ORDIT: A New Methodology to Assist in the Process of Eliciting and Modelling Organisational Requirements

A.J.C. Blyth, J. Chudge, J.E. Dobson and M.R. Strems

TECHNICAL REPORT SERIES

No. 456 November, 1993
TECHNICAL REPORT SERIES

No. 456 November, 1993

ORDIT: A New Methodology to Assist in the Process of Eliciting and Modelling Organisational Requirements

A.J.C. Blyth, J. Chudge, J.E. Dobson and M.R. Strens

Abstract

Requirements engineering from an organisational perspective needs to be viewed as social engineering. Thus in this paper a modelling language will be presented, which is visual in nature, and with which we assert that it is possible to diagrammatically represent and reason about the impact that an information technology system may have on an organisation, and thus derive organisational requirements.
Bibliographical details

ORDIT: A NEW METHODOLOGY TO ASSIST IN THE PROCESS OF ELICITING AND MODELLING ORGANISATIONAL REQUIREMENTS.

ORDIT: A New Methodology to Assist in the Process of Eliciting and Modelling Organisational Requirements.
[By] A.J.C. Blyth [and others]


(University of Newcastle upon Tyne, Computing Science, Technical Report Series, no.456)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE.
BLYTH, Andrew John Charles

Abstract

Requirements engineering from an organisational perspective needs to be viewed as social engineering. Thus in this paper a modelling language will be presented, which is visual in nature, and with which we assert that it is possible to diagrammatically represent and reason about the impact that an information technology system may have on an organisation, and thus derive organisational requirements.

About the author

Mr. A.J.C. Blyth is a Research Associate at the University of Newcastle upon Tyne. He is currently employed on an ESPRIT project called ORDIT researching into requirements engineering, while at the same time studying for a Ph.D.

Mr. J. Chidge is a Research Associate at the University of Newcastle upon Tyne. His first degree is in Philosophy and his second is Cognitive Science. He has been employed on ORDIT for the past three years.

Mr. J.E. Dobson is a Principal Research Associate at the University of Newcastle upon Tyne, with particular interest in a system approach to security. He has been active in the field since 1982 and is the author of more than a dozen papers.

Dr. M.R. Strens is a Research Associate at the University of Newcastle upon Tyne. She has a Ph.D. in Geology with a strong background in computer modelling. She has been employed on ORDIT for the past three years.

Continued/
Suggested keywords

CONVERSATION MODELLING
ENTERPRISE MODELLING
FUNCTIONAL RELATIONSHIPS
ORGANISATIONAL REQUIREMENTS
STRUCTURAL RELATIONSHIPS

Suggested classmarks (primary classmark underlined)
Dewey (18th): 621.381958 651.8 001.6425
U.D.C. 681.322.02 681.3.01 681.322.06
ORDIT: A New Methodology to Assist in the Process of Eliciting and Modelling Organisational Requirements

A.J.C. Blyth, J. Chudge, J.E.Dobson and M.R.Strens
Department of Computing Science
University of Newcastle upon Tyne
NEWCASTLE UPON TYNE NE1 7RU
United Kingdom

Abstract

Requirements engineering from an organisational perspective needs to be viewed as social engineering. Thus in this paper a modelling language will be presented, which is visual in nature, and with which we assert that it is possible to diagrammatically represent and reason about the impact that an information technology system may have on an organisation, and thus derive organisational requirements.

Key Words: Organisational requirements, enterprise modelling, structural relationships, functional relationships, conversation modelling.

1 Introduction

"Computers and coffee machines are perhaps the two most striking artefacts of the workplace today. To understand these artefacts we have to understand how people at work use them. For example the coffee machine is not just used to produce a stimulating drink: more importantly it offers an opportunity for people to meet, for communication in the workplace. Similarly, computers are not just instrumental means of production: they also condition and mediate social relations at work." [1].
With computers taking part in, and mediating social relationships new requirements engineering methodologies are required to address this very issue. Conventional system analysis has largely concentrated on the information processing rather than taking a wider view of the problem. Consequently a great number of systems have been developed which fail to greater or lesser extents to satisfy the needs of their human

Permission to copy without fee all or part of this material is granted provided that copies are not made or distributed for commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

In essence what is required is a methodology that views the engineering process with which information technology systems are developed as a social one. Traditional engineering disciplines such as electrical engineering have drawn upon the physical sciences to provide them with a rigor and formal basis. Social engineering is however built upon the concepts and ideas developed in the social sciences. These sciences attempt to understand the complex way in which social systems behave. Unfortunately rather than these models being built upon any formal logical system, they are built upon observations of the way in which social systems have behaved in the past, consequently they lack any rigor or formal basis upon which to draw conclusions.

Organisations are by their very nature complex and ever changing social systems. In an attempt to address some of the problems currently faced by methodologies in developing IT. Systems to meet organisational needs the concept of a socio-technical systems methodology has been developed. The socio-technical approach to systems analysis in one in which the system is viewed as a whole by placing it within the broad operational environment with the user as an integral part of the system. Examples of such systems analysis methodologies include SAMPO [2], ICN [3] and ERD[4].

The key difference between the models to be presented here as used by the ORDIT methodology, and those used in SAMPO, ICN and ERD is that the models presented here employ not only linguistic concepts but also social ones. In addition the ORDIT methodology expresses the need for requirements engineers to talk the same language as the problem owners as opposed to some high level abstract specification language.
Designers have generally ignored the importance of organisational issues in the design of IT products, and hence many of the difficulties encountered have been due not to limitations in technology but to the disregard of organisational requirements. It is only fairly recently that requirements of an organisational nature, while acknowledged to be essential for successful implementation of socio-technical systems, have become an explicit focus of attention in systems design. The central tenet of the ORDIT philosophy is that design methods appropriate for technical systems cannot simply be applied to socio-technical ones, and that consideration must be given equally to both human and technical issues, with success being seen as the construction of a relevant socio-technical system that meets the 'real' requirements of the organisation.

