COMPUTING SCIENCE

A Framework for Modelling Evolving Requirements

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

TECHNICAL REPORT SERIES

No. 455 November, 1993
A Framework for Modelling Evolving Requirements

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

Abstract

The most crucial aspect of software engineering is the gathering of requirements. Increasing the amount of time and improving the quality of effort expended at this stage will lead to fewer and more manageable problems later on in the development life-cycle. The purpose of this paper is to highlight one of the most crucial problems that afflict user organisations when engaging in I.T. systems: the problem that the requirements for the system may change and evolve during the construction of the system. What we propose in this paper is a framework in which evolving and emerging requirements can be dealt with in an iterative manner within a method constructed upon a user-centred design philosophy. This paper is a presentation of recent developments made in the ORDIT project which was reported on last year.

© 1993 University of Newcastle upon Tyne.
Printed and published by the University of Newcastle upon Tyne,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.
Bibliographical details

FRAMEWORK FOR MODELLING EVOLVING REQUIREMENTS
A Framework for Modelling Evolving Requirements
[By] A.J.C. Blyth [and others]
(University of Newcastle upon Tyne, Computing Science, Technical Report Series, no. 455)

Added entries
UNIVERSITY OF NEWCASTLE UPON TYNE.
BLYTH, Andrew John Charles

Abstract
The most crucial aspect of software engineering is the gathering of requirements. Increasing the amount of time and improving the quality of effort expended at this stage will lead to fewer and more manageable problems later on in the development life-cycle. The purpose of this paper is to highlight one of the most crucial problems that afflict user organisations when engaging in I.T. systems: the problem that the requirements for the system may change and evolve during the construction of the system. What we propose in this paper is a framework in which evolving and emerging requirements can be dealt with in an iterative manner within a method constructed upon a user-centred design philosophy. This paper is a presentation of recent developments made in the ORDIT project which was reported on last year.

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. Chudge 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. Suren 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

CHANGING REQUIREMENTS
ENTERPRISE MODELLING
SOFTWARE ENGINEERING
REQUIREMENTS ENGINEERING

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
A Framework for Modelling
Evolving 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

The most crucial aspect of software engineering is the gathering of requirements. Increasing the amount of time and improving the quality of effort expended at this stage will lead to fewer and more manageable problems later on in the development life-cycle. The purpose of this paper is to highlight one of the most crucial problems that afflict user organisations when engaging in I.T. systems: the problem that the requirements for the system may change and evolve during the construction of the system. What we propose in this paper is a framework in which evolving and emerging requirements can be dealt with in an iterative manner within a method constructed upon a user-centred design philosophy. This paper is a presentation of recent developments made in the ORDIT project which was reported on last year.

Key Words: Enterprise Modelling, Requirements Engineering,
Changing Requirements and Software Engineering.

1 Introduction

As part of their defences the Roman invaders of Britain built a series of forts across the north of England. On one of these they built a turret (36(b)). Soon after, the line along which Hadrians Wall was to be built was determined and in order to
connect it to the fort, they had to demolish turret 36(b) as soon as they had finished building it. The problem of evolving requirements has been with us for a long time!

There is now good reason to believe that changing rather than stable requirements are the norm in systems development [1]. The uncertainty of requirements has been paid little attention by existing systems development methodologies [2, 3, 4] most of which are underpinned by a sequential or 'waterfall' approach in which one stage of development is completed and agreed with the client before the next begins. Within these frameworks requirements are agreed at an early stage in order that system development can be undertaken to meet these requirements. In this way the requirements become 'fixed' at the outset.

In system life-cycle approaches [5, 6], however, uncertainty is recognised. But here, the focus of attention is usually solution rather than problem uncertainty and it proceeds by verifying proposed solutions against a static problem statement. We, in turn, propose a framework which recognises problem uncertainty and proceeds by validating problem representations against stakeholders perceptions. Here, an important feature is that successive problem representations often have the effect of changing stakeholders perceptions as to what the problem is so that 'facts' established in previous phases cannot be assumed to remain true in later phases. This in turn raises the issue of evolving requirements.

