SPDX FuSa Status

Elisa Workshop Lund 2025 Nicole Pappler, AlektoMetis



SPDX FuSa

Goal:

To create a SPDX profile, based on SPDX 3.0 that enabled the delivery of the documents created in a safety lifecycle to enable the automation of building, exchanging and processing safety evidences

Use Cases:

- Generation of the Safety Case documentation
- Safety SBOM as exchange format in the supply chain
- Integrating the build of the safety documentation into the pipeline





SPDX 3.0 model



Core profile



Use existing 3.0 model

SPDX

Meta amende descri modifi Struct Behavi config delega depend Pedig copied expand genera hasAdd hasDat hasDel Proven ancest availa descen variar Serial serial Build hasDis hasDoc hasDyn hasExa hasHos hasInp hasMet has0pt has0pt hasPre hasPro hasReg hasSta has7es has7es hasVar invoke patche usesTo Licens hasCon hasDec Securi affects doesNo exploi fixedB foundB hasAss publis report republ underI AI/Dat

hasEvi tested

ations		
RelationshipType	ExternalRefType	AnnotationType
	altDownloadLocation	other
iBy [Element -> Element]	altWebPage	review
(Element -> Element)	binaryArtifact	
dBy [Element -> Element]	bower	
[Element -> Element] (comment)	buildMeta	ExternalIdentifierType
	buildSystem	
ire	certificationReport	cpe22
is [Element -> Element]	chat	cpe23
	componentAnalysisReport	cve
oral	documentation	email
ires [Element -> Element]	dynamicAnalysisReport	getoid
edTo [Element -> Element]	eolNotice	other
On [Element -> Element]	exportControlAssessment	packageUrl
	funding	securityOther
e	issueTracker	swhid
To [Element -> Element]	license	swid
To [Artifact -> Artifact]	mailingList	urlScheme
es [Artifact -> Artifact]	mavenCentral	
dfile [Element -> Element]	metrics	
file [Element -> Element]	npm	RelationshipCompleteness
tedfile [Element -> Element]	nuget	
	other	complete [default]
ince	privacyAssessment	incomplete
orOf [Element -> Element]	productMetadata	Incomplete
eleFrom [Element -> Element]	purchaseOrder	nonssercion
iantOf [Element -> Element]	qualityAssessmentReport	
[Artifact -> Artifact]	releaseHistory	
	releaseNotes	LifecycleScopeType
zation	riskAssessment	
zedInArtifact [SpdxDocument -> Artifact]	runtimeAnalysisReport	build
	secureSoftwareAttestation	design
	securityAdvisory	development
andencyManifest [Element -> Element]	securityAdversaryModel	other
cributionArtifact [Element -> Element]	securityFix	runtime
umentation [Element -> Element]	securityOther	test
amicLink [Element -> Element]	securityPenTestReport	
sple [Element -> Element]	securityPolicy	
(Build -> Element)	securityThreatModel	ProfileIdentifierType
t (Build -> Element)	socialMedia	
adata [Element -> Element]	sourceArtifact	ai
ionalComponent [Element -> Element]	staticAnalysisReport	build
ionalDependency [Element -> Element]	support	core
out (Build -> Element)	VC8	dataset
requisite [Element -> Element]	vulnerabilityDisclosureReport	expandedLicensing
videdDependency [Element -> Element]	vulnerabilityExploitabilityAssessment	extension
irement [Element -> Element]		lite
Element -> Element]		security
icLink [Element -> Element]		simpleLicensing
[Element -> Element]		software
Case [Element -> Element]	HashAlgorithm	
[Element -> Element]		
iBy [Element -> Agent]	adler32	BroconsoTur
dBy [Element -> Element]	blake2b256	PresenceType
Element -> Element)	blake2b384	20
(Element -> Element)	blake2b512	notecortion
	blake3	1000
ing	crystalsDilithium	ye=
cludedLicense [SoftwareArtifact -> AnyLicenseInfo]	crystalsKyber	
laredLicense [SoftwareArtifact -> AnyLicenseinfo]	falcon	
	md2	SupportType
-y	md4	
[Vulnerability -> Element]	md5	deployed
Affect (Vulnerability -> Element)	md6	development
CreatedBy (Vulnerability -> Agent)	other	endOfSupport
(Vulnerability -> Agent)	shal	limitedSupport
(Vulnerability -> Agent)	sha224	noAssertion
issmentror [Vuinerability -> Element]	sha256 [default]	noSupport
(Welevenherability [Artifact -> Vuinerability]	sha384	support
[Vuinerability -> Agent]	sna512	
aby [Vuinerability -> Agent]	sna3_224	
[Vulnerability -> Agent]	sna3_256	
vestigationFor [Vulnerability -> Element]	sha3_384	
	sha3_512	
iset		
[Element -> Element]		
(Element -> Element)		
ion [Element -> Element]		





Similar SPDX profiles





Security Profile



... and many more

Dependencies in a FuSa Project









FuSa documentation structure

All FuSa related documentation is part of the Safety Case! Think of all these documents as part of the release - each document is part of the Bill of Material, as is each screw, each microcontroller and each piece of software!







