequently be detailed in the Role Model (sec. 4.4). Based on the role relationships, a first set of information requirements can be identified:

C1 = | Organ Request: a request for an organ

C2 = | Organ Availability Announcement: information on the availability of an organ

C3 = | Organ Enquiry: an enquiry for an organ from a donor

C4 = | Offer: information on organs that can be offered

These information requirements will be candidate information parcels.

Constraints and restrictions include the binding of participants to the roles, and the instantiation of roles. For an instance of this scenario only one participant is allowed for each role. We express this in a simple information model (Fig. 4.2).

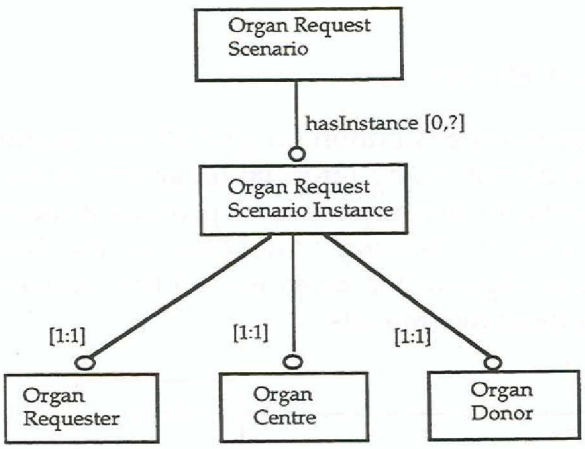

Figure 4.2: Role instantiation constraints

A further constraint we may want to impose on this scenario is that the same hospital (i.e. participant) should not be able to act in multiple roles in one instance of the scenario. For example, a hospital should not be able to act as an organ requester and organ centre simultaneously. We call this a role binding constraint, represented in Fig.4.3.

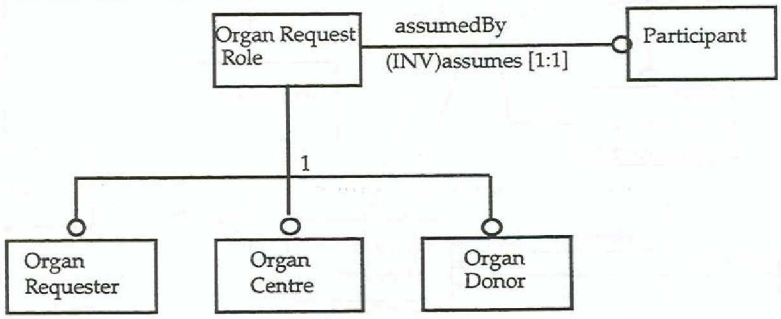

Figure 4.3: Role binding constraints

The three roles are specialisations of a general Organ Request Role entity. The relationship from the Organ Request Role entity to the Participant entity implies that a single role may be assumed by ("played") exactly one Participant, which also applies to the inverse relationship (INV).

Note, however, that a hospital may assume different roles in different instances of the scenario, i.e. being a requester in one instance of the scenario, and

possibly act as a donor in another. Several instances of the scenario may of course be active simultaneously.

Other candidate notations: Various process modelling notations that elaborate the role concept such as RINs, requirements modelling notations similar to the Enterprise diagrams in ORDIT.

4.2.3. Scenario Information Model

The Scenario Information Model represents a common (or global) information model for the roles in the scenario. It describes any information entity relevant to the application domain of the scenario. Its purpose is to provide a model from which information parcels can be derived. All roles in the Role Relationship Model are required to be represented as entities in the Scenario Information Model, thereby relating these two models.

