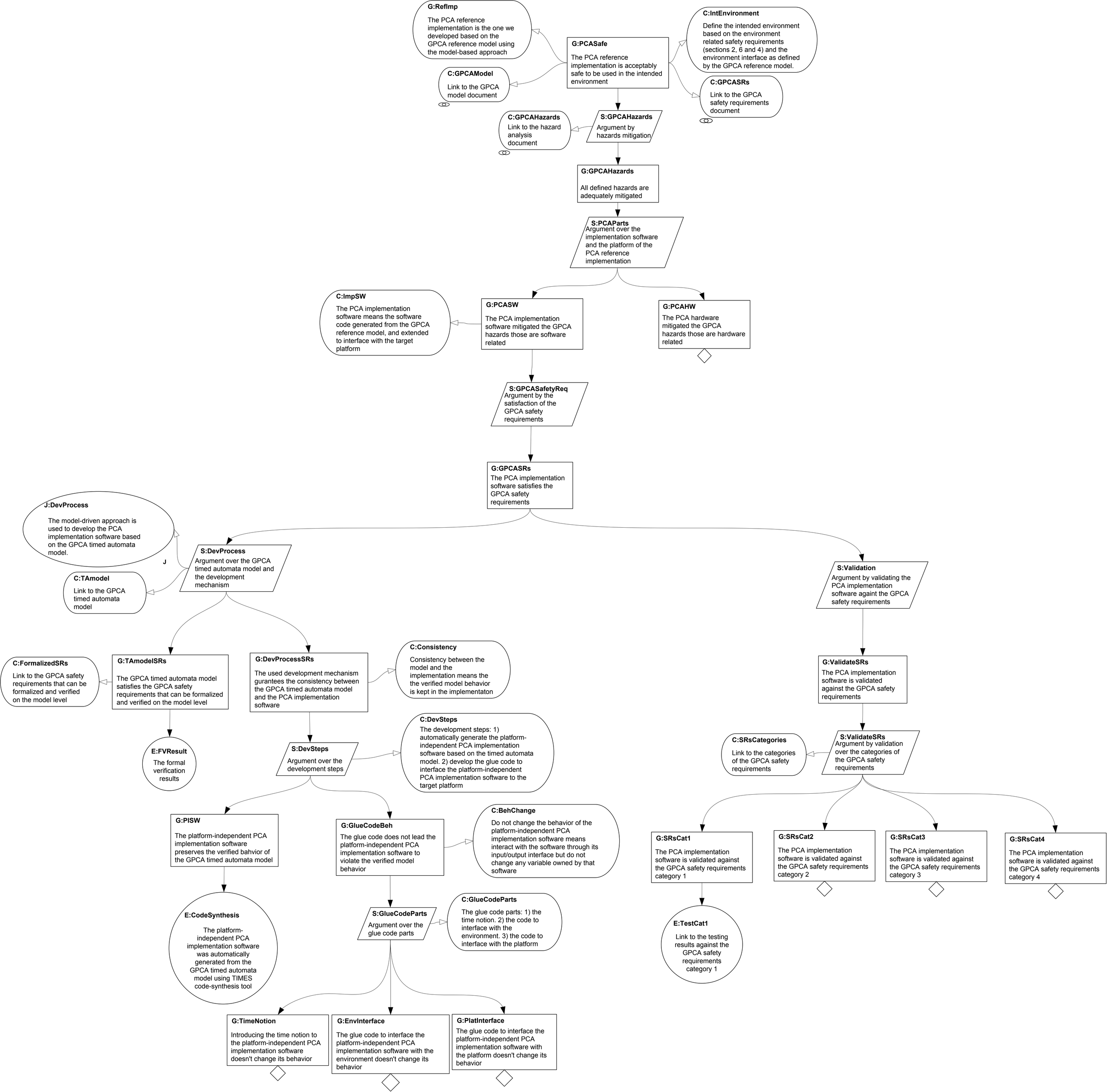
G:RefImp
The PCA reference implementation is the one we developed based on the GPCA reference model using the model-based approach
G:PCASafe
The PCA reference implementation is acceptably safe to be used in the intended environment
C:IntEnvironment
Define the intended environment based on the environment related safety requirements (sections 2, 6 and 4) and the environment interface as defined by the GPCA reference model.
