9 Causal Responsibility Models

Ian Sommerville

9.1 Introduction

In previous chapters, we have discussed the ways in which we can model how responsibility can be assigned to agents and how responsibility models can facilitate discussions about the nature of responsibilities in organisations. These models document responsibilities in an organisation, provide insights into possible vulnerabilities due to responsibility misassignment and facilitate discussion about the nature of specific responsibilities. However, we have not, so far, tried to model the responsibilities themselves. Such a model might include information about the attributes of the responsibility, the relationships between these attributes and how one responsibility is dependent on other responsibilities.

The difficulties of developing such a model of responsibilities as abstractions in their own right should not be under-estimated. We have already discussed how the word 'responsibility' is used in a very broad way and it is not possible, in our view, to have a single model that encompasses all different types of responsibility. A further difficulty arises because responsibilities are always interpreted by the holder of the responsibility and their culture, education, competence and experience influences that interpretation. This is one reason why it is often difficult to decide who should be blamed when some accident or incident occurs and a tribunal of some kind examines the ways in which individuals have discharged their assigned responsibilities. Because of these difficulties, I focus here on the more limited, but still challenging, problem of modelling causal responsibilities.

Recall that causal responsibility is the responsibility of making some state of affairs come about or of acting to ensure that some undesirable situation does not occur. Each causal responsibility has an associated consequential responsibility where the consequential responsibility defines who takes the blame in the event of failure or, sometimes, the credit in the event of success. The agent that is assigned a causal responsibility may, but need not, be the holder of the corresponding consequential responsibility. For example, an automated agent assigned a causal responsibility cannot be assigned the related consequential responsibility – computer systems cannot take the blame for failure.

Modelling causal responsibilities, without regard for the agent assigned these responsibilities, is helpful for a number of reasons:

1. It focuses attention on the responsibility itself – does the responsibility properly reflect the intention of the organisation? That is, if an agent

properly discharges the responsibility, will this achieve the goals of the organisation?

2. It allows us to look at the relationships between responsibilities to find inconsistencies and incompleteness. If, for example, there are related responsibilities such as the admission of a patient to a hospital and the completion of an initial health check, we can check that the information produced and required by these activities is consistent.

3. It provides a basis for deciding on the allocation of responsibilities. The responsibility model may include information about the resources and competences required to discharge the responsibility. This information can then be used to decide who or what should be assigned the responsibility and what support they might require.

4. When used in conjunction with a responsibility assignment model, it provides a basis for vulnerability analysis. Using information from these models, it may be possible to assess if an agent has the capacity, resources and competences to discharge his or her responsibilities in a proper way.

At this stage, it is important to emphasise that the work on modelling responsibilities as abstractions in their own right is still immature. Nevertheless, we think it important to introduce the ideas here as they are completely novel and reflect what we believe is an important step forward in understanding issues that influence the dependability of socio-technical systems.

So far, our work on responsibility modelling has not addressed the problem of modelling consequential responsibilities. Indeed, it is not clear what might be included in such a model. In some case, the consequential responsibility model would simply consist of the associated causal responsibilities but there are consequential responsibilities which are not really definable in this way. For example, the director of a railway company may be responsible for the safety of the public but defining this as a causal responsibility would not be meaningful. How to model and represent this type of responsibility is a problem for future work.

In the remainder of this chapter, I introduce an approach that may be used to define causal responsibilities and discuss the inherent uncertainties in responsibility modelling. I then go on to explain how information about responsibilities may be used in conjunction with responsibility assignment models to infer whether or not responsibility assignments have vulnerabilities that could lead to system failure. I illustrate this discussion with examples derived from discussions in earlier chapters of the book.

9.2 Causal responsibilities

We have introduced the notion of a causal responsibility as a responsibility for making something happen or ensuring that some undesirable state does not occur. Therefore, examples of causal responsibilities might be the responsibility of delivering drugs to a patient in a hospital, the responsibility of updating patient records or the responsibility of monitoring patients to ensure that their blood pressure has not increased or decreased to an unsafe level.

Slightly more formally, we can define a causal responsibility as follows:

A causal responsibility is an *obligation* to some *authority* to ensure that some state of affairs is achieved/avoided.

All causal responsibilities should have an associated authority as discussed in Chapter 8 where we introduced a notation for associating authority with responsibilities. This authority is not part of the responsibility itself but depends on the responsibility assignment. For causal responsibilities, we define the authority for the responsibility to be the agent who decides whether or not a causal responsibility has been properly discharged. To do so, they must receive a report of some kind from the agent holding the causal responsibility. The authority associated with a responsibility often depends on the assignment of that responsibility – hence, a statement of the authority should not be part of the responsibility model.

The authority of a causal responsibility who decides that that responsibility has not been properly discharged need not be need not be the holder of the associated consequential responsibility. For example, if a responsibility to provide patient information is assigned to a database system, the operator of that system may be the authority who decides whether or not the patient information is properly provided. However, they cannot assign blame and some other agent or body must decide why the database system is not operating as intended and who is consequentially responsible for this.

While causal responsibilities can be thought of as the responsibility for ensuring that some change in the world takes place or is avoided, it is sometimes convenient to group types of change under the heading of a single responsibility. For example, in a library there may be a responsibility for issuing books to readers and receiving books from readers to return to stock. These can be thought of as part of a single responsibility – 'Book Lending'. In some libraries, this might be assigned to a single agent, in others, separate agents would be responsible for dealing with the issuing of books and their return to stock. The 'Book Lending' responsibility therefore includes two simpler responsibilities namely 'Book Issuing' and 'Book Return'.

