2020 International Software Safety Conference

Software Safety Demonstration: Techniques and Case studies

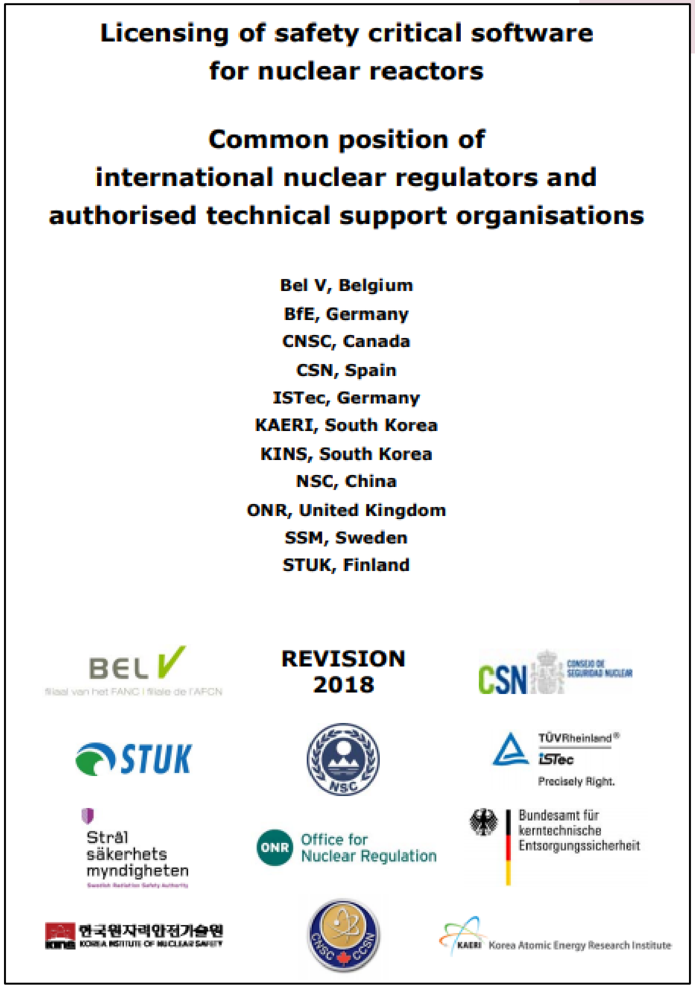
2020. 12. 3.

Eunkyoung Jee
ekjee@se.kaist.ac.kr
School of Computing
KAIST

KAIST Korea Advanced Institute of Science and Technology

Safety Demonstration

- *"The set of arguments and evidence elements which support a selected set of claims on the **safety** of the operation of a **system important to safety** used in a given plant environment."*



Licensing of safety critical software for nuclear reactors

Common position of international nuclear regulators and authorised technical support organisations

Bel V, Belgium
BfE, Germany
CNSC, Canada
CSN, Spain
ISTec, Germany
KAERI, South Korea
KINS, South Korea
NSC, China
ONR, United Kingdom
SSM, Sweden
STUK, Finland

REVISION 2018

Approaches to Establish Confidence in Systems

- Standards-Based Approach
  - Examples: DO-178C for avionics safety; Common Criteria for security
  - Development processes are evaluated against a standard
    - Adherence to good development processes is evidence of ability to produce good products
    - Product X has been developed using good development practices
    - Therefore Product X is sufficiently safe, secure, reliable, etc.
- Product-Based Approach → "assurance case" approach
  - Example: safety case in UK
  - Developer creates an assurance case with
    - Explicit claims about system behavior
    - Supporting evidence for claims
    - Arguments linking evidence to the claims
  - The case is evaluated by independent assessors

Assurance Cases

- *"A reasoned and compelling* ***argument****, supported by a body of* ***evidence****, that a system, service or organization will operate as intended for a defined application in a defined environment."*

  - GSN Standard 2011

- Often with a particular focus
  - Safety
  - Security
  - Dependability
  - Trust …

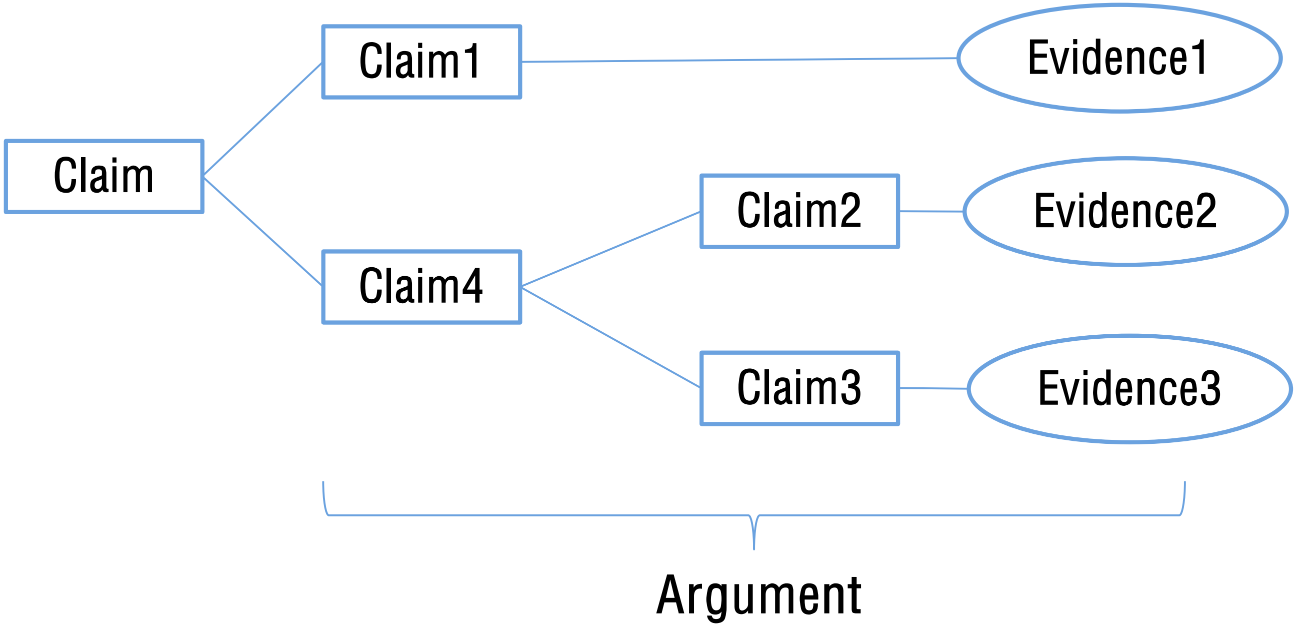
Assurance Cases: Increasing Importance

- Software for Dependable Systems: Sufficient Evidence?
  - *Daniel Jackson, Martyn Thomas, and National Research Council. 2007. Software for Dependable Systems: Sufficient Evidence? National Academy Press, USA.*
  - *"recommended approach ▸* ***dependability case*** *based on explicit claims, evidence, expertise"*
- ISO/IEC 15026-2:2011 Systems and software engineering — Systems and software assurance — Part 2: Assurance case
- U.S. FDA's Infusion Pumps Total Product Life Cycle: Guidance for Industry and FDA Staff (2014)
  - *"In determining whether your new, changed, or modified infusion pump is substantially equivalent, FDA recommends that you submit your information through a framework known as a* ***safety assurance case****."*
- ISO26262 has an explicit requirement for the safety case:
  - *"6.4.6.2 The* ***safety case*** *should progressively compile the "work products" that are generated during the safety lifecycle."*

  within Part 2 – Management of Functional Safety

Assurance Case

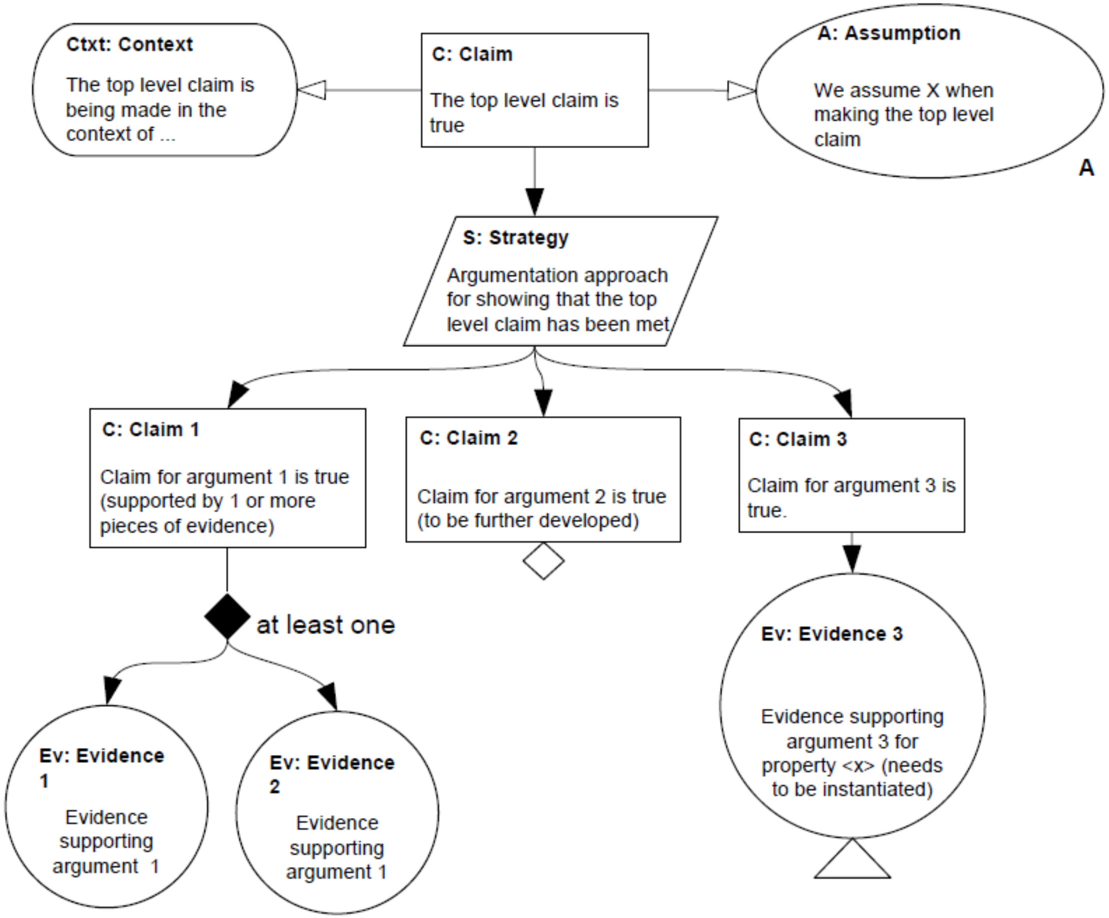
- A structured demonstration that a system is acceptably safe, secure, reliable, etc.
  - A comprehensive presentation of evidence linked (by argument) to a claim



Claim, Argument, and Evidence

- An assurance case requires claims, evidence, and an argument linking evidence to claims:
  - Claim
    - E.g., "The contributions made by the BSCU software to S18 WBS hazards are acceptable."
  - Argument
    - Usually by demonstrating compliance with requirements, sufficient mitigation of hazards, avoidance of hazards, etc.
      - Hazardous software contributions have been identified
      - Controls have been put in place to manage these contributions
      - Mechanisms are in place to monitor the performance of the controls and the system on an on-going basis
  - Evidence
    - E.g., tests, analyses, reviews, simulation, expert judgements and compliance with best practice

Goal Structuring Notation



Ctxt: Context

The top level claim is being made in the context of ...

C: Claim

The top level claim is true

A: Assumption

We assume X when making the top level claim

A

S: Strategy

Argumentation approach for showing that the top level claim has been met

C: Claim 1

Claim for argument 1 is true (supported by 1 or more pieces of evidence)

C: Claim 2

Claim for argument 2 is true (to be further developed)

C: Claim 3

Claim for argument 3 is true.

at least one

Ev: Evidence 1

Evidence supporting argument 1

Ev: Evidence 2

Evidence supporting argument 1

Ev: Evidence 3

Evidence supporting argument 3 for property <x> (needs to be instantiated)

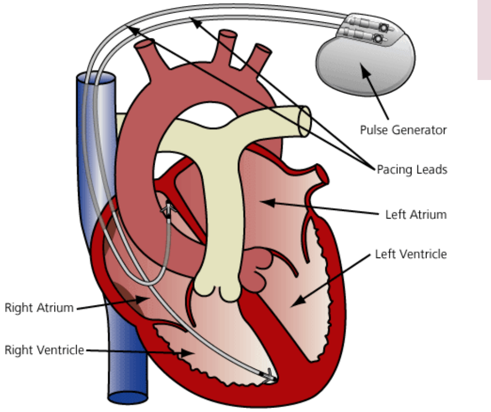
A notation for organizing and structuring assurance cases in a readily reviewable form

- ▭ - Claim
- ⬭ - Context
- ▱ - Strategy
- ○ - Solution/Evidence
- ⬭ᴬ - Assumption
- ➞ - Solved by
- ◇ - Requires further development
- △ - Requires instantiation
- ⇾ - In the context of

*The picture was taken from C.B. Weinstock, J.B. Goodenough, “Towards an Assurance Case Practice for Medical Devices”, SEI TECHNICAL NOTE CMU/SEI-2009-TN-018, 2009.

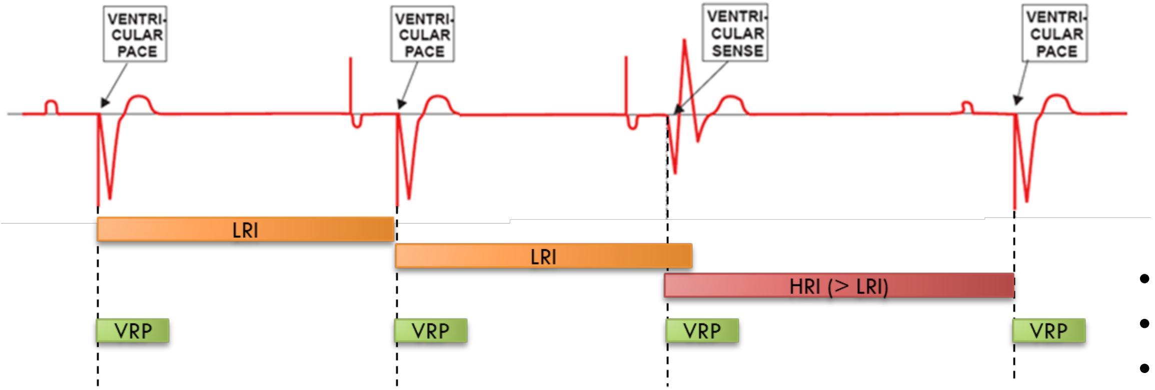
Case Study 1: Pacemaker

- Electronic device implanted in the body to regulate the heart beat
  - A life-critical real-time embedded system
- Two basic functions
  - Pace
  - Sense intrinsic rhythm and inhibit



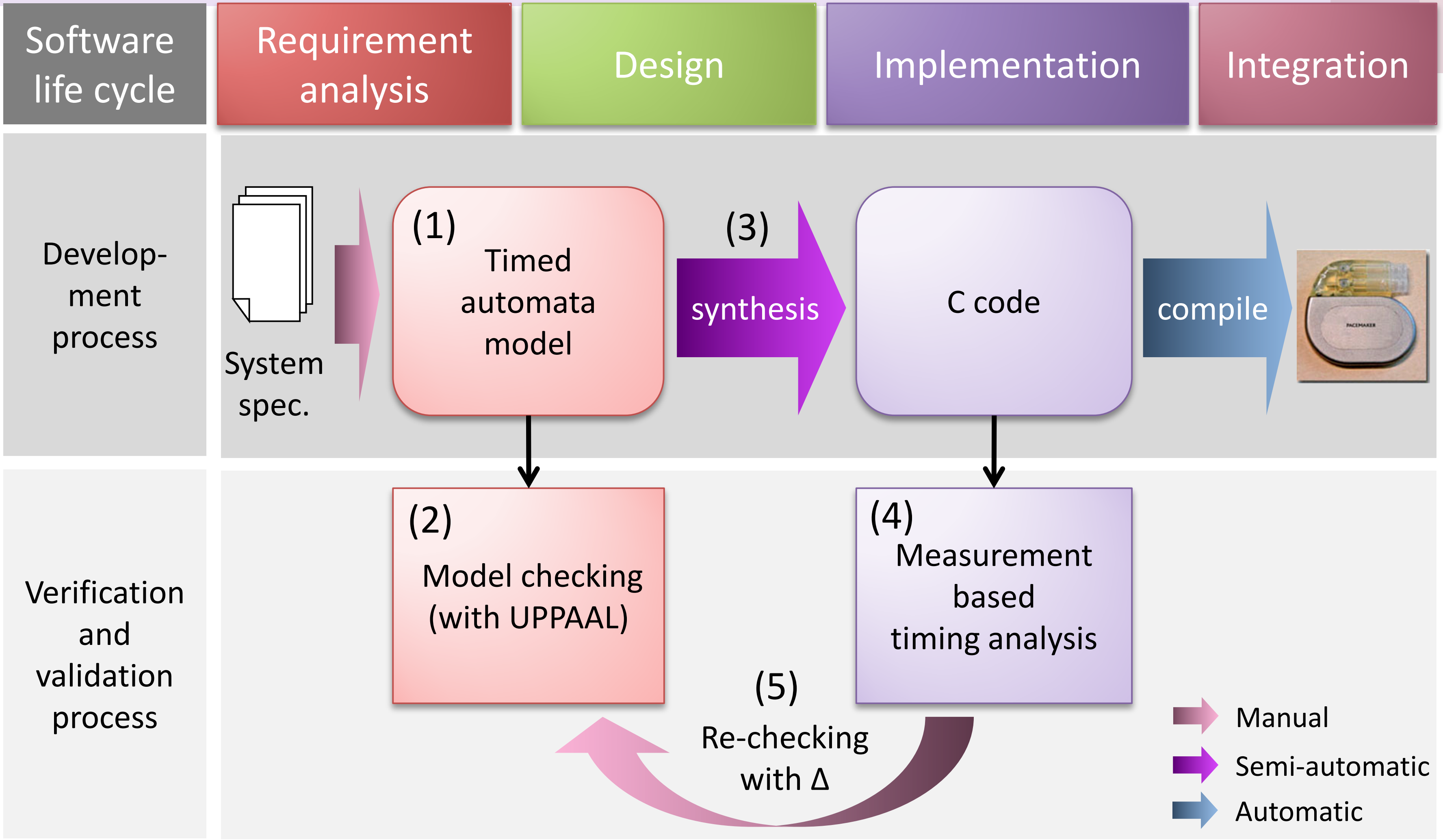
img src: http://www.odec.ca/projects/2007/torr7m2/images/pacemaker.gif

- Fundamental timing cycles of VVI mode (simplest mode)



- LRI: Lower Rate Interval (e.g., 1000ms)
- HRI: Hysteresis Rate Interval (e.g., 1200ms)
- VRP: Ventricular Refractory Period (e.g., 320ms)

A Safety-Assured Development



Software life cycle

Requirement analysis

Design

Implementation

Integration

Develop-ment process

System spec.

