McMaster University

Inspiring Innovation and Discovery

Practically Formal Development and Assurance of Complex Software-Intensive Safety-Critical Systems

Alan Wassyng

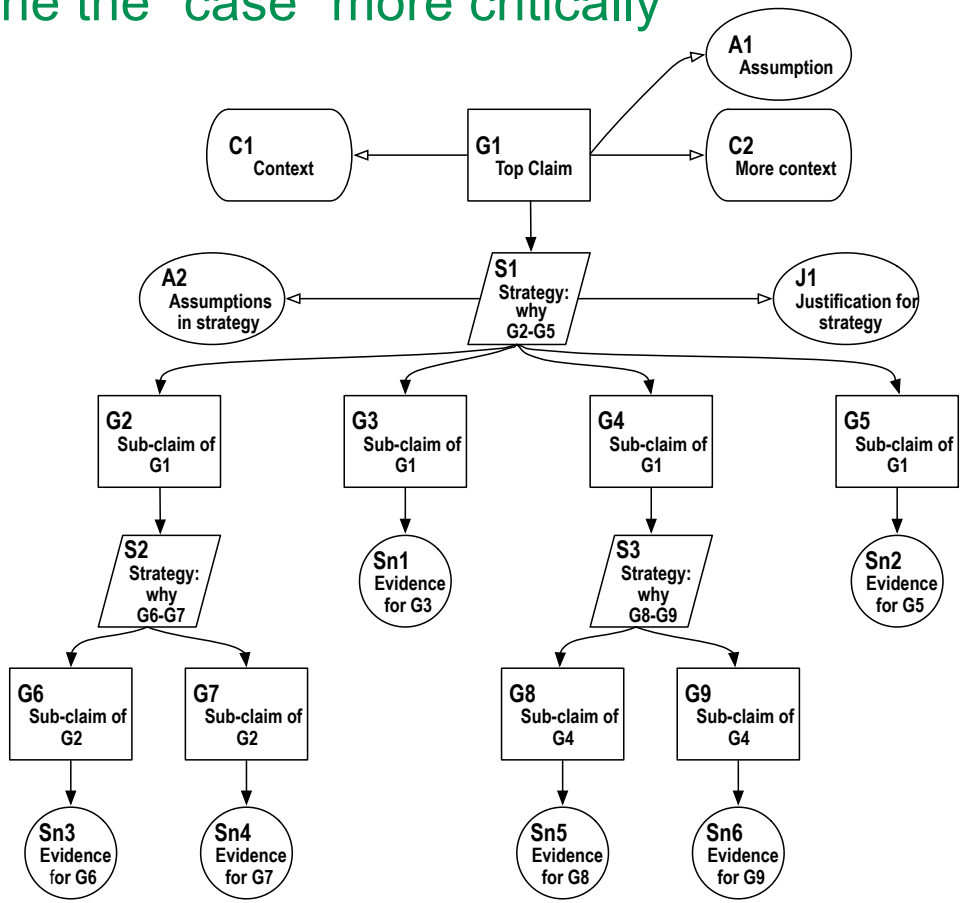
work with Zinovy Diskin, Nicholas Annable, and Mark Lawford

CyPhyAssure, 20 March 2019

GSN-like Assurance Cases

Pros

- Appealingly intuitive
- Does seem to improve safety (for example) by making people examine the “case” more critically



GSN was introduced by Tim Kelly in 1998 [Kelly1998]

He and others have turned it into the most popular notation for assurance cases

GSN-like is meant to include other similar notations such as Claims, Arguments and Evidence (CAE) and tabular approaches

McSCert

GSN-like Assurance Cases

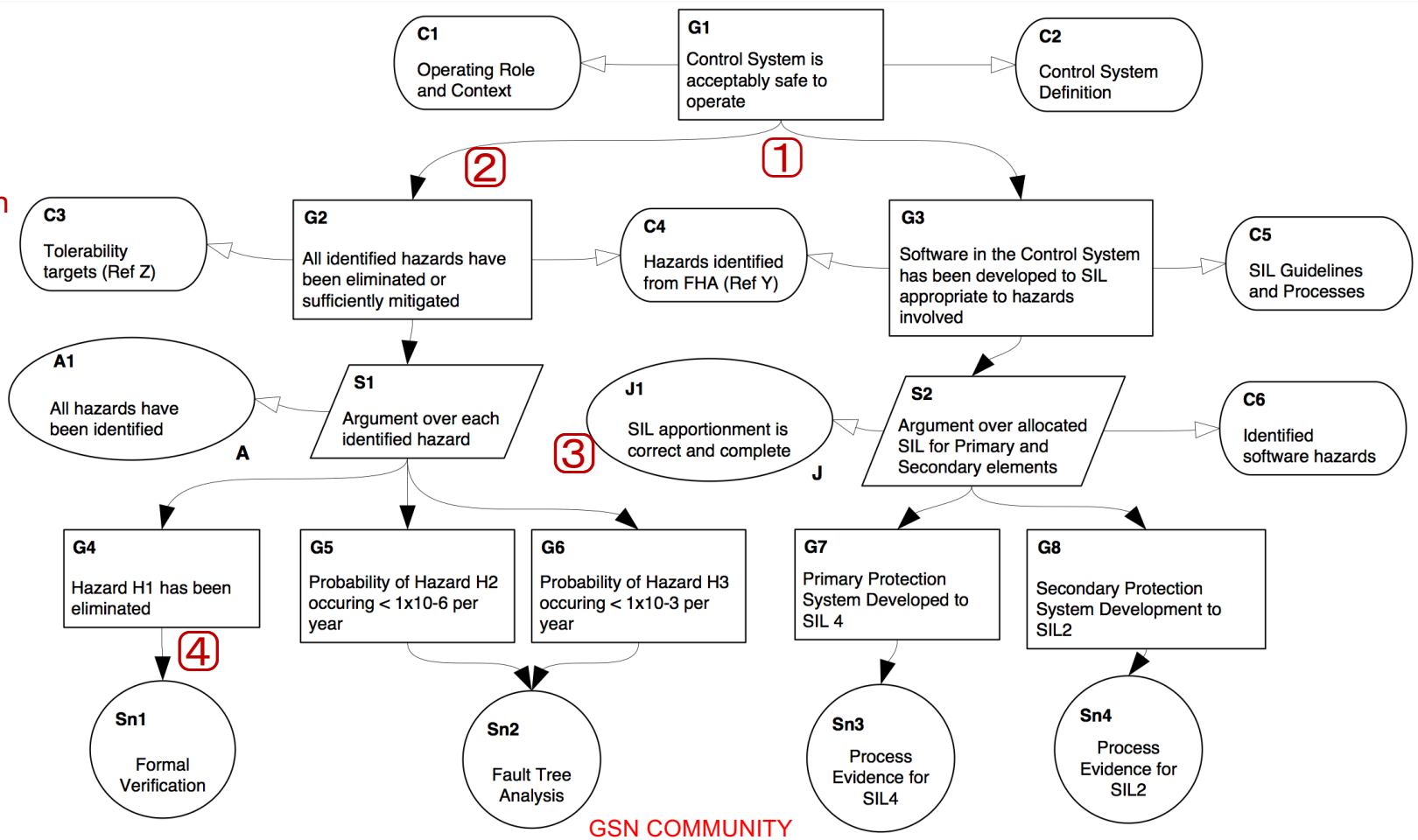
Cons [Wassyng2011]

- Cases tend to have an ad hoc structure dictated by experience & preference
  - Patterns help but they are not the “solution”
- Cross-cutting concerns abound
- There is an element of confirmation bias
- Provides a false sense of confidence (no matter how we “measure” confidence) since we think our reasoning is rigorous (most never claim formal) – but it is not
- Safety impact analysis is difficult to impossible
  - In general, effective traceability is tough

GSN Intent

The actual intent behind GSN was fundamentally flawed in some ways

1. Strategy is optional!
2. The arrows indicate decomposition, not an argument based on premises
3. The promised explicit argument is just not present – if justification was supposed to represent reasoning as some people claim, then why does it support strategy instead of the other way around?
4. GSN/CAE experts say that the only place an explicit argument is necessary is to support evidence, but there is no strategy node even – so, what argument? Better in SACM 2.0



GSN COMMUNITY STANDARD VERSION 1 - 2011 [GSN2011]

