Assurance Based Development of Critical Systems

AUTHORS: P. GRAYDON, J. KNIGHT, E. STRUNK
PRESENTED BY: MIKE MAKSIMOVO
Overview

1. Introduction to Assurance Cases
2. Overview of the Problem
3. Assurance Based Development (ABD)
   ◦ Candidate Development Choices
   ◦ Selection of a System Development Choice
   ◦ Applying System Development Choice
4. Illustrative Example of the ABD Process
5. Discussion
Introduction

- Definition of ABD – “synergistic construction of a critical computing system and an assurance case....”
Introduction

- Definition of ABD – “synergistic construction of a critical computing system and an assurance case....”
Introduction

◦ Definition of ABD – “synergistic construction of a critical computing system and an assurance case....”
◦ Definition of an Assurance case – “a documented body of evidence that provides a convincing and valid argument that a specified set of critical claims regarding a system's properties are adequately justified for a given application in a given environment” Scott and Krombolz (2005)
Safety Case

- Safety Cases are a subset of Assurance Cases that argue the safety of a system.

Q: What do they look like?
Safety Cases are a subset of Assurance Cases that argue the safety of a system.

Q: What do they look like?
A: It depends..

We have various types:
- Textual
- Graphical
Safety Cases are a subset of Assurance Cases that argue the safety of a system.

Q: What do they look like?
A: It depends..

We have various types:
- Textual
- Graphical (Ex. GSN Notation)
Current Development Practices

- Current dependability assurance approaches are ad hoc.

- Developers carry out dependability testing on isolated units without being able to evaluate the ensuing effects to the system as a whole.

- Assurance cases produced at the end of development might not have enough evidence from the development process.
Current Development Practices

- Current dependability assurance approaches are ad-hoc.

- Developers carry out dependability testing on isolated units without being able to evaluate the ensuing effects to the system as a whole.

- Assurance cases produced at the end of development might not have enough evidence from the development process.

All of this can lead to the revisiting of development steps after the development process is complete!
Assurance Based Development

- Confidence that the system will meet its dependability goals is evaluated throughout the development process.

- The system and its assurance argument are co-developed so that the impacts of a development choice are available at the time the choice is made.
Assurance Based Development

- Confidence that the system will meet its dependability goals is evaluated throughout the development process.

- The system and its assurance argument are co-developed so that the impacts of a development choice are available at the time the choice is made.

- This helps avoid and detect potential assurance difficulties as they arise.

- The Assurance Case can be exploited to drive development choices.

- You have confidence that you have enough evidence to support your claims.

- You have confidence that you are producing a dependable product.
ABD Workflow Overview

Assurance Based Development assumes:

- the availability of system dependability requirements
- the availability of a description of the given architecture
ABD Workflow Overview

1. Examine the unsatisfied goals
   - Developer actions in ABD workflow
   - Make a system development choice (architecture, language, etc.)

2. Select a development choice
3. Applying system development choice

1. Create a list of candidate development choices
1. Developers brainstorm choices that will lead to a system that meets its functional, cost, dependability and other goals.

2. Developers enumerates candidate development choices.

3. Developers then consider familiar choices or may solicit suggestions from colleagues.

There are costs associated with the consideration of more choices!
Selection of a Development Choice

Selection of a choice is based on 7 criteria:

1. **Functionality**
2. **Restriction on later choices**
3. **Evidence of dependability**
4. **Cost**
5. **Feasibility**
6. **Applicable standards**
7. **Non-functional requirements**
Example - Anti-lock braking system:

a) A single processor.

b) Two processors whose outputs are compared.

c) Three processors whose outputs will be voted on (TMR).

d) Many processors on a real-time bus.
Applying a Development Choice

Once a development choice is made:

1. The choice is applied to the system.

2. The assurance case is updated to reflect its effect.
Example system – Runway Incursion Prevention System (RIPS)

- Alerts pilots about potential runway incursions via IDS (Integrated Display System)
- Project developed for NASA