(1) Timed automata model

(3) synthesis

C code

compile

Verification and validation process

(2) Model checking (with UPPAAL)

(4) Measurement based timing analysis

(5) Re-checking with Δ

Manual

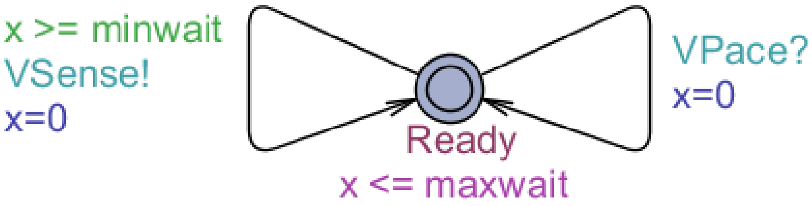
Semi-automatic

Automatic

Formal Modeling in UPPAAL

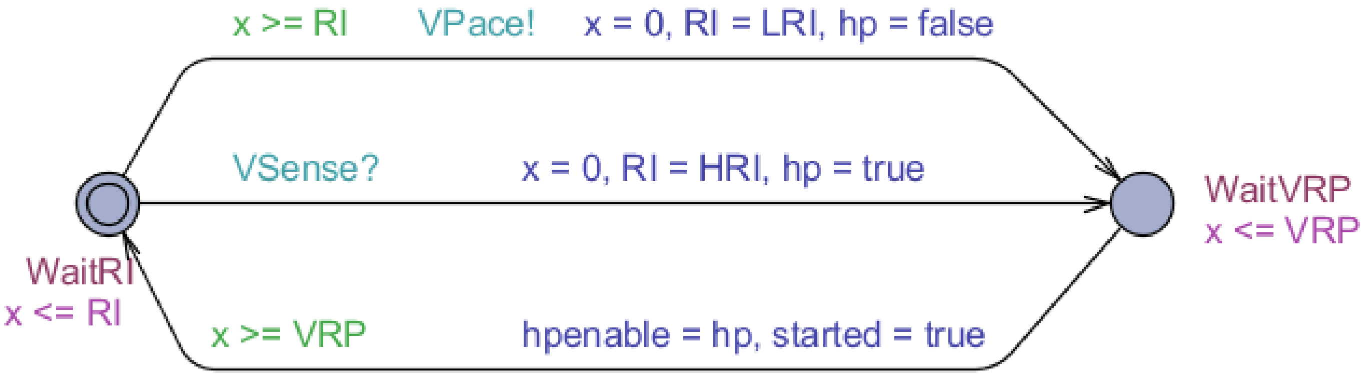
Pacemaker on VVI mode

Heart



Ventricular

controller

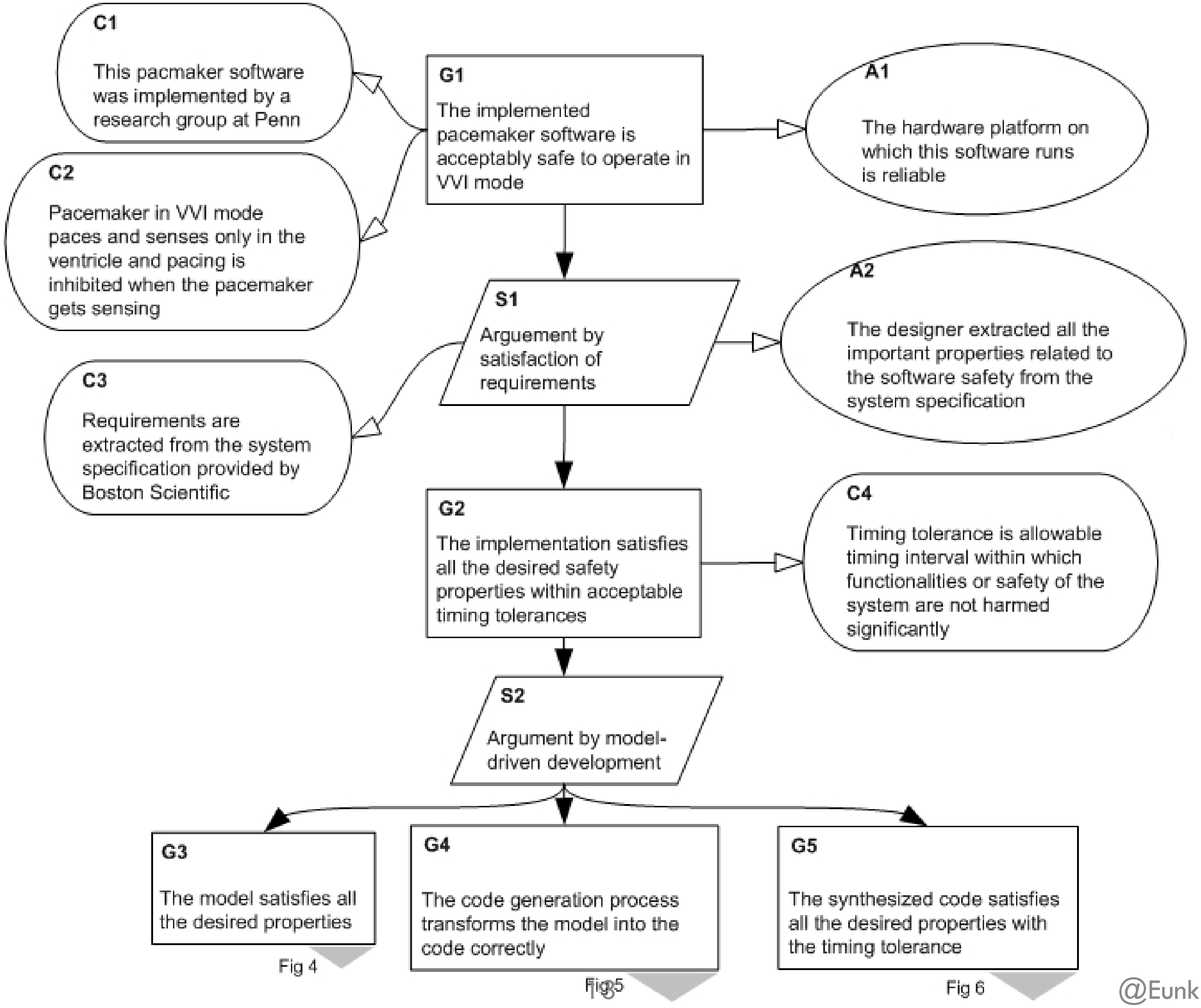

x >= RI
VPace!
x = 0, RI = LRI, hp = false
VSense?
x = 0, RI = HRI, hp = true
WaitVRP
x <= VRP
WaitRI
x <= RI
x >= VRP
hpenable = hp, started = true

Assurance of Model-Driven Development

- Model-driven development:
  - Formal modeling and verification
  - Synthesis of code from models
  - Testing
- Each step adds rigor to some aspect of system development
- How do these steps tie together and are they sufficient?

- Details of the development process:
  - E. Jee, S. Wang, J. K. Kim, J. Lee, O. Sokolsky, I. Lee, A Safety-Assured Development Approach for Real-Time Software, RTCSA, August 2010.

Top Level Claims



C1

This pacmaker software was implemented by a research group at Penn

C2

Pacemaker in VVI mode paces and senses only in the ventricle and pacing is inhibited when the pacemaker gets sensing

G1

The implemented pacemaker software is acceptably safe to operate in VVI mode

A1

The hardware platform on which this software runs is reliable

S1

Arguement by satisfaction of requirements

C3

Requirements are extracted from the system specification provided by Boston Scientific

A2

The designer extracted all the important properties related to the software safety from the system specification

G2

The implementation satisfies all the desired safety properties within acceptable timing tolerances

C4

Timing tolerance is allowable timing interval within which functionalities or safety of the system are not harmed significantly

S2

Argument by model-driven development

G3

The model satisfies all the desired properties

Fig 4

G4

The code generation process transforms the model into the code correctly

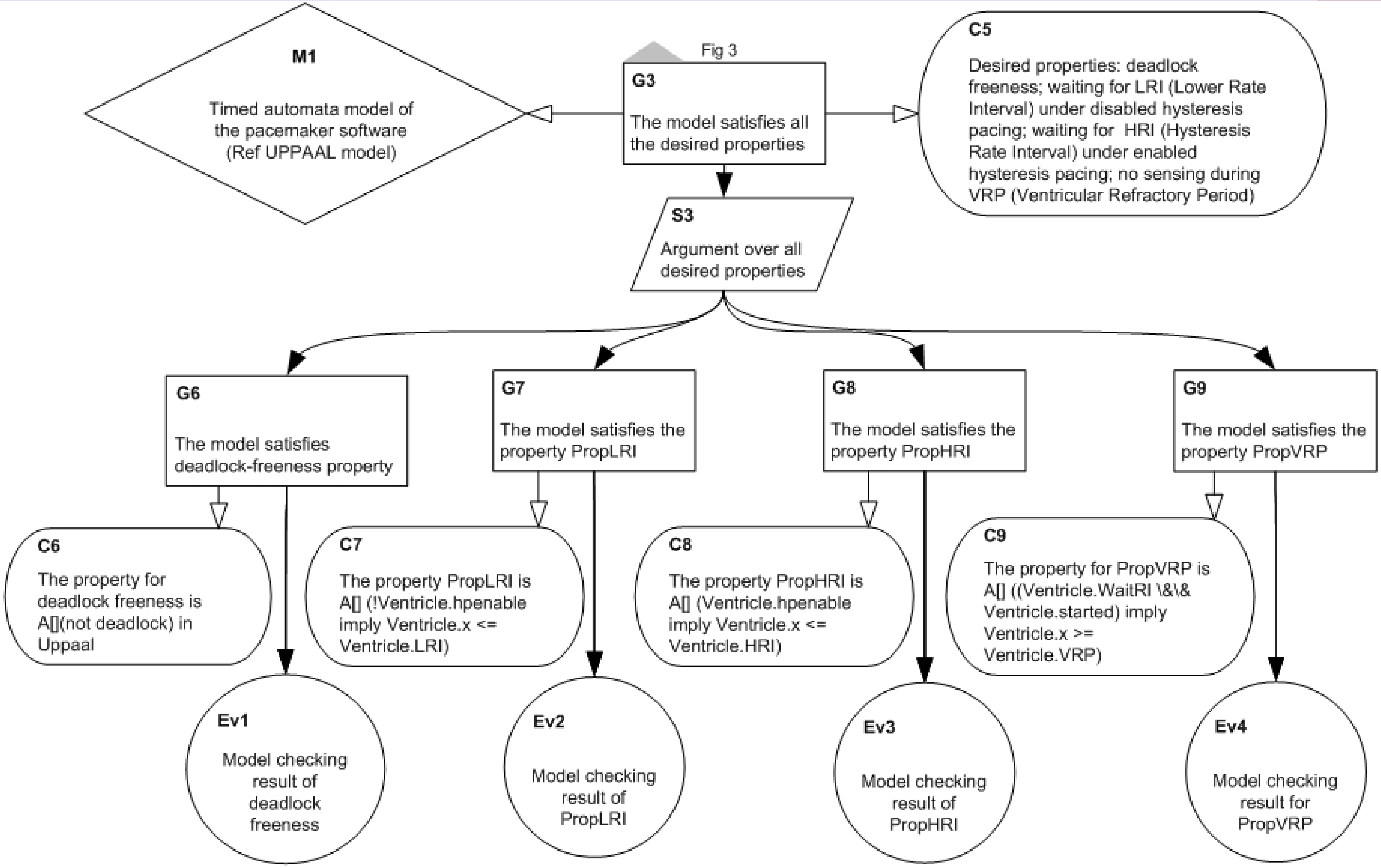
Fig 5

G5

The synthesized code satisfies all the desired properties with the timing tolerance

Fig 6

Modeling Claims



M1

Timed automata model of the pacemaker software (Ref UPPAAL model)

Fig 3

G3

The model satisfies all the desired properties

C5

Desired properties: deadlock freeness; waiting for LRI (Lower Rate Interval) under disabled hysteresis pacing; waiting for HRI (Hysteresis Rate Interval) under enabled hysteresis pacing; no sensing during VRP (Ventricular Refractory Period)

S3

Argument over all desired properties

G6

The model satisfies deadlock-freeness property

G7

The model satisfies the property PropLRI

G8

The model satisfies the property PropHRI

G9

The model satisfies the property PropVRP

C6

The property for deadlock freeness is A[](not deadlock) in Uppaal

C7

The property PropLRI is A[] (!Ventricle.hpenable imply Ventricle.x <= Ventricle.LRI)

C8

The property PropHRI is A[] (Ventricle.hpenable imply Ventricle.x <= Ventricle.HRI)

C9

The property for PropVRP is A[] ((Ventricle.WaitRI \&\& Ventricle.started) imply Ventricle.x >= Ventricle.VRP)

Ev1

Model checking result of deadlock freeness

Ev2

Model checking result of PropLRI

Ev3

Model checking result of PropHRI

Ev4

Model checking result for PropVRP

Code Synthesis Claims

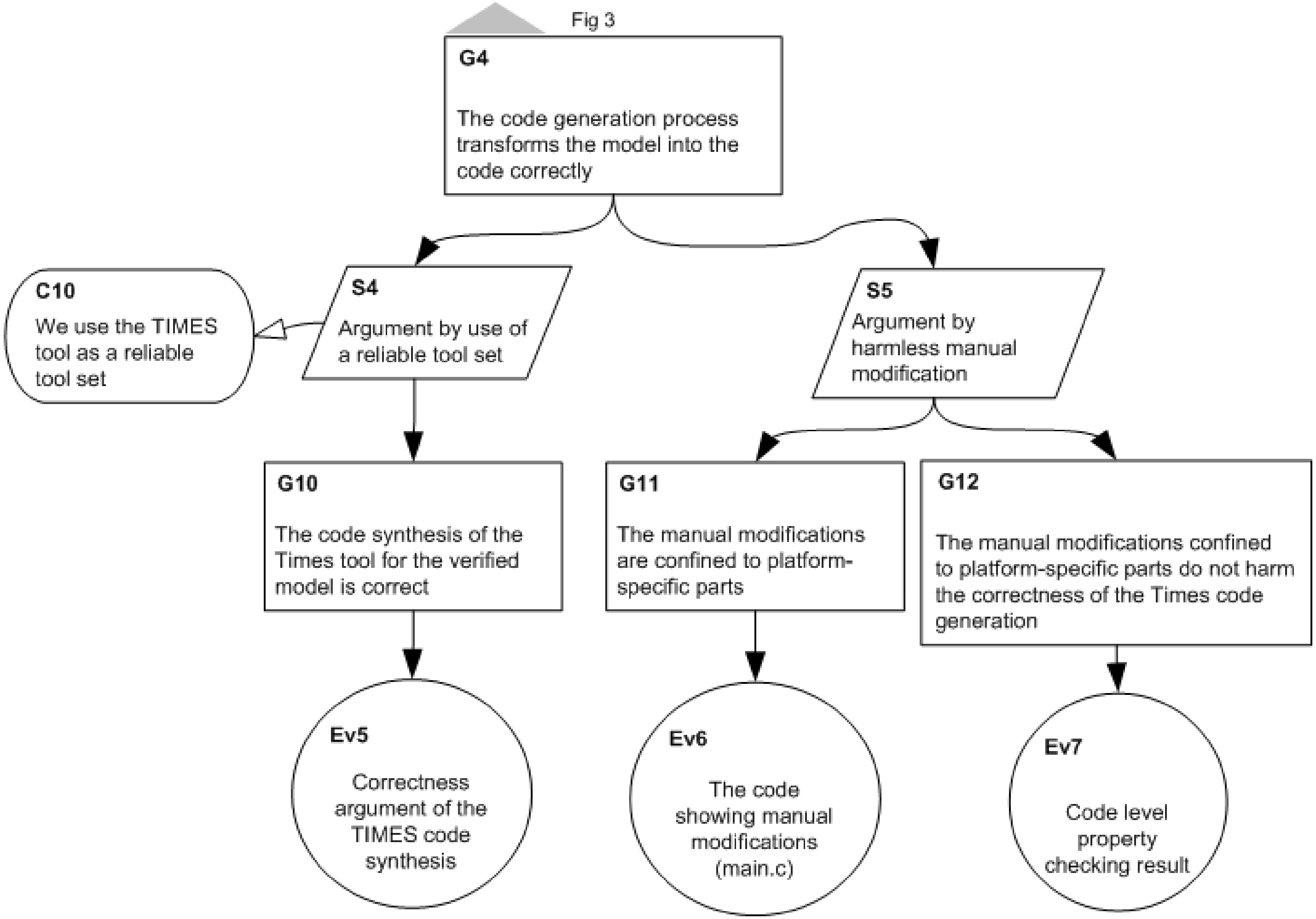


Fig 3

G4

The code generation process transforms the model into the code correctly

S4

Argument by use of a reliable tool set

C10

We use the TIMES tool as a reliable tool set

S5

Argument by harmless manual modification

G10

The code synthesis of the Times tool for the verified model is correct

G11

The manual modifications are confined to platform-specific parts

G12

The manual modifications confined to platform-specific parts do not harm the correctness of the Times code generation

