ECSEL Joint Undertaking

European Commission

AMASS

Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems

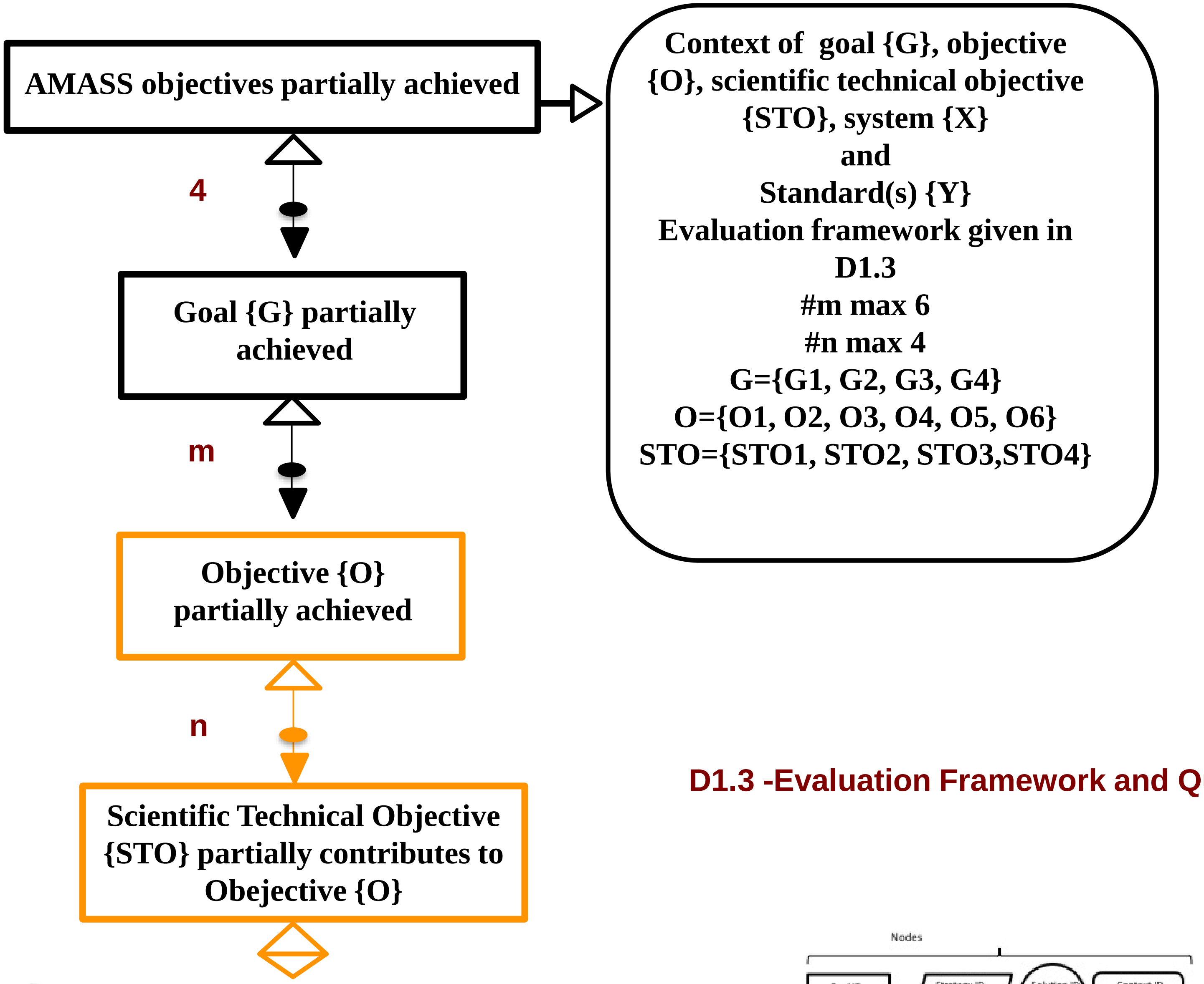
Technical Vision

Second EAB Workshop
Västerås, Sept 17, 2018

Barbara Gallina, Ph.D.
TM, WP6 Leader, T6.1-2 Leader

MÄLARDALEN UNIVERSITY SWEDEN

AMASS technical contribution



AMASS objectives partially achieved

4

Goal {G} partially achieved

m

Objective {O} partially achieved

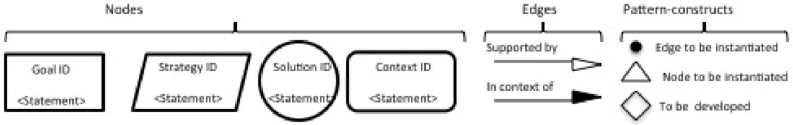
n

Scientific Technical Objective {STO} partially contributes to Obejective {O}

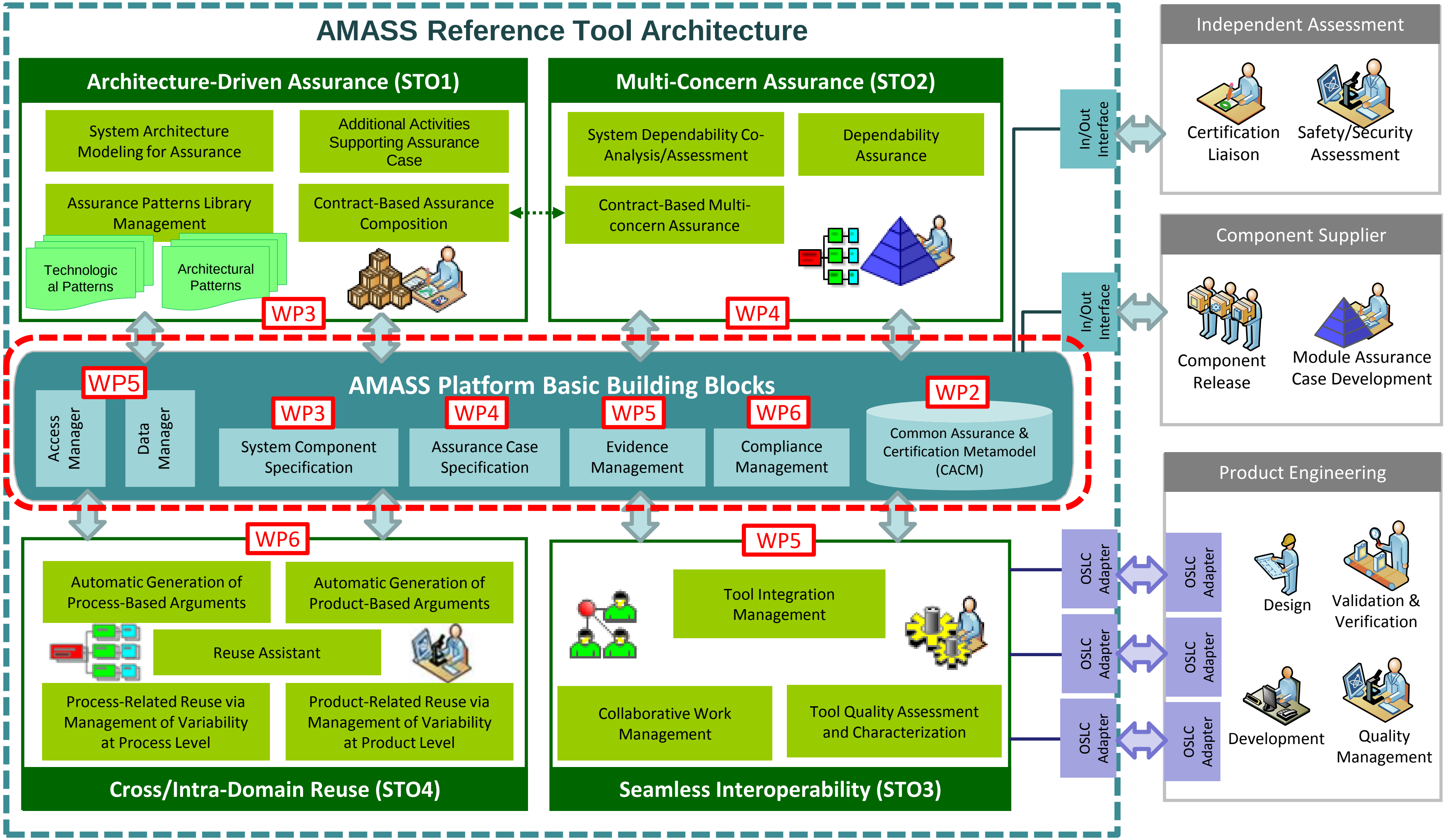
Context of goal {G}, objective {O}, scientific technical objective {STO}, system {X}
and
Standard(s) {Y}
Evaluation framework given in D1.3
#m max 6
#n max 4
G={G1, G2, G3, G4}
O={O1, O2, O3, O4, O5, O6}
STO={STO1, STO2, STO3,STO4}

Claim, claim decomposition, evidence

D1.3 -Evaluation Framework and Quality Metrics



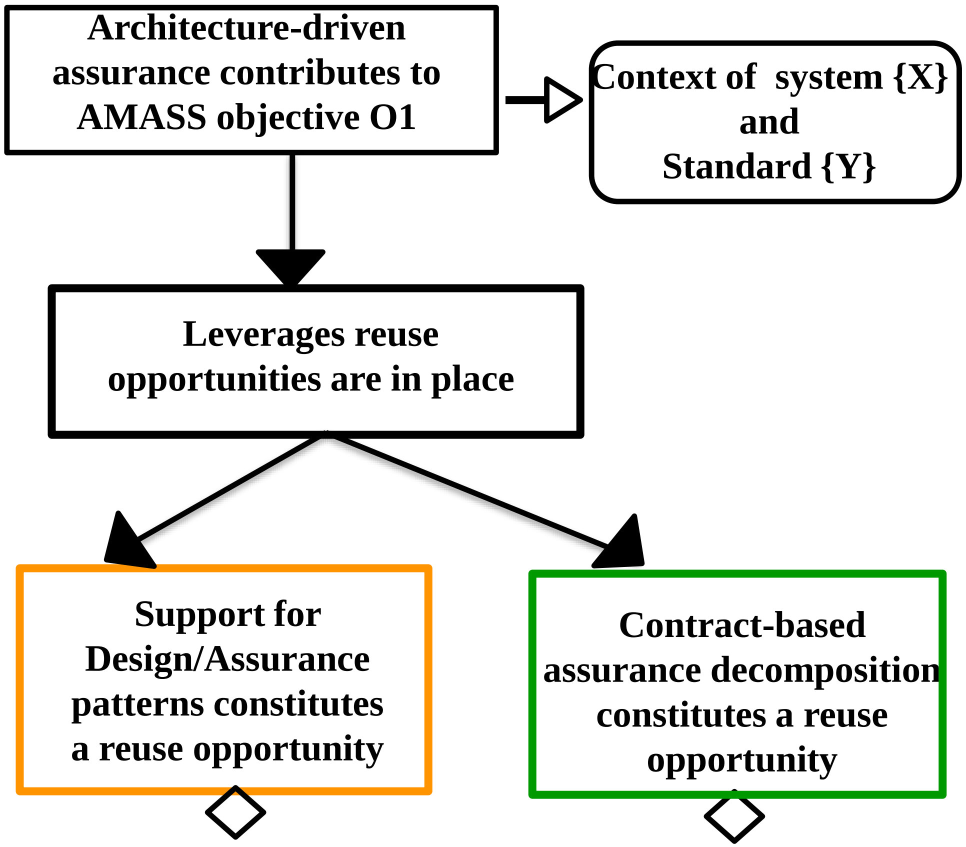
AMASS Reference Tool Architecture



AMASS Reference Tool Architecture
Architecture-Driven Assurance (STO1)
System Architecture Modeling for Assurance
Additional Activities Supporting Assurance Case
Assurance Patterns Library Management
Contract-Based Assurance Composition
Technological Patterns
Architectural Patterns
WP3
Multi-Concern Assurance (STO2)
System Dependability Co-Analysis/Assessment
Dependability Assurance
Contract-Based Multi-concern Assurance
WP4
In/Out Interface
In/Out Interface
Independent Assessment
Certification Liaison
Safety/Security Assessment
Component Supplier
Component Release
Module Assurance Case Development
AMASS Platform Basic Building Blocks
WP5
Access Manager
Data Manager
WP3
System Component Specification
WP4
Assurance Case Specification
WP5
Evidence Management
WP6
Compliance Management
WP2
Common Assurance & Certification Metamodel (CACM)
WP6
Automatic Generation of Process-Based Arguments
Automatic Generation of Product-Based Arguments
Reuse Assistant
Process-Related Reuse via Management of Variability at Process Level
Product-Related Reuse via Management of Variability at Product Level
Cross/Intra-Domain Reuse (STO4)
WP5
Tool Integration Management
Collaborative Work Management
Tool Quality Assessment and Characterization
Seamless Interoperability (STO3)
OSLC Adapter
OSLC Adapter
OSLC Adapter
OSLC Adapter
OSLC Adapter
OSLC Adapter
Product Engineering
Design
Validation & Verification
Development
Quality Management

Architecture-driven assurance

O1: define a holistic approach for ***architecture-driven assurance*** to leverage the reuse opportunities in assurance and certification by directly and explicitly addressing current technologies and HW/SW architectures needs.



