4

Responsibility Modelling and its Application to the Specification of Domain Knowledge

Andrew Blyth
Department of Computer Studies, University of Glamorgan,
Pontypridd, Mid Glamorgan, CF37 1DL. United Kingdom.
Tel. No +44 1443 48 2245 Fax. No +44 1443 48 2715
Electronic Mail ajcblyth@glamorgan.ac.uk

Abstract

The most crucial aspect of software engineering is the gathering of requirements. The best source of requirements is domain knowledge and the best source of domain knowledge is stakeholders. Current requirements engineering techniques are failing to a) identify stakeholders, and b) identify why a stakeholder is a stakeholder. In an attempt to address this issue the responsibility modelling technique developed in this paper focuses upon the specification of domain knowledge through the identification of stakeholders and the specification of the roles they play and the actions they perform in an organisation.

Keywords:

Domain Knowledge Acquisition, Requirements Engineering, Responsibility Modelling, Stakeholder Analysis.

1 INTRODUCTION

Over the last two decades two things have influenced organisations more than anything else - and these are the coffee machine and the computer as they both mediate social values (Ehn 1989). The result of this is that Information technology is rapidly becoming a cognitive and social artifact (Dahlbom & Mathiassen 1994). In (Gause & Weinberg 1989) the point is made that when we ignore the human aspect of requirements engineering we create ambiguities. In (Stamper 1985) it is argued that if we do not capture the correct set of values and their associated interactions then we should not be surprised when the system fails, or to put it another way: *if you put garbage in then you will get garbage out.*

All of the above highlight the point that it is vital that when we are developing an Information System we capture and validate requirements about a) the social values that the system will be required to support, and b) the interactions through which the social values will be mediated. However finding the correct set of values and interactions that the system is required to support is a difficult thing. Values are largely constituted in the realm of a community. Finding values and interactions that are relevant for IS design implies a need to identify a group of people sharing a pool of values that define what the desirable features if an IS are and how they should be obtained (Lyytinen & Hirscheim 1987)

The best source of requirements is domain knowledge and the best source of domain knowledge is stakeholders. Both (Gause & Weinberg 1989) and (Wiener 1993) observe that the most common mistake in the requirements engineering process is to

A. Sutcliffe et al. (eds.), *Domain Knowledge for Interactive System Design*

leave a stakeholder out of that process. Stakeholders are defined as all those claimants inside and outside an organization who have a vested interest in decisions faced by the organization in adopting and utilizing information technology (Mason & Mitroff 1981). Evidence has shown that failure to involve stakeholders in the development of an information system can directly lead to the system's failure (Alter & Ginzberg 1978; Bailey & Person 1983; Gause & Weinberg 1989; Markus 1983; Wise & Debons 1987).

Current stakeholder analysis techniques do not allow us to identify social values and their associated interactions, and from this to derive requirements (Lucas 1975; Markus 1983; Franz & Robey 1984; Kling & Iacono 1984). Stakeholder analysis techniques such as (Checkland 1986; Checkland & Scholes 1990) and (Tuzhilin 1995) concentrate upon the specification and analysis of activities that people (stakeholders) perform. They concentrate upon the execution of the activity and fail to examine why the activity is important in the organisational context. Office models such as (Beslmuller 1988; Rein & Singh 1992) concentrate upon modelling systems from a functional action oriented perspective. These approaches allow us to identify and specify a distinct set of actions that a stakeholder performs. These approaches fail to identify the the other stakeholders which monitor and direct the activity. In short the fail to examine and social context within which the activity is set. Evidence and experience has shown us that analysing critical systems using such methods can and does lean to system failures (Mellor 1994). Goal modelling techniques such as (Dardenne et al 93; Sutcliffe & Maiden 93; Yu & Mylopoulos 1994) in general model goals as states which are to be maintained, achieved or avoided. These approaches view goals as measurable and quantifiable concepts. The models do not allow us to clearly a) identify and enunciate user based social values, or b) the stakeholders which are there owners and mediators

Evidence (Executive 1993a; Executive 1993b) suggests that responsibility modelling offers us the ability to a) identify and validate social values, b) identify and validate the interactions required to mediate the social values, c) identify and validate the conditionals under which we can engage in the interactions that mediate the social values, d) identify and validate the stakeholders to whom the social values and interactions belong, and e) derive requirements from the stakeholder models.