Because responsibilities may be made up of other responsibilities, it is therefore useful to introduce the notions of simple and composite responsibilities. A simple responsibility is one where a single agent is assigned the responsibility and only that agent is involved in discharging the responsibility. A composite responsibility is one that is made up of other responsibilities, which may be (but need not be) assigned to different agents. Therefore, 'Book Lending' may be considered to be a composite responsibility in libraries where there are separate desks for the issuing and the return of books.

It is important here to distinguish between the notions of composite responsibility and role. A specific role in an organisation may be defined by the allocation of responsibilities to that role. Therefore, in a school, the role 'Head Teacher' might be defined by the associated responsibilities of 'Staff management', 'Expenditure approval', 'Student welfare', etc. These responsibilities are disparate and may have little in common. The responsibilities defining the role may therefore change with little impact on other responsibilities. For example, the school may decide to reduce the load on the head teacher by assigning the (causal) 'Student welfare' responsibility to a Deputy Head. It therefore makes little sense to define 'Head Teaching' as a composite responsibility.

Composite responsibilities only make sense when they are made up of simpler responsibilities that are coherent and mutually dependent. They should rely on shared information such as a shared database. For example, the simpler responsibilities of 'Book Issue' and 'Book Return' update a shared database of loans from the library and are obviously dependent in that a book cannot be returned without being issued. If the responsibilities in a collection are independent, then these define a role (as discussed in Chapter 1) rather than a composite responsibility.

Whether or not a responsibility is a simple or a composite responsibility is not inherent in the responsibility itself but depends on the organisation within which the responsibility is defined. In a small library, it is unlikely that the activities of issuing books and accepting them for return would be separate. 'Book Lending' is therefore a simple responsibility. In a large library, it may make sense to separate these functions so that people returning books do not need to queue alongside people waiting for books to be issued. 'Book Lending' in such settings is a composite responsibility.

This exemplifies the fact that responsibility descriptions are not context-free but depend on the organisation in which the responsibility is discharged. Therefore, an important function of these descriptive models is to allow responsibilities to be compared across organisations. By creating an explicit model of the responsibility, we may highlight the differences and similarities between responsibilities that have the same name in different organisations. This may help to avoid misunderstandings about 'who is doing what' when some task is shared across organisations.

While the general definition of causal responsibility as the obligation to achieve or avoid some state of affairs is universal, when we look at responsibilities that are assigned to agents in real systems, we see that simple causal responsibilities fall into three broad classes:

1. 'Doing' responsibilities whose aim is to affect some change of state in the world (although its normally more useful to think of some restricted part of the world such as a hospital).

2. 'Monitoring' responsibilities whose aim is to observe part of the state of the world and events that influence that state and report if the state is desirable/undesirable.

3. 'Avoiding' responsibilities whose aim is to ensure that some undesirable state does not occur.

'Doing' responsibilities may be transaction-oriented, where the start and end states are clearly defined or they may be creative responsibilities. Creative responsibilities are usually longer-term and involve the 'creation' of some output rather than the completion of some task. Their end state cannot be defined in an objective way but, rather, its achievement is socially determined. That is, the actors involved have to agree on when the end state has been reached. An example of a transaction-oriented doing responsibility is to admit a patient to a hospital. There is a clearly defined start state, which is the presentation of the patient for admission and an end state, which is the allocation of the patient to a hospital bed. An example of a creative responsibility is the writing of this book chapter. The author

and editors collectively decide when the chapter is 'finished' and acceptable for publication.

'Monitoring' activities are not transaction-oriented. They don't necessarily have a trigger event to initiate them and they may never end. They have inputs (what to monitor) but may never produce an output if the undesirable state does not occur. Monitoring responsibilities may involve the real-time monitoring of sensors or may be retrospective where data is monitored to ensure that an undesirable state has not arisen. An example of a real-time monitoring responsibility is where an automated agent is responsible for monitoring the state of a chemical process by observing sensors in the reactor vessel and reporting (by setting of an alarm) if the temperature and pressure falls outside some limits. An example of a retrospective monitoring responsibility is financial auditing. An auditor monitors the financial state of an organisation and reports on that state. In both cases, the monitoring agent does not take action to change that state.

In principle, a monitoring responsibility could be represented as a doing responsibility (i.e. Observe state; if state = X then report). However, from the perspective of the agent who is assigned the responsibility, this may not be a natural representation as, most of the time, the agent is simply observing rather than taking action. The 'doing' part i.e. the reporting, may rarely, if ever, arise. Of course, from the perspective of a different agent, monitoring responsibilities can be thought of as doing responsibilities. For example, carrying out an audit might be seen by the auditor as a doing responsibility but as a monitoring responsibility by the organisation being audited.

Alternatively, it might be argued that monitoring responsibilities should be considered to be a composite responsibility including the simpler responsibilities 'Monitor' and 'Report'. This has the benefit that it is possible to distinguish between monitoring failures and reporting failures. A monitoring failure might be the incorrect reading of a sensor; a reporting failure might be the failure to inform some other agent that a temperature sensor is reporting an abnormally high reading. However, I think that monitoring without some form of reporting is meaningless – otherwise, the monitored state is never exposed. Therefore, separating monitoring from reporting does not really make sense. I, therefore, do not consider monitoring responsibilities to be composite responsibilities

'Avoiding' responsibilities normally include both monitoring responsibilities (watch for indicators that suggest the undesirable state is becoming more probable) and doing responsibilities (do something to reduce the probability of that undesirable state). For example, in a hospital, an undesirable state is the state of having no beds available for emergency admissions. Avoiding this state involves monitoring the number of beds available and the likely future demands on these beds. If these indicate that the demand for beds is likely to exceed the supply then actions such as the early discharge of patients may be invoked.