Architecture-driven assurance contributes to AMASS objective O1

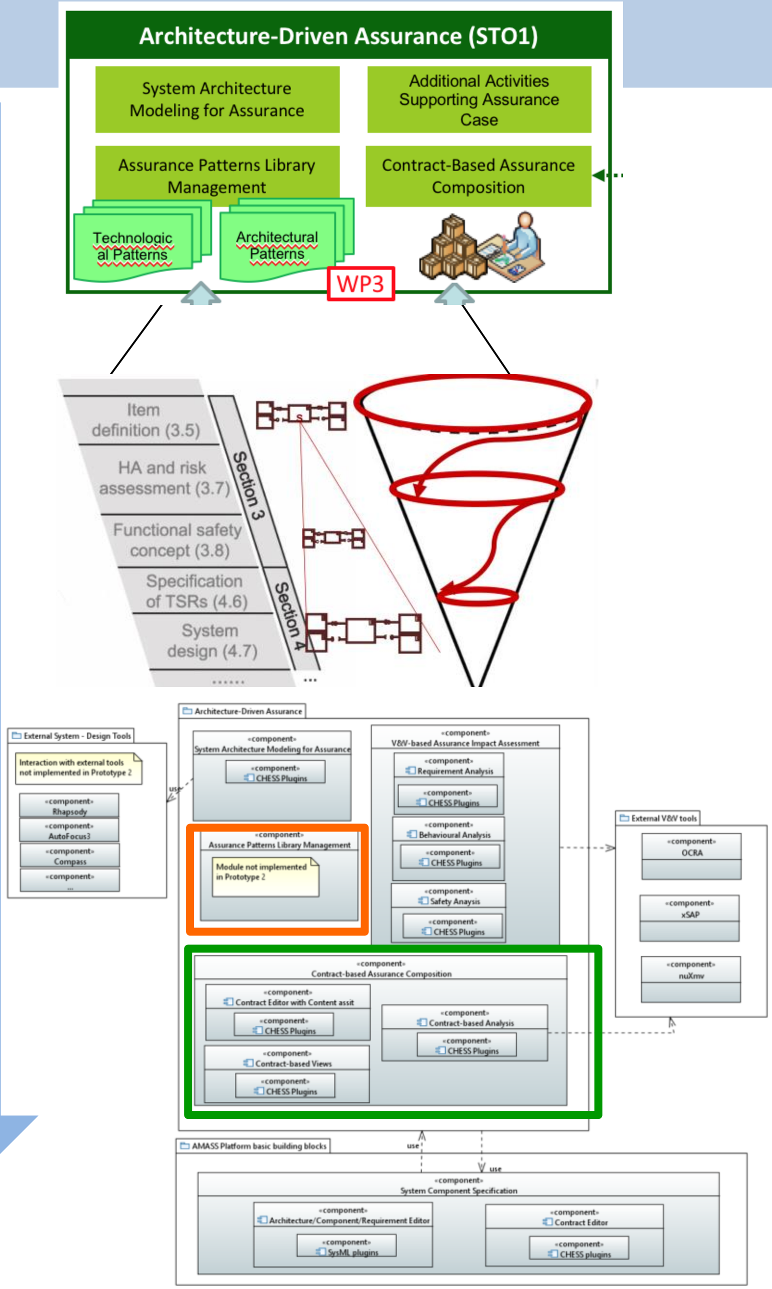
Context of system {X} and Standard {Y}

Leverages reuse opportunities are in place

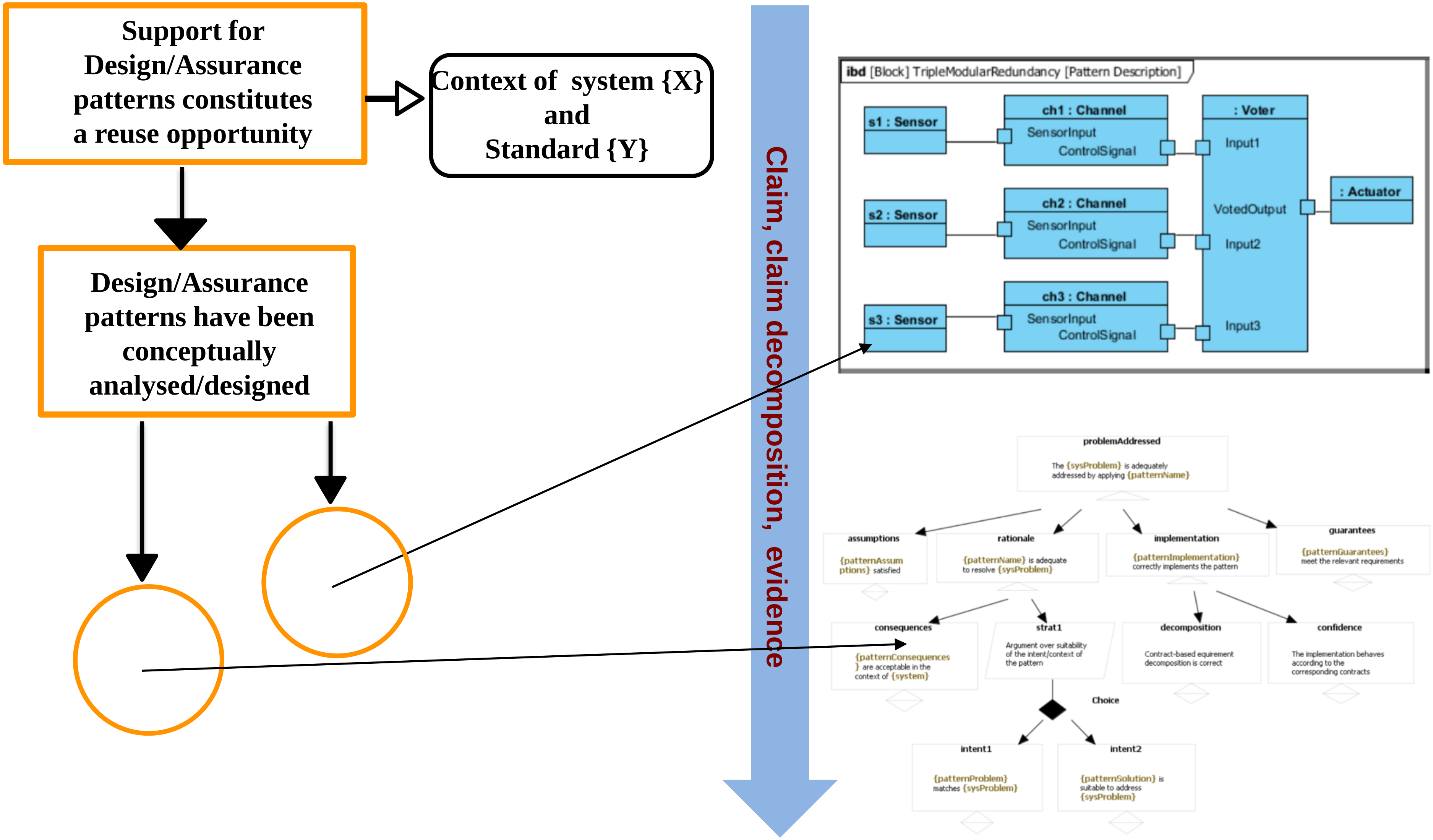
Support for Design/Assurance patterns constitutes a reuse opportunity

Contract-based assurance decomposition constitutes a reuse opportunity

Claim, claim decomposition, evidence


Architecture-Driven Assurance (STO1)
System Architecture Modeling for Assurance
Additional Activities Supporting Assurance Case
Assurance Patterns Library Management
Contract-Based Assurance Composition
Technological Patterns
Architectural Patterns
WP3
Item definition (3.5)
HA and risk assessment (3.7)
Functional safety concept (3.8)
Specification of TSRs (4.6)
System design (4.7)
Section 3
Section 4
External System - Design Tools
Interaction with external tools not implemented in Prototype 2
«component» Rhapsody
«component» AutoFocus3
«component» Compass
«component» ...
use
Architecture-Driven Assurance
«component» System Architecture Modeling for Assurance
«component» CHESS Plugins
«component» V&V-based Assurance Impact Assessment
«component» Requirement Analysis
«component» CHESS Plugins
«component» Behavioural Analysis
«component» CHESS Plugins
«component» Safety Anaysis
«component» CHESS Plugins
«component» Assurance Patterns Library Management
Module not implemented in Prototype 2
«component» Contract-based Assurance Composition
«component» Contract Editor with Content assit
«component» CHESS Plugins
«component» Contract-based Views
«component» CHESS Plugins
«component» Contract-based Analysis
«component» CHESS Plugins
External V&V tools
«component» OCRA
«component» xSAP
«component» nuXmv
AMASS Platform basic building blocks
use
use
«component» System Component Specification
«component» Architecture/Component/Requirement Editor
«component» SysML plugins
«component» Contract Editor
«component» CHESS plugins

Architecture-driven assurance



Support for Design/Assurance patterns constitutes a reuse opportunity

Context of system {X} and Standard {Y}

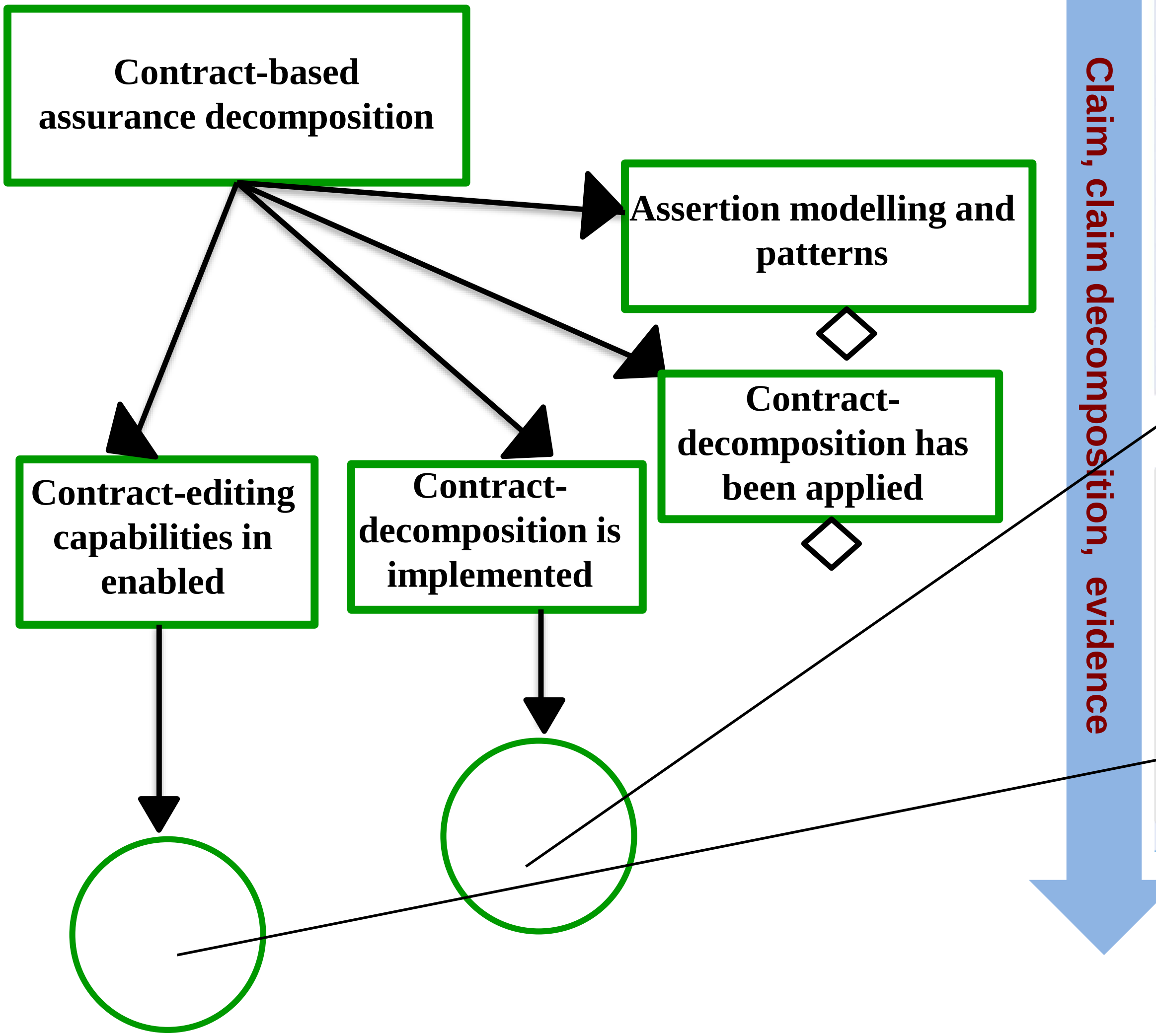
Design/Assurance patterns have been conceptually analysed/designed