2.0 MODELLING RESPONSIBILITIES

"No object is an island. All agents stand in relationships to others on whom they rely for services and control". (Beck and Cunningham 1983)

2.1 What is a Responsibility

Evidence has shown that the responsibilities concerned with a particular problem, and the agents that hold them, can be a great source of domain knowledge (Executive 1993b; Strens & Dobson 1994; Blyth 1995b). From Figure 1 we can view a responsibility as a relationship between two agents, or stakeholders, regarding a specific state of affairs, such that the holder of the responsibility is responsible to the giver of the responsibility, the responsibility principal. The full definition of a responsibility consists of the following:

- who is responsible to whom;
- the state of affairs for which the responsibility is held;
- a list of roles held by the responsibility holder (how the responsibility can be fulfilled);
- the type of responsibility (these include accountability, culpability, legal liability).

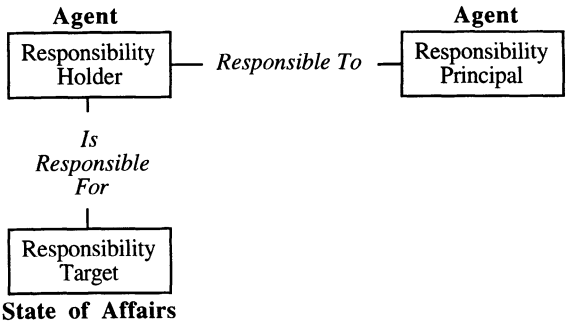

Figure 1 The Responsibility Relationship

2.2 Responsibilities, Roles and Actions

This brings us to the distinction between responsibilities, roles and actions. We use these concepts in the sense that agents execute actions in order to perform roles imposed on them by virtue of the responsibilities they hold. These roles are what the agents have to do and effectively describe their *'jobs'*. Roles are performed with regard to other agents. For example, in (Blyth 1995a; Blyth and Chudge 1995) a health care based case study is presented in which the role of *Problem-Report* stands in relationship to *Problem-Reception*. Roles are the link between their responsibilities and the actions they execute. Another way of describing this relationship is to say that responsibilities tell us *why* agents do something, roles tell us *what* they do and actions are *how* they do it. The distinction between responsibilities and roles is apparent from the words we use: a responsibility is *for* a state of affairs, whereas a role is *to do* something that will change or maintain that state of affairs. Roles only have meaning in relationship to other roles. For a selection of the types of relationships that can exist between roles the reader is reerred to (Blyth et al 1993). The distinction between roles and actions is that roles define *what* has to be done rather than *how* it should be done. As such we regard roles as an abstraction away from actions. Actions are defined as operations that change or maintain the state of the system or affect the outside world. There are three types of actions that a agent may perform: a physical action, a mental action, and a communication action. A physical action is an action that changes the physical state of the world, and an example of such an action is moving an object from place A to place B. A mental action is an action that changes the mental state of the agent, and an example of such an action is a decision. A communication action is an action through which agents interact with each other, and an example of such an act is an utterance.

When examining a responsibility for the purpose of defining the actions through which the responsibility is fulfilled we must define the following

- What roles the agents holding the responsibility perform,
- Under what conditions they perform the roles,
- What action the agents performing the role execute, and
- Under what conditions they execute the actions.