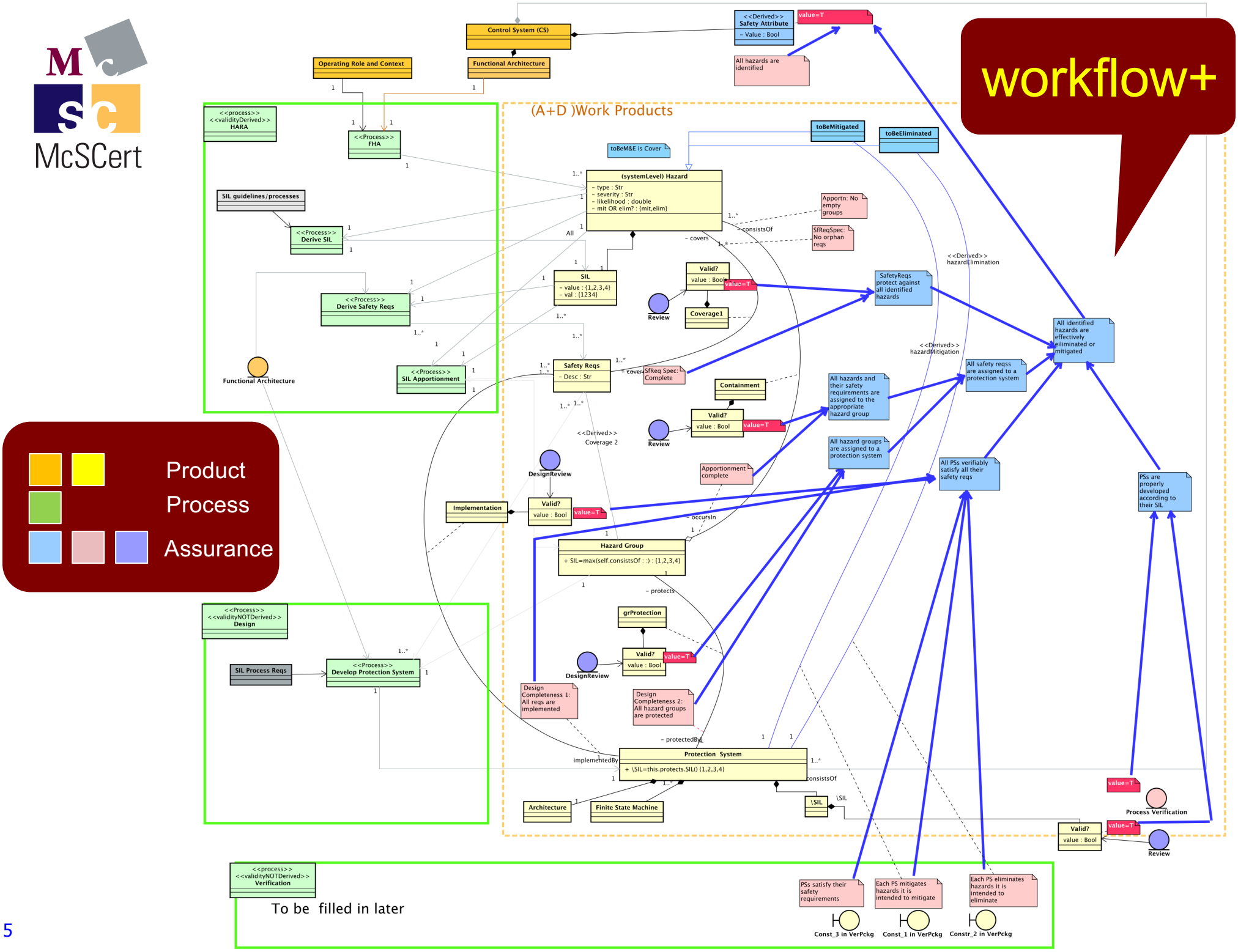
McSCert

A New Approach

- Before we explore why we came up with a different approach the following slide shows components of the new assurance – as an integral part of the development
- This gives us some idea of where we are headed
- The basis is metamodels and model transformations, refinements and instances

McSCert

workflow+



Product
Process
Assurance

Assurance Case Templates (ACTs)

McSCert

Describes an assurance case (AC) for a product line or domain.

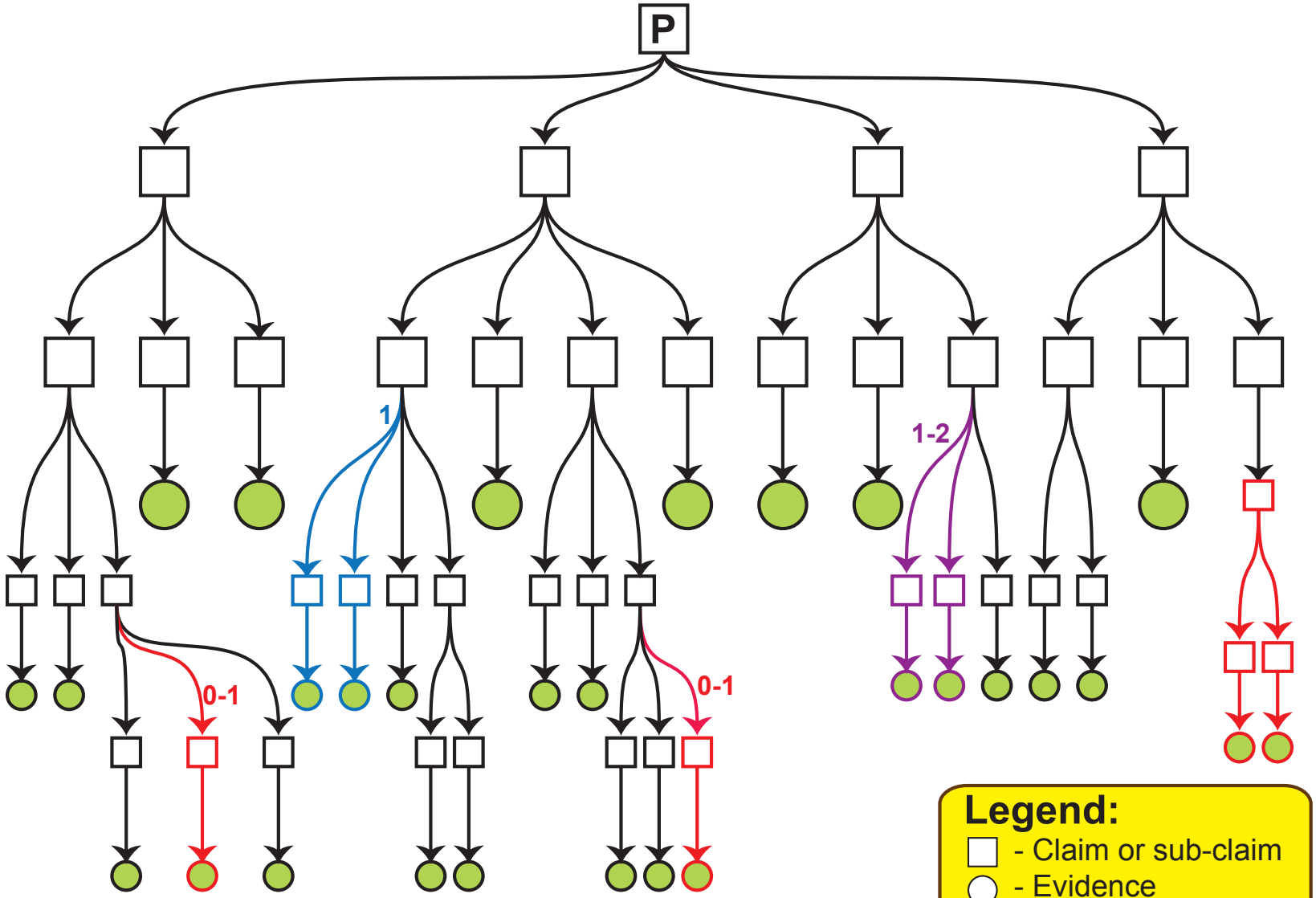
Needs to be instantiated for a specific product.

Developed BEFORE building products.

Reduces confirmation-bias.

Facilitates incremental assurance.

Could be used to direct development – as a process guide or even as a standard.



acceptance criteria on evidence required specified in the template

[Wassyng2016]

Assurance Case Templates (ACTs)

McSCert

Describes an assurance case (AC) for a product line or domain.

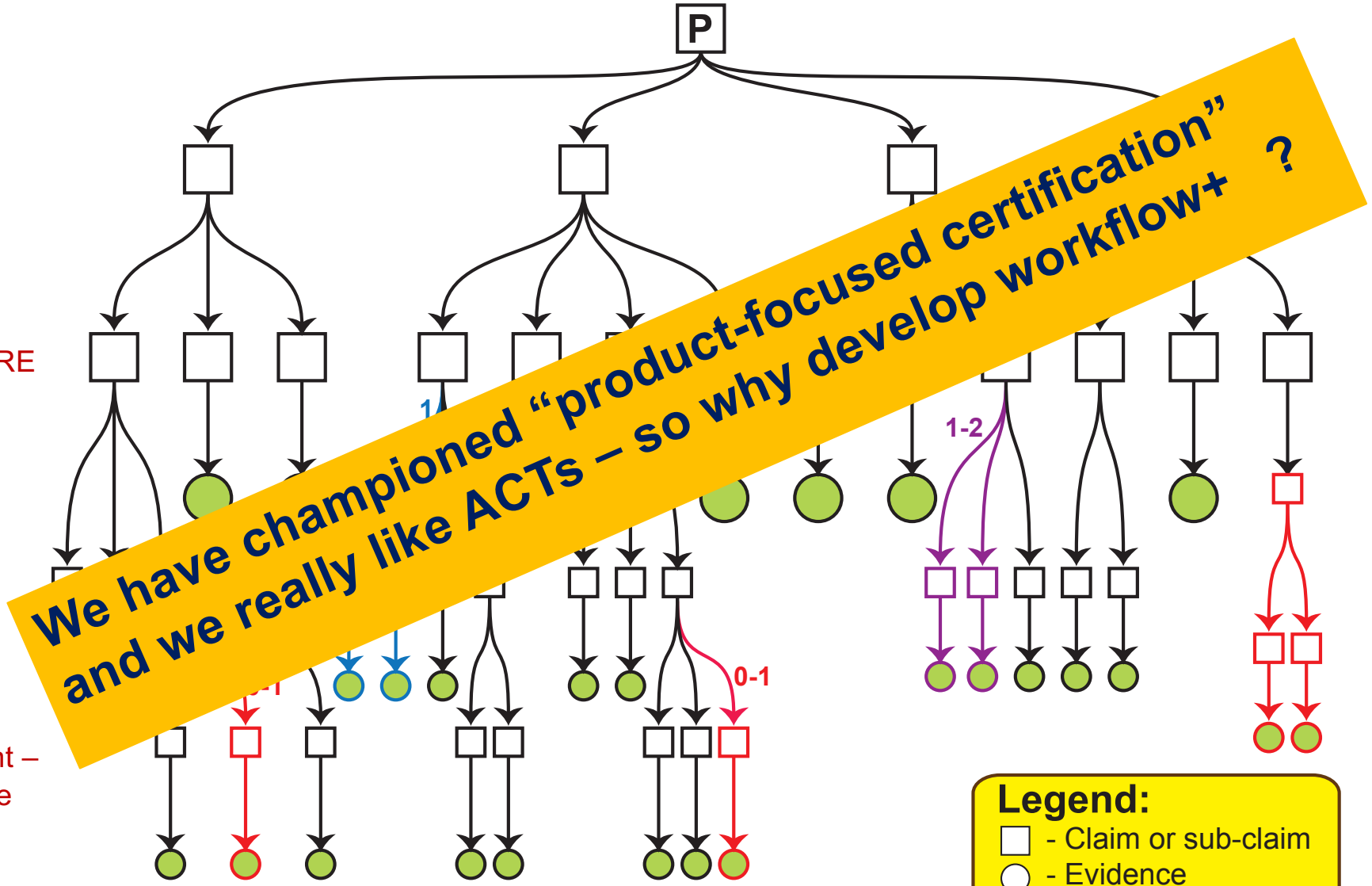
Needs to be instantiated for a specific product.