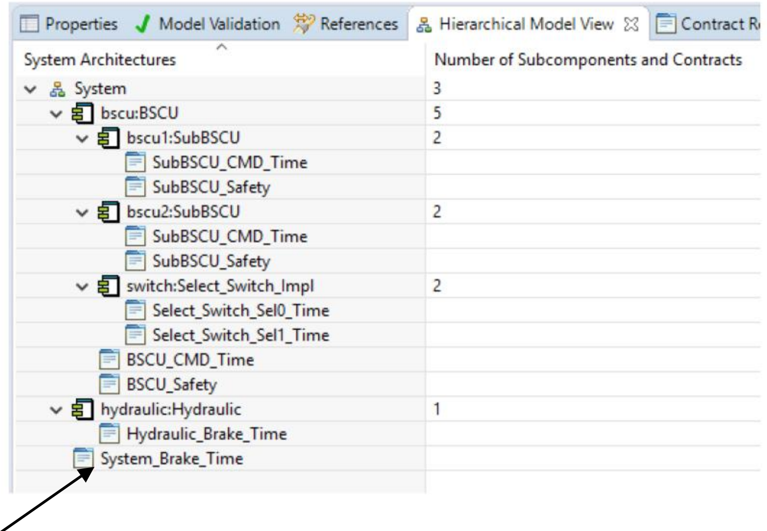
Claim, claim decomposition, evidence

ibd [Block] TripleModularRedundancy [Pattern Description]
s1 : Sensor
s2 : Sensor
s3 : Sensor
ch1 : Channel
ch2 : Channel
ch3 : Channel
SensorInput
ControlSignal
: Voter
Input1
Input2
Input3
VotedOutput
: Actuator
problemAddressed
The {sysProblem} is adequately addressed by applying {patternName}
assumptions
{patternAssumptions} satisfied
rationale
{patternName} is adequate to resolve {sysProblem}
implementation
{patternImplementation} correctly implements the pattern
guarantees
{patternGuarantees} meet the relevant requirements
consequences
{patternConsequences} are acceptable in the context of {system}
strat1
Argument over suitability of the intent/context of the pattern
decomposition
Contract-based equirement decomposition is correct
confidence
The implementation behaves according to the corresponding contracts
Choice
intent1
{patternProblem} matches {sysProblem}
intent2
{patternSolution} is suitable to address {sysProblem}

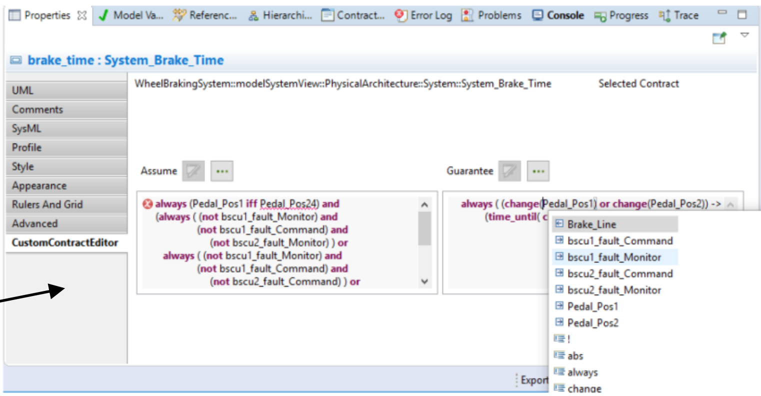
Architecture-driven assurance – Evidence on …

O1: define a holistic approach for ***architecture-driven assurance*** to leverage the reuse opportunities in assurance and certification by directly and explicitly addressing current technologies and HW/SW architectures needs.


Contract-based assurance decomposition
Assertion modelling and patterns
Contract-decomposition has been applied
Contract-editing capabilities in enabled
Contract-decomposition is implemented
Claim, claim decomposition, evidence



System Architectures	Number of Subcomponents and Contracts
System	3
bscu:BSCU	5
bscu1:SubBSCU	2
SubBSCU_CMD_Time	
SubBSCU_Safety	
bscu2:SubBSCU	2
SubBSCU_CMD_Time	
SubBSCU_Safety	
switch:Select_Switch_Impl	2
Select_Switch_Sel0_Time	
Select_Switch_Sel1_Time	
BSCU_CMD_Time	
BSCU_Safety	
hydraulic:Hydraulic	1
Hydraulic_Brake_Time	
System_Brake_Time	



brake_time : System_Brake_Time

WheelBrakingSystem::modelSystemView::PhysicalArchitecture::System::System_Brake_Time

Selected Contract

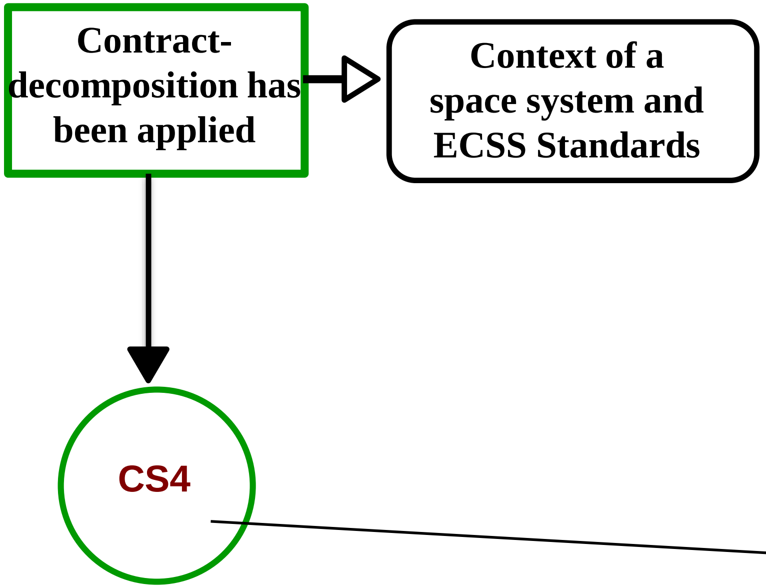
Assume

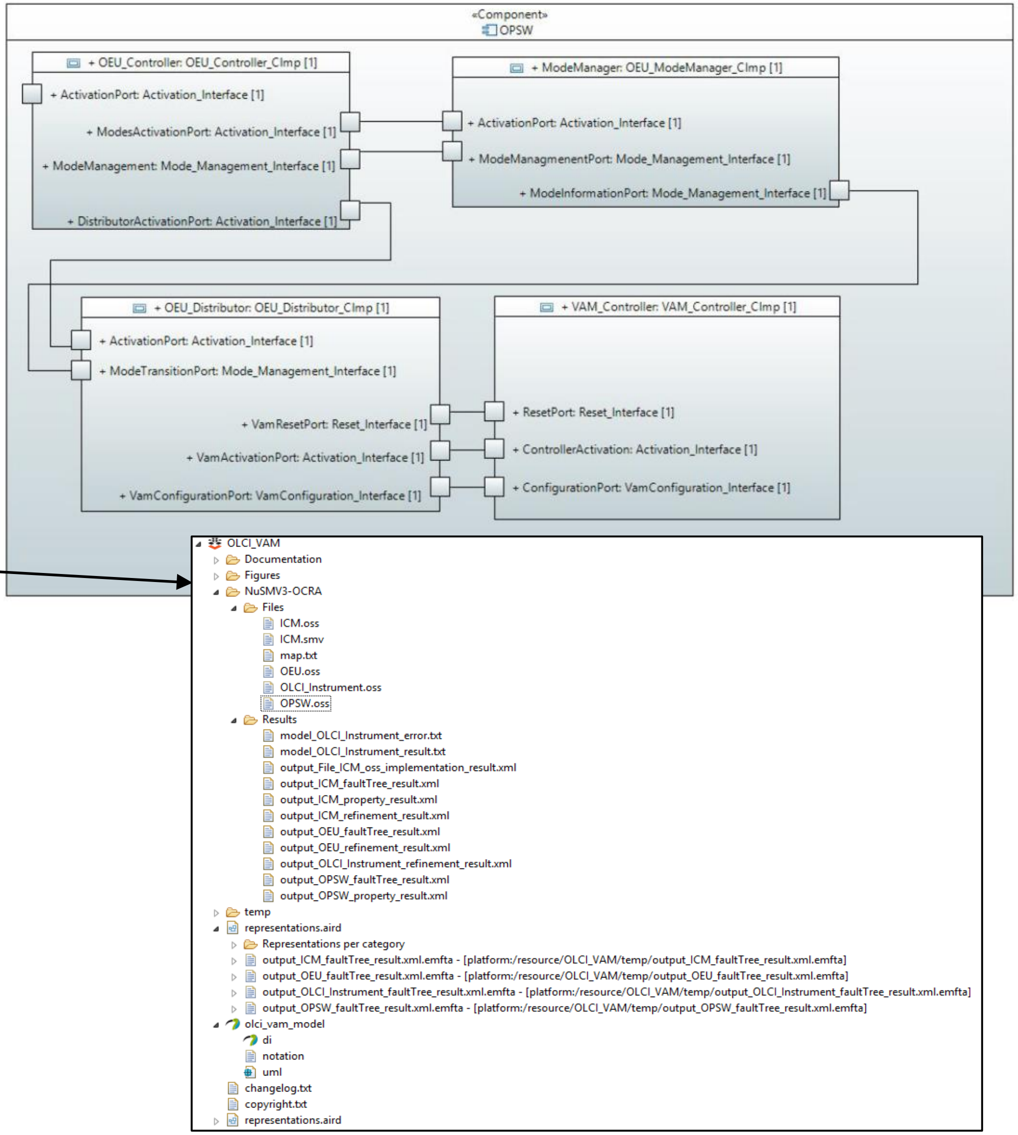
```
always (Pedal_Pos1 iff Pedal_Pos24) and
 (always ( (not bscu1_fault_Monitor) and
          (not bscu1_fault_Command) and
            (not bscu2_fault_Monitor) ) or
     always ( (not bscu1_fault_Monitor) and
          (not bscu1_fault_Command) and
            (not bscu2_fault_Command) ) or
```

Guarantee

```
always ( (change(Pedal_Pos1) or change(Pedal_Pos2)) ->
     (time_until( c
```

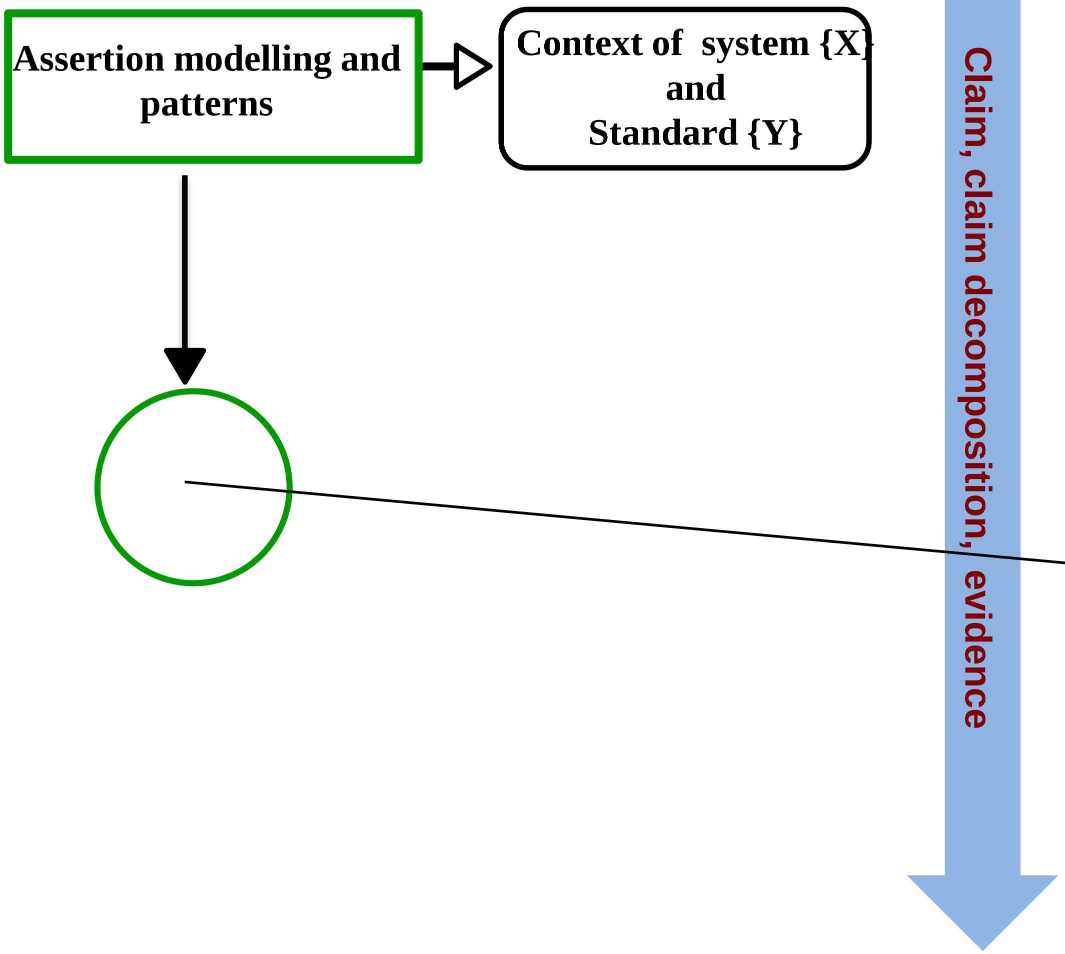
Architecture-driven assurance – Evidence on …