Data Structure of current FuSa projects...





Data Structure of current FuSa projects





Any guesses????



Emoji by emojidex







Emoji by emojidex



No 1 Safety Information Exchange Format

ALL MODERN DIGITAL INFRASTRUCTURE

draft_2005TemplateSafetyCase_thisproject_final_forTraceingv06.xls



Emoji by emojidex





	nationship type		ExternalHerType	Annotation type
ta			altDownloadLocation	other
endedBy	(Element -	> Element]	altWebPage	review
SCEIDES	[Element -	-> Element]	DinaryArtifact	
her	Element -	> Element] (comment)	buildMeta	ExternalIdentifierT
			buildSystem	LAternanoentiner
ructure			certificationReport	cpe22
ntains	[Element -	> Element]	chat	cpe23
			componentAnalysisReport	cve
havioral	101.0000		documentation	enail
logatedTo	[Element -	> Element)	dynamicAnalysiskeport	getoid
pendsOn	Element -	> Element)	exportControlAssessment	nackagellr]
			funding	securityOther
digree			issueTracker	swhid
piedTo	[Element -	<pre>> Element]</pre>	license	swid
pandaTo	[Artifact -	> Artifact]	mailingList	urlSchene
nerates	[Artifact -	> Artifact)	mavenCentral	
sAddedille	(E)ement -	> Element)	netrics	
aDeletedfile	Element -	> Element]	nuget	RelationshipComplet
			other	
ovenance			privacyAssessment	complete [defaul
cestorOf	[Element -	<pre>> Element]</pre>	productMetadata	ncâmertion
ailableFrom	[Element -	> Element]	purchaseOrder	nonssertron
scendantOf	(Element -	> Element]	qualityAssessmentReport	
EXOLC.	(Artiract -	- ALCIERCE)	releaseHistory	LifecycleSconeTh
rialization			risklassespert	Lifet y clest uper y
rializedInArtifact	[SpdxDocument -	> Artifact]	runtimeAnalysisReport	build
			secureSoftwareAttestation	design
ild			securityAdvisory	development
sDependencyManifest	[Element -	<pre>> Element]</pre>	securityAdversaryModel	other
sDistributionArtifact	[Element -	> Element)	securityFix	runtime
abocumentation	(Element -	> Elementj	securityOther	test
sEvannle	[Element -	> Element]	securityPeniestKeport	
aHost	(Build -	> Element)	securityThreatModel	ProfileIdentifierTy
sInput	[Build -	> Element]	socialMedia	
isMetadata	[Element -	> Element]	sourceArtifact	ai
sOptionalComponent	[Element -	<pre>> Element]</pre>	staticAnalysisReport	build
sOptionalDependency	[Element -	> Element)	support	core
Boutput	(D)	> Element)	vcs	dataset
ProvidedDependency	(E)ement	> Element)	VulnerabilityDisclosurekeport	expandedisteensin
aRequirement	Element -	> Element]	valuerabilicylaxproreabilicynaaesanane	lite
sSpecification	[Element -	> Element]		security
sStaticLink	[Element -	> Element]		simpleLicensing
aTest	[Element -	> Element]		software
IsTestCase	[Element -	-> Element]	HashAlgorithm	
svariant	[Element -	> Elementj	- 43	
ickagedBy	[Element -	> Element]	blake2b256	PresenceType
tchedBy	[Element -	> Element)	blake2b384	
esTool	[Element -	> Element]	blake2b512	no
			blake3	noAssertion
censing			crystalsDilithiun	Aco
sConcludedLicense [Sc	ftwareArtifact -	> AnyLicenseInto]	crystalsKyber	
specialednicense [50	itwareniciiace -	AnyLicenseinic)	Talcon	0
curity			nd4	oupporttype
fects	[Vulnerability -	> Element)	md5	deployed
NesNotAffect	[Vulnerability -	<pre>> Element]</pre>	md6	development
ploitCreatedBy	[Vulnerability -	> Agent]	other	endOfSupport
xedBy	[Vulnerability -	> Agent]	shal	limitedSupport
-unuby sAssessmentFor	[vulnerability -	> Nyentj	sha224	noAssertion
sassociatedVulnerahil	ity [artifact -	> Vulnerabilityl	sha256 [default]	noSupport
blishedBy	[Vulnerability -	> Agent]	sha512	aupport
portedBy	[Vulnerability -	> Agent]	sha3 224	
publishedBy	[Vulnerability -	-> Agent]	sha3_256	
derInvestigationFor	[Vulnerability -	> Element)	sha3_384	
			sha3_512	
/Dataset	(2)	> Discorti		
- Desidence -	Issement -	> prement]		
sEvidence	E ament	C Element]		
sEvidence stedOn ainedOn	[Element -	-> Element] -> Element]		
sEvidence stedOn ainedOn	[Element -	> Element) > Element]		
sEvidence stedOn ainedOn	[Element - [Element -	<pre>>> Element) >> Element]</pre>		

