Information Structures for Traceability for Dependable Avionic Systems

Sheena Pearson[1] *and Amer Saeed*[1,2]
[1]BAe Dependable Computing Systems Centre
[2]Department of Computing Science
University of Newcastle upon Tyne, NE1 7RU, UK

Abstract

As the complexity of computing systems increases, the provision of an effective traceability scheme is recognised as essential for the efficient management of system development. Several approaches have been advocated for enhancing traceability, a central concern is to provide appropriate structures for recording the artifacts produced during system development. In this paper, we present a traceability procedure applicable to the development of dependable avionic systems. The procedure was developed by customising a set of generic information structures that support the recording and manipulation of design rationale. The customisation process was directed by an exposition of development practices for dependable avionic systems, in terms of a set of complementary models of a development context. To examine the effectiveness of the procedure and guide further work a case study based on an high−integrity flight control subsystem is employed.

Keywords: traceability, dependable systems, design rationale, information structures.

1. Introduction

An essential aspect of managing system development is to provide effective means/mechanisms for achieving traceability between the artifacts produced during system development. Traceability has become more important in recent years due to the use of software in safety critical applications which require certification, and the development of more complex and larger systems leading to greater interaction between different domains. The growing emphasis on producing quality products to specified standards (e.g. IS0 9001) requires assessment of development practices (e.g. SEI Capability Maturity Model /Paulk 93/) where issues relating to traceability are raised.

Requirements traceability has therefore been one of the main subjects of recent requirements engineering research, /IEE 91/, /Gotel 94/, /Pohl 94/, /Gotel 95/, and /Ramesh 95/. The work in /Gotel 95/ concentrates on pre−requirements traceability ("refers to those aspects of a requirements life prior to inclusion in the Requirements Specification" /Gotel 94/) proposing an approach based on contribution structures addressing the problems related to the absence of adequate information relating to the human sources of requirements. Ramesh et al detailed the type of information which is required to support traceability, the different views which stakeholders have of this information and the results of

implementing a traceability process into an organization. The NATURE project /Pohl 95/ looks at traceability focusing on concurrent engineering and proposes implementing traceability by the use of formal product models with tool support to automate the interrelationships.

This paper proposes a traceability procedure applicable to the development of dependable avionic systems based on structures to support the recording and manipulation of design rationale /Klein 93/. Special characteristics of dependable avionic systems must therefore be considered, typically these are large scale projects which involve several teams each with specialist knowledge and often in numerous geographic locations. The systems are real time, complex and often safety critical, requiring certification by regulatory authorities and an examination of failure behaviours through the process of safety analysis.

To identify a traceability procedure numerous activities were carried out relating to the elucidation of the basic information categories and the linkages between them. These included: the identification of the information to be recorded, resulting in a top level information model; the identification of the artifacts produced (documents etc.); identification of the actual data and data types required (detail of an information model); a definition of the process model for the activities performed during the development including the activities required to support traceability; the identification of notations and structures to support traceability of the information and activities identified.

Discussion with engineers and management in the avionics industry highlighted concerns relating to the support for certification /MOD 91/ and change control. The proposed traceability procedure focuses on support in these areas providing a structure to record design rationale in context (found to be critical for effective change management), links with a safety analysis process, and an improved development process leading to the ability to perform a more rigorous analysis of the data at an early stage.

The rest of this paper is organized as follows. In Section 2, the context for the traceability is established by the identification of complementary models of system development. In Section 3, we discuss desirable characteristics for a traceability procedure and present the proposed procedure. Section 4 presents a case study, identifies lessons learnt and examines the effectiveness of the proposed procedure. Section 5 discusses issues related to tool support, Section 6 presents some concluding remarks.

2. System Development

In order to devise an effective procedure for traceability, it is essential that the structures employed to record the basic elements of information and their interrelationships reflect the system development context. We address this concern by constructing a comprehensive description of a development context. Such a description will need to encompass a number

of different projections of system development, and relate the projections to the artifacts between which traceability is to be achieved.

2.1. Development Projections

We have employed four complementary (interrelated) models to describe a development context for dependable avionic systems: *information model*, *process model, documentation model* and *enterprise model*.

- **Information Model**. Central to the provision of a procedure for traceability is the definition of an information model. The purpose of the information model is to identify the basic categories of information to be recorded (e.g. requirements, constraints) and the linkages between the different information categories, e.g. derives, verifies. Typically an information model is represented as an entity−relationship diagram, the entities representing the categories of information and the relationships representing potential links between them. Establishing an information model for a system context is a pre−requisite for the employment of several traceability support tools, such as RTM and DOORS.
- **Process Model**. Process models are the traditional and most extensively examined approach to describe system development, typically they present a view of systems development that describes the different development activities and their interrelationships. The development artifacts between which traceability is to be achieved will emerge from the development activities, and the relationship between the artifacts is a reflection of the activities performed on the artifacts.
- **Documentation Model**. The primary interest of system developers is not the primitive artifacts produced during system development, rather they are concerned with documents which "package" these artifacts into pre−defined formats. The purpose of the documentation model is to describe the different documents that are produced during system development and their interrelationships. After an examination of a proposed set of relationships for structures between document artifacts /Gotel 95/, three relationships were investigated in our study of the development context: *containment*, where a document is part of another document; *reference*, where a document refers to another document; *dependency*, where the existence of a document depends on the presence of another document.
- **Enterprise Model**. An important aspect for understanding system development is the structure of the organizations that will participate in system development, /Strens 94/. For the application domain of interest, this will include: the *developers*, *certification authorities* and the *user community*. For each organization the enterprise model should identify the different roles and the relationships between them; in addition the roles involved in the interactions between organisations should be delineated.

2.2. Relating the Projections

For the purpose of traceability the central projection is the information model, the relationships between this model and the other projections is discussed in terms of its components (i.e. set of entities and relationships).

- **Development Artifacts**. The elements of information that appear as inputs or outputs of the activities of the process model should be recorded in some part of the information model. When a process model is to be studied with respect to a proposed information model, the consistency of the information content of both models should be confirmed.
- **Enterprise Roles**. The enterprise model defines a domain of information that must be incorporated into the information model. Each of the roles must be connected to the information model, otherwise there is some redundancy in the enterprise model or the information model is incomplete. The roles of the enterprise can be partitioned into three meta−classes: *development*, *quality control*, *management*. Each of the meta−classes will focus on particular abstraction of the information model, e.g. roles derived from the quality control class will be primarily concerned with rationale.
- **Documentation**. In order to achieve effective traceability, the relationship between the documents and the primitive artifacts must be established. These relationships can be exploited to generate predefined documents from a repository of information and to extract elements of information from a document to add to a repository of information. Two roles can be envisaged for the repository of information: *project specific*, to support development of a project; *generic*, to support development of several projects.

2.3. Notations, Techniques and Tools

The notations that will be used to represent specifications and related information (e.g. rationale), must be examined for their adequacy with respect to the information model. In addition when a mixture of notations are used the information that they represent should be categorised in terms of the information model, as should the associations between the different notations. The degree of rigour with which the basic entities of the information model are represented, dictates the extent to which rigour can be employed for traceability. If the basic entities are represented informally then it will not be possible to achieve even syntactic traceability. If the basic entities are represented in some structured notation, the provision of syntactic traceability becomes feasible. For *semantic* traceability (i.e. semantic relations between subsets of information) the entities involved must be represented in a formal notation.

2.4. Example Models

In this section we present some examples of the projections presented in section 2.1; here we report on the information, process and documentation projections. Work on the enterprise model is currently in progress.

2.4.1 Information Model

Some of the entities and relationships of the top−level information model are depicted in Figure 1, in a structure which is similar to that proposed in /Ramesh 95/. The central entity refers to the requirements, this is directly related to a set of core entities (such as strategies) and other entitles are related indirectly to the requirements. The full model has been encoded into a number of traceability tools.

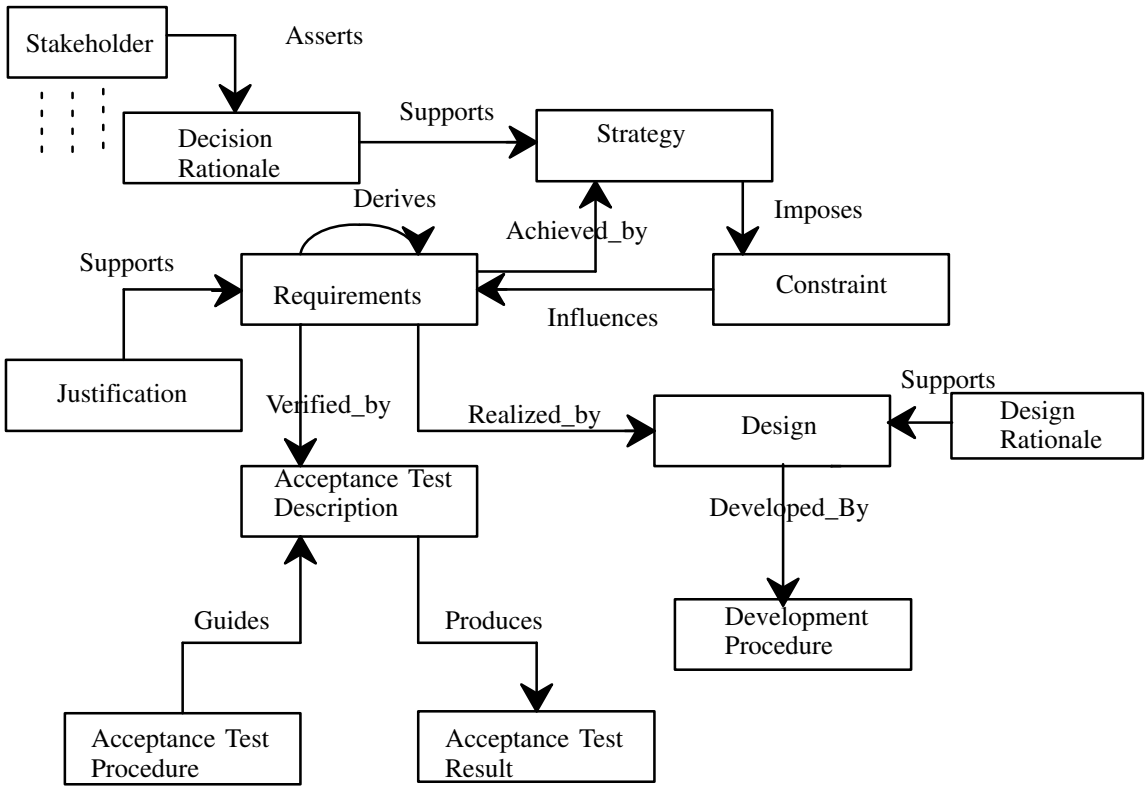

Figure 1: Extract of Top Level Information Model

2.4.2 Process Model

An extract of the process model for the early stages of development is depicted by a SADT diagram in Figure 2. The boxes represent activities, data transformed by an activity is represented by the side arrows, control data by arrows from the top and mechanisms by arrows from the bottom. To incorporate the traceability procedure an additional activity “Record Rationale” (depicted by the broken box) was introduced to collect information, that would otherwise be discarded or recorded in an unstructured format.

