8 Models for Responsibility Assignment

Ian Sommerville

8.1 Introduction

Responsibility assignment modelling is concerned with developing a picture of how the responsibilities in a socio-technical system are distributed across the different automated elements and actors in that system. At this stage, we are not concerned with the details of the responsibilities themselves, nor with what the actors in the system have to do to discharge these responsibilities. Rather, a responsibility model presents a succinct picture of 'who is responsible for what' that can be used to identify responsibilities that have not been assigned, responsibilities that have been misassigned and actors in the system who may be overloaded with responsibilities. We argue that these models have a role to play in identifying sources of undependability in a system. They can be used to help identify requirements that are inconsistent with the responsibility structures and to design robust and reliable operational processes.

The primary use of responsibility assignment models is to serve as a basis for facilitating discussions on how responsibilities are distributed in an existing system and for planning the responsibility structure of new systems. In any system, there is some flexibility over 'who does what' and individual responsibilities are always subject to negotiation. By making responsibilities explicit, a responsibility assignment model allows designers, users and managers to develop a shared understanding of the responsibility structure in a system. This helps designers understand who needs what information and when they need it. In addition, the responsibility assignment model may be a useful supporting mechanism for identifying possible responsibility vulnerabilities in a system.

Responsibility negotiation occurs during the specification and design phases of a socio-technical system. The system designers must negotiate with stakeholders to decide how the responsibilities associated with the system are distributed. Clearly, there are trade-offs to be made between automated and manual tasks. However, issues such as organisational structures and politics also affect the distribution of responsibilities.

Responsibility renegotiation occurs when some of the actors in a system renegotiate their assigned responsibilities so that the distribution of responsibilities in the system is changed. This may be required because the organisational structure has changed, because some actors are over or under-loaded, because of changes in the people in a system, or because of new responsibilities that emerge as a system is deployed and used. The role of a

responsibility assignment model is to show the actual responsibility structure that has to be changed and, critically, to reveal the relationships between causal and consequential responsibilities in the system. While causal responsibilities can be renegotiated by the actors involved, changing the consequential responsibility structure in a system is often more difficult. It will certainly require management involvement and, in regulated situations, may require external approval.

Generally, in socio-technical systems operation, a design assumption is that some actor or component is responsible for something and will properly discharge this responsibility. Under some circumstances, this assumption may be invalid. A responsibility vulnerability is a system state that can lead to a situation where some responsibility is not properly discharged and, as a consequence, a failure of the broader socio-technical system ensues. We have probably all encountered problems due to responsibility vulnerabilities. For example, some actor in a system may make clear that they did not realise they were supposed to do something or that they did not have time to discharge some responsibility.

There are six types of responsibility vulnerability:

1. *Unassigned responsibility*. Within a socio-technical system, the responsibility for some critical task is not assigned to any agent. This is most common in circumstances where the system designers only consider what normally happens and do not think of how exceptions are handled. When such exceptions arise, it is not clear who should take responsibility for dealing with them.

2. *Duplicated responsibility*. This occurs in a system when different agents believe that they are the holder of some responsibility and each acts to discharge that responsibility. If each agent interprets the responsibility in exactly the same way, then this simply results in inefficiency. If, however, they interpret it differently, inconsistent information may be created and problems may arise when one agent interprets information created by another.

3. *Uncommunicated responsibility*. In this situation, there is a formal assignment of responsibility (typically to a role) but this is not communicated to the agent assigned to that role. Therefore, they are not aware that they should discharge that responsibility.

4. *Misassigned responsibility*. In this situation, the agent who is assigned the responsibility does not have the competence or resources to discharge the responsibility. Therefore, the proper discharge of the responsibility cannot be guaranteed. To reason about misassigned responsibilities, you need to understand something about the nature of the responsibility (discussed in Chapter 9) as well as the responsibility.

5. *Responsibility overload.* This vulnerability arises when the agent who is assigned a set of responsibilities does not have the resources to properly discharge all of these responsibilities. This is particularly likely to arise when an agent must handle exceptions that arise at the same time as other responsibilities that they must discharge.

6. *Responsibility fragility*. This occurs when a critical responsibility is assigned but there is no backup assigned who can take over if the responsibility holder is unavailable. This is a particular problem for time-critical responsibilities where there is not an option of simply delaying the responsibility discharge until the holder becomes available again.

I return to a discussion of how responsibility assignment models may be used to identify responsibility vulnerabilities later in the chapter. In the remaining sections, I discuss the distinctions between causal and consequential responsibilities that are important for a responsibility assignment model and briefly describe a proposed modelling notation. I then introduce a case study associated with a hospital system and develop a model to represent the responsibilities in that case study. In the final section, I discuss strengths and weaknesses of responsibility models.

8.2 Causal and consequential responsibilities

As we have discussed elsewhere in the book, there is an important distinction between consequential and causal responsibility. Consequential responsibility reflects who gets the blame or credit for the occurrence of some state of affairs. Consequential responsibility can only be assigned to a person, a role or an organisation – automated components cannot be blamed. Causal responsibility reflects who or what is responsible for making something happen or avoiding some undesirable system state. It is often the case that these are separated. The holder of a consequential responsibility may assign the associated causal responsibility or responsibilities to some other actor or component in the system.

When modelling the assignment of causal and consequential responsibilities, we need to represent the responsibility itself, the agents (actors or automated elements) in a system and the nature of the relationships between responsibilities and agents. However, what we mean by a responsibility depends on whether we are talking about consequential or causal responsibility.

Typically, consequential responsibilities are expressed in fairly broad terms. For example, we might see statements of responsibility such as:

§ The security officer is responsible for all aspects of building security.

§ The IT manager is responsible for all aspects of computer security.

§ The sales director is responsible for ensuring that current sales targets are reached.