This classification of responsibilities is, I believe, helpful because it allows us to think about the resources and competences required to discharge each type of responsibility. In situations where several responsibilities are assigned to the same agent, we may get clues from the classification about whether that agent will be able to discharge all of the assigned responsibilities if some kind of problem arises. For example, if an agent is assigned several 'avoiding' responsibilities, what will

happen if the undesirable state for more than one of these responsibilities arises simultaneously?

Knowing something about the resource requirements for a responsibility is important as it provides a basis for deciding on the responsibility assignment and identifying vulnerabilities due to a lack of resources to discharge the responsibility. In general, the different types of responsibility have different levels of resource requirement:

1. Doing responsibilities always require some level of resource in order to transform inputs to outputs. The amount of resource required may be predictable if the responsibility is rule-based (see below) but often depends on the knowledge, experience and competence of the responsibility holder.

2. The resource requirements for monitoring responsibilities depend on the complexity of the information that is being monitored. If this information is simple, the resource requirements will be low but as it becomes more complex, these requirements increase. This can cause particular difficulties in the event of failure of an information provision system such as a sensor. Manual intervention may then be required to collect the data being monitored so the overall effort required for monitoring may increase significantly. Furthermore, the need to report the monitored result also requires resources – there must be sufficient available bandwidth in the reporting channel and the reporting agent must have the time to organise the information to be reported. It is difficult to predict these requirements as they depend on the system state that has to be reported.

3. The resource requirements to properly discharge avoiding responsibilities are difficult to predict. If the undesirable state does not occur, then the resources are whatever is required for monitoring. However, the more likely the undesirable state, the more effort that may have to be devoted to doing responsibilities to avoid the state. If an agent is assigned more than one avoiding responsibility, then they may not have the resources to cope if they have to cope with a situation where tow or more undesirable states are reached at the same time.

If an agent is assigned both doing and avoiding responsibilities and the doing responsibilities consume virtually all available resources, then discharging the avoiding responsibility may mean that, inevitably, a failure occurs in the doing responsibilities.

The resource requirements for a responsibility obviously depend on the competence of the agent assigned that responsibility. As a result, accurately predicting these requirements in advance can be very difficult. The more flexibility there is in discharging a responsibility, the more difficult it is to predict the resource requirements. This flexibility is reflected in different strategies that may be used to discharge responsibilities:

1. *Rule-based strategies*. In this approach, the responsibility can be discharged by following a set of clearly defined rules or instructions. These are a *definitive* description of the responsibility. In principle at least, a rule-based responsibility can be represented as a workflow which can be enacted by an automated agent.

An example of a responsibility that could be primarily discharged using a rule-based strategy is maintaining the temperature in a building within a given range.

2. *Experience-based strategies*. In this approach, the holder of the responsibility discharges that responsibility by adopting a strategy based on their experience of previous situations where that responsibility had to be discharged. The way that it is discharged may follow a standard pattern but this is adapted and configured depending on the experience of the responsibility holder. It is possible to describe experience-based strategies using a workflow but this is *indicative* rather than definitive. This means that the workflow indicates one way of discharging the responsibility. However, it is recognised that alternative approaches may also be adopted to cope with unusual circumstances. Because of this flexibility, experience-based responsibilities cannot be completely assigned to an automated system although software may be used in a supporting role.

   An example of an experience-based strategy is the approach used to allocate beds to incoming patients discussed in Chapter 8.

3. *Knowledge-based strategies*. In this approach, the holder of the responsibility uses their knowledge and skills to discharge the responsibility. It makes little sense to try and pin down exactly how this is done as it is very dependent on the individual holder of the responsibility.

   An example of a knowledge-based responsibility is the responsibility to write a chapter of a book on responsibility and dependability.

In practice, responsibilities may be classified as primarily rule-based, experience-based or knowledge-based, although most responsibilities probably have some elements of all of these. For example, the rule-based strategy that can be followed by an automated system to maintain temperatures may break down in the event of equipment failure. In such a situation, the responsibility may pass to a human who will adopt an experience-based strategy to try to discharge the key elements of the responsibility. Similarly, the knowledge-based responsibility of writing a book chapter does involve some rule-based activities such as formatting and checking spelling and grammar.

It is useful to identify the primary classification of a responsibility because it provides information about the scope for automating the responsibility and for understanding how the proposed responsibility model relates to the reality of discharging the responsibility.

9.3 Causal responsibility models

A causal responsibility model is a standardised representation of a responsibility that includes information that is central to understanding the nature of that responsibility. These models are designed for people to read so that they can understand the responsibilities that exist and how that responsibility might be discharged. By representing the responsibilities in an abstract, standard way, we can ensure that the responsibility is properly documented. We can compare models

more readily than textual descriptions and it may be possible to develop tool support to maintain and manage the responsibility descriptions.

The process of developing a responsibility model requires the modeller to acquire a thorough understanding of what is involved in discharging the responsibility and the resources and competences required for the responsibility discharge. Ethnographic studies, as discussed in Chapter 8, along with discussions with responsibility holders may be used as a means to develop this understanding. The information gained may then be organised and structured according to the responsibility pattern format that I discuss in section 9.4.

Almost inevitably, initial attempts at developing a responsibility model will be incomplete and inconsistent – it is hard for people to explain what they do. Therefore, developing responsibility models should be seen as an iterative process where models are proposed, presented to the actors involved and modified according to their comments.