The authors focus on a subcomponent of RIPS, called the Runway Safety Monitor (RSM).
The Given Architecture
Top Level Assurance Goal

Assume that RSM is required to meet the following two requirements:

- If the quality of the supplied data is adequate, detect runway incursions involving ownership within \( t \) time units after they begin with probability greater than or equal to \( P_0 \).
- If the quality of the supplied data is inadequate, report a failure of RSM with probability greater than or equal to \( P_1 \) within \( u \) time units.
Overall approaches for the real-time requirements:
1. Sequential
2. Concurrent
   ◦ Synchronous
   ◦ Asynchronous

Requirement for the detection of corrupt/missing data:
1. A system module can
   ◦ Generate an event
   ◦ Time-out
2. Other
First System Development Choice

Development Choices Made:

- Sequential code implementation
- Each software module is responsible for detecting and reporting errors in the data that it handles
Second System Development Choice

Available choices to address G4 (failure detection):

- New architectural pattern
- Implementing an object-oriented architecture
- Functional decomposition
Available choices to address G4 (failure detection):
- New architectural pattern
- Implementing an object-oriented architecture
- Functional decomposition
Second System Development Choice

Available choices to address G4 (failure detection):
- New architectural pattern
- Implementing an object-oriented architecture
- Functional decomposition
Second System Development Choice

G4
RSM data inadequacy detection is implemented with required dependability

C3
RSM system architecture

ST2
Functional decomposition

G4.1
Ownership runway locator reports failure with $p \geq p_2$ if quality of information is inaccurate

G4.2
Conflict detector reports failure with $p \geq p_3$ if quality of information is inadequate

G4.3
RSM detects and reports failures in called functions with $p \geq p_4$

G4.4
Ownership runway locator detects & reports failures in itself and in called functions with $p \geq p_5$

G4.5
Runway database reports failure with $p \geq p_6$ if it can’t locate the runway

G4.6
Ownership position component reports failure with $p \geq p_7$ if quality of information is inadequate

G4.7
Runway model reports failure with $p \geq p_8$ if quality of information is inadequate

G4.8
Traffic positions component reports failure with $p \geq p_9$ if quality of information is inadequate

G4.9
Conflict detector detects and reports failures in itself and in called functions with $p \geq p_{10}$
Third System Development Choice

- TPC must detect inadequate information received from ADS-B due to:
  - Other aircraft reporting incorrect data.
  - Data can be corrupted in transit.
  - Data can be stale due to no updated data received
Third System Development Choice

Available choices to address G4.8:

- Impose reasonableness criteria.
- Incorporate redundant source of information, such as a radar or a camera with which to compare information.
Third System Development Choice

Available choices to address G4.8:
- Impose reasonableness criteria.
- Incorporate redundant source of information, such as a radar or a camera with which to compare information.
Third System Development Choice

Available choices to address G4.8:

- Impose reasonableness criteria.
- Incorporate redundant source of information, such as a radar or a camera with which to compare information.
Fourth System Development Choice

Easiest choice to address G4.8.4:

- Use a fully verified implementation of the traffic position component.
Fourth System Development Choice

Easiest choice to address G4.8.4:
- Use a fully verified implementation of the traffic position component.
Re-addressing a Choice

- At any point in the process, a developer may discover that a previous choice leads to an unsatisfiable goal.
Then it might be necessary to re-address our previous choice.
Questions

1. Do you foresee any (development) costs that may be associated with using the Assurance Based Development approach?

2. ABD assumes the availability of system requirements, including functional requirements and dependability requirements, as well as the high-level architecture in which the system will operate. Do you believe this is reasonable?

3. Do you think development creativity might be impacted by strictly following the safety case feedback during each development decision? (I.e. The product is dictated by the safety case, not the safety case dictated by the product.)

4. General thoughts about the paper?