Evolving requirements are those which arise out of the interactions between the problem-solving and the problem-owning organisations. They occur over time as the problem space is continually explored and probed with feedback and suggestions from earlier sessions constituting the input to the later sessions. This process serves not only to clarify and broaden the target but also to enhance the sense of ownership and commitment to the eventual solution amongst the stakeholders in the client organisation. The context of the requirements with which ORDIT [7] is concerned are those for an IT system which supports people in their work; and, we limit our interest to non-functional or organisational requirements e.g. "It shall be secure" which have to be reconsidered at each stage in the design process, here for instance, to determine exactly what is meant by 'security'.

The aim of this paper is to present a framework which we believe represents the full range of evolving requirements. The intention is to represent the degree of uncertainty in the requirements in order that system solution strategies can be chosen which will cope with the evolving requirements. For example, where analysis demonstrates that a lengthy debate about organisational goals and priorities will be necessary to choose between two futures, an appropriate technical architecture will
have to be chosen to keep the options open between these futures. We will propose the outline of a new framework that is currently under development in the ORDIT project which seeks to integrate the capture of evolving requirements of both a technical and organisational nature with a means of representing them in a clear, meaningful, and communicable way. The end result of this will be a more encompassive and powerful organisational analysis tool.

The ORDIT philosophy is based on socio-technical systems theory [8], with its basic premise that any system contains within it two sets of resource: technical and social (human), and that these are so inter-related that any attempt to optimise only one of these sets of resources may adversely affect the other set so that the resultant utilisation is sub-optimal. Design methods appropriate for technical systems cannot simply be applied to socio-technical ones, and 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 requirements that come out of a system being placed in a social context rather than those that derive from the functions to be performed or the tasks to be assisted are termed organisational requirements, and the objective of the ORDIT project is to develop a methodology for the capture and specification of organisational requirements so that these become an explicit focus of attention in systems design.

We begin with the problem i.e. the uncertainty of requirements and then present our conceptual framework which is capable of capturing and representing it. We will then briefly outline our approach which is transformational in nature and offer some suggestions on how it may be used with the aid of a case study we have been involved in.

2 The Uncertainty of Requirements

What then is the nature of the problem, what are the sources of uncertainty? There are a number of sources of uncertainty associated with organisational requirements specification [1] all of which demand consideration. We will list at least a few here to present their 'flavour':

- Organisations are dynamic entities - continually having to change. Often in lengthy systems development the organisation's goals at the beginning are unlikely to be the same at the time of delivery; however, they are likely to remain in the same business e.g. manufacturing cars. Therefore we have to separate fairly stable or enduring requirements of a functional nature from the mutable or unstable requirements associated with organisational uncertainty.
Emergent requirements are those which emerge, particularly in user-centred approaches, when clients are engaged in problem-finding through the technique of working through possible future scenarios with the designers. This process, which we believe is facilitated by our conceptual framework, serves to develop and make more concrete the requirements of the would-be users.

Early systems constrained the way users undertook their tasks; they were rigid and inflexible and did not support the users in their varied roles. This ignores a range of requirements concerned with supporting a wide range of local ways of operating, namely, adaptive requirements. Hence, systems must be flexible enabling customisation and/or personalisation so that users can modify them to suit their own purposes.

Pro-active requirements are those which evolve after delivery of a system. Users will inevitably discover new ways of working, new possibilities etc. and as a result push for enhancements and upgrades.

Coping with planned organisational development from a current state to a desired state involves charting a migration path in which some requirements are met before others in order to accomplish orderly change. Here, constraints or further requirements may be created along the way if the organisations operations are not to be disrupted. These migration requirements often provide meaningful insights into the culture of an organisation, the constraints and (market) forces it operates under, the limits and extents of what can and cannot be changed, and it also acts as a test of the whole system development process providing invaluable feedback. In this way it furthers the process of refining requirements in an iterative and user-centred manner.

The above are the uncertainties which exist in requirements specification. The next section deals with how these may be captured, specified, and represented within a coherent and consistent framework.