For causal responsibility assignment modelling, however, these are too vague and it is necessary to recast these into statements that are more specific. Generally, it is possible to do this by restating the responsibilities in terms of one or more goals that must be attained. For example, the goals associated with building security might be:

1. No unauthorised person should gain entry into a controlled area.

2. Authorised persons should be permitted access to controlled areas according to their authorisation.

3. No injury to people or damage to property should result from the use of security equipment or procedures.

Implicitly, when we identify a goal, we create a responsibility to ensure that the goal is satisfied. Failure to ensure that a goal has been satisfied is a consequential responsibility failure. Whether or not the holder of the responsibility should be assigned blame for a failure, depends on whether or not they have done all that might be expected to discharge the responsibility. For example, if a responsibility holder has followed procedures but, due to some external agency, a failure arises then no blame should be attached to the individual. The blame may be assigned, perhaps, to the designer of the procedures or the person who authorised their use. On the other hand, if procedures have not been followed, then the responsibility holder should probably take the blame, unless there are good reasons why they could not follow these procedures.

The high-level goals are generally decomposed into a set of more detailed sub-goals and the consequential responsibility for these sub-goals may be assigned to different agents. Therefore, the goal of ensuring that no unauthorised person should gain access to a controlled area may be decomposed into the sub-goals of:

1. Maintaining perimeter security for all controlled areas.

2. Detecting any attempted or successful access to controlled areas by intruders.

3. Maintaining an identification system for authorised persons who may access controlled areas.

4. Limiting the damage or losses that might arise if an unauthorised person gains access to a controlled area.

In order that some authority (person or organisation) can decide whether or not there has been a consequential responsibility failure and whether or not blame should be attached to the holder of the responsibility, there must be some associated evidence associated with each goal. This evidence demonstrates what has been done to ensure the correct discharge of the responsibility. Therefore, for the first goal above, the evidence that shows that no unauthorised person should gain entry might include:

§ Logs which show who entered and left the building by the permitted doors.

§ Damage reports which show any forced entry to buildings.

§ Security reports which show any doors left unlocked or windows left open.

At some level of decomposition, it becomes appropriate to associate causal responsibilities with goals or sub-goals. Causal responsibility is, generally, a more detailed notion and, as we argue in Chapter 9, some causal responsibilities can be partially described as a process or workflow. The workflow sets out the actions that may be taken to discharge the responsibility. However, there may be

flexibility in how an agent discharges a responsibility so the workflow is indicative rather than definitive. This means that it shows one possible way of discharging the responsibility but some agents may discharge that responsibility in a different way, depending on their competence and experience.

Statements of causal responsibility might be:

§ A janitor is responsible for checking, every two hours, that all doors are locked outside of normal working hours.

§ The system manager is responsible for taking daily system backups.

§ The sales manager is responsible for producing monthly sales reports for the sales director.

Implicitly then, when we create a process, a causal responsibility exists to ensure that the process is properly enacted. Where this process is associated with a goal, it is assumed that the consequential responsibility associated with the process (i.e. who gets the blame if the process 'fails') falls, by default, on the agent who is consequentially responsible for the goal. However, this may be overridden by an explicit assignment of consequential responsibility. For example, the process designer rather than the process enactor may be responsible in the event of failure.

In some cases, these causal responsibilities may be associated with the evidence required to support consequential responsibilities. For example, a receptionist in a building may have the (causal) responsibility to maintain a log of all visitors who do not normally work in a building. In general, several causal responsibilities may be associated with each goal.

In summary, this discussion has suggested that key elements in a responsibility assignment model are goals associated with consequential responsibilities, and evidence and processes associated with causal responsibilities. These elements are associated, through responsibility relations, with agents, which may be individuals, roles, groups or organisations or, for causal responsibility, automated system components.

8.2.1 Authority

The notion of authority is fundamental to discussions of responsibility. When we say "X is responsible for Y", there is an implication that some authority exists who can decide whether or not that responsibility has been properly discharged. This authority, of course, is not necessarily a named individual. It can be a more diffuse body such as an organisation or society itself. In such cases, where there are allegations that a responsibility has not been properly discharged, formal procedures are invoked (e.g. there may be a legal enquiry) to decide how blame should be allocated.

Authority is important for both consequential and causal responsibility:

§ For consequential responsibility, the authority will assign the blame in the event of failure or (perhaps) praise if the responsibility has been successfully discharged. From a dependability point of view, this means that those who are in authority must be made aware of their responsibilities

and there must be procedures in place to ensure that they have sufficient information to decide whether or not these responsibilities have been properly discharged.

§ For causal responsibility, the authority can make decisions about responsibility transfer. If a holder of a responsibility is not available or unable for some other reason to discharge the responsibility, the authority should decide how that responsibility should be re-allocated. In this discussion, I assume that the authority for a causal responsibility receives a report from the responsibility holder on the discharge of the responsibility. In situations where an agent is both causally and consequentially responsible, the normal assumption is that the causal authority is also the consequential authority. However, this is not necessarily always the case.

Authorities are explicitly related to responsibilities rather than to the holders of these responsibilities. This can result in vulnerabilities when the authority structure does not match the management structure in an organisation. There are two possible types of vulnerability:

1. *Responsibility without authority*. This occurs when a holder of a responsibility does not have management authority over others to ensure that tasks necessary to discharge the responsibility are completed. For example, a manager may have the responsibility of reducing costs in his or her department. However, staffing decisions may be made centrally in the organisation – such a situation is common in government and public sector organisations. Individual managers may not have the authority to make staff redundant to discharge their responsibility of reducing costs.

2. *Conflicting authorities*. An agent may have the responsibility to complete some task but their manager may not be the authority responsible for that task. The agent then may have conflicting demands – from the authority for the responsibility and from their manager. If they give priority to their manager's instructions, then the responsibility may not be properly discharged. This is a relatively common situation which frequently occurs when people are assigned responsibilities that cut across the structure of an organisation.

   For example, in a hospital, the authority for a responsibility concerned with updating patient records may be the hospital records manager. However, nurses, who have to update patient records, may report to a nursing manager whose prime concerns are clinical rather than administrative. The nurse may therefore not give priority to the record management responsibility as this is outside the remit of their manager.

8.2.2 Shared responsibilities

Before going on to discuss the notation that you can use to model the assignment of responsibilities, there is one further idea that must be introduced. This is the notion that responsibilities may be shared. That is, the causal (or, more rarely, the

consequential) responsibility is not assigned to a single agent but to multiple agents who collaborate to discharge the responsibility.