Developed BEFORE building products.

Reduces confirmation-bias.

Facilitates incremental assurance.

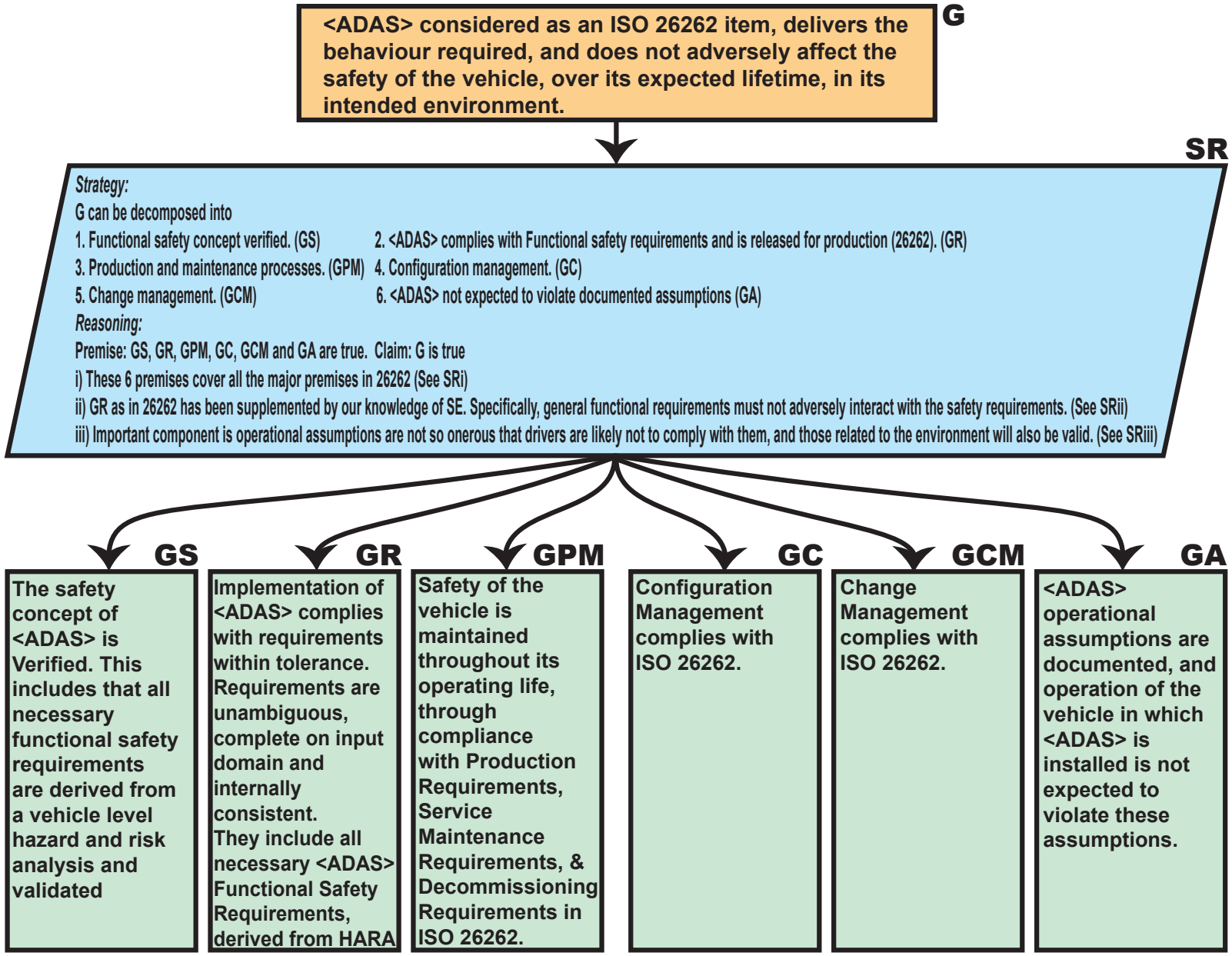
Could be used to direct development – as a process guide or even as a standard.



acceptance criteria on evidence required specified in the template

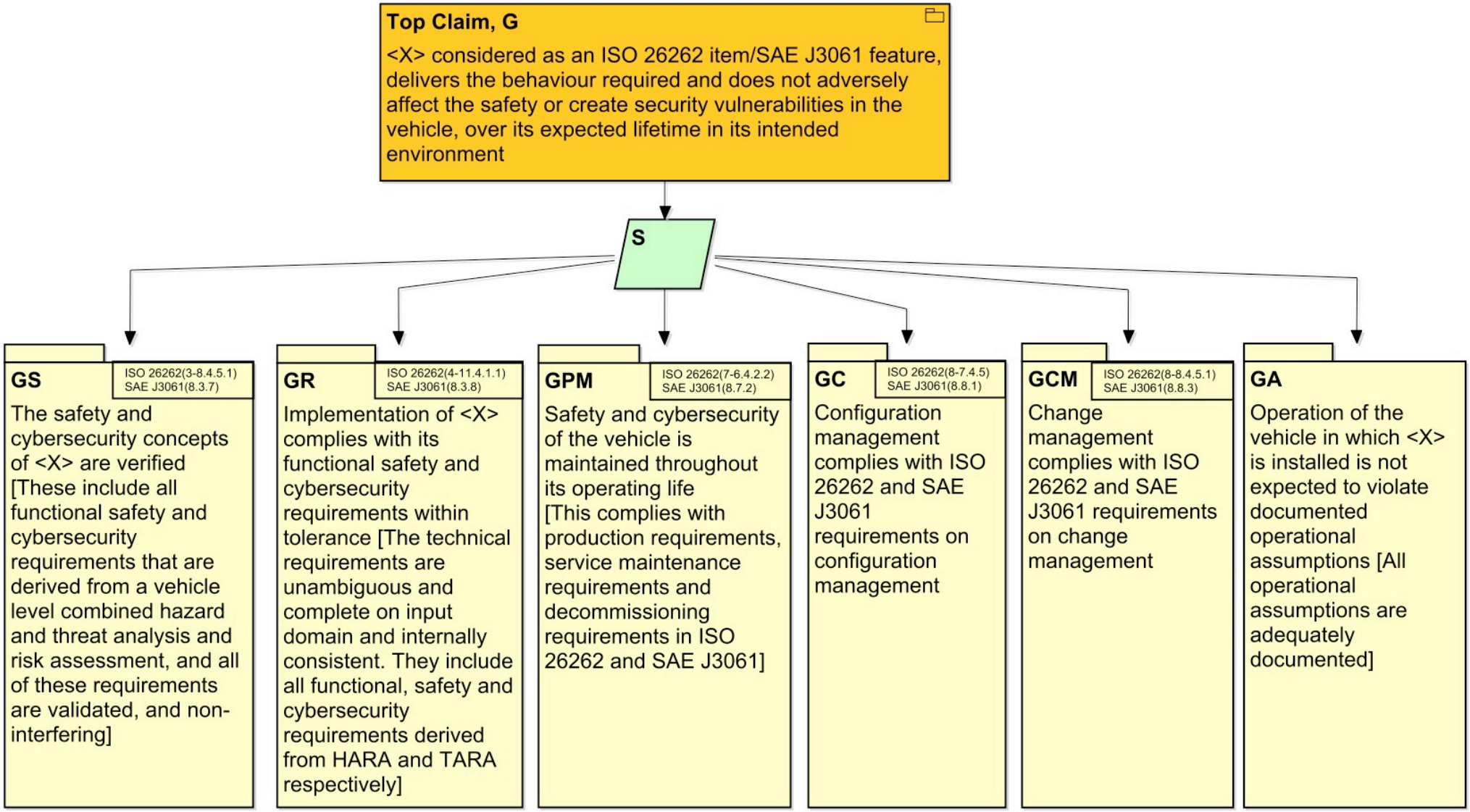
[Wassyng2016]

Extract from Assurance Case Safety Template for ADAS



[Chowdhury2017]

Extract from Assurance Case Safety & Security Template



[Chowdhury2018]

McSCert

We Like the Benefits of ACTs

- They are templates for a product line developed ahead of work on individual products
- They facilitate incremental assurance in the same way that *information hiding* makes software designs robust with respect to **anticipated** changes
- They provide some protection against confirmation bias
- They standardize an approach so that regulators (in particular) can develop expertise in evaluating them