Generate SBOMS when the data is known - by the projects





Exchange SPDX Safety SBOMs









Safety Information Exchange Format?





... instead of inconsistent Spreadsheets, manual import/export of half decent ReqIFs...

Dependencies in a FuSa Project







FuSa documentation structure



All FuSa related documentation is part of the Safety Case!

Think of all these documents as part of the release - each document is part of the Bill of Material, as is each screw, each microcontroller and each piece of software!





Classes for WPs - REQUIREMENT



Requirements Specifications Determining factors and assumptions:

- A <u>requirement</u> describes a functional, non-functional or design need placed on an item (HW, SW, system, whatever can be the product)
- There are different sources of requirements
- Atomic REQUIREMENTS entities can be packaged to Requirement sets that then can become part of specifications ⇒ no new class needed, use existing SPDX functionality



Classes for WPs - REQUIREMENT



Requirements Specifications

REQUIREMENT class



- Element

 + spdxID: xsd: anyURI[1]

 + name: xsd: string[0.. 1]

 + summary: xsd:string[0.. 1]

 + description: xsd:string[0.. 1]

 + comment: xsd:string[0.. 1]

 + creationInfo: CreationInfo [1]

 + verifiedUsing: integrityMethod [0.. *]

 + externalRef: ExternalRef[0.. *]

 + externalIdentifier: ExeternalIdentifier[0.. *]
 - + extension: /Extension/Extension[0... 1]



REQUIREMENT - relationships





SPDX



implemented has Requirement

Item 1 Item 2 Item 3

SoftwareArtifact

Evidence

(e.g. a test report) Item 2

Item 3

hasEvidence

REQUIREMENT & product line engineering SAFETY WIP - Open for suggestions!

Requirements Specifications



Example:

Requirement 1: The VEHICLE_TYPE vehicle with ENGINE_TYPE shall have a TRANSMISSION_SYSTEM between engine and wheels

Requirement 2:

Features-applicability FA1: VEHICLE_TYPE: small; medium, large Features-applicability FA2:

TRANSMISSION_SYSTEM: manual, automatic, fixed

Feature-applicability FA3:

BEV, PHEV, FUEL

Product Configuration 1:

SuperNewVehicle: FA1:small, FA2:fixed, FA3:BEV

Product Line Element

Instance of Features: Table with actual values for Features in Requirements

Configuration Element

Comes from the software artifact, defines the actual configuration of feature elements

SPDX

Classes for work products - TASK and Process SAFETY

Plans Processes Guidelines

Determining factors and assumptions:

- we can generalize what it is a plan, process or guideline as a set of things, that can be done/performed ⇒ TASKs
- A PROCESS is a list of TASKs,
- While a PLAN will look very much like a PROCESS, a PLAN is a project specific instantiation of a PROCESS
- There are a lot of PROCESS types, e.g. for software it can be things like, dev-process, test-process, build-process, assessment-process etc
- In the definition of IEC 61508, there are requirements for systematic capability, which in the engineering reality translate to process and methods, and therefore these "requirements for systematic capability" -

TASK class includes everything process, plan or guideline, as these types of documents always look the same



Other classes - TASK



Determining factors and assumptions:

- A TASK is a specific unit of work that contributes to the completion of a project or an item
- TASKs can be of different types
- Need at least the content of ELEMENT
- An TASK as a minimum needs:
 - An Objective what the tasks aims to achieve
 - Preconditions and neccessary inputs, resources
 - An Agent (human, tool) with assigned ROLE
 - Completion Conditions, Definition of Done
 - Optionally an environment and its configuration needed to perform the task
 - Optionally supporting information



TASK

Classes for work products - TASK







TASK class







Classes for work products - AGENT



AGENT class AGENT Agent Organization Person Tool Artifact **SPDX**



Classes for work products - AGENT



AGENT class

AGENT

- AGENT can be an actual person, a tool, script, infrastructure, organization...
- For FuSa (and other dependability topics) the agents performing tasks must have sufficient expertise/qualification