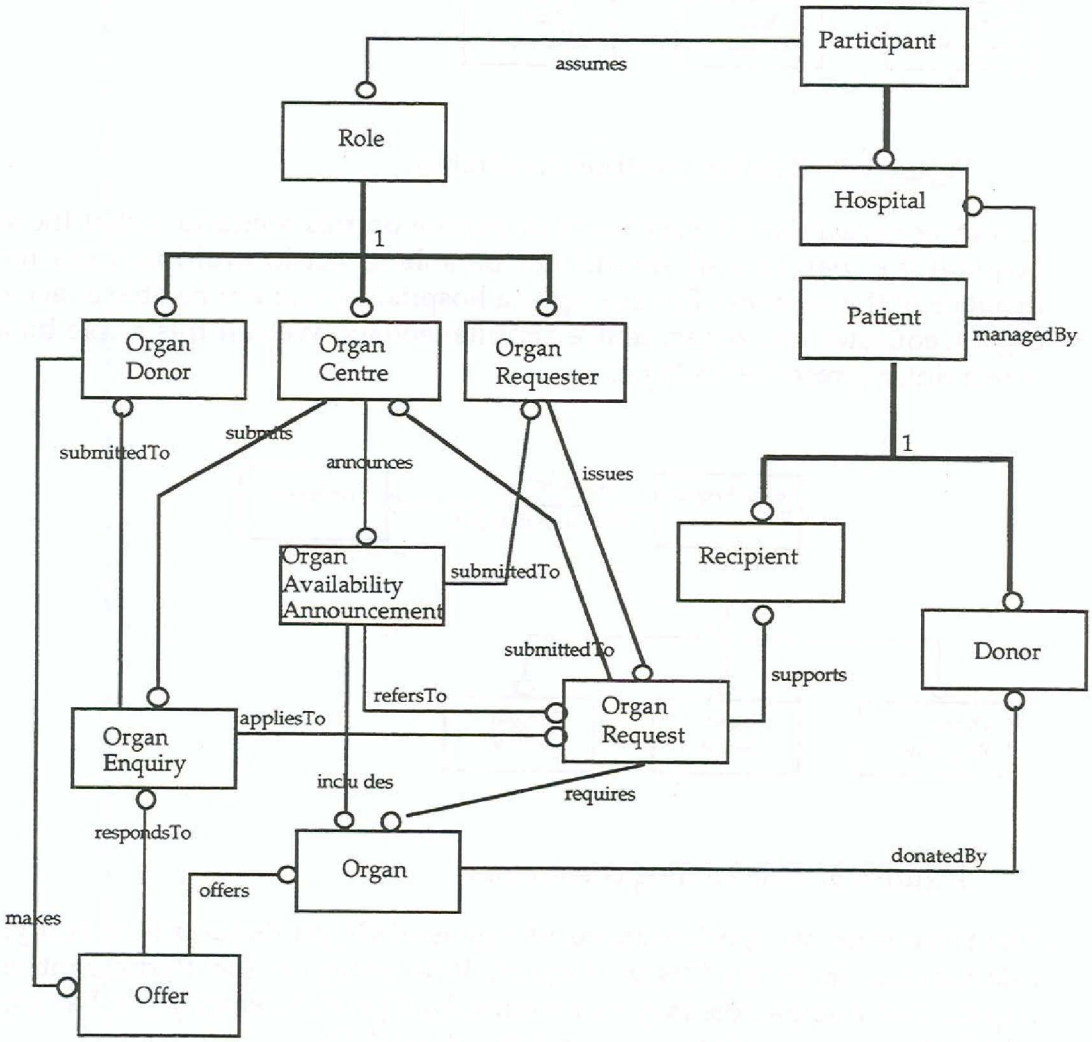

Figure 4.4: Scenario Information Model

The roles of the scenario are represented as specialisations of the general Role entity. The specialisation of the Participant entity implies that the users of this scenario are hospitals. Patients are included in the model, in terms of two non-overlapping subtypes (Recipient and Donor). A Patient cannot donate organs to himself. The Organ Requester entity issues an Organ Request, which is submitted to the Organ Centre. The Organ Centre provides an Organ Availability Announcement,. This may be dependent on an Organ Enquiry made to an Organ Donor. This dynamic aspect is not expressed in this model, it has to be modelled in terms of role behaviour.

This model only includes the most significant relationships between entities. The model should be seen as a conceptual model defining the context of the scenario, it does not prescribe a specific design.

Other candidate notations: Any information modelling notations focused on analysis.

4.3 Information Parcel Model

This section of the scenario description defines the set of information parcels used by all roles. These are described by an Information Parcel Model based on the Scenario Information Model previously defined.

Based on the relationships in the Role Relationship Model and the corresponding entity definitions in the Scenario Information Model, we select a number of views of the latter. These views include the central entities that are candidates for information parcel definitions. Each view should include at least two role entities corresponding to the role relationship in the Role Relationship Model, in this sense information parcels are always defined as relationships among roles.

Several views may thus be defined on the basis of role relationships, each one may result in one or more information parcel definitions. However, all information parcels will not have direct correspondence to entities in these views. We give an example of one such view with the Organ Request as the central entity and suggested information parcel (Figure 4.5).

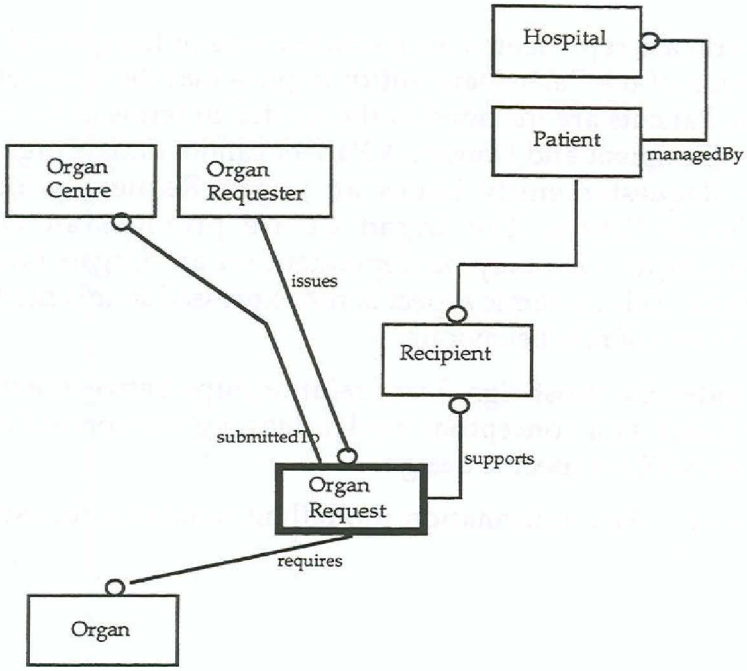

Figure 4.5: Information Parcel View of the Scenario Information Model for Organ Request.

Each view of the entities in the Scenario Information Model is now used as a basis for defining the information parcels. The total set of information parcel descriptions make up the Information Parcel Model, in which each information parcel should be directly or indirectly related. This model is clearly more focused on design than the previous one. Although it is not to be seen as a "message design", we are still dealing with specification information for the scenario, albeit on a more specific level.

