The Scarecrow MBSE Embassy presents...

The Evidence Pattern

Prof Jon Holt & Simon Perry
Hello from Jon & Simon!
Protocol and resources

• Please mute microphones
• Raise any questions via chat
  – We will get to you
• Resources available off line
  – Slides with hand-drawn notes
  – Video
  – Access to Scarecrow’s KnoB
  – Access to the MBSE Embassy YouTube channel
1. Overview

1. What are Patterns?
2. An Approach to Documenting Patterns
3. Example Patterns
4. Evidence Pattern
5. Summary
6. References
7. Questions
1. What are Patterns?

• “A pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice”.
  – Christopher Alexander, ‘A Pattern Language’, 1977

• Original ideas from (building) architecture but adopted by software engineering OO community
  – Made more widely known through “Gang of Four Book” (1995)

• Ideas spread to analysis and data modelling through books by Fowler (1997) and Hay (1996)

• Now adopted in the wider systems engineering community
Enabling Patterns

• Specific constructs of modelling elements whose combination and subsequent use enables a number of systems engineering applications

• E.g. one used for the definition of interfaces or one used to ensure traceability throughout a model of a system

• See Holt, Perry & Brownsword (2016)
2. An Approach to Documenting Patterns

• When defining Patterns it is important to use a consistent model-based approach

• Approach we use known as the Framework for Architecture Frameworks (the FAF)
  – Developed by Holt & Perry (2013)

• Now being widely used across multiple domains and organisations
The Framework for Architectural Frameworks (FAF)

• The FAF was developed to improve the definition of Architectural Frameworks (AFs)
  – Can also be used to define Enabling Patterns (EPs)
• The FAF is designed to force anyone defining an AF to consider the following six questions:
  – What is the purpose of the AF/EP?
  – What domain concepts must the AF/EP support?
  – What Viewpoints are required?
  – What is the purpose of each Viewpoint?
  – What is the definition of each Viewpoint in terms of the identified domain concepts?
  – What Rules constrain the use of the AF/EP?
The FAF – Where does it sit?
The FAF Viewpoints

The Rules Definition Viewpoint is related to ALL the other Viewpoints and defines the Rules that constrain the Architectural Framework. Relationships to other Viewpoints are omitted from this diagram for clarity.
3. Example Patterns

- Following Enabling Patterns are available:
  - Interface Definition Pattern (HPB, KnoB)
  - Traceability Pattern (HPB, KnoB)
  - Test Pattern (HPB, KnoB)
  - Life Cycle Pattern (HPB, KnoB)
  - Epoch Pattern (HPB, KnoB)
  - Context Pattern (HPB)
  - Description Pattern (HPB, KnoB)
  - Analysis Pattern (HPB, KnoB)
  - Model Maturity Pattern (HPB, KnoB)
  - Evidence Pattern (HPB, KnoB) [1]
  - Certification Pattern (KnoB) [1]
  - Evaluation Pattern (SP) [2]
  - Responsibility Pattern (SCL) [2]

Getting the Patterns

HPB: Holt, Perry & Brownsword (2016)
KnoB: Scarecrow Consultants Ltd’s Knowledge Base
SP: Stephen Powley; outside Scarecrow
SCL: Scarecrow Consultants Ltd.

[1] Developed by the INCOSE UK MBSE IG
[2] Under development/refinement
4. Evidence Pattern

- **Purpose:**
  - An aid to the definition of chains of evidence and arguments used to support claims made about a subject
  - Captures claims, arguments & evidence, along with the subject of the claim & the claimant
  - Allows counter-claims (& counter-counter-claims etc.) to be captured

- **Number of Viewpoints:**
  - 4

- Developed by the INCOSE UK MBSE IG
Evidence Pattern - Concepts

- Claim-Argument Link
- Argument-Evidence Link
- Claim
- Claimable Item
- Subject
- Claimant
- Refuter
- Counter-Claim
- Evidence
- AVP
- EUP

Copyright © 2020
CDV [Package] Example [CDV - Definition of Claims by Safety Officer about System safety]

- **Claimant**: Safety Officer
- **Claim**: System is safe to use
- **Subject**: System safety
- **Claim**: Safety requirements have been exceeded
- **Relates to**: Claims
AV [Package] Example [AV - Example showing Arguments supporting "...
«claim»
System is safe to use

«argument»
System has been tested

«argument»
Safety statistics are good

«supports»
«supports»
Evidence Pattern – Evidence View

- **Safety case results**
- **Simulation results**
- **Analysis of safety statistics**
- **System has been tested**
- **Safety statistics are good**

```
«evidence»
Simulation results
«reinforces»
«reinforces»
«reinforces»

«evidence»
Safety case results
«argument»
System has been tested

«evidence»
Analysis of safety statistics
«argument»
Safety statistics are good
```
Evidence Pattern – Counter-Claim View

CCV [Package] Example [CCV - Example showing Counter-Claim made against an Argument-Evidence Link]

- «evidence» Simulation results
- «evidence» Safety case results
- «counter-claim» Simulation doesn’t cover all cases.
- «counters»
- «claimant, refuter» Simulation Specialist
- «argument» System has been tested
- «reinforces»
Certification Pattern – An Extension to the Evidence Pattern

• Purpose:
  – An aid to the definition of certified chains of evidence and arguments used to support claims made about a subject
  – Allows timeliness of certified claims to be captured
  – Allows accreditation of claims certifier to be captured
  – Extension of the Evidence Pattern

• Number of Viewpoints:
  – 5 (4 + 1)

• Developed by the INCOSE UK MBSE IG
5. Summary

• Patterns are now seen as a useful and powerful tool in model-based systems engineering
• Patterns need to be described in a consistent fashion
  – Can be done using the FAF
• A number of Enabling Patterns have been created
  – Many available from our KnoB
• Evidence Pattern looked at
  – An aid to definition of chains of evidence & arguments used to support claims made about a subject
6. References

- Gamma, E., Helm, R., Johnson, R. & Vlissides, J. ‘Design Patterns – Elements of Reusable Object Oriented Software.’ Boston, MA: Addison-Wesley; 1995
References continued