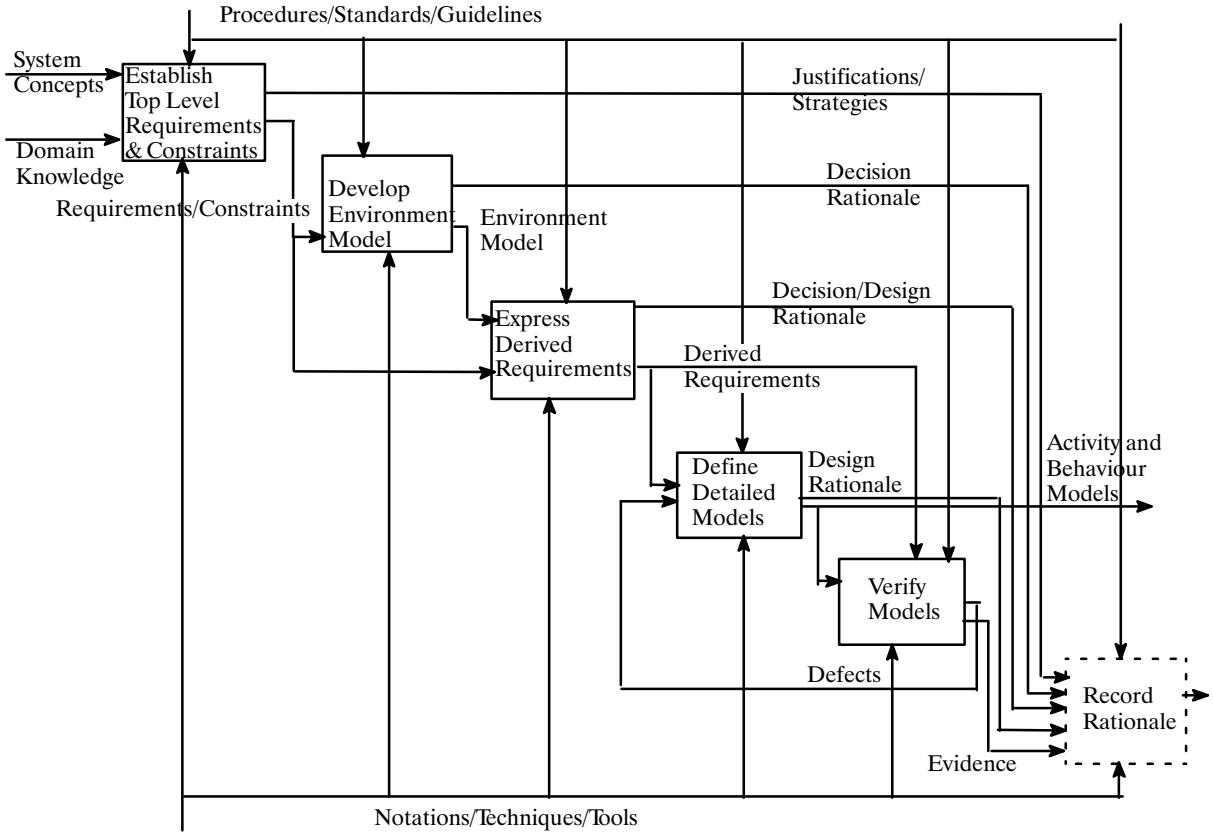

Figure 2: Extract of Process Model

2.4.3 Documentation Model

An extract of the documentation model is depicted in Figure 3. The System Concepts document contains several sections, including Functional Requirements and Non Functional Requirements. The Software Requirements document has a dependency to the System Concepts and refers to Standards and Guidelines.

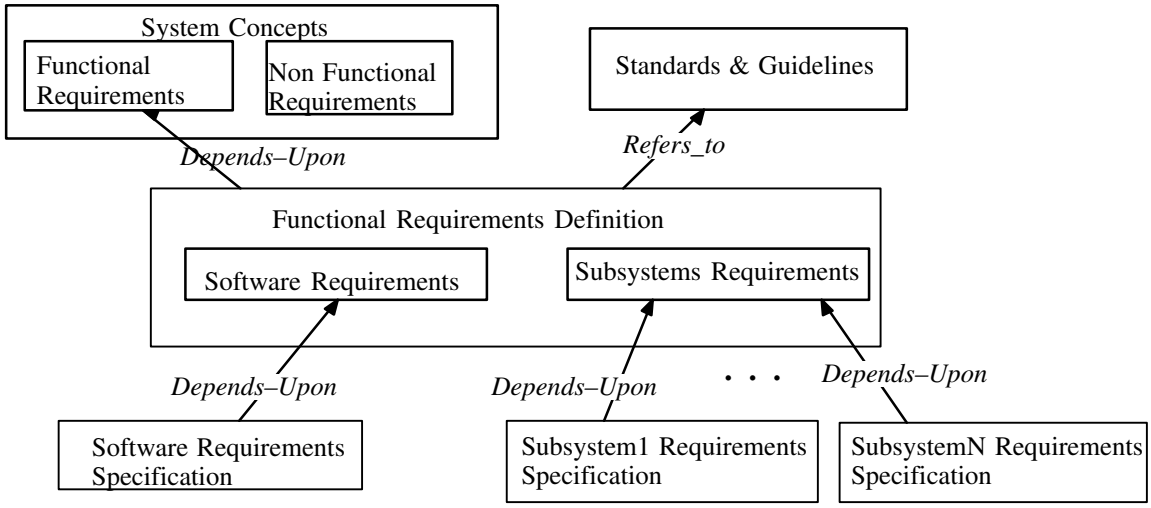

Figure 3: Extract of Documentation Model

3. Traceability Procedure

In this section, we present a procedure for enhancing traceability that can be incorporated into a development process. Firstly, we present a set of desirable characteristics for a traceability procedure, this is followed by a description of the underlying approach and then the customisation of the approach.

3.1 Desirable Characteristics

Desirable characteristics of a traceability procedure, are grouped into three broad categories, based upon the activities for requirements engineering: *elicitation*, *expression* and *analysis*.

- *Elicitaton*. An elicitation strategy that identifies relevant information categories during system development and the relationships between the information. An elaboration strategy which generates prompts and questions from the elicited information and incorporates responses.
- *Expression*. The collected requirements information should be organized in accordance with a set of structuring principles in order to facilitate the extraction of different projections of the information (e.g. safety properties). A set of interlinked information structures should be provided to encompass the full range of information that should be recorded. These information structures should permit flexibility in the granularity of encoded information and relationships.
- *Analysis*. The procedure should support the following activities: the preparation of a safety case; the exploration of design options, trade−off analysis between different properties; recording justifications; change management; checking consistency/completeness of requirements.

After further consultations with BAe Topic I customers, a subset of primary concerns were identified.

- *Preparation of a Safety Case*. Relevant aspects of design rationale should be recorded in structures that support the preparation of a safety case.
- *Change Control*. The information structures should support the prediction of the consequences of changing an element of a specification (i.e. assess the impact of a change request).

3.2 Underlying Approach

The crux of the problem, is to provide effective means for recording the information that emerges during system development in structures that enhance traceability. Approaches for recording development information have been most extensively examined in schemes for capturing design rationale (i.e. the information that explicitly records the reasons for making design decisions). Several approaches have been proposed for design rationale;

those based upon: a record of the exploratory design activity (e.g. the Questions Options Criteria /Maclean 91/ approach); specific representational schemes /Chandraeskaran 93/; and information structures to record the design rationale, such as the Design Rationale Capture System (DRCS) /Klein 93/.

The features of DRCS that make it the prime candidate are: provision of a rich set of information structures that are related to system engineering, and a generic model that can be customised for particular development environments. The DRCS approach consists of a rationale language, this provides a vocabulary of assertions which consists of entities (modules, tasks, specification, versions etc.), and claims about these entities, relational claims which also have a pre−defined vocabulary. The DRCS language is divided into five components, each can be viewed as a self−contained information structure focusing on a particular aspect of the development information.

- *Synthesis*. Records actions used to define artifacts and plans for their development. The basic entities for an artifact description include modules, attributes, interfaces and connections. The attributes can have values expressed using a constraint language (including ranges, inequalities etc). The basic entities for the plan description are subtasks and actions.
- *Evaluation*. Records the results of evaluations between different design versions of a specification.
- *Intent*. Records the association of a decision problem to an assertion (e.g. how a particular specification will be achieved) and relate this to a strategy to address the problem. For a given type of assertion a predefined decision problem is provided.
- *Versions*. Records the exploration of the spaces of design alternatives for a decision problem.
- *Argumentation.* Records the reasons for or against believing a claim.