Additional attributes may now be defined for the entities. In Figure 4.6 only the Organ Request entity is detailed in this way, although in general all entities in a view should be specified to a sufficient level of detail. Any inverse relationships to other entities should be made explicit, e.g. the *issues* relationship from Organ Requester to Organ Request is represented by its inverse *issuedBy*. The reason for this is to preserve the Organ Request entity as the defining (or central) entity for the information parcel. Some relationships in the information parcel view may have their identifiers changed in order to better correspond to attribute names. No additional entity relationships should however be introduced, so as to preserve the structural correspondence with the Scenario Information Model.

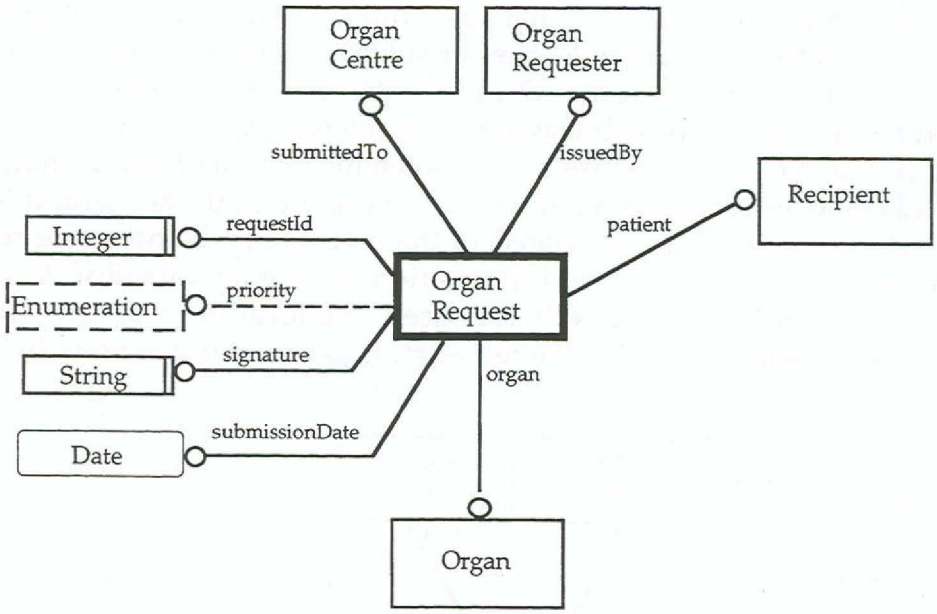

Figure 4.6: The Information Parcel Model including the single information parcel Organ Request.

The additional attributes for Organ Request are based on standard data types of this particular notation. An optional attribute (*priority*) has been added to represent possible priority classes for a request, a user defined enumeration data type is used to represent this attribute. The *signature* attribute could be intended for digital signatures in a request.

Although the Information Parcel Model is a description of a static aspect of the scenario, i.e. the structure of information exchanged, its definition requires an analysis of the dynamic aspect of the scenario. Thus, descriptions of the external role behaviour may supplement the modelling of information parcels (the internal behaviour is, however, left for the Role Model). We exemplify with a message sequence chart showing one part of the role interaction. Such descriptions will be the starting point for definition of the Role Model.

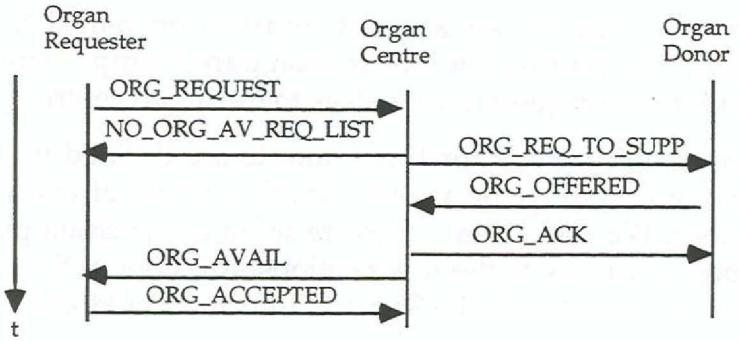

Figure 4.7: Information Parcel sequence chart

This simple chart does not show the complete role behaviour, just a specific sequence of information parcel exchanges. Thus several such descriptions may need to be provided. The Organ Requester submits an organ request. There are no locally available organs at the Organ Centre, hence the requester must be informed about this and possibly assigned to the waiting list, with the request pending (while the Organ Centre makes an enquiry at an Organ Donor). This introduces a new information requirement not present in the Scenario Information Model, although it can be related to the entities representing the request and the organ centre. A new information parcel is introduced in the Information Parcel Model (Fig. 4.8). It is used to inform the requester that the organ was not available and that the requester is assigned to a waiting list.