9.3.1 Requirements for a responsibility model

As discussed in the introduction to this chapter, the purpose of a responsibility model is to help people understand the nature of a responsibility, decide who should be allocated a responsibility and identify possible responsibility vulnerabilities. The causal responsibility model therefore has to include information that allows this analysis to take place. At the very least, a responsibility model should include:

1. Information about the context in which the responsibility is discharged.
2. Information about what is assumed to be true when the responsibility is discharged.
3. Information about how the responsibility might be discharged, including required inputs and expected outputs.
4. Information about exceptions that might arise during the discharge of the responsibility.
5. Information about how the discharge of the responsibility affects the state of the world.
6. Information about the resources that are normally required to discharge the responsibility.
7. Constraints that might apply to the holder of the responsibility (e.g. in a military context, the responsibility holder may have to have a certain level of security clearance).

As responsibility models are intended for analysis by people rather than programs, readability is an essential requirement. The form of the model must allow for flexibility as different people may wish to define the same responsibility in different ways.

As I have discussed earlier in the chapter, there are different types of responsibility (doing, monitoring, avoiding) and different strategies for responsibility discharge (rule-based, experience-based and knowledge-based). Responsibilities may also be simple or composite responsibilities. Ideally, all of

these should be accommodated within a single model although the detail that is normally included in different parts of the model may differ for each responsibility type.

9.3.2 A pattern-based responsibility model

The approach that I propose for modelling individual responsibilities is based on the notion of a pattern. Patterns were first proposed by Alexander (Alexander, Ishikawa et al. 1977; Alexander 1979) who identified approaches to architecture that worked effectively in a range of settings. He defined a pattern as:

> "Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice." (Alexander, Ishikawa et al. 1977)

The essence of this definition is that a pattern is a generalisation that can be instantiated in different ways in different settings. The notion of patterns has received a great deal of attention from the software engineering community and have been used to represent standard software architectures and designs (Gamma, Helm et al. 1995; Schmidt 1997; Coplien 1998; Erickson 1998; Larman 2002; Martin and Sommerville 2004). These have been somewhat different and rather more specific than Alexander's patterns but the differences are not of interest here.

The notion of a pattern as a generalisation that may be instantiated in many different ways reflects the essential characteristic of responsibilities. Different agents who are assigned a responsibility (such as writing a chapter of this book, say) will approach this in completely different ways. Nevertheless, all of these agents have a basic understanding of the fundamental notion of writing a chapter. Therefore, patterns are the basis for my definition of responsibilities.

Patterns are usually represented as structured entities with a number of different fields describing different aspects of the pattern. To define causal responsibilities, the template shown in Figure 9.1 is a flexible framework for responsibility description. The ways in which the different components of this pattern are completed is partially dependent on the type of responsibilities. For example, for rule-based responsibilities, the normal process may be defined using a diagrammatic workflow notation. The requirements field may set out the resources that are normally required to enact the workflow. However, for a knowledge-based responsibility, the normal process may be a simple description in free text and the requirements field may simply set out some initial requirements before chapter writing can commence.

To illustrate how these responsibility patterns may be used, I have defined patterns for four different responsibilities:

1. The responsibility to maintain the temperature in a plant house within a certain range (say 5 to 30 degrees Celsius). This is a rule-based responsibility than could be assigned to an automated system. Assume there are temperature sensors in the plant house and actuators to switch on heating if the temperature gets too low and to open windows and doors as the temperature increases. The normal process could be defined using a graphical workflow notation. This is illustrated in Figure 9.2.

Figure 9.1 A pattern for responsibility description

Component	Description
Name	A short, meaningful name for the responsibility.
Goal	What the responsibility is trying to achieve. This should normally be explained in a single sentence.
Context	A description of the environment or the context where the responsibility will be assigned. This may be a simple textual explanation or a more detailed model that shows the actors and other systems in the environment.
Type	The type of the responsibility – simple or composite. This depends on how the responsibility is considered within a particular context and may differ for the same responsibility in different contexts. As discussed earlier, simple responsibilities will normally be assigned to a single agent; composite responsibilities may be assigned to several agents.
Classification	The classification of the responsibility in two dimensions - (Doing, Monitoring, Avoiding) and (Rule-based, Experience-based, Knowledge-based). This should reflect the judgement of the modeller as to the primary classification – in reality, responsibilities are mixtures of all of these.
Pre-conditions	Context conditions that must normally hold before the responsibility can be discharged. Assumptions that are made about the context where the responsibility is to be discharged may be included as pre-conditions.
Post-conditions	Context conditions that hold after the responsibility has been discharged. These reflect how the state of a system or its environment has been changed by the discharge of the responsibility.
Normal process	A description of how the responsibility may be discharged. For simple responsibilities, this should be expressed as a workflow or process description. The process description should include a specification of the required inputs and expected outputs. For composite responsibilities, this should include a list of the other responsibilities (simple or composite) in the composition.
Variations	Ways in which the normal process may vary. (These are not exceptions i.e. things going wrong but rather less common situations that require different actions).
Exceptions	Exceptions that may arise in the course of responsibility discharge.
Advice	Information about how exceptions might be handled. This might reflect previous experience of dealing with exceptions in similar situations.
Requirements	Requirements that must be satisfied for the normal discharge of the responsibility. These may include requirements for a specific resource such as time, constraints on the assignment of the responsibility and the handling of exceptions. Information and communications requirements are particularly important.

Figure 9.2 The Maintain Temperature responsibility

Component	Description
Name	Maintain Temperature
Goal	Ensure that the temperature in some area is always maintained within given limits.
Context	A plant house where the temperature must be maintained between 5 and 30 degrees.
Type	Simple
Classification	Avoiding, Rule-based
Pre-conditions	Heating and ventilation equipment for temperature control must be installed.
Post-conditions	None. The responsibility does not terminate.
Normal process	The normal process is, essentially, an endless loop of checking sensors and activating actuators to switch heating on and off and open and close ventilators. See Figure 9.6.
Variations	None
Exceptions	Equipment failure.
Advice	Heating equipment failure in cold weather can lead to frost damage to plants. Wrap plants in insulating material. Ventilation equipment failure in hot weather can lead to overheating. Manually jam open all openable windows and doors. Drape material over windows to provide shade. Spray vulnerable plants with water to keep cool.
Requirements	If automated discharge, then activity log must be maintained and checked by human operator every hour.