3.3 Customisation of DRCS Approach

The customisation of the DRCS approach is guided by the development models (see 2.4) and the characteristics of dependable avionic systems. To establish a relationship between the DRCS and the top−level information model, the contents of the information model are partitioned into two groups: *Products & Procedures*, and *Rationale*. Products & Procedures encompass the information of the basic product or procedure entities (e.g. requirements, strategies, designs, constraints, test description procedures etc.) and related stakeholders. Rationale encompasses the information of basic supporting entities (e.g. decision rationale, design rationale and justifications) and related stakeholders. These two partitions can be used to derive macro−structures from the basic structures of the DRCS and localize where enhancements are needed. A top−level relationship between the information model and DRCS structures is shown in Figure 4.

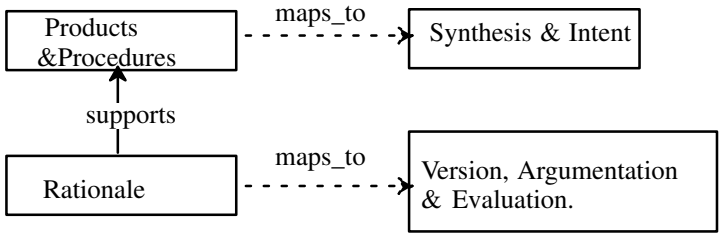

Figure 4: Relationship Between Information Model and DRCS Structures

The traceability procedure will support concurrent engineering, by enabling the inclusion of information at varying degrees of commitment (e.g. an artifact model can represent an actual susbsystem or the specification of a subsystem).

In addition to the specific issues for the two partitions of the information model, there are two generic issues relating to the organization of the information structures. Firstly to manage relationships between entities of a single structure, the notion of grouping entities into "families" which can be related to another entity (or family) is introduced; guidelines have been identified to ensure this feature is applied in a disciplined way (an example of this is illustrated in Figure 9). Further, to better organize the overall information captured using the customised DRCS structures, the application of the procedure is steered towards the identification of layered models, each layer representing a level of abstraction in a decompostion of the overall system (see Figure 5).

3.3.1 Products & Procedures Structure

The products and procedures information is *elicited* and *expressed* by construction of the synthesis and intent structures. The customisation process involves modifications in both the artifacts and plans aspects of the synthesis structure and strengthening the relationship between the intent structure and rationale.

Synthesis: Artifact

The basic DRCS structure for an artifact is extended in three ways to reflect the characteristics of the application domain. Firstly a set of avionic specific data types are introduced for the attributes of an artifact. Secondly, the notion of different projections (functional and behavioural) of an artifact that represent a design is introduced. Thirdly, a new component is introduced: the behavioural structure, which also has some additional relational claims: the *has−mode*, *has−state* relationships, the state claims can have *has−transition* relationships (where a an event and follow on state are defined) and the modes can have *has−function* and *has−transition*. We are currently investigating further components that can be related to notations used in the detailed design of avionic systems, e.g. models for a functional specification.

Synthesis: Plans

The basic DRCS structure for a plan is extended by introducing additional relations /Gotel 95/, including: role relationships, mood of transcription, and extending relationships between artifacts (see Figure 8). The plans structure is closely related to the process model, the tasks (subtasks) corresponding to selected activities of the process model, and can be supported by configuration management tools.

Intent

The assertions of the DRCS model reflect an intent of the developer, the introduction of more specific types facilitates the introduction of more targeted decision problems. In addition, to accommodate the increased emphasis on evidence, the convention is adopted that each assertion should be associated with some part of the rationale structure; either through an exploration of possible options or justification for the stated assertion.

3.3.2. Rationale Structure

The rationale information is *elicited* and *expressed* by construction of the version, argumentation and evaluation structures; and an additional structure the responsibility model.

Responsibility Model

The responsibility model adds to the design reasoning model the stakeholder (this can be an individual or a review team) who asserted the claim (resulting in the choice of options) and the role which they played in this particular development at that time. This not only indicates the validity of claims, but also records their capacity in making the claim and records the people involved if future reference is required.

3.4. Traceability Procedure and Information Model

Application of the traceability procedure will lead to a collection of information structures focusing on different aspects of system development (see Figures 6−17), and a composite structure relating the different information structures (see Figure 5). For the contents of the structures to be more manageable, mappings from the entities and relationships of the information structure to the top−level information model have been identified.

4. Case Study

The system discussed in this paper is an Attitude Monitor, this case study was a subsystem on the Experimental Aircraft Programme (EAP). The study has been used previously to demonstrate several complementary development techniques proposed by the DCSC (Dependable Computing Systems Centre), each focussing on different aspects of the system, including safety properties, functionality and structure see /Coombes 95/.

The Attitude Monitor is a component of the Gust Alleviation Filter which was required to stop or reduce cross−winds from adversely affecting the performance of the aircraft. The

filter required as input the acceleration of the aircraft with the component due to gravity eliminated, this is provided by the Attitude Monitor.

Building upon earlier work performed by the DCSC concerning traceability, this case study was utilised to examine issues related to a traceability procedure by the re−expression of the Attitude Monitor following BAe's proposed process improvements.

4.1 Conducting the Case Study

The examples in this section are taken from the case study to demonstrate the support provided for traceability. The work performed represents a post mortem style investigation identifying the information from the original project documentation and from the documentation produced as the result of earlier research. This process highlighted gaps in the information originally recorded with respect to the information required to support the traceability procedure.

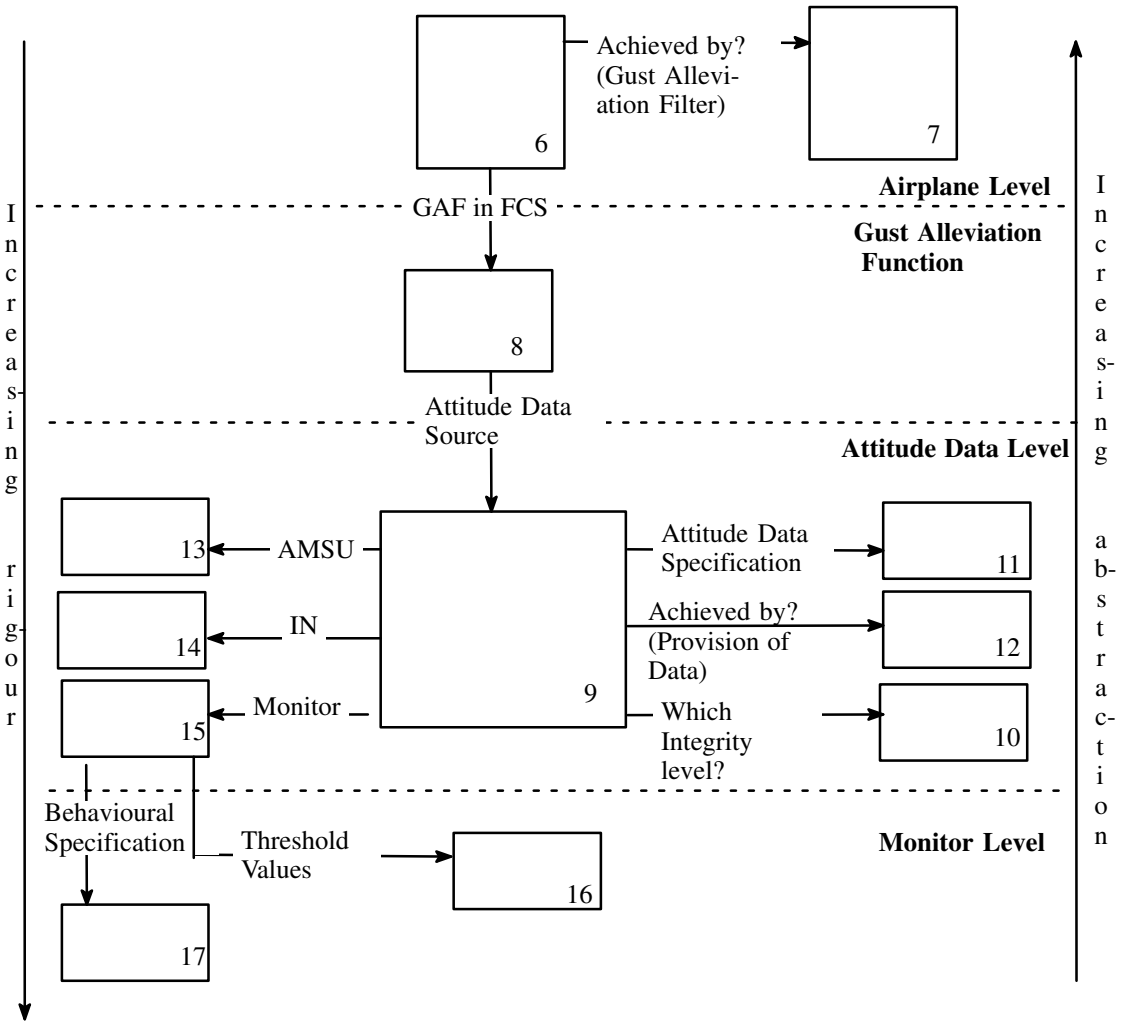

Figure 5: Composite Structure

Route Map

A composite structure is provided, see Figure 5, to help guide the reader through the figures (depicted by the indices in the boxes) and their links, it also reflects a four level model. The top level represents the Airplane level and is at a high level of abstraction, the second level is the system specification level which in this case contains the Gust Alleviation Function (GAF). The third and fourth levels contain more detailed specifications and represent respectively the Attitude Data Level, which still contains some actual data values on the artifact models, and the Monitor level where the specifications represent solely the required values of the system to be built.

Airplane (Conceptual Level)

The top level view of the system relates to the whole Airplane and only contains those elements which were relevant during the re−expression see Figure 6, therefore we are initially only interested in the `Airplane` attribute of `Wind Tolerance` and the value which it must provide. The artifact synthesis model is used to express this view, the `Airplane` *has−attribute* `Wind Tolerance`, which *has−value* <V, and *has−submodule* `Flight Control System (FCS)` etc. The artifact model is extended and demonstrates the intent model where decision problems are examined (there are decision problem types for every relation defined). The decision problems raised here are how to achieve the desired attribute, this has a strategy to `Develop a Gust Alleviation Filter` and this assertion raises the decision problem of how this is to be achieved.