Contract-decomposition has been applied
Context of a space system and ECSS Standards
CS4


«Component»
OPSW
+ OEU_Controller: OEU_Controller_CImp [1]
+ ActivationPort: Activation_Interface [1]
+ ModesActivationPort: Activation_Interface [1]
+ ModeManagement: Mode_Management_Interface [1]
+ DistributorActivationPort: Activation_Interface [1]
+ ModeManager: OEU_ModeManager_CImp [1]
+ ActivationPort: Activation_Interface [1]
+ ModeManagmenentPort: Mode_Management_Interface [1]
+ ModeInformationPort: Mode_Management_Interface [1]
+ OEU_Distributor: OEU_Distributor_CImp [1]
+ ActivationPort: Activation_Interface [1]
+ ModeTransitionPort: Mode_Management_Interface [1]
+ VamResetPort: Reset_Interface [1]
+ VamActivationPort: Activation_Interface [1]
+ VamConfigurationPort: VamConfiguration_Interface [1]
+ VAM_Controller: VAM_Controller_CImp [1]
+ ResetPort: Reset_Interface [1]
+ ControllerActivation: Activation_Interface [1]
+ ConfigurationPort: VamConfiguration_Interface [1]

```
OLCI_VAM
  Documentation
  Figures
  NuSMV3-OCRA
    Files
      ICM.oss
      ICM.smv
      map.txt
      OEU.oss
      OLCI_Instrument.oss
      OPSW.oss
    Results
      model_OLCI_Instrument_error.txt
      model_OLCI_Instrument_result.txt
      output_File_ICM_oss_implementation_result.xml
      output_ICM_faultTree_result.xml
      output_ICM_property_result.xml
      output_ICM_refinement_result.xml
      output_OEU_faultTree_result.xml
      output_OEU_refinement_result.xml
      output_OLCI_Instrument_refinement_result.xml
      output_OPSW_faultTree_result.xml
      output_OPSW_property_result.xml
  temp
  representations.aird
    Representations per category
    output_ICM_faultTree_result.xml.emfta - [platform:/resource/OLCI_VAM/temp/output_ICM_faultTree_result.xml.emfta]
    output_OEU_faultTree_result.xml.emfta - [platform:/resource/OLCI_VAM/temp/output_OEU_faultTree_result.xml.emfta]
    output_OLCI_Instrument_faultTree_result.xml.emfta - [platform:/resource/OLCI_VAM/temp/output_OLCI_Instrument_faultTree_result.xml.emfta]
    output_OPSW_faultTree_result.xml.emfta - [platform:/resource/OLCI_VAM/temp/output_OPSW_faultTree_result.xml.emfta]
  olci_vam_model
    di
    notation
    uml
  changelog.txt
  copyright.txt
  representations.aird
```

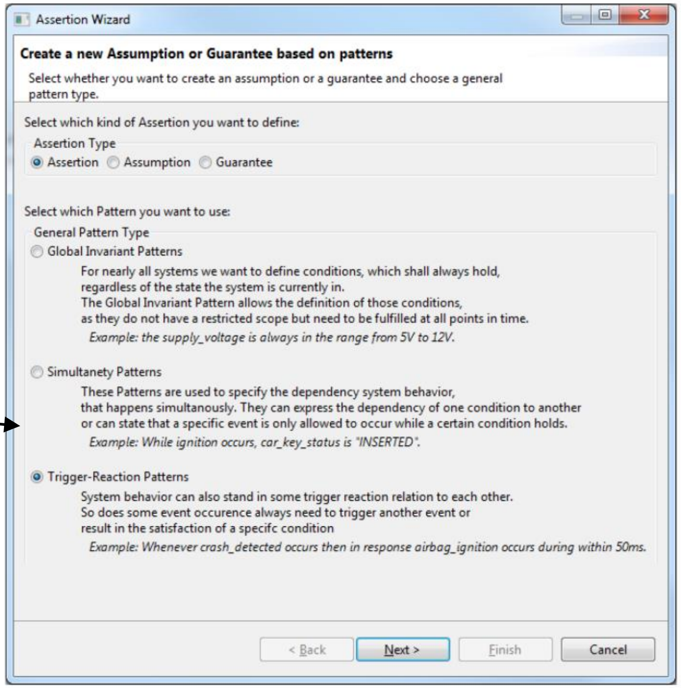
Architecture-driven assurance – Evidence on …



Assertion modelling and patterns

Context of system {X} and Standard {Y}

Claim, claim decomposition, evidence



Create a new Assumption or Guarantee based on patterns

Select whether you want to create an assumption or a guarantee and choose a general pattern type.

Select which kind of Assertion you want to define:

Assertion Type

● Assertion ○ Assumption ○ Guarantee

Select which Pattern you want to use:

General Pattern Type

○ Global Invariant Patterns

For nearly all systems we want to define conditions, which shall always hold, regardless of the state the system is currently in.
The Global Invariant Pattern allows the definition of those conditions, as they do not have a restricted scope but need to be fulfilled at all points in time.

Example: the supply_voltage is always in the range from 5V to 12V.

○ Simultanety Patterns

These Patterns are used to specify the dependency system behavior, that happens simultanously. They can express the dependency of one condition to another or can state that a specific event is only allowed to occur while a certain condition holds.

Example: While ignition occurs, car_key_status is "INSERTED".

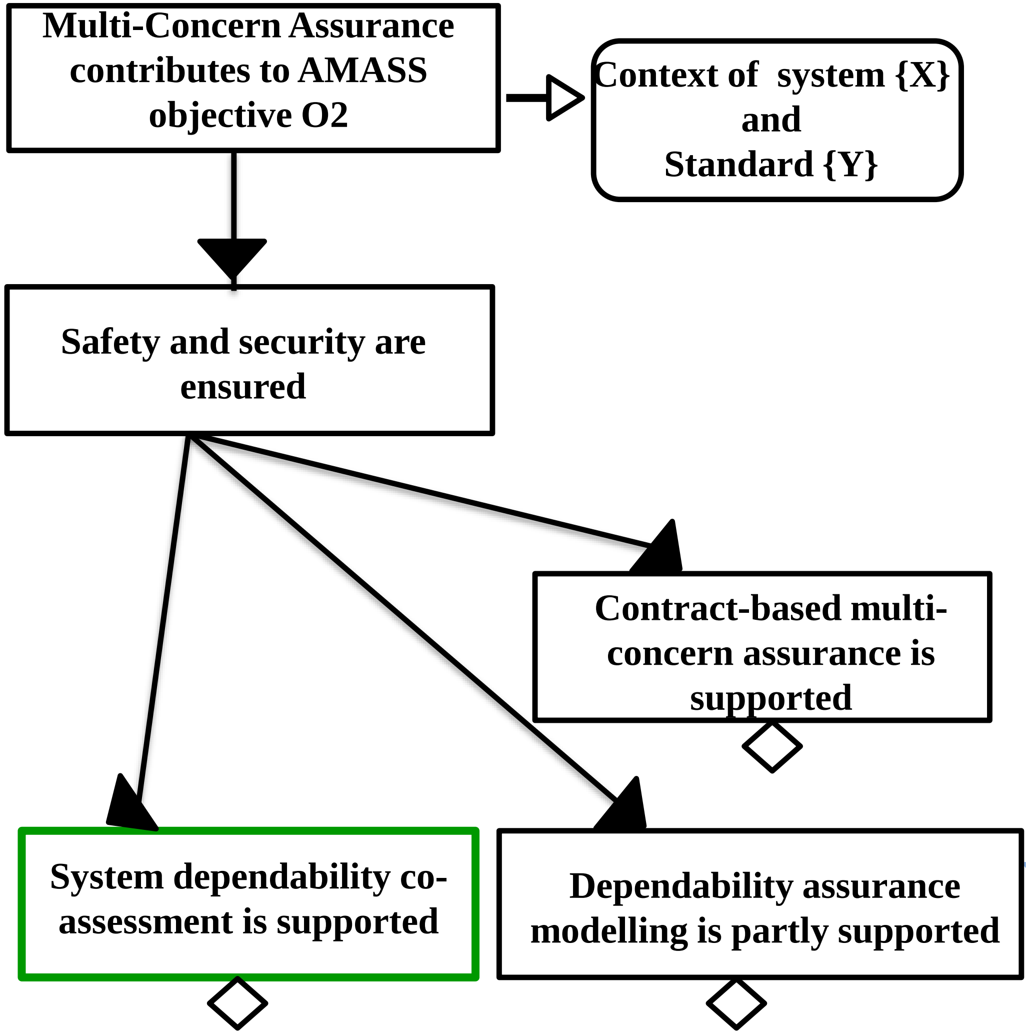
● Trigger-Reaction Patterns

System behavior can also stand in some trigger reaction relation to each other. So does some event occurence always need to trigger another event or result in the satisfaction of a specifc condition

Example: Whenever crash_detected occurs then in response airbag_ignition occurs during within 50ms.

Multi-Concern Assurance

O2: define a *multi-concern assurance* approach to ensure not only safety and security but also other dependability aspects such as availability, robustness and reliability.



Multi-Concern Assurance contributes to AMASS objective O2

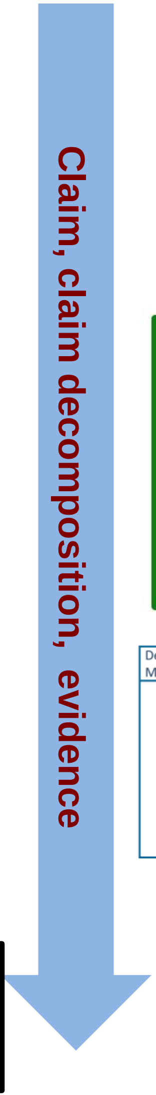
Context of system {X} and Standard {Y}

Safety and security are ensured

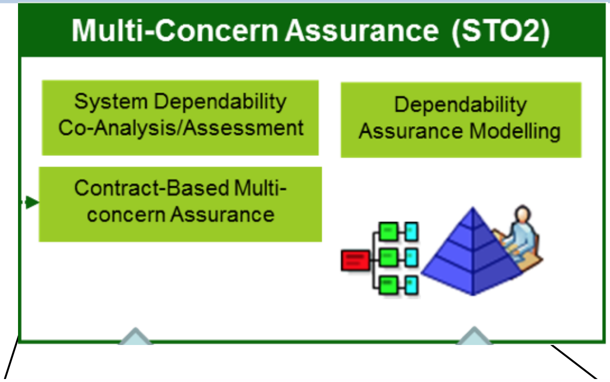
Contract-based multi-concern assurance is supported

System dependability co-assessment is supported

Dependability assurance modelling is partly supported



Claim, claim decomposition, evidence



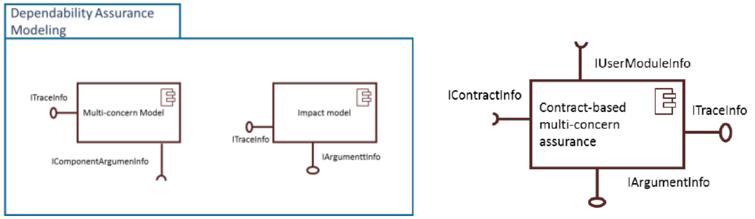
Multi-Concern Assurance (STO2)

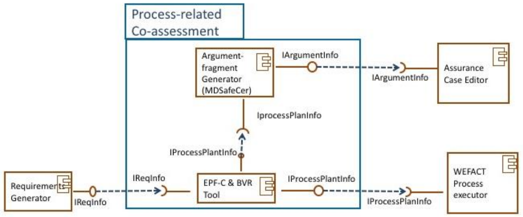
System Dependability Co-Analysis/Assessment