2. The responsibility for bed management in a large hospital. The bed management responsibility involves ensuring that beds are available for patients being admitted to the hospital and that the most effective use is made of the hospital's stock of beds. Beds should not be left empty for long periods of time. This is a composite responsibility including the operational responsibilities of bed allocation and bed release and planning responsibilities to take into account the expected demand for admission. This is illustrated in Figure 9.3. Notice that this satisfies the requirement for a composite responsibility that the simper responsibilities should be dependent. In this case, all of these simpler responsibilities use the same shared beds database.

3. The responsibility to allocate a bed to patients being admitted to a hospital. This is an experience-based responsibility which is part of the composite responsibility of bed management. There is a standard way of doing this but the admissions officer will often have to deal with unusual cases which can't be handled in a routine way (e.g. a patient with a very infectious disease who has to be isolated, patients who are suffering from dementia, etc.) In these cases, the admissions officer uses his or her experience to decide how best to complete the admissions process. A graphical description of the normal process may be useful but there would be many exceptions to it. This is illustrated in Figure 9.4.

Figure 9.3 The composite Bed Management responsibility

Component	Description
Name	Bed Management
Goal	To ensure that patients are assigned a bed within a reasonable time of admission to the hospital and to ensure that the hospital's stock of beds is efficiently used.
Context	A large general hospital treating a wide range of conditions.
Type	Composite
Classification	Doing, Experience-based
Pre-condition	N/A for composite responsibilities
Post-condition	N/A for composite responsibilities
Constituent responsibilities	Bed Allocation, Bed Release, Capacity Planning, Reporting
Variations	May include Patient Transport Planning where disabled patients have to be transported by ambulance or where patients have to be moved between dispersed units of the hospital.
Exceptions	N/A for composite responsibilities
Advice	Careful coordination of Bed Allocation and Bed Release is essential when the hospital is close to capacity. The capacity plan has to be revised on a twice-daily basis in such circumstances.
Requirements	The holder of the responsibility should have had some clinical experience e.g. as a nurse so that they can understand clinical priorities.

4. The responsibility to write a book chapter on responsibility modelling. This is a knowledge-based responsibility that is a 'creating' responsibility. I know from experience of writing this chapter and other chapters that I couldn't articulate the process of writing that I have followed. Nor could the requirements be articulated in anything other than a rather trite way (e.g. I needed time free of interruptions close to the deadline). This responsibility is illustrated in Figure 9.5.

The Maintain Temperature responsibility is an example of a rule-based responsibility that could be assigned to an automated system. As this is a monitoring responsibility, there is no associated post-condition as the responsibility is not episodic. That is, you can't really say when the discharge of the responsibility has been completed – it is a continual process that never terminates. There is a single requirement associated with the responsibility, which is intended to help discover if an automated system is operating correctly. Of course, there are other implicit requirements such as the need for sufficient computational capacity in an automated system. However, there is no need for a responsibility pattern to be complete and to define requirements at a very fine level of detail. Remember, the model is intended for use by intelligent people not for enaction by computers.

Figure 9.4 The Bed Allocation responsibility

Component	Description
Name	Bed Allocation
Goal	Assign a bed to all patients being admitted to the hospital.
Context	A large general-hospital treating a wide range of conditions.
Types	Doing, Experience-based
Pre-conditions	Hospital must be in an 'admitting patients' state.
Post-conditions	All patients that are presented for admission are assigned a hospital bed.
Normal process	The normal process of allocating a bed is shown as a workflow in Figure 9.7.
Variations	Where the database reports that no beds are available, manual intervention is required to check actual bed availability by calling wards to see if patients have left the ward but the bed has not been released and by liaising with clinical staff to speed up bed release.
Exceptions	Equipment failure; Exceptional patient (e.g. senior politician)
Advice	If an exceptional patient, ensure that bed in a single room is assigned. In the case of equipment failure, call around wards to discover bed status. Delay admission of patients with less serious conditions.
Requirements	Discharge of patients to free up bed must be approved by doctor in charge of ward; Bed management database must be deployed and properly configured. Admissions staff must be trained in use of bed management system and be authorised to use it. No more than 30 patients an hour can be admitted/discharged.

Bed management is an example of a composite responsibility. While the overall responsibility would normally be assigned to a bed manager, the responsibilities included might be assigned to different agents. For example, in the system that we studied, Bed Allocation was the responsibility of the Admissions Officer (part of the hospital administration) whereas Bed Release was the responsibility of nurses in the ward where the bed was to be released. Capacity Planning and Reporting were the responsibility of the bed manager. The bed manager became involved in Bed Allocation and Bed Release when the database reported that there were no available beds for incoming patients. Notice that for composite responsibilities, it is not normally helpful to include descriptions of pre and post conditions or exceptions. These are more applicable to simple responsibilities.