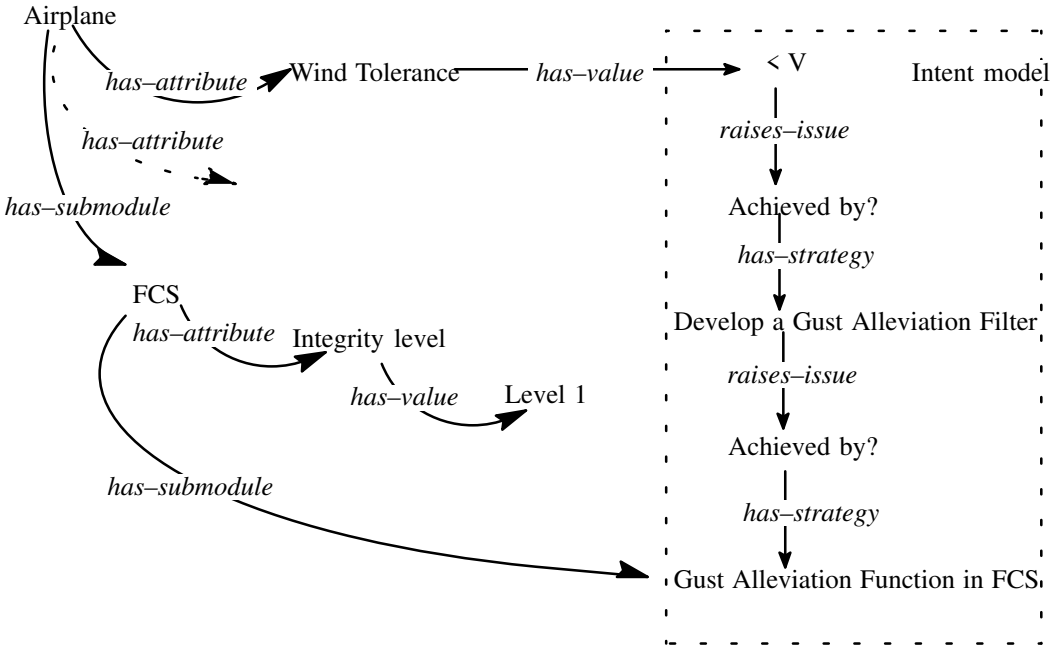

Figure 6: Artifact Synthesis – Airplane Level

This decision problem is more complex and has several options which are explored via the version and argumentation models in Figure 7, before establishing the strategy to provide

a gust alleviation function in the FCS. This strategy is further refined via an Artifact synthesis model, reference Figure 8.

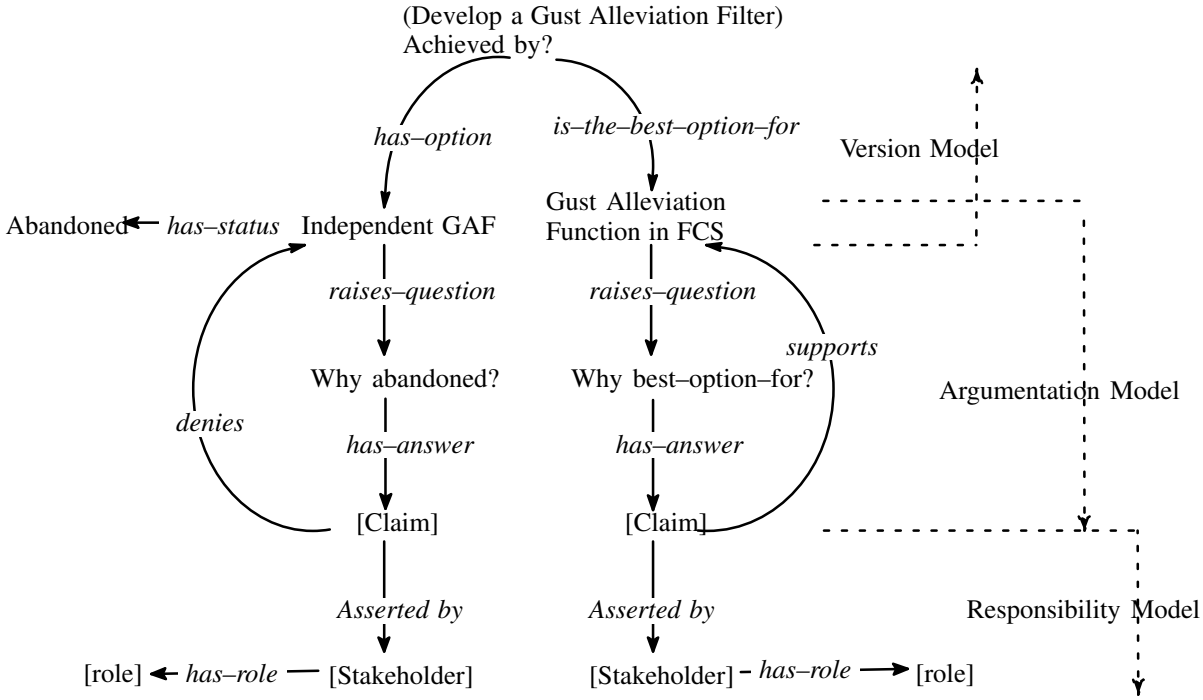

Figure 7: Exploration of Design space and Argumentation structure – Rationale

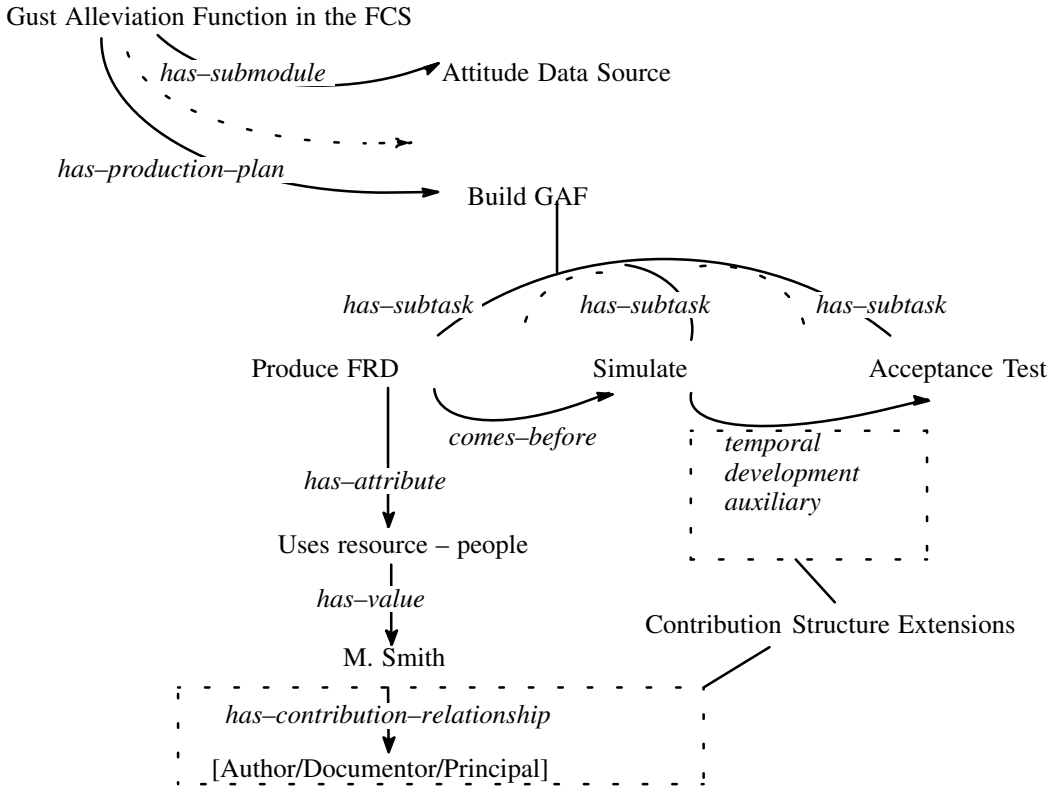

Figure 8: Artifact Synthesis model – GAF in the FCS – extending to Plan synthesis

The version model, in Figure 7, explores the design options for developing a gust alleviation filter (i.e. an `Independent GAF` or a `GAF within the FCS`), which leads into the argumentation model where the question is raised as to which option is best and the answer should be supported by a claim. By applying the DRCS structure we are prompted for supporting design rationale this highlights some of the information not included in the original documentation.

Gust Alleviation Function (System Level)

Moving down a level from that of the aircraft we have the `Gust Alleviation Function` which will be a component of the FCS, reference Figure 6. This decision therefore associates with the system an `integrity level of 1` (high integrity) which is the integrity level of the FCS. The artifact synthesis model in Figure 8 includes an example of the plan synthesis model which has been extended to include notions from contribution structures /Gotel 95/. The plan is to build the GAF which consists of many sub tasks which may be further decomposed into actions (relating to a project plan).