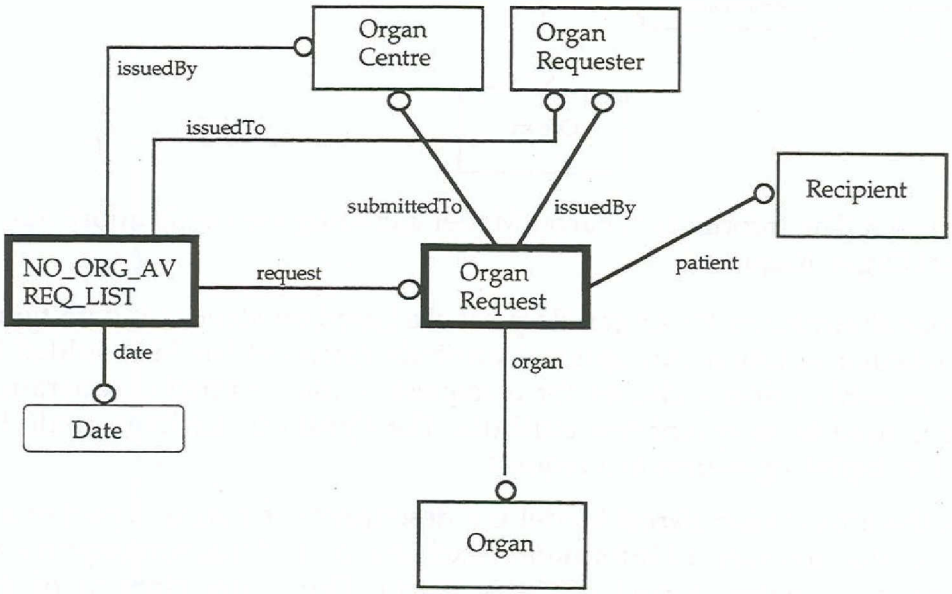

Figure 4.8: Information parcel for waiting list assignment added

Some of the attributes of the Organ Request information parcel have been suppressed for clarity. The waiting list information parcel simply refers to the request, and indirectly to the requester, as well as to the organ centre.

The additional information parcels for this example are defined in the same way, some as result of detailing the views, others a result of analysing the external role behaviour. We now list a complete set of information parcels for this example, grouped according to the role relationships. Four of these (*) have a direct correspondence to entities in the Scenario Information Model.

C1 =	ORG_REQUEST	(Organ Request) *
	ORG_REQ_REM	(Organ Request Reminder)

	ORG_ACCEPTED	(Organ Accepted)
	STOP_REQ	(Cancel a pending organ request)
C2 =	ORG_AVAIL	(Organ Availability Announcement) *
	NO_ORG_AV_REQ_LIST	(Waiting list assignment)
	RE_REQ_ACK	(Acknowledgement of a request reminder)
C3 =	ORG_REQ_TO_SUPP	(Organ Enquiry) *
	ORG_ACK	(Acknowledgement of offered organ)
	STOP_REQ	
C4 =	ORG_OFFERED	(Offer) *

Following the EXPRESS notation, the Information Parcel Model could correspond to a Schema.

Other candidate notations: In principle any information modelling notation, combined with data flow and activity composition notations with explicit separation of control flow.

4.4 Role Model

The Role Model provides a detailed description of each role. It should clearly identify the information parcels that are the interaction points between each pair of roles. The Role model is developed based on the Information Parcels Model.

SDL graphs are used in our example to describe each role. SDL is well-suited to all systems whose behaviour can be effectively modelled by extended finite-state machines and where the focus is placed on interaction aspects. The correspondence between scenario concepts and SDL modelling constructs is given below:

Scenario	**SDL**
Role	Process
Information Parcel	Signal
Request Information Parcel	Receive Signal
Submit Information Parcel	Send Signal
Internal Functions	Task
State	State

In addition we will in our example model an Internal Event as an Internal Signal, i.e. a Signal received not from the external world to the role, but within the role itself. Note the importance of separating internal behaviour of the role from its external behaviour.

Each role from the Role Relationship Model will be described in a separate Role Model using SDL graphs (Figure 4.9).

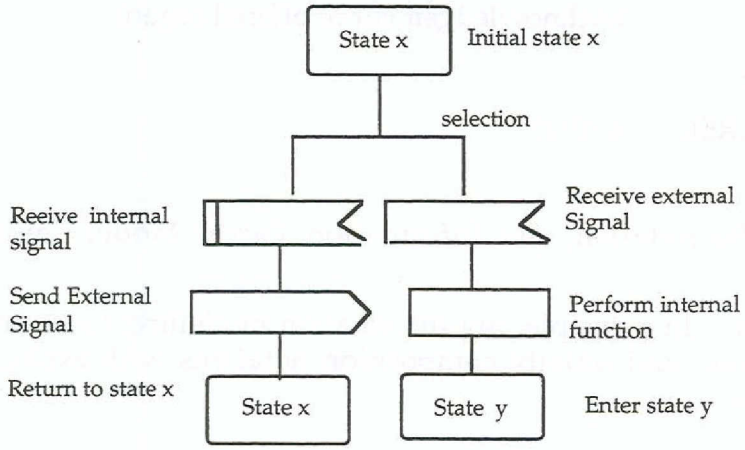

Figure 4.9: SDL graph legend

The Organ Requester role (Figure 4.10) will typically be played by a hospital within an instance of this scenario. The Organ Requester is the initiator of the Organ-Request Scenario. The need for an organ for some patient drives the Organ Requester to request an organ from the Organ Centre.

