Sûreté de fonctionnement : point de vue du régulateur sur les concepts, modèles et méthodes de démonstration

P.-J. Courtois
AV Nuclear, Authorized Nuclear Inspection and Licensing Body
Departement Ingenierie Informatique, UCL, Louvain-la-Neuve, B

courtois@info.ucl.ac.be

Atelier thématique

Justification de sûreté de fonctionnement (*safety case*) : approches industrielles, méthodes de construction et structures

Mardi 18 mars 2003 – LAAS-CNRS

Issues

To elicit and to justify a variety of claims (french: revendications, assertions) on the dependability of the system , on its design, on its operations.

To cope with different sources of evidence (french: _l_ments probants) to justify these claims: tests, static analyses, operational feedback, system architecture, procedures, development process, experience

To choose between different justification approaches; acceptance conditioned on either:

- Regulatory reviews (project, process, product)
- Compliance with a set of predefined rules, standards, design criteria,
- Specific Evidence (e.g. multi-legged or other bottom-up approach)
- Acceptable level of residual risk (per function class)
- Achievement of predefined safety goals, properties, design criteria,
- Belief-, influence- netsÅ

- **Or any empirical combination of these rule _ based classical approachesÅ**

Rule- versus claim- oriented justification

- Rule based approach weaknesses:
  - fail to demonstrate safety properties;
  - collect evidence in orthogonal, unrelated directions;
  - lack generality;
  - fail to cover the wide gap between the safety requirements at application level and the lower levels of the implementation;
  - can be expensive in terms of lengthy negotiations and interpretations, failing to demonstrate how much safe is safe enough.

→ claim _ oriented approaches

Cost Effectiveness

Costs of the safety justification supplied in support of a license application can be reduced in different ways:

By restricting attention to safety issues and claims dictated by the new application, or by the replacement/upgrade and/or by new technology, avoiding endless rambling about rules/standards not necessarily appropriate to the specificities of a project;

By limiting the required evidence to what is arguably necessary and sufficient;

By allowing assessors and regulators to identify the most critical evidence;

Cost-Effectiveness (2)

Under given conditions: by re-use of sub-claims, arguments and supporting evidence, possibly from other safety justifications;

By the capability to transform some non-functional claims into functional ones;

When safety case and design are in phase: by allowing designers/ developers to evaluate at application level the risks associated with design decisions before actually implementing them;

By making (the indispensable) consensus amongst stakeholders easier to reach.

Å

What is needed?

A unifying and pragmatic structure (a method, a discipline?) to support an effective justification of safety, with:

- A prescriptive part:
  - To determine a set of initial claims;
  - To organize sub-claims, evidence, and arguments so as to make the system safety justification easier to construct, to communicate, to assess and to agree upon.
- A descriptive part:
  - To identify models, representations and interpretations of the system that are required at different levels to formulate claims, evidence and arguments.

Differences between claims and requirements

- **A claim is a statement (true or false) claiming a functionality or a property of the specifications or of the behaviour of a system important to safety (SIS) in view of a particular application.**

- A claim can be identical to a system requirement specification (in particular when the safety justification is part of the development), but most often a claim is either :

  an Å*extra-requirement*Å addressing a need that was not part of the original specifications; e.g. components, COTÅs, PAMS functions;

  or a Å*meta-requirement*Å when claiming a property that a set of system requirement specifications must enjoy (completeness, soundness,Å).

- Requirements address the design and emphasize functionality and system properties; claims address a specific use and need also to cover threats and hazards.

How to Start a Safety Justification?

Safety Justification = the set of detailed arguments and evidence elements which support a selected set of initial claims on the dependability of the operations of a SIS in a given environment.

Examples of initial dependability claims (level 0):

pfd.clm0 **: The reliability and availability of the new digital Instrumentation System is at least that of the system being replaced.**

no_rot.clm0 **: Rotation of machine carousel is impossible while nuclear fuel material is being transferred.**

The set of level 0 initial dependability claims, approved as being complete, consistent, and sound defines the scope of the safety justification, (i.e. what has to be justified).