The purpose of the modelling language is to provide a common conceptual framework and communications medium within which both the problem owners and the problem solvers can engage in a dialectical [5] exploration of both the problem and the solution space. In providing a common language whose semantics are neutral to both the solution and the problem domains, a clarity of expression and understanding can be achieved. As a direct consequence miscommunication can be avoided as the true solution for the true problem can be developed and stated.

We begin by describing the rationale behind ORDIT and how this in turn obviated a need for the development of a modelling language through which organisations could be analysed. We then examine the interactions that occur between the agents within an organisation and present a means of representing these that is in accord with the ORDIT modelling language.

2 The ORDIT Methodology

The aim of the ORDIT project is to develop a methodology that will enable systems designers to reason about organisational goals, policies and structures, and the work roles of intended end users in a way which will facilitate the identification and expression of organisational requirements for IT and CSCW (Computer Supported Co-operative Work) systems, and furthermore, one which will support these structures and roles. Our concern therefore is not solely with the creation of a framework in which such issues can be identified, but also with the development of a language with which to discuss human requirements of socio-technical systems, and to demonstrate how these are linked to the technical features of the system design.

We believe that our approach addresses the problems that arise when designing large and complex systems. The areas of complexity are both 'technical' and
'organisational'. Technical complexity is fairly well understood, the principal tool being that of abstraction whereby the overall picture is constructed, with more detail being added as the design process proceeds. Organisational complexity, however, is another matter. The traditional notion of the software development life-cycle with requirements capture being completed before the design stage is no longer satisfactory. Requirements capture and design are now seen to be symbiotic. The initial set of requirements needed to start off the design process is gradually refined into a systematic and coherent statement of requirements hand in hand with the refinement of the design.

2.1 Components of the Methodology

There are three main components of the ORDIT methodology:

A Process Model

This is a model of the process of eliciting and modelling requirements. One of the main characteristics of the ORDIT process model is the way it has separated these two functions and shown the relation between them. Requirements are not considered as butterflies: they are not there to be captured and pinned down in a specification cabinet. Rather, the process of finding them is one involving at least three roles (requirements owner, requirements elicitor, requirements modeller) and a number of separate tasks involving some interesting feedback loops.

An Enterprise Modelling Language

The ORDIT project has devised an enterprise modelling language to represent the structure of the organisation in order to serve two related but distinct purposes: to determine the requirements owners and their positions and roles within the organisation; and to determine the user community (and others affected by the proposed IT system) and their roles and responsibilities within the organisation. The first purpose is in order to demonstrate completeness of the requirements elicitation process, and the second is required in order to demonstrate completeness of the requirements modelling process.

A Role Reference Model

The ORDIT concept of 'role' is perhaps the most sophisticated of all current role models, covering such things as functional and structural relationships, responsibilities, information access modes and rights, conversation structures and role
evaluation criteria. The concept of the Role Reference Model has been developed by ORDIT in order to capture all these aspects of 'role' that are so important for a full definition of the organisational requirements placed on a system by the roles with which the system will interact.

In addition there is an information modelling language used to derive the data models which are required and there are also supporting tools which assist the modelling but these two items will not be discussed in this paper.

3 Requirements Engineering

3.1 Setting the Context for Organisational Requirements

Within organisations, large tasks tend to be devolved to groups of people who work together in complex ways to achieve an overall objective. This has always been the case, and yet technical systems design tends to assume a single user with a discrete task. The failure to recognise that users work in a collaborative or co-operative way, and to design systems to support this way of working, can account for the relatively low success rates of many complex technical systems. One of the aims of the ORDIT methodology is therefore to enable design teams to address these organisational requirements, and thereby to produce IT systems that match not only the organisational and functional needs of the individual end user, but also those of groups of users and their associated usability and acceptability requirements.

The requirements identified in order to achieve the aim of supporting different members of the design team are as follows:

- to identify the full range of relevant requirements in a specific organisational context;
- to derive appropriate functional specifications for any IT systems which take account of organisational as well as individual task requirements;
- to compare the organisational requirements match of different design alternatives;
- to represent the range of organisational requirements to the problem owner and the systems designers in an iterative way;
to identify organisational mechanisms or processes for fulfilling critical non-functional requirements.

The method used to determine requirements must allow the system designer to explore possible solutions (involving both the IT system and possible organisational change) and their consequences at the same time as specifying the problem, thereby refining the understanding of the problem and developing the solution by an iterative process. We have found it convenient to represent the general requirements process within Ordit as four broad interactive component sub processes, namely scoping, modelling, requirements capture and solution options, as shown in figure 1. Important points to note are that the activities shown within the sub processes and the sub processes themselves are in no way sequential and there is no set route through the diagram, the whole process being iterative with feedback to the client at every stage. For example, in contrast to other methods, modelling may be started at a very early stage to help in exploring the system boundaries and in identifying stakeholders. This process is in contrast to the traditional 'waterfall' approach to modelling in which the output from one stage forms the input to the next stage and so on, with all the stages following a pre-determined order. The objective of the ORDIT model is to support the identification and transformation of organisational requirements into precise statements which can be operated upon by systems designers without being prescriptive as to the order in which the various operations involved in this process are carried out.