Attitude Data Source (Architectural Level)

The attitude data source specification starts with an artifact synthesis model, reference Figure 9. The requirements of the attitude data source are specified as the values relating to

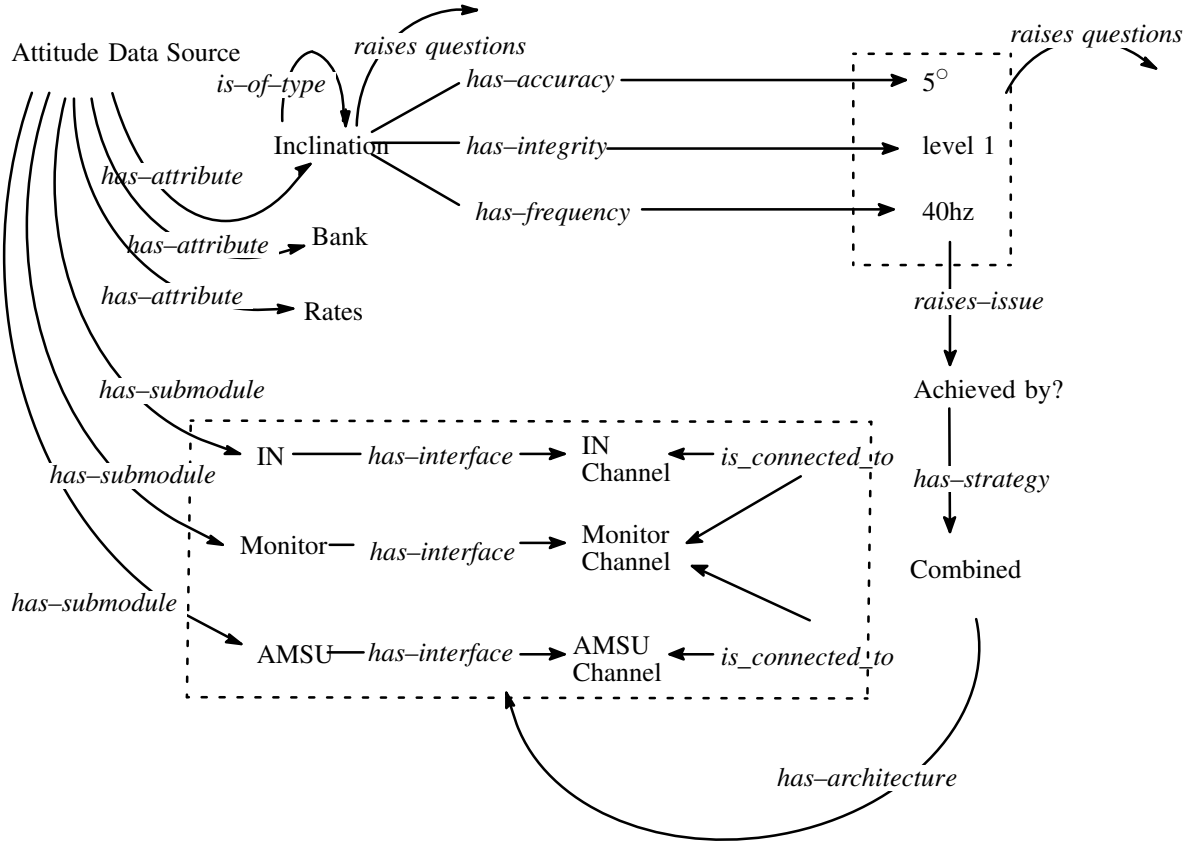

Figure 9: Artifact Synthesis – Attitude Data Source

the attributes. Some of these attributes are of an attribute type relevant to the attitude data,

and this type has associated with it specialized relationships, of the *has−specification* relationship, for attributes of this type values must be established for the *has−accuracy*, *has−integrity* and *has−frequency* relations. The attribute type also raises specific questions relating to this information, for example, which integrity level is to be chosen (expanded in Figure 10) and justification for the choice of desired values e.g. *has−accuracy* < 5 degrees see Figure 11, both sets of questions may be relevant, the first question leads to an exploration of options providing rationale behind the choice, whereas the question raised regarding the choice of desired values only records the justification for the chosen option.

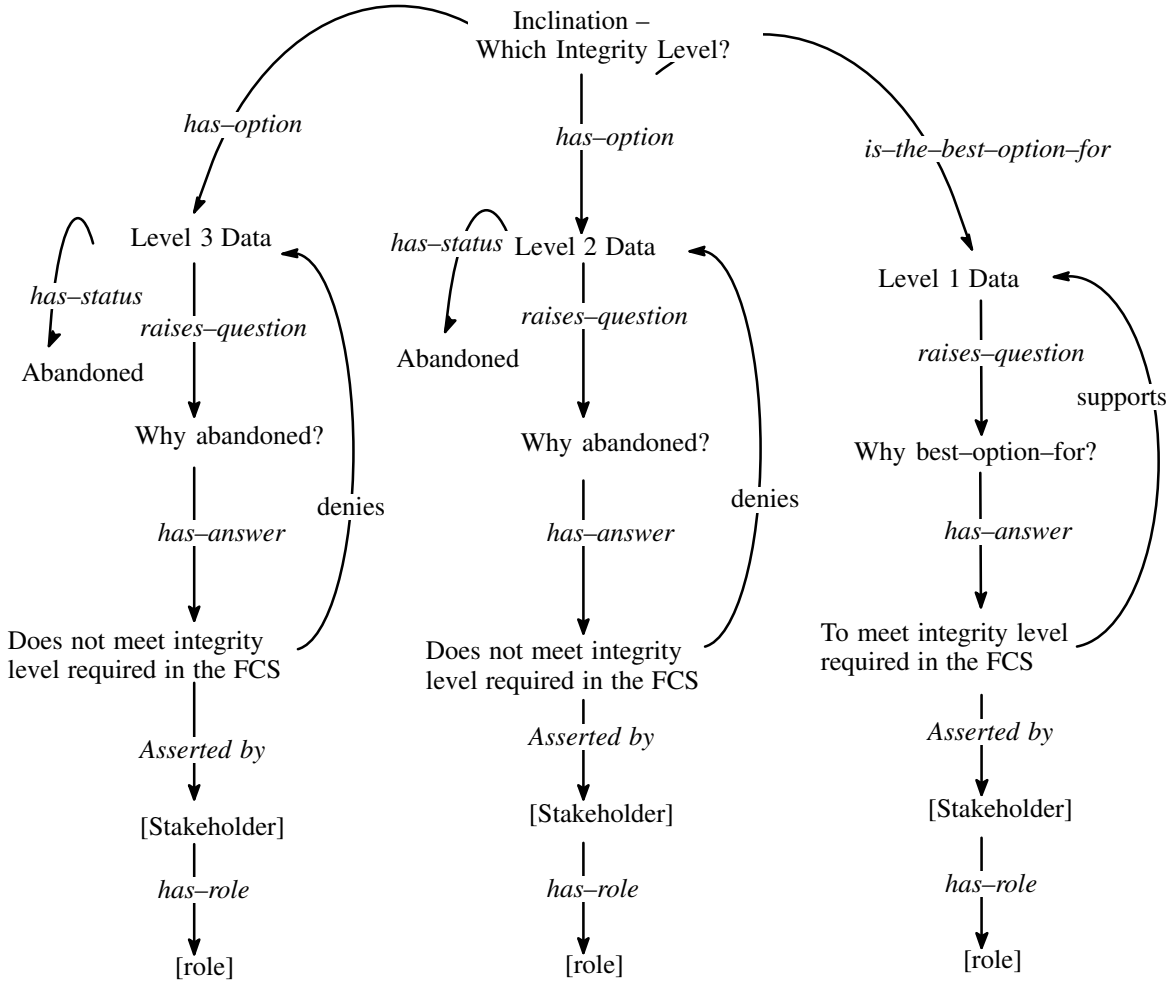

Figure 10: Identification of Attitude Data Integrity Values

The options raised to the query `Attitude Data Source - Which Integrity Level?` are depicted in Figure 10. The design rationale is then recorded through the intent, version, argumentation and responsibility models. The attribute type would have other related queries regarding the degree of accuracy and the frequency required of this data.

For clarity Figure 9 does not detail the complete specification for all the attributes, and several of the attribute values are addressed as a group raising the decision problem of how to achieve the requirements. The concept of being able to group the attributes to address

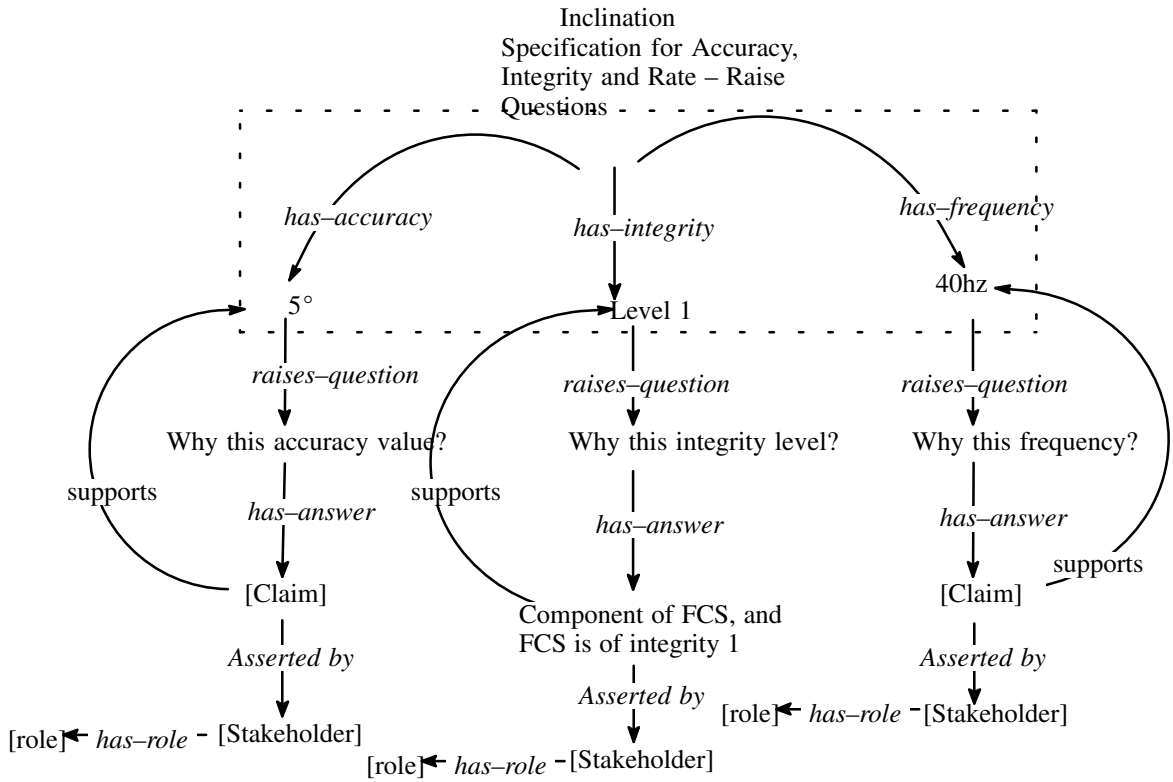

Figure 11: Attitude Data Source Specification

related issues must be carefully managed, in this case we group together the attributes of the same type which raise the same questions and are to be achieved by a single strategy. The strategy chosen to achieve the integrity, accuracy and frequency requirements is to provide a `combined approach` (see Figure 12) which utilises both the Inertial Naviga-

