Open Source Project: Framework Modeling

Comments:

1. We will not at classes to the Standard Use Case Model

Step 0: Framework Elements

* Introduce Domains, Aspects and Concerns (fixed finite sets as classes)
  + Domains (give a partition of function) Modeled in a NEW DIAGRAM (separate from the CPS Framework diagram.
    - Created System Class Object in the CPS Framework model and created many to many association from this class to itself
    - Created Device Class Object in the CPS Framework model and inserted ‘part’ arrow from device to system
    - Created Domain Class Object (XSDcomplexType label?) and created bidirectional many to many association between Domain and Device classes
      * Domain class has attributes: name, technicalID and description
  + Aspects (convenient categorization or types of concerns)
    - Modeled but they need to switch roles with concerns in the current model. This item was addressed at the 25Mar2016 meeting.
    - Aspect Class with Concerns Classes that inherit from the Aspect Class
    - Aspects class now consists of all of the nine individual aspects
  + Concerns (voices of stakeholders from consumer/user to engineering level)
    - Modeled but see previous bullet
    - Properties class was added 25Mar2016 (a property is a closed statement over a finite set of propositional variables)
    - There is an association from a concern class to 0 or more Properties
* Introduce Facets (finite sets of activities associated with these modes of thinking about CPS)
  + Facet Is modeled as a class
  + The 3 facets are each modeled as classes with inheritance from the Facet Class
    - Conceptualization Facet class
    - Realization Facet class
    - Assurance Facet class
    - Each will have inheritance from the Facet class
  + Activities (finite sets of Framework defined activity associated with each mode of thinking about CPS – their steps and artifacts may vary by domain)
    - Model each activity in each facet as a class with inheritance from that facet?
  + Artifacts (these are the products of the activities and their form may also vary by domain); class Artifacts has a set of attributes one of which is Type that can take values in an ‘enumeration class’, e.g., artifact type would have the enumeration: Design, Test, Test Results, Simulation, etc.
    - Create a class Artifact Class
    - Should be an association between Property and Artifact? Should this be one or more
    - Create additional classes for Design Artifact and Test Artifact Other Artifacts
    - For each activity there will be a 1 to ‘1 or more’ association with the Artifact Class
    - the Design Artifacts and the Test Artifacts are ‘counted’ as values of Artifact attributes
    - If there is an issue here it is specifying the artifacts – they are not known in advance! Initially allow there to be a 0…\* association between concerns and properties
  + link concerns to activities (to indicate application of the concern)
    - Each activity class should ‘be subject to all concerns’

Step I: Conceptualization Facet (Artifacts are properties and design properties)

* perfect the property modeling or concern class model – done: there is a 1 to 1 or more association between concerns and properties (what if the concern is not relevant to the CPS – perhaps there should be a ‘null property’ that is true of any CPS but does not put any constraint on the CPS
* properly reflect it in the use case UML model (use existing structure or add new class) – check to see that the Framework UML Model correctly pushes all the concerns correctly out over the Use Case UML Model with the current linkage!
* complete the concerns class model – with the current modification, this seems to be adequate
* which concern levels to be included – should we cascade the concern decomposition down to the level of Cyber Security or even CIA below that? Should this be done for multiple ‘level 2’ concerns?
* new ones proposed – see above
* support the functional decomposition of actors (subsystems) into functions all the way down to subsystem components, with SW and HW and interfaces (this should include SLA/SenseLogicActuation, i.e., the allocation of function to components) – actors are already modeled in the Use Case UML model as connected to the Framework UML model
* determine how to put the meat of actual CPS into the model
* introduce a class for ‘design properties’
  + Create a class Design Properties that inherits from Properties class
  + Should this be identified with Design Artifact class (only need one)?
* answer the question about whether we need ‘trace’ attribute
* get all notions of tracing modeled
* tracing of property to concern – Is this achieved by the association between Aspect and Property
* tracing of design requirements to CPS properties (optional) – see above the possible classes Design Artifact, Design Property and Test

Step II: Realization Facet (Artifacts are design elements, test and test results)

* subsystems and their components, defined during conceptualization – these are the actors and are currently captured in the Use Case UML Model
* set of design elements, test is associated with each (subsystem and component system level test and component unit test, together with the test results) – modeled above as Artifact class with Design and Test Artifact subclasses (inheritance)
* Need to include the 1 to 1 or more associations between Design Property and Design Element – this is an issue; perhaps a Property becomes a Design Property when it is associated with a system/subsystem/actor; in that case Property is associated with Concern for a CPS – does this make sense?
* ‘full system level test/exercising a use case’ - sometimes called ‘acceptance criteria’ – these are generally derived from the use cases defined in the conceptualization facet and are sometimes referred to as ‘functional objectives/property of a use case’
* Logical Tracing: a relationship between the test of artifacts aimed at realizing properties of actors and their ‘logical decomposition’ (remain properties of same artifacts and are decomposed to facilitate the understanding of what is evidence for them – is this necessary?)

Step III: Assurance Facet (Artifacts are evidence, judgements and assurance cases)

* We want to assure the properties of use cases!
* represent assurance relation between artifacts produced in the realization facet and the conceptualization properties, including
  + using tracing of design property decomposition of each conceptualization property get associated judgements that the assurance of each conceptualization property is achieved by using the evidence associated with the judgements that evidence for the assurance of the traceability children of that conceptualization property