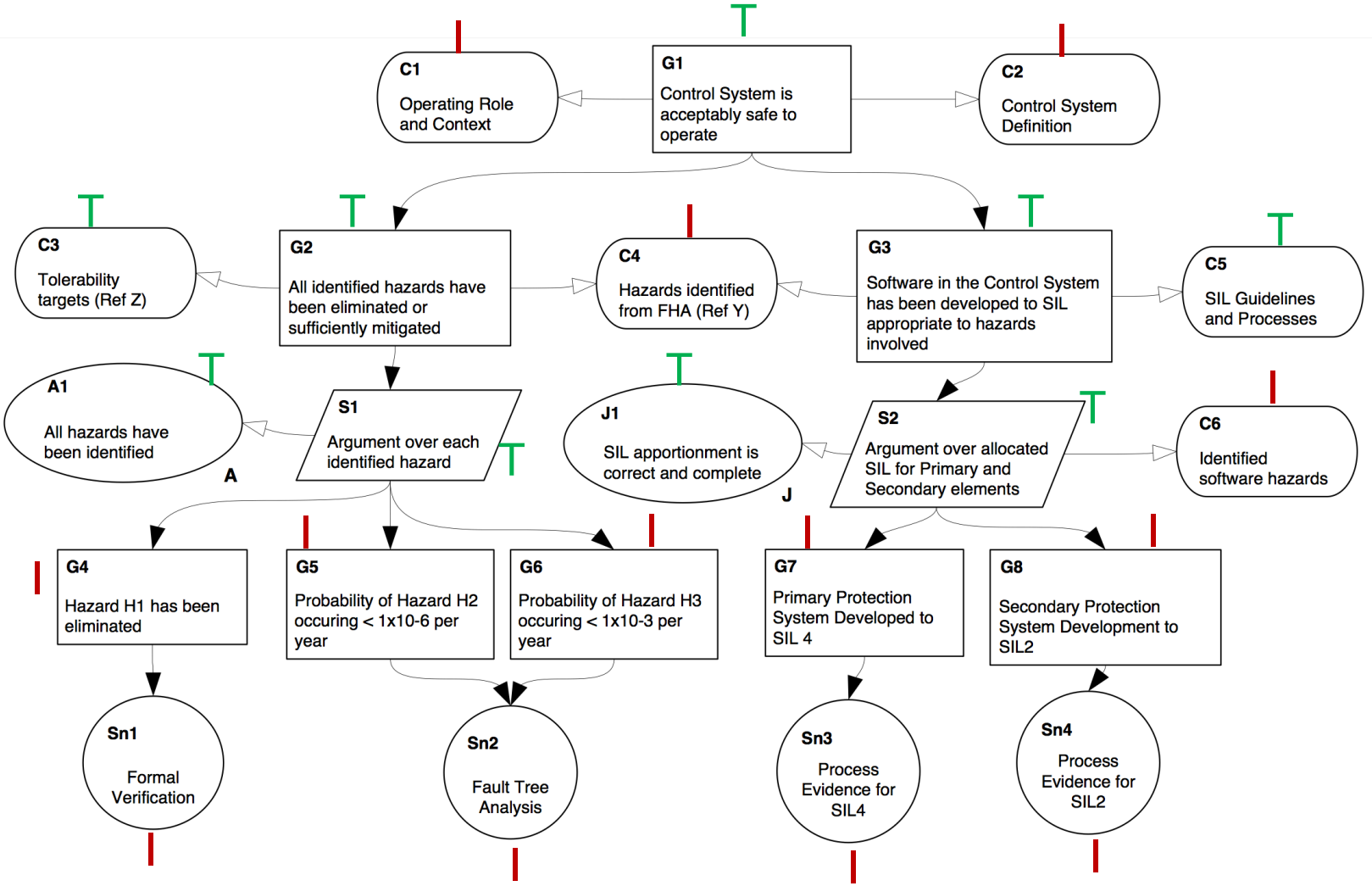
McSCert

But Could Not Overcome the Flaws in GSN

- GSN is inherently ad hoc – there is no theoretical foundation for its assurance steps
- The “logic” for the explicit argument is either non-existent or hopelessly inadequate to deal with a property like safety
  - We have not managed to define safety semantics for the arrows in GSN – No, SACM does not do this [SACM2.0]
  - Safety or security impact analysis in GSN is fundamentally unsound
- Cross-cutting concerns are unavoidable
- GSN diagrams typically are a mixture of template and instance – see next slide

Template or Instance?

This is not inherent in GSN – but GSN is so ad hoc in its approach that this is what we see most of the time



McSCert

Why workflow+?

- We need process!
  - Even if we want product-focused certification we have always said it has to be based, at the very least, on notions of an idealized process
- We can cope with process, product, people, in much more structured ways
- workflow+ is a formal notation and the inherent assurance steps are not ad hoc
- workflow+ provides a rigorous way of producing and maintaining traceability links providing a method to turn incremental analysis into syntactic checks

Where did it come from?

- Last year we published an early version of the framework in MoDELS [Diskin2018]

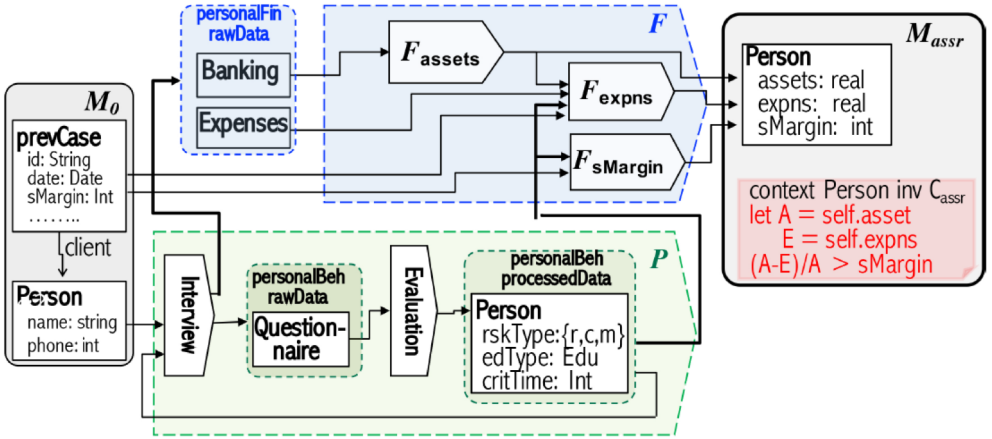
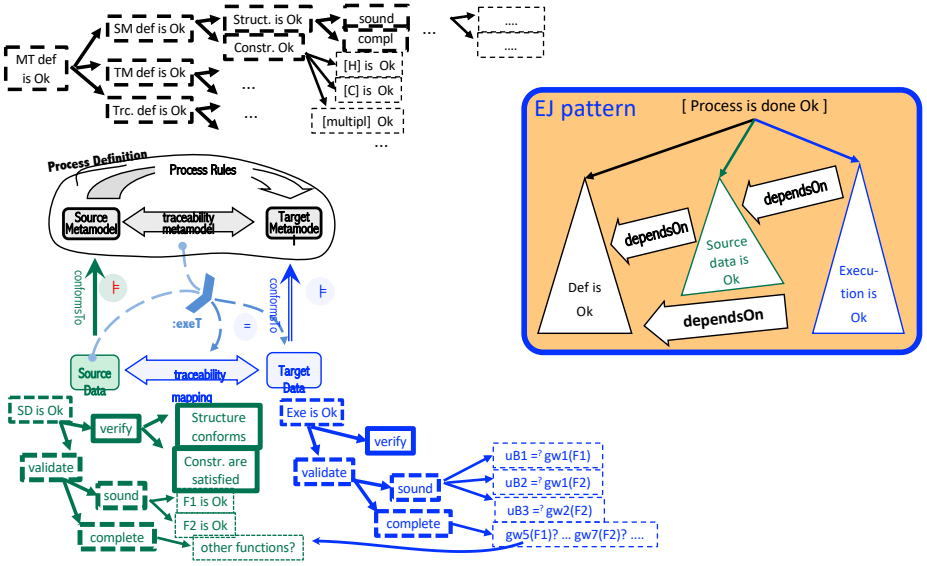


Figure 3: The process in detail



These “assurance steps” are suggested by the mathematical structure - no longer ad hoc.