Figure 2 depicts the structure of the responsibilities with regard to the performance of roles, and also depicts the structure of the roles with regard to the performance of actions. Each component in Figure 2 is a propositional sentence and we can therefore use propositional logic to glue sentences together. From Figure 2 we can see that a responsibility implies a set of guarded roles. The guards are used to specify the prerequisites under which the role may be performed. For example, the agent Peter may be responsible for receiving and evaluating complaints about dentists. This

responsibility would involve Peter executing the roles of problem reception and problem evaluation. The guards would allow us to express the fact that the roles may only be performed between 9:00 and 17:00 hours.

In addition, Figure 2 also implies that a role is performed by an agent. Consequently we may ask and answer the question: "Is the agent that performs a role the same agent that holds the responsibility?" For example, within a university a Head of Department may be responsible for hiring and firing staff. In order to fulfill this responsibility two roles have to be performed. The first role is that of interviewing an applicant, and this role may be performed by a number of people including the Head of the Department. The second role is that of offering the job to the applicant, and this role may not be performed by the Head of Department, but may be performed by some person in the university's administration. Thus by examining who performed the roles involved in discharging a responsibility we may identify additional stakeholders, and consequently build up confidence that the final delivered system will meet all the needs of its stakeholders and thus be fit for purpose.

Examination of the roles associated with the fulfillment of responsibilities has shown that they fall into two broad categories. The first category is concerned with the delegation of the responsibility to some other agent. The second category of roles is concerned with the performance of some activity, the result of which is the discharging of the role and the fulfillment of the responsibility. From the specification of this role it is possible to derive a set of interactions. For example, the role of ambulance controller involves the execution of the conversation that schedules an ambulance for a job.

In addition, from Figure 2 we can see that a role implies a set of guarded actions. A guard are used to specify the conditions under which the action is executed. For example, the role of problem evaluation which is performed by Peter may involve the making of a mental action (decision) about whether the complaint is valid or not. This action may only be performed once Peter has collected data from both the Dentist and the Complainant, and has access to that data.

Responsibility_1 **is fulfilled by** [
$[\text{Guard}_1]$ **pre** Role_1 **is performed by** $\text{Agent}_1 \wedge$
$[\text{Guard}_n]$ **pre** Role_n **is performed by** Agent_n].
Role_1 **is discharged by** [
$[\text{Guard}_1]$ **pre** Action_1 **is executed by** $\text{Agent}_1 \wedge$
$[\text{Guard}_n]$ **pre** Action_n **is executed by** Agent_n].

Figure 2 - The Structure of Responsibilities and Roles

In addition, Figure 2 also implies that an action is executed by an agent. Consequently we may ask and answer the question: "Is the agent that executes an action the same agent that performs the role?" For example, in the U.K. when a person phones for an ambulance, from the perspective of the ambulance service they are performing the role of problem reporter. This role involves the execution of several actions. The first action is the communicative action of stating a) why an ambulance is required, b) where an ambulance is required, and c) where the problem reporter is phoning from. All three of these communicative actions are executed by the problem reporter. The second action is the communicative action of stating the location of the phone which is being used by the problem reporter, and this action is executed by the British Telecom operator.

Actions are considered to be of three basic types (See Figure 6). Communicative actions are spoken or written utterances that result in meaning being assigned to a linguistic expression (Searle 1969; Searle & Vanderveken 1985). Communicative acts

always involve at least two agent roles: speaker and hearer (though these can on occasion be the same individual). Communicative acts form larger wholes called *conversations*, which exhibit systematic regularities that can be studied and analysed. An example of a conversation is the process of authenticating someone; it consists of a number of communicative acts such as requesting a token of identification, confirming its authenticity with an authority, and finally accepting the individual as genuine (or not). Mental actions are actions which change the mental state of the agent making them. These actions can only be performed by one agent and can be either deterministic or non-deterministic. An example of a mental action is a pilot deciding to land a plane or subordinate deciding to perform an other activity. Physical actions are deeds performed by human or mechanical agents. Their preconditions, initial states, and results can thus be described by referring to states in the entity domain. An example of an physical act is that of delivering some goods to a customer.

