Requirements Stability Assessment Using Scenarios

David Bush
UK National Air Traffic Services Ltd
David.Bush@nats.co.uk

Anthony Finkelstein
University College London
Anthony.Finkelstein@cs.ucl.ac.uk

Abstract

This paper presents a new approach to assessing Requirements Stability as a contribution to the building of more stable long lifetime systems. A goal-based model is used to analyze the requirements in a number of possible future worlds described by scenarios of the possible future environment of the system. The result is an assessment of possible instability of the requirements and the assumptions, which can be presented to system developers to inform alternative requirements definition and architectural choices supporting 'targeted design for change'. A process for carrying out the analysis is described and practical tool support outlined. Results from an industrial scale, live case study validating the approach, process and tool support is reported, and possible developments of the concept, process and tool are discussed.

1. Introduction

UK National Air Traffic Services is responsible for providing safe and efficient air traffic management to aircraft flying in UK airspace, and over the Atlantic Ocean. A significant proportion of NATS value is tied up in its infrastructure, which is costly to develop and certify. As a result systems tend to have long lifetimes, and the cost of making changes to those systems is significant. Many of these changes are generated by evolution of the operational, regulatory or technical environment, leading to changed, or new, requirements placed on the system.

Therefore motivation for this work falls strongly in the category of *Adaptive Change* [1] - i.e. change which stems from uncertainty about the future environment in which a system might operate - and the effect this has on the requirements and assumptions [2,3]. Dealing with change as one of the major research challenges facing the requirements engineering community [4]. Identifying and documenting possible future changes is important in order to be able manage the future changes [5] and to make and evaluate architectural choices [6]. This paper presents a new approach to the identification of possible future instability in requirements. A goal-based model is used to describe the requirements in a number of possible future worlds described by scenarios of the possible future environment, and descriptions of possibly instabilities in those worlds can be generated. The approach is distinctive because:

- It is proactive, and takes place within the requirements development activity, rather than reflect its results.
- It provides a 'creative' approach to discovering the possible risks of requirements change.
- It provides an output that developers can use to implement targeted 'design for change'.

This paper is structured as follows: Section 2 describes the existing approaches to dealing with change in requirements. Section 3 presents a 'World-Machine' view of the change environment for a system, and outline an approach based in this view. Section 4 introduces environmental scenarios and goal based requirements as vehicles for carrying out assessments of requirement stability. Section 5 describes a process for identifying, recording and presenting requirement instability assessments, and the tools available to support it. Section 6 presents examples from a case study evaluation of the process and tool. Section 7 discusses the results and identifies future work required.

2. Existing Approaches

In managing change to systems that may emerge in the future considerable emphasis is placed on the system architecture as the key artefact involved [7]. The most established route to handle it is a universal 'design for change' philosophy, where the whole of the architecture/design is conceived and developed such that evolution is possible [8].

From the requirements engineering viewpoint metrics (e.g. [9]), are often used in assessing requirements stability. Unfortunately, this approach is based in the requirements specification and so stems from a historical perspective. They seem to address the question: How much ***have*** the requirements changed as we have documented them (possibly over sequential releases); rather than how much ***might*** the requirements change over the next 10-20 years?

However, it is one thing to accept change as inevitable (which it is), it is another to concentrate exclusively on managing and measuring change when it has occurred. The more proactive approach, described in this paper, involves a greater engagement with what the possible future changes might be in a systems lifetime, and so provide an insight that could allow those possibilities to be specifically *designed for*.

This is a distinct emphasis from both historical work in requirements stability, and from software architecture approaches; a change which will include *a-priori* engagement with the likely stability of the requirements and the assumptions.

Risk Management engages with a similar problem in identifying risks up front. However, while for example, the SEI Risk Taxonomy [10] does recognize requirements stability as a risk area, there is a fundamental assumption that these risks are already known - just not well communicated. Hence, its elicitation techniques are unsuitable for identifying long-term changes in the environment or the requirements - where often the possibility of change is not currently know them. More fundamentally, experience in the strategic management domain [11] suggests that *identifying and reporting* these risks is itself insufficient, and that achieving early action in response to them requires a much more interactive engagement with them and their consequences.

3. A World-Machine View of Evolution

To model the relationship between the system we propose to build, and the environment in which it will sit we have applied Jackson's [12] concept of 'The World and the Machine' to make clear the importance of the environment within which the solution system will eventually sit. In Jackson's formulation the requirements (R) for the system, and the domain knowledge (D) or assumptions about the environment that surround it are 'world' artifacts.

But what is this 'world' or environment? Classically in systems development we might understand it as the world NOW when we analyze a current system, and then extend this to an EXPECTED world for which we develop a set of requirements and assumptions. However, there is considerable uncertainty in the future environment for the machine. We therefore extend Jackson's concept to allow for this uncertainty by conceiving possible future worlds in which our machine might operate.

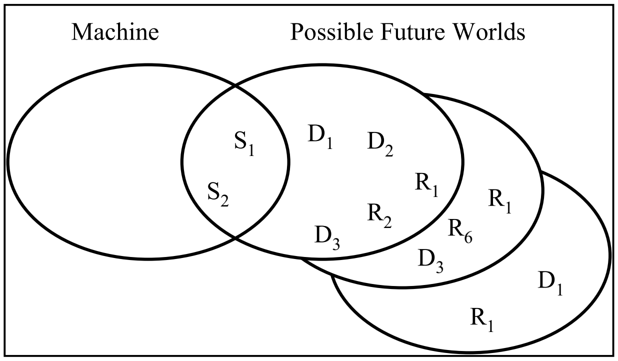


Figure 1 - Alternative Future Worlds

Each of these can have its own requirements (R_n) and domain knowledge (D_n), and it may share some of these with other possible future worlds. As we cannot predict which of these worlds might eventually come about, then to minimize the impact of change in the systems lifetime we need to build a system with the best chance of being stable in all of them (or as many as possible). However two other considerations face developers: they must design for the expected world - even if some of it's requirements and assumptions are not present in the possible future worlds; and secondly they must develop cost-effective solutions - and this is hard enough when developing for just one expected world.

Therefore in order to provide a feasible approach we follow a process that is based in the current practice - of starting with the requirements for a system for the expected world. We then move on to analyzing those requirements in the possible future worlds ('**A**'). Elements of stability and volatility in requirements and domain knowledge can then be identified, and the developer is then in a position to decide what, if anything, they wish to do about the specification for the machine in order to address that change risk ('**B**').