![Diagram of the ORDIT Process Methodology]

**Figure 1: A Representation of the ORDIT Process Methodology**

### 3.2 The Requirements Process

The four major sections in figure 1 outline a process for arriving at a set of organisational requirements on an information technology system.

**Scoping**

This component sub process deals with determining the scope of the requirements determination team and the contractual agreement with the client,
establishing boundaries for the contract, gaining an understanding of the purpose and structure of the organisational unit(s) which are to be involved, and identifying the principal stakeholders involved.

Modelling

The purpose of this component sub process is to represent the current understanding of the socio-technical system by producing a set of models. This provides not only information about the environment in which the IT system is to function, but also a context for understanding later policy and design decisions. One particularly important use of the description is to act as a focus for discussion on what sort of system the new system should be, and on how responsibilities and authorities in the organisation are going to change as a result of introducing the new system. Another important function is to assist in the scoping sub process, by providing models for the determination of relevant stakeholders and system boundaries.

The existing system is described in terms of the system itself and its organisational environment and then agreed with the problem owners. The next step is to model the existing system in terms of abstract agencies and a general information and computational system. In addition models are annotated with relevant requirements policies concerning issues such as combinations of agents into roles, access, authorisation, scope and objectives.

Requirements capture

The purpose of this component sub process is to identify those organisational requirements which must be observed in any implementation. Natural language statements of requirements are elicited from problem owners in order to agree upon how the system should be rather than how it is, and to identify defects in the current system and how these might be overcome. Statements of requirements are then fed back to problem owners and agreed. Requirements are then classified in order to identify conflicts, particularly conflicts in definitions of boundary objects and organisational aspects. It is an important feature of the ORDIT methodology that this stage of the requirements process is more concerned with the determination of what organisational responsibilities need to be supported by the IT system than what functions the IT system is to perform. Experience in a number of case studies has convinced us that taking a functional or activity-based view of a system at this stage leads to significantly lower quality (i.e. less fit-for-purpose) requirements than taking a view of a system as supporting contractual and work-related obligations.
Solution options

Requirements and some priority ordering are used to generate possible design options for the socio-technical system, with conflicts and trade-offs being resolved with the client. One of the purposes of the earlier stakeholder analysis is to answer not only the question of how and by what criteria to prioritise stakeholders, but also whether different prioritisations are needed for different purposes (e.g. data collection, validation/feedback, policy decisions). The implications of the design options are analysed and discussed with relevant stakeholders in an iterative fashion. The acceptability of the preferred option is agreed with the problem owners and other stakeholders, ensuring that the option meets the formal model of requirements.

The 'space' in which solution options are discussed is a socio-technical space, i.e. it encompasses possible changes in the organisation and new organisational constructs as well as changes in the technical system and new IT constructs. These must be considered together. For example, an organisation which wishes to improve its repair service carried out at customer premises might wish to consider making its repair service an integral part of its operations, or a separate organisation (wholly-owned or independent), or make its repairers a set of independent agents with a common co-ordinating service to which they all subscribe (the taxi-driver model). Each of these options will place its own distinctive set of requirements on an IT system to support the requesting, scheduling, use and monitoring the repair service.

This is the final step in problem space of requirements analysis. Subsequent stages consist of a response by the system architects and designers to the requirements, and are therefore expressed in solution space terms.

4 Enterprise Modelling

The modelling concepts [6] are perhaps the key aspect that makes ORDIT different from more conventional approaches to design. Enterprise modelling provides a framework for representing and reasoning about the IT system as a component of a wider environment which is the organisation whose needs it is designed to serve. We have found that this form of modelling ensures an adequate representation of the structural and organisational aspects of the problem making explicit policy issues and assumptions which cannot so easily be stated. One of the main characteristics of the ORDIT approach is that we model responsibilities and relationships rather than activities and base our understanding of the architectural framework on this and not on the current implementation.
4.1 Basic concepts of the Enterprise Model

The three essential elements of the Enterprise Model are agency, actions and resources. The relations between these entities are defined in Figure 2 [7]. Agencies are regarded as primary manipulators of the state or structure of the system, and agency is the only object that can create, modify or destroy other objects. Actions are the operations that change the state of the system, and they are performed by agencies. All actions must induce state changes in the system that are visible to one or more agencies. The resources can be of two types: physical or logical, where physical resources are tangible objects such as nuts and bolts, and logical resources include information, time etc. When modelling organisations at the enterprise projection level [ANSA 1989], resources act either as tokens of responsibility signifying that an agency has a binding responsibility upon it, or as objects for which some agency is responsible. An important type of logical resource is data. When data are passed from one action to another interactions occur, the data being the bearer of those interactions.

![Figure 2: The Basic Elements of the Enterprise Model and their Relationships.](image)

The enterprise model also provides us with a framework from within which it is possible to examine the kinds of relations between entities of the same type and of different types. By examining these relationships a problem owner and solver can begin to understand how the basic objects influence and are related to each other. In our enterprise modelling we have chosen to concentrate on the concepts of 'role' and 'agency' as being central to our procedure.

4.2 The Concept of Role

We choose to describe an organisation as a set of related work roles for the following reasons:
A role is a descriptive concept that can be used to represent many different organisational realities from the formal and structured to the fluid and unstructured.

Treating role as a basic building block makes it possible to move between organisational requirements and the requirements of individual users (e.g. from the organisation's role in a project to the way these responsibilities devolve to the roles of members of the project team).

A role defines task responsibilities and thereby functionality requirements.

A role defines the relationships between role holders and the behaviour they expect of one another which in turn defines many non-functional requirements.