3 Conceptual Framework

It is important to note that satisfying such requirements can have implications for the structure and process within the organisation as well as for the architecture and design of the IT system. We have therefore devised a method that is applicable to the design of an enterprise as well as to the design of an IT system in its environment within the enterprise. It is based on the ANSA [9] projections which presents a set of
models along with a set of questions that the models attempt to address. The projection that will concern us in this paper, and is vital to the ORDIG methodology, is the enterprise projection. The modelling techniques that we have developed are sufficiently generic as to be capable of satisfying both sets of needs: to model the organisation and the requirements in it that the IT system will assist or induce, and to model a system architecture. For us, modelling is a process that is essentially iterative with requirements emerging as the analysis of the system is presented to and debated with the client.

The modelling techniques must therefore be capable of supporting the entire requirements process. We see this as a multi-faceted process encompassing the capture of requirements made explicit, the elicitation of implicit requirements and the generation of requirements that emerge only when possible futures are explored. Techniques that concentrate on functional requirements, i.e., those that can immediately be expressed in behavioural terms, are inadequate to deal with organisational requirements. We avoid this behaviourist trap by approaching the problem through responsibilities. This approach enables us to reason about organisational goals, policies and structures, and the work roles of intended end users in a way which facilitates the identification and expression of evolving organisational requirements.

4 Responsibility Modelling

As stated earlier requirements modelling is concerned not only with the functionality of the system but in ascertaining such aspects as the effect of the new system on work roles, what human interactions must be mediated by the system and what are the characteristics of the user interfaces in terms of appropriateness to the responsibilities of both the user and computer. To this end ORDIG has approached the problem of modelling complex socio-technical systems by developing a basic set of three inter-related models that each represent a major aspect of the enterprise: a responsibility model, an obligation model based on the obligations that describe the role or job, and an activity model that describes function.

The distinction between responsibilities, obligations and activities, and the relationship of activities to responsibilities through obligations is the central tenet of our conceptual modelling framework. This is based on the precept that a worker executes activities in order to discharge the obligations imposed on him by virtue of the responsibilities he holds. These obligations effectively describe his 'job' or role, and are the link between his responsibilities and the activities he executes.
The distinction between responsibilities and obligations is apparent from the words we use: a responsibility is *for* a state of affairs, whereas an obligation is *to do* (or not do) something that will change or maintain that state of affairs. Thus a set of obligations must be discharged in order to fulfil a responsibility. As such, obligations define in what way the responsibility holder is responsible, and how the responsibilities can be fulfilled. For example a hospital doctor may have responsibility for the medical condition of his patients. To fulfil this responsibility he must discharge obligations that may include to diagnose, treat, monitor and prescribe.

The distinction between obligations and activities is essentially one of specification of detail; we think of it in terms of abstraction of obligations away from activities. Activities are defined as operations that change the state of the system. The role holder may (or may not) have a wide choice of activities that discharge the obligation he holds. Consider again the hospital doctor role that includes an obligation to make a diagnosis. The actual activities undertaken by the role holder may be one or more of several: he may examine the patient, order x-rays or do tests.

5 The ORDIT Enterprise Models

We have therefore a set of three inter-related models based on responsibilities, obligations and activities respectively. Each of these models includes the same three basic entities and examines the relationships between them. Our modelling is centred on the agent entities in the organisation. These are the primary manipulators of the state or structure of the system and may be of any size from an individual to a whole organisation. It may also be a machine. The other entities are an action entity and a resource entity, where an action entity is an operation that changes the state of the system, and a resource entity is that which enables the agent to do the action. An icon showing these generic entities and relationships is shown in figure 1. We include the relationship between agent entities, termed a structural relationship, which is essentially a responsibility relationship.

The specific icons for our three enterprise models, corresponding to the generic icon in Figure 1, are shown in Figure 2:

The models are arranged vertically in the diagram to emphasise the fact that the vertical links join concepts of the same type, and that the intermediate level concepts (i.e. the role model) are the links between the more abstract and more real concepts. As stated above, obligations are recognised to be the link between responsibilities and the execution of activities in the real world. Similarly capability tokens, are the link between rights to access resources and resources themselves.
These relationships form the infrastructure that binds together the separate models into a coherent framework.