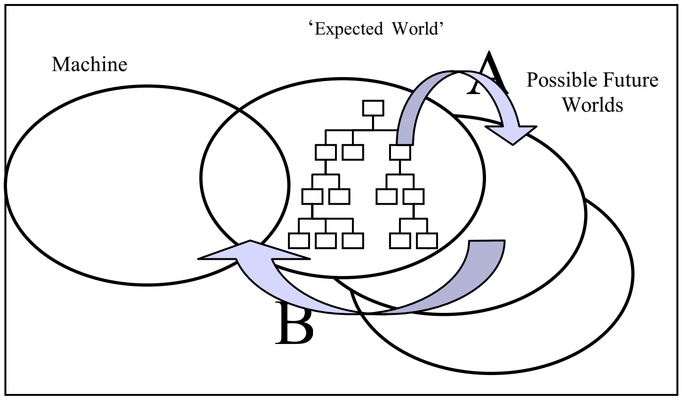


Figure 2 - Analysis and Response

4. Environmental Scenarios and Goal Models

In [13] we introduced the use of environmental scenarios from the strategic management domain to describe possible future worlds with a view to assessing requirements stability of requirements represented in a goal based model.

This paper presents the follow on of that study, defining a process for evaluating stability, tool support to assist the developer; and provides illustrations of the output of the process as artifacts for developers to review their requirements or architectures to address the instabilities identified.

We selected scenarios because of the close match between their use in business, and the extended 'World/Machine' view of alternative worlds described in Section 2. Secondly they were attractive because of the established strengths they have in anticipating and understanding risks and identifying new scope and opportunities [14]. They effectively represent a global 'WHAT-IF' environmental analysis.

We selected a goal model as our representation mechanism because they are well established as a mechanism for dealing with change, both requirements change [15,15a, 15b] and safety case argumentation [16], and for the established work in identifying classifications of change types in goal structures [17]

There are many different types of goal based representations, [18,19,20,21] In order to keep our approach as open as possible to different goal modeling approaches we have limited the hard requirements and constraints placed on the content and structure of the goal graph. These are the essential characteristics for use in this approach:

- A parent goal is satisfied when its child goals are adequately satisfied.
- Goals should include clear information expressing why their refinements are adequate - (we have called this 'refinement argument'.) It will also provide an indication of the level of confidence we have in the refinement being adequate.
- Goals should contain a notion of the measurable extent to which they must be performed. (So to 'Achieve Safety' is not enough - the level of 'Safety' will need to be defined). This aspect is important because it is often in the detail of the 'level' of performance required in a goal that instabilities occur.
- The goal graph should explicitly include assumptions as a form of goal (often these can be 'allocated' to the environment) linked as necessary to ensure adequate refinement.

5. The Stability Assessment Process

5.1 General

This section defines an assessment process to be followed in establishing the stability of the set of goals modeling the system requirements. There are three distinct and sequential stages of the process: the Preparation; the Assessment; and the Resolution. The preparation stage involves the development of the environmental scenarios and the baseline goals for the system. In the Assessment phase the stability of the goals are judged and recorded. In the Resolution stage the results of the stability assessments are used to make decisions about changes to the requirements, and/or to the architecture of the solution.

There is an important distinction to be drawn between Assessment and Resolution. Assessment activities are based in the existing goal details, and the activity makes assessments of those goals, it does not change them, or add further goals. It is the Resolution stage that makes changes to the actual goals.

The emphasis of this work is in assessing the stability of the goals, and so the process description is most detailed in that area. In the case study examination, some detail will be provided about the results of the Preparation stage, and pointers will be provided to the possible content of the Resolution stage, although these are not core aspects of this work.

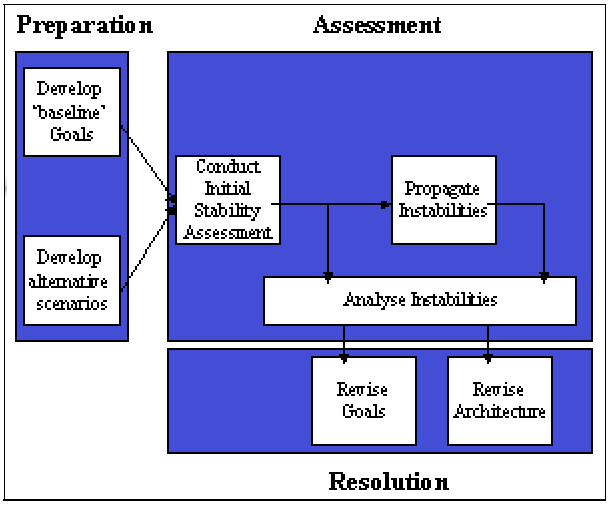


Figure 3 - Stability Assessment Process

5.2 Preparation

The preparation stage contains the activities necessary to develop the pre-requisites for the detailed goal based assessment. These pre-requisites are shown on the left hand side of Figure 3. The pre-requisites are: a set of requirements and assumptions structured in a goal directed form; and a number of possible future worlds described in scenarios related to the context of the system being developed. While the development of goal-based models of requirements may be familiar to readers [16, 18,19] the development of scenarios of possible future worlds may not be. It is beyond the scope of this paper to address this, but established texts in this area provide considerable guidance in this activity. [21a,21b]

5.3 Assessment

Given the two pre-requisites: a goal graph and a set of scenarios, the detailed Assessment stage can begin. There are three activities involved in the Assessment stage, these are: an Initial Stability Assessment, where a goal-by-goal assessment of stability is made in each environmental scenario; an Analysis activity, where the results of the stability assessment can be reviewed – so allowing decision on resolution action; and a Propagation activity, which overcomes the 'localization' inherent in the initial piecemeal assessment. The preferred route through these activities is:

Initial Stability Assessment → Analyze Instabilities → Instability Propagation → Analyze Instabilities

Where the first of the analysis stages can be used to make a decision about whether it is necessary or worthwhile to carry out the propagation activity. The following sub-sections describe the process following this recommended path.

5.3.1 Initial Stability Assessment

The initial assessment is a piecemeal passthrough of each item in the goal graph, against each of the scenario descriptions in turn. This involves reading into the scenario description, and then imagining the system in the context of the 'alternative' world that the scenario describes. Then for each goal, an assessment can be made of the stability, both in a numerical score, and in a textual description.