3.0 THE CASE STUDY

3.1 Background Information

The information about the London Ambulance Service contained in this paper is taken from (Commns 1995; Flowers 1994; Page 1993; Wells 1995). The London Ambulance Service (LAS) was founded in 1930 and is currently managed by South West Thames Regional Health Authority (RHA). It is broadly divided into an Accident and Emergency Service (A&E) and a non-emergency Patient Transport Service (PTS). LAS covers a geographical area of just over 600 square miles, and its area of operations is broadly coterminous with the London Fire and Civil Defence Authority and Metropolitan Police. It is the largest ambulance service in the world. It covers a resident population of some 6.8 million, but its daytime population is larger, especially in central London. The LAS carries over 5,000 patients every day and it receives between 2000 and 2500 calls daily; this includes between 1300 and 1600 999 calls. In addition, the London Ambulance Service makes 0.5 million A&E patient and 1.3 million PTS journeys each year. The London Ambulance Service is financed on the A&E tier by service agreements with four Thames RHAs, and via contracts with some 80 hospitals and community units for PTS. In 1992/3 its budgeted income was 69.7 million: 53 Million for A&E services and 16.7 million for PTS services.

Responsibility Label: *Responsibility-One*		
Responsibility Principal	**Responsibility Holder**	**Responsibility Target**
South Thames Regional Health Authority.	London Ambulance Service	Provision of emergency ambulance service across the Greater London Area.

Figure 3 - The Definition of the Responsibility

Figure 3 states that the London Ambulance Service is responsible to South West Thames Regional Health Authority for the provision of emergency ambulance service across Greater London. Figure 3 also states the label *Responsibility-One* will be used to refer to the responsibility. Within this case study I will focus upon the resource dispatch system.

3.2 A Description of the Dispatch System

When an urgent call is received in the Central Ambulance Control the Control Assistant (CA) writes down the details of the call on a pre-printed incident form (AS1). The incident location is then located and the map reference points are recorded on incident form. On the form the Control Assistant records the details of the incident, the name and address of the caller, and the location of the phone used by the caller. This form this then passed to a Central Resource Allocator which decides which of the three London Divisions - North East, North West and South, are going to deal with this call. It is at the point of allocating calls to divisions that duplicate calls are removed.
The division resource allocator examines the form and, using the status and location information provided through the radio operator and noted on forms maintained in the activation box for each vehicle, decides which resource should be mobilised. This resource is then also recorded on the form which is passed to the despatcher. The despatcher will telephone the relevant ambulance station (if that is where the resource is located) or will pass the mobilisation instructions to the radio operator if the ambulance service is already mobile. The ambulance crew then proceed to the incident location and deliver the emergency service.

3.3 Modelling the Fulfillment of the Responsibility

From the problem described in section 3.2 we see that there is a clear process through which the responsibility for the delivery of emergency ambulance service is discharged. This process involves a series of agents performing a series of roles. Figure 4 is derived from section 3.2 and it shows us how the London Ambulance Service fulfil their responsibility for the delivery of emergency ambulance service. Figure 4 depicts the roles that various agents perform and the sequence in which the roles are performed.
In Figure 4 we can see two sequencing operators at work. The first operator is the parallel operator which is depicted by the :: symbol. This symbol tells us that the roles on either side of the operator are performed in parallel. For example, in Figure 4 we can see that the role of *Problem-Report* is performed in parallel with the role of *Problem-Reception.* The second operator is the sequential operator which is depicted by the : symbol. This symbol tells us that the roles on either side of the operator are performed one after the other. For example, in Figure 5 we can see that the role of *Problem-Allocator* is performed before the role of *Resource-Allocator.* When the operators are combined the parallel operator has priority. For example, in Figure 4 we can see that the role of Problem-Allocator is performed after the roles of *Problem-Report* and *Problem-Reception.*