Dependability Assurance Modelling

Contract-Based Multi-concern Assurance

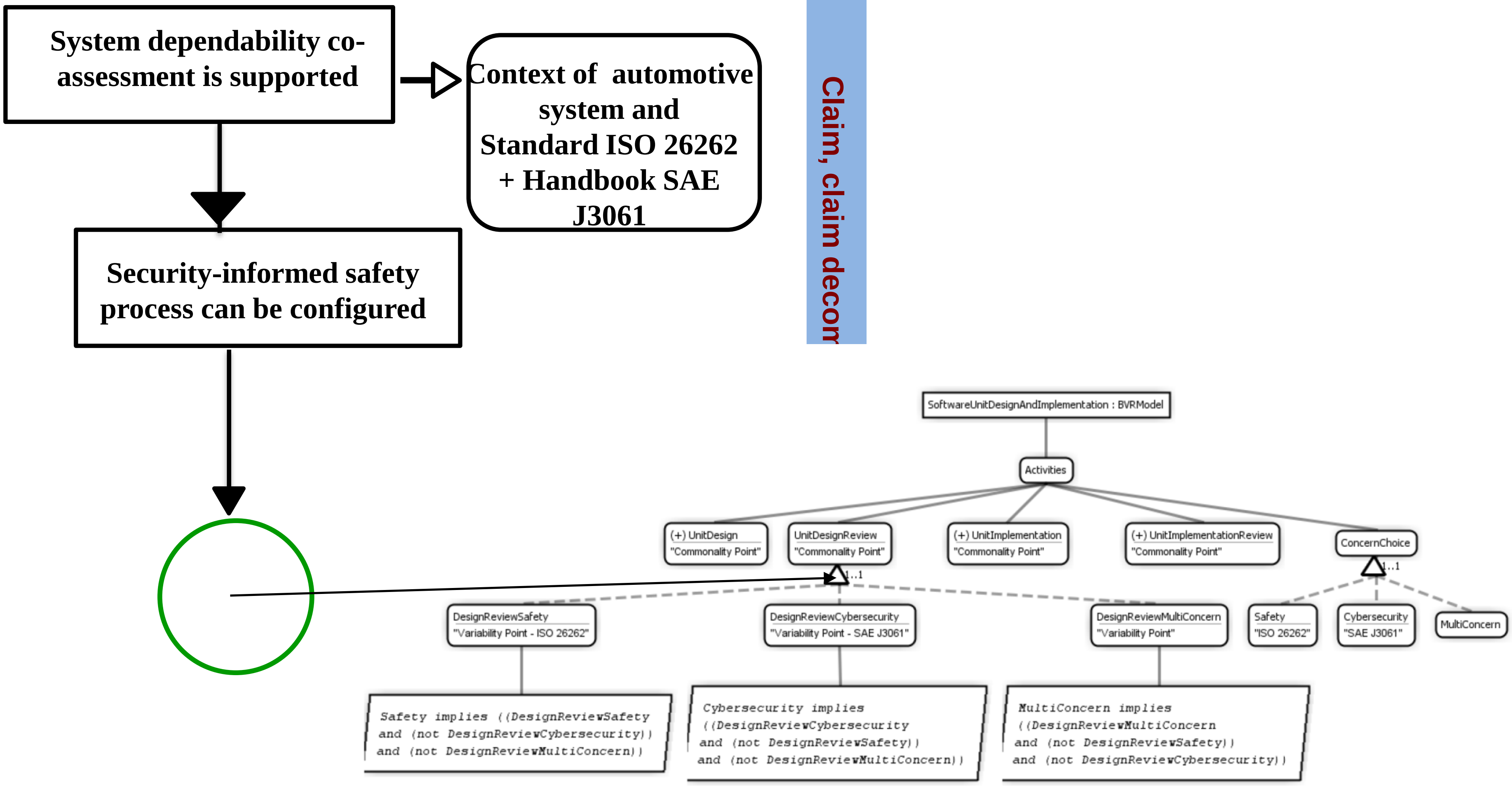
- Process engineer(s) addressing the security & safety process
- Architect jointly interacting with safety and security managers





Multi-Concern Assurance – Evidence on …

O2: define a *multi-concern assurance* approach to ensure not only safety and security but also other dependability aspects such as availability, robustness and reliability.



System dependability co-assessment is supported

Context of automotive system and Standard ISO 26262 + Handbook SAE J3061

Security-informed safety process can be configured

Claim, claim decom

SoftwareUnitDesignAndImplementation : BVRModel
Activities
(+) UnitDesign "Commonality Point"
UnitDesignReview "Commonality Point"
(+) UnitImplementation "Commonality Point"
(+) UnitImplementationReview "Commonality Point"
ConcernChoice
1..1
1..1
DesignReviewSafety "Variability Point - ISO 26262"
DesignReviewCybersecurity "Variability Point - SAE J3061"
DesignReviewMultiConcern "Variability Point"
Safety "ISO 26262"
Cybersecurity "SAE J3061"
MultiConcern

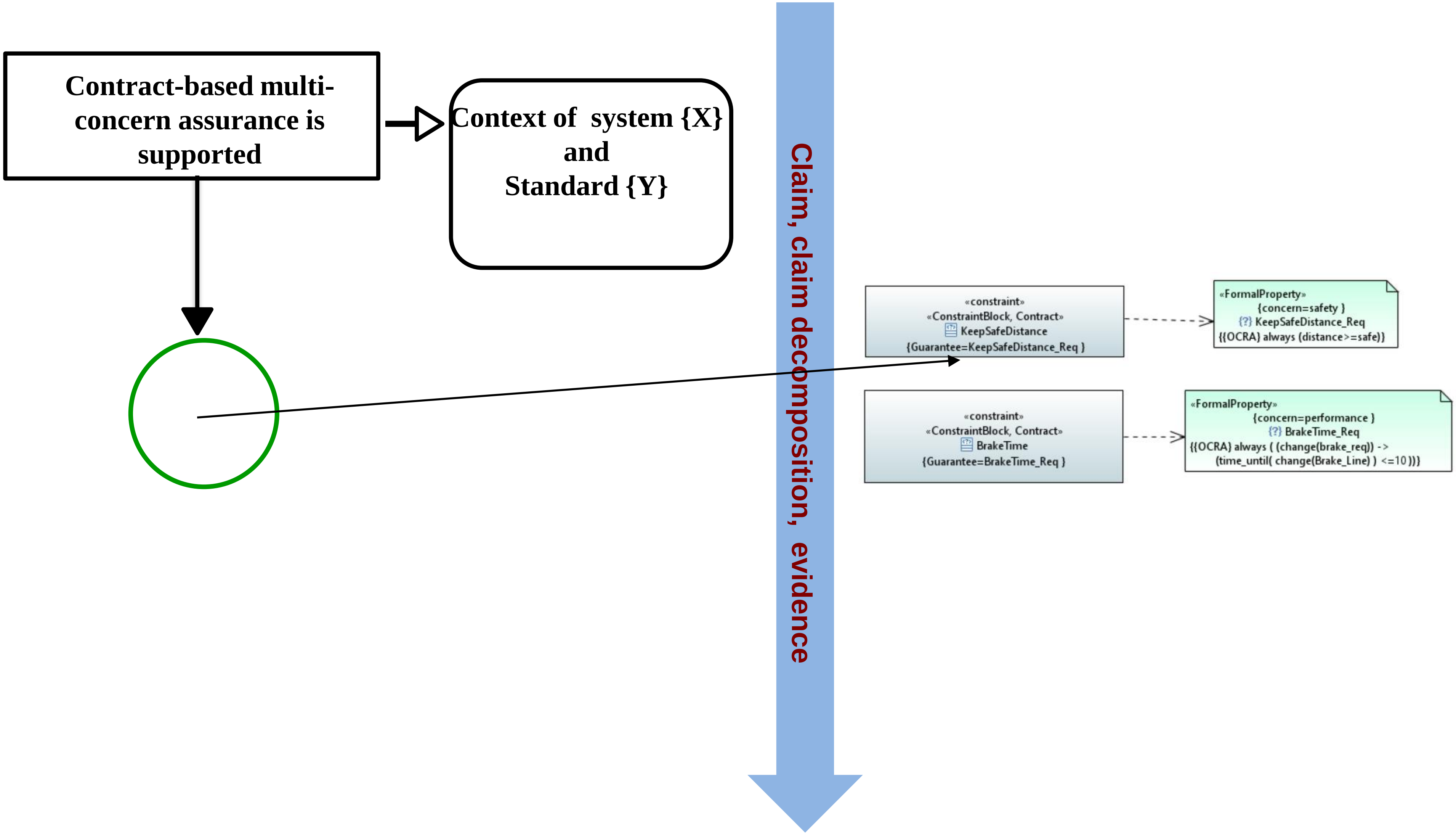
```
Safety implies ((DesignReviewSafety
and (not DesignReviewCybersecurity))
and (not DesignReviewMultiConcern))
```

```
Cybersecurity implies
((DesignReviewCybersecurity
and (not DesignReviewSafety))
and (not DesignReviewMultiConcern))
```

```
MultiConcern implies
((DesignReviewMultiConcern
and (not DesignReviewSafety))
and (not DesignReviewCybersecurity))
```

Multi-Concern Assurance – Evidence on …

O2: define a *multi-concern assurance* approach to ensure not only safety and security but also other dependability aspects such as availability, robustness and reliability.



Contract-based multi-concern assurance is supported

Context of system {X} and Standard {Y}

Claim, claim decomposition, evidence

«constraint»
«ConstraintBlock, Contract»
KeepSafeDistance
{Guarantee=KeepSafeDistance_Req }

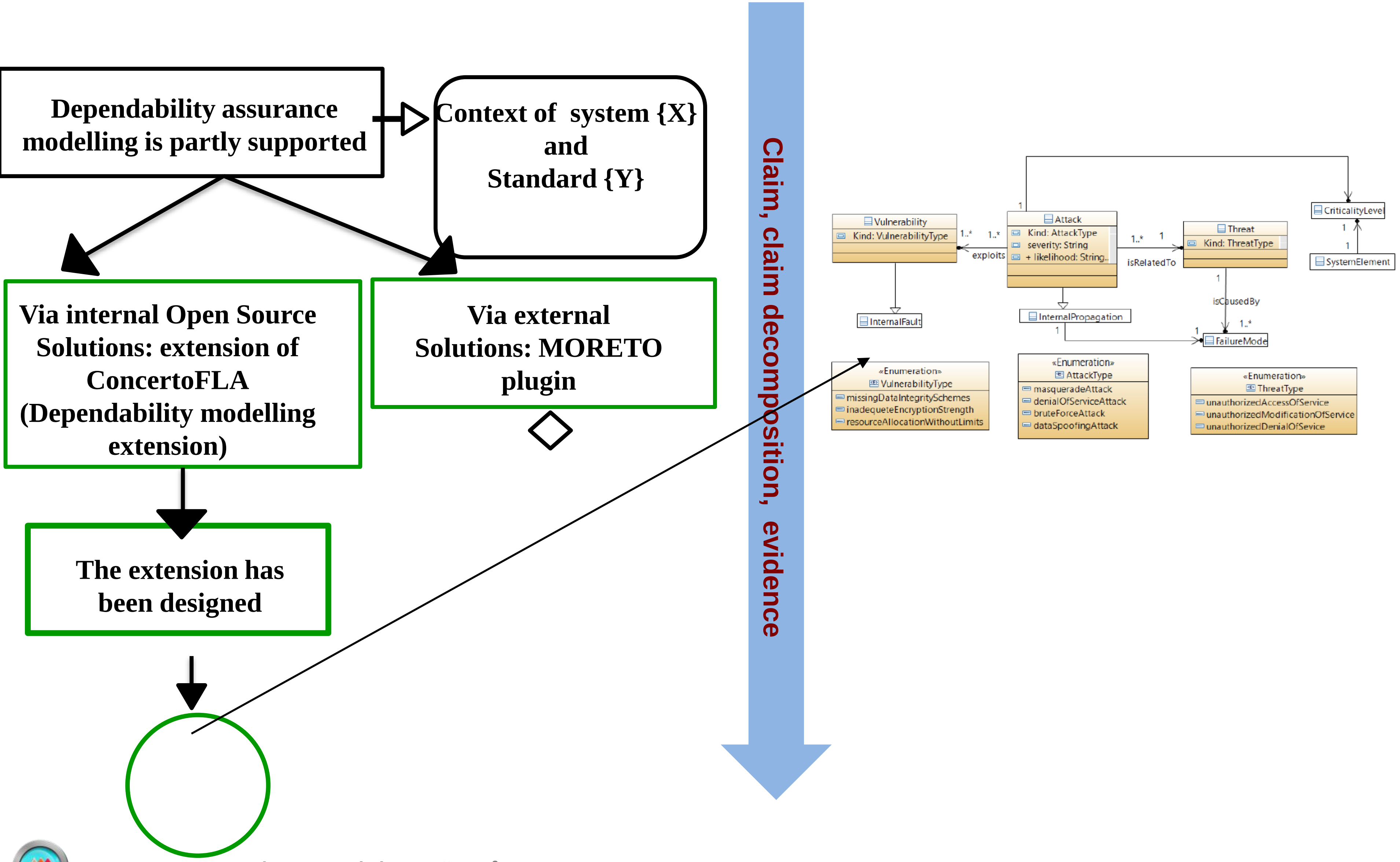
«FormalProperty»
{concern=safety }
{?} KeepSafeDistance_Req
{{OCRA} always (distance>=safe)}

«constraint»
«ConstraintBlock, Contract»
BrakeTime
{Guarantee=BrakeTime_Req }

«FormalProperty»
{concern=performance }
{?} BrakeTime_Req
{{OCRA} always ((change(brake_req)) ->
(time_until(change(Brake_Line)) <=10))}

Multi-Concern Assurance – Evidence on …

O2: define a *multi-concern assurance* approach to ensure not only safety and security but also other dependability aspects such as availability, robustness and reliability.