The classification of possible change types in a goal based requirements structure was introduced in [17]. Although there the context was for supporting the derivation of rationale for version changes of the requirements, we have found the classification a useful basis for extension into a classification of goal instabilities. Such a classification is useful because:

- It assists users of the process to think about the categories of instability that may occur.
- It provides a standard terminology which assist in the collection of data and metrics.
- It serves as the basis for providing propagation heuristics (discussed later).
- It may provide a mechanism for establishing 'Resolution Heuristics'.

Our extended set of goal and assumption instability types are listed in Table 1.

- Goal Addition
- Goal Removal
- Change Goal Functionality
- Change Goal Level
- Add Goal Refinement
- Remove Refinement
- Assumption Addition
- Change Assumption Level
- Assumption Removal
- Change Allocation

Table 1 - Goal Change Types

At the completion of this activity the goals in the goal graph will each be annotated with instability measures and remarks – for each of the environmental scenarios considered. This initial assessment can then be analyzed.

5.3.2 Analyze Instabilities

The analysis of the initially assessment can be used to identify the Goals most likely to be unstable - either overall across all scenarios, or within in each scenario. At this stage the result of the analysis might be either:

- A decision to use the initial assessment as the basis for deciding on resolution action, or
- A decision to carry out a propagation of the identified instabilities, in order to address the 'localization' limitation of the piecemeal assessment.

The description of this Propagation activity, and the rationale for it is addressed next.

5.3.3 Propagate Instabilities

While this first pass assessment is useful, and can provide a basis for identifying resolution actions, it suffers from one particular limitation, and that is that it ignores the non-locality of instability in the goals. The most trivial example of such a non-locality would be 'Goal Removal – where if we assessed a particular goal to be unstable in that way – then all of the leaf goals[*] should really be unstable in the same way. This process therefore includes a stage based on the strategy used in goal based safety Cases [16] - a second pass stage, where goal instabilities can be propagated up, down and across the goal graph, to provide a more thorough and considered assessment of possible instability.

In order to provide guidance to those assessing goal instability, we have identified a series of propagation heuristics to assist in the consideration of possible propagation actions. Each heuristic is based on a particular type of goal instability (introduced in Table 1), and includes guide questions and considerations for propagating *up*, *down* or *across* the goal graph. An example of one of these heuristics is shown in Table 2 below.

Propagation Heuristic for: Goal Addition

Trace Up: Identify any parent goal, which is itself inadequately broad in scope to include the need to have this Goal as a refinement. That Goal is a candidate for instability, and has the instability type Change Goal Functionality or Change Goal Level.

Trace Down: Identify any child goal that is itself inadequately broad in scope to include the need to have this Goal as a parent. That Goal is a candidate for instability type Change Goal Functionality or Change Goal Level.

Table 2 - Sample Propagation Heuristics

Once the propagation is complete, a revised set of values and comments about possible instability in each environmental scenario is available, allowing a second pass of the analysis activity.

5.3.4 Analyze Instabilities (2nd Pass)

With the revised set of assessments the assessors once again can review the assessments. These potential instabilities identified can then inform decisions about whether or not to resolve the issues identified – possibly by changing the requirements, or by making architectural design decisions.

6. Tool Support to the Process

In order to confirm the feasibility of the concept and process, and to provide a degree of automation and support to the case study, a supporting toolset was developed to guide users through the process and to record and present the results of the analysis. In order to minimize the cost and effort, and to maximize the possible take-up of the tool support, the development strategy was based on using existing desktop tools, adapted and integrated as needed.

[*] More strictly 'all of the leaf goals that do not have another parent'.

The tool had to meet two distinct needs, firstly the presentation of the goals in a graph structure, and secondly the recording and processing of the analysis data as it is entered. For the graphical element Microsoft Visio© has provided a widely available tool to support goal graph development and presentation. For the data storage and manipulation, a database is the most appropriate mechanism. Given he relatively simple requirements we selected Microsoft Access©. As these represent widely available 'standard' tools, and can be easily integrated with Visual Basic©, they met the needs admirably. Related work we have been involved with has identified that the integration of Visio with ODBC database is relatively straightforward, and recently we have been developing and applying a Visio to DOORS link, which would provide for more rigorous requirements for data storage, management and tracability.

7. Instability Assessment: A Case Study

7.1 Case Study System

In order to validate the concept, process and tool support we conducted a case study evaluation of the stability of the requirements for a real-life NATS problem - the Minimum Safe Altitude Warning (MSAW) system. The MSAW system is intended to reduce the number of occurrences of Controlled Flight into Terrain (CFIT) through a ground based early warning to aircraft flying below the mandated minimum safe altitude. This particular system was selected because we had adopted a goal based approach to the development of the requirements for the system as part of an earlier evaluation of goal based requirements engineering [23] and so not only was the goal graph already available, but we were also familiar with the requirements, and confident in its completeness and accuracy.

7.2 Case Study Scenarios

Full details of the scenarios and the techniques used in their derivation are reported in [24]. These scenarios were developed by a small team of MBA students, with significant pre-course experience in air transport and business. Each scenario provides a narrative description of a possible future world, and an example of one of the scenarios is visible in Figure 5. The four scenarios described worlds we have characterised as: Affluent Nationalism; Consumerism; Controlled Development and Survival of the Fittest. Figure 4 shows the highlights of these scenarios.

7.3 Case Study Examples

In this section examples are presented, based on the results developed during the case study. The sub-sections are arranged based on the assessment activities, Initial Assessment, Analysis and Propagation.

7.3.1 Initial Stability Assessment

An example screenshot of the tool used for stability assessment is shown in Figure 5. The figure shows, at the top, a full description of the text of the scenario. Alternative page selectors on the tab strip allow the use to view the key drivers of the scenario in list form or a bulleted list of change drivers for requirements in general.