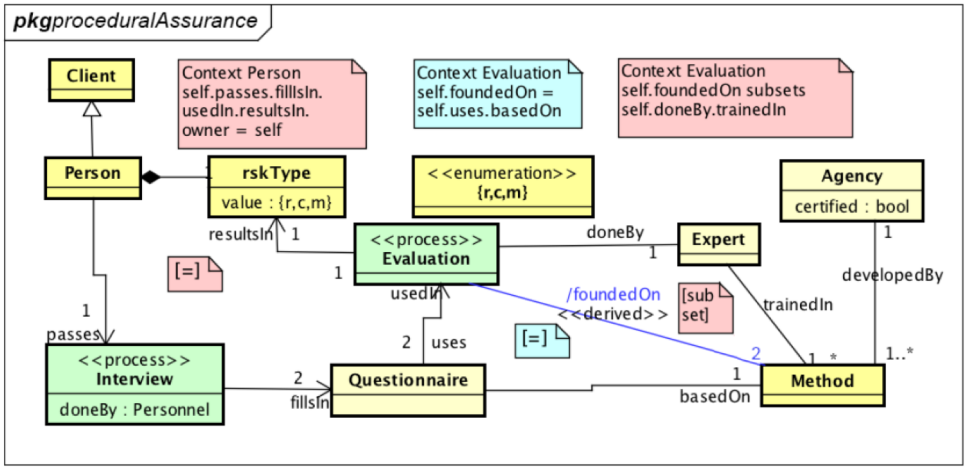


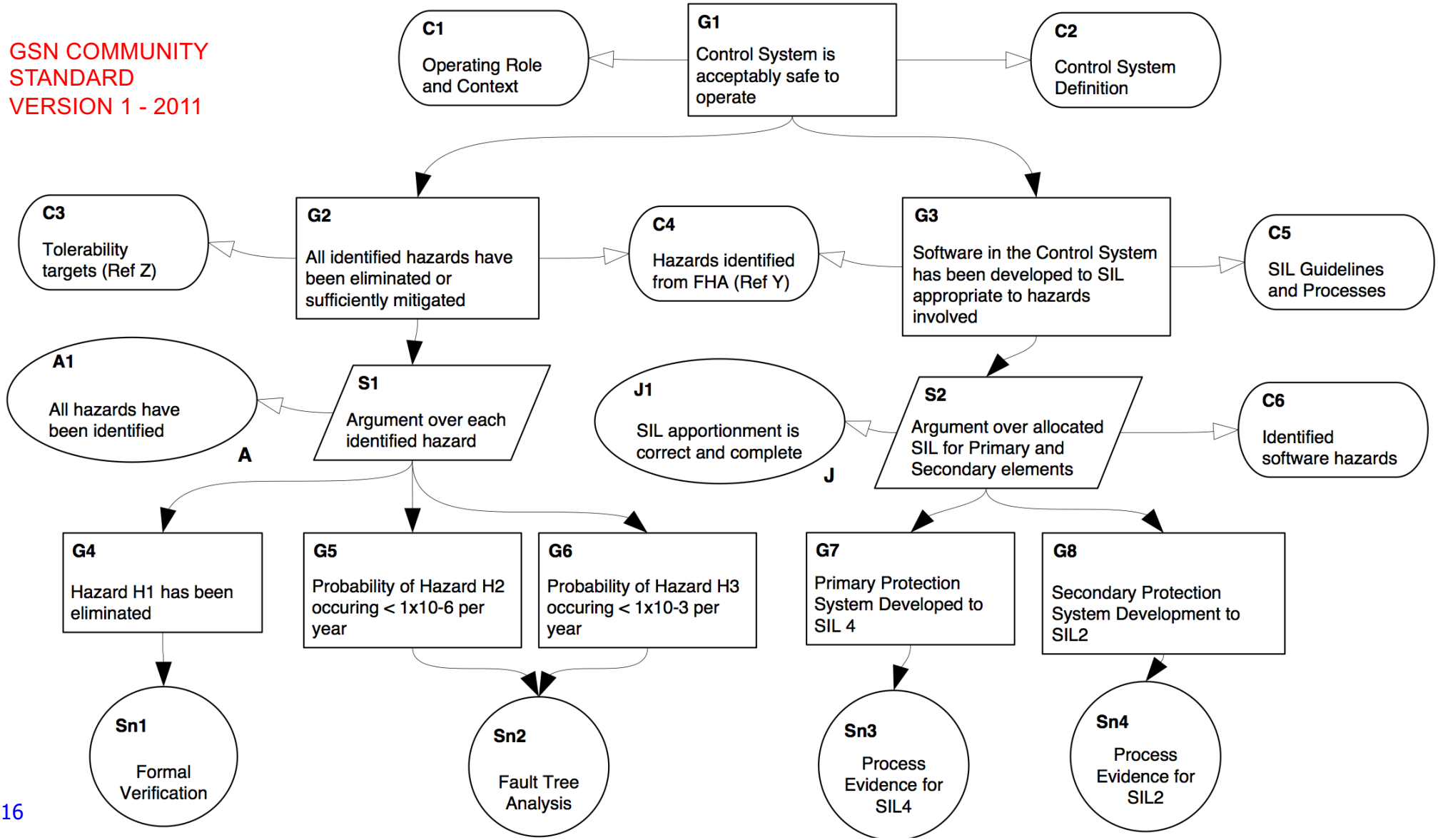
Figure 4: Workflow and metamodels together

Demonstrate by Way of Example

Example of an Assurance Case Segment

Note: We do realize that this was produced to illustrate GSN constructs not as an example of a safety case, but it is useful for our purpose since it is (sort of) plausible and is simple enough but with some (implicit) detail

