Assurance 2.0 in a Nutshell

Robin Bloomfield (City, Univ. of London) and John Rushby (SRI)

SRI CSL Technical Note, 14 October 2024

This is intended as a memory aid, not a replacement for reading the longer documents that can be found at https://www.csl.sri.com/users/rushby/assurance2.0.

Purpose of Assurance 2.0: it's a rigorous and systematic approach to developing, presenting, and examining assurance cases to support *indefeasible confidence* in safety or other critical properties

- **Structure**: Claims, Argument, Evidence (CAE), plus Theories and Defeaters
  - **Claims**: precise and meaningful statements about system and environment, presented as atomic propositions in natural language. Some may be marked as **assumptions**
    - Claims may state probabilistic properties and uncertainties (e.g., $\texttt{pfd} < 10^{-4}$)
  - **Argument**: typically presented as a tree-like structure of nodes; each node has a **parent** claim, one or more **subclaims**, and usually a **side-claim**
    - Just 5 kinds of (building) **blocks** for argument nodes: **concretion**, **substitution**, **decomposition**, **calculation**, **evidence incorporation**. See Figure 1
    - **Conjunction** of subclaims and side-claim should *deductively* entail parent claim; otherwise flag as *inductive* & apply special care such as confirmation theory (below)
    - **Disjunctive** decompositions are available (useful in *refutational* subcases, see over)
    - **Side-claim** typically factors out deductiveness conditions (e.g., subclaims partition parent claim, or parent claim distributes over components enumerated in subclaims)
    - A **narrative justification**. . . *justifies* all this; may cite an external **theory**
    - LLMs can interpret claims as *knowledge graphs* over standardized ontology, which can then be checked for consistency using *answer set programming* [1]
  - **Evidence**: a coherent *assembly* of reviews, analyses, tests etc. that *measures* some property of the system. The measurement in turn supports some *useful* inference. This is justified by a **narrative description** that may cite an external **theory**
    - Parent claim of an evidence incorporation block is called the **measured claim**: it says what the evidence is (e.g., testing achieved MC/DC coverage with no faults)
    - Above that is a substitution block that derives a **useful claim** from the measured claim; it says what the evidence means (e.g., there is no unreachable code)
    - Weight of evidential support for the useful claim is examined using the measures of **confirmation theory**, e.g., (Keynes): $\log \frac{P(C \mid E)}{P(C)}$, or (Good): $\log \frac{P(E \mid C)}{P(E \mid \neg C)}$
- **Theories** are self-contained technical descriptions and assurance arguments for specific assurance methods (e.g., static analysis) or (sub)systems (e.g., altitude hold). They include narrative justifications for their arguments and may serve as **templates** for assurance (sub)cases
  - Subcases can be instantiations of parameterized (and ideally *pre-certified*) theories
  - Instantiations can be *expanded* in place (like a macro), or *referenced* (like a subroutine)
  - Much of a case can be *synthesized* from a library of such parameterized theories
  - Standards bodies should deliver theories not guidelines.
  - Overall case can be summarized by enumerating its theories

- **Defeaters** are used to challenge a case, have their own subcases to *refute* or *support* them
  - **Exact Defeaters** introduce negation & refutation: support *eliminative argumentation*
  - Other kind are called **exploratory defeaters** and must eventually be *refuted* (but can then be retained as commentary), or *accepted* as **residual risks**
- An **assurance case** is a package of claims, argument, evidence, plus all supporting theories and narratives; deployment **decision** may be justified in a **sentencing statement**
  - The argument must be **completed**: a *connected* tree/graph where leaves are either evidence, assumptions, or residual risks (or references to completed subcases)
  - Must have no unrefuted defeaters, except those identified as **residual risks**
- **Assessment** employs 4 perspectives: logical, probabilistic, dialectical, and residual risks
  - **Logical assessment** requires a completed argument that is logically valid and **indefeasibly sound**: no credible new information would change the judgement
  - Also, there are (fairly weak) ways to externally assess **probabilistic confidence** in a case. Main value is supporting principled ways of graduating effort vs. risk.
  - **Dialectical examination** combats complacency and confirmation bias: uses defeaters (for claims and argument nodes) and confirmation measures (for evidence).
  - **Residual doubts** are assessed for quantity & risk and all but negligible risks eliminated

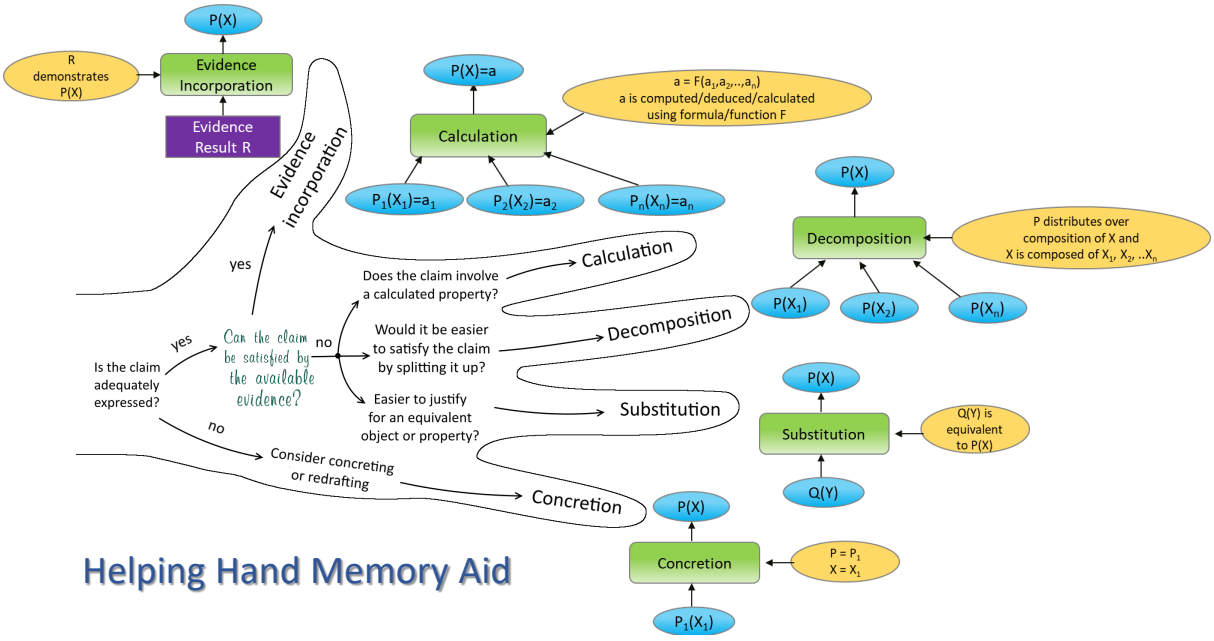

Figure 1: Assurance 2.0 Building Blocks and "Helping Hand" Mnemonic (from [2])

References

[1] Anitha Murugesan et al. Automating semantic analysis of system assurance cases using goal-directed ASP. arXiv:2408.11699, August 2024. To appear in a special issue of TPLP.

[2] Srivatsan Varadarajan et al. CLARISSA: Foundations, tools and automation for assurance cases. In *42nd AIAA/IEEE Digital Avionics Systems Conference*, Barcelona, Spain, October 2023.