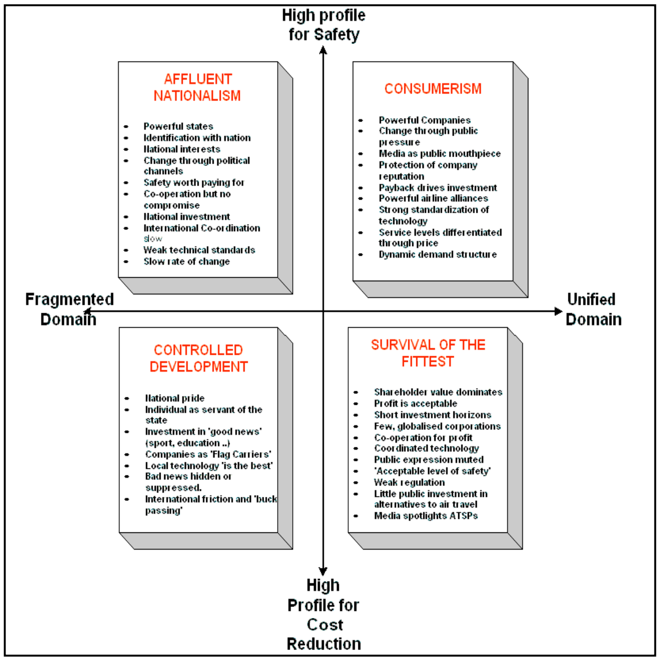

High profile for Safety
AFFLUENT NATIONALISM
Powerful states
Identification with nation
National interests
Change through political channels
Safety worth paying for
Co-operation but no compromise
National investment
International Co-ordination slow
Weak technical standards
Slow rate of change
CONSUMERISM
Powerful Companies
Change through public pressure
Media as public mouthpiece
Protection of company reputation
Payback drives investment
Powerful airline alliances
Strong standardization of technology
Service levels differentiated through price
Dynamic demand structure
Fragmented Domain
Unified Domain
CONTROLLED DEVELOPMENT
National pride
Individual as servant of the state
Investment in 'good news' (sport, education ..)
Companies as 'Flag Carriers'
Local technology 'is the best'
Bad news hidden or suppressed.
International friction and 'buck passing'
SURVIVAL OF THE FITTEST
Shareholder value dominates
Profit is acceptable
Short investment horizons
Few, globalised corporations
Co-operation for profit
Coordinated technology
Public expression muted
'Acceptable level of safety'
Weak regulation
Little public investment in alternatives to air travel
Media spotlights ATSPs
High Profile for Cost Reduction

Figure 4 Scenario Summaries

A diagram of the local extract of the goal graph is presented and the user can make an assessment of the instability in the box in the bottom left, by providing a checkbox assessment, a description of the instability effect and a classification of the type of the instability. For the case study a 4-point scale was selected, with the responses converted to a linear 0-3 scale. An example of the kind of instability recorded at this stage is shown in Table 3, this represents a single assessment of a goal against a scenario.

Goal	***Scenario***
1	4

Title

Achieve: Controller Alerted to proximity

Description

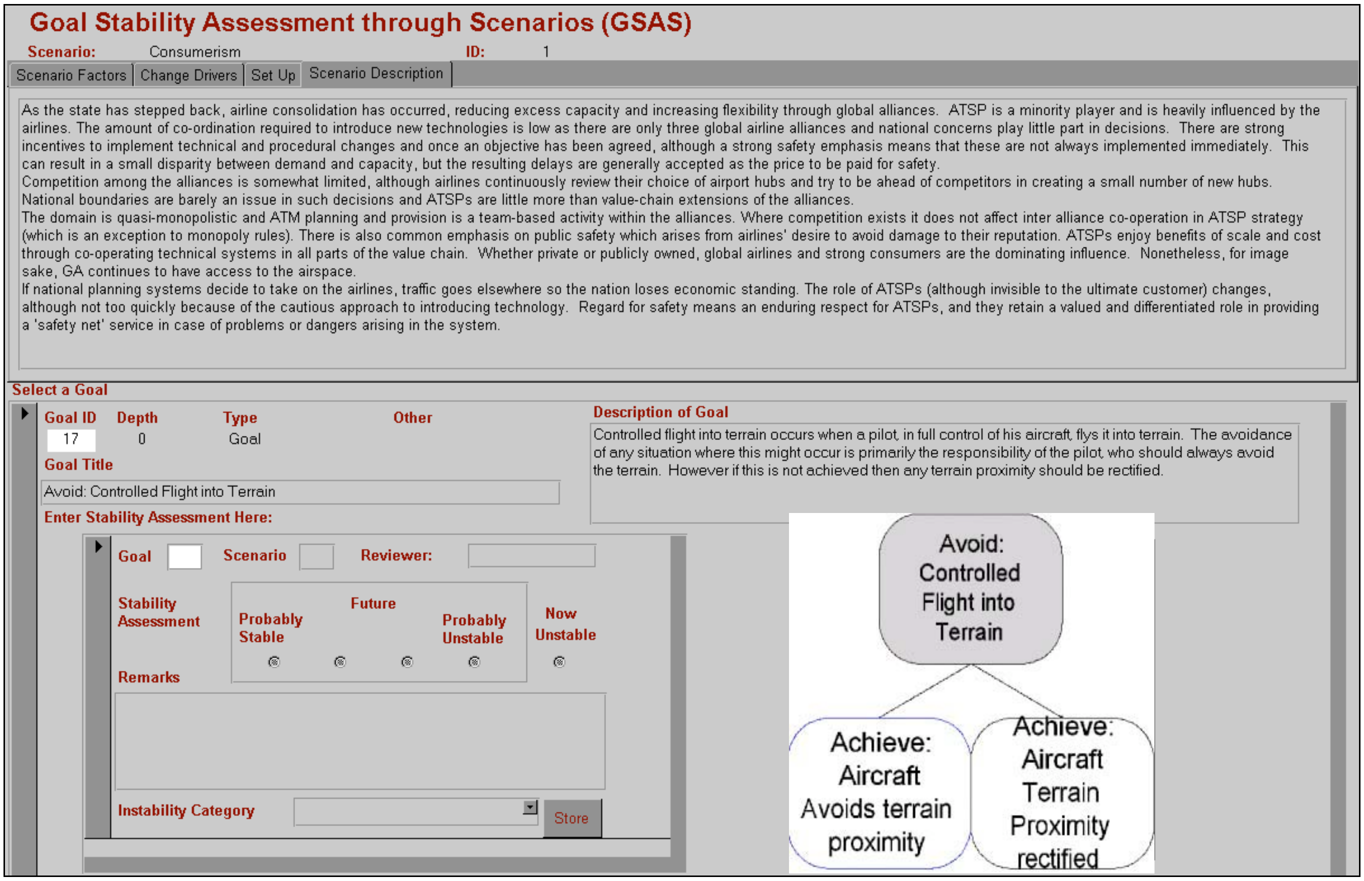
The controller is alerted to the proximity of an aircraft to the ground. This is currently possible by identifying that an aircraft has hit the ground and/or by predicting that it will hit, and then informing the controller.