A key observation: Sub-Claim Inductive Structure

A claim that a SIS prevents or mitigates a hazard at the application level

implies the subclaims that:

iff

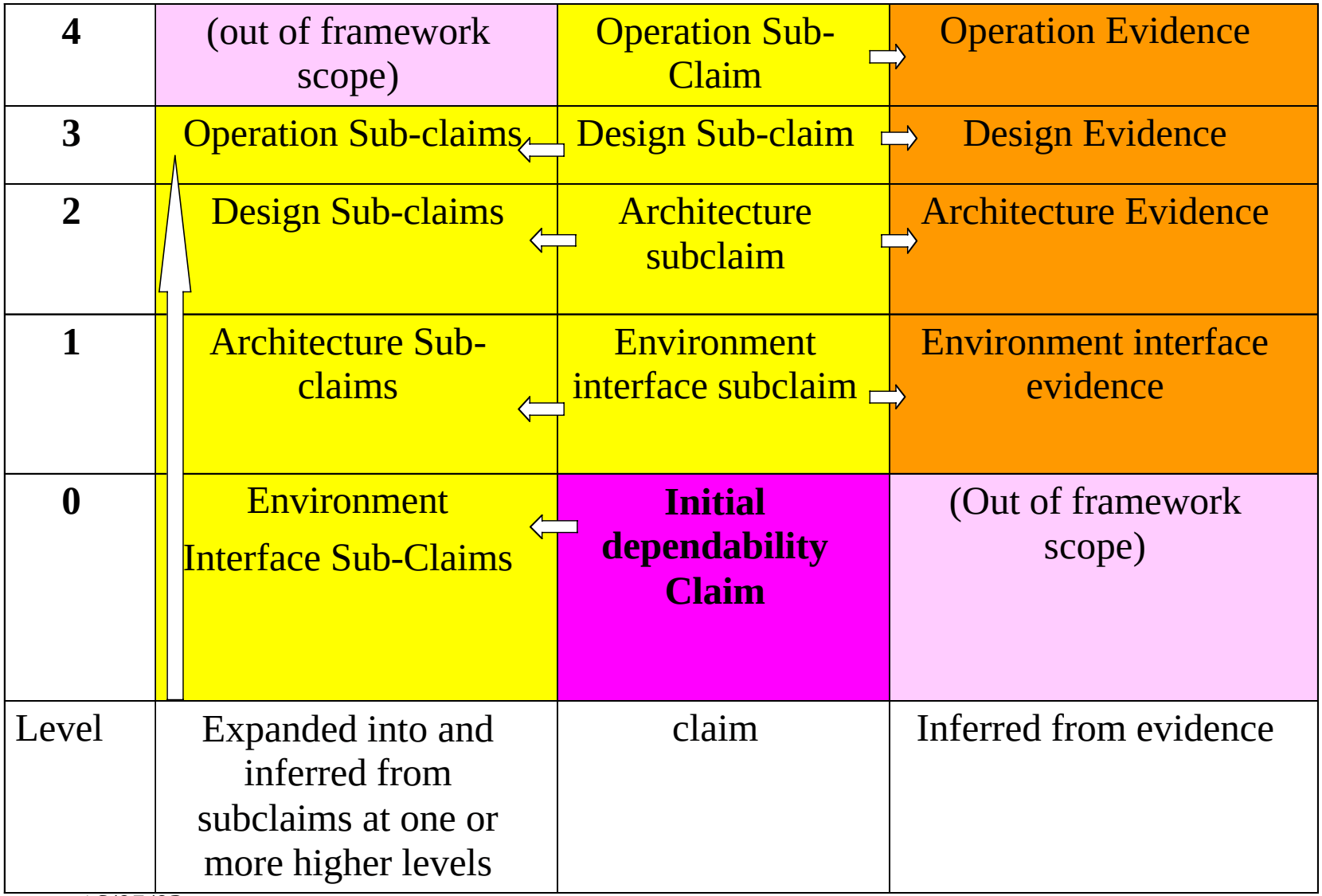
- **the environment _ system interface specifications of how the SIS deal with this hazard are valid (sound, consistent, complete);**
- **the SIS H/S architecture and design adequately implement these specifications;**
- **operations and maintenance maintain the integrity of this implementation.**