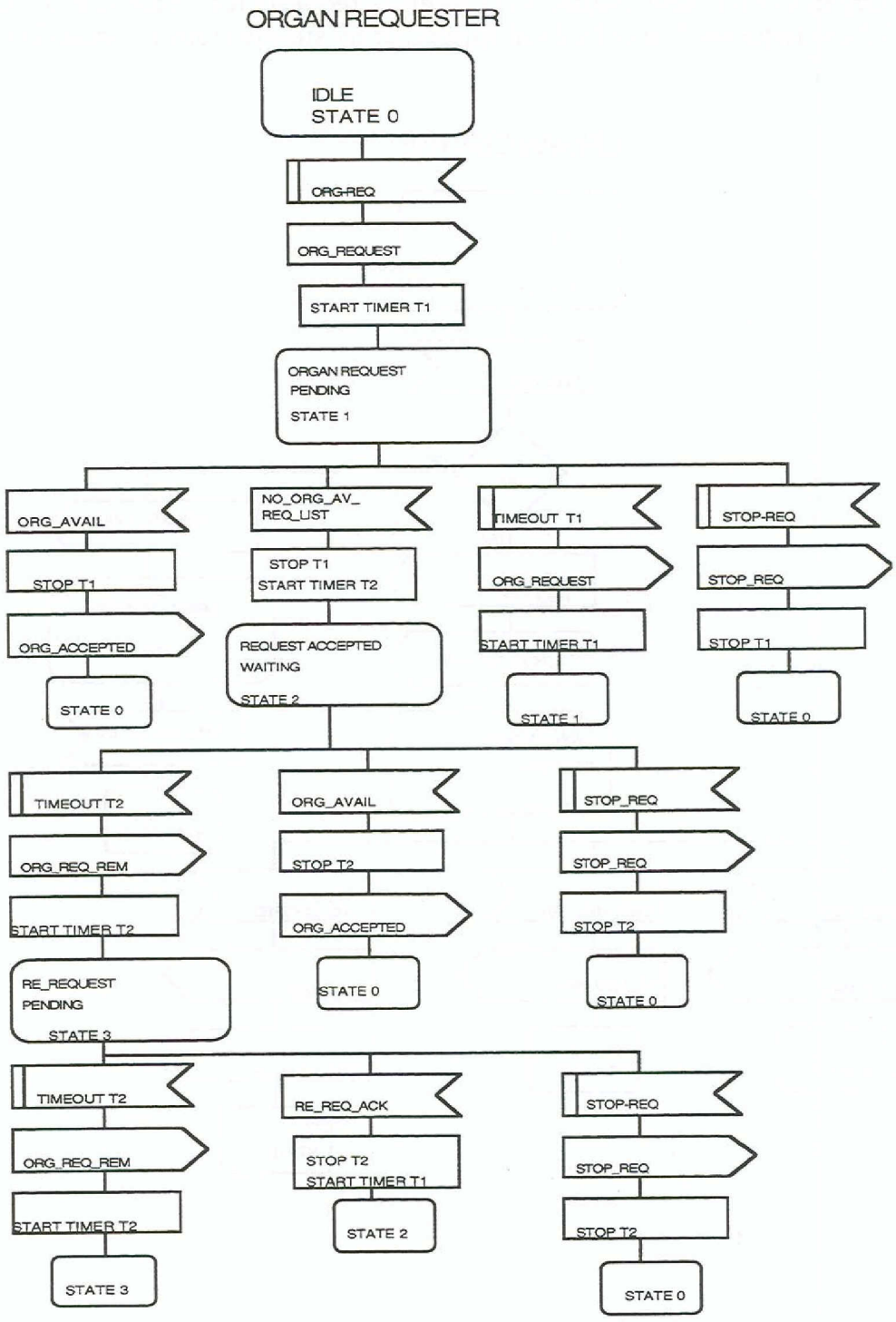

Figure 4.10: The Organ Requester Role

The Organ Centre Role (Figure 4.11) receives requests for Organs, and tries to fulfil these requests. Either by local supply, or by supply from an organ Donor.