![Diagram of Structural Relationship](image)

**Figure 1**

Looking now at the three models we can describe the organisation at the **activity** level in terms of agents, what they do and what resources they use. Alternatively, at a slightly higher level of abstraction, we can look at the agent's activities in terms of its job or **role**. Here we describe what the role entails in terms of what the role holder is obliged to do and whether he has been allocated the capability tokens needed to discharge those obligations.

At the highest level of representation we can describe the **responsibilities** and rights held by the agent entities (agencies). In essence these three models describe the organisational process in terms of **why** an agent does something (responsibilities), **what** an agent does (obligations) and **how** it is done (activities).

The words chosen to describe concepts of a similar type are different in each model to accentuate the different aspect of the organisation represented by that model. For example we use the term **agency** for the agent entity when we model responsibilities and rights to emphasise the high level of abstraction and the intensional nature of the concept, whereas we use the specific term **agent** to convey an extensional concept in models of the real world containing activities and resources. Again at an intermediate level we use the term **role** when we describe the agent entity in terms of the obligations and capabilities that it holds.

We use the terms **rights** and **capability tokens** to denote high and intermediate level concepts concerned with access to **resources** in the broadest sense. Rights are permissions or authorisations and an agency can only fulfil its responsibilities within the bounds of the rights it holds. For example if an agency within a poorly structured organisation holds only a very restricted set of rights then its freedom to carry out the tasks associated with its responsibilities will be limited.
Similarly at the role level when job design is being considered, it is essential that the capability tokens allocated to a role should be commensurate with its obligations. Again if the set of capabilities held is restrictive it will limit the freedom of action and constrain how the obligations that constitute the role are discharged, if in fact it is possible to discharge them at all.

![Diagram of Responsibility Model](image)

**Responsibility Model**

![Diagram of Obligation Model](image)

**Obligation Model**

![Diagram of Activity Model](image)

**Activity Model**

The functional, access and structural relationships in each model (see Figure 1) are referred to by different terms in the different models to convey their specific nature in the context of the model in which they are used (see Figure 2). As mentioned above we speak of fulfilling responsibilities, discharging obligations and executing activities when describing the functional relationship in each model. We have chosen to speak of exercising rights, applying capabilities (capability tokens)
and accessing resources when describing the access relationship in each of the three models. Different terms for the requirements relationship have not been used in the various models as the term 'requires' seems to be the most appropriate in all three cases although requirements can be treated in a more flexible manner made possible by the three models.

It is this requires relationship that we will now turn our attention to in more detail by looking at the modelling process and the qualitatively different requirements generated in each of the three levels or models we have.

6 The Requirements Relationship

Let us begin with the question: what is the relationship between responsibilities and requirements? From the following two statements we can explore an aspect of this relationship: 'a responsibility is a statement about the current state of affairs', and, 'a requirement is a statement about a future state of affairs'. We can suggest that high-level organisational requirements once adopted become high-level policies or responsibilities (which in turn have to be filtered down into the levels of operationalisation i.e. roles and activities).

At a rather simple level it could be said that at the agency level requirements are to do with policies and what responsibilities and rights are to be associated with them; at the agent level they are to do with making work easier, more satisfying etc. with regard to the allocation of activities and resources; and at the intermediary role level, requirements are concerned with how the organisation itself functions and what obligations and capability tokens are necessary to support this.

Expanding from the above, at the activity level model we are concerned with how agents perform their activities and with what resources. The requirements relationship would examine work issues concerned with the agent itself (as an individual); concerned with who will work with the system, what task is to be performed, the individuals relationship with IT, allocation of functions etc.

At the role model level the notation is concerned with what a role does i.e. its obligations and what capability tokens it needs to fulfil these. The requirements relationship is here concerned with how the organisation itself functions, what its work processes are, and what obligations and capability tokens are necessary to support this (i.e. what tokens need to be bestowed upon a role). This is at an abstract level and the requirements are fundamentally to do with work processes and procedures, job design, and the social aspects of working.
The requirements relationship at the level of the responsibility model is involved with how the organisation is going to be affected by systems development i.e. with organisational processes. This has to be operationalised in terms of key organisational processes, the culture of the organisation, human resources management policies and practices and the technology of the organisation. Furthermore, at this high level requirements are to do with policies and what responsibilities and rights are to be associated with them. We are here asserting that at the responsibility model level actual responsibilities are derived from the policy level, i.e. it could be said that over and above the responsibility level there exists a more strategic layer of organisational life which is concerned with policy-making, future directions for the organisation etc. It is these policies or high-level issues which, initially help set the boundary for us and, are then translated into responsibilities, agencies, and rights.