Responsibility-One **is fulfilled by** [
[Guard$_1$] **pre** Problem-Report **is performed by** Citizen ::
[Guard$_2$] **pre** Problem-Reception **is performed by** Control-Assistant :
[Guard$_3$] **pre** Problem-Allocator **is performed by**
Central-Resource-Allocator :
[Guard$_4$] **pre** Resource-Allocator **is performed by** Control-Assistant :
[Guard$_5$] **pre** Problem-Notifier **is performed by** Dsipatcher :
[Guard$_6$] **pre** Health-Care-Deliver **is performed by** Ambulance-Crew :

Figure 4 - Discharging the Responsibility

We can analyse the responsibility in several ways. Firstly we can analyse the responsibility by examining the guard associated with a role and asking the question,

"what are the preconditions and prerequisites under which agents perform their roles?" For example, from Figure 4 we can ask the question, "what are the preconditions and prerequisites for the *Citizen* agent to perform the *Problem-Report* role?" The answer to this question is that the performer of the role deems there to be a legitimate need on behalf of themselves or a third party for an emergency ambulance. We can also ask the question, "to whom is the agent performing the role held responsible?" In this way we are a) defining a new responsibility - See Figure 1, b) identifying stakeholders, and c) making the responsibility model recursive.

We can also analyse the responsibility model by examining the relationships that exist between the various role holders. For example, if we examine the relationship that exists between the Citizen and the Control Assistant (See Figure 4) we can see that the relationship is one of service. The service relationship is such that the citizen invokes the delivery of a predefined service to a third agent. We can use the relationships that exist between the various role holders to define the nature of the interactions and information flows by which the roles are discharged. For instance, from the above example the nature of the relationship directs us to examine what information must flow between the Citizen and the Control Assistant in order for the Citizen to invoke the service of provision of emergency ambulance service.

In addition, we can also analyse the responsibility by examining and defining what actions are required in order for a role to be discharged. Figure 5 shows us some of the actions by which the Control Assistant discharges the role of Problem Reception.

Problem-Reception **is discharged by** [

[Guard$_1$] **pre** Call-Reception **is executed by** Control-Assistant :

[Guard$_2$] **pre** Provide-Details **is executed by** Citizen ::

[Guard$_3$] **pre** Record-Details **is executed by** Control-Assistant :

[Guard$_4$] **pre** Transfer-Job **is executed by** Control-Assistant :

Figure 5 - Discharging the Role of Problem-Reception

In Figure 5 we can see that the role of *Problem-Reception* begins with the Control Assistant performing the action of Call-Reception. The Citizen and the Control Assistant then perform the roles of *Provide-Details* and *Record-Details* in parallel. The role is finally discharged when the Control Assistant transfers the job to the Central Resource Allocator.

One way in which we can analyse the role depicted in Figure 6 is by examining the guards associated with the various actions. If we examine the guard associated with the action of *Record-Details* we can ask the question, "what are the pre-conditions and pre-requisites for the action?" From the answer to this question we can derive requirements that will feed straight into the systems specification. In order for the Control-Assistant to be able to execute the role of Record-Details the Control-Assistant must be able to a) create a form on which to record the data, b) edit (read and write) the newly created form, and c) commit the form to the information store. From the above description of the requirements of the Control-Assistant we can deduce the requirement that Control-Assistant does not have the right to destroy a form once it has been created.

We will now examine the actions shown in Figure 5 in a little more detail. From Figure 6 we can see that the action *Recording-Details* is a communicative action. This communication comes from the Control-Assistant and is directed at the Information-System. There are several points that should be noted about the communicative act. The first is that the executor of the act is the same agent who produces the communicative act. If the executor of the act and the agent who produces the communicative act had been different then this would have point to the executor of the act acting as a surrogate for another agent and stakeholder. For example, when the

postman delivers a bill from the tax-man, the postman is executing the communicative act of Deliver-Bill on behalf of the tax-man.