Dependability assurance modelling is partly supported

Context of system {X} and Standard {Y}

Via internal Open Source Solutions: extension of ConcertoFLA (Dependability modelling extension)

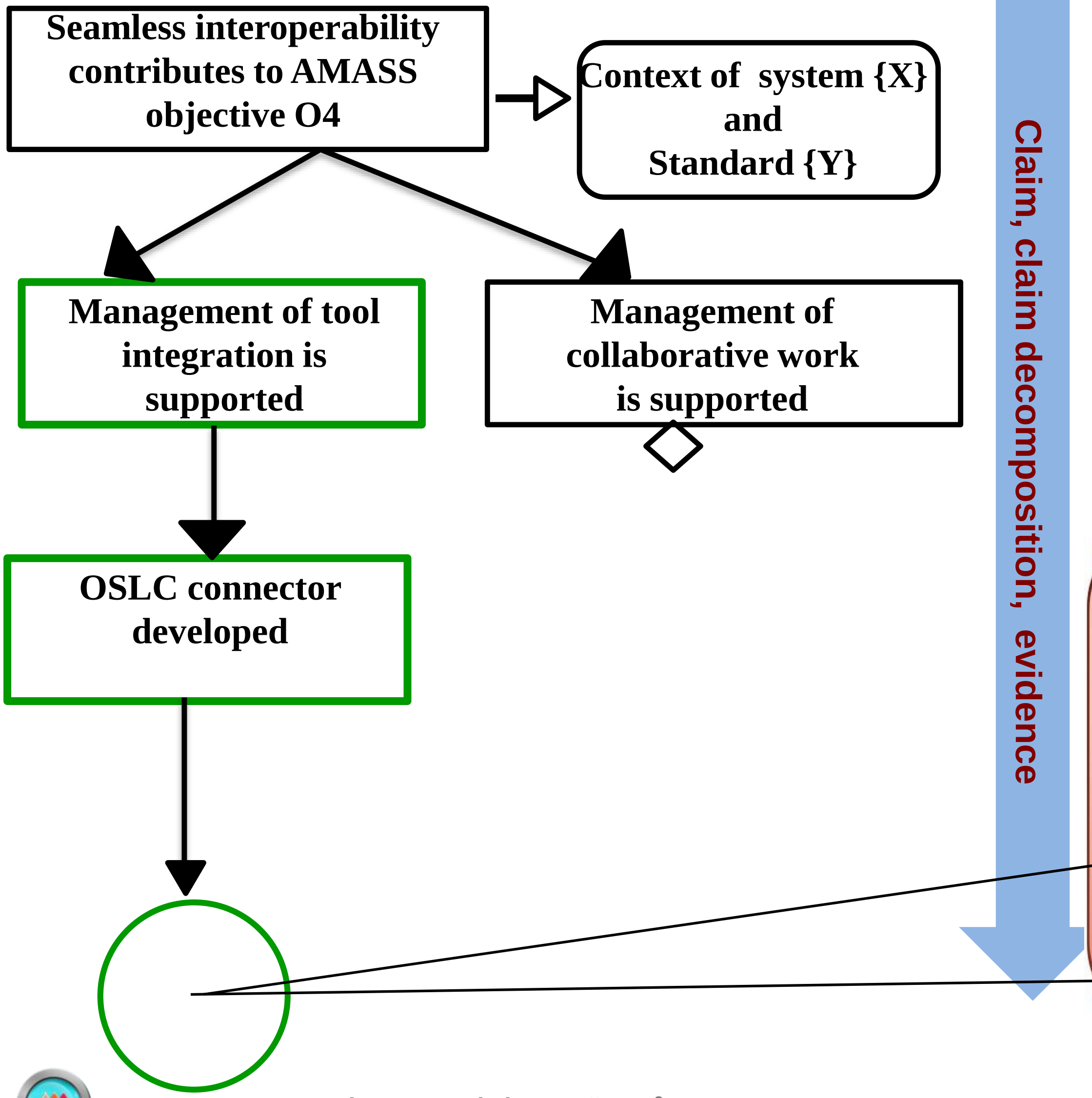
Via external Solutions: MORETO plugin

The extension has been designed

Claim, claim decomposition, evidence

Seamless Interoperability – Evidence on …

O4: develop a fully-fledged open tool platform that will allow developers and other assurance stakeholders to guarantee ***seamless interoperability*** of the platform with other tools used in the development of CPSs.



Seamless interoperability contributes to AMASS objective O4

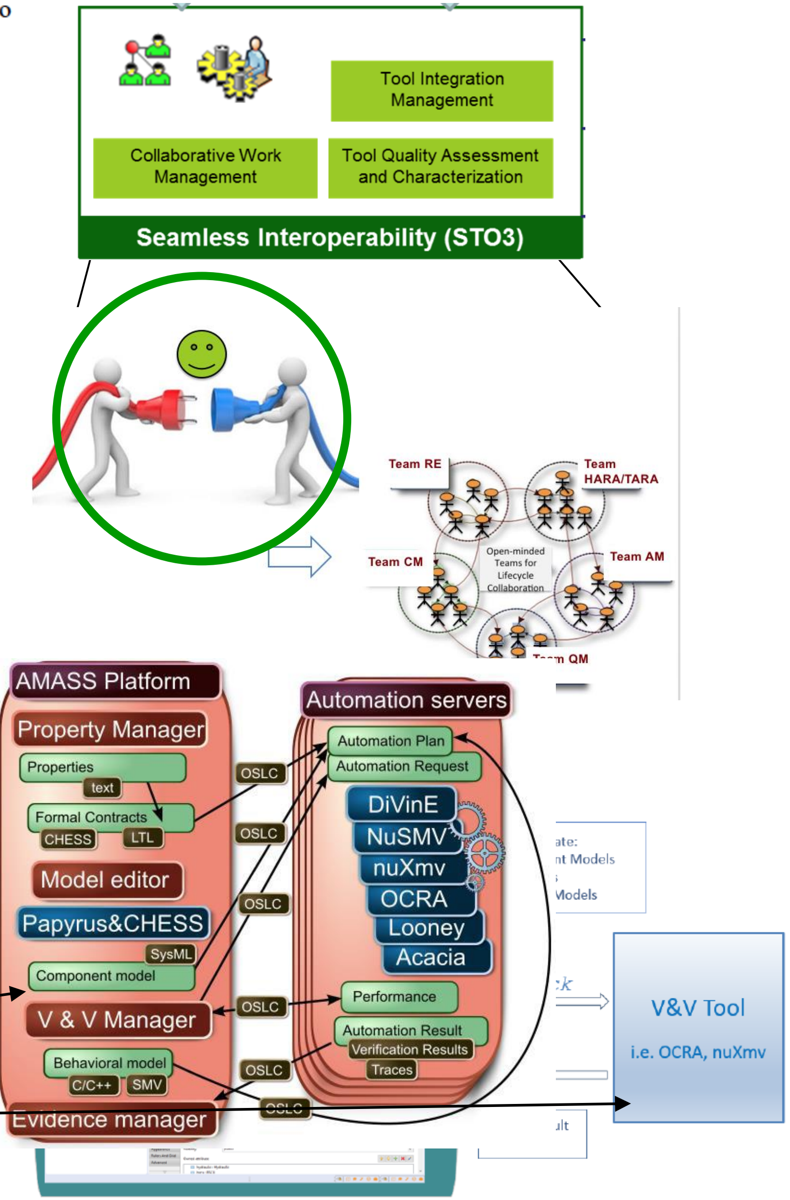
Context of system {X} and Standard {Y}

Management of tool integration is supported

Management of collaborative work is supported

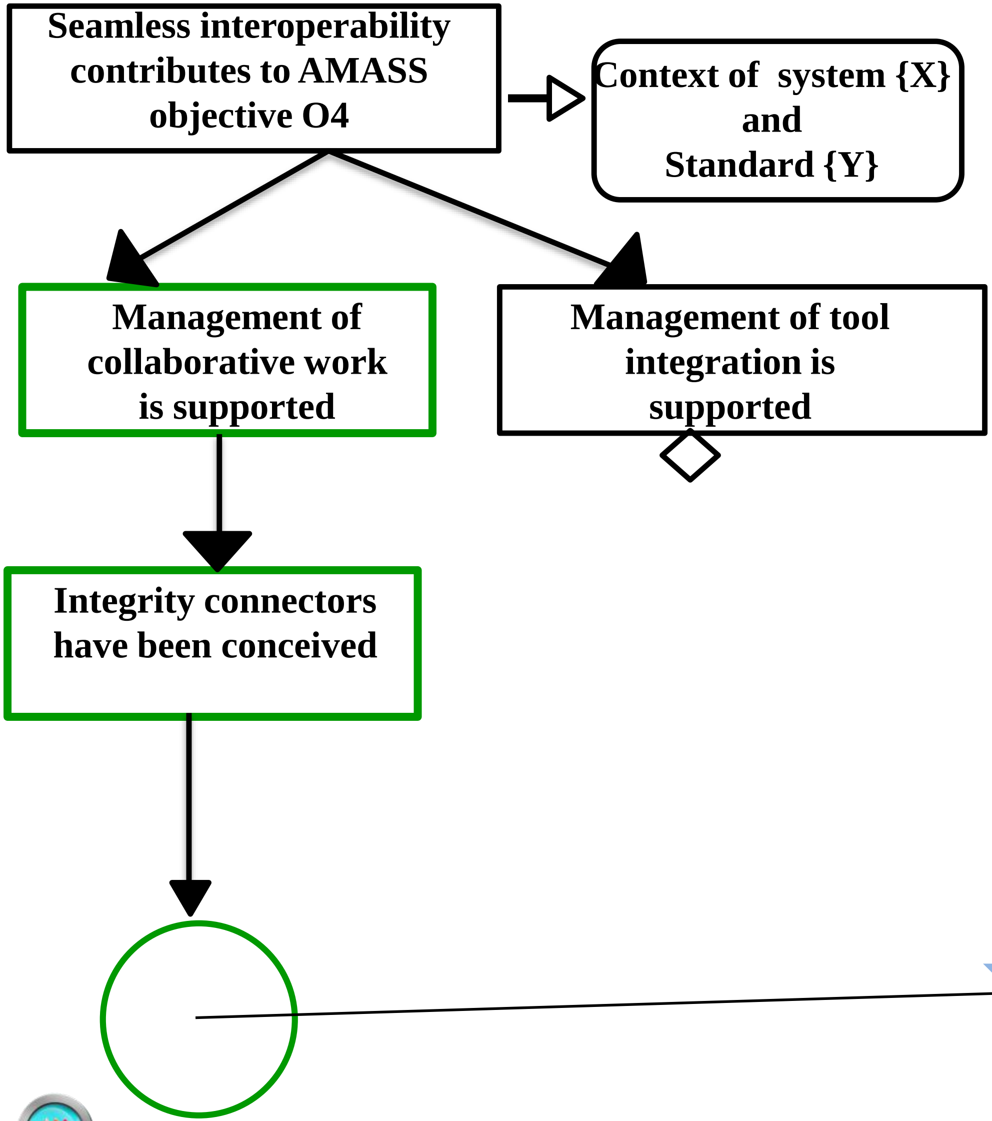
OSLC connector developed

Claim, claim decomposition, evidence

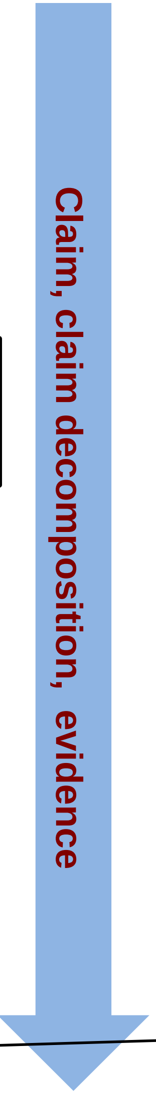


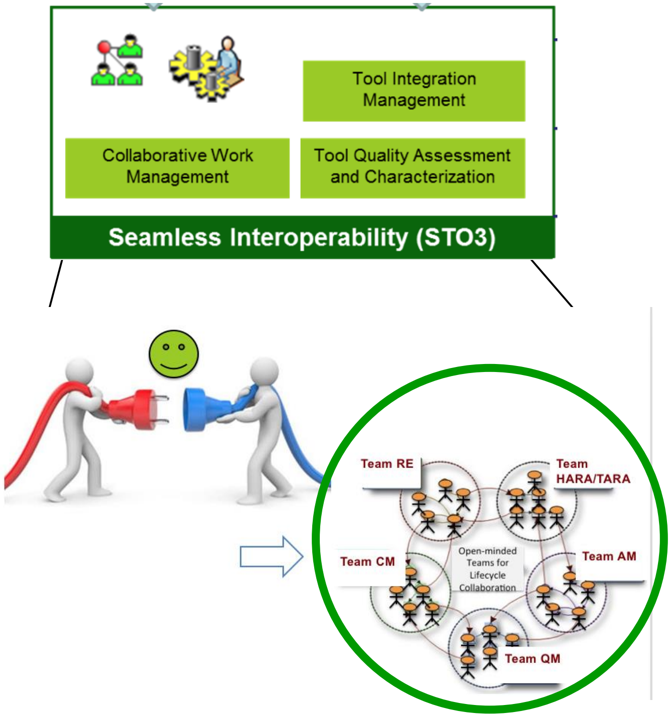
Seamless Interoperability – Evidence on …

O4: develop a fully-fledged open tool platform that will allow developers and other assurance stakeholders to guarantee ***seamless interoperability*** of the platform with other tools used in the development of CPSs.