Ev5

Correctness argument of the TIMES code synthesis

Ev6

The code showing manual modifications (main.c)

Ev7

Code level property checking result

Timing Tolerance Claims

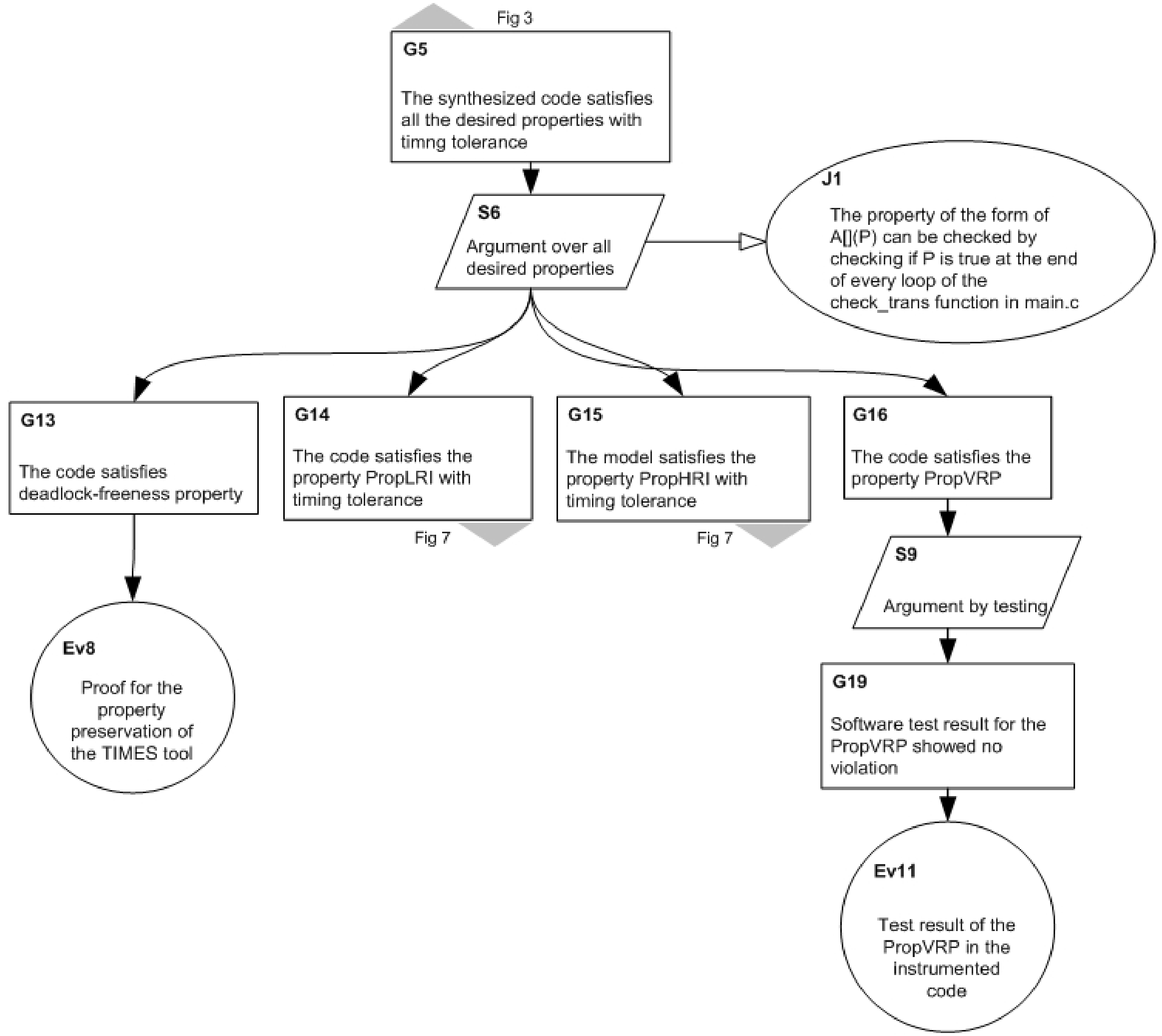


Fig 3

G5

The synthesized code satisfies all the desired properties with timng tolerance

S6

Argument over all desired properties

J1

The property of the form of A[](P) can be checked by checking if P is true at the end of every loop of the check_trans function in main.c

G13

The code satisfies deadlock-freeness property

G14

The code satisfies the property PropLRI with timing tolerance

Fig 7

G15

The model satisfies the property PropHRI with timing tolerance

Fig 7

G16

The code satisfies the property PropVRP

S9

Argument by testing

G19

Software test result for the PropVRP showed no violation

Ev8

Proof for the property preservation of the TIMES tool

Ev11

Test result of the PropVRP in the instrumented code

Korea Advanced Institute of Science and Technology

Timing Tolerance Claims (cont.)

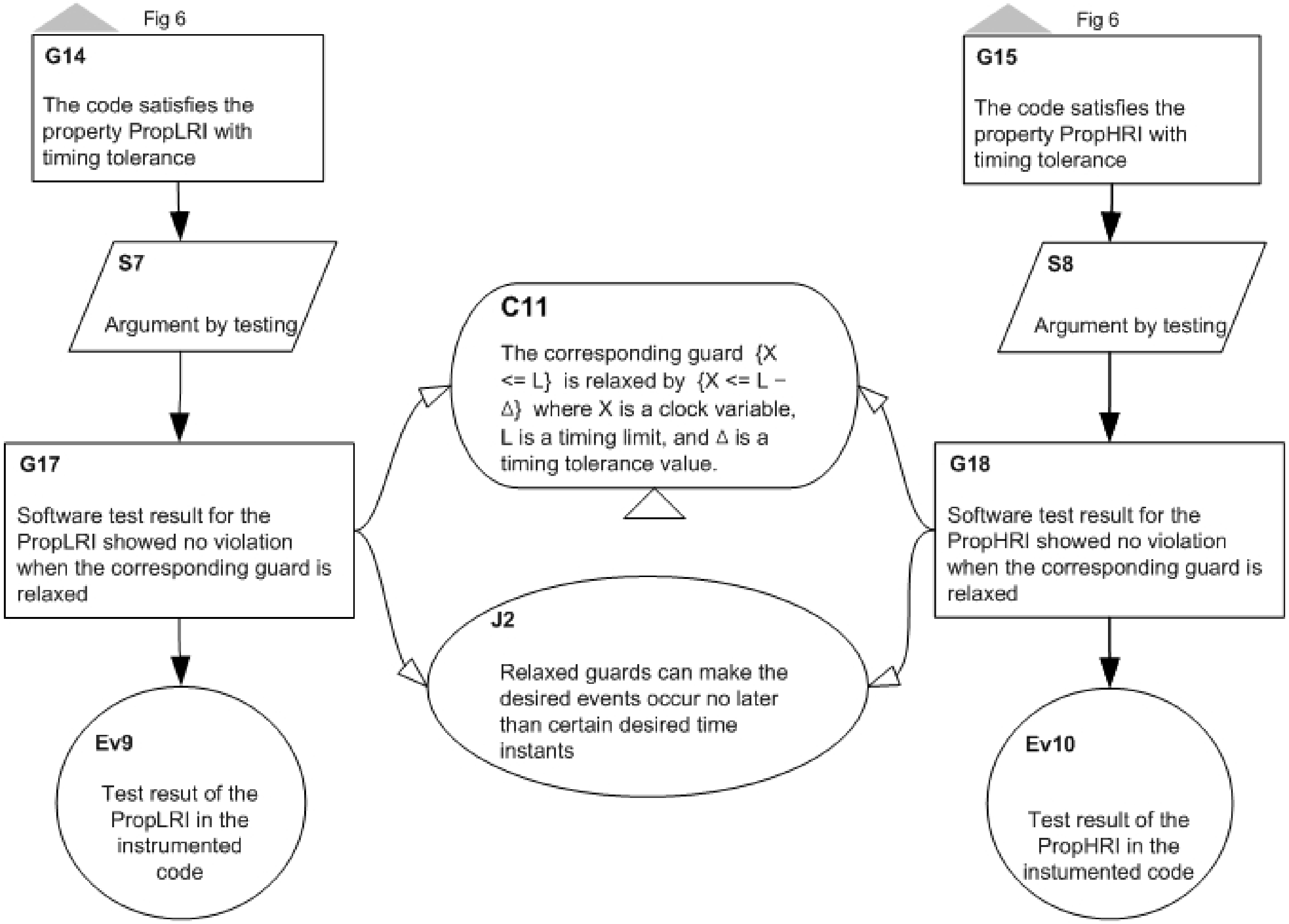


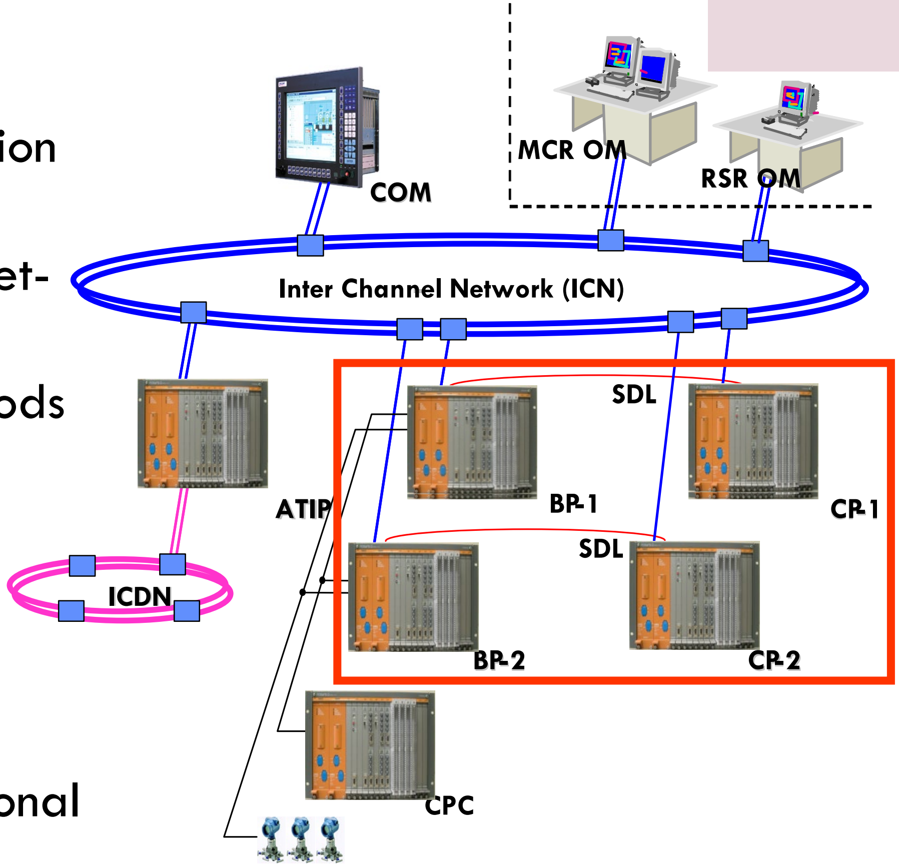
Fig 6
G14
The code satisfies the property PropLRI with timing tolerance
S7
Argument by testing
G17
Software test result for the PropLRI showed no violation when the corresponding guard is relaxed
Ev9
Test resut of the PropLRI in the instrumented code
C11
The corresponding guard {X <= L} is relaxed by {X <= L – Δ} where X is a clock variable, L is a timing limit, and Δ is a timing tolerance value.
J2
Relaxed guards can make the desired events occur no later than certain desired time instants
Fig 6
G15
The code satisfies the property PropHRI with timing tolerance
S8
Argument by testing
G18
Software test result for the PropHRI showed no violation when the corresponding guard is relaxed
Ev10
Test result of the PropHRI in the instumented code

Lessons Learned: Potential Assurance Case Benefits

- Improves comprehension of existing arguments
- Improves discussion and reduces time-to-agreement on what evidence is needed and what the evidence means
- (Having identified argument structure up front) focuses activities towards the specific end-objectives
- Recognition and exploitation of successful (convincing) arguments becomes possible (assurance case patterns)
- Supports monitoring of project progress towards successful certification

Case Study 2: Reactor Protection System

- Bistable Processor (BP)
  - Part of the KNICS* reactor protection system (RPS)
  - Compares processing values with set-points
  - Developed using prescriptive methods
- KNICS project
  - Goal: to achieve technical self-reliance in the area of nuclear instrumentation and control
  - Over 1,000 documents were generated, conforming to international standards and guidelines
    - NUREG-0800, IEEE STD-1228, etc.

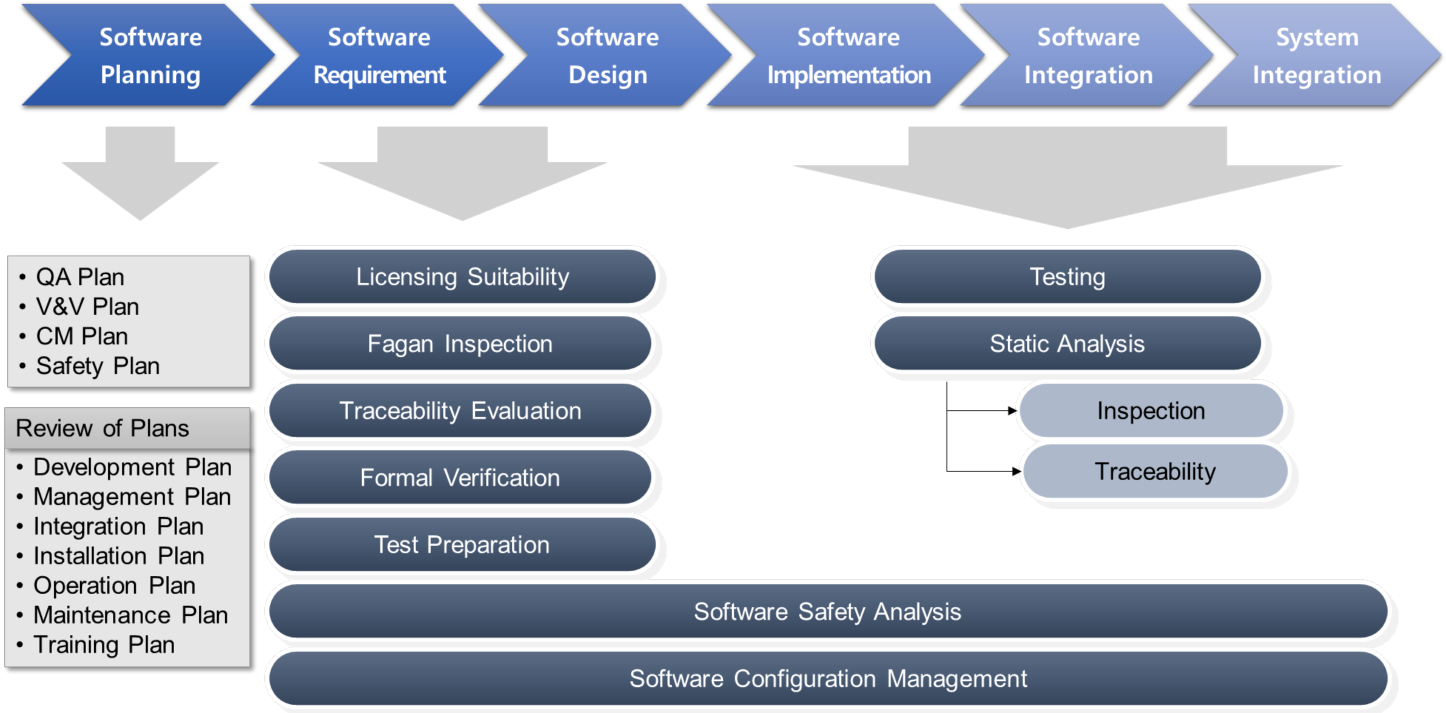


KNICS RPS

*KNICS: Korea Nuclear Instrumentation and Control System (원전계측제어시스템개발사업단)

Software V&V Activities of KNICS RPS

- The software used in the KNICS RPS was developed under a rigorous procedure.
  - V&V activities were performed following the software development life cycle.