The Bed Allocation responsibility is an experience-based responsibility that is part of the composite bed management responsibility. It is an episodic responsibility where each discharge episode involves allocating a patient to a bed so the defined post-condition holds after each discharge of the responsibility. Notice that a key part of this responsibility is the discussion on variations in discharging the responsibility as these reflect previous experience. Similarly, the

Figure 9.5 The Chapter Writing responsibility

Component	Description
Name	Chapter Writing
Goal	Write a chapter of a book.
Context	The production of a book on Responsibility and Dependability.
Types	Doing, Knowledge-based
Pre-conditions	Approval given by book editors to chapter synopsis.
Post-conditions	Chapter delivered to book editors.
Normal process	*No workflow for knowledge-based responsibilities. It is up to the chapter writer to decide how to discharge the responsibility.*
Variations	
Exceptions	Failure of required material from other chapter authors to be available.
Advice	Re-oriented chapter with an alternative focus; Combine chapter with another chapter.
Requirements	Chapter author must have problem knowledge and writing skills. Chapter author must have time available to complete chapter and must provide an estimate of the time required. Editor time must be available to review chapter.

advice on exception management explains how these problems have been handled in the past.

This responsibility pattern also shows how the requirements field can be used to provide information about required resources and competences. The training requirement essentially defines a required competence and the capacity requirement indicates that the responsibility holder requires at least 2 minutes to complete the bed allocation process. This is important in planning the workload of the admissions officer and making provision for support in circumstances (such as a serious accident) where many patients are presented for admission at the same time.

The Chapter Writing responsibility description shown in Figure 9.5 is rather shorter than the patterns defining the rule-based and experience-based responsibilities. The reason for this is that knowledge-based responsibilities are discharged in different ways depending on the competencies, knowledge and experience of responsibility holders. People who have written different chapters of this book have tackled them in completely different ways. For example, I was responsible for writing Chapter 8 and Chapter 9. I based Chapter 8 on an existing, unpublished article on responsibility assignment and modified and extended it for this book. This chapter was written from scratch and the pattern-based approach that I have discussed was developed, refined and extended as the chapter was written.

Patterns are abstract descriptions that are designed to represent a range of instances. A criticism that can be levelled at pattern-based approaches is that the descriptions they use are too abstract and, sometimes, inherently vague. There is no

doubt that this criticism can be made of responsibility patterns. For knowledge-based patterns in particular, the descriptions of what is involved in discharging the responsibility are general and informal. However, you must remember that the principal function of these responsibility descriptions is to facilitate discussion and analysis by of the responsibility by people, not by computers. You should not think if these models as definitive and complete specifications of a responsibility – rather, they are a useful starting point for communicating the essence of the responsibility to people who need to understand it.

9.3.3 Workflow description

In Chapter 8, I suggested that causal responsibilities should be represented as a process. The reason for this is that a process of some kind is followed to discharge the responsibility although that process can depend on the knowledge and experience of the responsibility holder. For rule-based and experience-based responsibilities, I believe that it is helpful to make the process associated with the responsibility explicit as this provides a clearer and more complete definition of what is involved in discharging the responsibility. The explicit process description also means that it is possible to discuss what components of the responsibility can be transferred and delegated. The notation that I suggest using for the process description is a workflow notation.

Workflows represent business process models and are usually represented using a graphical notation such as BPMN (White 2004) or YAWL (van der Aalst and ter Hofstede 2005) At the time of writing, the process modelling language which seems most likely to emerge as a standard is BPMN. This is a graphical language which has been developed as a basis for workflow programming in service-oriented systems. It is reasonably easy to understand and mappings from the language to lower-level descriptions in an XML-based workflow language, WS-BPEL, have been defined.

Figures 9.6 and 9.7 are examples of BPML workflow descriptions that show the definitive process for maintaining temperatures (a rule-based responsibility) and an indicative process for bed allocation in a hospital (an experience-based responsibility). The key difference between definitive and indicative responsibility models is that a definitive model sets out how the responsibility is normally discharged whereas an indicative model defines how it could be discharged.

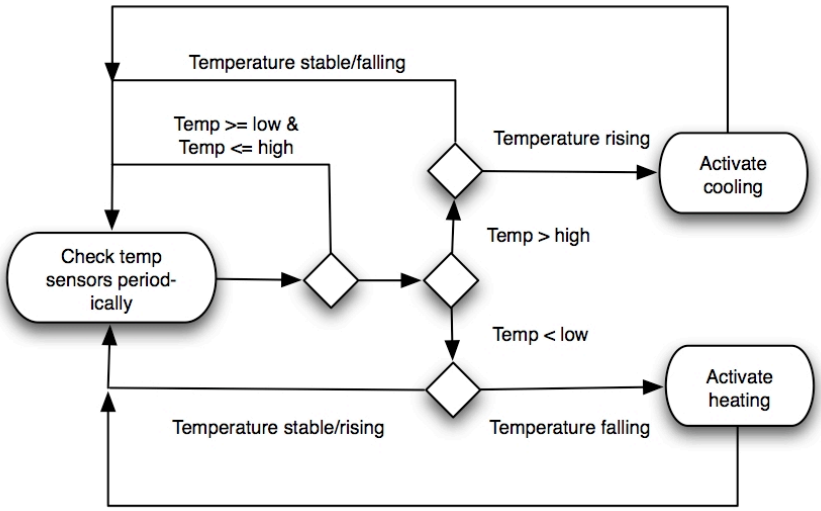
The process models shown in Figures 9.5 and 9.6 introduce some of the core concepts of BPMN that are used to create workflow models:

1. Activities are represented by a rectangle with rounded corners. An activity can be executed by a human or by an automated service.

2. Events are represented by circles. An event is something that happens during a business process. A simple circle is used to represent a starting event and a darker circle to represent an end event. A double circle (not shown) is used to represent an intermediate event. Events can be clock events thus allowing workflows to be executed periodically or timed out.

3. A diamond is used to represent a gateway. A gateway is a stage in the process where some choice is made. For example, in Figure 9.6, there is a choice made on the temperature reading returned from a sensor.

