Arguing Successful Development

M. Anthony Aiello and Benjamin D. Rodes
Dependable Computing
Charlottesville Virginia, USA
(tony.aiello,ben.rodes)@dependablecomputing.com

***Abstract*—This paper presents a new top-level pattern for an assurance case. The pattern consists of a claim "The system development is successful" and an argument structure based on the Seven Artifact model. The pattern addresses concerns often raised by engineers by explicitly identifying the role of critical development artifacts in the argument structure.**

I. Introduction

This paper presents a new top-level claim for the system assurance case and associated top-level argument structure based upon the Seven Artifact model [1].

Rigorous arguments are increasingly used for assurance cases [2]. Nevertheless, a significant challenge in the application of rigorous arguments is determining what the top-level claim should be. Many recommendations have emerged in the literature and in practice and are captured in patterns and pattern libraries [3], [4]. Candidate top-level claims include: fitness for use, satisfaction of requirements, and mitigation of hazards. Unfortunately, each of these claims has significant drawbacks.

Fitness for use is a poor choice because the claim is too flexible. Without refinement, fitness for use is meaningless: we must say what is meant by fitness for use by identifying criteria for assessment. As a result, in our experience, engineers do not know how to interpret the claim. Engineers ask what fitness for use means and, when shown that the claim requires a multi-legged argument that can include a variety of assessments, request instead a more concretely stated goal that has more relevance for their system.

Satisfaction of requirements is an obvious claim, but does not readily appear to cover assessment of safety. Requirements for safety critical systems often include safety requirements, so there is a degree of assessment of safety that may be addressed through satisfaction of requirements. However safety assessment cuts across requirements to look more broadly at what can go wrong for a system through the identification of hazards. Additionally and most importantly, satisfaction of requirements leaves open the question of where the requirements come from, if they are complete, and if they are correct. In our experience, engineers typically ask these questions when faced with a top-level claim of satisfaction of requirements.

Mitigation of hazards is an appealing choice for a safety case, an assurance case that focuses primarily on safety, but is insufficient if more comprehensive assurance is sought. Additionally, as noted above, the question of where the hazards come from, if they are complete, and if they are correct remains. In our experience, engineers typically ask these questions when faced with a top-level claim of hazard mitigation.

We advance a new top-level claim of successful development that is supported by a pattern based upon the Seven Artifact model [1]. The top-level claim and pattern speak immediately to the most important concern facing the development of any system: that the development activity succeeds. Informally, development is successful when the customer's problem is fully understood and completely solved by the developed system. The Seven Artifact model defines successful development more rigorously:

> System development is considered successful if (1) the problem is adequately defined and (2) the problem is solved. [1]

The top-level claim is supported by an argument structure that explicitly includes artifacts from the Seven Artifact model, ensuring that questions such as "where is the argument that requirements are correct" are easily and obviously addressed.

II. Motivation

Typical arguments focus on assessment activities, such as safety assessment, security assessment, regulatory compliance and satisfaction of requirements. These activities are extremely important and are rightly emphasized. To a large degree, they are all that matter for acceptance of the system.

Each of these assessment activities evaluates the system with respect to a set of previously identified criteria. Requirements satisfaction, for example, evaluates the system with respect to requirements. Safety assessment, similarly, evaluates the extent to which the system has mitigated identified hazards.

Typical uses of rigorous argumentation identify these criteria in one of two ways.

1) The most common way in which these criteria are introduced is as part of the argument in which assessment is discussed. Identification of criteria is effectively mixed with system assessment. This has the advantage of clearly associating the identification argument with the assessment argument, but has the disadvantage of mixing two distinct concepts in one argument.
2) An alternative approach is to use assurance-claim points (see [5]) to link from the assessment argument to the

Approved for Public Release; Distribution Unlimited (Case Number: 88ABW-2017-0489)

associated identification argument. In this approach, the identification argument takes on the role of a confidence argument: the identification argument provides confidence to the reviewer that the assessment criteria are correct and complete. This has the advantage of clearly separating the assessment argument from the identification argument. However, this use of assurance-claim points has the disadvantage of relegating the identification argument to the role of confidence argument.