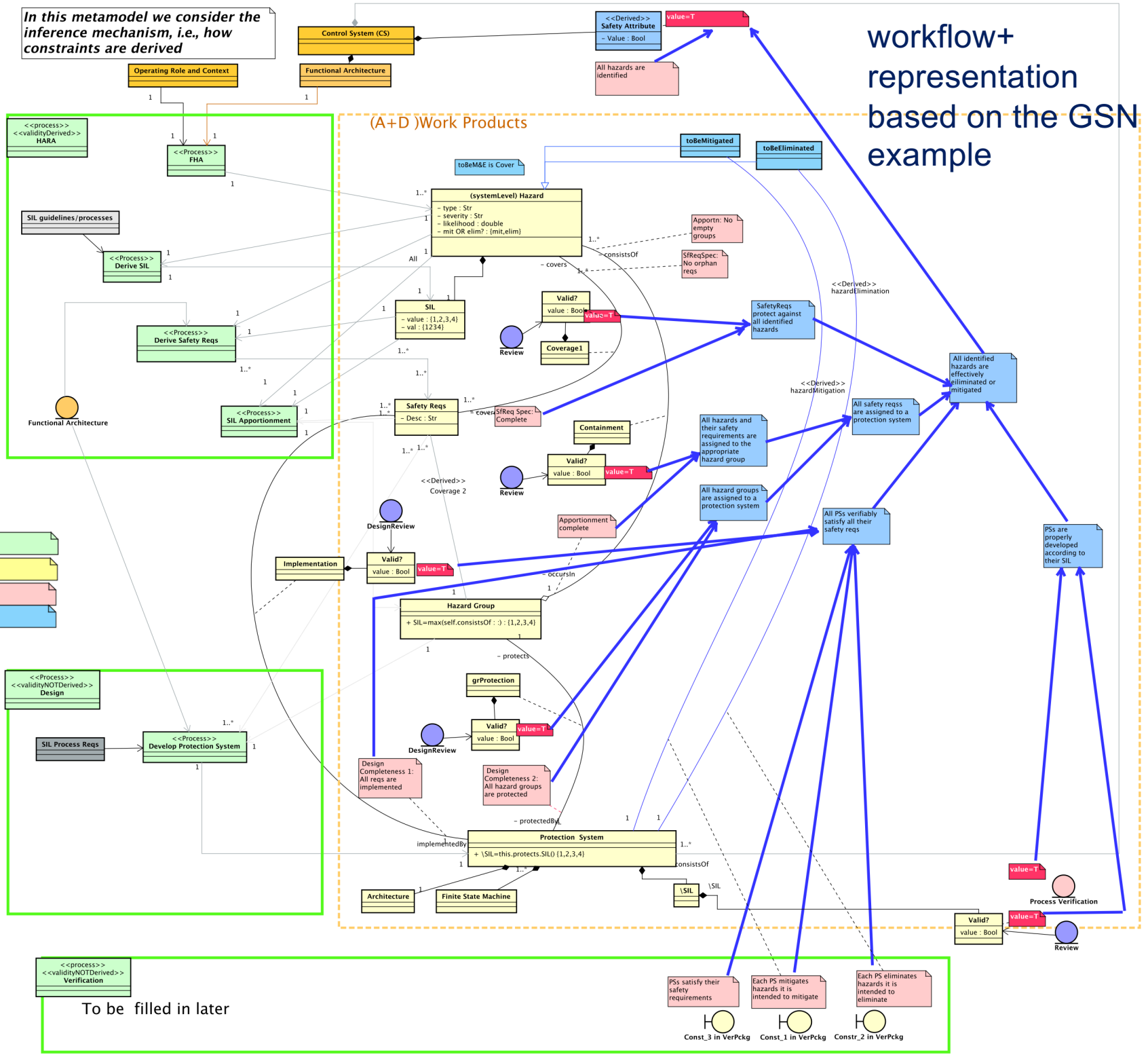
GSN COMMUNITY STANDARD VERSION 1 - 2011



McSCert

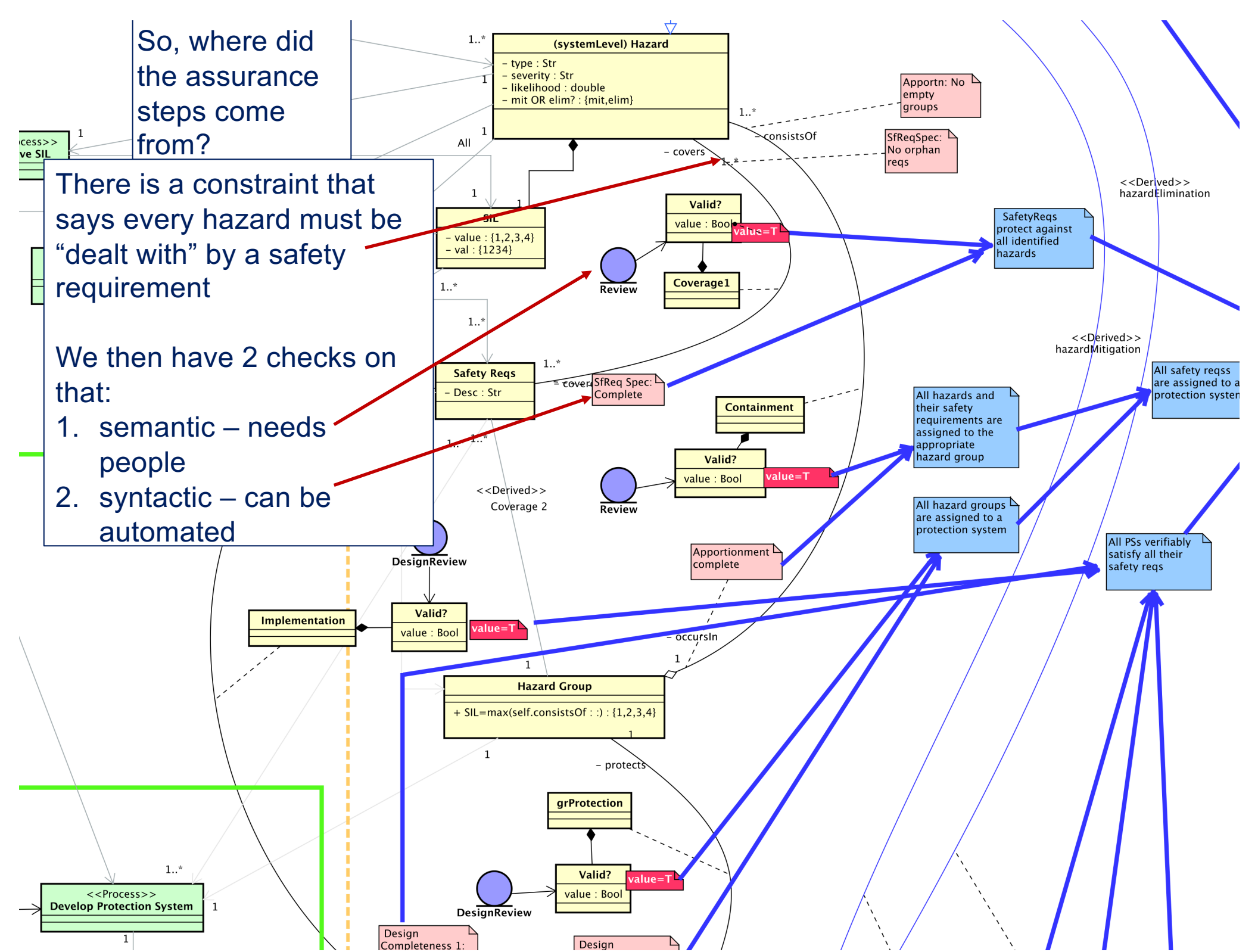
In this metamodel we consider the inference mechanism, i.e., how constraints are derived

workflow+ representation based on the GSN example



Legend

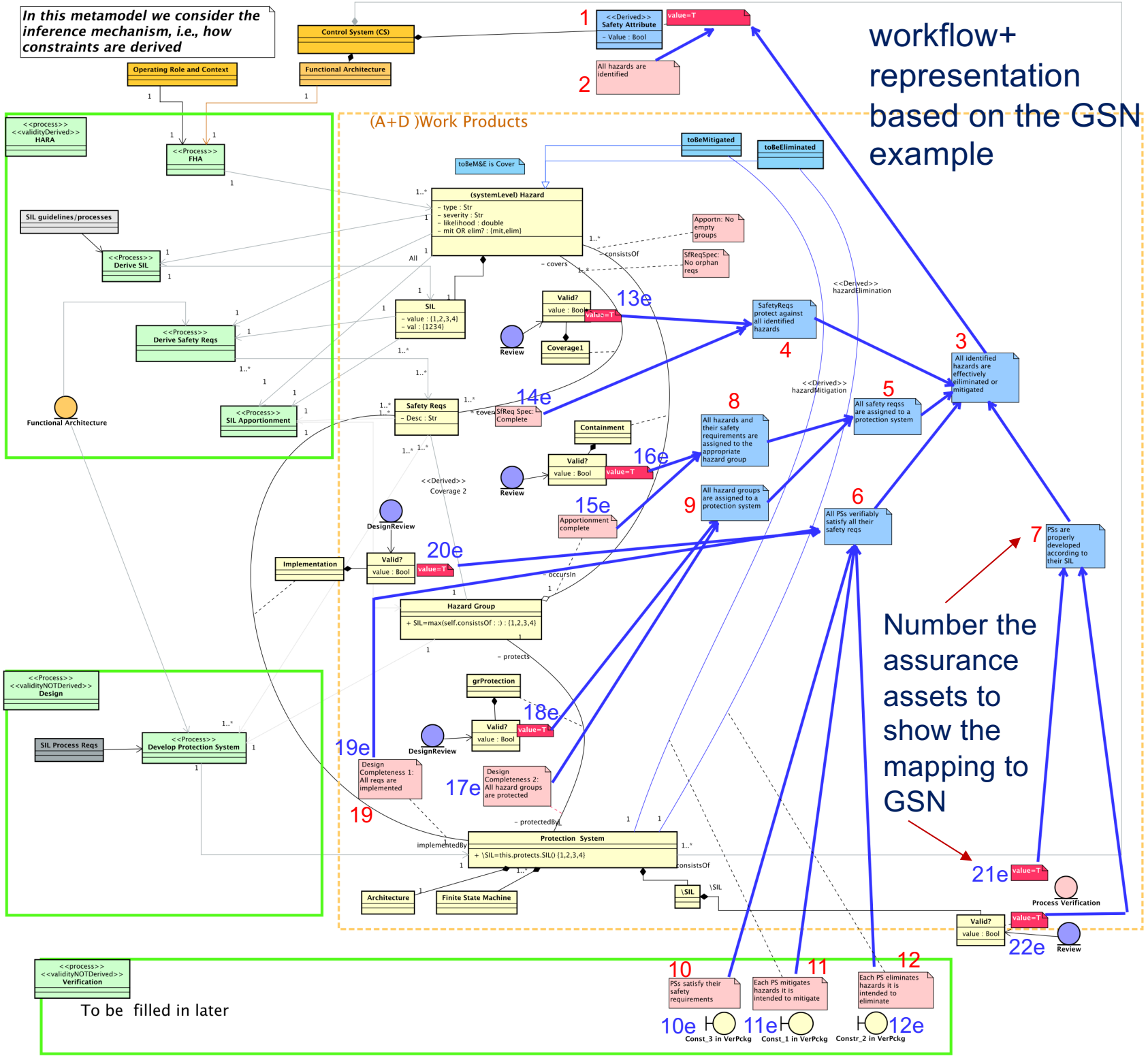
- *lightGreen -- Processes*
- *Yellow--Data*
- *Red--Constraints*
- *Blue--Derived elements*


So, where did the assurance steps come from?
There is a constraint that says every hazard must be "dealt with" by a safety requirement
We then have 2 checks on that:
1. semantic – needs people
2. syntactic – can be automated
(systemLevel) Hazard
- type : Str
- severity : Str
- likelihood : double
- mit OR elim? : {mit,elim}
SIL
- value : {1,2,3,4}
- val : {1234}
Safety Reqs
- Desc : Str
Valid?
value : Bool
value=T
Review
Coverage1
Containment
Implementation
Hazard Group
+ SIL=max(self.consistsOf : :) : {1,2,3,4}
grProtection
DesignReview
<<Process>>
Develop Protection System
- covers
- consistsOf
- occursIn
- protects
All
Apportn: No empty groups
SfReqSpec: No orphan reqs
SfReq Spec: Complete
<<Derived>>
Coverage 2
<<Derived>>
hazardElimination
<<Derived>>
hazardMitigation
SafetyReqs protect against all identified hazards
All hazards and their safety requirements are assigned to the appropriate hazard group
All hazard groups are assigned to a protection system
Apportionment complete
All safety reqss are assigned to a protection syster
All PSs verifiably satisfy all their safety reqs
Design Completeness 1:
Design

McSCert

We have not added all the reviews in the process nor shown how we deal with people in this example

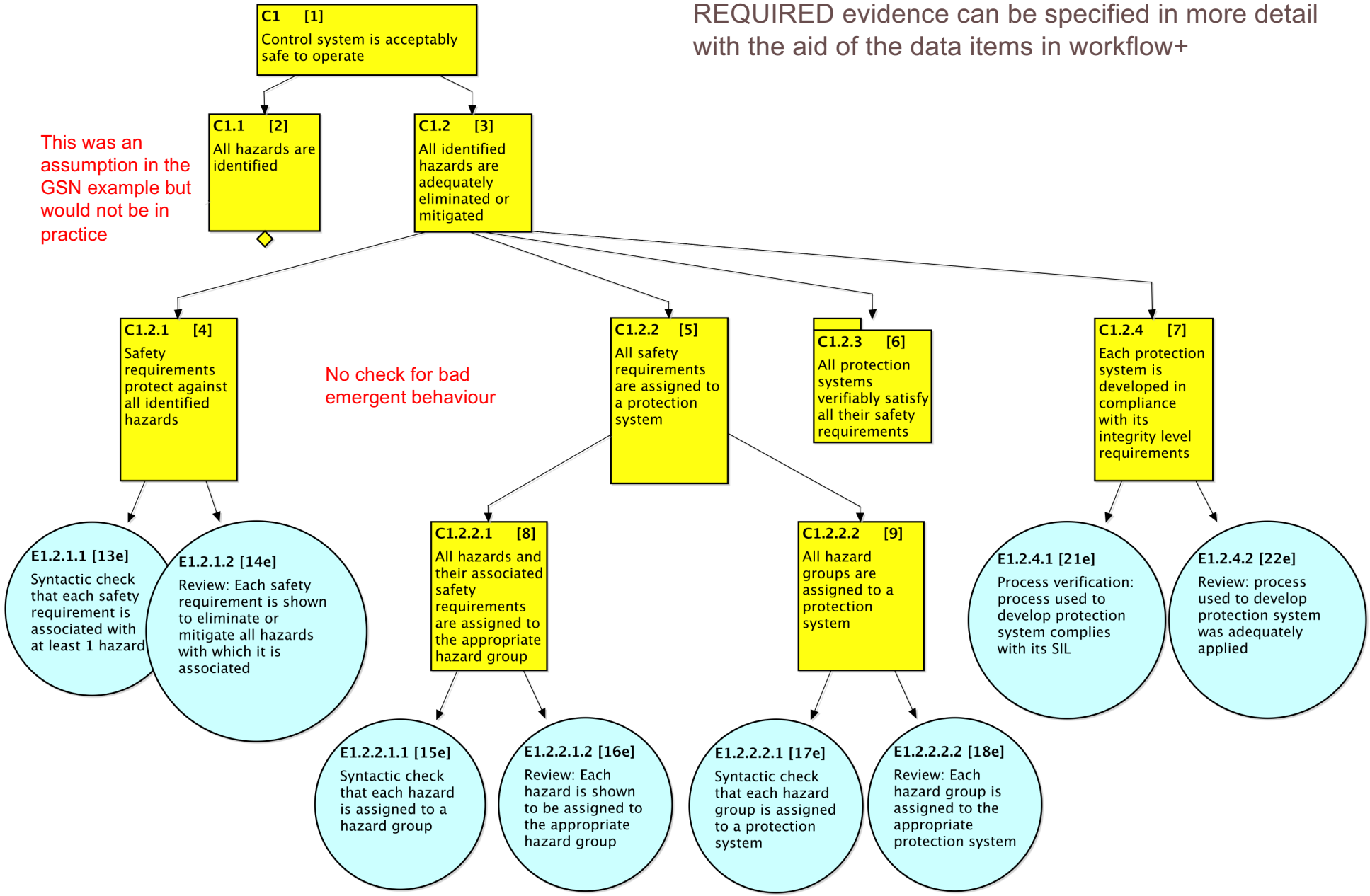
The framework for each of those can be defined using aspects