Agent
+ AgentType: agentType [1 *]
+ AgentQualification: QualificationType [1*]
+ AgentQualificationVerified: QualificationVerificationType [01]

QualificationType
+ qualifiedFor: RoleType [1]
+ qualificationLevel: QualificationLevelType [1]
+ qualificationCompleteness: qualificationCompletenessType [1]
+ qualificationEvidence: ExternalRef [1]

Classes for work products - enums



Role (enum) RequirementsEngineer VerificationEngineer SoftwareDeveloper SoftwareGenerator SoftwareDeveloper Tester SafetyAssessor TaskCoordinator ProcessPlanner SafetyAnalysisModerator SafetyEngineer SafetyManager RiskAnalyst DeploymentManager DeploymentTester

Plans Processes Guidelines

Enums to describe roles and qualifications
--

QualificationLevelType (enum)
Beginner
Advanced
Expert
Senior Expert
Genius
QualifiedSoftware
other

QualificationCompletenessType (enum)	
Complete	
Incomplete	
other	
QualificationVerificationType (enum)	
Verified	
Unverified	
other	

SPDX



Definition:

Task: a task is a specific unit of work that contributes to the completion of a project. E.g. a plan is created, code is changed, a test is specified, a test log is recorded

Action: something that is physically done to a physical object. E.g. a PCB is manufactured, an ECU is mounted into a machine,



Task vs HW Action





SPDX

Task vs HW Action



Example:

For a vehicle ECU, the hardware (PCB, housing, assembling of all of it) is manufactured, which is recorded by an ACTION.

The software that has been created to run on this ECU, has been created following work steps that are defined in TASKs. These TASKs define e.g. how a software requirement must be created, how it must be written, which tooling is needed. The code for the software is created using TASKs that define how code is written (coding guidelines) and how it is managed in the config management system (e.g. GitHub workflow)

The software is then tested, using test procedures also defined by TASKs. The way the test results are captured and evaluated is also described in its TASKs.

Building the software and running the tests, as well as the creation of the Evidence artefacts for it, is all described by TASKs, there is no specific need for a Location, Duration or similar. The buildTime is already defined by ARTEFACT.

Once we talk about something physically being manufactured, location and duration will become interesting also for safety.

Corner Cases: Is flashing a binary to an ECU a TASK or an ACTION? What about downloading a binary to model to run on an FPGA?

SPDX

Classes for WPs - VERIFICATION



TBD - Open for suggestions!

Verification Analysis Test Evidences

Determining factors and assumptions:

- There are different types of verifications, eg.
 - Test
 - Review/Inspection
 - Analysis
 - Demonstration
- Verification means we have a PROCESS how to do VERIFICATION and some evidence that this verification was performed and what were the environmental and runtime conditions of these tests
- While the verification PROCESS is a process that can be defined using the PROCESS class, a test case/suite/checklist looks very much like a REQUIREMENT, but not exactly

 \Rightarrow need class for VERIFICATION specification to have something that describes test cases



Classes for WPs - VERIFICATION



TBD - Open for suggestions!





Classes for WPs - VERIFICATION



TBD - Open for suggestions!



VERIFICATION class



Classes for WPs - EVIDENCE



TBD - Open for suggestions!

Evidence (test reports, build logs, etc.) Determining factors and assumptions:

- EVIDENCEs are created based on a PROCESS with a ProcessType for verification
- EVIDENCEs are created applying ACTIONs by an Agent
- EVIDENCES attest a certain level of compliance of
 - a tested item (code) with its acceptance criteria (requirement), using the test process and



Classes for WPs - EVIDENCE



TBD - Open for suggestions!





Classes for WPs - EVIDENCE



TBD - Open for suggestions!

Evidence (test reports, build logs, etc.)

EVIDENCE class - enums we need

- **EvidenceType**: report, log, video, ...
- EvidenceResult: passed, partially passed, partially run, skipped, failed, ...

... to be continued

Talk to us: <u>nicole@alektometis.com</u> <u>kstewart@linuxfoundation.org</u> <u>Mailing List</u> <u>Weekly meeting Friday 18:00 CET/CEST</u>



Appendix



Zephyr Requirements Management

Requirements Management Knowledge Model

Safety Committee View *







Dependencies of Safety Plan, Safety Claim, Req, Design and Code



Design SBOM to Source SBOM



Source SBOM to Build SBOM



Dependency Identification on Component Level ? Specification_For ? Specification_For ?











Specification