There are 3 types of shared responsibility:

1. *Joint responsibility*. This is a situation where a causal responsibility is assigned to more than one agent and these agents have the autonomy to decide how to discharge that responsibility. They may renegotiate how to discharge the responsibility as circumstances change. In some cases, consequential responsibility may also be a joint responsibility – the agents assigned the responsibility all take the blame or praise in the event of failure or success.

   For example, a team of three people may have the responsibility to produce a newsletter. They decide amongst themselves who does the writing, who does the layout, etc.

2. *Divided responsibility*. This means that under some circumstances, the responsibility is assigned to one agent but, under different circumstances, it is assigned to an alternative agent. Divided responsibilities are most often used when exceptional situations arise and the agent assigned the responsibility does not have the competence, resources or authority to discharge the responsibility.

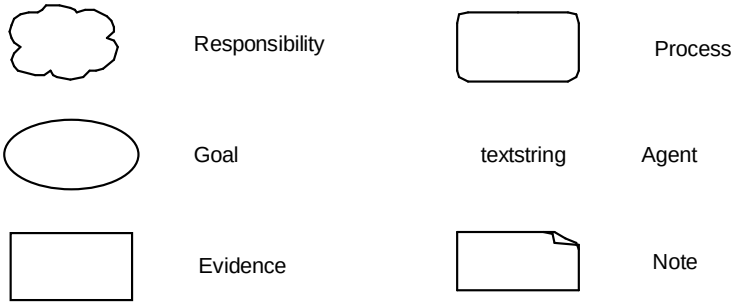
   For example, a junior doctor on night duty in a hospital may have the responsibility to ensure that proper medical treatment is prescribed for patients whose condition deteriorates. However, they may not have the competence to deal with some situations and, if these arise, they must call on a more experienced doctor to take over the responsibility for that patient.

3. *Delegated responsibility*. This is a situation where some agent, who has been assigned a responsibility, delegates that responsibility (or part of it) to some other agent. The consequential responsibility remains with the originally assigned agent. Divided responsibilities often arise as a consequence of delegation. The discharge of the responsibility in 'normal' situations is assigned to some agent but, in abnormal circumstances, the responsibility reverts to the delegating agent. Normally, when a responsibility is delegated, the delegating agent becomes the authority for the delegated part of the responsibility.

   For example, in a university, an admissions officer may delegate the routine processing of student admissions to an admissions secretary. If applicants have standard qualifications, he or she can make decisions on entry. However, if the applicant's qualifications are unusual in some way, the admissions secretary has to hand over the application to their more senior colleague to make the decision on whether or not admission should be approved.

The above examples illustrate the distinction between these different types of shared responsibilities. In the case of joint responsibilities, the actors

Figure 8.1 Entities in a responsibility assignment model



themselves negotiate how the responsibility is to be discharged. In the case of divided responsibilities, some external authority imposes rules or regulations about the limits of responsibilities but the interpretation of these rules depends on the actors in the system. In the case of delegated responsibility, one of the actors in the system decides how the responsibility will be shared. They are the authority associated with the delegated responsibility.

Issues of shared responsibility become more difficult both to analyse and to implement when the responsibility holders belong to different organisations. This may be handled formally by establishing a contract between the organisations, but often this is not sufficiently well defined, and may be entirely implicit. As discussed in Chapter 5, recourse to the law may well be the only way of clarifying the ambiguities and omissions, but this will normally only be done after some kind of failure has occurred.

8.3 The modelling notation

In this section, I introduce a graphical notation that may be used to show the assignment of responsibilities. I have chosen to use a graphical notation because this allows for responsibilities to be seen ‘at a glance’ so that informed individuals can rapidly evaluate a responsibility model. Furthermore, graphical notations are generally more accessible to and understandable by system stakeholders who are not experts in reading responsibility models. However, they do have the disadvantage that they take up a lot of space and need specialised editors to support model development.

A responsibility model includes entities of different types (nodes) and relations (links) between these entities. The representations of the entity types that may be used in a responsibility assignment model are shown in Figure 8.1:

1. The responsibility icon is used to represent some generic responsibility which may be a causal or consequential responsibility. You may use this when you are unsure of the precise details of a responsibility or where you

want to insert a placeholder for a responsibility that is decomposed and defined at a lower level in the model.

2. The goal icon is used to represent consequential responsibilities and shows the goal or goals that define that responsibility. One or more goals may be associated with a responsibility.

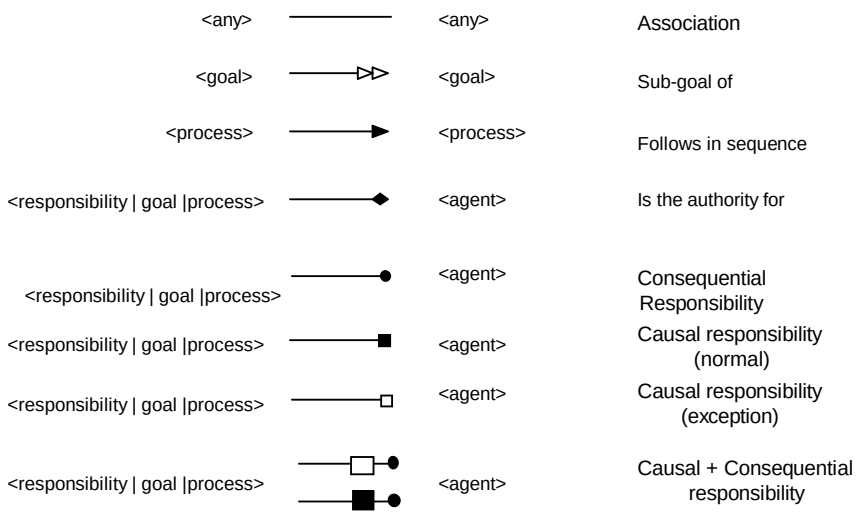
3. The evidence icon is used to represent the evidence that is collected to ensure that a consequential responsibility has been properly discharged. There may be one or more evidence icons associated with each goal icon. Evidence icons will normally be associated with at least one process icon, where the evidence (or part of the evidence) is generated during that process.

4. The process icon denotes a causal responsibility. Each goal icon may have one or more associated process icons associated with it.