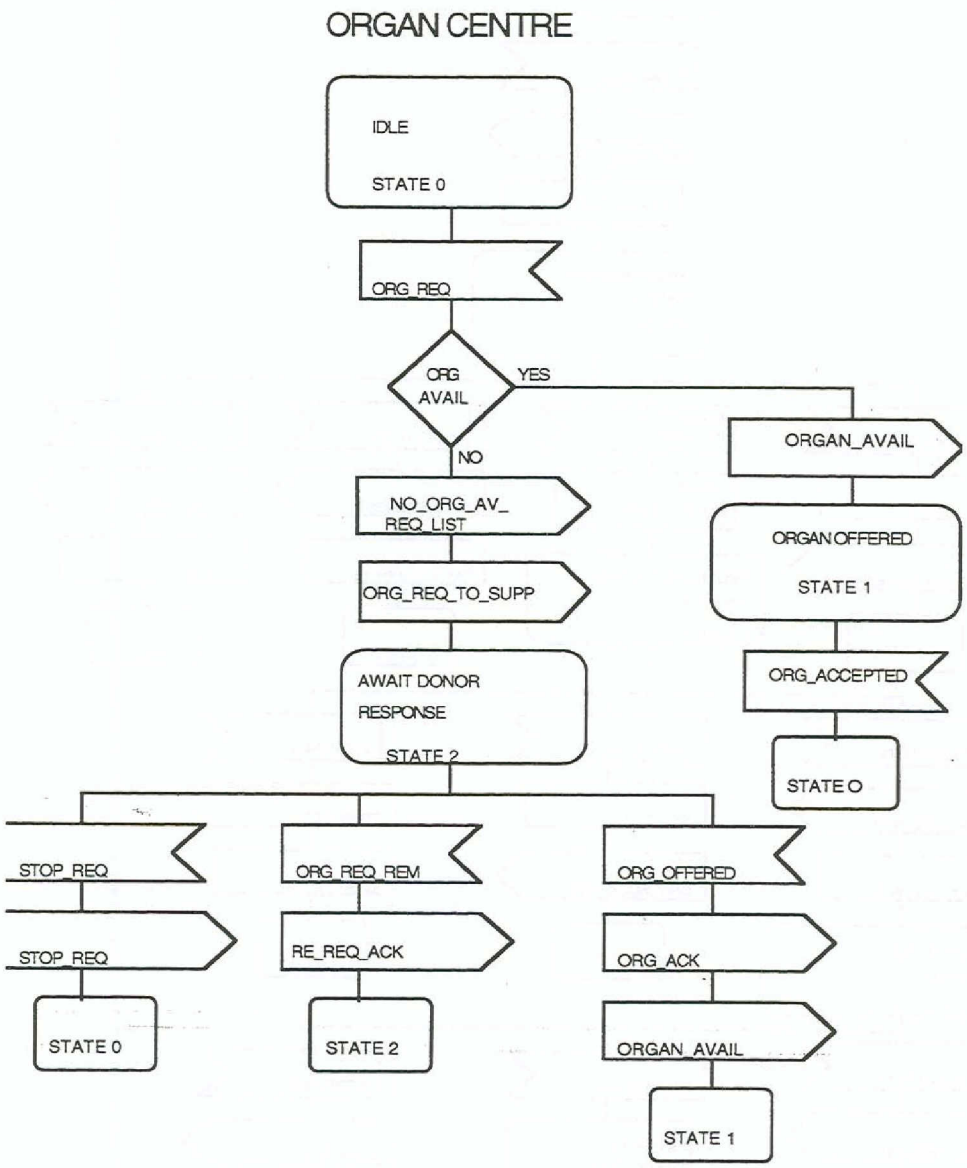

Figure 4.11: The Organ Centre Role

The Organ Donor Role (Figure 4.12) receives requests for Organs and supplies the requester (Organ Centre) with organs if available.

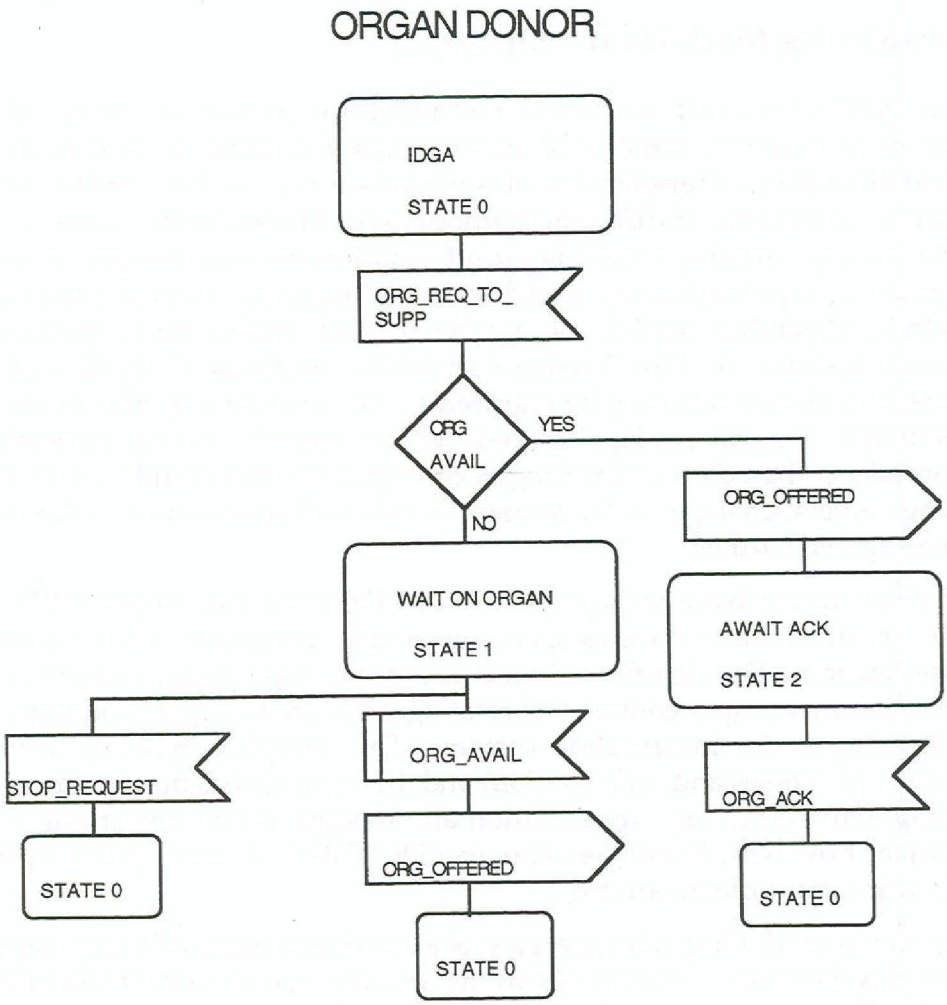

Figure 4.12: Organ Donor Role

Other candidate notations: Other notations based on Petri Nets such as Coloured Petri Nets, or other state based notations.

4.5 Usage Case

The aim of this final section of the scenario description is to provide an instantiated description of the scenario defined. The objective is to provide walk-throughs of the different models. Ideally, this could be supported by simulations depending on the notations used and the tool support available. We do not elaborate this further in this report.