Software Planning → Software Requirement → Software Design → Software Implementation → Software Integration → System Integration

- QA Plan
- V&V Plan
- CM Plan
- Safety Plan

Review of Plans

- Development Plan
- Management Plan
- Integration Plan
- Installation Plan
- Operation Plan
- Maintenance Plan
- Training Plan

Licensing Suitability

Fagan Inspection

Traceability Evaluation

Formal Verification

Test Preparation

Testing

Static Analysis

Inspection

Traceability

Software Safety Analysis

Software Configuration Management

Safety Case vs. Prescriptive Approach

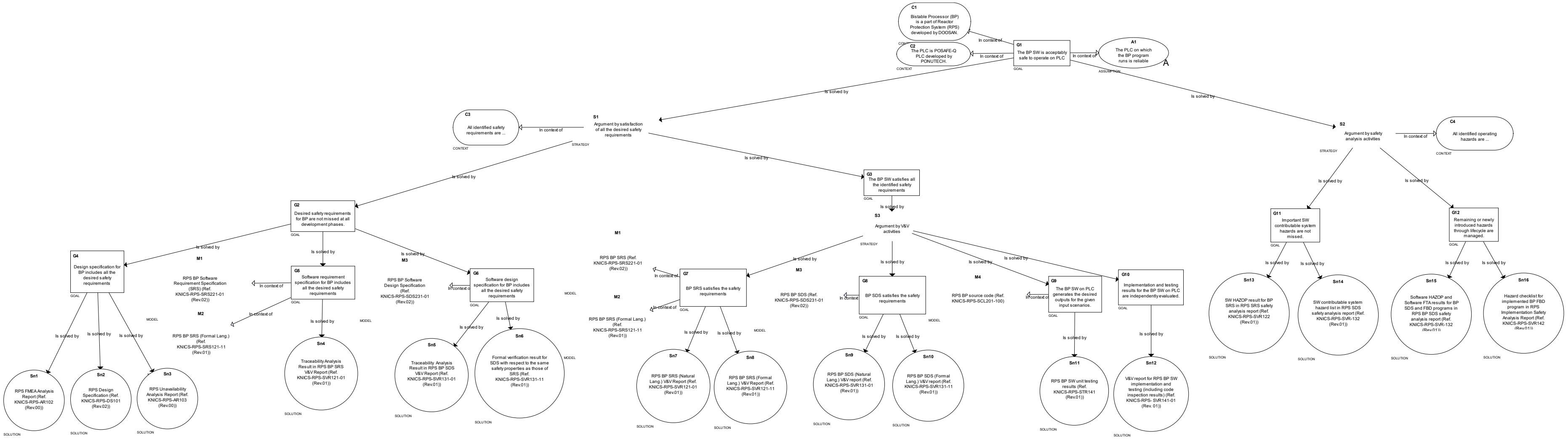
- Safety case
  - Structured argument, supported by a body of evidence that provides a compelling, comprehensible, and valid case that a system is safe for a given application in a given operating environment [1]
  - Considered an effective way to argue for and evaluate system safety
- Prescriptive (or process-based) approaches
  - Developers demonstrate software safety assurance by appealing to the satisfaction of objectives that the safety standards require for compliance.
  - Assumes that following the process prescribed in safety standards will generate evidence for safety [2]
- In this case study,
  - Created a safety case for a part of the reactor protection system
    - Illustrated how a safety case can be created with real-world industrial project data
  - Analyzed the results of applying the safety case approach to the target system developed through prescriptive methods

[1] MoD, Defence Standard 00-56 Issue 4 (Part 1): Safety Management Requirements for Defence Systems, UK Ministry of Defence.

[2] R. Hawkins, et al., "Assurance Cases and Prescriptive Software Safety Certification: A Comparative Study," Safety Science, vol. 59, pp.55–71, 2013.

The BP SW safety case

- Top claim
  - The BP SW is acceptably safe to operate on the PLC.



G1
The BP SW is acceptably safe to operate on PLC

C1
Bistable Processor (BP) is a part of Reactor Protection System (RPS) developed by DOOSAN.

C2
The PLC is POSAFE-Q PLC developed by PONUTECH.

A1
The PLC on which the BP program runs is reliable

S1
Argument by satisfaction of all the desired safety requirements

C3
All identified safety requirements are ...

S2
Argument by safety analysis activities

C4
All identified operating hazards are ...

G2
Desired safety requirements for BP are not missed at all development phases.

G3
The BP SW satisfies all the identified safety requirements

S3
Argument by V&V activities

G4
Design specification for BP includes all the desired safety requirements

G5
Software requirement specification for BP includes all the desired safety requirements

G6
Software design specification for BP includes all the desired safety requirements

M1
RPS BP Software Requirement Specification (SRS) (Ref. KNICS-RPS-SRS221-01 (Rev.02))

M2
RPS BP SRS (Formal Lang.) (Ref. KNICS-RPS-SRS121-11 (Rev.01))

M3
RPS BP Software Design Specification (Ref. KNICS-RPS-SDS231-01 (Rev.02))

Sn1
RPS FMEA Analysis Report (Ref. KNICS-RPS-AR102 (Rev.00))

Sn2
RPS Design Specification (Ref. KNICS-RPS-DS101 (Rev.02))

Sn3
RPS Unavailability Analysis Report (Ref. KNICS-RPS-AR103 (Rev.00))

Sn4
Traceability Analysis Result in RPS BP SRS V&V Report (Ref. KNICS-RPS-SVR121-01 (Rev.01))

Sn5
Traceability Analysis Result in RPS BP SDS V&V Report (Ref. KNICS-RPS-SVR131-01 (Rev.01))

Sn6
Formal verification result for SDS with respect to the same safety properties as those of SRS (Ref. KNICS-RPS-SVR131-11 (Rev.01))

M1
RPS BP SRS (Ref. KNICS-RPS-SRS221-01 (Rev.02))

M2
RPS BP SRS (Formal Lang.) (Ref. KNICS-RPS-SRS121-11 (Rev.01))

G7
BP SRS satisfies the safety requirements

M3
RPS BP SDS (Ref. KNICS-RPS-SDS231-01 (Rev.02))

G8
BP SDS satisfies the safety requirements

M4
RPS BP source code (Ref. KNICS-RPS-SCL201-100)

G9
The BP SW on PLC generates the desired outputs for the given input scenarios.

G10
Implementation and testing results for the BP SW on PLC are independently evaluated.

Sn7
RPS BP SRS (Natural Lang.) V&V Report (Ref. KNICS-RPS-SVR121-01 (Rev.01))

Sn8
RPS BP SRS (Formal Lang.) V&V Report (Ref. KNICS-RPS-SVR121-11 (Rev.01))

Sn9
RPS BP SDS (Natural Lang.) V&V report (Ref. KNICS-RPS-SVR131-01 (Rev.01))

Sn10
RPS BP SDS (Formal Lang.) V&V report (Ref. KNICS-RPS-SVR131-11 (Rev.01))

Sn11
RPS BP SW unit testing results (Ref. KNICS-RPS-STR141 (Rev.01))

Sn12
V&V report for RPS BP SW implementation and testing (including code inspection results) (Ref. KNICS-RPS- SVR141-01 (Rev. 01))

G11
Important SW contributable system hazards are not missed.

G12
Remaining or newly introduced hazards through lifecycle are managed.

Sn13
SW HAZOP result for BP SRS in RPS SRS safety analysis report (Ref. KNICS-RPS-SVR122 (Rev.01))

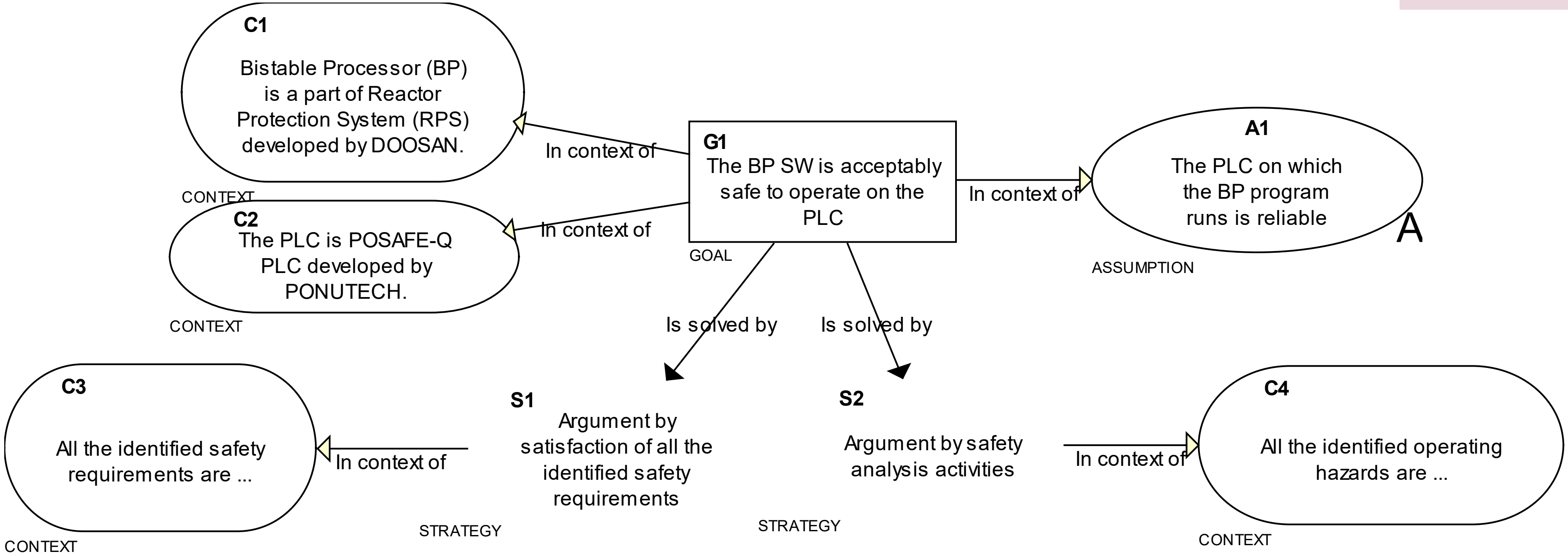
Sn14
SW contributable system hazard list in RPS SDS safety analysis report (Ref. KNICS-RPS-SVR-132 (Rev.01))

Sn15
Software HAZOP and Software FTA results for BP SDS and FBD programs in RPS BP SDS safety analysis report (Ref. KNICS-RPS-SVR-132 (Rev.01))

Sn16
Hazard checklist for implemented BP FBD program in RPS Implementation Safety Analysis Report (Ref. KNICS-RPS-SVR142 (Rev.01))

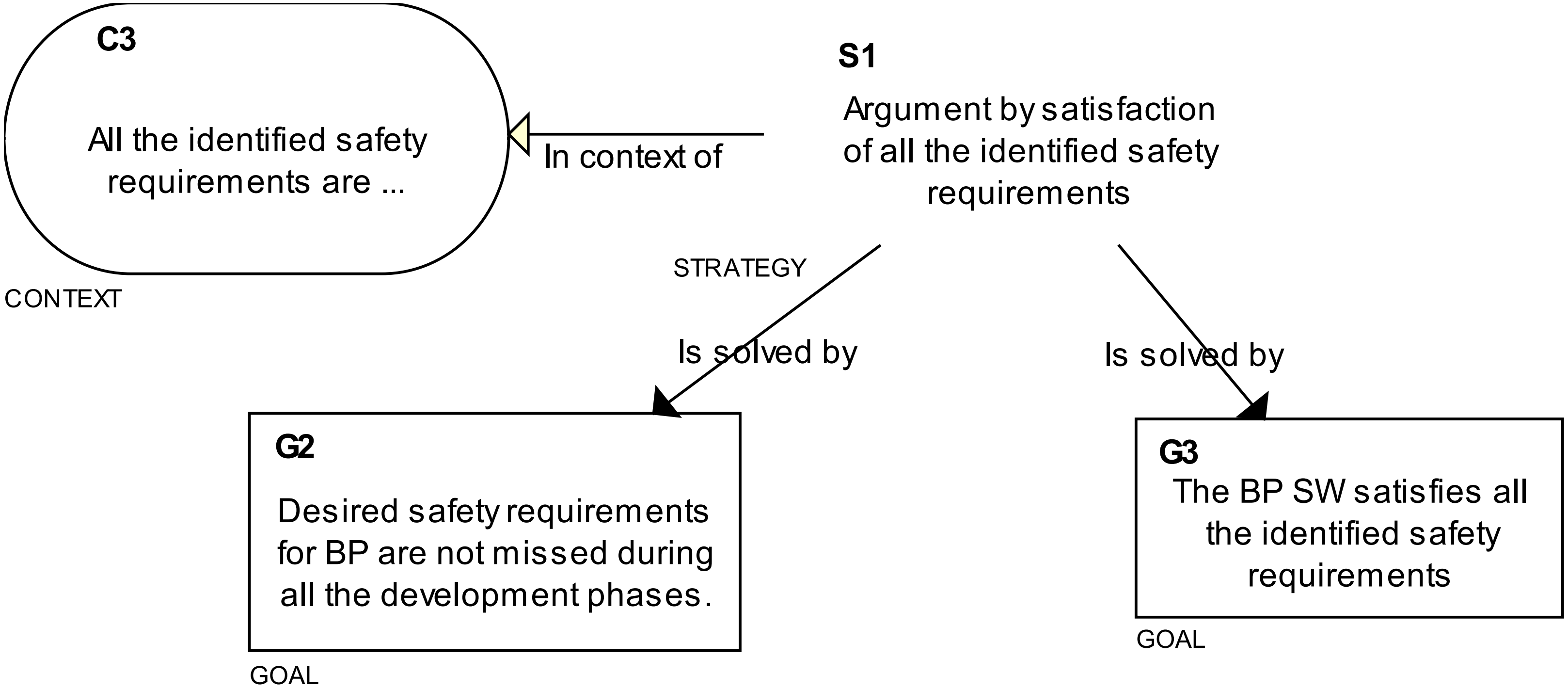
The BP SW safety case – Bird's eye view

Structuring the Safety Case

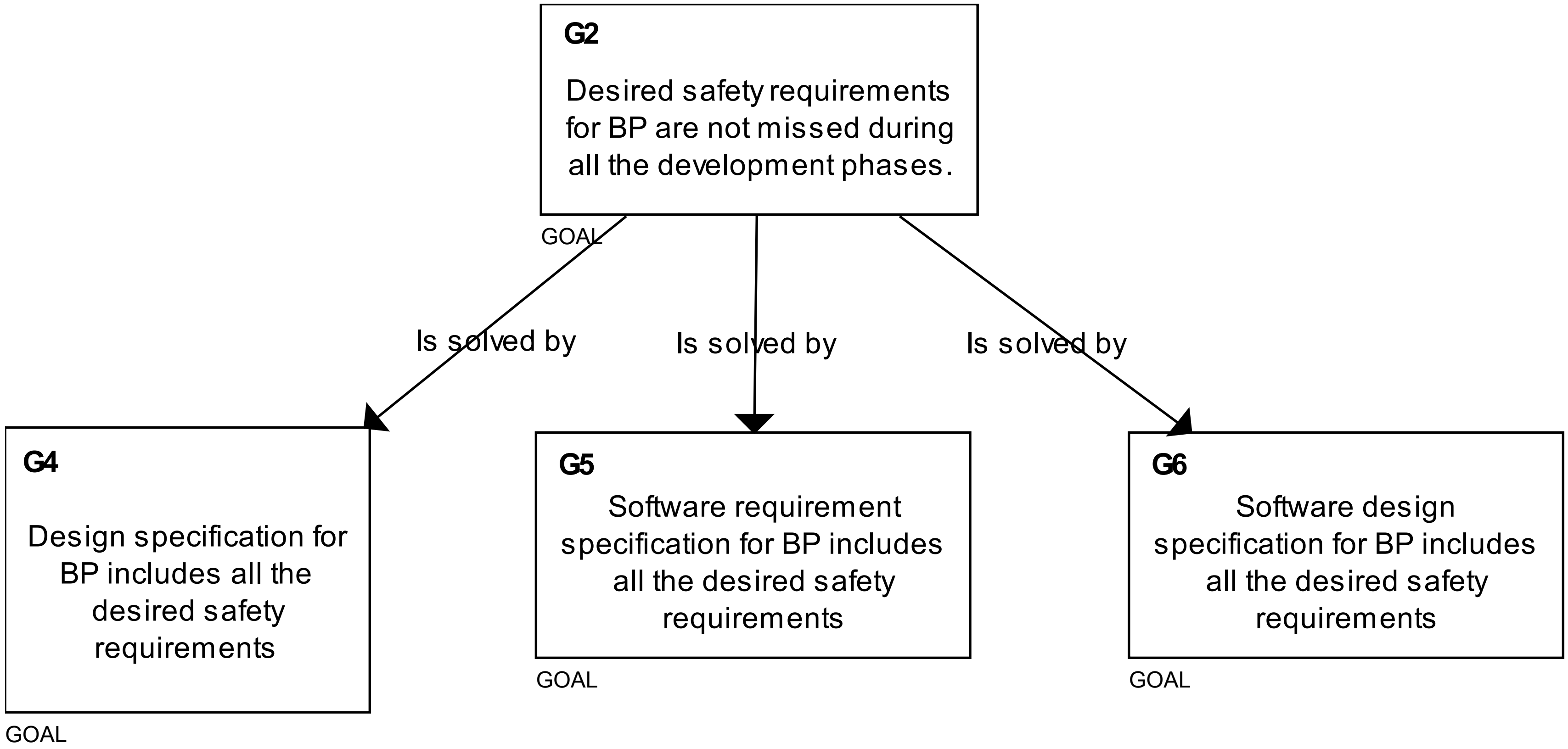


C1
Bistable Processor (BP) is a part of Reactor Protection System (RPS) developed by DOOSAN.
CONTEXT
In context of
G1
The BP SW is acceptably safe to operate on the PLC
GOAL
In context of
A1
The PLC on which the BP program runs is reliable
A
ASSUMPTION
C2
The PLC is POSAFE-Q PLC developed by PONUTECH.
CONTEXT
In context of
Is solved by
Is solved by
C3
All the identified safety requirements are ...
CONTEXT
In context of
S1
Argument by satisfaction of all the identified safety requirements
STRATEGY
S2
Argument by safety analysis activities
STRATEGY
In context of
C4
All the identified operating hazards are ...
CONTEXT

Argument by Satisfaction of Safety Requirements


C3
All the identified safety requirements are ...
CONTEXT
In context of
S1
Argument by satisfaction of all the identified safety requirements
STRATEGY
Is solved by
Is solved by
G2
Desired safety requirements for BP are not missed during all the development phases.
GOAL
G3
The BP SW satisfies all the identified safety requirements
GOAL

Safety Requirements are Not Missed



G2

Desired safety requirements for BP are not missed during all the development phases.

