

An Integrated Approach to Dependability Requirements Engineering

Ian Sommerville

Computing Dept., Lancaster University, LANCASTER LA1 4YR, UK.
E-mail: is@comp.lancs.ac.uk

Abstract. This paper discusses an approach to system requirements elicitation that integrates safety requirements elicitation and analysis with more general requirements analysis. We propose that the analysis should be organised round pervasive ‘concerns’ such as safety and security which can drive the requirements engineering process. The paper introduces the notion of concerns based on business goals and discusses how concerns are used to elicit information about system requirements from stakeholders. I also discuss briefly how concerns may be part of a more general requirements engineering method called DISCOS that integrates requirements engineering with high-level design. I use examples from a medical information system to illustrate how concerns may be used.

1 Introduction

The discipline of safety requirements engineering is well-established in industries such as the chemical industry and the nuclear industry where safety issues have been paramount for many years. Conventionally, for computer-based systems, safety requirements engineering is considered to be a separate process from more general system requirements engineering with safety requirements being derived either before or in parallel with more general system requirements. This is embodied in, for example, the IEC 61508 process [1] [2] which was designed to support the safety engineering of protection systems.

The requirements of a protection system are distinct from the requirements of the system being protected so there is some rationale for separating the safety analysis of these systems from more general requirements analysis. The approach may also be applied in a more general system requirements engineering process for critical systems. It focuses attention on the importance of safety and may derive critical requirements that take precedence over other functional and non-functional system requirements.

The identification of safety requirements usually follows a process of hazard and risk analysis. In this process, hazards are identified along with the risks of these hazards arising, the risks of an associated accident and the potential consequences of such an accident. Various techniques may be used to support hazard analysis such as HAZOPS [3] and fault-tree analysis [4] and these can be the basis for deriving safety requirements.

However, the separation of safety and system requirements engineering can lead to problems. There are three related difficulties with this separation:

1. It assumes that safety is a system property that can be considered in isolation from other system properties such as integrity and timeliness.
2. It assumes that safety requirements can be isolated and clearly identified.
3. It increases the difficulty of identifying requirements conflicts and the costs of resolving these conflicts.

Separating safety analysis from requirements analysis suggests that we clearly understand what is meant by ‘safety’ for the system being analysed. From a common sense point of view we can say that a safe system is one that does not cause damage to people or its environment. However, when we try to pin this down into a more precise definition that can be used as a basis for assessing a system we run into problems. A train which does not move or a turbine that does not turn is clearly safe but of little use. A critical information system that provides correct information but in a font that can only be read with difficulty by operators with normal eyesight may not, in itself, be unsafe but can lead to safety-related failures in the encompassing socio-technical system. Delays in delivering an information system for cancer screening as a result of problems in reconciling safety requirements and other requirements may mean that several patients die unnecessarily. Availability, usability and timeliness of delivery all may affect whether or not the system is ‘safe’ – safety is not a simple system property that can or should be isolated.

Because safety is a holistic system property, I am convinced that the notion of a ‘safety requirement’ as a special type of requirement that is distinct from other system requirements is potentially dangerous. While it may be applicable in the limited context of protection systems, I argue that it is more effective to consider safety as a pervasive ‘concern’ for the procurers, developers and operators of critical systems. Safety must be considered along with other concerns and that these concerns should drive the requirements engineering process for critical systems. While there may be specific requirements that focus on safety issues in a system, all system requirements can potentially affect system safety.

In the remainder of this paper, I develop this notion of concerns that reflect safety and other dependability properties and outline an approach to requirements elicitation and analysis where concerns structure this process. By using generic and specific questions associated with concerns, requirements can be elicited from system stakeholders. To illustrate the approach, I use examples drawn from a case study of an information system used to help manage the care of patients with mental health problems. In the final section of the paper, I briefly discuss a requirements and design method that uses concerns and that integrates system requirements engineering and high-level system design.

2 The MHCPMS system

In this section, I briefly introduce an example system that I draw on as a source of examples throughout the paper. This system is intended to help manage the care of patients suffering from mental health problems who may attend different clinics within a region. This based on a real system that is in use in a number of hospitals. For reasons of commercial confidentiality, I have changed the name of the system and have not included information about any specific system features.