C:GPCAModel
Link to the GPCA model document
C:GPCASRs
Link to the GPCA safety requirements document
C:GPCAHazards
Link to the hazard analysis document
S:GPCAHazards
Argument by hazards mitigation
G:GPCAHazards
All defined hazards are adequately mitigated
S:PCAParts
Argument over the implementation software and the platform of the PCA reference implementation
C:ImpSW
The PCA implementation software means the software code generated from the GPCA reference model, and extended to interface with the target platform
G:PCASW
The PCA implementation software mitigated the GPCA hazards those are software related
G:PCAHW
The PCA hardware mitigated the GPCA hazards those are hardware related
S:GPCASafetyReq
Argument by the satisfaction of the GPCA safety requirements
G:GPCASRs
The PCA implementation software satisfies the GPCA safety requirements
J:DevProcess
The model-driven approach is used to develop the PCA implementation software based on the GPCA timed automata model.
J
S:DevProcess
Argument over the GPCA timed automata model and the development mechanism
C:TAmodel
Link to the GPCA timed automata model
S:Validation
Argument by validating the PCA implementation software againt the GPCA safety requirements
C:FormalizedSRs
Link to the GPCA safety requirements that can be formalized and verified on the model level
G:TAmodelSRs
The GPCA timed automata model satisfies the GPCA safety requirements that can be formalized and verified on the model level
G:DevProcessSRs
The used development mechanism gurantees the consistency between the GPCA timed automata model and the PCA implementation software
C:Consistency
Consistency between the model and the implementation means the the verified model behavior is kept in the implementaton
G:ValidateSRs
The PCA implementation software is validated against the GPCA safety requirements
E:FVResult
The formal verification results
S:DevSteps
Argument over the development steps
C:DevSteps
The development steps: 1) automatically generate the platform-independent PCA implementation software based on the timed automata model. 2) develop the glue code to interface the platform-independent PCA implementation software to the target platform
C:SRsCategories
Link to the categories of the GPCA safety requirements
S:ValidateSRs
Argument by validation over the categories of the GPCA safety requirements
G:PIsW
The platform-independent PCA implementation software preserves the verified bahvior of the GPCA timed automata model
G:GlueCodeBeh
The glue code does not lead the platform-independent PCA implementation software to violate the verified model behavior
C:BehChange
Do not change the behavior of the platform-independent PCA implementation software means interact with the software through its input/output interface but do not change any variable owned by that software
G:SRsCat1
The PCA implementation software is validated against the GPCA safety requirements category 1
G:SRsCat2
The PCA implementation software is validated against the GPCA safety requirements category 2
G:SRsCat3
The PCA implementation software is validated against the GPCA safety requirements category 3
G:SRsCat4
The PCA implementation software is validated against the GPCA safety requirements category 4
E:CodeSynthesis
The platform-independent PCA implementation software was automatically generated from the GPCA timed automata model using TIMES code-synthesis tool
S:GlueCodeParts
Argument over the glue code parts
C:GlueCodeParts
The glue code parts: 1) the time notion. 2) the code to interface with the environment. 3) the code to interface with the platform
E:TestCat1
Link to the testing results against the GPCA safety requirements category 1
G:TimeNotion
Introducing the time notion to the platform-independent PCA implementation software doesn't change its behavior
G:EnvInterface
The glue code to interface the platform-independent PCA implementation software with the environment doesn't change its behavior
G:PlatInterface
The glue code to interface the platform-independent PCA implementation software with the platform doesn't change its behavior