Hence our concept of role allows us to distinguish: a) an agent with associated obligations such as accountabilities and responsibilities to other agents; and b) activities that interact through information flows and are structured into tasks and operations. This enables us to represent and analyse the relations between these concepts and to represent the way in which they operate in real organisations.

4.3 The Concept of Agency

ORDIT aims to describe and reason about organisations that embody both a social and a technical system. These however comprise one single system, a socio-technical system, and, as such, cannot be described or modelled in terms of state and behaviour only as a purely technical system might be, since there is a fundamental difference between social and technical systems. It is to be able to differentiate between social and technical objects (i.e. between people and computers) that we introduce the idea of agency. A machine may perform the same tasks as a person, but the person will hold responsibilities for those tasks in contrast to the machine which cannot hold responsibility. The person is said to be an agent and hold the agency.

It is important to realise that an agent is distinct from both an individual human and a role. An agent holds the particular set of responsibilities that comprise an agency. Thus depending on how responsibilities in a social system are allocated and combined, so agencies are composed and decomposed. An agent also differs from an individual in that an individual may hold more than one agency simultaneously. An agent differs from a role in that a role is not merely an agency or a collection of
agencies but also includes a set of relationships with other agents. These are structural or social in nature, arising from responsibilities that relate to the other agents.

This concept of agency is one of the strengths of the ORDIT approach to reorganisation of a socio-technical system, since it facilitates the reallocation of agency in a way that takes into account as fully as possible the structural and organisational implications of the change. Since agency is considered as a coherent set of responsibilities, it permits the discussion of issues related to the change in and reallocation of responsibilities when some functions or agents in the system are proposed to be automated.

4.4 Enterprise Diagrams

Enterprise diagrams were first presented in [8] as a graphical means of representing and reasoning about organisational policy and structure. This notation makes a clear distinction between the behavioural component of a system and its structural component.

![Problem Owner: Problem Repair
Consumer: Supplier](image)

Figure 3: An Example of an Enterprise Diagram

Figure 3 is an example of the basic building block of our enterprise modelling diagrams, which shows the roles involved in repairing a customer's appliance. We have used these diagrams in an actual case study to discuss options for reorganising an electricity supply company's policy on the future organisation of its customer premises repair service. The main point to note about this diagram is that each box contains the names of what we term a functional role and a structural role. The functional roles (problem owner, problem repair) are shorthand for the kind of behaviours that the parties may engage in (e.g. the problem owner can report the existence of a problem, the problem repairer can mend it). The structural roles (consumer, supplier) are shorthand for the framework of obligations that permit and give meaning to these behaviours. The lines that join the functional and the structural roles represent the fact that a relation between two agents may be one of interaction or one of commitment. Commitments and interactions are related to each other through activities such that commitments only arise and are fulfilled through interactions. So, properly speaking, the links between them are in fact functional and structural relationships.
5 Structural Aspects of the System

5.1 What a Structural Role Represents

While agents act as the primary manipulators of the systems state, agencies act as repositories for responsibilities and obligations and structural roles act as the binding point for those organisational attributes. A structural relationship serves as a means for the responsibilities and obligations to flow from one agency to another and thus facilitate in the flux of responsibilities and obligations through an organisation.

The key to understanding the nature of structural roles and their relationships with each other is in understanding the primary purpose of the enterprise projection and hence the uses to which they will be put. The main aim of the projection is to allow a problem solver to model, and thus to comprehend, how organisational attributes like obligations and responsibility are established, flow through an organisation and are then discharged or fulfilled. By understanding such organisational attributes the problem solvers are better equipped to comprehend how the organisation interprets not only such attributes but also the binding of such attributes to agencies within the organisational setting. Thus by understanding how the organisation interprets such attributes the problem solver can better understand how a computer system would function within the organisation.

The set of structural roles that an agent can hold is divided into three types, a power relationship, a peer relationship and a service relationship. These relationships are described in the following sections.

5.1 The Basic Types of Structural Relationships

The peer and power relationships may be viewed as relationships at different ends of a co-worker relationship spectrum. Thus a total power relationship is where one agent is totally subservient to another agent, and a total peer relationship is where neither agent is subservient in any way to the other agent. There are of course an infinite number of possible relationships in the middle.

The difference between a power / peer relationship and a service relationship is that power / peer relationships tend to exist within organisational boundaries, whereas service relationships tend to cross organisational boundaries. Thus we may view these relationships as intra and inter organisational relationships respectively.
5.2 Power Relationships

The essence of a power relationship is that one agency has the power to make and enforce demands on another agency. It is important to note however that the enforcement of these demands may be made via a third agency. An example of a power relationship is the \textit{supervisor-subordinate} relationship that can exist in most organisations, there are however, many different types of this relationship. In this relationship the supervisor has the power to define the responsibilities and obligations that a subordinate is required to fulfil, and to judge whether or not the responsibilities were correctly discharged.

The subordinate is not totally subservient to the supervisor in that the responsibilities and obligations that the subordinate is required to fulfil are defined by means of interaction between the two agencies. The types of power relationships that can exist between two agencies within an organisation can be defined with reference to the types of interactions that are meaningful for the two agencies to engage in. For example if a person's boss punishes them and they think that the punishment was unfair then they may appeal to a higher authority, the final authority being, of course, the law courts. The nature of these relationships is very complex and something that the problem solver would need to explore carefully and in depth with the problem owners.

5.3 Peer Relationships

The peer relationship is a far more subtle relationship than the power relationship as this relationship appears to be more social in nature than the power relationship. The nature of a peer relationship is that of equality. In a peer relationship there is no implication of enforcement, in fact, it is exactly the lack of this attribute that is characteristic of peer relationships and makes them special. Thus when two agents are in this relationship they may request that each other perform various tasks however they lack the facility to enforce execution. Hence agreements to perform actions are achieved by means of mutual agreement. An example of a peer relationship is that of the colleague-colleague relationship.