The overall goals of the MHCPMS system are twofold:

- To generate management information that allows health service managers to assess performance against local and government targets.
- To provide medical staff with timely information to facilitate the treatment of patients.

The health authority has a number of clinics that patients may attend in different hospitals and in local health centres. Patients need not always attend the same clinic and some clinics may support ‘drop in’ as well as pre-arranged appointments.

The nature of mental health problems is such that patients are often disorganised so may miss appointments, deliberately or accidentally lose prescriptions and medication, forget instructions and make unreasonable demands on medical staff. In a minority of cases, they may be a danger to themselves or to other people. They may regularly change address and may be homeless on a long-term or short-term basis. Where patients are dangerous, they may need to be ‘sectioned’ – confined to a secure hospital for treatment and observation. These factors mean that safety is one of the issues that must be considered in the development and operation of this system.

Users of the system include clinical staff (doctors, nurses, health visitors), receptionists who make appointments and medical records staff. Reports are generated for hospital management by medical records staff. Management have no direct access to the system.

The system is affected by two pieces of legislation (in the UK, Acts of Parliament). These are the Data Protection Act that governs the confidentiality of personal information and the Mental Health Act that governs the compulsory detention of patients deemed to be a danger to themselves or others.

3 Concerns

Systems exist in an organisation to help that organisation deliver its organisational goals. We know that many systems that are developed are unused or fail to meet expectations. One reason for this is that the requirements for these systems either conflict with or are irrelevant to the organisational goals and constraints. Consequently, a good requirements engineering process should relate requirements to organisational goals and constraints.

Organisational goals reflect the overall purpose and priorities of the organisation so the goals of a hospital (for example) are derived from its purpose to treat people who are ill or injured. Examples of the goals of a hospital might therefore be:

- Provide a high standard of medical care for patients.
- Ensure that a high proportion of resources are spent on patient care.

As well as goals, organisations must operate within a set of externally imposed constraints. These constraints may be legal, governmental or social and reflect the environment in which the organisation operates. Examples of constraints on a hospital might therefore be:

- Operate within the funding budget as set by the local health authority.
- Provide monthly reports to government on numbers of patients treated.

Goals and constraints may be conflicting so trade-off decisions on how best to satisfy the goals while meeting the constraints are inevitable.

The need to satisfy organisational goals has resulted in the development of a number of goal-based approaches to requirements engineering [5-10]. These are based on refining vague objectives into concrete formal goals then decomposing these further into sub-goals until a set of primitive goals which can readily be expressed as system requirements has been derived. These approaches have the advantage that they expose different goals from different stakeholders and provide a structured approach to assessing alternatives.

However, the vague and abstract nature of organisational goals and constraints poses difficulties for goal-based approaches. The inherent messiness of the world means that a hierarchical decomposition of goals is inherently artificial. Furthermore, I believe that stakeholders prefer goals to remain loosely defined and hence resist detailed goal decomposition. Loosely defined goals allow for flexibility of interpretation in whether or not these goals have been reached. Consequently, I think that validating the goal-decomposition hierarchy is practically impossible. Other problems with goal-based approaches are discussed in a good summary of these techniques in a web site maintained by Regev [11].

To address the problems with goal-based approaches, we have introduced an intermediate concept called a 'concern' which helps bridge the gap between organisational goals and the requirements for systems being developed to support that organisation. Originally, we proposed that concerns should be integrated with a viewpoint-oriented approach to requirements elicitation [12, 13]. However, we are now convinced that 'concerns' are more generally applicable and can be applied in conjunction with any systematic approach to requirements elicitation and analysis.

Concerns, as the name suggests, reflect issues that the organisation must pay attention to and which are central to its operation. Concerns are identified from organisational goals by asking 'What do we need to think about if we are to achieve goal X or meet constraint Y'. Notice that concerns are not about what to 'do' but rather are a way of explicitly identifying the key issues around a goal.

Concerns correspond to high-level strategic objectives for the system. They are established after discussion with strategic management and are first expressed at a very high level of abstraction. They are frequently common to applications within the same domain. In general, it should be possible to express concerns using a single word or phrase and to explain in one or two sentences how these concerns are linked to organisational goals.

Some of the concerns that affect systems being developed in a hospital and their link to organisational goals might include:

1. Safety – hospitals must ensure that patient care is safe; from a legal standpoint, hospitals must work within national health and safety legislation.
2. Information quality – information quality is important for patient care and for providing timely and accurate reports to government about the functioning of the hospital.
3. Staffing – recruiting and retaining high-quality staff is essential to deliver a high standard of patient care.

Of course, there are many more concerns in a hospital and, when considering computer-based system development, it is important to decide which concerns are relevant. For example, privacy is obviously central in a medical records system but much less significant in a system that schedules ward cleaning. Concerns should

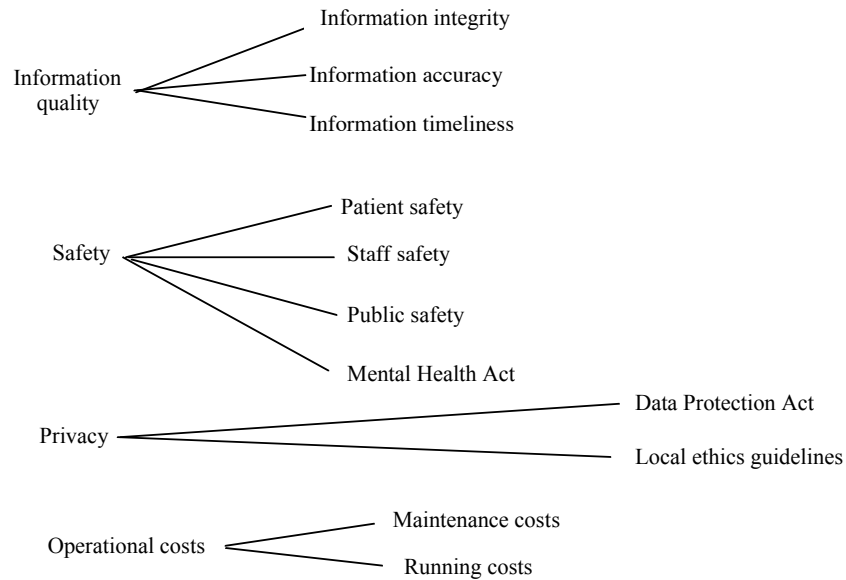


Figure 1 Decomposition of concerns in the MHCPMS

be ruthlessly prioritised so that only a small number of key concerns should drive the requirements engineering process.

For critical systems development, concerns are closely aligned with the dependability attributes that are most important for that system. Therefore, for a business-critical e-commerce system, the principal concerns are likely to be security and availability; for a control system for a radiation therapy machine, the principal concern is probably safety; for a telescope control system, the concerns may be reliability and accuracy.

Concerns, therefore, are a means of addressing the problems that arise when safety requirements elicitation is separated from more general systems requirements engineering. Where safety is important, it should be one of the concerns that drive the requirements engineering process. Safety issues can still be highlighted and analysed separately but the safety analysis is integrated with other analyses based around other concerns. By concern-cross checking, we can look for potential conflicts between requirements corresponding to the different concerns and can consider how other system requirements may have safety implications.

In the MHCPMS system, we have identified the principal concerns as:

1. Safety – the system should help to reduce the number of occasions where patients cause harm to themselves or others. The provisions of the Mental Health Act must be considered.
2. Privacy – patient privacy must be maintained according to the Data Protection Act and local ethical guidelines.
3. Operational costs – the operational costs of the system must be ‘reasonable’.
4. Information quality – the information maintained by the system must be accurate and up-to-date.

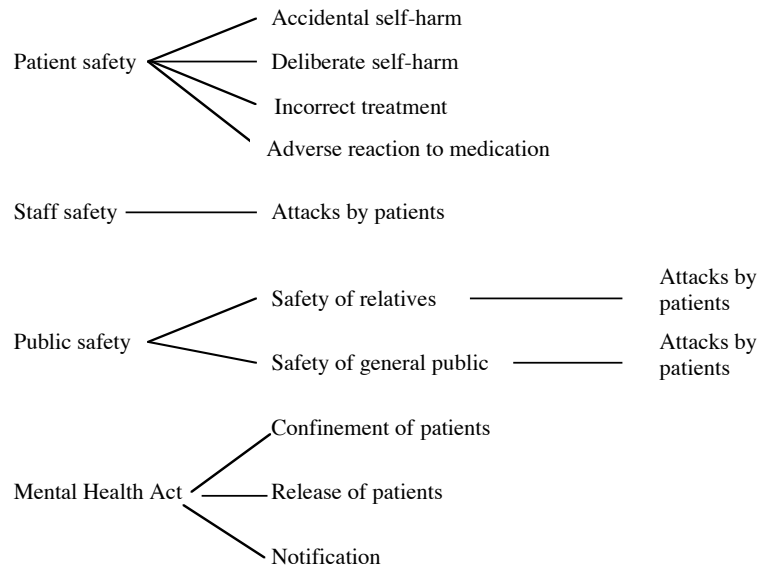


Figure 2: Further decomposition of the safety concern

High-level abstract concerns are decomposed into sub-concerns which reflect different facets of the concern. The first level decomposition of concerns into sub-concerns for the MHCPMS is shown in Figure 1. From these sub-concerns, a set of questions are derived. The outcome of the concern decomposition process is a set of questions, grouped by concern which are used during the requirements elicitation and analysis process. These questions are used to elicit information from system stakeholders and, from this information, system requirements are derived.

At this level of decomposition, concerns are still vague reflections of issues that the organisation considers to be important. To break these down into more detailed concerns, we ask ‘what are the issues’ questions such as ‘what are the issues around patient safety that are of concern for the system’. This results in a further level of decomposition as shown in Figure 2.

Patient safety concerns the health and well-being of the patient themselves. Two of these are generic to all medical situations namely incorrect treatment and adverse reactions to treatment. The other two are more specific to mental health situations where the often confused nature of patients can result in accidental self harm and, sometimes, deliberate self-harm to gain attention.

The nature of patients suffering from mental health conditions means that they may attack other people. Although the threat is the same for medical staff, relatives and the general public, the risks and the situations where attacks might take place are different. Consequently, these are identified as separate concerns.

Finally, the Mental Health Act is concerned with both the safety of the public and the rights of patients. Legal formalities have to be followed when patients are confined and released, confinement can only be for a limited time without further

examination and various people have to be notified when a patient is confined and released.

The process of decomposition continues by asking ‘what are the issues’ questions until these become difficult to answer. So the issues around the safety of relatives and the general public are ‘Attacks by patients’ but further decomposition into, say, types of attack is not needed for requirements derivation.

At this stage, the decomposition of concerns switches from identifying sub-concerns to identifying questions that may be used to elicit a deeper understanding of these issues and possible system requirements. In the case of an information system such as the MHCPMS, there are a number of generic questions that can be used as a starting point for these questions:

- What information from the system relates to the sub-concern being considered?
- Who requires this information and when do they require it?
- How is this information delivered to users of the information?
- What constraints does this concern impose on the system?
- What are the consequences of failing to deliver this information?

These generic questions may be decomposed into more detailed questions for system stakeholders or may be sufficient on their own to gather information about system requirements.

Possible answers to these questions about deliberate self-harm are:

1. Information about previous history of self-harm or threats of self-harm made during consultations
2. Medical staff during consultations. The patient’s relatives or carers.
3. Can be delivered directly to medical staff using the system. Must be delivered to relatives and carers via a message from the clinic.
4. No obvious constraints are imposed by this.
5. Failing to deliver the information may mean that a preventable incident of self-harm takes place.

As well as the generic questions that may be used to gather information from system stakeholders, sub-concerns may be decomposed into more specific questions. For example, if we consider the information accuracy concern, then the generic questions are inappropriate and more specific questions associated with the concern may be derived. For example:

- How can potential inaccuracies in manual records be detected?
- If clinical staff selected inputs from a menu of options (to avoid inaccurate inputs) what problems might this cause?
- Is accuracy in some parts of the record more critical than others?

Notice that system functionality is not normally a concern. Goals are usually unaffected by details of the functionality provided by a system. However, where there is a clear link then ‘System functionality’ could be identified as a concern. We assume that, as part of the requirements engineering process, information on

functionality is elicited by asking questions such as ‘what user tasks are supported by the system?’, ‘what information is needed to carry out these tasks?’, etc.

2.1 Concern cross-checking

A generic problem in complex systems is requirements conflicts where different system requirements are mutually contradictory. Conflicts are inevitable because different system stakeholders have incompatible goals and because of the interactions between the overall organisational goals and the constraints imposed on the organisation. Ideally, conflicts should be identified at an early stage of the requirements engineering process and resolved through negotiation. In practice, however, conflicts can be subtle and difficult to find in the detail of the system requirements. They may only emerge at later stages of the development process with the consequence that requirements changes and consequent rework become necessary.

The notion of concerns provides a mechanism that partially addresses the problem of detecting requirements conflicts as it allows cross-checking to be carried out at a higher level of abstraction than the requirements themselves. Rather than looking for conflicts in the requirements, we can use the concerns and our general background knowledge of these concerns to discover areas of potential conflict.

As there should be a relatively small number of concerns we can compare them in pairs to assess whether or not conflicts are likely to arise. For example:

1. *Safety and information quality.* Safety is dependent on accurate information. Conflicts are only likely if requirements on information quality allow records that are known to be erroneous or out-of-date to remain in the system.
2. *Safety and privacy.* Redundancy is an important mechanism for achieving safety but this requires information sharing. Privacy may impose limits on what information can be shared and who can access that information.
3. *Safety and operational costs.* There is always a trade-off between costs and safety so there is a potential for conflict here.

All we have done here is to highlight areas where conflicts are most likely and we need to turn to more detailed decomposition of the concerns to questions. We then can explore them in more detail.

Consider the privacy concern and its sub-concern of the Data Protection Act and the possible answers to the above generic questions about information, information users, information delivery, constraints and consequences of non-delivery. Possible answers to the generic questions identified in the section above are:

1. All information in the system that relates to identifiable individuals is covered by the data protection act.
2. All staff using the system need to be aware of the requirements imposed by the data protection act.
3. There are no information delivery requirements generated from this concern.

4. The constraint imposed is that personal information may only be disclosed to accredited information users where information users are people who need to do the information to do their jobs such as doctors or nurses.
5. Failure to address this concern could result in legal action being taken by data subjects.

The starting point for concern cross-checking is the constraints that are identified with each concern. We take these constraints and look at the relationship between them and the answers to questions generated in other concerns. Here we see an immediate conflict between the privacy constraints imposed by the DPA and the safety issue of making information about the possibility of self-harm known to the patient's relatives and carers (and possibly some medical staff). To improve safety, it makes sense to tell the patient's relatives about the possibility of self-harm. However, the patient may not wish their relatives to know of previous incidents so any dissemination of this information is not allowed by the DPA. There should not, therefore, be a requirement to generate information for patient's relatives included in the system.

2.2 Requirements derivation

Requirements are derived from the answers to the concern questions that are provided by system stakeholders. There is not a simple 1:1 relationship between the answers and requirements and it is up to the analyst to assess the answers and generate requirements from them. These should then be taken back to the stakeholders for validation.

Some examples of requirements that might be generated for the MHCPMS are:

1. The system shall provide fields in each patient record which allow details of incidents or threats of deliberate self-harm to be maintained.
2. The records of patients who have a history of deliberate self-harm shall be highlighted when accessed to bring them to the attention of clinical system users.
3. The system shall have a facility to generate e-mails to other accredited medical staff that warn about at-risk patients who may harm themselves deliberately.
4. The system shall only allow the transmission of personal patient information to accredited staff and to the patient themselves.

By using the answers to questions to generate requirements, we can avoid the problem of system stakeholders generating requirements that are too specific or which reflect their pre-conceptions about how the system should be designed.

4 Concerns and hazard analysis

The notion of concerns set out here has been illustrated using an system where hazard-driven analysis is not the norm. However, the concerns-based approach is completely consistent with hazard analysis for deriving safety requirements. To use concerns in conjunction with hazard analysis, the safety concern is decomposed as before into sub-concerns. However, after the principal sub-concerns have been

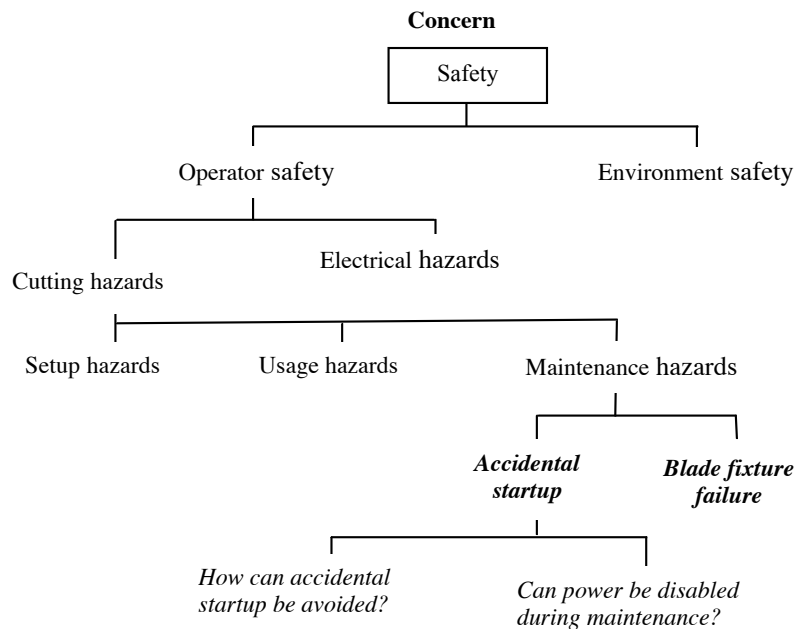


Figure 3: Concerns as hazards

identified, rather than decompose the concern into questions, each sub-concern is may be decomposed into a set of related hazards.

This is illustrated in Figure 3 which shows a hazard decomposition for a paper guillotine. You can see from this diagram that each identified hazard is represented as a sub-concern of safety and that the questions associated with concerns are designed to elicit information about how to avoid the hazard or mitigate the consequences of an accident.

5 Using concerns in a requirements engineering process

The notion of concerns as a driver for the requirements engineering process was first set out by us in a number of papers that described work on a requirements engineering method called Preview [12, 13]. Since then, we have developed a new requirements engineering method called DISCOS where we have extended the notion of concerns and have integrated early conceptual design with the process of requirements elicitation and analysis.

In the DISCOS method, we propose a spiral approach to requirements engineering as shown in Figure 4. Each stage in this process is:

1. *Define concerns.* As discussed here, establish the main concerns that influence the system and decompose these to sub-concerns and questions.

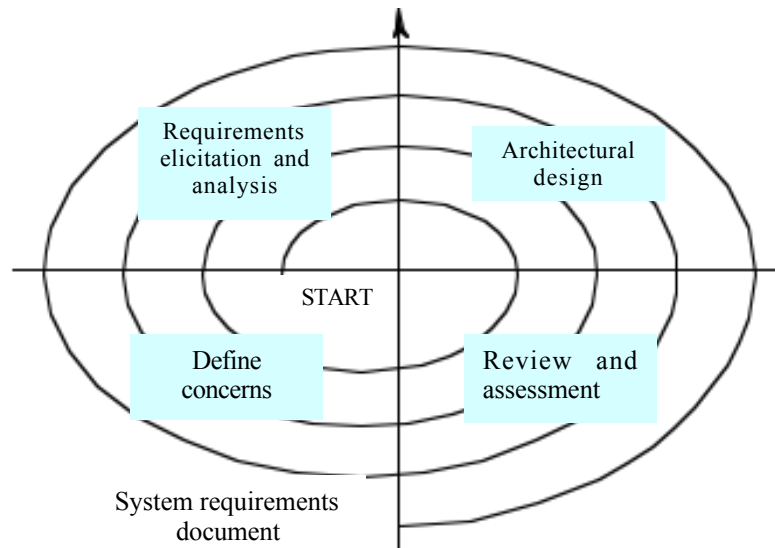


Figure 4: The spiral model of the DISCOS process

2. *Define system requirements.* Using the questions as a basis for the requirements elicitation, collect answers from system stakeholders and use these to define the requirements for the system.
3. *Propose system architecture.* Define an architectural model for the system showing sub-systems and their interactions.
4. *Assessment and review.* Review the requirements and model. If these are incomplete or incorrect, enter a new round of the spiral where more information may be collected and review issues addressed.

We propose a spiral process as we believe that creative human activities such as requirements engineering are never sequential processes. Concern definition, requirements elicitation and analysis and systems architectural design are always inter-leaved. In practice, irrespective of concerns, some requirements may be fixed at a very early stage in the system definition process. Equally, decisions on the system architecture may be made to allow legacy systems to be integrated, to reuse architectures with known characteristics and to help structure the process of requirements definition.

During each round of the spiral, more or less information may be added for each stage. The initial round is, essentially, a baseline round where high-level concerns are established as well as any requirements that are pre-defined for the system and, perhaps, an outline architectural design. Further early rounds focus on decomposing concerns with requirements ‘popping out’ as these concerns are established. Later rounds of the spiral are more focused on detailed requirements definition. We do not separate requirements engineering and high-level system design. This reduces the probability of proposing requirements which cause conflicts at the design stage.

The notion of concerns is pervasive and critical to the DISCOS method. We have discussed here how concerns can support the elicitation of requirements but

they are also used to structure the review and assessment activity. Using concerns, this activity can assess whether design proposals may conflict with organisational goals.

6 Conclusions

The essential message of this paper is that separating the processes of eliciting and analysing safety requirements from more general processes of requirements engineering is undesirable. Such a separation is likely to result in safety requirements that conflict with other requirements with rework required to resolve these conflicts. Furthermore, it tends to isolate safety rather than emphasise that it is an issue of universal concern. The notion of cross-cutting concerns derived from high-level organisational goals avoids this separation while still maintaining the possibility of separate risk-based safety analysis.

Our original work on concerns focused on control systems but, more recently, we have been exploring how to use this approach for critical information systems. Methods of risk and failure based safety analysis are rarely applicable to this type of system. System failure does not have immediate implications for safety but, as we have seen with the MHCPMS, the system design should take safety into account. By highlighting critical issues such as safety and security as concerns, we can ensure that they pervade the requirements elicitation process and are not considered after the system functionality has been established. We are therefore now focusing on the use of concerns to support the requirements engineering of critical information systems and are planning further experiments in this area to develop the DISCOS method.

7 Acknowledgements

This research has been partially supported by the European Community's Framework Programme of IT research in the BANKSEC project (IST-1999-20711) where the DISCOS method was developed as project deliverable WP2.D3. For more information see <http://www.atc.gr/banksec>.

References

1. Redmill, F., *IEC 61508: Principles and use in the management of safety*. IEE Computing and Control Engineering J., 1998. **9**(10): p. 205-13.
2. IEC, *Standard IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems*. 1998, International Electrotechnical Commission, Geneva.
3. Chudleigh, M.F. and J.R. Catmur. *Safety assessment of computer systems using HAZOP and audit techniques*. in *Proc. SAFECOMP'92*. 1992: Pergamon Press.
4. IEC, *International standard 1025: Fault tree analysis*. 1990, International Electrotechnical Commission: Geneva.
5. van Lamsweerde, A., R. Darimont, and P. Massonet. *Goal-Directed Elaboration of Requirements for a Meeting Scheduler*. in *Proc. RE'95*. 1995. York, England: IEEE Computer Society Press. 194-203.

6. Anton, A.I. *Goal Based Requirements Analysis*. in *Proc. 2nd Int. Conf. on Requirements Engineering (ICRE'96)*. 1996. Colorado Springs: IEEE Computer Society Press. 136-44.
7. Dardenne, A., A. van Lamsweerde, and S. Fickas, *Goal-Directed Requirements Acquisition*. *Science of Computer Programming*, 1993. **20**: p. 3-50.
8. Fickas, S., Van Lamsweerde, A., and Dardenne, A. *Goal-directed concept acquisition in requirements elicitation*. in *6th Int. Workshop on Software Specification and Design*. 1991. Como, Italy: IEEE CS Press. 14-21.
9. Mylopoulos, J., L. Chung, and E. Yu, *From Object-oriented to Goal-oriented*. *Comm. ACM*, 1999. **42**(1): p. 31-7.
10. Rolland, C., C. Souveyet, and C. Ben Achour, *Guiding Goal Modeling using Scenarios*. *IEEE Trans. on Software Eng.*, 1998. **24**: p. 1055-71.
11. Regev, G., *Goal-Driven Requirements Engineering Overview*. <http://lamswww.epfl.ch/Reference/Goal/Default.htm>. 2002.
12. Sommerville, I. and P. Sawyer, *Viewpoints: principles, problems and a practical approach to requirements engineering*. *Annals of Software Engineering*, 1997. **3**: p. 101-30.
13. Sommerville, I., P. Sawyer, and S. Viller. *Viewpoints for requirements elicitation: a practical approach*. in *Proc. Int. Conf. on Requirements Engineering*. 1998. Colorado. 74-81.