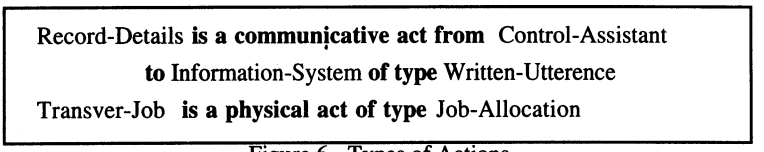

Figure 6 - Types of Actions

It is possible for the same act to be of different types at the same time. For example, from the perspective of the postman when the postman delivers a bill from the tax-man it is a physical act. Yet the same act from the perspective of the tax-man is a communicative act. It is important that when we identify an act that is of two or more different types at the same time that we ask the question and answer the question, "Do the people who are involved with this act understand the different ways in which the act is being interpreted?"
The communicative act depicted in Figure 6 is of type Written-Utterance. This type of act tells us that some information is recorded in some permanent form. Consequently we can ask the question, "what resource will be used to record the utterance and who has access rights and control over that resource?"

5.0 CONCLUSIONS

It is my experience that constructing responsibility models and determining their bindings and boundaries aids us in suggesting issues of domain knowledge and exploring stakeholder issues in a manner that pays heed to an organisation's requirements 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. I claim that the responsibility modelling techniques provide a sufficiently rich environment in which organisational structure and the roles of stakeholders, information flow, resource management, and the relationships between all of these are capable of being represented.
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. The notations presented in this paper all stem from the responsibility model presented in Figure 1. This model acts as the source from which all the other notations flow. All of the models presented in this paper are bi-directional thus from an activity model it is possible to work backwards and to define what responsibility this activity supports. I have found that this close coupling of the various notations greatly aids the problem owners and problem solvers in the process of requirements engineering.
To sum up, the responsibility modelling approach recognises that the process of elicitation and representation of requirements through domain knowledge is a process that is best accomplished by combining the social aspects of a system with the functional, and the approach also adheres to the principle of giving the customer what the customer needs and not what the system designers think the customer wants.

6.0 REFERENCES

Alter, S., & Ginzberg, M. J. (1978). Managing Uncertainty in MIS Implementations. *Sloan Management Review*, **19**, 23 - 31.

Bailey, J., & Person, S. (1983). A Tool for Computer User Satisfaction. *Management Science*, **29(5)**, 530 - 545.

Beck, K., & Cunningham, W. (1989). A Laboratory for Teaching Object Oriented Thinking. In *Object-Oriented Programming Systems Languages and Applications (OOPSLA)*, (pp. 1 - 6). ACM Press.

Beslmuller, E. (1988) *Office Modelling Based on Petri Nets*. ESPRIT 1988 Part 1. North-Holland. 977 - 987.

Blyth, A. (1995a). Modelling the Business Process to Derive Organisational Requirements for Information Technology. *ACM SIGOIS Bulletin*, **16(1)**, 25 - 33.

Blyth, A. (1995b). Responsibility Modelling and Its Application to the Management of Change. *Focus on Change Management - Cases in Business Process Re-Engineering*, **17**.

Blyth, A., & Chudge, J. (1995). Modelling and Eliciting Organisational and Information System Requirements. In *International Medical Informatics Association 8th World Congress on Medical Informatics (MEDINFO)*. Vancouver, British Columbia, Canada.

Blyth, A., Chudge, J.,Dobson, J., & Strens, R. (1993). ORDIT: A New Methodology to Assist in the Process of Eliciting and Modelling Organisational Requirements. In S. Kaplan (Ed.), *Organisational Computing Systems (COOCS)*, . Milpitas, California, USA: ACM Press.

Checkland, P. (1986). *Systems Thinking, Systems Practice*. Wiley.

Checkland, P. & Scholes, J. (1990). *Soft Systems Methodology in Action*. Wiley.

Commons, H. o. (1995). *Health Committee Second Report London Ambulance Service*. Her Majesties Stationary Office (HMSO).

Dahlbom, B. & Mathiassen, L. (1994). *Computers in Context: The Philosophy and Practice of Systems Design*. NCC Blackwell.