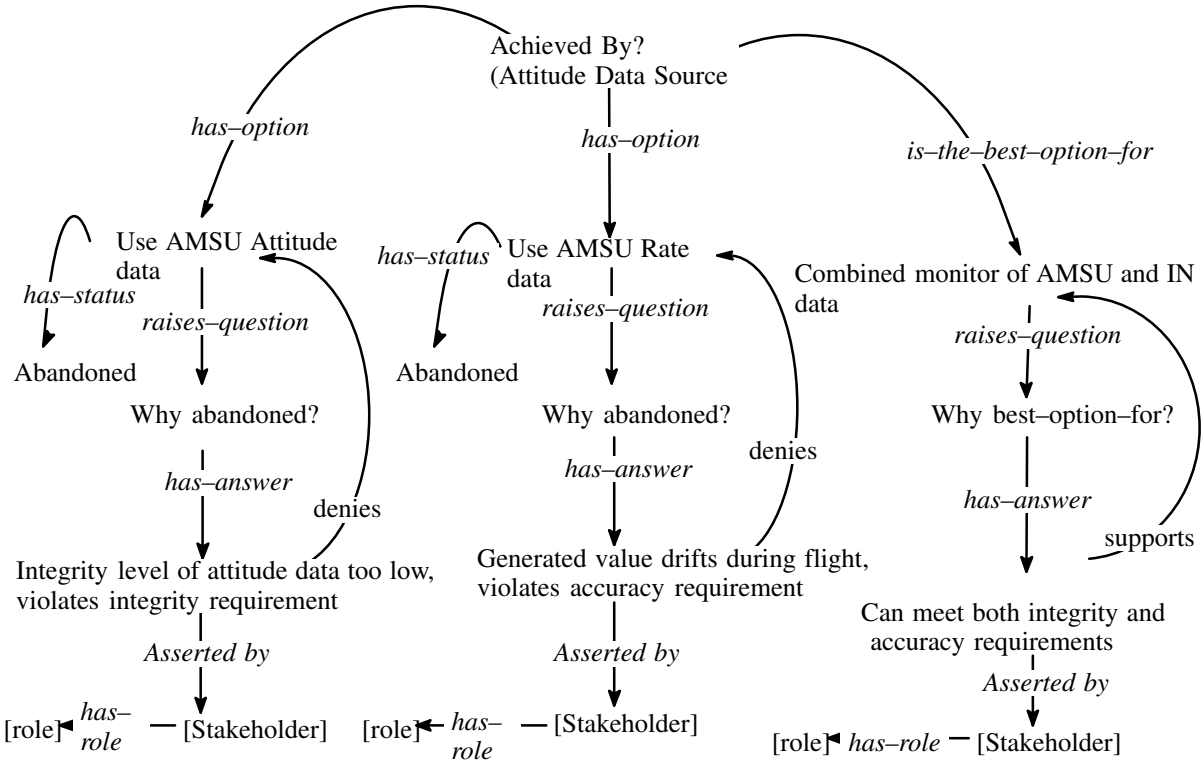

Figure 12: Exploration for Provision of Attitude Data Strategies

tion unit (IN) and the Air Motion Sensor Unit (AMSU), (rate and attitude data), sources to provide the required data. This function is to be provided by the Monitor.

The Attitude Data Source has three sub modules, the IN, AMSU and the Monitor. These modules constitute an architecture for the combined strategy represented by the *has−architecture* relationship which provides the structure of a top level design. The IN and the AMSU are sub modules which were already in existence and the Monitor (to be developed) interfaces with them. The IN and AMSU are specified via artifact models, see Figure 13 for the AMSU model and Figure 14 for the IN model. The Monitor is further refined via an artifact synthesis model.

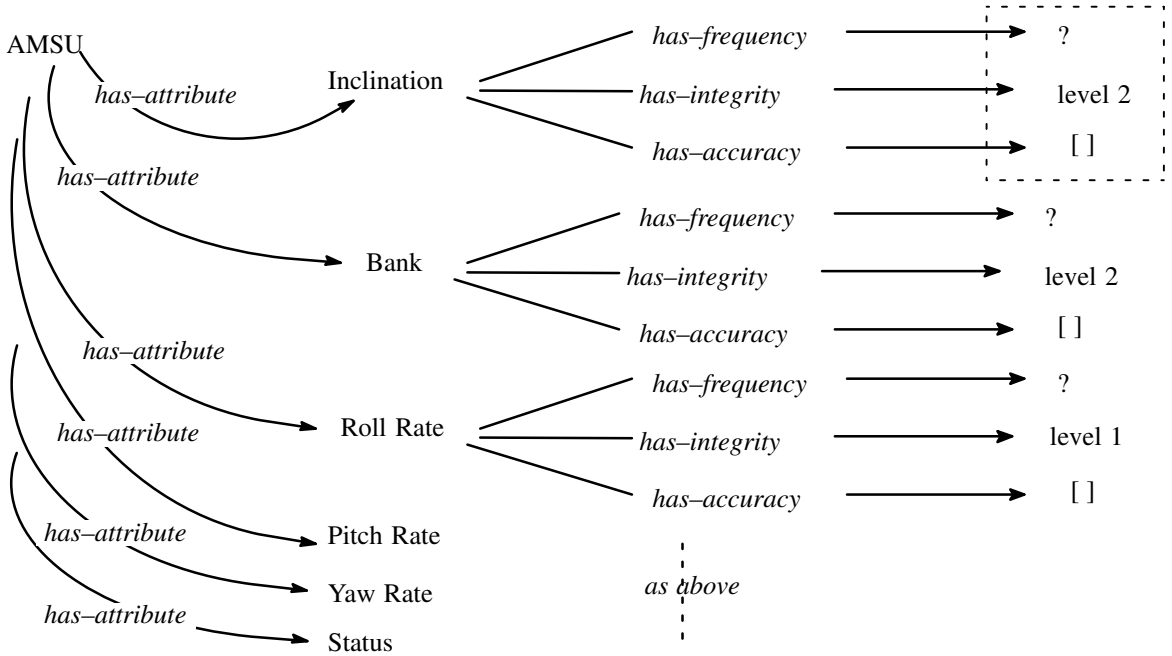

Figure 13: Artifact Model – AMSU

The AMSU artifact description (see Figure 13) *has−attributes* (inclination, bank etc.) which are of the same attribute type as those required for the Attitude Data Source and therefore raise similar questions and require similar specifications. Here, however we record the values to model the components. It is evident from the model why the AMSU alone could not provide the attitude data, i.e. due to the integrity level of 2 for the Inclination and Bank attributes, it is not clear however from this figure why the rates could not be used as they are of the required integrity, the explanation for this is detailed in the argumentation structure relating to how to achieve the provision of attitude data to the required specifications. This represents an artifact description of an existing system module, whereas the Monitor artifact synthesis model represents the model of a planned module

The artifact synthesis model for the Monitor (see Figure 15) includes the attributes which are required to perform its function e.g. Inclination and Bank Thresholds, questions are raised against these relating to how the thresholds will be calculated. The bank

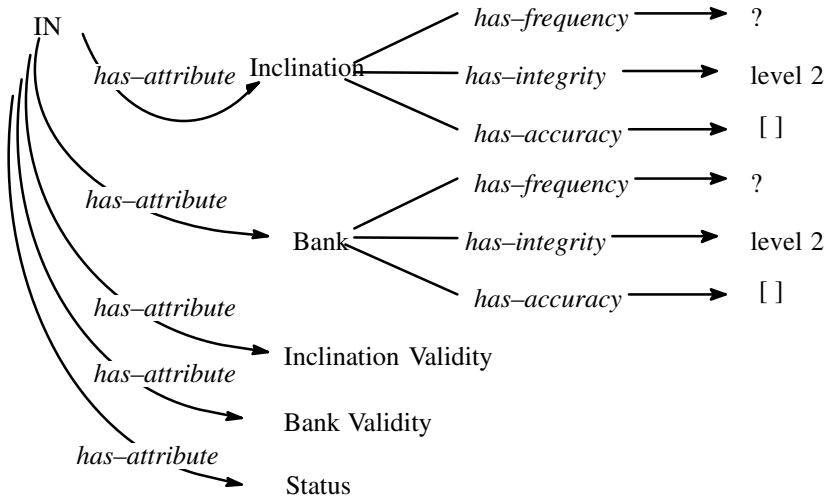

Figure 14: Artifact Model – IN

`threshold` design space is explored in a version, argumentation, and responsibility model as in previous examples see Figure 16. The `Monitor` requires a new relationship type which is specific to the methods of development employed within a domain, this is the *has–projection* relationship. Two types of projection are employed to specify a system model for the monitor, the `Functional Specification` (which consists of the data flow model and algorithm specification) and a `Behavioural Specification` (which is a model of a state transition diagram).