Assessment	***Instability Category***
2	Goal Functionality Changes

Remarks

Likely that this would be removed altogether and replaced with direct alerting.

Table 3 - Example Assessment



Goal Stability Assessment through Scenarios (GSAS)

Scenario: Consumerism ID: 1

As the state has stepped back, airline consolidation has occurred, reducing excess capacity and increasing flexibility through global alliances. ATSP is a minority player and is heavily influenced by the airlines. The amount of co-ordination required to introduce new technologies is low as there are only three global airline alliances and national concerns play little part in decisions. There are strong incentives to implement technical and procedural changes and once an objective has been agreed, although a strong safety emphasis means that these are not always implemented immediately. This can result in a small disparity between demand and capacity, but the resulting delays are generally accepted as the price to be paid for safety.
Competition among the alliances is somewhat limited, although airlines continuously review their choice of airport hubs and try to be ahead of competitors in creating a small number of new hubs.
National boundaries are barely an issue in such decisions and ATSPs are little more than value-chain extensions of the alliances.
The domain is quasi-monopolistic and ATM planning and provision is a team-based activity within the alliances. Where competition exists it does not affect inter alliance co-operation in ATSP strategy (which is an exception to monopoly rules). There is also common emphasis on public safety which arises from airlines' desire to avoid damage to their reputation. ATSPs enjoy benefits of scale and cost through co-operating technical systems in all parts of the value chain. Whether private or publicly owned, global airlines and strong consumers are the dominating influence. Nonetheless, for image sake, GA continues to have access to the airspace.
If national planning systems decide to take on the airlines, traffic goes elsewhere so the nation loses economic standing. The role of ATSPs (although invisible to the ultimate customer) changes, although not too quickly because of the cautious approach to introducing technology. Regard for safety means an enduring respect for ATSPs, and they retain a valued and differentiated role in providing a 'safety net' service in case of problems or dangers arising in the system.

Select a Goal

Goal ID	Depth	Type	Other
17	0	Goal	

Goal Title: Avoid: Controlled Flight into Terrain

Description of Goal: Controlled flight into terrain occurs when a pilot, in full control of his aircraft, flys it into terrain. The avoidance of any situation where this might occur is primarily the responsibility of the pilot, who should always avoid the terrain. However if this is not achieved then any terrain proximity should be rectified.

Enter Stability Assessment Here:

Goal Scenario Reviewer:

Stability Assessment: Probably Stable ○ ○ Future ○ Probably Unstable ○ Now Unstable ○

Remarks

Instability Category

Store
Avoid: Controlled Flight into Terrain
Achieve: Aircraft Avoids terrain proximity
Achieve: Aircraft Terrain Proximity rectified

Figure 5 - Initial Stability Assessment

7.3.2 Initial Instability Analysis

Once assessments were complete for all goals in each of the four scenarios the initial assessment was complete. Figure 6 is an overview of the goal graph, and while the detail is now readable, it does show the results of the instability scores reflected back into the goals in colour.

The goal graph is colour coded using the instability ratings given. Lighter colours represent little or no stability, while the darker colours are the more unstable goals/assumptions. The coding scheme selected in the case study ensured that the severity of the instability was a function both of the total instability scores across all scenarios, and of the number of different scenarios in which a particular goal/assumption turned out to be unstable graph. A particular strength of this display is that it allows a rapid targeting of the most unstable goals and assumptions, allowing users to target follow up activities on the more critical areas. The main weakness is that it does not provide any detail on the nature of the instability and how it varies across scenarios. Indeed, as some instability may actually be welcome a different view is needed to carry out a detailed analysis of the instabilities.

The tool provided such an alternative view, allowing the drilling into the detail of the instability in a way that supported the making of decisions about what might be done about the identified instability. The goal detail view for one of the goals in the case study is shown in Figure 8. This shows the assessments for each of the four scenarios on one page, and allows the user to make the necessary decisions about resolving the instability. Such a resolution might involve changing goals; making architectural decisions about the solution, which might mitigate the risks; or, indeed, doing nothing.

The process description highlighted the potential limitation of the single pass assessment, that it ignored the cascading effects of change around the goal graph. The goal detail display shown in Figure 7 allows the user to begin to address this by propagating instabilities around the goal graph, because against each scenario is a connection to the propagation tool.

7.3.3 Instability Propagation

Instability propagation required a source Goals and at least one target goal. The propagation heuristics described in Section 6 are used to identify which are the candidate target goals. Figure 8 shows the tool display for one of the case study source goals that has been selected for instability propagation. The display presents the source goal information and the recorded instability types for it. Based on these the lower box contains the specific propagation heuristics for the given instability type.

Using the heuristics as a guide the goal graph was scanned, and a suitable target goal to which the instability could be propagated was identified. The target goal was then edited to provide revised, *propagated*, assessment. Once the goals were propagated in this way, an alternative 'propagated' version of the goal graph was generated. The Analysis process was then able to be can then be repeated for the (potentially new) unstable goals to make the appropriate decisions about resolution actions.

7.4 Using Instability Results

The process and tool discussed here specifically excludes the mechanism by which the instability results will be used by analysts or designers to change the requirements or to make architectural decisions. This is an approach and process to assess and present stability/instability measures for requirements. Our vision is that the developers, having worked through this process will be able to use the displays already presented to make their resolution decisions. The following examples from the case study illustrate the way in which this approach has provided new information, which can be used to drive further requirements or design activities.

7.4.1 Goal Addition Example

The subject goal is "**Avoid: Controlled Flight into Terrain". ** During *Initial Assessment*, against the scenario it was identified that this top level goal was potentially unstable because that scenario suggested that MSAW alerting might become a 'subscription service'. This suggested that there might be an additional goal to bill the aircraft for the use of the service.

During *Instability Propagation*, it became clear that the impact of such an instability also spread to the goal: **Achieve: Controller Notifies Pilot,** as pilots of airlines who had not subscribed to the service would not need to be informed. In fact, after consideration, it was decided that this propagated instability was probably best assigned to the child goal: **Achieve: Controller Informed of Proximity**, as there was no point warning a controller if no action was going to be taken.