GOAL

Is solved by

Is solved by

Is solved by

G4

Design specification for BP includes all the desired safety requirements

GOAL

G5

Software requirement specification for BP includes all the desired safety requirements

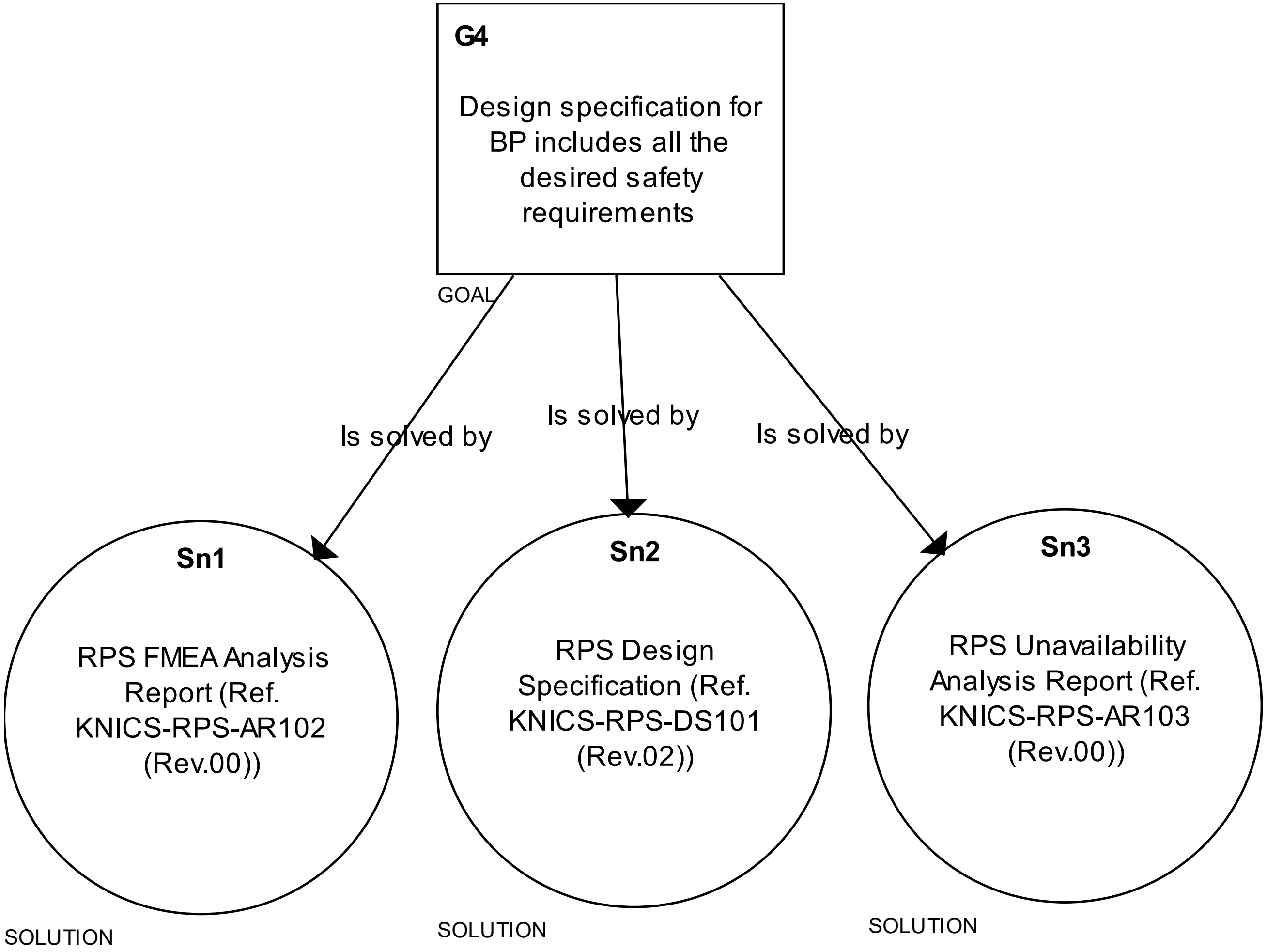
GOAL

G6

Software design specification for BP includes all the desired safety requirements

GOAL

Design Spec. Includes All the Safety Requirements



G4

Design specification for BP includes all the desired safety requirements

GOAL

Is solved by

Is solved by

Is solved by

Sn1

RPS FMEA Analysis Report (Ref. KNICS-RPS-AR102 (Rev.00))

SOLUTION

Sn2

RPS Design Specification (Ref. KNICS-RPS-DS101 (Rev.02))

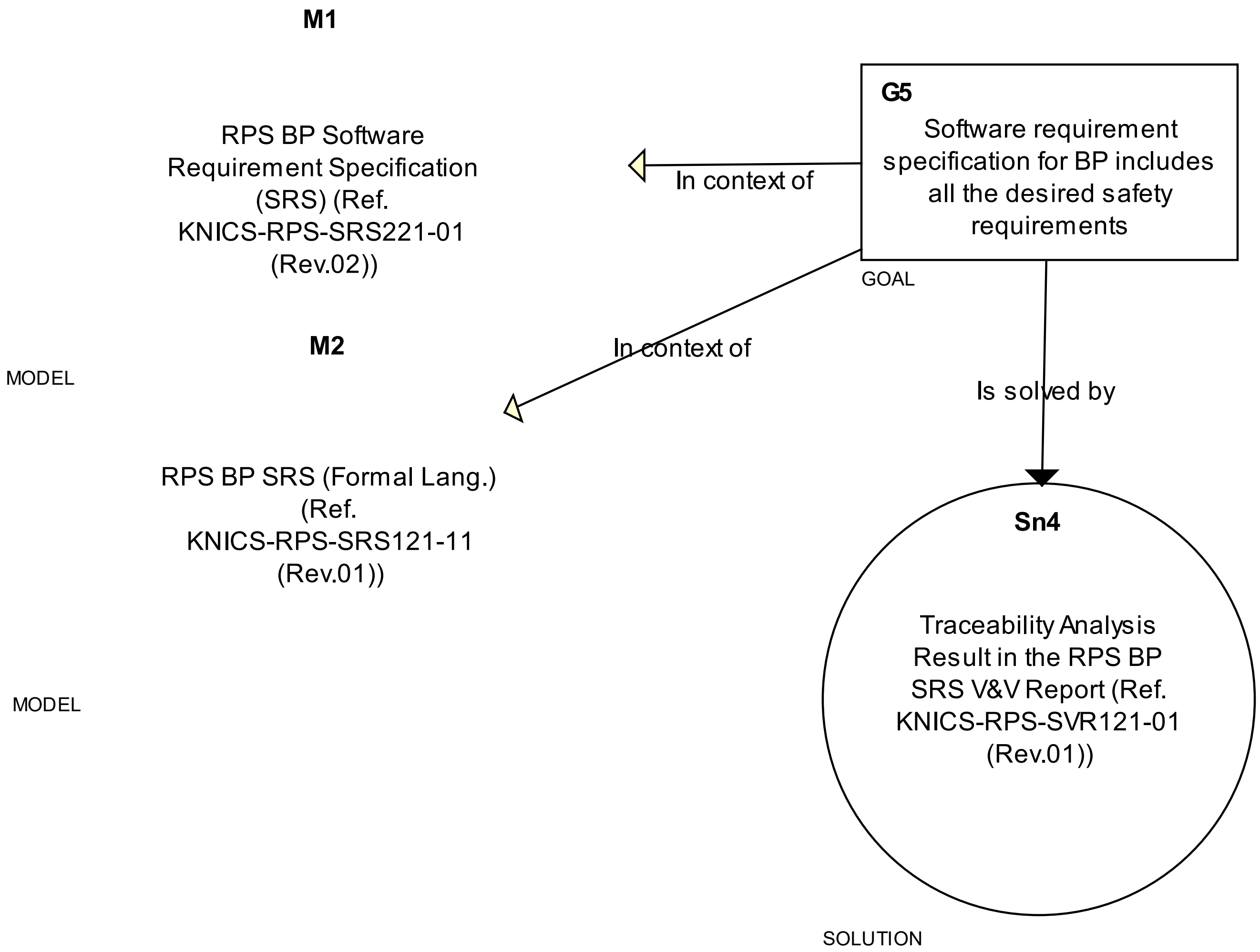
SOLUTION

Sn3

RPS Unavailability Analysis Report (Ref. KNICS-RPS-AR103 (Rev.00))

SOLUTION

SRS Includes All the Safety Requirements



M1

RPS BP Software Requirement Specification (SRS) (Ref. KNICS-RPS-SRS221-01 (Rev.02))

MODEL

M2

RPS BP SRS (Formal Lang.) (Ref. KNICS-RPS-SRS121-11 (Rev.01))

MODEL

G5

Software requirement specification for BP includes all the desired safety requirements

GOAL

In context of

In context of

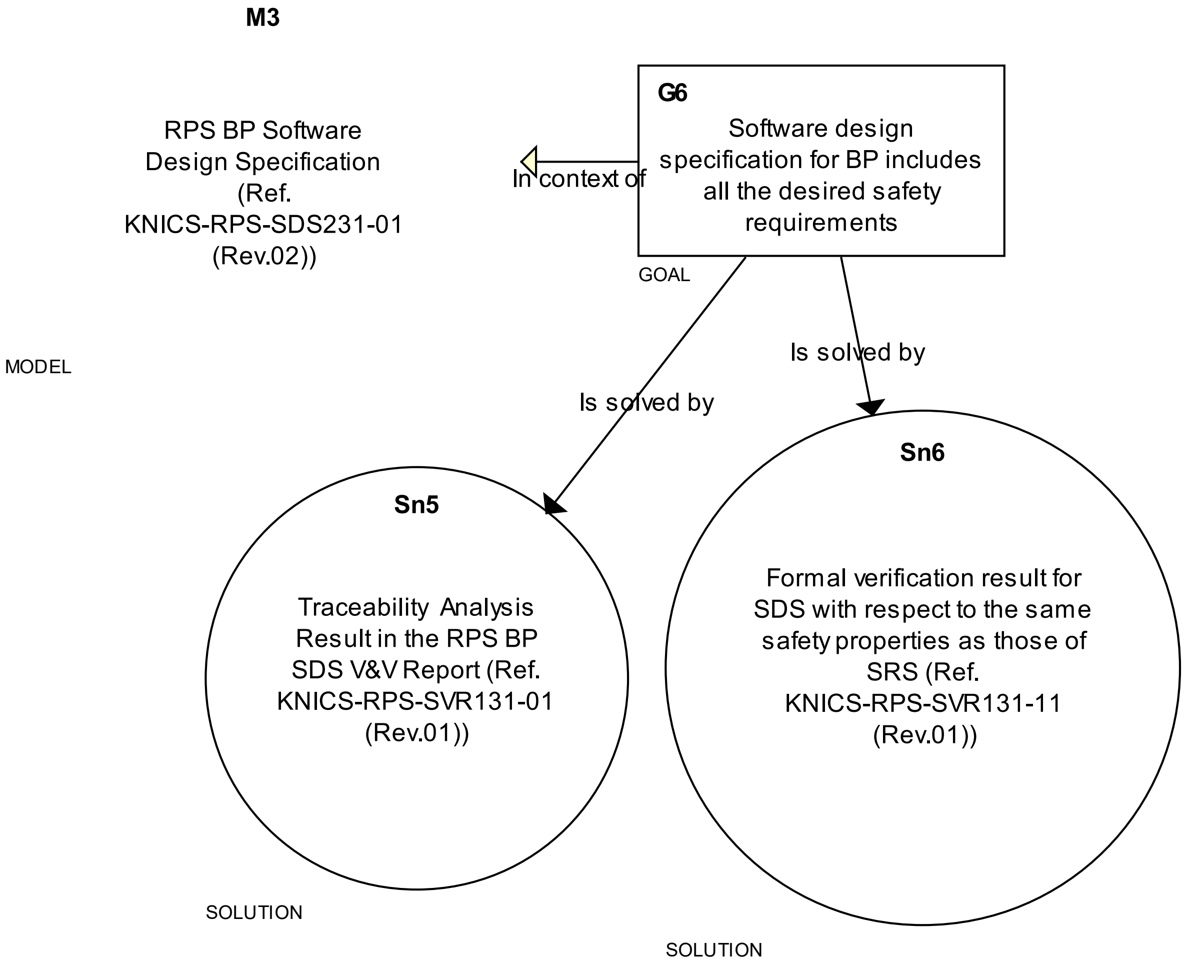
Is solved by

Sn4

Traceability Analysis Result in the RPS BP SRS V&V Report (Ref. KNICS-RPS-SVR121-01 (Rev.01))

SOLUTION

SDS Includes All the Safety Requirements



M3

RPS BP Software Design Specification (Ref. KNICS-RPS-SDS231-01 (Rev.02))

MODEL

In context of

G6

Software design specification for BP includes all the desired safety requirements

GOAL

Is solved by

Is solved by

Sn5

Traceability Analysis Result in the RPS BP SDS V&V Report (Ref. KNICS-RPS-SVR131-01 (Rev.01))

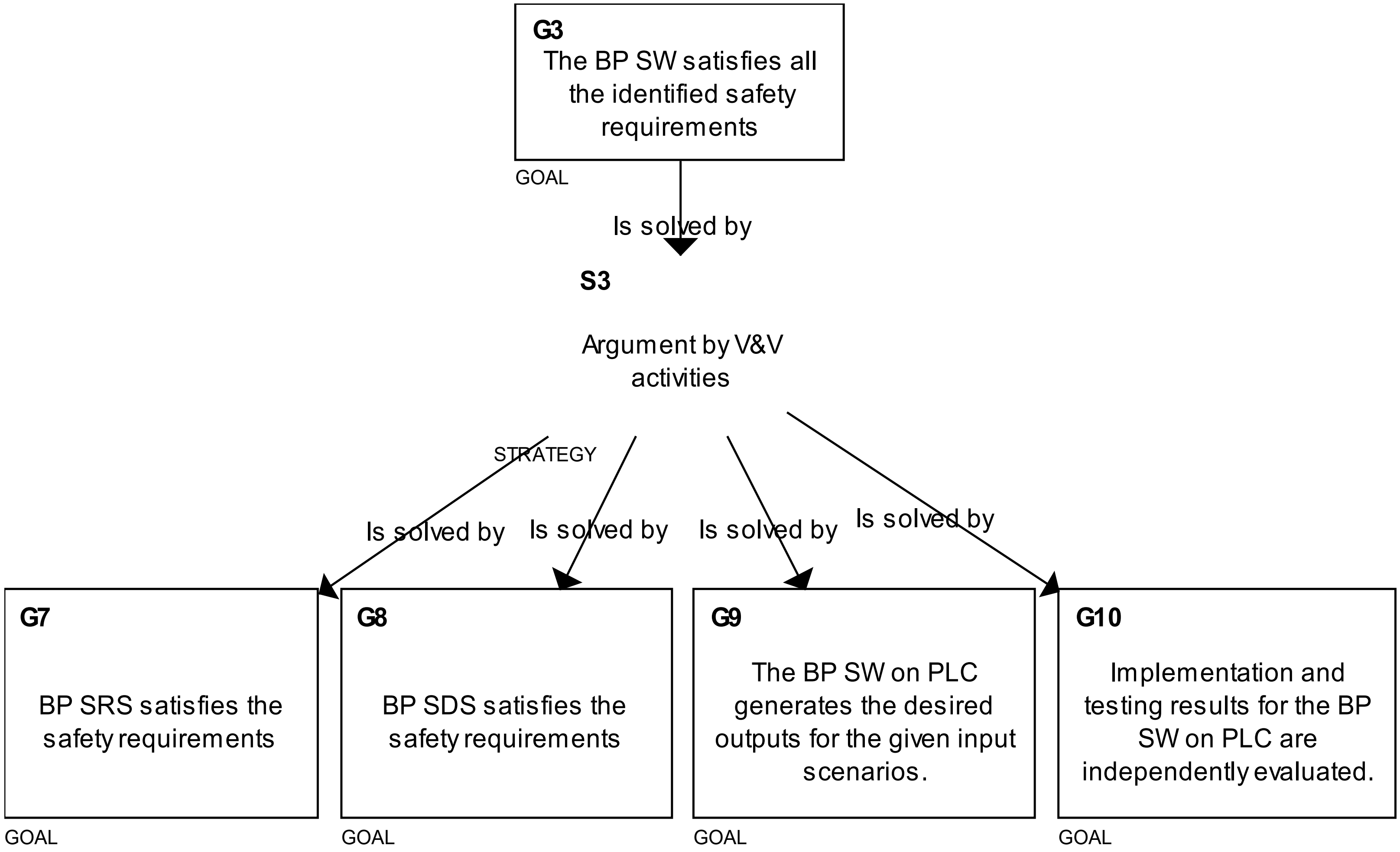
SOLUTION

Sn6

Formal verification result for SDS with respect to the same safety properties as those of SRS (Ref. KNICS-RPS-SVR131-11 (Rev.01))