5.4 Service Relationships

In a service relationship one or both of the agents have the power to invoke the execution of a predefined and agreed task by another agent. This task will in some way relate to both the invoking and executing agents. An example of a service relationship is the consumer-supplier relationship. An example of this relationship is
that relationship that most people can be said to hold with an electricity board. In this relationship one agent acts as the consumer of a service another agent acts as the supplier of the service. The difference between a service relationship and a power relationship is that when the consuming agent is dissatisfied with the service provided by the supplying agent then the consuming agent may appeal to a third agent. It is this third agent that has the ability to enforce its judgements on both the supplying and consuming agents. A service relationship is in essence one agent invoking the performance of a predefined task by another agent with predefined rules for the enforcement of the correct execution of that task. Thus a consumer does not have a power relationship with the supplier.

5.5 Explications of Structural Relationships

The structural relationship diagrams that will be introduced in this section are normative. That is they attempt to explain what is required for a particular structural relationship in order for it to be such a relationship. Therefore we term such a diagram an explication of the structural relationship that it represents.

When modelling the structural relationships that can exist within an enterprise there are three questions that the model should be able to answer. The first question is what responsibilities and obligations are allowed to exist within the relationship. The second question is what enterprise objects i.e. resources etc., must exist in order to support and give meaning to the responsibilities and obligations. It is important to note that these objects can act in one of three ways: firstly they can act to show that a responsibility or obligation has either been established, discharged or altered in some way. Secondly they can act as objects over which responsibilities and obligations are established, and finally they can act as a medium of communication.

For example a problem specification document may act as a token of an agent's acceptance to perform an action and thereby the creation of a responsibility to perform an action. Whereas in a hospital the drugs may act as a resource over which and for which responsibility is awarded. The third and final question is under what conditions are the relationships between the objects valid. E.g. An identifier and password for a computer are only valid between the hours of 8:00 am and 9:00 pm, or, a ships captain can only perform the marriage ceremony while his ship is at sea.

The answers to these questions can be examined by placing the access to the object, and the creation of their attributes within the context of an interaction. Part of the purpose of a structural relationship is to define what types of interactions the two
agencies may engage in. Thus implications of changes to objects and attributes of the system may be explored and examined.

![Diagram](image)

**Figure 4: A Structural Relationship Diagram**

A structural relationship diagram is depicted in Figure 4. In this diagram there are a few things to notice. The first is that each object type is represented as a distinct shape, i.e. the agents are drawn as rectangles, the tasks as ovals and the resources as rhomboids. The type of arcs i.e. access rights are defined by the enterprise modelling language [6]. The arcs may also have a condition associated with them. The task that is depicted at the centre of Figure 4 is derived from the responsibilities and obligations that a particular agency may hold. As was stated in [9] a responsibility is a three place relationship between two agencies and a state of affairs. For this relationship we say that the agency A is responsible (in some way) to the agency B for bringing about or maintaining a state of affairs C. Consequently it is from this that the task definition is derived. A structural relationship diagram can be used in one of two ways by the problem owners. The first is to help them in their task of requirement elicitation by prompting them to ask certain questions. For example when and under what conditions is this relationship between two objects meaningful. The second is in allowing them to explore the ramifications, implications and possible contradictions of policy statements.

6 Functional Aspects of the System

6.1 What a Functional Role Represents

Functional relationships links together two functional roles in different agents or agencies where each agency or agency can also be called a role holder. One of the purposes of functional relationships is to define the behaviour that a role holder may engage in with another role holder within the context of a structural relationship (see Figure 3). Thus we may say that one of the purposes of a structural relationship is to define the context for a functional relationship. In defining and modelling the behaviour of a role holder, the problem solvers are in fact defining and modelling the set of allowable functional relationships that can exist for that particular role holder.
Functional relationships facilitate the problem owners and problem solvers in their tasks of defining and modelling the interaction that can exist between two or more role holders. In addition functional relationships aid in the identification of the organisational objects that are required to give meaning to the interaction. The purpose of a functional relationship from the perspective of its role holders is to facilitate in the correct discharge of their responsibilities and obligations.

From the perspective of functional relationships organisational objects can be classified into one of two types. The first type of objects are the enterprise attributes of obligations and responsibilities that are placed upon role holders. These enterprise attributes are placed upon the role holders by virtue of either the tasks that they are required to perform, or, the state of affairs that they are required to achieve or maintain. The second type of objects are the enterprise objects and their relationships that are required be to present in order for the role holders to successfully discharge their enterprise attributes.

The modelling and defining of the life cycle of both enterprise objects and enterprise attributes is achieved via the modelling of the set of interactions that the role holders of an organisation may engage in.

6.2 Interaction between Role Holders

In the enterprise projection the interaction between two role holders defines how, when, where and under what circumstances projection attributes like obligations and responsibilities are established, flow through the organisation and are finally discharged or fulfilled. By modelling the life cycle of such attributes the problem solvers may attempt to answer several types of questions.

The first type of question allows for the examination of the possible conflicts that could arise for any given role holder. The term conflict is used to denote a situation where a role holder is either obliged or responsible to perform an action or bring about some state of affairs whilst at the same time being obliged or responsible to either not perform the action or not bring about some state of affairs. The second type of question is concerned with the elucidation of the conditions under which an agent can either not discharge an obligation or not fulfil a responsibility. The third type of question is concerned with the elucidation of what objects act as tokens of either obligations or responsibilities. The fourth type of question is concerned with the delineation of the valid accesses to objects that act as token of either responsibilities or obligations. The final type of question is concerned with the examination and
comprehension of the correct creation and deletion of the objects that act as tokens of either responsibilities or obligation.