5. The agent icon denotes the holder of a responsibility. For compactness, this has not been assigned an explicit graphical icon. Rather, it is simply written as a text string. If this string is written in plain text, then it names an individual responsibility holder; if it is enclosed in pointed brackets <>, then it names a role; if it is underlined, then the agent name is the name of an automated system.

6. The note icon may be associated with any other node or link in a responsibility assignment model. It is used to give any additional information that may be useful to the reader in understanding the model.

Figure 8.2 Links in a responsibility assignment model



The representations of the different links between the nodes in the responsibility model (Figure 8.2) are all derived from a simple association relation. This is represented as a line between two nodes. This line may be decorated with the following symbols:

1. A square denoting causal responsibility. The association is normally between a process and an agent. However, if a goal is not decomposed into associated processes, then a causal responsibility relationship may exist between the goal and the agent.

   If the square is filled, this means that the agent is responsible for the enaction of the process in normal circumstances. If the square is unfilled, this means that the agent is responsible for the enaction of the process in the event of some exceptional circumstance. If there is no 'exception' link associated with a causal responsibility, this means that the associated agent is responsible for the process under all circumstances.

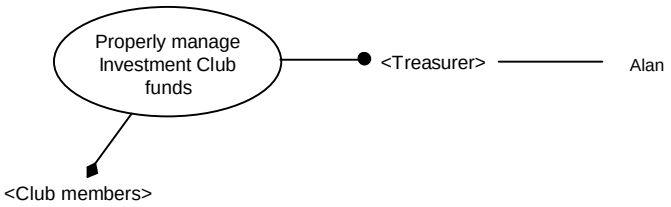
2. A circle denoting consequential responsibility. The association must be between either a responsibility icon or a goal icon and an agent.

3. A double arrow is used to link goals and sub-goals with the arrowhead pointing at the sub-goal.

4. A single arrow used to link processes with the arrowhead pointing at the process which is executed after the source of the link.

5. A diamond indicating that the agent at the diamond end of the link has authority over the agent at the source of the link.

Figure 8.3 Association of a role with a responsibility



To illustrate this notation, consider the following, very simple example.

> *Alan is the treasurer of an investment club where a group of people pool their resources to invest in shares of companies listed on the stock market. Alan's overall responsibility is the proper management of the funds of the club but they also take responsibility for the buying and selling of shares as decided by the club members. However, to ensure that these buying and selling decisions can always be enacted quickly, this responsibility is shared with Bob, the club chairman who is also authorised to make transactions on behalf of the club. To demonstrate that he has discharged his responsibility properly, Alan makes a monthly report to club members which sets out the current holdings and transactions made. Claire is responsible for tracking the performance of investments made and helps Alan prepare this monthly report by providing details of the prices of shares held by the club.*

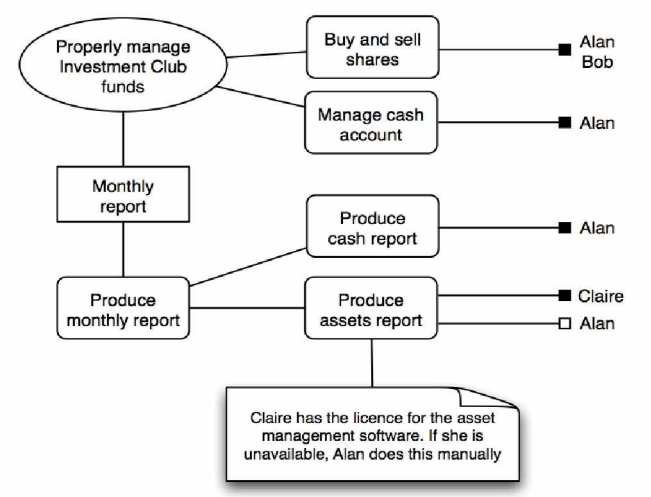
Figure 8.3 shows the goal of proper management of the funds of the club, the assignment of the consequential and causal responsibility for this goal to the Treasurer and the association of Alan with the Treasurer role. Notice that the authority for the responsibility, who decides if the Treasurer has properly discharged the responsibility, is deemed to be the club members.

Figure 8.4 shows how the goal of properly managing the club funds has three processes associated with it – managing the cash account, producing a monthly report for club members, and buying and selling shares. It shows how the causal responsibilities associated with these processes may be shared. Notice that the production of the monthly report is shared between Alan and Claire but, if problems arise in the reporting of the assets, then Alan is responsible for dealing with these.

Even from a very simple model such as that in Figure 8.4, you can identify issues that might affect the dependability of the system. For example:

1. The management of the cash account and cash reporting depends on Alan. What happens if Alan is unavailable? How do club members get the report?
2. How do Alan and Bob coordinate the buying and selling of shares?

Figure 8.4 Decomposition of high-level responsibility



It may be the case that the answers to these questions reveal that there are vulnerabilities but the club members may decide these are tolerable. However, this is then an explicit decision rather than an accidental consequence of the ways that responsibilities have been organised.

Notice that I have used two models here – one showing the consequential responsibility structure and the other showing the decomposition of consequential into causal responsibilities. In a very simply example such as this one, these could have been combined. However, in most cases, to avoid clutter and complexity, you will need separate consequential and causal responsibility models.

8.4 Bed management

As we have discussed in Chapter 5, understanding the actual responsibilities in a socio-technical system and how these responsibilities interact is not easy. We suggested that an ethnographic approach was one way to try and identify the actual distribution of responsibilities and to understand how these responsibilities were discharged by agents in the system. In this section, I describe a socio-technical system in a hospital that we have observed and illustrate the complexity of the responsibilities in that system (Clarke, Hartswood et al. 2001; Clarke, Hartswood et al. 2002; Clarke, Hughes et al. 2003; Clarke, Procter et al. 2003). In section 8.5, I illustrate how the responsibilities in that system may be set out in a responsibility assignment model.

When patients arrive in a hospital for in-patient treatment, they have to be allocated to a bed in a ward. Bed management is complex as large hospitals have a constant stream of admissions and discharges as well as planned, routine surgery and emergency treatments. In general, patients should not have to wait more than a few hours in a holding area before being assigned to a bed in a hospital ward.