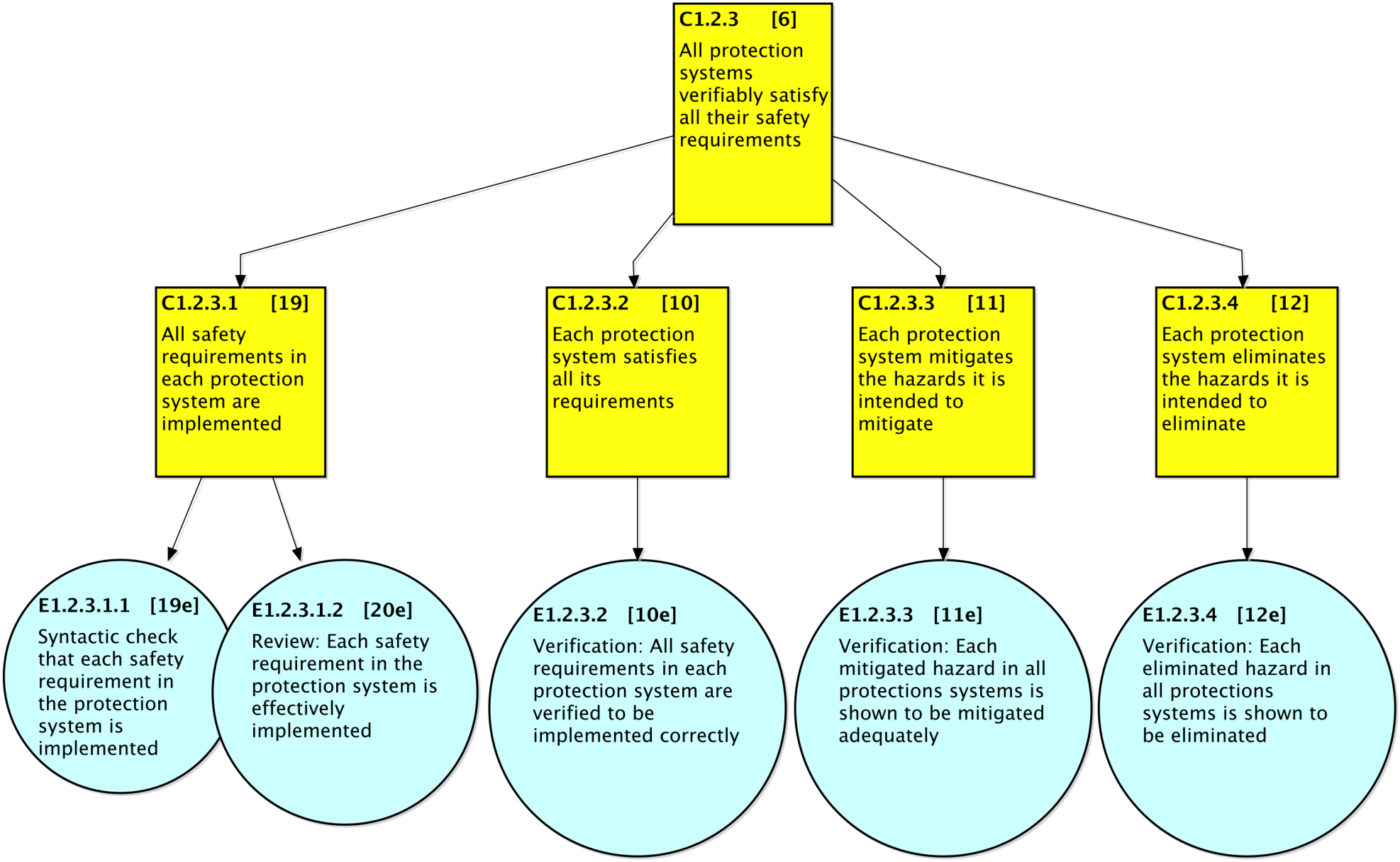
McSCert

Partial GSN model derived from workflow+ model. Showing only claims and evidence so as not to clutter the diagram. Numbers in node title maps to workflow+ in previous slide.

REQUIRED evidence can be specified in more detail with the aid of the data items in workflow+



McSCert

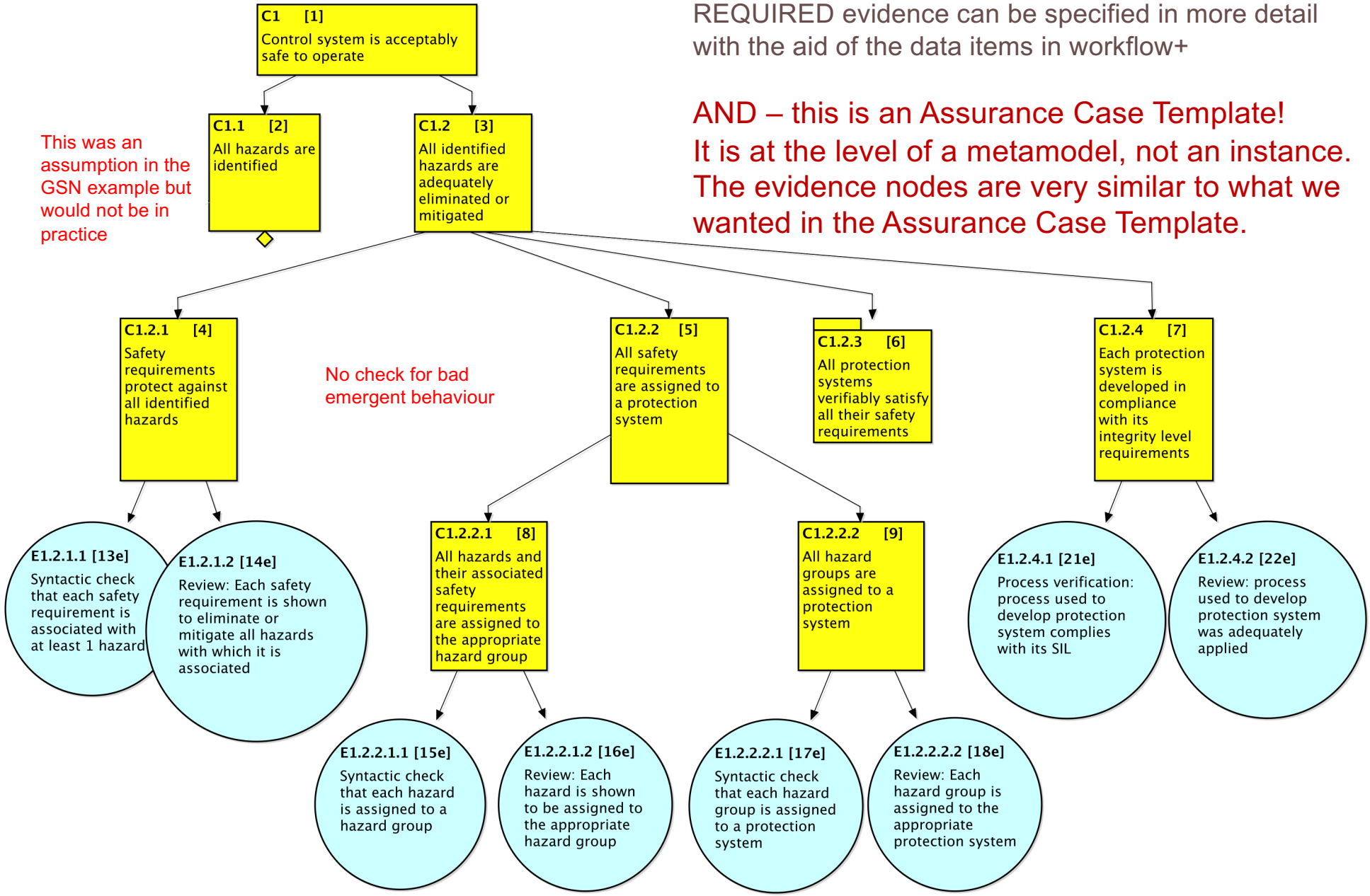

C1.2.3 [6]
All protection systems verifiably satisfy all their safety requirements
C1.2.3.1 [19]
All safety requirements in each protection system are implemented
C1.2.3.2 [10]
Each protection system satisfies all its requirements
C1.2.3.3 [11]
Each protection system mitigates the hazards it is intended to mitigate
C1.2.3.4 [12]
Each protection system eliminates the hazards it is intended to eliminate
E1.2.3.1.1 [19e]
Syntactic check that each safety requirement in the protection system is implemented
E1.2.3.1.2 [20e]
Review: Each safety requirement in the protection system is effectively implemented
E1.2.3.2 [10e]
Verification: All safety requirements in each protection system are verified to be implemented correctly
E1.2.3.3 [11e]
Verification: Each mitigated hazard in all protections systems is shown to be mitigated adequately
E1.2.3.4 [12e]
Verification: Each eliminated hazard in all protections systems is shown to be eliminated

McSCert

Partial GSN model derived from workflow+ model. Showing only claims and evidence so as not to clutter the diagram. Numbers in node title maps to workflow+ in previous slide.