This concludes the Organ-Request Scenario description.

5. Considering Methods for Open-edi

We have previously not discussed methodology in relation to notations, although most notations mentioned above are part of some method or at least have guidelines for their application. It is important to consider whether Open-edi requires additional or different methodological issues to be focused, than those covered by existing methods for information systems design. It should be stressed that Open-edi is concerned with inter-organisational applications. It focuses the co-operation model, and not the internal organisation of participant information systems or how "business should be performed". In view of this, designers of such systems may be required to consider aspects like autonomy, responsibilities, contracting, legal procedures and security. We have mentioned some formalisms based on deontic logics or linguistics that could be a basis for modelling some such aspects. However, the practical application of this needs to be investigated further.

Method frameworks have been proposed for EDI [Peric and Simon 1991] that prescribe an information systems specification and design process for message based applications. The significant aspect covered is that message structure and design is put in its proper context, by relating it to conventional modelling and design techniques for information systems. The interplay between different models (e.g., processes and information) and different abstraction levels as well as development stages (e.g. specification and design) is one important aspect here, which, however, is not peculiar to EDI. This is a weak point in most methods using multiple notations.

With respect to modelling, the emerging object-oriented methods may improve the interplay between different notations and the use of uniform modelling constructs. Object-oriented analysis and design methods do in general include notations for both static, dynamic and functional aspects of an information system. There is currently no unified object-oriented notation for analysis and design, although the object-oriented concepts are fairly uniform and most methods employ a similar set of notations based on a combination of the notations described above. Existing methods are also similar in their focus on design. They do, however, introduce new modelling concepts like the object life cycle, and the principle of locality and modularity (c.f. encapsulation). Both these aspects are used to model object behaviour, different from other forms of conceptual modelling of behaviour. The dynamics in terms of individual object state changes, are typically based on state transition diagrams and state charts. As a complement, message sequence charts are used to model interactions between a set of objects. Functional and computational aspects are modelled separately, including the use of dataflow diagrams or flow charts. Another aspect is the support for object layering and clustering, as a way of reducing the complexity of models.

Not surprisingly, the most well developed modelling perspective in object-oriented methods is the information (or object) structuring, describing the

static aspects of a system. Typically, conventional ER notations are enhanced with object-oriented concepts. There is an evident shift of emphasis in object-oriented methods compared to traditional methods (e.g. SA/SD). Although they generally support the separate views of an information system in terms of information, dynamics and function; object-oriented methods focus the analysis (and design) around the object (information) model, whereas many traditional methods focus functional decomposition. This is obviously reflected in the notations.

The choice of object-oriented modelling and design in a method for Open-edi can also be motivated by the availability of de facto or de jure standard architectures for interoperability like CORBA [OMG 1991].

Regardless of the choice of modelling and design approach, the Open-edi reference model should not define a modelling process per se. It identifies those aspects of a scenario (i.e. different concepts and components and their relationships) that must be defined in order to arrive at a complete scenario specification and the required support services. For each aspect of a scenario the reference model could recommend suitable methods and description techniques. These methods and techniques will support the actual modelling process. In general, an Open-edi method should at least support these four phases:

- analysis of information requirements -> information parcels
- identification of agents, responsibilities, commitments etc. -> roles
- analysis of behaviour and co-ordination -> role states and synchronisation
- analysis of rules, regulations or restrictions -> scenario attributes

The elaboration of these phases and their dependencies could be the starting point for an Open-edi methodology.

6. Concluding Remarks

In this report we have dealt with the basic properties of Open-edi scenarios, mainly using conventional notations and models. One objective being to place Open-edi in the context of information and communications systems modelling. A careful combination of existing notations and modelling approaches will probably suffice to model the basic aspects of scenarios. We have, however, indicated the possibilities of investigating other formalisms for analysis and representation of commitments, obligations and other constraints that may be attributable to the user roles of scenarios. An interesting aspect of the Open-edi reference model is that it may bring together different modelling and design approaches from various application domains. However, the Open-edi model in general and the scenario concept in particular, need to be further refined and strictly defined to enable a proper choice of notations. Regardless of the choice

of modelling notations, scenario design will also require a methods framework. Further, the potential acceptance and use of the type standards that will result from this model, should be investigated more deeply as a part of the future development.

7. References

Auramäki, E., E. Lehtinen and K. Lyytinen, (1988), *A Speech-Act Based Office Modelling Approach*, ACM TOIS, 6, (2). 126 - 152.

Avison, D. E. and G. Fitzgerald, (1988), *Information Systems Development: Methodologies, Techniques and Tools*, Blackwell Scientific Publications, Oxford.

Azari, M., (1993), *Manufacturing Systems Engineering*, Doctoral Thesis, 1993, The Royal Institute of Technology, Dept of Manufacturing Systems.

Blyth, A. J. C., J. Chudge, J. E. Dobson and M. R. Strens, (1993), *ORDIT: A New Methodology to Assist in the Process of Eliciting and Modelling Organisational Requirements*, Conference on Organizational Computing Systems, Milpitas, California, ACM Press.

Bolognesi, T. and E. Brinksma, (1987), *ISO Specification Language LOTOS*, Computer Networks and ISDN Systems, 14, 25-59.

CEN/TC251, (1992), *PT004 Syntax: Investigation of Syntaxes for Existing Interchange Formats to be used in Healthcare*, Draft Technical Report v 1.1, N318, November 1992, CEN TC251 PT004.