During *Instability Analysis*, the two potential instabilities (the possible new Goal, and the possible change to the goal to 'inform the controller'), prompts thought on a number of possible resolutions actions. Firstly, the new goal could be added (although it seems likely that it would be allocated outside of the MSAW system, to a supporting administrative process), or the potential new requirement could be ignored. Secondly, the potential change in the goal associated with informing the controller could be ignored for now or alternatively the possibility could be engaged with, in which case the following considerations might be made:

- How should the goals be changed to address the possible change? (should the controller be informed or not if a non-subscribing aircraft is in danger?)

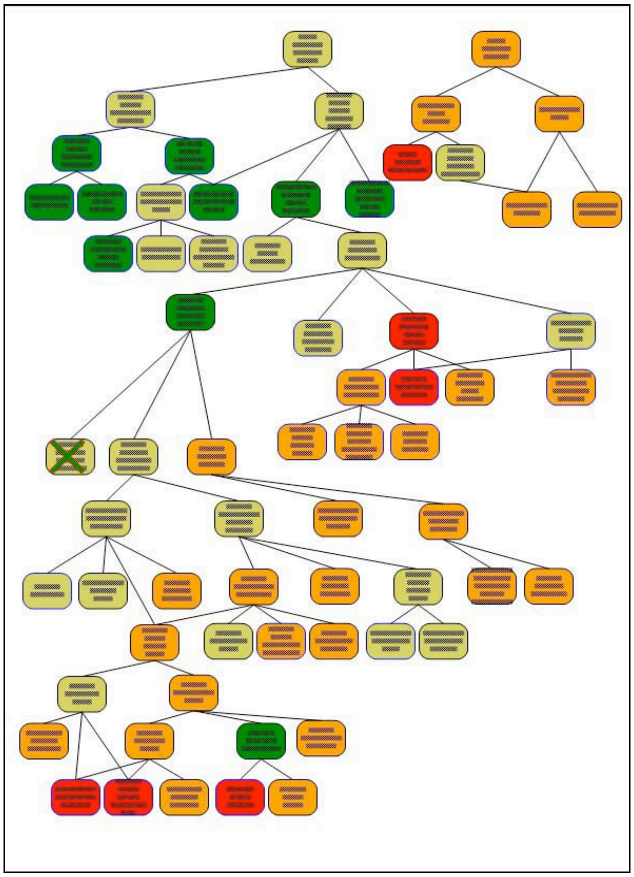


Figure 6 - MSAW Goal Graph - Extract

- What additional requirements are implied to meet this possible new need (e.g. Inhibition of alerts for non-subscribing Aircraft)?

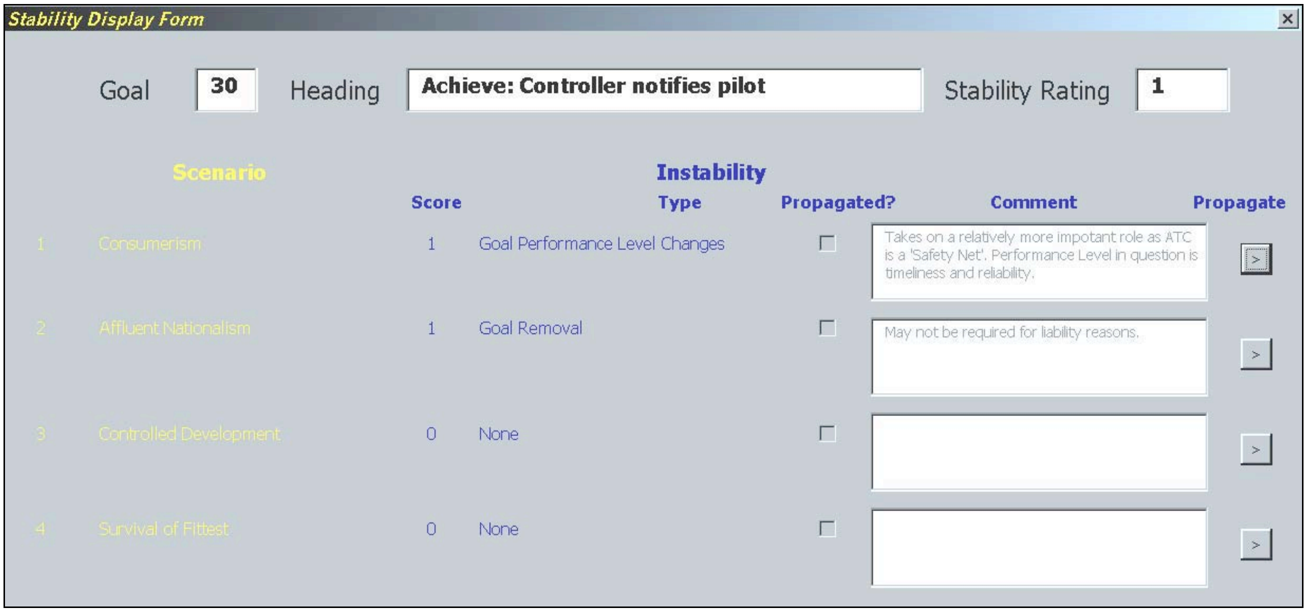
  How should the solution be addressed:

  - With the 'inhibition' functionality built in and all aircraft automatically subscribed unless and until the possible change materializes?
  - With the 'alerting' functionality 'designed for changed' to accommodate inhibition capabilities some time in the future if needed?

7.5 Analysis of the Case Study

Overall the case study was very successful at providing insights to the possible instabilities that might emerge in the longer term. A number of possible new and changed requirements, and assumptions were identified. These were genuinely new, as they had not been identified in the original goal derivation exercise. Furthermore, the use of the textual commenting of the assessors' reasons for marking items as unstable was useful, because when coupled with the database trace to scenarios it provided significant rationale for the choices made.

Notwithstanding this existing trace, we are of the opinion that the stability assessment exercise is most likely to be successful if carried out by those who will need to act on its results – the requirements analysts, system architects and designers.