SOLUTION

Argument by V&V Activities



G3
The BP SW satisfies all the identified safety requirements
GOAL

Is solved by

S3
Argument by V&V activities
STRATEGY

Is solved by

Is solved by

Is solved by

Is solved by

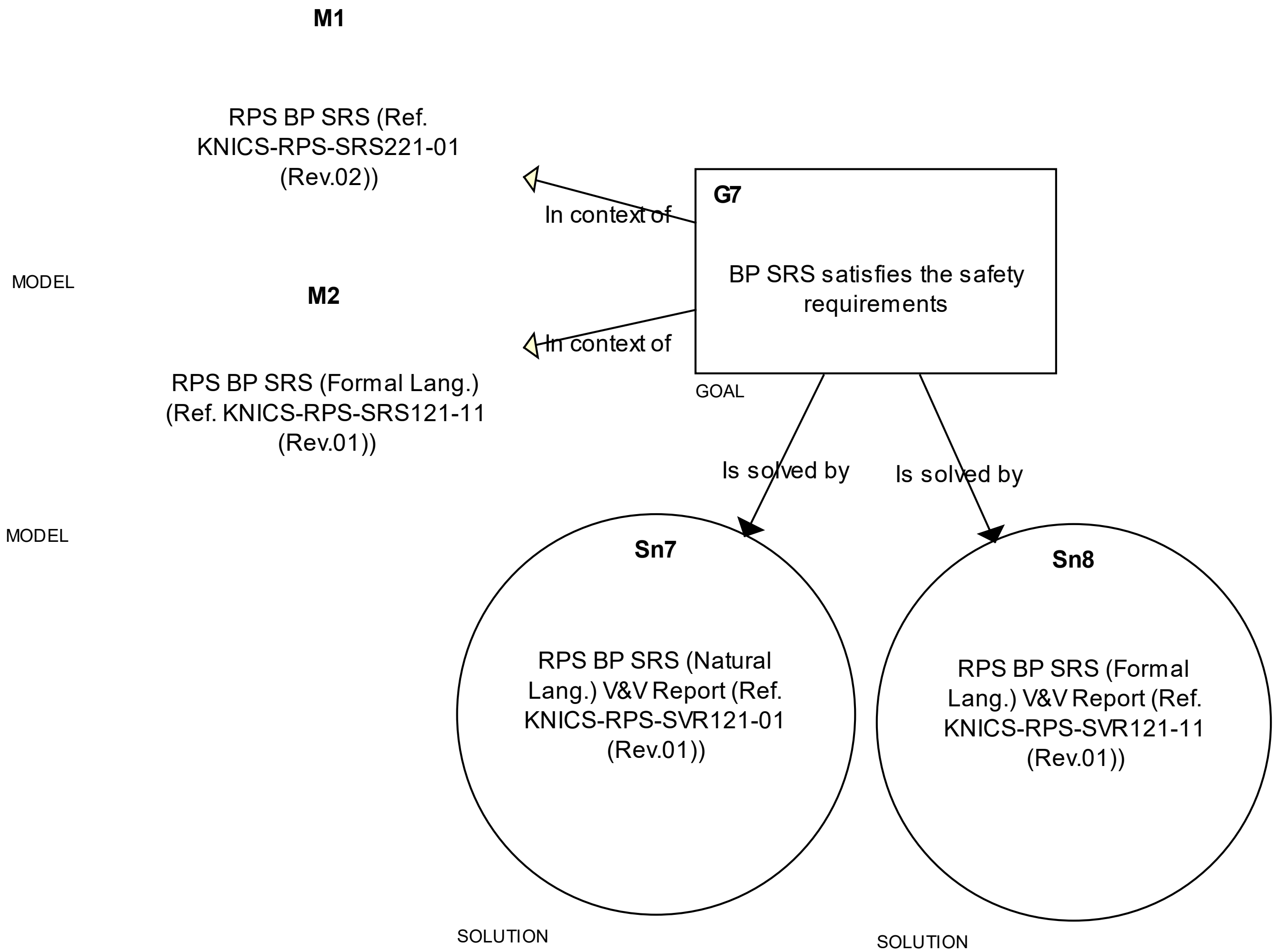
G7
BP SRS satisfies the safety requirements
GOAL

G8
BP SDS satisfies the safety requirements
GOAL

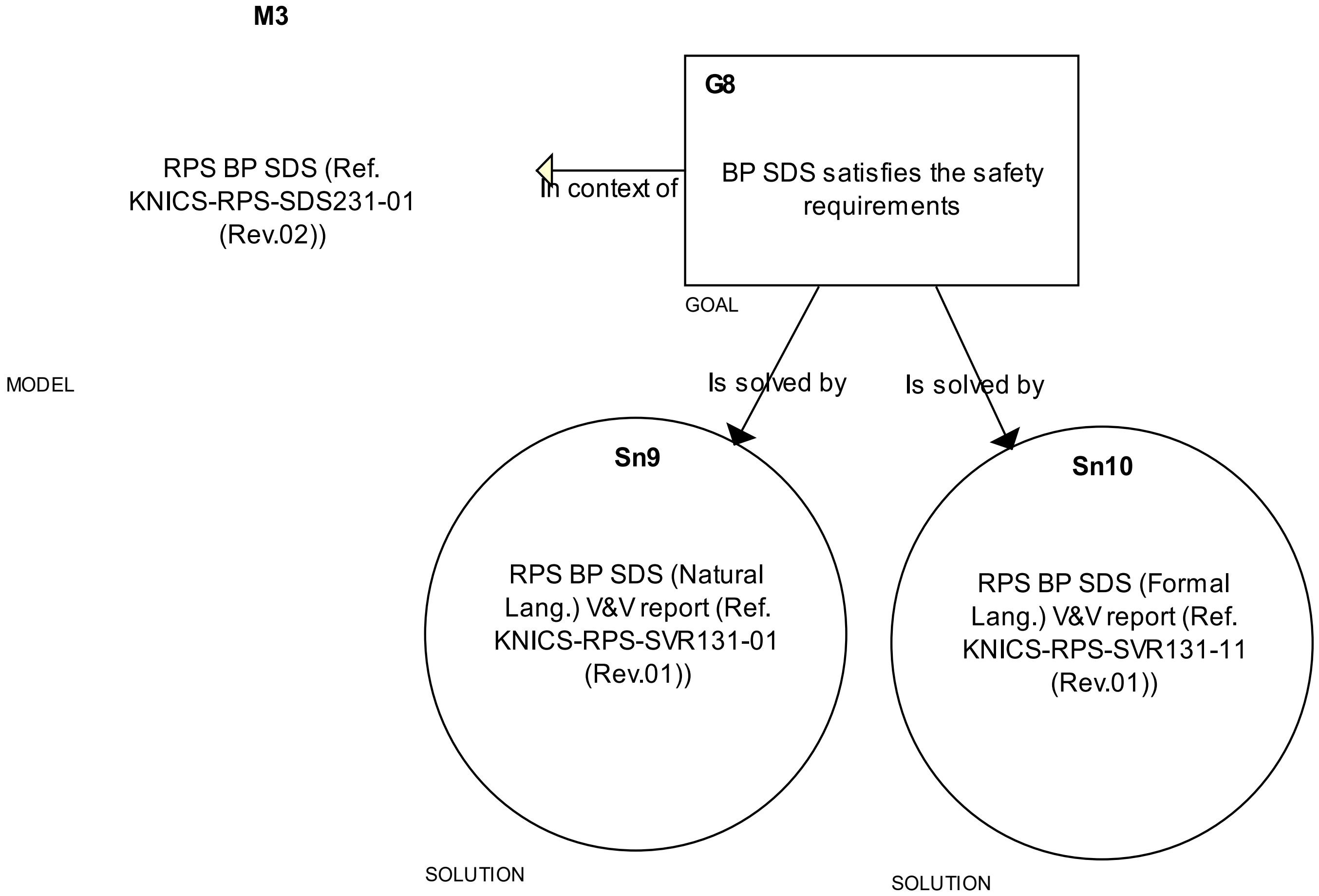
G9
The BP SW on PLC generates the desired outputs for the given input scenarios.
GOAL

G10
Implementation and testing results for the BP SW on PLC are independently evaluated.
GOAL

BP SRS Satisfies All the Safety Requirements

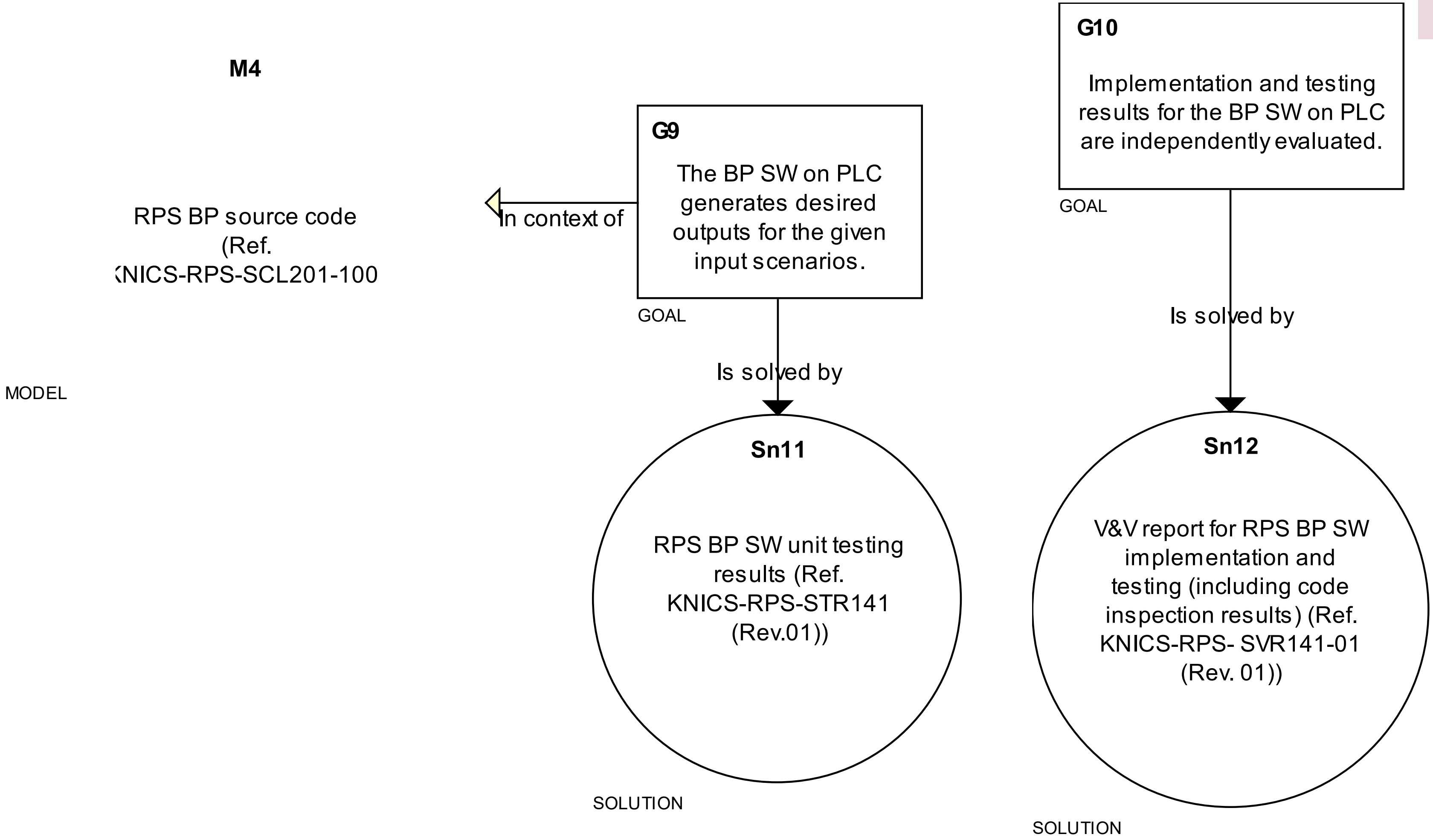

M1
RPS BP SRS (Ref. KNICS-RPS-SRS221-01 (Rev.02))
MODEL
M2
RPS BP SRS (Formal Lang.) (Ref. KNICS-RPS-SRS121-11 (Rev.01))
MODEL
In context of
In context of
G7
BP SRS satisfies the safety requirements
GOAL
Is solved by
Is solved by
Sn7
RPS BP SRS (Natural Lang.) V&V Report (Ref. KNICS-RPS-SVR121-01 (Rev.01))
SOLUTION
Sn8
RPS BP SRS (Formal Lang.) V&V Report (Ref. KNICS-RPS-SVR121-11 (Rev.01))
SOLUTION

BP SDS Satisfies All the Safety Requirements



M3
RPS BP SDS (Ref. KNICS-RPS-SDS231-01 (Rev.02))
MODEL
In context of
G8
BP SDS satisfies the safety requirements
GOAL
Is solved by
Is solved by
Sn9
RPS BP SDS (Natural Lang.) V&V report (Ref. KNICS-RPS-SVR131-01 (Rev.01))
SOLUTION
Sn10
RPS BP SDS (Formal Lang.) V&V report (Ref. KNICS-RPS-SVR131-11 (Rev.01))
SOLUTION

Argument for BP SW Implementation Safety



M4

RPS BP source code (Ref. KNICS-RPS-SCL201-100

MODEL

In context of

G9

The BP SW on PLC generates desired outputs for the given input scenarios.

GOAL

Is solved by

Sn11

RPS BP SW unit testing results (Ref. KNICS-RPS-STR141 (Rev.01))

SOLUTION

G10

Implementation and testing results for the BP SW on PLC are independently evaluated.

GOAL

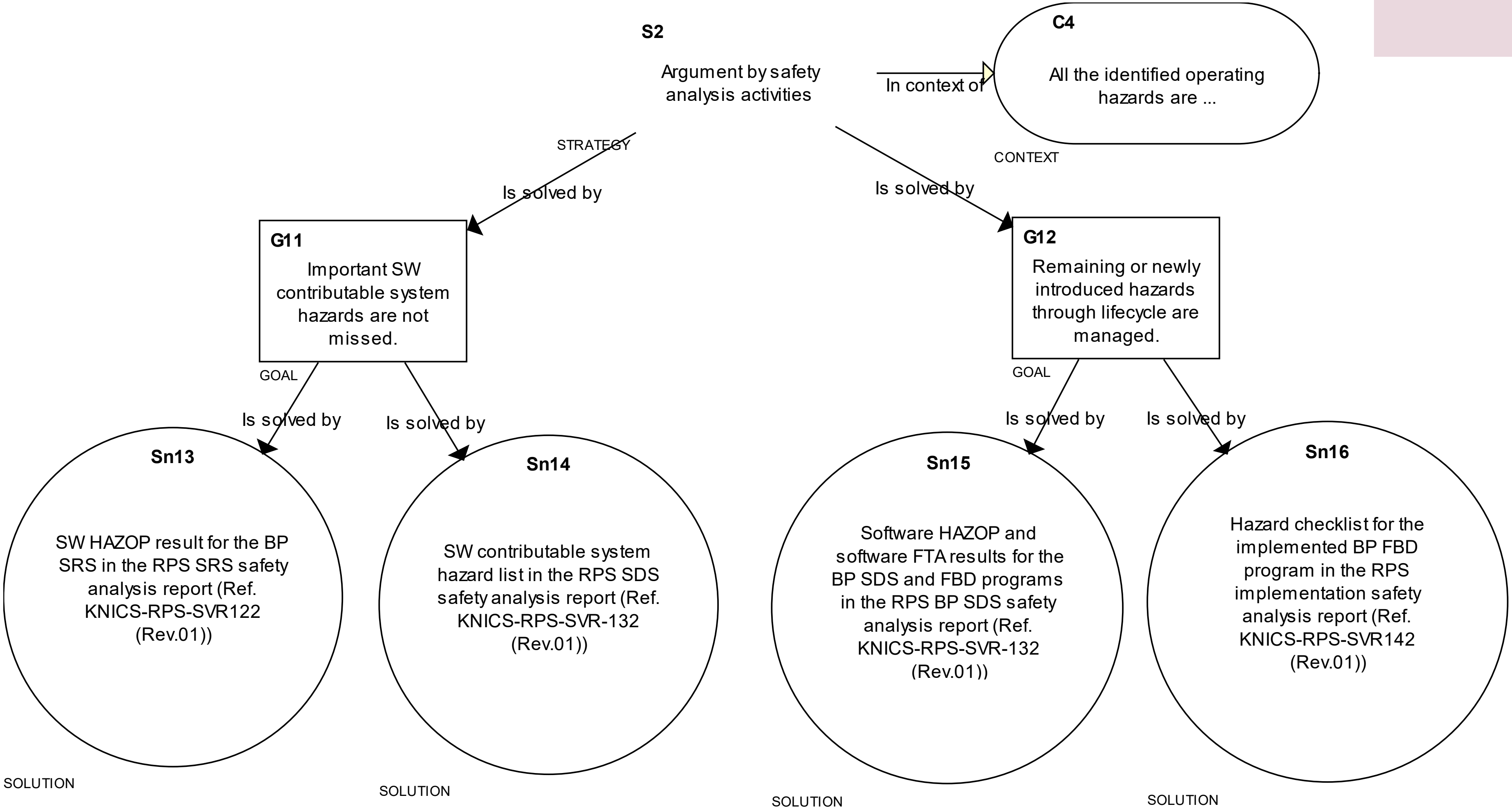
Is solved by

Sn12

V&V report for RPS BP SW implementation and testing (including code inspection results) (Ref. KNICS-RPS- SVR141-01 (Rev. 01))

SOLUTION

Argument by Safety Analysis Activities


S2
Argument by safety analysis activities
STRATEGY
In context of
C4
All the identified operating hazards are ...
CONTEXT
Is solved by
Is solved by
G11
Important SW contributable system hazards are not missed.
GOAL
G12
Remaining or newly introduced hazards through lifecycle are managed.
GOAL
Is solved by
Is solved by
Is solved by
Is solved by
Sn13
SW HAZOP result for the BP SRS in the RPS SRS safety analysis report (Ref. KNICS-RPS-SVR122 (Rev.01))
SOLUTION
Sn14
SW contributable system hazard list in the RPS SDS safety analysis report (Ref. KNICS-RPS-SVR-132 (Rev.01))
SOLUTION
Sn15
Software HAZOP and software FTA results for the BP SDS and FBD programs in the RPS BP SDS safety analysis report (Ref. KNICS-RPS-SVR-132 (Rev.01))
SOLUTION
Sn16
Hazard checklist for the implemented BP FBD program in the RPS implementation safety analysis report (Ref. KNICS-RPS-SVR142 (Rev.01))
SOLUTION

Limitations

- The presented safety case is not complete.
  - Revisions and/or corrections are needed.
    - E.g.) concretization of each safety requirement, concretization of each operating hazard, addition of a claim for safety of PLC, etc.
- The presented safety case was created with existing artifacts of an already developed system.
  - How the prescriptive approach and the safety case approach can complement each other during development was not evaluated.