6.3 Modelling Interaction

When modelling the interactions that can exist within an organisation a clear comprehension of the types of questions that such models will attempt to solve is required. In maintaining a clear understanding of such questions, the users of the interaction modelling notation can achieve a clear perception of the notations primary objectives. By maintaining this clear perception the problem solver will not be lulled into misusing the notation by attempting to answer questions which lie within the scope of other projections.

The types of questions that the model of interaction will attempt to address can be classified into two broad categories. The first category is concerned with the establishment and discharge of obligations and the creation and fulfilment of responsibilities. The questions attempt to aid the problem owners and problem solvers in their understanding of how, when, where and under what conditions such enterprise attributes exist. Hence they may begin to address such question as *when does X become responsible for Y*. The second class of question is concerned with the accessing of objects (resources) that act as tokens for the existence of enterprise attributes.

6.4 Speech Act

A speech act object aids in the elicitation, comprehension and formalization of how, when and where organisational attributes such as responsibilities and obligations are either brought into being or fulfilled.

![Figure 5: A Speech Act Object](image)

A speech act object has four distinct attributes, the first expresses the idea of an agent owning a speech act object, this being the agent that makes the utterance. The second attribute is the type of utterance being made in the object, this in classical speech act theory [10, 11], is called the *speech act*. The third attribute is the enterprise
attribute that the utterance manipulates in some way and this attribute is optional. Finally the fourth attribute is the enterprise object that is accessed in some way as a result of the utterance and this attribute is also optional.

6.5 Decision Act

A decision act is used to identify when a role holder has to make a decision of some kind. Thus it is used to allow the problem solvers and problem owners to examine when, where and under what conditions decisions are made.

![Figure 6: A Decision Act Object](image)

A decision act allows you to not only express the decision that is being made but also the role holder that is making the decision. In addition conditions that are necessary for the role holder to make the decision can be expressed and explored. An important point to note about the decision act is that the conditions part of the act is optional. The conditions section of the act allows for the representation and examination of the state of the environment that is necessary for the role holder to make a decision. The terms state of the environment are used to encapsulate several concepts. The encapsulation includes concepts such as what enterprise objects and enterprise attributes are required to be present in the system in order for the role holder to make the decision. In addition decision acts aid in the classification and delineation of acts that are affected by, and affect, organisational policy.

The primary purpose of a decision act is to aid in the comprehension of what a decision means and thus what its implications are, i.e. what decisions are critical or not for the fulfilment of the organisation's responsibilities. In addition, decision acts facilitate the elicitation of what resources are not only necessary, but also sufficient for the role holder to make the decision. The decision act is the only act that combines issues of data flow with control flow and for this reason this act should have special attention paid to it.
6.6 Instrumental Act

Instrumental acts not only facilitate in the examination and formalization of when, where and by whom an instrumental act, (task) is performed. They also aid in the elucidation and identification of what enterprise objects and attributes are required and manipulated by the agent entity performing the act. The expression of the projection objects and attributes that are required in the performance of an instrumental act are optional and is thus denoted by a shaded areas in the diagram.

![Diagram of an Instrumental Act](image)

**Figure 7: An Instrumental Act**

An instrumental act can either be decomposed down into small instrumental acts or sets of interactions.

6.7 Start/Finish Indicators

The start/finish indicators are used to denote when an interaction begins and when an interaction terminates.

![Diagram of Start/Finish Indicators](image)

**Figure 8: Start/Finish Indicators**

Thus are thus used purely as control points to denote the entry and exit points of the interaction.

6.8 The Flow of Control

Multi-agent or group dialogues invariably create complications for the conversation modeller. He needs to be able to keep track of not only who says what and to whom, but also more importantly, the ensuing establishment of responsibilities and obligations. It is through these that we model the flow of control and we do so as
a black box with a set of input and a set of outputs and this is depicted in Figure 9. In this diagram the inputs are shown on the top of the black box and numbered from left to right \(i_1, i_2, \ldots, i_n\) and denoted by a semi-circle at the bottom of the input arc. The outputs are shown emerging from the black box and are numbered from left to right \(k_1, k_2, \ldots, k_m\).

The black box facilitates the problem solvers and problem owners in the examination and comprehension of the relationships that can exist between the input and the outputs without having to be concerned about the various types of speech acts. Thus a distinction is drawn between the data flow and the control flow parts of the interaction. It is important that when modelling an interaction each control block be carefully examined and its structure explored. This examination and exploration is to make sure that both the problem solving and the problem owning parties understand the implications of the various structures are used to represent the flow of control.

![Diagram Depicting Flow of Control](image)

**Figure 9: Diagram Depicting Flow of Control**

An example of the control block construct being used to aid in the process of interaction analysis can be constructed quietly easily. Consider a clerk in a dispatch office, the clerk can makes a request to a group of two engineers and then wait for a reply from both of them. The control box allows us to explore the conditions under which the clerk may perform certain actions such as make another request. Hence questions such as what does the clerk do if none of the engineers reply. And what does the clerk do if just one engineer replies can all be examined, and their answers explored by means of the control box.