What this gives us is a platform on which we can discuss modelling and requirements issues. For instance, if a problem solver wishes to discuss a requirement at a particular level he can now place it within a modelling framework by using the language appropriate to that level. Requirements can now be discussed in the context of modelling issues and this is unique in organisational analysis making ORDIT the only methodology attempting to do this.

7 Design Process

As will be evident from the previous sections our modelling process is essentially one of transformation from policies through to responsibilities, obligations, and activities. Policies, and the perspectives and driving-forces for change they generate are essentially very high-level requirements. These can be transformed into responsibilities, obligations, and at the level of abstraction pertaining to specification, activities. And, of course, each of these three models suggest qualitatively different requirements.

We will now examine the relationships between our modelling concepts in a little bit more detail in the context of how an ORDIT analysis is actually carried out. With the aid of a role-play case study that was used to test the developing methodology we present a brief report which highlights our system modelling process. Our approach is not sequential; backtracking and iteration are essential and, in practice, modellers will be working at different levels of detail, on several stages simultaneously.
We begin with the awareness that there is a problem, that something is not quite right in the organisation. The owner(s) of the problem(s) are then determined and their concerns elicited. This is an intensive process and it is quite likely there will be many iterations here. We are effectively beginning to bound the nature of the problem: its scope, the constraints on change, the important perspectives or driving forces etc. Once these have been established in sufficient detail we proceed to the next stage where we refer to our conceptual framework. Each of the three models in our conceptual framework ignores detail irrelevant to its purpose and concentrates on the set of features it is designed to exhibit.

We may begin with responsibilities but basically our framework allows us to place whatever data we have gathered in the appropriate model (therefore, in reality we will most likely adopt a top-down - bottom-up approach). The responsibility model will help us to focus on why we are pursuing a particular perspective; the obligation model then enables us to describe in terms of roles, obligations, and capability tokens, what is going on in the responsibility model in more practical terms. The activity model facilitates a further level of detail describing how things will be done in the language of agents, actions, and resources. It is absolutely vital to the understanding of the conceptual framework that it is realised that each of the three models are applicable to any organisation at any level and to whatever entities it is composed of - whether they be departments, sections, groups, or even individuals.

8 Case Study

The "Payfund" [10] organisation is a large UK insurance agency with several thousand staff. One of its main tasks involves the calculation and issue of payments to clients according to a set of established rules. The case study related to the design of a new computer system to support the administration of these payments. Payments are made to clients through a nationwide network of local offices ranging in size from, typically, 20 - 50 staff, most of whom are in clerical jobs organised under Section Heads who in turn are responsible to a Branch Manager.

Policy and procedures to be carried out in each office are closely specified by Head Office, based in London, but not the way in which the local offices are organised. The main tasks carried out by staff in the local offices are to collect personal details from clients which are entered into the computer system. If specified conditions are met the system calculates the appropriate payment and issues cheques for payments direct to clients.
A decision to introduce a new bespoke computer system has been taken by the Payfund organisation. It is intended not only to replace the functions performed by the existing system but to improve the efficiency and effectiveness of the payment system and to create more rewarding jobs for staff. The role of ORDIT was to investigate organisational requirements for the proposed system and to present socio-technical design options for consideration by Payfund as a result of this.

Here, with regard to examining evolving requirements in the framework of our modelling schema, we will focus only on the job design and work organisation issues associated with the implementation of the technology.