We say "relegating" with full knowledge of the negative connotation associated with the term. In our experience, engineers are unhappy with the use of confidence arguments to fill roles such as correct and complete identification of requirements or correct and complete identification of requirements. While the confidence argument is not and should not be viewed as a second-class citizen to the spinal argument, engineers nevertheless often see the confidence argument in this way.

Concern about the correctness and completenes of critical engineering artifacts such as the requirements and the environment in which the system will operate has led to the development of reference models for systems engineering. In prior work, we introduced the Seven Artifact model to describe a rational and defensible problem-oriented approach to building systems.

The Successful Development argument pattern is therefore aligned with the Seven Artifact model, providing explicit argument claims under which arguments for the correctness and completeness of each artifact may be included. The pattern therefore addresses the concerns voiced by engineers by providing a top-level argument structure in which the identification arguments and the assessment arguments exist side-by-side in a single, inferentially consistent argument. Although arguments do not impose temporal ordering based on the order in which claims are presented, the Successful Development pattern presents each artifact from the Seven Artifact model in the order in which the artifact is developed, moving from left to right, and ends with assessment of the system.

III. The Successful Development Pattern

A high-level overview of the Successful Development pattern is shown in Figure 1; the complete pattern, documented in GSN [6], is shown at the conclusion of this report in Figure 2. The top-level claim is "The {system} development is successful", where {system} is meant to be replaced by the name of the system in question. The claim is supported by arguing the success of each component of development success, which are drawn from the Seven Artifact model: problem definition, context identification, solution definition, optionally solution development, and solution assessment.

A. Problem Definition

The claim associated with problem definition is "Definition of the problem to be solved is successful". The claim is supported by a strategy that links successful definition to adequate elicitation: we should thus conclude that problem definition is successful when adequate elicitation of the problem has taken place. The pattern therefore asserts success when an artifact is determined to be adequate; this linking of success to adequacy is used throughout the pattern. Adequate must be defined for the given application of the pattern, based on, for example, best practices for the domain. The supporting claim is therefore "Elicitation of the problem to be solved is adequate". This claim may be supported directly by evidence, if an appropriate and acceptable evidence scheme (see [7]) exists or a subargument, as necessary.

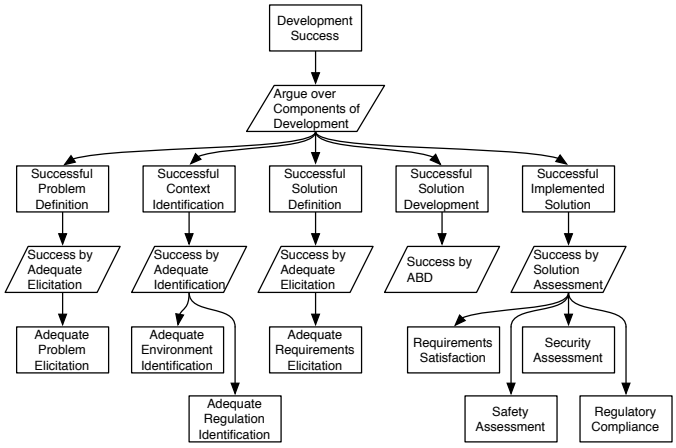

Fig. 1. Successful Development — High-level Overview

B. Context Identification

The claim associated with context identification is "Identification of the context in which the problem will be solved is successful". The claim is made with reference to the problem statement that resulted from problem definition. Similar to the above, the claim is supported by a strategy that links successful identification to adequate identification of the environment, the regulatory environment and any other elements of context that are relevant for the system development. We should conclude that the context identification is successful when each element of context has been adequately identified. Adequate must be defined for the given application of the pattern, based on, for example, best practices for the domain [8]. The supporting claims are each of the form "Identification of the environment in which the problem will be solved is adequate". These claims may be supported directly by evidence, if an appropriate and acceptable evidence scheme exists, or a subargument, as necessary.

C. Solution Definition

The claim associated with solution definition is "Definition of the solution to the problem is successful". The claim is made with reference to the problem statement and to each element of context that resulted from context identification. The claim is supported by a strategy that links successful definition to adequate elicitation of system requirements. Adequate must be defined for the given application of the pattern, based on, for example, best practices for the domain. The supporting

claim is therefore "Identification of system requirements is adequate". This claim may be supported directly by evidence, if an appropriate and acceptable evidence scheme exists or a subargument, as necessary.