### 7 MODELLING ORGANISATIONAL CAPABILITIES

The enterprise diagram depicted in Figure 3 facilitates in the delineation of the behavioural component of the organisation from its structural component. Structural relationships are used to define the various attributes of the enterprise model, such as what responsibilities and obligations can be created and manipulated, along with, what resources are required to support the fulfilment of the responsibilities and obligations. In addition the explication diagram defines any
conditions that might imposed on the existence of any enterprise object, attribute or relationship. The behavioural component of the enterprise diagram is modelled with the use of functional relationships and conversations. These define the dynamic components of the organisational model. The nature of the relationship between the functional and structural components of the model is one of capabilities. That is to say that agents are said to have the capability to discharge their responsibilities and obligations, when they possess the relevant functional roles and relationships, and can thus engage in the correct conversation.

For example, the head of a university department may be said to hold the responsibilities for hiring people, yet by examining the interactions that he may engage in we can examine whether or not he has the capability to discharge his responsibilities, e.g. does he have the power to say "I offer you this job", or is that utterance made by some other person in some other department.

8 The Man in the Van Case Study

Electricians in the Area Board provide a wide range of services on customer premises, such as repairing appliances and restoring supply. This can be a costly and wasteful process with electricians in vans inefficiently deployed, clerical staff unable to process customer requests quickly, and long delays in communication. One problem is that the communication to arrange visits to premises for emergency work is undertaken by radio. This can often be sub-optimal, e.g. when the electrician is not in the van, the radio operator has to stack calls and keep trying. This case study is concerned with the installation of mobile data terminals in vans which can receive and store calls. The purpose is to examine the organisational implications of setting up a technical system of this kind - but here we will present only the use of the modelling techniques.

8.1 The System as it Exists

The Customer Service Unit consists of a team of clerks and a radio operator controlling communications with the fleet of service vans, each service van belonging to an electrician. There are four depots each with a foreman supervising the work of a team of electricians who are of two types. Day electricians mostly undertake specialist activities requiring specialist knowledge, e.g. fixing and maintaining washing machines, whilst the shift electricians are responsible for the statutory duties of the Board, e.g. attending to main fuses on a property to restore supply.
8.2 The System as Proposed

The mobile data terminal put in each van will be capable of receiving and storing up to 25 complete job messages. These can be displayed on a screen or printed at the electrician’s discretion. The messages will be transmitted from a computer terminal operated by the radio operator in the Customer Service Unit. The management are looking for the following kinds of benefit from the system: faster response to emergencies, prompter response to non-emergencies, better customer services, better utilisation of electrician resources and more efficient use of office staff.

To this end the Customer Service Unit clerks will not only pass emergency Work Instruction Sheets (WIS) to the radio operator but also non-emergency requests which the clerks records show electricians could undertake that day. The radio operator will then send them to the terminal in the van of the appropriate electrician and from this point on the jobs are treated as the electricians responsibility.

8.3 The Enterprise Model of the Case Study

A set of Enterprise diagrams can be constructed using the ideas presented earlier in this paper.

In Figure 10 the two main agent entities that are involved in the case study are depicted - the house-holder and the electricity board. This diagram shows us that the house-holder has a consumer - supplier structural relationship with the electricity board, this being represented in real life as the service contract that exists between them. The functional relationships show us the nature of structural relationship in that the house-holder can notify the electricity board of a problem and that the electricity board then has the responsibility to fix the problem. These two functions are illustrated the diagrams by the functional relationships problem notifier - problem reception and problem owner - problem repair.

![Figure 10: The Main Agent Entities](image-url)
In this case study the questions that we are attempting to answer are concerned with the reorganisation and re-structuring of the electricity board. Consequently we are concerned with what the responsibilities of the restructured organisation will be and how those responsibilities will flow through the organisation. Thus we proceed by decomposing the electricity board into its sub-components.

By decomposing the electricity board into its sub-agencies a picture of how the organisation functions can be composed. We can already begin to use these diagrams to elicit requirements by asking such questions as *what is the communication medium through which the functional relationships flow?*, and *what are the responsibilities that reside within the structural relationships?*

![Diagram]

Figure 11: The Sub-Agent Entities

Figure 11 depicts the sub-agencies and the relationships that exist not only with each other but also with the outside world for the problem repair agent entity. This diagram clearly shows the three sub agent entities *Foreman*, *Emergency Man* and *Ordinary Man* along with how the structural and functional relationships have been decomposed. It shows us how within the organisation the problem is classified, scheduled, and that the organisation discharges its responsibilities to the house holder through its electricians.

9 Interaction Analysis of the Case Study

In this case study the communications occur in the first instance between the consumer (the problem owner) and the Electricity Board itself (the problem repairer). Then, through the Board's own internal channels of communication, the nature of the problem is received by the electrician whose task it is to repair it. Supervision occurs
between the foreman and both the emergency electricians and the ordinary electricians.

In the existing system a clerk receives the customer's call at the Customer Service Unit (CSU) and makes a decision as to whether it is an emergency or not. If the customers' premises have to be visited the problem is recorded as a task for the electricians. For a non-emergency task the clerk raises a Work Instruction Sheet and allocates it to a day electrician at the appropriate depot. The foreman has access to the Work Instruction Sheet via a computer. When the foreman has checked it, he may then discusses the job with the electrician and raise the necessary parts from the store. Under certain conditions the foreman may even reallocate the job. For an emergency task the radio operator allocates the job to a shift electrician at an appropriate depot via radio. When work is completed the electricians complete job-sheets, which are checked by the foreman at the depot, and are then sent to the Customer Service Unit for billing and such like.

Here, we will present the conversations arising out of non-emergency situations because from the point of view of analysing them by interaction analysis, their dynamics make them more interesting and therefore they are likely to prove more insightful. Since the structural relation between the two individuals is one of supervision, we want to be able to assess whether the notation is able to represent this adequately.