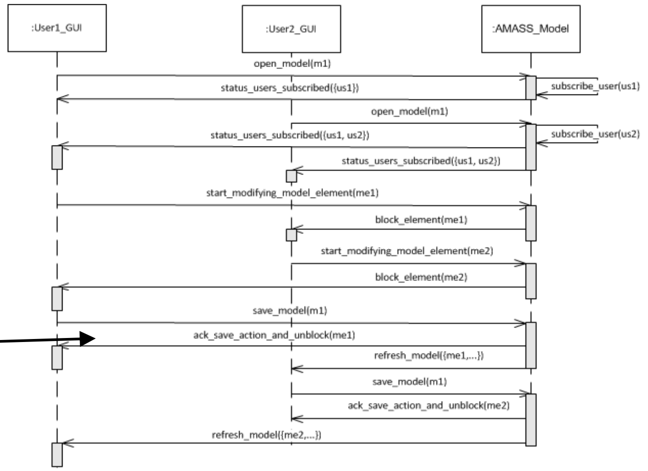


Seamless interoperability contributes to AMASS objective O4
Context of system {X} and Standard {Y}
Management of collaborative work is supported
Management of tool integration is supported
Integrity connectors have been conceived





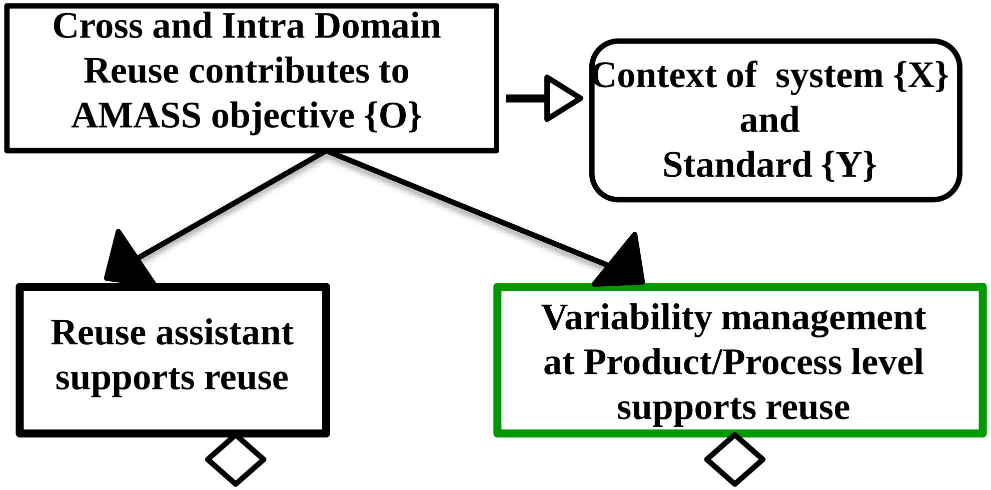
Tool Integration Management
Collaborative Work Management
Tool Quality Assessment and Characterization
Seamless Interoperability (STO3)
Team RE
Team HARA/TARA
Team CM
Open-minded Teams for Lifecycle Collaboration
Team AM
Team QM



:User1_GUI
:User2_GUI
:AMASS_Model
open_model(m1)
status_users_subscribed({us1})
subscribe_user(us1)
open_model(m1)
status_users_subscribed({us1, us2})
subscribe_user(us2)
status_users_subscribed({us1, us2})
start_modifying_model_element(me1)
block_element(me1)
start_modifying_model_element(me2)
block_element(me2)
save_model(m1)
ack_save_action_and_unblock(me1)
refresh_model({me1,...})
save_model(m1)
ack_save_action_and_unblock(me2)
refresh_model({me2,...})

Cross and Intra Domain Reuse

O3: consolidate a ***cross-domain and intra-domain assurance*** reuse approach to improve mutual recognition agreement of compliance approvals and to help assess the return of investment of reuse decisions.



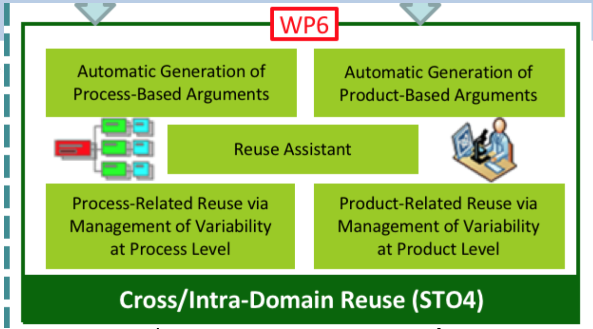
Cross and Intra Domain Reuse contributes to AMASS objective {O}

Context of system {X} and Standard {Y}

Reuse assistant supports reuse

Variability management at Product/Process level supports reuse

Claim, claim decomposition, evidence



WP6

Automatic Generation of Process-Based Arguments

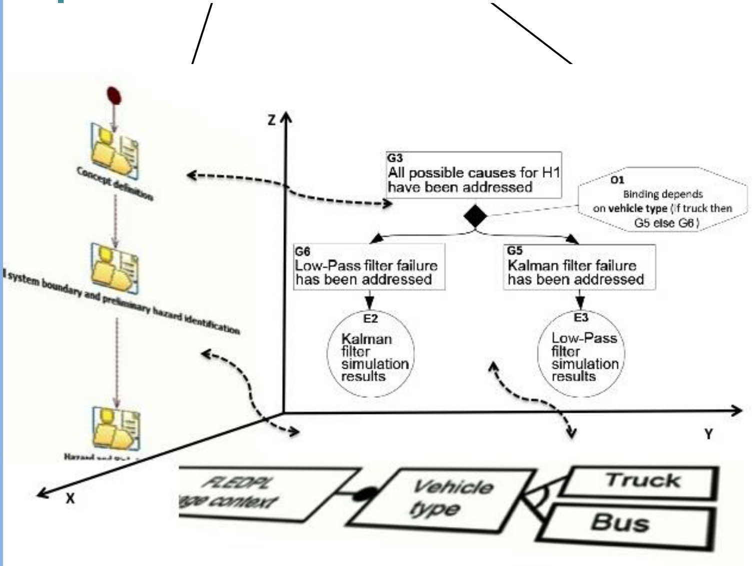
Automatic Generation of Product-Based Arguments

Reuse Assistant

Process-Related Reuse via Management of Variability at Process Level

Product-Related Reuse via Management of Variability at Product Level

Cross/Intra-Domain Reuse (STO4)


Concept definition
system boundary and preliminary hazard identification
Hazard and ...
Z
G3
All possible causes for H1 have been addressed
O1
Binding depends on vehicle type (if truck then G5 else G6)
G6
Low-Pass filter failure has been addressed
G5
Kalman filter failure has been addressed
E2
Kalman filter simulation results
E3
Low-Pass filter simulation results
Y
X
FLEDPL ...age context
Vehicle type
Truck
Bus

- OpenCert
- PolarSys CHESS **+ BVR Tool**
- eclipse PROCESS FRAMEWORK

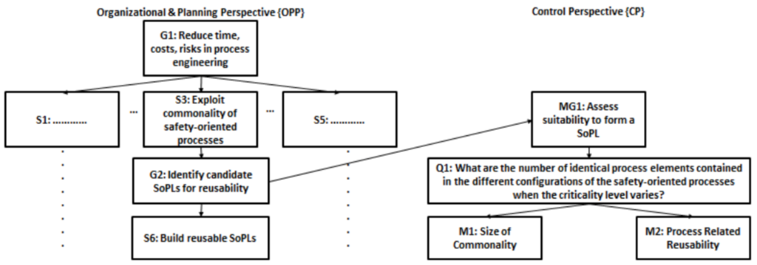
Cross and Intra Domain Reuse – Evidence on …

O3: consolidate a *cross-domain and intra-domain assurance* reuse approach to improve mutual recognition agreement of compliance approvals and to help assess the return of investment of reuse decisions.

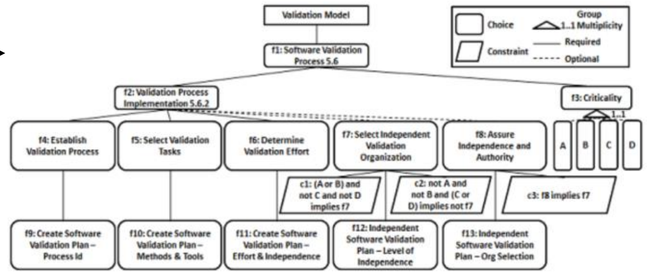
Variability management Process level supports reuse

Context of system ACS and Standard ECSS-40

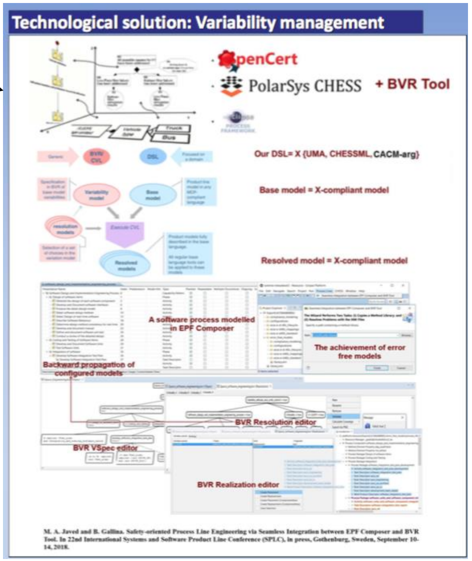
Claim, claim decomposition, evidence



Organizational & Planning Perspective {OPP}
Control Perspective {CP}
G1: Reduce time, costs, risks in process engineering
S1: …………
…
S3: Exploit commonality of safety-oriented processes
…
S5: …………
G2: Identify candidate SoPLs for reusability
S6: Build reusable SoPLs
MG1: Assess suitability to form a SoPL
Q1: What are the number of identical process elements contained in the different configurations of the safety-oriented processes when the criticality level varies?
M1: Size of Commonality
M2: Process Related Reusability



Validation Model
Choice
Constraint
Group
1..1 Multiplicity
Required
Optional
f1: Software Validation Process 5.6
f2: Validation Process Implementation 5.6.2
f3: Criticality
1..1
f4: Establish Validation Process
f5: Select Validation Tasks
f6: Determine Validation Effort
f7: Select Independent Validation Organization
f8: Assure Independence and Authority
A
B
C
D
c1: (A or B) and not C and not D implies f7
c2: not A and not B and (C or D) implies not f7
c3: f8 implies f7
f9: Create Software Validation Plan – Process Id
f10: Create Software Validation Plan – Methods & Tools
f11: Create Software Validation Plan – Effort & Independence
f12: Independent Software Validation Plan – Level of Independence
f13: Independent Software Validation Plan – Org Selection