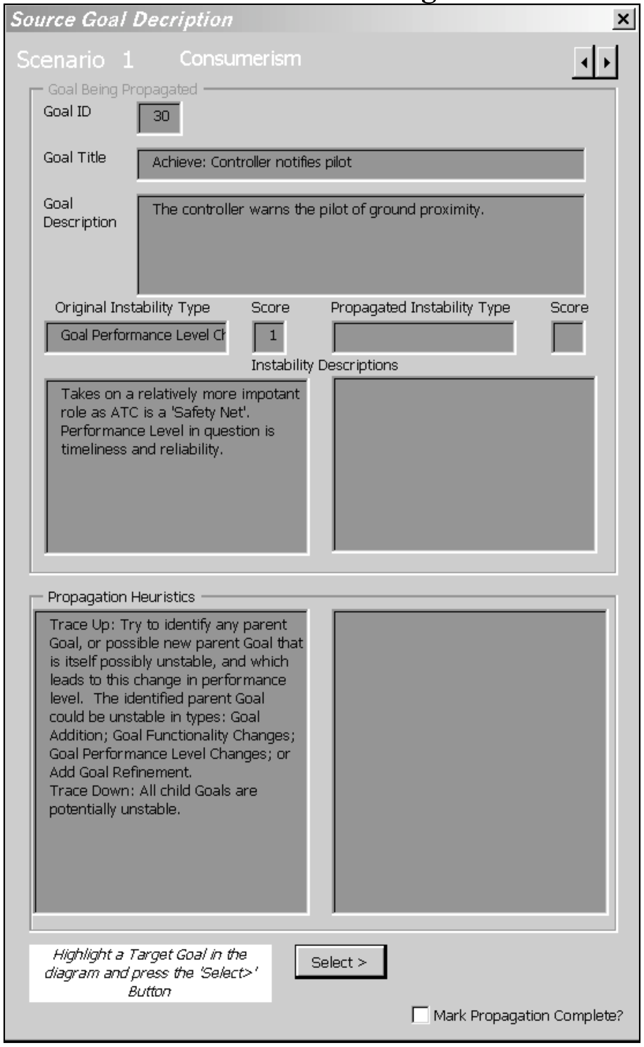


Stability Display Form

Goal 30 Heading Achieve: Controller notifies pilot Stability Rating 1

Scenario		Score	Instability Type	Propagated?	Comment	Propagate
1	Consumerism	1	Goal Performance Level Changes	□	Takes on a relatively more impotant role as ATC is a 'Safety Net'. Performance Level in question is timeliness and reliability.	>
2	Affluent Nationalism	1	Goal Removal	□	May not be required for liability reasons.	>
3	Controlled Development	0	None	□		>
4	Survival of Fittest	0	None	□		>

Figure 7 - Instability Report for a Goal



Source Goal Decription

Scenario 1 Consumerism

Goal Being Propagated

Goal ID: 30

Goal Title: Achieve: Controller notifies pilot

Goal Description: The controller warns the pilot of ground proximity.

Original Instability Type	Score	Propagated Instability Type	Score
Goal Performance Level C	1		

Instability Descriptions

Takes on a relatively more impotant role as ATC is a 'Safety Net'. Performance Level in question is timeliness and reliability.

Propagation Heuristics

Trace Up: Try to identify any parent Goal, or possible new parent Goal that is itself possibly unstable, and which leads to this change in performance level. The identified parent Goal could be unstable in types: Goal Addition; Goal Functionality Changes; Goal Performance Level Changes; or Add Goal Refinement.
Trace Down: All child Goals are potentially unstable.

Highlight a Target Goal in the diagram and press the 'Select>' Button

□ Mark Propagation Complete?

Figure 8 - Propagation Heuristics

There are a number of reasons why this is likely:

- It is valuable that those with responsibility for making decisions about resolving possible instabilities feel that hey own the instabilities identified.
- Views on the nature of the possible future worlds will change as time progresses. If the people working on the system have internalized the system-to-worlds relationships they are more likely to identify risks associated with emerging new environments. This effect has been called 'Memory of the Future' [25].
- When reviewing or resolving potential instabilities recourse to rationale traces will be minimized as those involved will retain memory of their assessments.

In the case study we wanted to provide the maximum choice about how to assess stability. The tool incorporated three possible mechanisms: a taxonomy of change risks; the 'key drivers' used to develop the scenarios; and the scenarios themselves. It rapidly emerged that the scenario approach was the most easy to use, and the most intuitive. The preference was so pronounced that only the scenarios were really used in the end.

The initial assessment process could be completed relatively quickly, once the scenario had become relatively well internalized in the consciousness. For large goal graphs, the scope could be limited to higher-level goals, which would have most effect on the system if they changed, although this would risk losing valuable information in the process.

The propagation activity was quite time-consuming, and sometimes one pass of propagation was not sufficient. For example, propagation of a particular goal might have been completed, but would have to be re-visited if that goal were itself the target for propagation from another goal. Usually this was not too much of a problem for parent-child propagations, but when the effect was a peer-to-peer propagation it sometimes re-opened up the whole propagation problem for the target goal. Nonetheless, propagation was effective in a number of ways:

- It acted as a review mechanism for assessments. In all we removed 5 assessments at this stage, as when we came to propagate them we were not satisfied that they actually represented an instability.
- Inconsistencies in scoring the instabilities were identified and were able to be rectified when carrying out propagation.
- Fairly subtle peer-to-peer propagations were identified only through the propagation activity.

However, it must be said that only the third of these could be classed as a significant contribution to the quality of the assessments overall. In the first case it is likely that these incorrect instability assessments would very likely have been identified at the resolution stage anyway. In the second case the score inconsistencies were due to a natural tendency to propagate instabilities anyway in the initial assessment stage. In other words, a particular instability of the type 'Goal Removal' was often cascaded down the goal graph anyway on the initial assessment.

During propagation it was tempting to slip away from pure 'assessment' and move into 'resolution'. This was particularly noticeable with the instability types 'Goal Addition', 'Assumption Addition' and 'Add Goal Refinement'. Each of these – even in its name – is suggesting a resolution to a problem – making a change to the goal model. One option was to remove these as instability types, but on reflection we decided that they should stay because:

- Sometimes they had propagation effects that suggested instability in a different goal.
- The addition of new functionality was a legitimate reason for change, and so could not be ignored.

Having made this decision it emphasized the difficulty of deciding which goal should have the (for example 'Goal Addition') instability attached to it – when the original goal was not really unstable. We adopted an entirely practical approach to this, and attached the instability to the original goal that made us think about the new goal that might be needed!

Overall the contribution made by the propagation stage, bearing in mind the effort involved, was questionable. The conclusion to draw therefore is that the instability analysis stage preceding propagation need to take a very critical view on the need for the propagation activity to take place at all.