Safety Justification Inductive Claim Structure

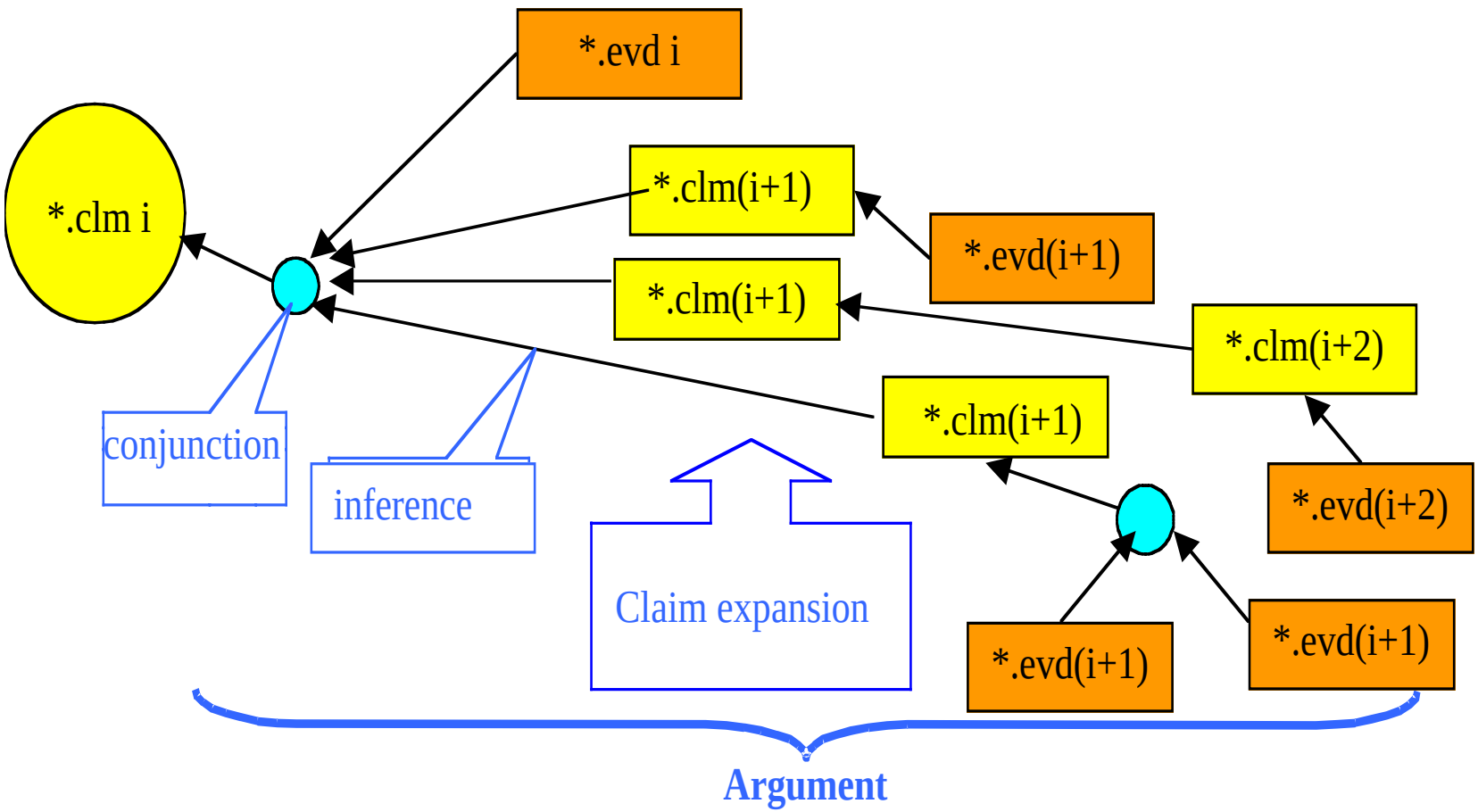
Every *initial claim* implies sub-claims formulated and NESTED at one or more of four levels:

1. Environment Interface *(system functions and properties expected by the environment)*: *sub-claims* on the validity of the system specifications to satisfy the initial claim;
2. Architecture*: sub-claims* that the architecture satisfy the system specifications;
3. Design: *sub-claims* that the system is designed and implemented so as to perform according to the system specifications;
4. Operations *(system response to its environment)*: *sub-claims* that the initial claim will remain satisfied during operations. Includes sub-claims on maintenance, recalibrations, operator procedures and behaviour, alarms, ….