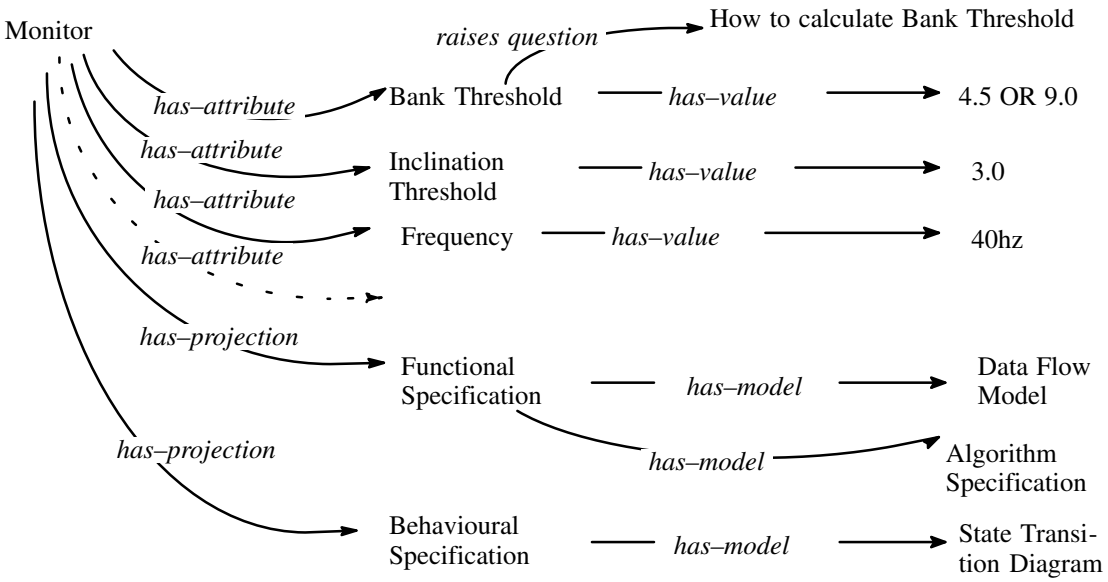

Figure 15: Artifact Synthesis Model – Monitor

Monitor Level (Detailed Design)

The `Monitor State Transition Diagram` is represented via a behavioural model (see Figure 17). We can see from the model that the monitor performs two functions, one when it is in the `Idle` mode (providing raw AMSU data) and the other in the `Generate Attitude Data` mode (providing calculated attitude data). The states represent the

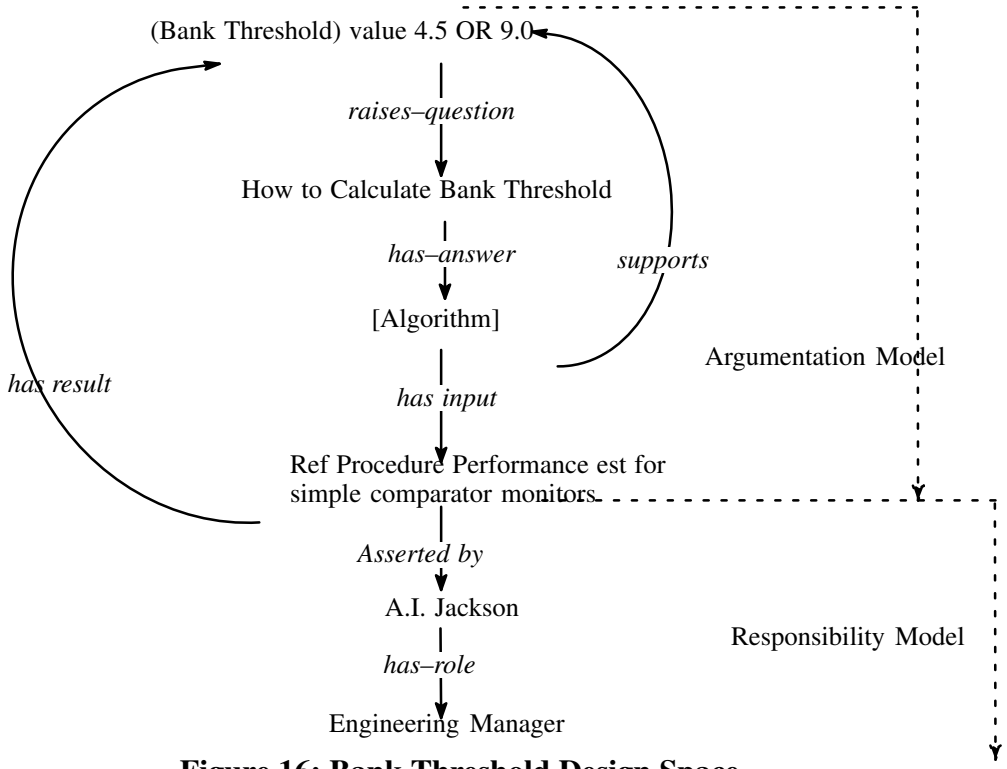

Figure 16: Bank Threshold Design Space

monitoring of the data against thresholds and determine which function will be provided. Questions can be raised against these claims as in the other models, e.g. the `Calculate Generated Attitude Data` claim would raise the question of how to provide this function, the answer in this case would relate to the algorithm specification.

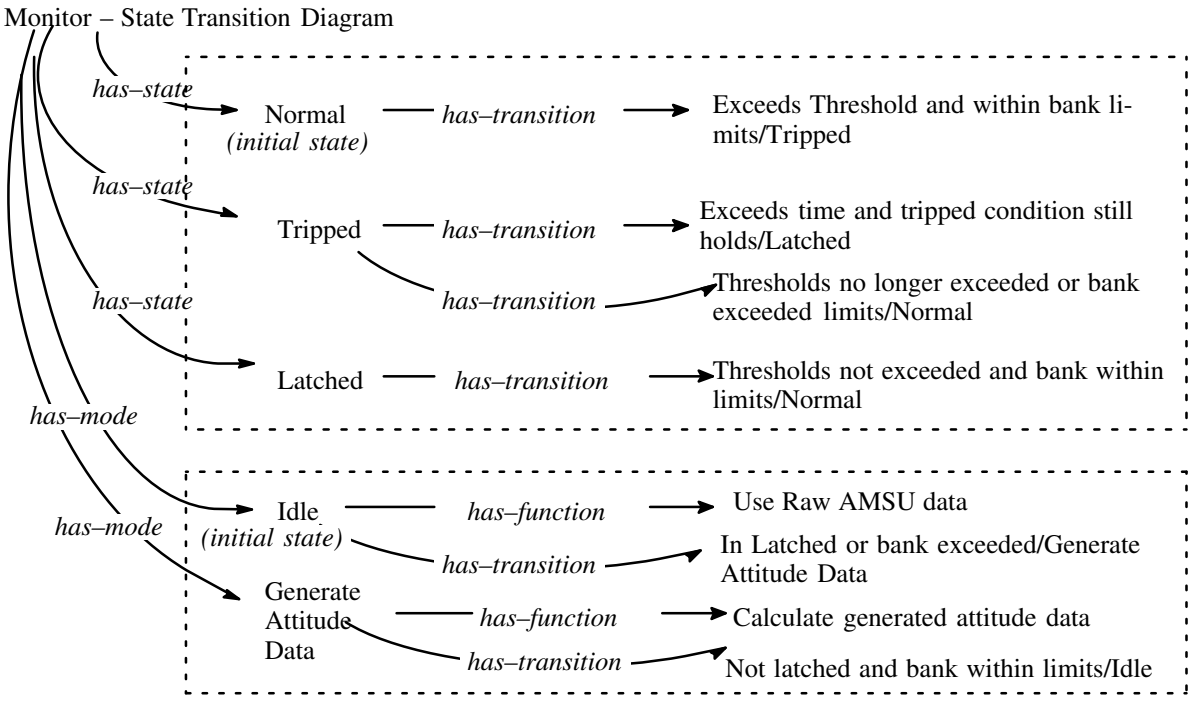

Figure 17: Monitor Behavioural Specification

4.2 Lessons Learnt

The proposed procedure for traceability has been discussed with engineers and initial reactions have been favourable. Indications were that the DRCS was intuitive and could be related to the reasoning process they adopt during system development. However, they stressed that flexibility would be required in the approach and where possible the relevant information should be extracted from structures already employed, utilizing this information to build the model and support the engineer in the development process. Other issues identified are listed below.

- *Overheads inevitable*. The extra effort involved in recording the additional information required to support traceability is considerable, especially for the first project, where user training, introduction of tools to support traceability etc. is required. A loss leader strategy can be followed /Ramesh 95/, by selecting a project which can accommodate the overheads.
- *Benefits*. Unlike many environments/domains the benefits of traceability may be seen during the development of dependable avionics systems. This is due to the domain specific characteristics which typically involve a very long project life cycle, often ten years or more. This introduces many of the problems which traceability helps to overcome into the scope of a single project as opposed to post project e.g. changing requirements, staff movement, evolutionary development, hardware and software upgrades.
- *Decisions in context.* It is very important that the decisions made are recorded in context with the information which may have influenced the decision to permit understanding of the reasoning and that it is recorded in an easily retrievable format. This is particularly important when multi−disciplinary teams are involved.
- *Variants*. The structures permit the re−use of design decisions and modules supporting future project variants. To make this more effective project specific issues should be separated from generic issues.

4.3 Examine the effectiveness of the proposed procedure

Desirable characteristics of a traceability process were discussed earlier in the paper and the proposed approach will be examined against these to identify its effectiveness.

Elicitation. Information can be recorded as entities, attributes, assertions etc. associated with some artifact. The identification of this information is via a pre−defined set of relationships e.g. *has−attribute*, *has−submodule*, *has−value* which may be extended to include domain specific information as demonstrated in the case study. Information may be categorised as being of a specific type which may have associated with it a set of questions/issues to be asked or raised thus prompting the developer for additional information where it is appropriate.

Expression. The structure of the artifact, and argumentation models, allows them to be applicable at several levels of the development process, the models only changing when detailed design is represented in a structure which supports the specific techniques employed in a particular domain. This facilitates the recording of the information at different degrees of granularity in the same model (*has−attribute*, *has−submodule*) and between the four levels. Projections of the recorded information can be made to support both the documentation model (see Figure 3) and the safety properties of the system, see section 4.4.

Analysis. Design options are explored in the version model and this may be extended into the argumentation model to support the recording of justifications and into the responsibility model incorporating the stakeholders asserting the claims. The evaluation model details the information relating to how well a strategy meets the specification and may be extended to support trade off analysis. The analysis of the completeness of requirements is supported during the process by the pre−defined set of questions and issues which are raised, these may be domain specific and build upon previous experience.