Dardenne. A., van Lamsweerde, A. & Fickas, S. (1993) Goal-directed Requirements Acquisition. *Science of Computer Programming*. **20(1-2)** 3 - 50.

Ehn, P. (1989). *Work-Oriented Design of Computer Artifacts*. Arbetslivscentrum.

Executive, N. M. (1993a). *Integrated Clinical Workstation: User Requirements (Acute Hospital) - Part A*, Report No. F6018/ICWS/135. NHS Centre for Coding and Classification.

Executive, N. M. (1993b). *Integrated Clinical Workstation: User Requirements (Acute Hospital) - Part B*, Report No. F6018/ICWS/136. NHS Centre for Coding and Classification.

Flowers, S. (1995). The London Ambulance Service Computerised Despatch System. In *Mega Mistakes - Lessons From Information System Failures* J. Wiley.

Franz, C. R., & Robey, D. (1984). An Investigation of User-Led System Design: Rational and Political Perspectives. *Communications of the ACM*, **27(12)**, 1202 - 1209.

Gause, D. C., & Weinberg, G. M. (1989). *Exploring Requirements: Quality Before Design*. New York: Dorset House Publishing.

Kling, R., & Iacono, S. (1984). The Control of Information Systems Development After Implementation. *Communications of the ACM*, **27(12)**, 1218 - 1226.

Lucas, H. C. (1975). *Why Information Systems Fail*. New York: Columbia University Press.

Lyytinen, K., & Hirscheim, R. A. (1987). Information System Failures:A Survey and Classification of Empirical Literature. *Oxford Surveys in Information Technology*, **4**, 257 - 309.

Markus, M. L. (1983). Power, Politics and MIS Implementation. *Communications of the ACM*, **26(6)**, 430 - 444.

Mason, R. O., & Mitroff, I. T. (1981). *Challenging Strategic Planning Assumptions*. New York: John Wiley and Sons.

Mellor, P. (1994), CAD: Computer Aided Disaster. *High Integrity Systems*, **1(2)**, 101 - 156

Page, D. (1993). *Report of the Inquiry into the London Ambulance Service.*, Communications Directorate South West Thames Regional Health Authority.

Rein, G. L. &Singh, B. (1992) Interaction Nets (RINs): A Process Description Formalism. Technical Report Number CT-083-92. Micro-Electronics and Computer Technology Corporation (MCC).

Searle, J. R. (1969). *Speech Acts - An Essay in the Philosophy of Language.* Cambridge University Press.

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

Stamper, R. (1985). Management Epistemology: Garbage In, Garbage Out. (And What About Deontology and Axiology ?). In *Knowledge Representation for Decision Support Systems* (pp. 55 - 77). Elsevier Science.

Strens, R., & Dobson, J. E. (1994). Responsibility Modelling as a Technique for Requirements Definition. *Intelligent Systems Engineering*, **3(1)**, 20 - 26.

Sutcliffe, A. G. & Maiden N. A. M. (1993) Bridging the Gap: Policies, Goals and Domains. *Proceedings of the Seventh International Workshop on Software Specification and Design.* IEEE Computer Press.

Tuzhilin, A. (1995). Templar: A Knowledge-Based Language for Software Specification Using Temporal Logic. *ACM Transactions on Information Systems*, **13(3)**, 269 - 304.

Wells, W. (1995). *Report on the London Ambulance Service*,. South Thames Regional Health Authority.

Wiener, L. R. (1993). *Digital Woes: Why We Should Not Depend on Software.* Addison-Wesley.

Wise, J. A., & Debons, A. (Ed.). (1987). *Information Systems: Failure Analysis,* Springer-Verlag.

Yu, E. S. K. & Mylopoulos, J. (1994) Using Goals, Rules and Methods to Support Reasoning in Business Process Re-Engineering. *Proceedings of the Twenty Seventh Hawaii International Conference on System Sciences.*