Our client was identified as a senior executive of the Payfund organisation. He provided us with the necessary background information on our remit and as the purse-holder was responsible for all policy decisions. The running of local offices was seen as our main area of concern and here the manager was given the freedom to utilise his staff in his own way as long as the policy objectives were met. Initially we set the organisational boundary through eliciting information about the role and function of the organisation, the forces for change, issues and perspectives to be adopted, the constraints it operates under, external parties it comes into contact with etc. with the aid of our responsibility model. This provided us with an approximate overview of high-level requirements and an insight into the relationships that exist within and without the organisation. Our next step was to explore our understanding of this information by going back to the senior executive, the managers, and other stakeholders with various suggestions and proposals, the feedback from which assisted in further defining the scope for change within the organisation.

Eliciting information on the roles and requirements of all the stakeholders is the most time consuming task. The level of detail entered into is largely up to the problem solvers discretion but we would recommend that the responsibilities and obligations of each stakeholder are ascertained along with their whole range of requirements. For example, within a number of offices it was found that requirements among the stakeholders ranged from: a manager wanting more autonomy to decide on the grades of staff required in his office; a senior clerical officer wanting to concentrate more on staff assessment and training, away from routine administration; a junior clerical officer wanting more contact with clients because it was the aspect of their job they enjoyed the most, and so on.

Now we can see that we have three different types of requirements each of which corresponds to the requirements relationship in the responsibility, role, and activity models respectively. To select just one: The reasons the senior clerical
officers put forward for a change in their roles were to do with a disenchantment with
too much paperwork and a desire to be more involved with other staff. When this was
further explored with them it was found that not all the consequences such a change in
role would entail were welcomed. For instance, it would be more efficient for a local
office to train just one of its senior clerical officers in this role and this could
potentially cause conflict amongst them; a proposal which was more amenable to
them was for their duties to train and assess other staff to be recognised by
management and resourced adequately.

We have seen, albeit from this rather brief example, that ORDIT is presenting
a methodology, a crucial element of which is based on an ongoing process of
interchanges, where notions and proposals are put forward by problem solvers, based
upon an elicitation of stakeholder requirements in the context of following a particular
perspective which too is in a state of ongoing refinement.

9 Summary

Organisational change is dynamic. It is about movement from a current state
to a future desired state. We hope that we have been able to show that organisational
analysis is more than just the treatment of evolving requirements: fundamentally it is
the treatment of evolving problems and evolving solutions. Evolving and emerging
requirements provide an exploration space in which ideas and proposals are put
forward, discussed and developed, and this process is repeated until the client is
satisfied with the end result. The exploration of the problem space merges slowly into
an exploration of the solution space. We believe that our conceptual framework and
our user-centred approach pave the way forward for clearer requirements capture and
representation. The aim is not to create the system the organisation wants but one that
it needs.

Philosophically, the approach we are advocating is based upon the 'dialectic';
used by Plato to describe the method of Socrates who sought philosophical truth
through disputation, and by Hegel [11] to mean the propensity to transcend
contradictions (whereby the thesis created the anti-thesis and the resolution between
the two occurred in the synthesis which in turn became the new thesis and so on).
Evolving requirements, as the term suggests, evolve in response to perceived changes
in the environment. Once elicited, or even during the process of elicitation (on which,
unfortunately, we have not been able to say very much) the problem solver treats
requirements as states of affairs which represent aspects of organisational realities. It
is these organisational realities which are then discussed and explored further with the
stakeholder. By placing requirements in their environment of change the process of
evolution occurs as a response to perceived environmental changes and it is in this way that we hope to provide a more harmonious fit, ultimately, between requirements specifications and their owners.

The models in the conceptual framework we have presented form a set of modelling languages where the different models fit well together and can all be seen as models of different aspects of the underlying situation. We do emphasize however that it is work-in-progress and that this paper must be viewed as a signpost showing where we have come from and where we are trying to go.

10 Acknowledgements

We gratefully acknowledge the contributions made by our colleagues in the ORDIT project to the ideas presented here. ORDIT, which is an Esprit II project, includes collaborators from industry and academia with expertise both in human factors and organisational issues and software design and engineering (Algotech srl, HUSAT Research Institute, the MARI Group Ltd, the Computing Science Department of the University of Newcastle upon Tyne, and the Work Research Centre). We also gratefully acknowledge the support of the University of Newcastle upon Tyne in particular that of our colleague Tony Dale, and the financial support afforded by the ESPRIT programme of the CEC (Project 2301).

References