Admissions fall into two classes – planned and unplanned. Planned admissions are people who have been scheduled to receive some treatment such as diagnostic investigations or surgery. Unplanned admissions are people who require emergency treatment. To accommodate unplanned admissions, it is normal to have to reorganise planned treatments – e.g. a routine operation, such as a joint replacement, may be cancelled and re-scheduled for a later date. Patients in hospital may also be discharged earlier than planned to free up a bed. The bed manager, working with clinical staff, is closely involved in the process of deciding how to make required beds available.

However, hospitals in the UK National Health Service are regulated and must meet a range of externally imposed targets. One of these targets is waiting time for routine surgery. No-one should have to wait more than a given number of months for such surgery. This complicates the process of rescheduling treatments as failure to meet these external targets can lead to financial penalties for the hospital. To meet the waiting-time target, therefore, patients whose waiting time is approaching the target time may be given priority in the assignment of a bed, irrespective of the urgency of their treatment. For example, a patient waiting for a simple operation (e.g. to remove an ingrowing toenail) who has waited a long time, might be allocated a bed before another patient who requires more significant and urgent surgery.

Within the hospital, there is an administrative role of bed manager who has the (consequential) responsibility of ensuring that incoming patients are assigned to beds. The bed manager does not have the causal responsibility of allocating beds to patients – this is the responsibility of an admissions secretary. The admissions secretary uses a bed database that tracks the status of all beds in the hospital to discover if a bed is available and to associate a patient with that bed. When a patient is discharged, the causal responsibility of updating the bed database to reflect the change in bed status falls on the nursing staff in a ward. Nursing staff also update the database when patients are moved from one ward to another. The split model where different people are responsible for related actions suggests a possible vulnerability as the successful operation of the system requires some coordination between nurses and administrative staff.

From discussions with staff in the hospital, we discovered that the information in the bed database was rarely accurate (for reasons we discuss later). The number of available beds as reported by the system did not usually reflect the number of beds that were actually available in the hospital. Nevertheless, the bed allocation system works well enough most of the time. When there are several beds available, the accuracy of the information on the bed database is not critical. The database may report that there are 6 beds available when there are actually 5 or 7 but, so long as a request for a bed can be satisfied, then it doesn't matter.

However, when the database reports that there are no beds available, then problems arise. To discharge her consequential responsibility, the bed manager must then take over the causal responsibility of finding beds for incoming patients. At this stage, we discovered that she does not trust the data from the bed database. Rather, she takes action to discover the situation on the ground rather than in the database. This may involve calling round wards to discover if patients are shortly

to be discharged, re-negotiating planned admissions or, in extreme cases, walking around the hospital to see if any beds are free. Once a bed is discovered, the responsibility can then be discharged by the bed manager.

It may appear that this problem is one that could be solved by technical means. If the dependability of the bed database was improved so that it maintained an accurate record of the number of available beds, then this problem would not arise. Our ethnographies showed that this was, almost certainly, impossible to achieve. They revealed the reasons why the inaccuracies arose (these were not generally technical faults) and it became obvious that responsibilities had a prominent role to play.

The (causal) responsibility of updating the bed database when a patient leaves falls on the nursing staff in wards. They release a bed when a patient is discharged or is transferred to another ward. However, it is an inherent part of the training of nurses to instil the notion of professional responsibility – they are responsible for ensuring that patients receive proper and timely treatment. We discovered that there were conflicts between the nurses' professional responsibility, their responsibility to update the bed database and their everyday responsibilities of caring for patients.

There were two important reasons why the bed database was generally inaccurate:

1. Nurses were slow in updating the information about bed availability.

2. Nurses deliberately did not update the information when beds became available.

Nurses, primarily, have responsibilities for patient care and, generally, they see these as their prime responsibilities. They consider them to be more important than administrative responsibilities such as updating the bed database. In the time between a patient being discharged from a ward and the database update, patients often required care and immediate attention and this distracted the nurse from the database update. As a consequence, database updates were delayed and, in some cases, completely forgotten.

This situation is predictable and understandable. It reflects a normal professional situation where the individual has to decide how to prioritise their responsibilities. The second situation, where the database was deliberately not updated, was more surprising. We discovered that it arose because of a conflict between the responsibility to update the system and the professional responsibility of 'doing the right thing' for patients.

A strategy used by the hospital to make beds available was to postpone and reschedule planned surgery where this was non-critical. Therefore, an elderly patient scheduled to have a knee joint replaced might have their operation cancelled because the bed that was assigned to them was then re-assigned to some other patient with a shorter term (although not necessarily less urgent) demand.

Nursing staff were, of course, aware of planned surgery and often knew the patients concerned from previous stays in the hospital. While the surgery may have been routine, it was important to the patient's quality of life and very distressing to have this cancelled. In some situations, the nurses used their

judgement and deliberately did not update the beds database when a patient was discharged so that the bed was not released. Rather, they delayed the update until they knew that their patient was available. This ensured that when the patients with planned operations came to the hospital, a bed would be available for them.

Here, the nurses were making judgements about which of their responsibilities should take priority and coming down on the side of professional responsibility over assigned responsibility to enact the process of updating the database. The bed manager (who had been a nurse), of course, knew of such practices and hence she used the walk around the hospital to discover beds that might be used. Interestingly, she did not consider the inaccuracies of the database to be a problem – they were part of the way in which the hospital operated. If she found an available bed, she discussed with the ward nurses whether it should be released and added to the database – there was no question of simply overriding their judgement.

Because of the responsibility conflicts, it is probably impossible to design a process where the database would always be updated immediately with accurate data on bed availability. In fact, the system as it stands allows clinical judgement to override administrative demands. Even hospital administrators recognised that this was often the best way to balance the needs of patients and administrative requirements.

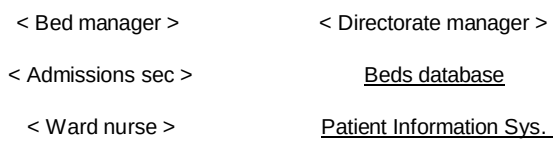
8.5 Bed management modelling

In this section, I illustrate how a model of the responsibilities in the bed management system may be developed. I also discuss how this model may be used as a means of communicating responsibilities and for highlighting issues of responsibility that may influence the system dependability.