Figure 9.6 The definitive workflow for Maintain Temperature



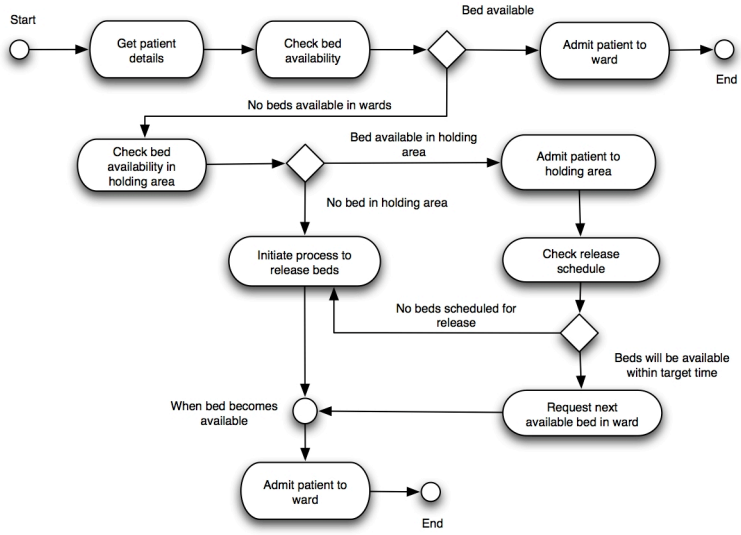
4. A solid arrow is used to show the sequence of activities; a dashed arrow represents message flow between activities.

These key features are enough to describe the essence of most workflows. However, BPMN includes many additional features that I don't have space to describe here. These add information to a business process description that allows it to be automatically translated into an executable form.

When writing workflows for responsibility description, you should try and make these as general as possible and minimise specific environmental details. This makes it easier to reuse the responsibility description in a different setting and provides some flexibility in how the responsibility is discharged. Therefore, in Figure 9.6, you can see that the specific low and high temperatures are not mentioned but I refer to these as 'low' and 'high'. Similarly, the specifics of the heating and ventilation system are not shown – the processes are simply shown as 'Activate heating' and 'Activate cooling' without regard for how this is accomplished. Figure 9.6 is a description of a rule-based responsibility and you can see how this process description could be translated, fairly easily, into an algorithm that could be followed by a computer system.

Figure 9.7 shows a description of the indicative workflow that describes the allocation of a bed to a patient who is being admitted to hospital. Essentially, the admissions offer checks the database and if a bed is available it is allocated. If there are no beds available in wards, then bed availability in a holding area is checked. If there is a bed then this is allocated to the patient but the patient is added to a queue to patients to be allocated beds in a ward. If there are no beds available in either wards or the holding area, then a process of releasing beds is initiated and, once a bed becomes available, the patient is assigned to it.

Figure 9.7 An indicative workflow for Bed Allocation



Bed allocation is an experience-based responsibility so the workflow is indicative rather than definitive. This means it is a description of how the responsibility might be discharged but, in reality, holders of the responsibilities will develop their own process depending on their experience, their workload and the environment where the responsibility of discharged. For example, if two patients are presented for admission at the same time with only one bed available, the admissions officer will make a decision on which patient should have priority. The workflow model should therefore be seen as a way of exposing the responsibility so that the people involved can discuss it. They can plan for exceptional situations, such as the need to admit many patients who have been injured at the same time in a major accident. In such circumstances, it may be impossible to follow normal procedures as many less urgent patients may have to be discharged. All doctors may be busy so procedures for identifying non-urgent cases (e.g. all patients scheduled for surgery but not yet in theatre) may be defined.

You should not think of the indicative workflow model as a template for process design. Responsibility models may be created during the requirements engineering stage of system development and they should be considered as an operational description that might but need not be adopted in the final system design. In such cases, they should be seen as an input to the design process rather than an output from it. It may be sensible to go through the processes of responsibility assignment and vulnerability analysis before arriving at a final

process design. Of course, if an alternative process design is agreed, it may then be sensible to update the operational model of the responsibility to reflect this.

9.4 Using responsibility models

The explicit modelling of responsibility involves effort and, by exposing what is often implicit, has the potential to create political and personal tensions in an organisation. It is therefore important that such models are not simply taken as a means of documenting responsibility (although this can be valuable, especially when the responsibility changes) but as a tool to improve dependability in a socio-technical systems or, more widely, across an organisation. I believe that there are three ways in which explicit responsibility models can contribute to improved dependability:

1. The models support the contingent assumption of responsibility in cases where the principal responsibility holder is unavailable.
2. The models help with responsibility allocation and reduce the probability that an inappropriate agent is assigned the responsibility.
3. The models may be used in conjunction with responsibility assignment models for vulnerability analysis.

Ethnographic studies of teamwork have, without exception, revealed that the division of labour (and hence responsibilities) in an effective team is contingent and dynamic (Anderson, Hughes et al. 1989; Ackroyd, Harper et al. 1992; Bentley, Rodden et al. 1992). Who does what is continually renegotiated, often without the need for explicit communication between the team members. This contingent assumption of responsibility reduces dependencies on individuals, makes people aware of other's work and hence able to check for mistakes and allows teams to cope with high demands. It is inherent in dependable working.

Of course, in tightly-knit teams, there is no need for explicit responsibility models for team members to be aware of other's responsibility. However, in situations where the dynamic assumption of responsibilities is less common, then an explicit responsibility model makes it easier for someone who is unfamiliar with the responsibility to get started with the work. For example, say the admissions officer in a hospital is called away urgently because a relative is seriously ill. In such situations, someone else would be called to cover but, before they arrive, patients still have to be admitted to the hospital. The responsibility model would allow a nurse who has used the system for bed release to be aware of what's involved in admitting patients. They would be less likely to make errors in the process. Overall, system dependability is improved because the admissions service remains available.