REQUIRED evidence can be specified in more detail with the aid of the data items in workflow+

AND – this is an Assurance Case Template!
It is at the level of a metamodel, not an instance.
The evidence nodes are very similar to what we wanted in the Assurance Case Template.



McSCert

Benefit 1

- Traceability
  - Have a look at the GSN view
  - Now consider traceability – remember our discussion on traceability between artifacts (development, assurance etc) and the problem of granularity of evidence
  - If we look at the workflow+ example, the traceability links are obvious and they link between artifacts in all components of the model(s)
    - They can extend into the environment as well!
  - This will become important when we extend this to incremental assurance

McSCert

Benefit 2

- workflow+ is a metamodel – it is automatically a template for all products produced using the same processes, etc
  - Yes, it is a lot of work, but it is done once used many times. Also, a lot of the work has to be done anyway
  - It is developed before you build the products and thus reduces confirmation-bias
  - The evidence at this level (like in ACTs) describes *what* evidence must be produced for a product, and should be linked to the appropriate data items in the model. This will provide definitive acceptance criteria
  - This approach satisfies both process and product assurance – and the product assurance is much more specific than competing methods (see the previous bullet)

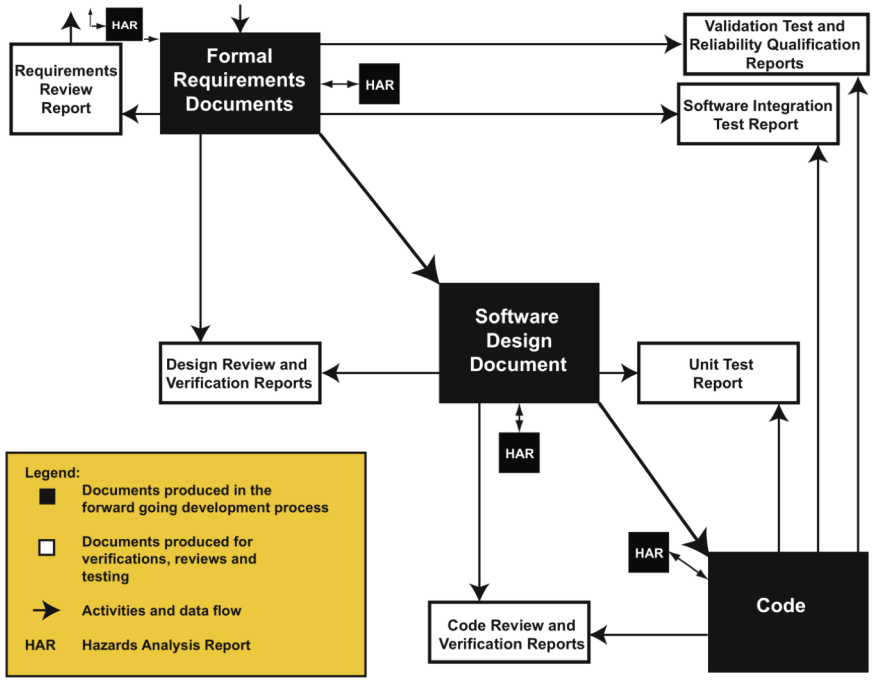
McSCert

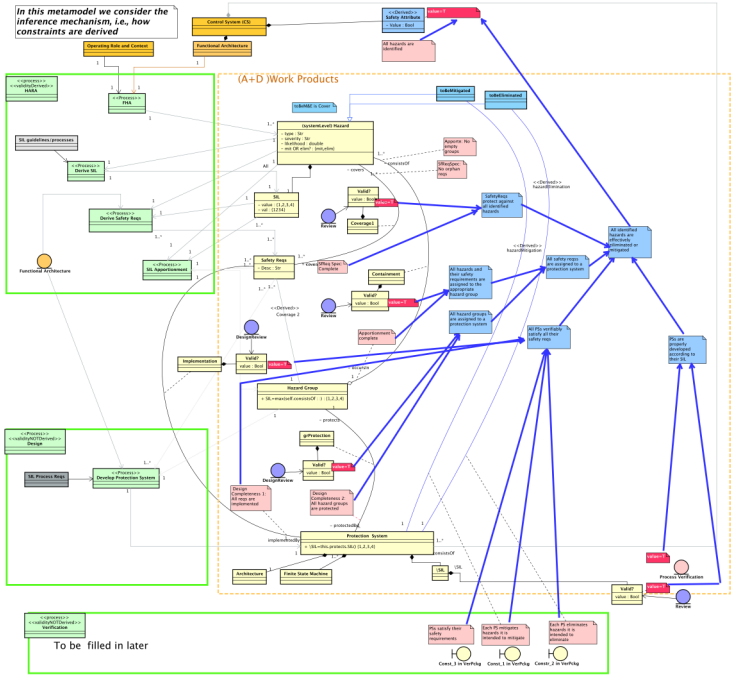
Benefit 3

- With appropriate tooling, workflow+ facilitates many automatic checks – aids in development as well as assurance
- The technology is pretty standard and commercial tool vendors are starting to produce tools based on very similar notations (not for assurance cases etc – I was thinking of tools like Medini Analyze when I wrote that, it aids safety analysis)

Benefit 4

- Compared with something like GSN, workflow+ is much closer to the way in which many companies have built safety-critical systems in the past





McSCert

Benefit 5

- workflow+ facilitates separating process, product and people –intuitively what many people would prefer. Our experience is that this is fundamentally difficult in GSN
  - We often need a combination of these as premises for a claim
  - And need the same premises elsewhere in the GSN graph
  - That means we either have to resort to duplication, or we have links stretching across different slices of the graph

McSCert

Benefit 6

- Current standards such as ISO 26262 can be modeled using workflow+ and could be used as normative models for conformance checks of internal company processes
  - We have experience of constructing UML models of ISO 26262 – before we defined workflow+
  - workflow+ models would be very similar

McSCert

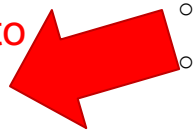
What is missing?

- The explicit demonstration that the system is safe!!!!
- We are working on that
- There are various options
  - Best I think is to do what we have always done – demonstrate that
    - The requirements will result in a safe system
    - The system as implemented complies with its requirements

What is missing?

- The explicit demonstration that the system is safe!!!!
- We are working on that
- There are various options
  - Best I think is to do what we have always done – demonstrate that
    - The requirements will result in a safe system
    - The system as implemented complies with its requirements

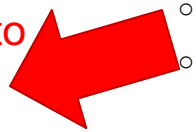
Actually, also show that (all) obstacles to achieving these 2 results have been overcome



What is missing?

- The explicit demonstration that the system is safe!!!!
- We are working on that
- There are various options
  - Best I think is to do what we have always done – demonstrate that
    - The requirements will result in a safe system
    - The system as implemented complies with its requirements
    - These are already process steps so it should not be difficult to augment them appropriately in workflow+

Actually, also show that (all) obstacles to achieving these 2 results have been overcome



McSCert

Thank You!

References

[Chowdhury2017] T. Chowdhury, C.W. Lin, B.G. Kim, M. Lawford, S. Shiraishi, A. Wassyng, "Principles for Systematic Development of an Assurance Case Template from ISO 26262", ISSRE Industry Day, 2017.

[Chowdhury2018] T. Chowdhury, E. Lesiuta, K. Rikley, C-W. Lin, E. Kang, B. Kim, S. Shiraishi, M. Lawford, A. Wassyng. "Safe and Secure Automotive Over-the-Air Updates." SAFECOMP 2018, Springer, 172-187.

[Diskin2018] Zinovy Diskin, Tom Maibaum, Alan Wassyng, Stephen Wynn-Williams, Mark Lawford, "Assurance via model transformations and their hierarchical refinement," MoDELS 2018: 426-436.

[GSN2011] GSN Community, GSN Community Standard, Std., Rev. Ver. 1, 2011. [Online]. Available: http://www.goalstructuringnotation.info/documents/GSN Standard.pdf

[Kelly1998] T. Kelly, "Arguing safety – a systematic approach to managing safety cases," Ph.D. dissertation, University of York, September 1998.

[SACM2.0] Obtainable from: https://www.omg.org/spec/SACM/About-SACM/

[Wassyng2011] A. Wassyng, T. Maibaum, M. Lawford, and H. Bherer, "Software certification: Is there a case against safety cases?" in Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems, ser. Lecture Notes in Computer Science, R. Calinescu and E. Jackson, Eds. Springer Berlin Heidelberg, 2011, vol. 6662, pp. 206–227.

[Wassyng2016] A. Wassyng, P. Joannou, M. Lawford, T. Maibaum, N.K. Singh, "New Standards for Trustworthy Cyber-Physical Systems." A. Romanovsky, F. Ishikawa, Trustworthy Cyber-Physical Systems Engineering, CRC Press, 2016, 341-371.