There is no definitive process for developing a responsibility model – it depends on the knowledge of the system that you have and your access to information about the system. However, I suggest that the process should normally include the following activities, although not necessarily in the order presented here.

1. Identify the agents in the system.
2. From discussions with these agents and other information, understand the responsibilities that have been assigned to each of these agents. It makes sense at this stage to introduce the idea of causal and consequential responsibility.
3. Identify the goals and evidence that are associated with the consequential responsibilities in the system. Identify the processes required to maintain the evidence and reporting structures.

Figure 8.5 Agents in the bed management system



4. Identify the processes that are associated with the causal responsibilities in the system. Pay particular attention to exception handling.

5. Draw up a consequential responsibility model, showing the goals and associated sub-goals and the agents associated with these goals.

6. Draw up a causal responsibility model showing the processes in the system and the associated allocation of these responsibilities.

7. Check the consistency of the causal and consequential models of responsibility. Where appropriate, they may be integrated into a single model.

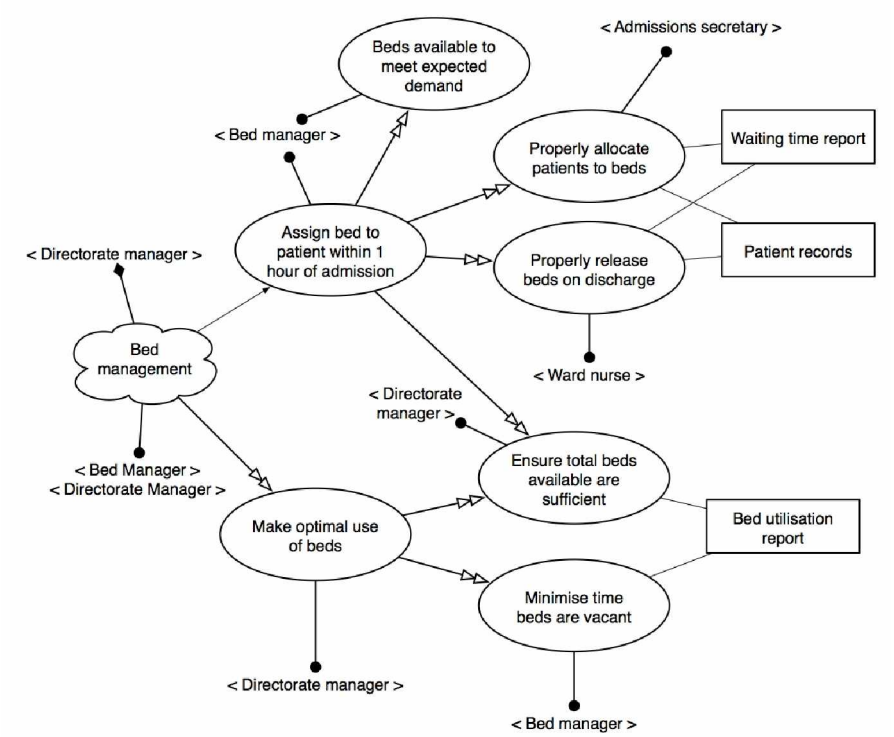
Graphical notations trade-off readability for compactness and so it may be impossible to fit all information onto a single level. In those cases, you need to break down the model into several related models, linking these through common model elements.

Figure 8.5 shows the agents in the bed management system that have causal or consequential responsibilities for decision making or information management. Notice there are two automated agents used in this system the bed management database and a patient information system (PIMS).

Figure 8.6 shows the overall goals of the bed management system, the sub-goals associated with these goals and the evidence that is required to demonstrate that these goals have been reached. By implication, the evidence that is associated with the leaves on the goal tree (e.g. properly allocate patients to beds) is also associated with the goals (e.g. assign bed to patient within 1 hour of admission) at higher levels in that tree.

Notice that the system has two goals that may, sometimes, be conflicting. The goal of assigning a bed to patients quickly may conflict with the goal of making the most effective use of beds in the hospital. This is the classic availability/efficiency problem – to guarantee availability, you need spare capacity (extra beds) but these must often be empty so this is not an efficient use of resources. Though it probably could not be used for beds, one way of dealing with this where there are a number of providers is to have a spare pool of resources assignable on demand to any provider. This pool can be managed as a shared resource or available as a service provided by a separate organisation. For example, specialised intensive care equipment is sometimes shared between hospitals.

Figure 8.6 Goals of the bed management system

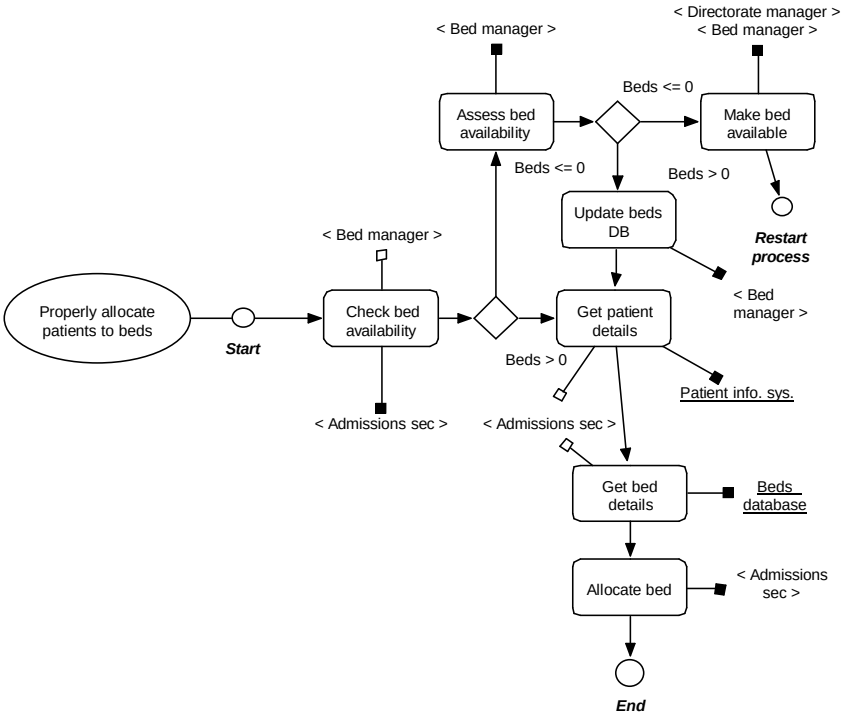


In Figure 8.6, I also show the overall responsibility of bed management using a cloud icon and illustrate that the authority for this responsibility lies with the directorate manager. The directorate manager is also an agent involved in the discharge of the responsibility. Clearly, in the event of a failure of a process involving the directorate manager, some more senior authority would be involved if there was a need to assign blame. This could be the hospital director, the management board or some external enquiry.