Safety Justification Inductive Structure



Argument


*.evd i
*.clm i
*.clm(i+1)
*.evd(i+1)
*.clm(i+1)
*.clm(i+2)
*.clm(i+1)
conjunction
inference
Claim expansion
*.evd(i+2)
*.evd(i+1)
*.evd(i+1)
Argument

Claim Assertion Atomicity

For simplicity of arguments and re-usability, a claim assertion should be as simple (atomic) as possible:

whenever possible, not a conjunction of assertions that could be justified separately, and used independently in different arguments.

Example:

Instead of a single claim:

watchdog.clm2: (system is in initstate and watchdog is disarmed
OR (system is not in initstate and watchdog is armed)

make two claims:

watchdog_disarmed.clm2: system in initstate ⊂ watchdog disarmed

watchdog_armed.clm2: system not in initstate ⊂ watchdog armed

Claim Quantifiers

Completeness usually is a difficult property to establish.

Quantifiers such as “all”, “always”, “never” should therefore be “avoided” in claim assertions. Claims should be as specific as possible, i.e. address properties of finite sets.

Ex: Instead of

testplan_x.clm3: the test set x is exhaustive

make two claims:

testplan_x.clm3: test set xx covers set yy of failure modes

fmeca_y_clm3: yy is the set of unsafe failure modes anticipated by software fmeca zz

CONJUNCTION PROPERTY:

A *(sub)claim argument* is a "backward proof"similar to proofs formalized in sequent calculus.

= a tree of implications, whose root is the (sub)claim to be proved, whose intermediary nodes are sub-claims, and whose end nodes are evidence components:

$$\alpha.\text{clmi} \leftarrow (\upsilon.\text{evd}[i] \wedge \ldots \omega.\text{evd}[i]) \wedge (\beta.\text{clm}.[j] \wedge \gamma.\text{clm}[k] \wedge \ldots)$$

where levels j, k,…> i.

Single Argument Property

A (sub-) claim is supported by one argument only (possibly multi-legged). If the (sub-) claim appears in more than one expansion, it must be supported with the same argument. This univocal property is a necessary condition for the consistency of the safety justification, and for allowing sub-claims to be reused with their arguments.

Argument Termination Property

Arguments must be finite and evidence components at a given level are the terminal nodes in the backward proof trees of one or more (sub) claims at that or at the lower levels. If the plausibility of an evidence component remains questionable, it means that *additional arguments are needed to assert this plausibility and that sub-claims must be added and justified to assert the correctness, the completeness or other attributes of this evidence* .

Causal Implication

The inference mechanism is not the (so-called material) implication of first order logic, an implication which is true if the antecedent is false, whatever is the value of the consequent.

In a safety justification, a relevant "causal connection" is needed between the antecedent and the consequent. Relevance and the relevance implication modal operator might be more appropriate to formalize the type of implication used in "*claim proving*".

Justifiability of Claims (1)

- **A claim is *justified* if and only if there exists a logically valid argument based on *justified* sub-claims and on *plausible* evidence.**
- **Unrealistic (e.g. on completeness) or unsound claims are not *justifiable*, the former because they lack plausible evidence, the latter a valid argument.**
- **Realistic and sound claims, however, may not be justifiable, for lack of plausible evidence: additional evidence is then needed, or additional sub-claims to claim plausibility.**

Consensus (2)

- Justifiability, and in particular plausibility of evidence is unavoidably based on the existence of a consensus between stakeholders. This consensus is required for the acceptance of:
  - ✓ Model assumptions
  - ✓ Correctness of arguments
  - ✓ Plausibility of evidence
- All proofs, even those of theorems and computer programs rest on some sort of social consensus (Lipton, Perlis, DeMillo, 1977)
- Framework should make the reaching of this consensus easier and more cost-effective