A common vulnerability that was identified in Chapter 8 is that of misassigned responsibility where the responsibility holder does not have the competence or resources to discharge the responsibility. Hence, there is a higher probability that they will make mistakes that compromise the dependability of the system. As I discuss below, the models may be used to help detect such misassigned responsibility but it is best to avoid such a problem rather than detect

it after it has occurred. Explicit responsibility models help decide who has the required competencies to discharge a responsibility in two ways:

- The requirements associated with a responsibility may set out the required competencies. For example, a requirement might be that the agent holding the responsibility for health and safety in an office has completed an approved first-aid course.

- Specific skills that an agent requires or conditions that would make it difficult for an agent to discharge a responsibility may be identifiable from the responsibility description even if these are not made explicit as competency requirements. For example, a responsibility that involves monitoring the status of a process may involve checking colour changes in a display. This suggests that this responsibility should not be assigned to an agent who is colour-blind.

These applications simply require an explicit responsibility model without regard for how the responsibility has been assigned. However, when you use responsibility models in conjunction with responsibility assignment models, as discussed in Chapter 8, more extensive vulnerability checking is possible. Recall that I identified 6 types of responsibility vulnerability in Chapter 8:

1. *Unassigned responsibility.* Within a socio-technical system, the responsibility for some critical task is not assigned to any agent.
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.
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.
4. *Misassigned responsibility.* In this situation, the agent who is assigned the responsibility does not have the competence or resources to discharge 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.
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.

Causal responsibility models are not required to detect unassigned or uncommunicated responsibility, but they have a role to play in detecting the other types of responsibility vulnerability.

Duplicated responsibility is problematic where there is an overlap in responsibilities and parts of the underlying process are common. For example, both agent A and agent B may believe that they are responsible for updating some information in a database. If they interpret that information differently, then

inconsistencies may be introduced depending on who added or modified the information. However, when the responsibility is made explicit, different responsibilities can be compared and areas of overlap may be detected.

Misassigned responsibility, as discussed above, may result from an agent's lack of competence or because an agent has too many other demands on their resources. The first of these has been discussed above but the second relies on a responsibility assignment model to identify all of the responsibilities assigned to an agent. The pattern-based models of these different responsibilities may then be compared to check that the total resource requirements do not exceed the capacity of the agent. It is particularly important to check whether the agent has the capacity to handle all of the responsibilities if problems arise simultaneously in more than one assigned responsibility. While it may not be realistic to ensure that agents always have spare capacity for such situations, there should be an explicit plan of how responsibilities should be prioritised and how the service offered by the socio-technical system should be gracefully downgraded.

A similar approach is used to check for responsibility overload. Overload is particularly likely in situations where responsibilities may be assigned from different sources. Hence, an agent may be assigned some responsibility by their manager and some other responsibility because they are a member of a planning group that cuts across departments in an organisation. By examining the explicit model of each of the responsibilities, it is possible to detect whether or not the agent has the capacity to dependably discharge all of them.

Finally, while explicit responsibility models are not required to detect responsibility fragility, they are useful, as discussed above, when responsibilities are dynamically assumed. Hence, in situations where there is no explicit backup agent, a responsibility model may help team members cope with the situation.

Our work on modelling responsibilities as patterns is still at an early stage and we need more experience to fully understand how these models can be useful in socio-technical systems design. However, the discussion here has shown that explicitly documenting responsibilities in a standard way can reveal vulnerabilities and hence we believe that responsibility models can be useful in designing dependable socio-technical systems.

9.5 References

Ackroyd, S., Harper, R., Hughes, J. A., et al. (1992). *Information Technology and Practical Police Work*. Milton Keynes, Open University Press.

Alexander, C. (1979). *A Timeless Way of Building*. Oxford, Oxford University Press.

Alexander, C., Ishikawa, S. and Silverstein, M. (1977). *A Pattern Language: Towns, Building, Construction*. Oxford, Oxford University Press.

Anderson, R. J., Hughes, J. A. and Sharrock, W. W. (1989). *Working for Profit: The Social Organisation of Calculability in an Entrepreneurial Firm*. Aldershot, Avebury.

Bentley, R., Rodden, T., Sawyer, P., et al. (1992). Ethnographically-informed Systems Design for Air Traffic Control. *Proc. CSCW'92*, Toronto, Canada, ACM Press.

Coplien, J. (1998). Patterns and Pattern Languages for Organisational Design. http://www.bell-labs.com/people/cope/Patterns/Process/process.html

Erickson, T. (1998). Towards a Pattern Language for Interaction Design. *Recovering Work Practice and Informing Systems Design*. C. Heath, J. Hindmarch and P. Luff, In preparation.

Gamma, E., Helm, R., Johnson, R., et al. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, Mass., Addison-Wesley.

Larman, C. (2002). *Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and the Unified Process*. Englewood Cliff, NJ, Prentice Hall.

Martin, D. and Sommerville, I. (2004). 'Ethnomethodology, Patterns Of Cooperative Interaction and Design'. *ACM Trans. on Computer-Human Interaction* **11**(1): 58-89.

Schmidt, D. C. (1997). Applying design patterns and frameworks to develop object-oriented communications software. *Handbook of Programming Languages, Vol. 1*. New York, Macmillan Computer Publishing.

van der Aalst, W. M. P. and ter Hofstede, A. H. M. (2005). 'YAWL: Yet Another Workflow Language'. *Information Systems* **30**(4): 245-275.

White, S. A. (2004). An Introduction to BPMN. http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf