Requirements Elicitation for Complex Safety Related Systems

David Bush [†], Anthony Finkelstein [‡]

[†] National Air Traffic Services Ltd.
[‡] Department of Computer Science, University College London

Abstract: *National Air Traffic Services Ltd. (NATS) has a major and ongoing investment programme in infrastructure to support its role of providing safe Air Traffic Management Services for the UK. An important enabler for the delivery of safe and successful systems is correct and traceable requirements. This paper describes early experiences of a case study in system development using a goal-directed approach to requirements engineering. It concludes that the work so far promises many benefits to NATS in the early concept and development stages of new systems.*

1. Introduction.

National Air Traffic Services Ltd. (NATS) plans, provides and operates a safe integrated Air Traffic Management Service for the United Kingdom. Approximately 25% of NATS staff are engineers and about 70% of its fixed assets are in technical infrastructure – nearly half of this currently in the course of development and implementation. This investment rate will continue - in part the current Government plans to introduce a Public Private Partnership (PPP) for NATS is driven by the future need for large scale investment in technical infrastructure to match capacity with future demand – which is currently growing at between 5% and 7%. It is in this context, with the added necessity to deliver technical systems that can be shown to be safe, that NATS has identified the importance of a rigorous and reliable approach to identifying, recording and maintaining correct requirements for its systems. This paper describes the experiences of NATS' Department of Technical Research and Development in the validation and ongoing development of a model and method for Requirements Engineering.

2. Requirements Model.

Requirements Engineering has been described (Zave [1]) as:

"……… the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families."

The approach we have adopted is grounded in a model of the information needed in order to achieve this, and a comprehensive description of the way such information inter-relates to achieve the tracability through time and through function. The whole model draws on three complementary and interlocked aspects: the goals for the system; the real world assumptions made about its environment; and the services and architecture planned to meet the goals.

[1] Zave, P. (1997). Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys*, 29(4): 315-321.

The goal directed approach draws on previous work in this area such as that of KAOS[2] starting from high-level concepts of system purpose and concept and progressively refining these until they can be operationalised. Goals are then linked in to the architecture from which they will be met. These are described in a way compatible with the 'component and connector' approach to software architecture to allow the information recorded in the requirements model to be incorporated into an Architectural Description Language (ADL) for analysis. Finally the components in the model can be traced to the real-world assumptions through system interfaces. This enables the separation between phenomena of the world in which our system will operate, and the phenomena we wish our system to make true in the world - identified by Jackson[3] as being critical to requirements clarity.

However, while the model itself covers the breadth of this remit, this paper reports only the experiences gained in elicit the system goals and documenting them. To this extent it has only addressed part of the whole information model and elicitation process described above.

3. The Case Study.

The case study chosen for the validation of the model is a live R&D project in NATS – the Minimum Safe Altitude Warning system (MSAW) a system mandated by ICAO. This is intended to reduce the occurrences of Controlled Flight into Terrain (CFIT) by providing advance warning to pilots flying below the mandated minimum altitude. This project is in an early R&D stage where overall system scope and stakeholder goals are being identified, and critical aspects of the system are prototyped before full development is carried out.

4. MSAW Requirements Elicitation.

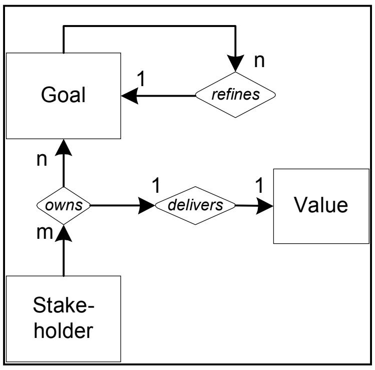

Figure 1 – Goals Information

The relevant extract of the requirements information model is shown in Figure 1. This stage of the MSAW project was associated with populating the information model with goals, stakeholders and associated values.

Two methods were used to identify potential goals: group meetings and individual semi-structured interviews. The notes from these meetings were recorded and later analysed by the requirements team. It soon became clear that, partly because of the novelty of the requirement (and partly because it is the nature of requirements elicitation), the opinions offered in the meetings could not often be described as goals. We decided to term these artefacts 'contributions', and the team then needed to take on the role of classifying these, and recording them.

It was found that these contributions fell into a number of categories:

[2] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-Directed Requirements Acquisition. *Science of Computer Programming*, 20:3–50, 1993.

[3] M. Jackson. The World and the Machine. In *Proceedings of the 17th International Conference on Software Engineering*, pages 283–292. ACM Press, 1995.

They could indeed be goals; they could contribute to an existing goal; they could be a value statement about goals; they might be constraints – all of which we could fit into this part of our information model. Frequently however, they were none of these things - they might be assumptions about the real world, they might be opinions about solutions or the identification of further stakeholders, these required recording elsewhere in the information model. It was then necessary to include tracability from these contributions to practically every other information type in the model.

Two types of goal structures were created, one describing the system context, and one describing the system itself. Because the goals have a 'refinement' relationship these structures essentially formed trees, with the most general goal at the top, and the most detailed goal at the bottom. A simplified example of the 'system context' goal hierarchy is shown in Figure 2.

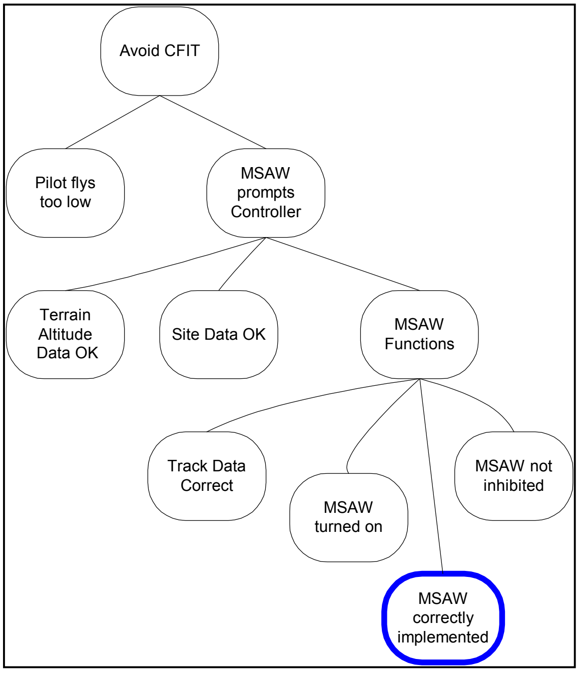

Figure 2 – System Context Goals

This shows the dependency of the MSAW system on its environment, which helps system designers to ensure that goals beyond the technical system are explored early in the lifecycle. In this case people and procedures will need to be developed to obtain and input correct terrain and site data, that the system providing the track data is identified and characterised, and that procedures exists for activating and deactivating the system. In addition to these external activities the system context goals help to identify interfaces needed within the system under development.

A goal hierarchy for the system itself partners this context hierarchy. This is essentially the refinement of the bold goal in the context hierarchy.

4. Some Experiences

The Information Model

We believe that having an information model as the basis of requirements elicitation is essential if any degree of control is to be maintained. Although we have found a need to adapt and extend the model, without the initial starting point we would have had no common concept of what the variety of information we were receiving was representing, and no way of classifying it. By using the information model the requirements team was able to address each contribution, discuss it in a common agreed framework, and classify it.

Without the model, contributions would either be classified as 'requirements' or would be omitted altogether, both of which would have been mistakes. This was particularly important when issues of system boundaries were under discussion, on a number of occasions we found real-world assumptions buried within a contribution. Without the framework of an information model that distinguished between these they could easily have accidentally ended up as system requirements.

Using a Goal Directed Approach

By using the goal directed approach we found ourselves constantly trying to define our top-level goal, challenging our own understanding of the scope and purpose of the system. Indeed our identification of a need for two goal hierarchies stemmed out of the search for clarity in the 'top level' goal of the MSAW system. We also found a number of other advantages:

- We were able to incorporate contradictory contributions – where different stakeholders wanted or did not want a certain goal.
- We found the ability to present goals graphically as in Figure 1, which assisted in gaining a common understanding through the team. This 'tree-structure' also provided a mechanism to boost our confidence in the completeness of the goals – as each goal refined into only 3-10 more detailed goals we became more confident that they represented a complete description of what was required – or else prompted further questions.
- We found the goal based approach allowed us to think of MSAW in terms of a product family, with 'core goals' being capable of separate refinement depending on the specific customer needs.

We also had some difficulties. It proved difficult to maintain our own focus on the general nature of the goals without refining them so far that they became solutions. Further, the graphical version of the drawing hierarchy resembled structured design drawings, which led occasionally to a tendency to imply sequence into the goals, which proved a distraction.

We also moved too quickly into recording our goals and contributions into our formal requirement management tool. At the earliest stages the goal hierarchy was subject to revision at almost every meeting, and the overhead of keeping the database up to date was overwhelming. We found that such tracability was not necessary at an early stage, while our understanding of the problem was being formed, and the goal structure was fluid. Paper and pencil provided all the necessary tracability at the early stages, which could then be transferred into the database once it reached a fairly firm stage in its evolution.

6. Conclusions

The activity underlying the work reported in this paper is the validation of the information model and a process supporting requirements elicitation within that model. Although much still needs to be done in examining the use of the goal-based model, we have concluded that it offers a number of very useful characteristics for NATS future systems development, particularly:

- It encourages and enables the system context to be discussed and recorded, so aiding allocation of responsibilities to people or procedures at an early stage.
- It aids in explicit discussion and decision on system boundaries.
- It aids communication of high-level system behaviour.
- It provides tracability from high level behaviour to stakeholders.