![Diagram](https://example.com/diagram.png)

**Figure 12: The Offer Stage**

Figure 12 begins with a start/finish indicator. The next component is a speech act, and it is composed of three components, the first of which is the agent entity that
is making the utterance, the second is the type of utterance being made, and the third is the enterprise object that is being accessed in some way by the utterance.

The largest of the basic building blocks that is shown here is called a decision act. This act is also composed of three components, the first expresses the type of decision that is being made. The second is an optional component to the act and express what conditions are necessary for an agent entity to make the decision, and the third component is the agent entity that is engaged in making the decision. The final basic building block shown here is used to denote the termination of the interaction and is called the finish indicator.

Figure 12 shows us the state in the conversation at which the foreman offers the job to the electrician. This diagram serves to point out the three possible ways in which the electrician may respond to the job offer speech act made by the foreman. The electrician may either postpone the job, reject the job, or, accept the job. In performing the latter speech act the electrician is entering into an obligation with the foreman, that obligation being to perform the job specified by the foreman.

![Diagram](image)

**Figure 13: Job Accepted**

In Figure 13 the electrician is shown accepting the job offer made by the foreman and thus entering into an obligation with the foreman. The electrician then performs the task specified by the foreman and in so doing consumes some spare parts. Upon completion of the task the electrician reports back to the foreman and raises a work sheet (WS).

Once the electrician has raised a work sheet the foreman then reads the it and decides if the electrician was successful in the completion of the task as depicted in Figure 14. Thus the decision of the foreman has two possible outputs, either yes or no. If the foreman chooses the former, then he agrees to the electrician's appraisal of the situation and thus the electrician's obligation that was created earlier is discharged. If the foreman chooses the latter, then the job sheet must be amended to the satisfaction of both of them before being passed to the CSU.

When the job is allocated to the electrician and the electrician either rejects it, or postpones it, then the foreman may request an explanation from the electrician. In
the interaction model that has been presented here the conversations are shown as stopping. In reality it would be the problem solvers and problem owners who would dictate when the interaction modelling should stop. Nonetheless, it can be seen that understanding the relationships that can exist between the foreman and the electrician, as highlighted by the diagrams, is vital in comprehending how the organisation discharges its obligations and responsibilities to the house-holder.

![Task Completed Diagram](image)

**Figure 14: Task Completed**

### 10 Summary

We have found that our experience in ORDIT of constructing enterprise models and determining their bindings and boundaries has aided us in suggesting issues of job design and exploring organisational issues in a manner that pays heed to an organisation's requirements, goals, and policies. The problem of determining system boundaries of complex IT systems has meant that mistakes have occurred where the boundaries turned out to have been drawn in the wrong place. We claim that our modelling techniques provide a sufficiently rich environment in which organisational structure and the roles of agents, information flow, resource management, and the relationships between all of these are capable of being represented, and we are therefore able to capture the complexities of organisational structure.

The most common problem of requirements engineering in the design and implementation of complex IT systems is combining differing representations of the system and its environments: the operational, organisational, and social environments of a system all possess different characteristics. Hence the driving thrust of the
ORDIT philosophy is its advocacy of involving policy makers/problem owners throughout the design of the system. It is a process of shifting the balance of responsibility between system owner and system designer away from the 'owner states, designer solves' model towards a relationship in which the problem solver helps the problem owner understand the problem, and the problem owner helps the problem solver understand the implications of possible solutions.

Furthermore, used in the context of the enterprise projection, interaction analysis has shown, through the use of various case studies, its usefulness in dealing with an organisations requirements and policies. A major advantage is the degree of flexibility it places within the analysts control. For instance, it may be evident that in taking an object-oriented view, both of the organisation and of the concepts we use to model it, that enterprise modelling is capable of describing an organisation at any number of different levels, and the power of interaction analysis as a tool lies in being able to support this. However, because it is a time consuming exercise its most important domains of use may well be safety-critical systems, or wherever security and privacy is at a premium. Its power lies in enabling us to examine an actors responsibilities in a more dynamic way since the conversation exchange is a dynamic on-going process. Furthermore, the responsibilities are examined in direct relation to other agents’ responsibilities and this greatly enhances the capacity to check for conflicts within and between agents with regard to their respective roles and requirements. Responsibilities can now be allocated to agents or roles in the knowledge that questions of conflict, job satisfaction and so on, have been considered and addressed.

The diagrammatic representation we have developed is adept not only at capturing the dynamism inherent in a conversation but is also capable of conveying change-control from one agent to another, iteration, decision points and so on. Currently, we believe our representation technique to be the most powerful and flexible conversation representation tool. It has the added advantage that it can be used in systems or organisational design to construct scenarios of possible future states describing the relationships between agents, responsibilities, and resources. Interaction analysis, we believe, significantly narrows the gap between what the user wants and what the analyst can represent. Because of its emphasis on communication and interaction the system design it facilitates is very much a case of 'what you see is what you get'. The modelling language it is supported by facilitates a more open discussion of organisational requirements of both a technical and a human nature.

To sum up, the ORDIT approach recognises that the elicitation, representation, and re-presentation of requirements is an iterative process that is best accomplished
by a methodology that combines the social aspects of a system with the technical, and adheres to the principle of giving the customer what the customer needs and not what the system designers think the customer wants.

Acknowledgements

We gratefully acknowledge the contributions made by our colleagues in the ORDIT project (Algotech srl, HUSAT Research Institute, the MARI Group Ltd, and Work Research Centre) to the ideas presented here. We also gratefully acknowledge the support of the University of Newcastle upon Tyne and financial support afforded by the ESPRIT programme of the CEC (Project 2301).

References