Chen, P. P., (1976), *The Entity-Relationship Model –Toward a Unified View of Data*, ACM Transactions on Database Systems, 1, (1). 9-36.

Connor, D., (1985), *Information system specification & design road map*, Prentice Hall.

Curtis, B., M. I. Kellner and J. Over, (1992), *Process Modelling*, CACM, 35, (9). 75-90.

DeMarco, T., (1978), *Structured Analysis and System Specification*, Yourdon Inc. New York.

Downs, E., P. Clare and I. Coe, (1988), *Structured Systems Analysis and Design Method –Application and Context*, Prentice Hall.

Fekete, A., (1993), *Formal Models of Communication Services: A Case Study*, IEEE Computer, 1993, (August). 37-47.

ISO, (1992), *Industrial Automation Systems – Product data representation and exchange - Part 11 : The EXPRESS Language reference Manual*, DIS, 10303-11, 1992, ISO/TC 184/SC 4.

ISO/IEC, (1991), *Guidelines for the application of Estelle, LOTOS and SDL.*, Technical Report, TR 10167, 1991-11-15, ISO/IEC.

ISO/IEC/JTC1/WG3, (1994), *Open-edi Reference Model Standard –Working Draft*, Draft, N255, 1994, ISO.

Jensen, K., (1988), *Coloured Petri Nets – Basic Concepts, Analysis Methods and Practical Use*, Aarhus University, Denmark.

Johansen, T., (1993), *EDIFACT Information Modelling. Message Repository Meta Model*, 40-RA93025, March 1993, SINTEF Delab.

Kontio, J., (1994), *Comparison of Process Modelling Notations for VITAL*, ESPRIT 2 Project Report, Nokia/T113/2, NOKIA Research Center/ESPRIT 2 project ITAL.

Lee, R. M., (1988), *Bureaucracies as Deontic Systems*, ACM Tr. OIS, 6, (2). 87-108.

Loucopoulos, P. and R. Zicari, (1992), Conceptual Modelling, Databases, and CASE – An Integrated View of Information Systems Development, John Wiley & Sons.

Medina-Mora, R., T. Winograd, R. Flores and F. Flores, (1992), *The Action Workflow Approach to Workflow Management Technology*, CSCW 92,

MetaSoftware, (1992), *Design/CPN MAC Version 1.9.1*, Release Notes, May 1992, Meta Software Corporation, Cambridge, MA, USA.

MetaSoftware, (1993), *Design IDEF Tutorial for the IBM PC and Close Compatibles. V1*, Tutorial, 1993, Meta Software Corporation, Cambridge, MA, USA.

Möller-Pederson, B., D. Belsnes and H. P. Dahle, (1987), *Rationale and Tutorial on OSDL: An Object-Oriented Extension of SDL*, Computer Networks and ISDN Systems, 13, (2). 97-117.

Nijssen, G. M. and T. A. Halpin, (1989), *Conceptual Schema and Relational Database Design: A Fact Oriented Approach*, Prentice Hall, East Brunswick, Victoria.

OMG, (1991), *The Common Object Request Broker: Architecture and Specification*, OMG Document Number 91.12.1, Object Management Group & X/OPEN.

Peckham, J. and F. Maryanski, (1988), *Semantic Data Models*, ACM Computing Surveys, 20, (3).

Peric, K. and F. Simon, (1991), *Computer Aided System Engineering for EDI*, Research Report, CR-DTS/91-03, July 1991, S.W.I.F.T Research & Engineering.

R.J. Wieringa, J.-J. C. M. (Ed.), (1993), *Deontic Logic in Computer Science*, John Wiley & Sons.

Rock-Evans, R. and B. Engelien, (1989), *Analysis Techniques for CASE: a Detailed Evaluation*, Ovum.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, (1991), *Object-Oriented Modelling and Design*, Prentic-Hall.

Schäl, T. and B. Zeller, (1993), *Supporting Cooperative Processes with Workflow Management Technology*, Tutorial Notes ECSCW'93. Milan, Italy.

Singh, B. and G. L. Rein, (1992), *Role Interaction Nets (RINs): A process description formalism*, CT-083-92, Microelectronics and Computer Technology Corp.

SWG-EDI, (1991), *Report on the Open-EDI Conceptual Model*, Report, ISO/IEC JTC1/SWG-EDI N222-1, 21/5/91, ISO/IEC.

Theodoulidis, C., B. Wangler and P. Loucopoulos, (1992),*The Entity-Relationship-Time Model*, in Conceptual Modelling, Databases, and CASE, P. Loucopoulos and R. Zicari (Ed.), Wiley.

Vanslembrouck, J. and B. Verdonk, (1991), *ESPRIT Project KIWIS: A comparison of service description techniques*, Esprit project deliverable, CASE 31, August 1991, Alcatel Bell, Belgium.

Veijalainen, J., (1992), *Issues in open EDI*, Research Notes, 1323, 1992, VTT-Technical Research Center of Finland.

Wangler, B., (1993), *Business Rule Capture in TEMPORA*, E2469/SISU/NT3.1/2/1, March, 1993, SISU.

Svenska Institutet för Systemutveckling, SISU, bedriver forskning, följer utvecklingen och förmedlar kunskap om informationsteknologins tillämpning på informationsanvändning och informationsförsörjning i företag, myndigheter och andra organisationer. Institutet verkar inom detta område som ett opartiskt nationellt kompetenscentrum.