4.4 Establishing a Safety Projection

To illustrate how the traceability procedure can be used to support a safety case, we examine how the connection between an element of the GAF and the safety properties of the Airplane can be established. The connection is realized by the Safety Specification Graph (SSG), this is a directed graph in which the nodes represent the specifications and the edges the relationships between the specifications /de Lemos 95/. Figure 18 represents a partial SSG indicating the association of a safety requirement with the attitude monitor. The top level accident is the airplane crashes, hazards are then identified which may lead to this accident e.g. the airplane is in a spin. The FCS must therefore be of high integrity to ensure the safe flight of the airplane, components within the FCS are therefore safety critical by association.

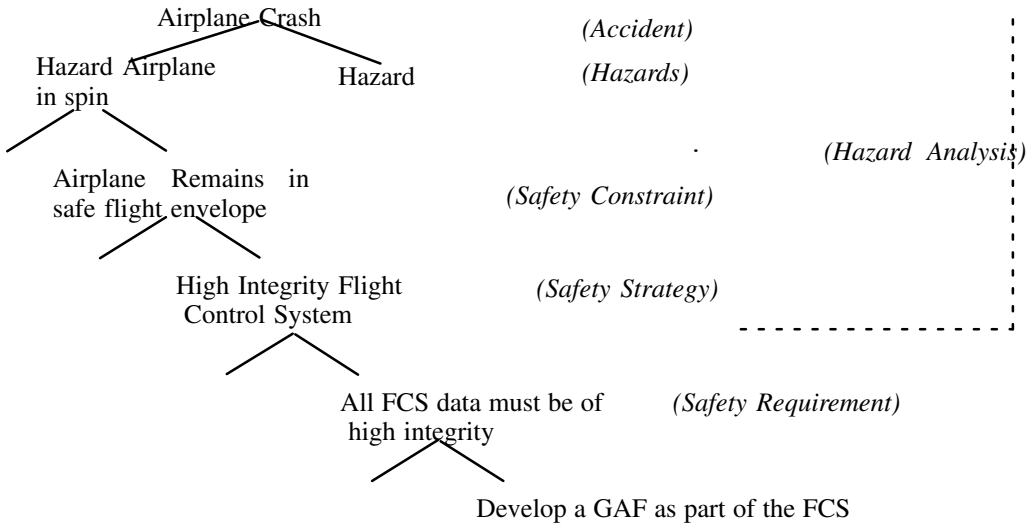

Figure 18: Example Safety Specification Graph (SSG)

The SSG in Figure 19 is a projection of the information structures extracting the information related to safety. Primarily, this information comes from the artifact synthesis and version models. The link between the SSG and DRCS for the attitude monitor is made when the decision to develop the GAF inside the FCS is chosen, Figure 19. The strategies of the graph relate to strategies in the DRCS structures and the requirements relate to the specification of desired attributes. This leads to the requirement on the data to be of integrity level 1, as in Figure 9 of the DRCS supporting the decision depicted in Figure 12. The only feasible option we are left with is to follow the combined approach. The DRCS structure permits the easy extraction and identification of the information required to support the SSG.

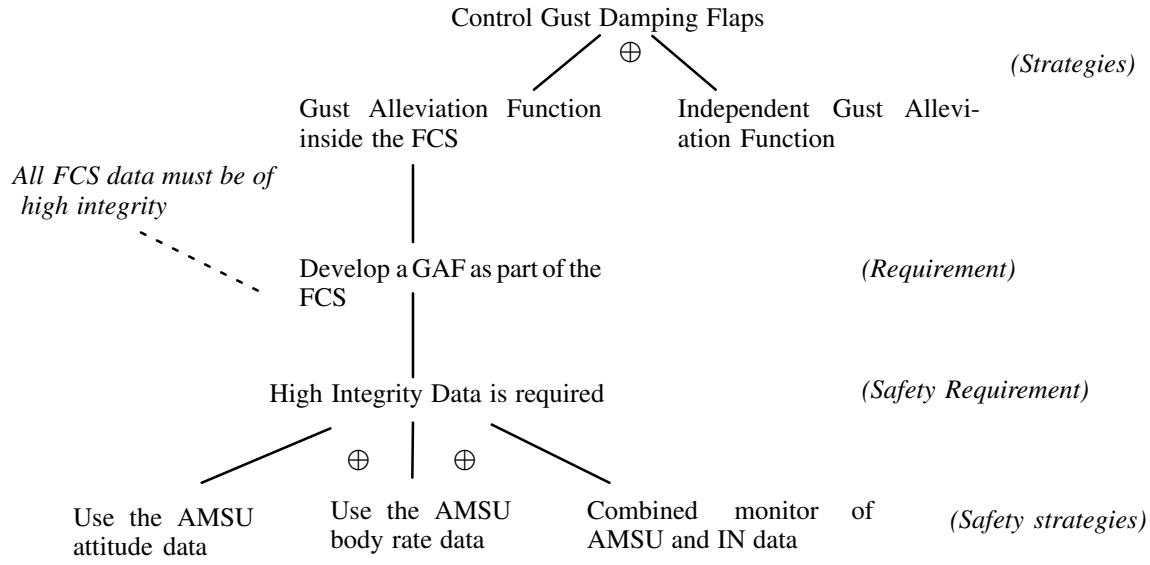

Figure 19: SSG associated with the attitude monitor

5. Tool Support

Integration of the proposed procedure into an organisation's development practices and toolsets is of paramount importance. Problems immediately arise due to a lack of a common data interchange format between most tools. Advances are being made in this area with tool vendors providing more open access to their data formats and by the provision of facilities within tools to write interfaces to access the structures produced by other tools.

A tool exists to support the DRCS process /Klein 93/, but Klein recognised the importance of the capture of this information during the development process and through the tools used to support that process. The planned tool use and support includes tools for: requirements capture, design modelling, configuration control, and the management of traceability; these will be investigated to identify how the traceability information can be captured and expressed in existing tools.

6. Concluding Remarks

Structures for the proposed traceability procedure were defined by: identification of a set of complementary models of a development context; establishing desirable characteristics for a traceability procedure and identification and extension of an approach for the capture of design rationale (DRCS). The models of the development context and characteristics of dependable avionic systems were used to direct the extensions for the DRCS. The feasibility of the approach was examined by applying it to a case study. The data collected by conducting the case study was used to examine the effectiveness of the proposed traceability procedure against the pre−defined characteristics.

In future work, we will focus on identifying further models to support domain specific techniques and notations. It will be necessary to further augment existing structures to accommodate the notion of grouping entities and more domain specific attributes. By extending the application of the traceability procedure to more detailed levels, the opportunity emerges to adopt a more formal approach to support semantic traceability. We also intend to consider the incorporation of other design rationale approaches, for example integration of the QOC approach to the argumentation model. In parallel, the issues involved in the introduction of the traceability procedure into the development practices of an Avionics company will be further examined. This will include issues related to tool support and integration with process improvement initiatives.

Acknowledgements

Earlier discussions with T. Anderson, R. de Lemos, J. Fitzgerald and J. Haveman helped in the development of this work. Support for the case study was provided by BAe Defence (Military Aircraft), in particular M. Johnston. The authors would like to acknowledge the financial support of British Aerospace (DCSC).

References

/Chandrasekaran 93/. B. Chandrasekaran, A.K. Goel, Y. Iwasaki. “Functional Representation as Design Rationale”. *IEEE Computer*. January 1993. pp. 48−56.

/Coombes 95/ A. C. Coombes, L. M. Barroca, J. S. Fitzgerald, J. A. McDermid, A. Saeed and L. Spencer “Formal Specification of an Aerospace System: the Attitude Monitor”, in M. G. Hinchey and J. P. Bowen (eds.) *Applications of Formal Methods*, Prentice−Hall International Series in Computer Science 1995. pp. 307−332.

/de Lemos 95/ R. de Lemos, A. Saeed, T. Anderson. “Analysing Safety Requirements for Process−Control Systems”. *IEEE Software*. May 1995. pp. 42−53.

/Gotel 94/ O.C.Z Gotel & A.C.W Finkelstein. *“An Analysis of the Requirements Traceability Problem”. Proceedings of the First IEEE International Conference on Requirements Engineering*, Colorado Springs pp. 94−101, April 1994.

/Gotel 95/ O.C.Z Gotel & A.C.W Finkelstein. "Contribution Structures". *Proceedings of the Second IEEE International Symposium on Requirements Engineering*, York pp. 100−107, March 1995.

/IEE 91/ IEE Colloquium on "*Tools and Techniques for Maintaining Traceability During Design"* IEE Savoy Place London, Digest No: 1991/180, December 1991.

/Klein 93/ M. Klein. "Capturing Design Rationale in Concurrent Engineering Teams". *IEEE Computer*. January 1993. pp. 39−47.

/MacLean 91/ A. MacLean, R. Young, V.M.E Bellotti and T. Moran. "Questions, options and criteria: Elements of design space analysis. *Human Computer Interaction*. March 1991. pp. 201−250.

/MOD 91/ Ministry of Defence DEF STAN 00−55, *"The Procurement of Safety Critical Software in Defence Equipment"* Interim, April 1991

/Paulk 93/ M. C. Paulk, B. Curtis, M. B. Chrissis, C. V. Weber, "Capability Maturity Model, Version 1.1". *IEEE Software*. July 1993. pp. 18−27.

/Pohl 94/ K. Pohl, S. Jacobs. "Traceability between Cross−Functional−Teams". *1st International Conference on Concurrent Engineering Research and Application*, Pittsburgh, August 1994

/Ramesh 95/ B. Ramesh, T. Powers, C. Stubbs, M. Edwards. "Implementing Requirements Traceability: A Case Study". *Proceedings of the Second IEEE International Symposium on Requirements Engineering*, York pp. 89−95, March 1995

/Strens 94/ M.R. Strens & J.E. Dobson. "Responsibility Modelling as a Technique for Organisational Requirements Definition". *Intelligent Systems Engineering*, Vol 3, No 1, Spring 1994.