Lessons Learned: Possible Advantages

- Possible advantages of using safety cases with the prescriptive approach.
  - How the BP software safety issues had been addressed could be explicitly presented by creating a safety case.
    - Reviewing over 500 documents, in the case of KNICS RPS, took significant effort and time.
    - It is not easy to figure out whether a specific part of those documents is more or less important in the aspect of system safety.

→Safety cases can facilitate clearer and more efficient communication focusing on safety between the developers and the regulators in the certification process.

Lessons Learned: Possible Drawbacks

- Possible drawbacks of using safety cases with the prescriptive approach
  - Creating additional safety cases to the artifacts required by safety standards entails extra efforts and costs.
    - Still, a significant portion of safety case creation and management relies on manual work.

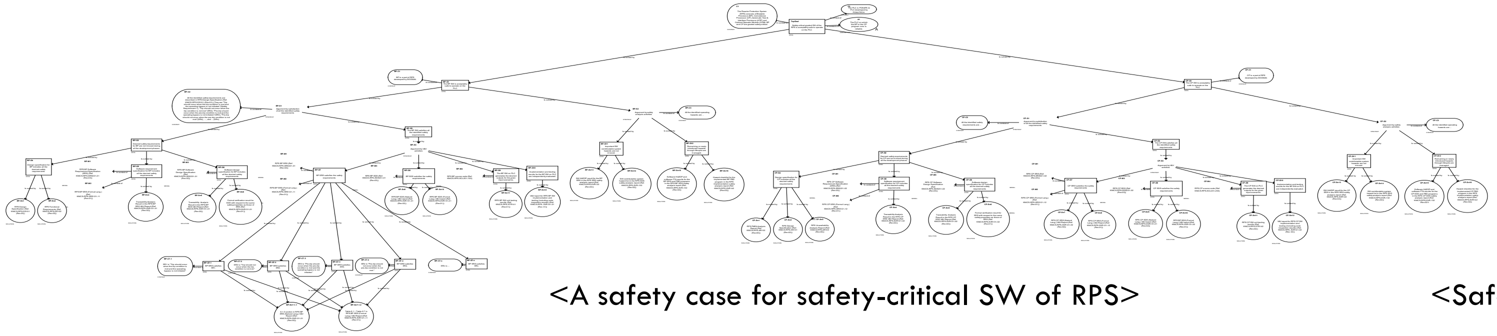
  → Efforts to develop proper guidelines and tools for creating and managing safety cases should be continued.

  - The safety case approaches are not enough to cover all the requirements.
    - The prescriptive approaches consider not only safety requirements, but also other quality attributes, e.g., security, performance, etc.
    - The safety case approaches may not be able to replace the prescriptive approaches.

  → Combining prescriptive approaches and safety case approaches in an effective and efficient way needs to be studied further.

Case Study 3: Safety Case Review

- Conducted a case study of safety case review
  - Target: Safety-critical software in KNICS RPS
- A safety case for safety-critical software from KNICS RPS was reviewed by a safety case expert from IFE in Norway
  - in the position of the regulator/licensing agency
  - the evaluation opinion was documented
- Derived considerations when using safety case technology for system safety demonstration and licensing



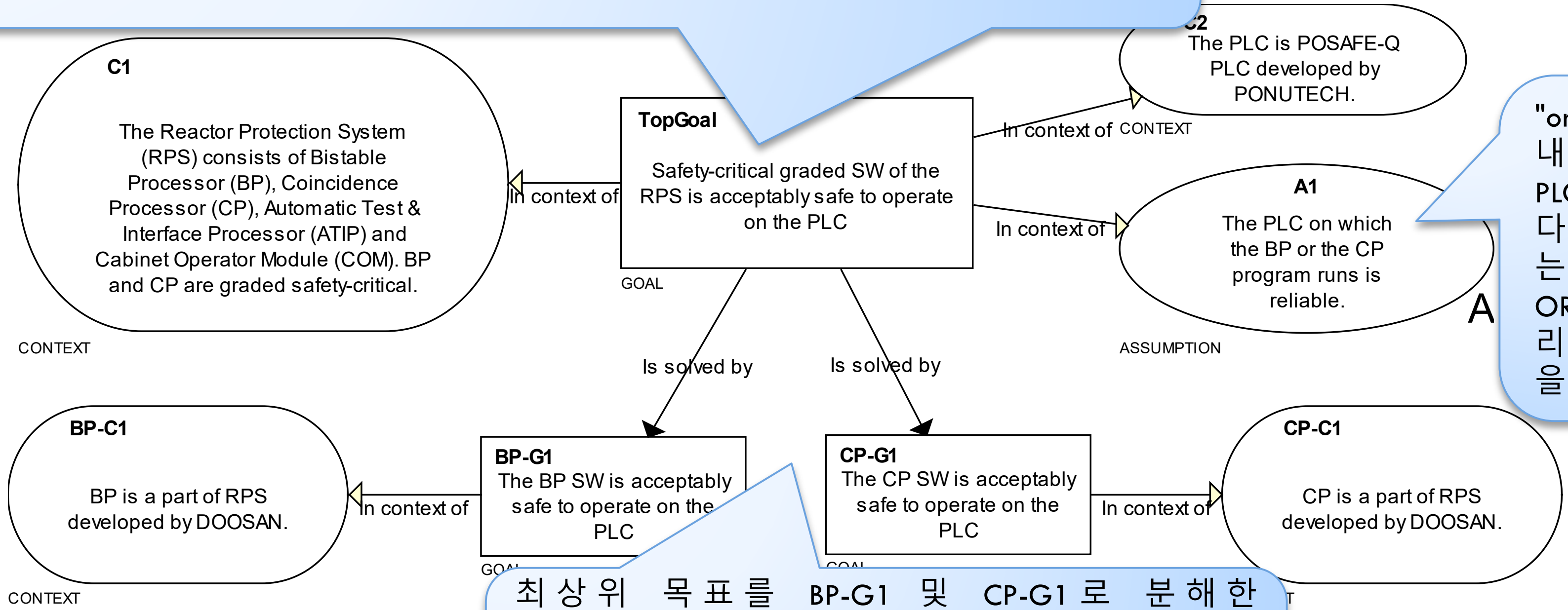
<A safety case for safety-critical SW of RPS>

<Safety case review result>

Review on Top Claim

- Top claim
  - Safety-critical graded SW of the RPS is acceptably safe to operate on the PLC.

Top Goal, BP-G1 및 CP-G1에서 "수용될만큼 안전한(acceptably safe)"을 어떻게 이해할 것인지를, 예를 들어 "BP가 수용될만큼 안전하다는 것은 [ref.]에 있는 안전 요구사항 SF_X와 SF_Y에 의해 정의된다."와 같이, 컨텍스트(context)에서 명확히 할 필요가 있다.


C1
The Reactor Protection System (RPS) consists of Bistable Processor (BP), Coincidence Processor (CP), Automatic Test & Interface Processor (ATIP) and Cabinet Operator Module (COM). BP and CP are graded safety-critical.
CONTEXT
In context of
TopGoal
Safety-critical graded SW of the RPS is acceptably safe to operate on the PLC
GOAL
In context of
C2
The PLC is POSAFE-Q PLC developed by PONUTECH.
CONTEXT
In context of
A1
The PLC on which the BP or the CP program runs is reliable.
A
ASSUMPTION
Is solved by
Is solved by
BP-C1
BP is a part of RPS developed by DOOSAN.
CONTEXT
In context of
BP-G1
The BP SW is acceptably safe to operate on the PLC
CP-G1
The CP SW is acceptably safe to operate on the PLC
In context of
CP-C1
CP is a part of RPS developed by DOOSAN.

"or"가 무엇을 나타내는가? 동일한 PLC인지? 두 개의 다른 PLC를 나타내는 것인지? 논리 OR, 논리 XOR, 논리 AND 중 어느 것을 의미하는가?

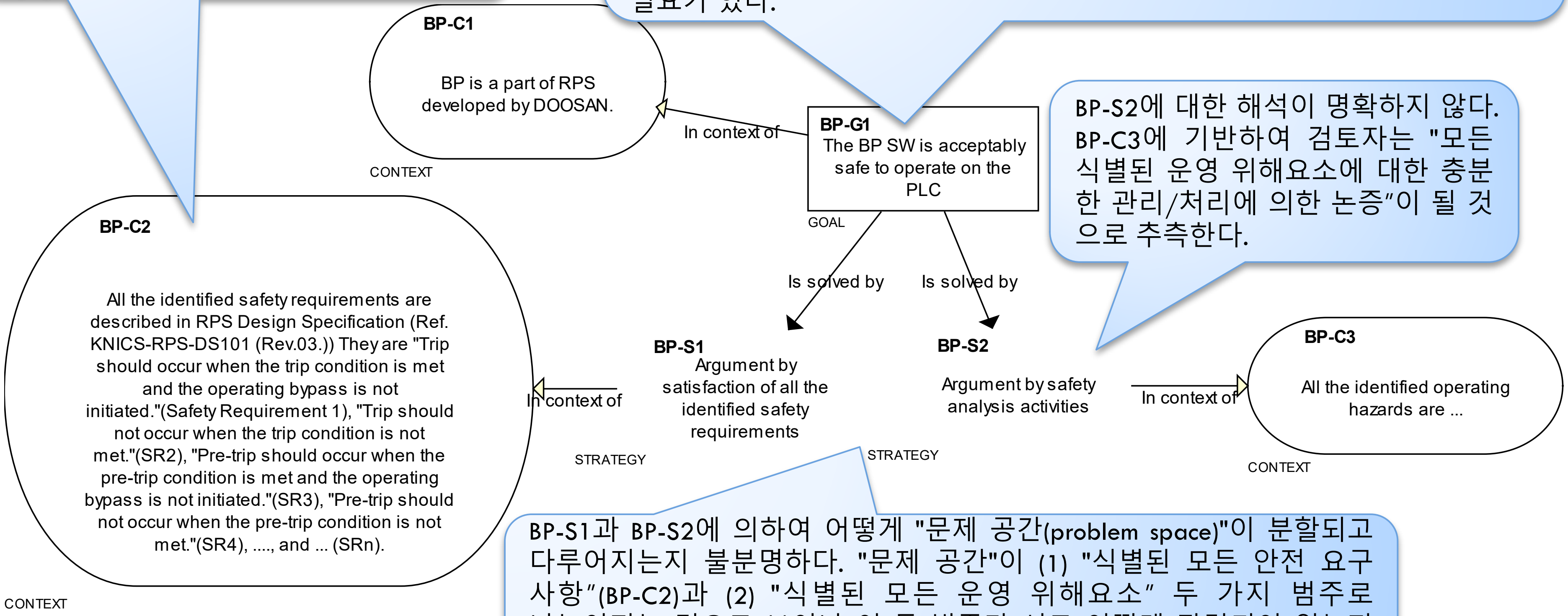
최상위 목표를 BP-G1 및 CP-G1로 분해한 논리/전략은 컨텍스트 C1에서 설명하고 있는대로 RPS 아키텍처를 기반으로 한다. 검토자는 여기에 제시된 논증 구조에서 분해 논리/전략이 명확하다고 생각한다.

KAIST Korea Advanced Institute of Science and Technology

Review on Argument

분명한 이해를 위해 안전진술에서 안전, 소프트웨어, 요구 사항의 개념이 서로 어떻게 관련되어 있는지 차별화되고 명확해져야 한다.

BP-G1의 분해 논리/전략이 보다 명시적으로 표현되어야 한다. 즉 "수용될만큼 안전한(acceptably safe)...”이란 정확히 무엇을 의미하는가? 예를 들어, BP-S1과 BP-S2를 기반으로 "BP SW는 BP SW가 [ref.]에 식별된 안전 요구사항들을 만족시키고(AND) [참조문헌과 함께 새로 구성된 BP-S2 부분]이면 PLC에서 작동하기에 수용될만큼 안전하다.“ 와 같이 “수용될만큼 안전한…”의 의미를 정의할 필요가 있다.



BP-C1

BP is a part of RPS developed by DOOSAN.

CONTEXT

In context of

BP-G1

The BP SW is acceptably safe to operate on the PLC

GOAL

Is solved by

Is solved by

BP-C2

All the identified safety requirements are described in RPS Design Specification (Ref. KNICS-RPS-DS101 (Rev.03.)) They are "Trip should occur when the trip condition is met and the operating bypass is not initiated."(Safety Requirement 1), "Trip should not occur when the trip condition is not met."(SR2), "Pre-trip should occur when the pre-trip condition is met and the operating bypass is not initiated."(SR3), "Pre-trip should not occur when the pre-trip condition is not met."(SR4), ..., and ... (SRn).

CONTEXT

In context of

BP-S1

Argument by satisfaction of all the identified safety requirements

STRATEGY

BP-S2

Argument by safety analysis activities

STRATEGY

In context of

BP-C3

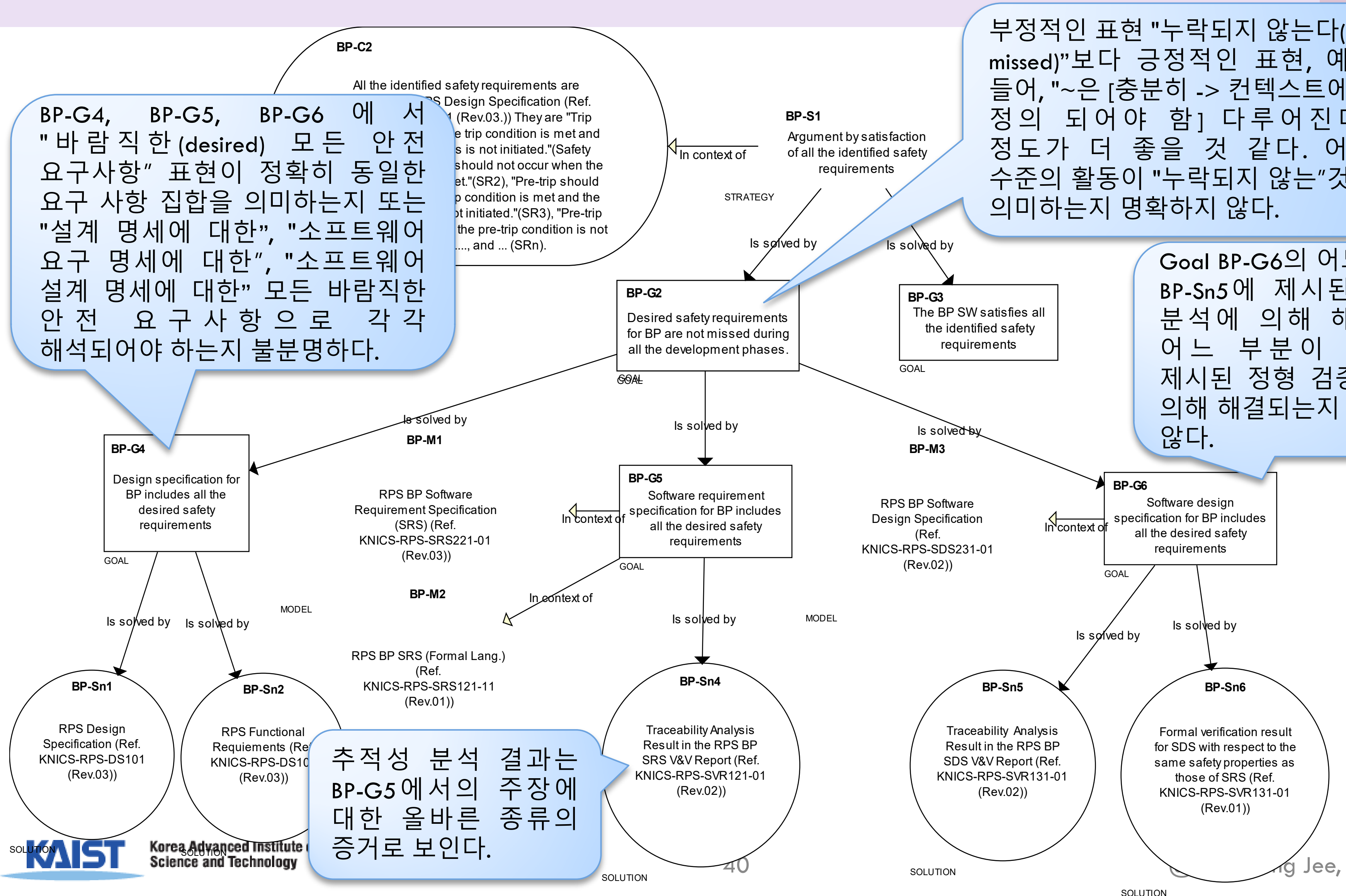
All the identified operating hazards are ...