The next stage of the modelling process involves identifying the processes involved in the system, with the assumption that these processes are associated with a causal responsibility. For simplicity, I will focus on the sub-goal in Figure 8.6, concerned with the proper allocation of patients to beds. The causal responsibility model is shown in Figure 8.7. All causal responsibility models should start with a single goal with the processes that could achieve this goal shown in the model.

When developing a model of causal responsibilities, you describe these responsibilities as a workflow – a sequence of processes and decisions. I use a notation that is a subset of the notation used in BPML. This has been developed as

Figure 8.7 Causal responsibilities in the bed management system



a modelling language for business applications. I explain the workflow notation that I have used here in Chapter 9.

Figure 8.7 illustrates the assignment of causal responsibilities to achieve the sub-goal of properly allocating patients to beds. As I discussed in the previous section, this is a simple and straightforward process so long as the beds database indicates that beds are available. If not, then the bed manager first tried to discover if there really are beds available. If so, he or she updates the beds database. If not, the bed manager works with the directorate manager to make beds available either by cancelling treatment for which a bed has been pre-allocated or by discharging patients. Once this has been done, the bed can then be re-allocated – essentially, as shown on the model, the allocation process is restarted. Notice the role of automated agents here – they have the responsibility to provide information for their associated processes.

Responsibility problems may arise when there is a mismatch between the authority structure for responsibilities and the management structure in an organisation. These arise because authorities are associated with responsibilities but managers are associated with people. In the bed management system, one such problem arose because the responsibility for allocating and releasing beds was

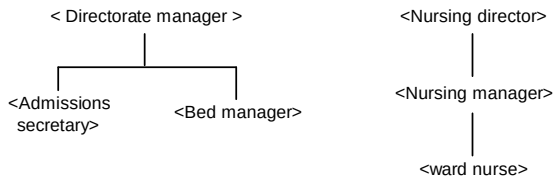
assigned to different roles (admissions secretary and nurse). The authority for the responsibility (the directorate manager) was in the same leg of the management hierarchy as the admissions secretary but the ward nurse was in a different branch of the hierarchy. The nurse's manager is the nursing director in the hospital who has a clinical role, rather than the directorate manager which is an administrative role. Figure 8.8 shows a fragment of the organisational hierarchy in the hospital. Both the nursing director and the directorate manager are at the same level but in different branches of the organisation tree. The directorate manager, therefore, cannot issue instructions to the nurse to disregard what they see as their professional responsibilities for what (the nurses see as) administrative convenience.

To highlight this type of vulnerability, you should compare the organisational management structure with the authority structure in a system. If there are serious mismatches between them, this may imply that a responsibility holder has insufficient authority to ensure that the responsibility is properly discharged. Alternatively, holders of a responsibility may have conflicting demands made on them by both the authority for that responsibility and their own manager.

8.6 Responsibility assignment models and system dependability

In this section, I discuss how responsibility assignment models may be used to help improve the dependability of a complex socio-technical process. To improve dependability, you either have to ensure that faults are avoided, that faults do not lead to system failure (fault tolerance) or, if a failure of part of the system occurs, that recovery is possible without complete system failure. Responsibility models provide information that allow you to assess whether or not responsibilities have been appropriately assigned (fault avoidance), whether there is sufficient redundancy in a system in the event of responsibility failure (fault tolerance) and who must be involved in restoring the system in the event of a failure (fault recovery). In reality, fault avoidance, tolerance and recovery all overlap and the analysis of the responsibility models that I discuss in this section is applicable to all of these.

Figure 8.8 The hospital organisational structure



As with all modelling, much of the value of a responsibility assignment model comes from the process of understanding the situation in enough detail to create the model. To create a model, you need to analyse the responsibilities of the system in detail, questioning and discussing these with the responsibility holders. This process teases out problems, issues and uncertainties that can be immediately resolved. For example, there may be a misunderstanding about who has been assigned a responsibility. Once this has been resolved, the new situation, with correctly assigned responsibilities, is reflected in the responsibility model. The model itself is useful but the understanding comes from the process of developing the model.

Once a responsibility assignment model has been created, its principal benefit, perhaps, is that it provides a basis for discussing whether the assignment of responsibilities is correct and whether the level of responsibility is appropriate for the role (or the person) which has been assigned that responsibility. Ensuring that the right person is assigned appropriate responsibilities reduces the chances of a system fault arising through responsibility failure.

In many organisations, the assignment of responsibilities is historical (X has always done job Y) and may not have been updated to reflect changes in the organisational structure and processes. An explicit model brings this out into the open and can be used to question the established responsibility structure. The types of question that might be asked include:

1. Do all holders of responsibilities understand these responsibilities?
2. Are managers aware of the responsibilities of people that they manage? This is particularly important in situations where responsibility holders are professionals with some autonomy in deciding what tasks they undertake. Individuals may take on extra responsibilities that are not assigned by their managers.
3. Is the responsibility structure consistent with the organisational structure? If not, what problems might result as a consequence?
4. Do the holders of responsibilities have the right knowledge, competence, experience and commitment to discharge the responsibility?

5. What are the consequences for the system as a whole if the responsibility is not properly discharged?
6. If a responsibility is shared, do the responsibility holders understand their individual responsibilities? Are the responsibility holders co-located? If not, how do they communicate?
7. If there are related tasks (e.g. allocating and releasing a bed) with different responsibility holders, how do they agents involved communicate with each other? Are there vulnerabilities in this communication mechanism?
8. For all consequential responsibilities, are responsibility holders aware of the evidence that they must maintain to demonstrate that the responsibility has been properly discharged?
9. For all causal responsibilities, is it clear who has the responsibility for exception handling?
10. For all causal responsibilities, what happens if the responsibility holder is unavailable?