The original tool was limited to allowing one instability assessment per goal per scenario. While this generally served acceptably for initial assessment, it was not adequate when propagating. Overall it would probably be worthwhile removing this limitation.

As the case study represented a well-defined description of the requirements, there was little remaining 'alternative' refinements to work on. Applying this approach to less well-defined requirements, or to emerging requirements, where significant 'OR' type requirements remain, could result in some improvements to the propagation heuristics. Intuitively, the use of this approach could provide requirements analysts help in making selections between alternative refinements – to chose those representing the most stable choice overall.

As both the speed and effect of changing requirements and assumptions varies significantly in different domains and applications, we would be interested to carry out further examination and validation off this approach in a different setting. The mobile telephony domain is appealing as it has a much quicker change cycle that the domain studied; it is more competitive – and so changing requirements and assumptions are a business advantage to be seized, rather than a cost to be resisted; and there is a strong embedded concept of product families and family architectures. Ericsson developed business scenarios for this domain over a 10-year horizon, although these may now be a little dated.

8. Summary

This paper has presented a new approach to assessing the stability of requirements for long-lifetime systems. By comparing a goal-based model of the requirements, and scenario descriptions of possible future worlds, an assessment can be made of possibly instabilities in the requirements.

- The advantages of this approach are that it encourages imaginative engagement with possible futures in a way that existing risk-based approaches do not.
- It provides a predictive view of requirements stability in a way that existing historic measures of requirements change do not.
- It engages with but the requirements and assumptions about the world, and how both of these might cause change in the future.

A robust process for the assessment has been defined, and using easily available tool support this was successfully exercised in a significant industrial case study. Some examples of the case study results have been presented and an example provided of the way in which these results would be of practical use to developers.

Although the focus of this work was in assessing requirements stability, we believe that there is significant scope for future work, not least in the following areas:

- Given the appeal of goal-based approaches in product line requirements, can a scenario based approach to assessing the potential for these requirements to change be used to guide product development paths?

- Can greater use be made of this approach as an aid to 'creativity' in identifying and exploring alternative goal refinements?
- Can a set of 'resolution' heuristics be developed to aid developers in making decisions about resolving identified instabilities?

9. Acknowledgements

I would like to acknowledge the significant contribution made by those in NATS and UCL who collaborated in the development of the Goals for MSAW; and of my fellow students at London Business School who collaborated in the development of the Scenarios.

10. Disclaimer

The Scenarios described here were developed as part of a Group Project at London Business School in May/Jun 2001. None is necessarily NATS' view, either of the likely or desirable future for Air Traffic Services. Indeed all of the opinions expressed in this paper are those of the author, and not necessarily NATS.

11. References

[1] IEEE Standard 610.12 "Glossary of software engineering terminology," in Software Engineering Standards Collection, IEEE CS Press, Los Alamitos, Calif. 1993.

[2] Lehman, M., "Programs, Life Cycles and Laws of Software Evolution", Proc. IEEE Special Issue on Software Engineering, pp. 1060-1076, Sep 1980.

[3] Lubars, Potts & Richter, "A Review of the State of the Practice in Requirements Modeling", Proc. IEEE International Symposium on Requirements Engineering, 1993.

[4] Finkelstein, A. & Kramer, J., "Future of Software Engineering", in *Future of Software Engineering* ed. Finkelstein, A., ACM Press, 2000.

[5] Lehman, M., "The Future of Software - Managing Evolution", IEEE Software, Jan. 1998.

[6] Nuseibeh, B., & Easterbrook, S., "Requirements Engineering: A Roadmap", in *Future of Software Engineering* ed. Finkelstein, A., ACM Press, 2000.

[7] Garlan, D., "Software Architecture: A Road Map.", in *Future of Software Engineering* ed. Finkelstein, A., ACM Press, 2000.

[8] Parnas, DL., "Designing Software for Ease of Extension and Contraction", IEEE Trans.on Software Engineering, Volume 5 (2), 1979.

[9] Costello, R. & Liu, D., "Metrics For Requirements Engineering," *Journal Of Systems and Softwar*e, Vol. 29, 1995.

[10] Carr, M., Konda, S., Monarch, I., Ulrich, F., & Walker, C., "Taxonomy Based Risk Assessment", Tech Report, CMU/SEI-93-TR-6, ESC-TR-93-183, Jun 93.

[11] de Geus, A., "Planning as Learning", Harvard Business Review, Mar-Apr 1988.

[12] Jackson, M., "The World and the Machine."

[13] Bush, D., & Finkelstein, A., "Environmental Scenarios and Requirements Stability.", Proc. International Workshop on Principles of Software Evolution, ACM Press, 2002.

[14] Wack, P., "Scenarios: Shooting the Rapids", Harvard Business Review, Nov-Dec 1985.

[15] Dobson

[15a] Anton. A., & Potts, C., "The Use of Goals to Surface Requirements for Evolving Systems", Proc. ICSE, Apr 1998.

[15b] Chung L., Nixon B., Yu E., "Dealing with change: An approach using non-functional requirements." Requirements Engineering Journal, 1(4), 1996.

[16] Kelly, TP., "Arguing Safety – A Systematic Approach to Managing Safety Cases.", PhD Thesis, Sep 1998.

[17] C. Potts and K. Takahashi, "An Active Hypertext Model for System Requirements", Proc. IWSSD 7,, IEEE, Dec 1993.

[18] Anton, A., "Goal Identification and Refinement in the Specification of Software-Based Information Systems", Ph.D. Thesis, June 1997.

[19] A. Dardenne, A. van Lamsweerde & S. Fickas. "Goal-Directed Requirements Acquisition.", Science of Computer Programming, 20(1-2), Apr 1993.

[21] Mylopoulos, J., Chung, L., & Nixon, B., "Representing and Using Nonfunctional Requirements: A Process-Oriented Approach." IEEE Trans. on Software Engineering, 18(6), 1992.

[21a] Galt, M., Chicoine-Piper, G., Chicoine-Piper, N. & Hodgson, A., "Idon Scenario Thinking", Idon Ltd., 1997.

[21b] Schwartz, P., "The Art of the Long View", Wiley, 1998.

[24] Bush, D., Durand, H., Ellison, D., Rhodes-James, C. & Tulloch, A., "Alternative Futures for Air Traffic Service Provision in Europe", LBS Project, Jun 2001.

[25] Ingvar DH. "Memory of the future: An essay on the temporal organization of conscious awareness." Human Neurobiology 4/1985.