CONTEXT

BP-S2에 대한 해석이 명확하지 않다. BP-C3에 기반하여 검토자는 "모든 식별된 운영 위해요소에 대한 충분한 관리/처리에 의한 논증”이 될 것으로 추측한다.

BP-S1과 BP-S2에 의하여 어떻게 "문제 공간(problem space)"이 분할되고 다루어지는지 불분명하다. "문제 공간"이 (1) "식별된 모든 안전 요구사항”(BP-C2)과 (2) "식별된 모든 운영 위해요소” 두 가지 범주로 나누어지는 것으로 보이나 이 두 범주가 서로 어떻게 관련되어 있는지 명확하지 않다.

Review on Goal-Solution Relationship



BP-C2
All the identified safety requirements are
...PS Design Specification (Ref.
...1 (Rev.03.)) They are "Trip
...e trip condition is met and
...s is not initiated."(Safety
...should not occur when the
...et."(SR2), "Pre-trip should
...p condition is met and the
...ot initiated."(SR3), "Pre-trip
... the pre-trip condition is not
...., and ... (SRn).
In context of
BP-S1
Argument by satisfaction of all the identified safety requirements
STRATEGY
Is solved by
Is solved by
BP-G2
Desired safety requirements for BP are not missed during all the development phases.
GOAL
BP-G3
The BP SW satisfies all the identified safety requirements
GOAL
Is solved by
BP-M1
Is solved by
Is solved by
BP-M3
BP-G4
Design specification for BP includes all the desired safety requirements
GOAL
RPS BP Software Requirement Specification (SRS) (Ref. KNICS-RPS-SRS221-01 (Rev.03))
In context of
BP-G5
Software requirement specification for BP includes all the desired safety requirements
GOAL
RPS BP Software Design Specification (Ref. KNICS-RPS-SDS231-01 (Rev.02))
In context of
BP-G6
Software design specification for BP includes all the desired safety requirements
GOAL
MODEL
BP-M2
In context of
MODEL
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
RPS BP SRS (Formal Lang.) (Ref. KNICS-RPS-SRS121-11 (Rev.01))
BP-Sn1
RPS Design Specification (Ref. KNICS-RPS-DS101 (Rev.03))
SOLUTION
BP-Sn2
RPS Functional Requiements (Re... KNICS-RPS-DS10... (Rev.03))
SOLUTION
BP-Sn4
Traceability Analysis Result in the RPS BP SRS V&V Report (Ref. KNICS-RPS-SVR121-01 (Rev.02))
SOLUTION
BP-Sn5
Traceability Analysis Result in the RPS BP SDS V&V Report (Ref. KNICS-RPS-SVR131-01 (Rev.02))
SOLUTION
BP-Sn6
Formal verification result for SDS with respect to the same safety properties as those of SRS (Ref. KNICS-RPS-SVR131-01 (Rev.01))
SOLUTION

BP-G4, BP-G5, BP-G6 에서 "바람직한(desired) 모든 안전 요구사항" 표현이 정확히 동일한 요구 사항 집합을 의미하는지 또는 "설계 명세에 대한", "소프트웨어 요구 명세에 대한", "소프트웨어 설계 명세에 대한" 모든 바람직한 안전 요구사항으로 각각 해석되어야 하는지 불분명하다.

부정적인 표현 "누락되지 않는다(not missed)"보다 긍정적인 표현, 예를 들어, "~은 [충분히 -> 컨텍스트에서 정의 되어야 함] 다루어진다" 정도가 더 좋을 것 같다. 어떤 수준의 활동이 "누락되지 않는"것을 의미하는지 명확하지 않다.

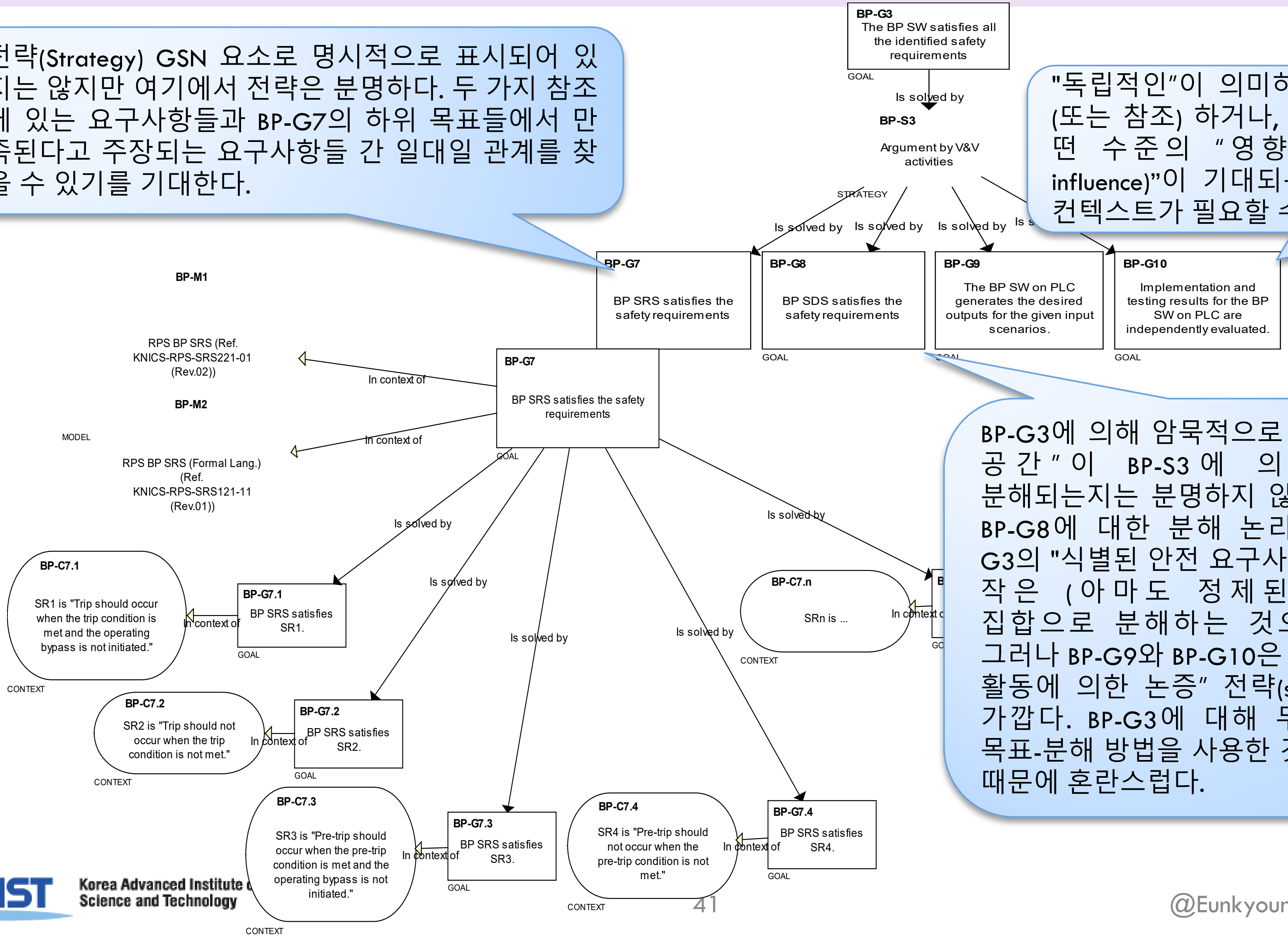
Goal BP-G6의 어느 부분이 BP-Sn5에 제시된 추적성 분석에 의해 해결되고, 어느 부분이 BP-Sn6에 제시된 정형 검증 결과에 의해 해결되는지 분명하지 않다.

추적성 분석 결과는 BP-G5에서의 주장에 대한 올바른 종류의 증거로 보인다.

Review on Argument by V&V

전략(Strategy) GSN 요소로 명시적으로 표시되어 있지는 않지만 여기에서 전략은 분명하다. 두 가지 참조에 있는 요구사항들과 BP-G7의 하위 목표들에서 만족된다고 주장되는 요구사항들 간 일대일 관계를 찾을 수 있기를 기대한다.

"독립적인"이 의미하는 것을 정의(또는 참조) 하거나, 검토자에게 어떤 수준의 "영향력 없음(non-influence)"이 기대되는지를 기술한 컨텍스트가 필요할 수 있다.



BP-G3
The BP SW satisfies all the identified safety requirements
GOAL

Is solved by

BP-S3
Argument by V&V activities
STRATEGY

Is solved by
Is solved by
Is solved by
Is s

BP-G7
BP SRS satisfies the safety requirements

BP-G8
BP SDS satisfies the safety requirements
GOAL

BP-G9
The BP SW on PLC generates the desired outputs for the given input scenarios.
GOAL

BP-G10
Implementation and testing results for the BP SW on PLC are independently evaluated.
GOAL

BP-M1
RPS BP SRS (Ref. KNICS-RPS-SRS221-01 (Rev.02))

BP-M2
MODEL
RPS BP SRS (Formal Lang.) (Ref. KNICS-RPS-SRS121-11 (Rev.01))

In context of
In context of

BP-G7
BP SRS satisfies the safety requirements
GOAL

Is solved by
Is solved by
Is solved by
Is solved by
Is solved by

BP-C7.1
SR1 is "Trip should occur when the trip condition is met and the operating bypass is not initiated."
CONTEXT

In context of

BP-G7.1
BP SRS satisfies SR1.
GOAL

BP-C7.2
SR2 is "Trip should not occur when the trip condition is not met."
CONTEXT

In context of

BP-G7.2
BP SRS satisfies SR2.
GOAL

BP-C7.3
SR3 is "Pre-trip should occur when the pre-trip condition is met and the operating bypass is not initiated."
CONTEXT

In context of

BP-G7.3
BP SRS satisfies SR3.
GOAL

BP-C7.4
SR4 is "Pre-trip should not occur when the pre-trip condition is not met."
CONTEXT

In context of

BP-G7.4
BP SRS satisfies SR4.
GOAL

BP-C7.n
SRn is ...
CONTEXT

In context o

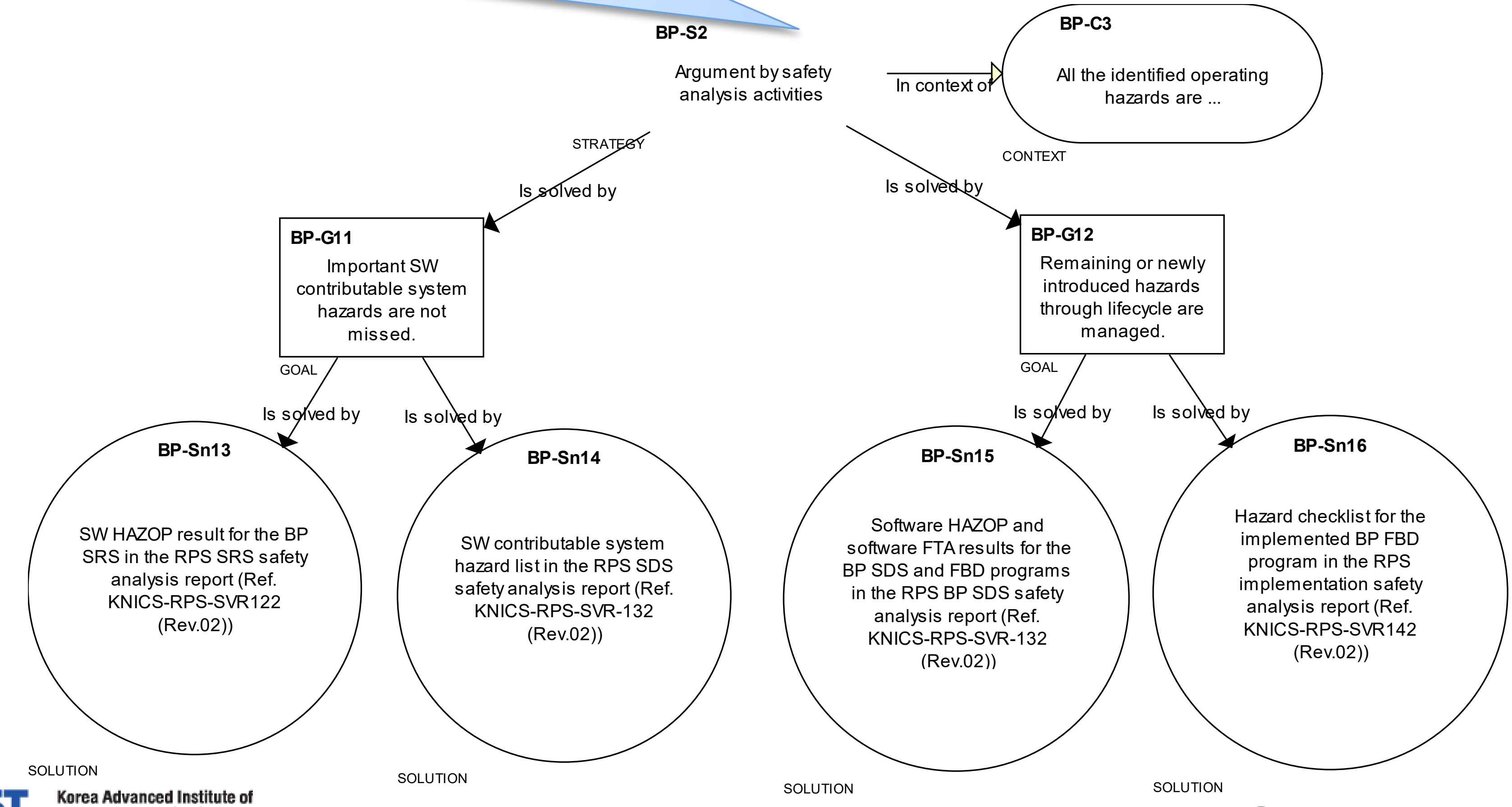
B

GC

BP-G3에 의해 암묵적으로 정의된 "문제 공간"이 BP-S3에 의해 어떻게 분해되는지는 분명하지 않다. BP-G7 및 BP-G8에 대한 분해 논리/전략은 BP-G3의 "식별된 안전 요구사항" 집합을 더 작은 (아마도 정제된) 요구사항 집합으로 분해하는 것으로 보인다. 그러나 BP-G9와 BP-G10은 BP-S3의 "V&V 활동에 의한 논증" 전략(strategy)에 더 가깝다. BP-G3에 대해 두 가지 다른 목표-분해 방법을 사용한 것으로 보이기 때문에 혼란스럽다.

Review on Argument by Safety Analysis

BP-S2의 해석이 불분명하며 "문제 공간"이 어떻게 분리되는지는 분명하지 않다. BP-G11과 BP-G12 두 목표가 BP-S2에 의해 정의된 "문제 공간"을 완전히 커버하고 있다고 평가할 수 있는 방법은 무엇인가?



BP-S2

Argument by safety analysis activities

STRATEGY

In context of

BP-C3

All the identified operating hazards are ...

CONTEXT

Is solved by

Is solved by

BP-G11

Important SW contributable system hazards are not missed.

GOAL

BP-G12

Remaining or newly introduced hazards through lifecycle are managed.

GOAL

Is solved by

Is solved by

Is solved by

Is solved by

BP-Sn13

SW HAZOP result for the BP SRS in the RPS SRS safety analysis report (Ref. KNICS-RPS-SVR122 (Rev.02))

SOLUTION

BP-Sn14

SW contributable system hazard list in the RPS SDS safety analysis report (Ref. KNICS-RPS-SVR-132 (Rev.02))

SOLUTION

BP-Sn15

Software HAZOP and software FTA results for the BP SDS and FBD programs in the RPS BP SDS safety analysis report (Ref. KNICS-RPS-SVR-132 (Rev.02))

SOLUTION

BP-Sn16

Hazard checklist for the implemented BP FBD program in the RPS implementation safety analysis report (Ref. KNICS-RPS-SVR142 (Rev.02))

SOLUTION

Lessons Learned

- Safety case can support efficient communication during the licensing process.
  - It enables clearer and more efficient communication with a focus on system safety between the developer and the regulator.
- Necessary to describe the context information as specific as possible when the common understanding b/w the developer and the regulator is insufficient
  - Specific and clear descriptive information on the terms, strategies and assumptions used is required so that the regulators can understand it accurately.
- Multiple cycles of construction and review may be needed.
  - Safety cases can be clarified through repetition.
  - Additional time and effort is required.
- Should exclude problems caused by ambiguity and subjective interpretation
  - by using objective, formal and quantitative phrases and evidence.

Thank you for your attention.

QUESTIONS?