There are no right and wrong answers to these questions. However, by asking such questions, you are likely to discover if there are responsibility vulnerabilities that could lead to a system failure.

Turning now to the six types of responsibility failure that I identified in the introduction to this chapter, you can see how responsibility models can contribute to avoiding these failures:

1. *Unassigned responsibility*. This may be detected by looking at each of the responsibilities in the system and checking that agents have been assigned to that causal responsibility. Pay particular attention to whether or not an agent has been assigned to handle the responsibility in the event of things going wrong.
2. *Duplicate responsibility.* This could be detected if the same explicit responsibility was assigned to different agents in the model. However, this is unlikely. More likely, duplication occurs when parts of responsibilities overlap. Therefore, models of responsibilities as discussed in Chapter 8 are necessary to detect this vulnerability.
3. *Uncommunicated responsibility.* This can be detected in meetings when the allocation of individuals to responsibilities is discussed. By making responsibilities explicit, people can see what they are responsible for.
4. *Misassigned responsibility*. This may be detectable by individuals making clear that they do not have the competence or the resources to discharge the responsibility. However, this task is made easier if the assignment models are used alongside models of the responsibility itself, as discussed in Chapter 9.
5. *Responsibility overload.* This can be detected in a responsibility assignment model by putting different responsibility models together and

checking that a role or an individual has not been given too many things to do. Again, models of the responsibility as discussed in Chapter 9, may be helpful here.

6. *Responsibility fragility*. This can be detected by identifying critical responsibilities and ensuring that these are shared responsibilities with more than one agent assigned to them.

The benefits of a responsibility assignment model then are that it provides a basis for discussing and planning responsibilities in a complex system in such a way that vulnerabilities can be avoided. Once it has been established, it informs responsibility holders and their managers of the responsibility structures in a system. This is particularly valuable when the responsibility assignment changes. It is a universal problem that new people coming into an organisation never really know who is responsible for what. They are therefore less likely to make assumptions about responsibility which could lead to system failure.

The decision to model consequential responsibilities as goals was taken to make it easier to relate responsibility assignment models to analyses of system dependability. The notion of goals (claims) and associated evidence is used in dependability cases. These are cases that set out the reasons why a system should be considered to be dependable. By identifying the responsibilities, we can check that the goals in the dependability case are the same as the goals as seen by the holders of the responsibility. Mismatches imply that more work on the dependability case may be required or that there are responsibility vulnerabilities that must be addressed.

However, there are three possible 'failure modes' for responsibility models which may limit their usefulness in responsibility planning and analysis. These are:

1. The responsibility model may not be an accurate representation of organisational responsibilities.

2. Models may not be sufficiently detailed or may be too detailed and hence hard to understand.

2. Models may be out of date.

Producing an accurate responsibility models depends on understanding the responsibilities in an organisation. Responsibilities are not usually represented in a tangible way so there is no doubt that it is difficult to understand the real allocation of organisational responsibilities, especially if these are not directly reflected in an individual's everyday work. If you simply ask people about their responsibilities, they may find these difficult to describe or may forget about what they are responsible for.

In some cases, the modelling of responsibilities may be politically sensitive. While some organisations that have responsibility for safety-critical systems (such as air traffic control) may wish to explicitly identify who is responsible in the event of a failure, many organisations may prefer to conceal rather than reveal consequential responsibilities. This gives the opportunity to 'pass the buck' in the

event of a system failure and for individuals to try to avoid blame for incidents and accidents. While we understand that this is a reality in many organisations, we believe that mature organisations that are concerned with developing dependable processes can benefit from documenting these responsibilities.

At the moment, the best tool that we have for responsibility understanding is ethnography. However, in reality, neither the resources nor the time will normally be available for detailed ethnographic studies. More work is required on guidelines and support processes for responsibility elicitation and the initial creation of responsibility models.

Like all models, responsibility models are simplifications rather than reflections of reality. The notion of responsibility is a universal one and it is possible to decompose responsibilities to a very fine level of detail. However, it rarely makes sense to do so. Not only would this lead to a model that was cluttered and difficult to understand, people use their initiative in discharging their responsibilities. Too much detail in the model suggests that there is a prescriptive way of discharging a responsibility and this is rarely the case. Therefore, the challenge is to develop a model with enough detail to be useful but which is not so detailed that it is impossible to use. At this stage, we have to leave this to the judgement of the modeller; we have not yet developed any guidelines for responsibility decomposition.

The problem of keeping models up to date is universal for all kinds of system model. As the real-world changes (e.g. assigned responsibilities change, new responsibilities are created, etc.) the responsibility model should be updated to reflect these changes. If this is not done, then the model becomes less and less useful as a document describing the responsibility structures. To make change easier, we need tool support for model creation and editing and simple ways to integrate model updating with other work activities.

However, because many of the benefits of a responsibility model come from the fact that it can facilitate discussion, keeping the model up to date at all times may not be necessary. So long as it is an accurate reflection of the responsibility structure when changes to the socio-technical system have to be made, it can serve its principal purpose as a discussion support mechanism.

8.7 References

Clarke, K., M. Hartswood, et al. (2001). "Hospital Managers Closely Observed: Some Features of New Technology and Everyday Managerial Work." *J. of New Technology in the Human Services* **14**(1/2): 48-57.

Clarke, K., M. Hartswood, et al. (2002). “Minus nine beds”: Some Practical Problems of Integrating and Interpreting Information Technology in a Hospital Trust. *Proc. BCS Conf. on Healthcare Computing*, Harrogate, BCS.

Clarke, K., J. Hughes, et al. (2003). 'When a bed is not a bed: The situated display of knowledge on a hospital ward'. *Public and Situated Displays. Social and interactional aspects of shared display technologies.* K. O'Hara, M. Perry, E. Churchill and D. Russell. Amsterdam, Kluwer.

Clarke, K., R. Procter, et al. (2003). "Trusting the Record." *Methods of Information in Medicine,* **42**: 345-52.