Technological solution: Variability management

- OpenCert
- PolarSys CHESS + BVR Tool
-

Our DSL= X {UMA, CHESSML, CACM-arg}

Base model = X-compliant model

Resolved model = X-compliant model

A software process modelled in EPF Composer

The achievement of error free models

Backward propagation of configured models

BVR Resolution editor

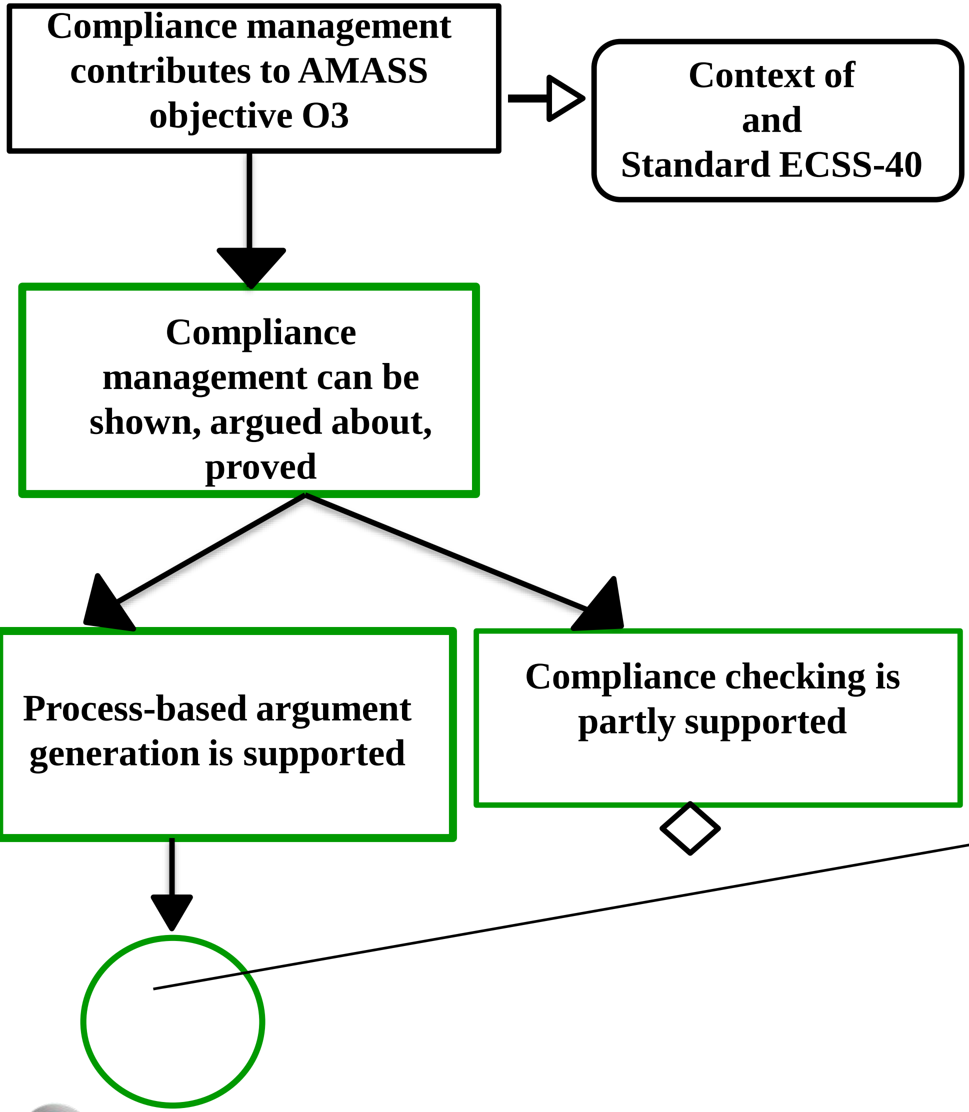
BVR VSpec editor

BVR Realization editor

M. A. Javed and B. Gallina. Safety-oriented Process Line Engineering via Seamless Integration between EPF Composer and BVR Tool. In 22nd International Systems and Software Product Line Conference (SPLC), in press, Gothenburg, Sweden, September 10-14, 2018.

Compliance Management – Evidence on …

O3: consolidate a *cross-domain and intra-domain assurance* reuse approach to improve mutual recognition agreement of compliance approvals and to help assess the return of investment of reuse decisions.



Compliance management contributes to AMASS objective O3

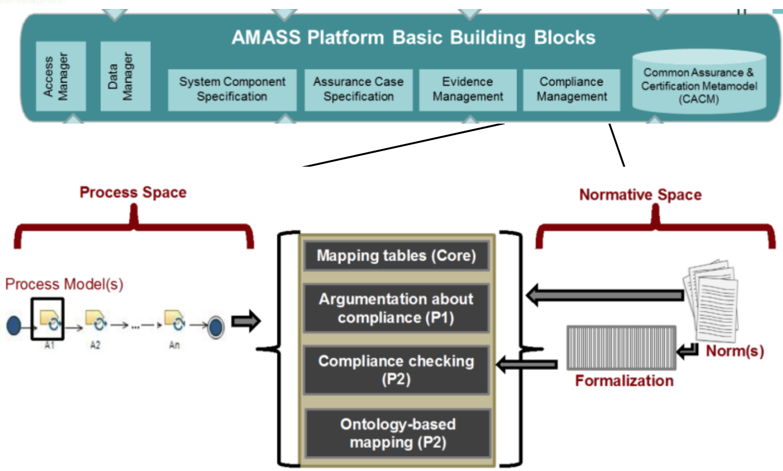
Context of and Standard ECSS-40

Compliance management can be shown, argued about, proved

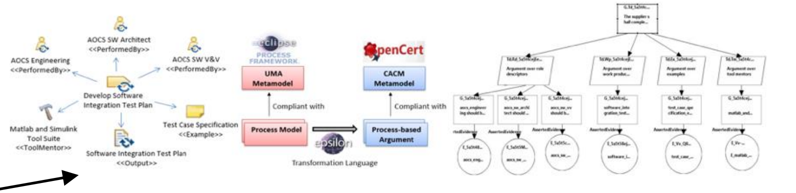
Process-based argument generation is supported

Compliance checking is partly supported

Claim, claim decomposition, evidence



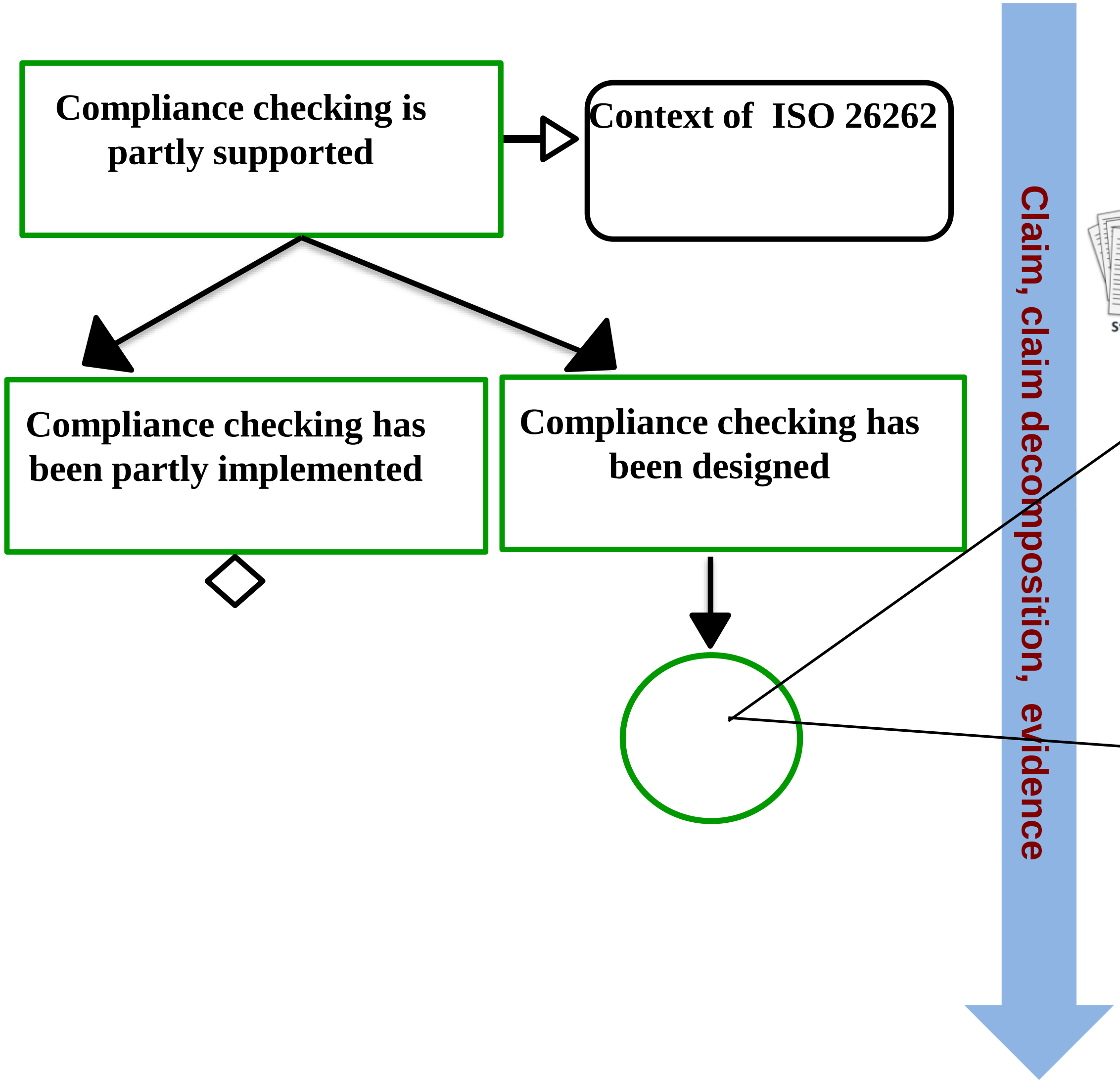
AMASS Platform Basic Building Blocks
Access Manager
Data Manager
System Component Specification
Assurance Case Specification
Evidence Management
Compliance Management
Common Assurance & Certification Metamodel (CACM)
Process Space
Normative Space
Process Model(s)
A1
A2
An
Mapping tables (Core)
Argumentation about compliance (P1)
Compliance checking (P2)
Ontology-based mapping (P2)
Norm(s)
Formalization



AOCS SW Architect <<PerformedBy>>
AOCS Engineering <<PerformedBy>>
AOCS SW V&V <<PerformedBy>>
Develop Software Integration Test Plan
Matlab and Simulink Tool Suite <<ToolMentor>>
Test Case Specification <<Example>>
Software Integration Test Plan <<Output>>
eclipse PROCESS FRAMEWORK
UMA Metamodel
Compliant with
Process Model
epsilon
Transformation Language
OpenCert
CACM Metamodel
Compliant with
Process-based Argument

Compliance Management – Evidence on …

O3: consolidate a *cross-domain and intra-domain assurance* reuse approach to improve mutual recognition agreement of compliance approvals and to help assess the return of investment of reuse decisions.



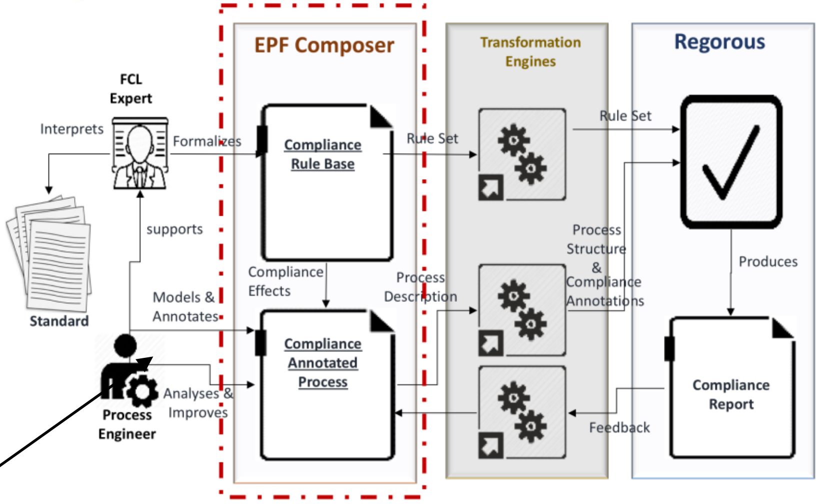
Compliance checking is partly supported

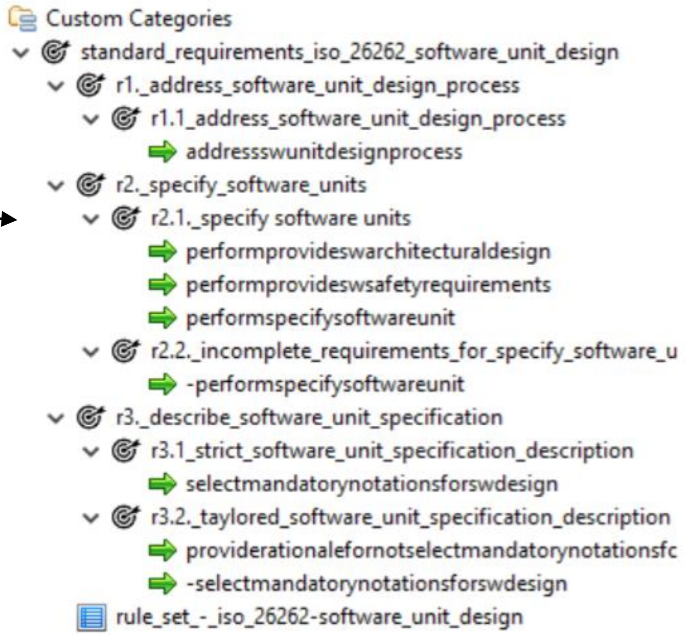
Context of ISO 26262

Compliance checking has been partly implemented

Compliance checking has been designed

Claim, claim decomposition, evidence





Custom Categories
- standard_requirements_iso_26262_software_unit_design
  - r1._address_software_unit_design_process
    - r1.1_address_software_unit_design_process
      - addressswunitdesignprocess
  - r2._specify_software_units
    - r2.1._specify software units
      - performprovideswarchitecturaldesign
      - performprovideswsafetyrequirements
      - performspecifysoftwareunit
    - r2.2._incomplete_requirements_for_specify_software_u
      - -performspecifysoftwareunit
  - r3._describe_software_unit_specification
    - r3.1_strict_software_unit_specification_description
      - selectmandatorynotationsforswdesign
    - r3.2._taylored_software_unit_specification_description
      - providerationalefornotselectmandatorynotationsfc
      - -selectmandatorynotationsforswdesign
- rule_set_-_iso_26262-software_unit_design

Thank you for your attention!

Any questions