D. Solution Development

Perhaps counterintuitively, the pattern defines the claim for solution development as optional. The reason this claim is optional is that, when system development is complete, details about how solution development was undertaken are relevant only insofar as they appear in the system assessment argument. This argument would most naturally follow the patterns laid out in assurance-based development [9].

E. Solution Assessment

The final claim is the claim associated with solution assessment: "The implemented problem solution is successful". The claim is supported by a strategy that link successful implementation to each identified element of assessment, including adequate satisfaction of requirements, adequate safety, adequate security, and adequate compliance with regulations. The form of each supporting claim is particular to the type of assessment expressed.

For example, the claim associated with requirement satisfaction is "The implemented problem solution adequately satisfies its requirements". This claim is made with reference to the requirements defined in the solution definition.

The claims associated with safety is "The implemented problem solution is adequately safe". If adequate safety is argued through hazard mitigation, the argument under this claim must include both identification of hazards as well as hazard mitigation. The argument for hazard mitigation will, moreover, typically reference the safety requirements and the argument for satisfaction of the safety requirements.

IV. Conclusion

We present a pattern that argues successful development. The pattern consists of a top-level claim that system development is successful and an argument structure that is closely aligned with artifacts from the Seven Artifact model. The pattern is addresses concerns that are often raised by engineers faced with other patterns that do not include claims such as the correct and complete identification of requirements explicitly and up-front in the argument structure.

A. Pattern Flexibility

We view argument patterns as less rigid and prescriptive than is often suggested by more traditional argument patterns. Patterns should be used to guide discussion and to provide a basis for argument development. We view the concepts captured with the Successful Development pattern as having primary importance.

Fine-grained argument structures and phrasing used within arguments can often be adequately expressed in more than one way. Furthermore, each domain of application likely requires subtle variations and refinement to argument structure and phrasing to meet the expectations of relevant stakeholders and argument reviewers. The purpose of an argument is communicate the rationale for justifiable assurance — rigidity in argument structure rarely facilitates good communication.

Acknowledgment

This work was funded in part by USAF AFLR/RQQA contract FA8650-14-C-2528.

References

[1] M. A. Aiello and B. D. Rodes, "The seven artifact model," Dependable Computing, Tech. Rep. DC-2017-01, January 2017.

[2] D. J. Rinehart, J. C. Knight, and J. Rowanhill, "Understanding what it means for assurance cases to 'work'," NASA, Contractor Report To Appear, October 2016.

[3] T. Kelly and J. McDermid, "Safety case patterns-reusing successful arguments," in *Understanding Patterns and Their Application to Systems Engineering (Digest No. 1998/308), IEE Colloquium on*, apr 1998, pp. 3/1 –3/9.

[4] R. Hawkins, K. Clegg, R. Alexander, and T. Kelly, "Using a software safety argument pattern catalogue: Two case studies," in *Proceedings of the 30th International Conference on Computer Safety, Reliability, and Security*, ser. SAFECOMP'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 185–198. [Online]. Available: http://dl.acm.org/citation.cfm?id=2041619.2041640

[5] R. Hawkins, T. Kelly, J. Knight, and P. Graydon, "A new approach to creating clear safety arguments," in *Advances in Systems Safety*, C. Dale and T. Anderson, Eds. Springer, 2011, pp. 3–23.

[6] K. Attwood, P. Chinneck, M. Clarke, G. Cleland, M. Coates, T. Cockram, G. Despotou, L. Emmet, J. Fenn, B. Gorry *et al.*, "GSN community standard version 1," *Origin Consulting Limited, York*, November 2011.

[7] P. Graydon and C. Holloway, ""evidence" under a magnifying glass: Thoughts on safety argument epistemology," 2015.

[8] J. C. K. Jonathan Rowanhill, "Domain arguments in safety critical software development," in *27th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, October 2016.

[9] P. J. Graydon, J. C. Knight, and E. A. Strunk, "Assurance based development of critical systems," in *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*, June 2007, pp. 347–357.

Fig. 2. Successful Development — Complete Pattern