Axiomatic Approach and Uncertainties

- Two different viewpoints must be reconciled:
  - Justification of safety must be as rationale as possible;
  - Unresolvable uncertainties must be dealt with explicitly;
- Two main sources of uncertainties:
  - impossibility of ruling out all possible undesirable events;
  - impossibility of ruling out faws in the consensus (on correctness of assumptions, models, arguments, plausibility o f evidence)
- Axiomatisation and uncertainties are reconciled by the following principles, which comply with safety:
  - Arguments are always deterministic and based on explicit assumptions accepted by consensus.
  - Uncertainties are minimised (by additional (probabilistic)claims, assumptions, models which assert confidence);
  - Remaining uncertainties are made explicit, stated and accepted by consensus.

Risk minimisation and safety justification

- Prevention
  - Anticipation of accidents and of mitigating features
  - Uncertainty is not considered (at best relegated to outside design basis accidents)
  - Instrumental to identify initial claims, PIE's and to construct expansions
- Precaution
  - Uncertainty is accounted for
  - Results from deficit of evidence plausibility, not from detreministic expansions
  - Residual risk determined by claim expansions insufficiently justified or incomplete
  - Framework should maximize plausibility and expansion completeness
- Enlightened Catastrophism
  - Accidents are considered certain
  - Instrumental to identify initial claims, PIE's and to construct expansions

Models

- **A (sub)claim is formulated in some language as an assertion which takes the value true or false.**

- **A language and a model are needed to interpret the predicate symbols, the functions, variables, and constants that are used to formulate the claim, as well as the supporting argument and evidence.**

Models (2)

- Four models are needed:
  - a model of the system-environment interface
  - a model of the system architecture
  - a model of the hardware and software design/implementation;
  - a model of the system modes of use, operations and maintenance.

- The safety justification is a logical analysis based on these four very different models and on their interrelations.

Proof Obligations and Model Preservation Relations (3)

At each level i, each subclaim must be inferred from level i evidence and/or from an expansion of subclaims of the next levels (i + 1, i + 2,…).

A subclaim at one level is thus taken as evidence at that level and supported by evidence at some upper level.

For this "demonstration" to be "right" - i. e. sound, consistent, complete – relations must exist between the models used at the different levels.

Misconceptions

Claim expansion is different from – and more difficult than - the stepwise refinement of specifications (structured programming).

Claim expansion - Misconceptions

Claim expansion is harder than specification stepwise refinement:

- The real universe of a sub claim is not included in the universe of the claim from which it is expanded, and does not include the universes of the sub claims into which it is expanded (unlike a hierarchy of layers or abstract machines);
- The sub claims of a given expansion are not mutually independent (unlike the nested functions of a compound function in structured programming);
- The four levels are fixed (unlike refinement levels) and quite apart from each other;
- Claims may have to be expanded into sub claims at all higher levels, and not of the next level only.

Main features of a safety justification framework (1)

- A prescriptive and a descriptive part;
- An inductive structure for the organisation of claims, evidence and arguments.
- A concept/mechanism for claim expansion
- A transformation of non-functional claims into functional ones;
- Properties of evidence: weight and plausibility;

Main features of a safety justification framework (2)

- Identification of model structures, and languages needed for the semantic, syntactic and logical treatment of a claim, and for the valuation of evidence.
- Distinction between assumptions and evidence;
- Preservation relations across model structures.
- “Guarantee” and “Rely” conditions between system and components independently developed and/or validated.
- Seeks consensus by rational arguments to back up and to limit subjective expert judgement.

Other Benefits

- Impact on design.
- Evolvable dependability case.
- Conditions for characterising components in view of their encapsulation and of re-use of evidence and sub-claims.
- More flexible and general than rule-based approach.

FAQ’s and Final Remarks

- Process versus Product
  - Evidence versus Claims
- Deterministic versus Probabilistic Analysis
  - Deterministic arguments
  - Deterministic/probabilistic claims
  - Deterministic/statistical evidence/evidence plausibility

- Safety Justification versus Safety Case

  Technical /Factual documented Justification

  versus

  Legal, social , consensual aspects .

Safety Justification Framework

“Computation = Logic + Control"
(Kowalski and Hayes in the 1970s)

“Safety Justification = Logic + Models ” ?