AoURN to RAM Transformation

Features

- Add a new feature: Fea Create
  (Feature name format is FeaNameOfFeature)

FeatureModeling : ---+ Feature Modeling

jUCMNav supports combined feature and goal modeling as well as analysis by defining feature models as a special case of goal models. Feature and goal models are analyzed together at the same time using the *Feature Model Strategy Algorithm*. First, select the "Feature Model Strategy Algorithm" in the jUCMNav preferences (top menu: Window - Preferences - jUCMNav - GRL Strategy Evaluation Algorithm - GRL Evaluation Algorithm Select = Feature Model Strategy Algorithm; GRL Evaluation Algorithm Tolerance = 0; check Automatically select mandatory features). Right click the background of the editor and select "Add Feature Model diagram" to create a feature model diagram. Then, add models elements to the feature model with the help of the new palette:

The contribution values of mandatory or optional links are not set by the modeler. The *Feature Model Strategy Algorithm* will set these values automatically to preserve the semantics of mandatory and optional links.

Features can be used directly in goal models to capture the impact of features on high-level goals and system qualities:

Scenario models describe features in more detail (e.g., the Password feature is added to this scenario model):

The variable "_GRL_Password" enables feature configurations to be synchronized with scenario models. In this case, if the feature Password is selected (i.e., its evaluation value equals 100), then the branch with the Password stub is chosen when traversing the scenario model. Feature configurations are defined with the help of GRL strategies. Switch to the jUCMNav Execution perspective to define and evaluate strategies. When a strategy is evaluated, the strategy algorithm verifies whether the configuration is valid:

Several strategies can be compared against each other in terms of whether the feature configuration is valid and in terms of the impact of the configuration on goal model elements:

With the help of the variables mentioned earlier, the feature configuration also influences the traversal of the scenario model:

The *Feature Model Strategy Algorithm* can be set to automatically select mandatory features:

FeaMiscValidateUniqueRamClassName : - Propagating the warnings messages from RAM to jUcmNav could be challenging.

- The fact that RAM does not support namespace is the root cause of the name clashes with custom RAM

aspect

FeaWorkflowInstantiationClassDiagram : - Do not underestimate, the class diagram for workflow instantiation is more complex than the class diagram for steps.
- contains associations
- contains mapping to multiple aspects (one workflow with many steps)

FeaValidateAspectMapOnlyBoundByAspectMarker : -- StephaneLeblanc - 11 Jan 2012

FeaValidateNoConditionOnBoundStartPoint : -- DanielAmyot - 12 Jan 2012

FeaValidateNoRamMetadataOnStubs : -- StephaneLeblanc - 11 Jan 2012

FeaValidateNotBoundInPaths : -- StephaneLeblanc - 11 Jan 2012

FeaValidateNotBoundOutPaths : -- StephaneLeblanc - 11 Jan 2012

FeaValidatePartialConditions : -- StephaneLeblanc - 11 Jan 2012

FeaValidateRamExpression : -- StephaneLeblanc - 11 Jan 2012

FeaValidateRecursiveRootMap : -- StephaneLeblanc - 11 Jan 2012

FeaValidateStubWithoutPluginBinding : -- StephaneLeblanc - 11 Jan 2012

FeaValidateUniqueInputName : -- StephaneLeblanc - 11 Jan 2012

FeaValidateUniqueOutputName : -- StephaneLeblanc - 11 Jan 2012

FeaValidateMisc : -- StephaneLeblanc - 19 Jan 2012

FeaTransformAsychnronousTriggerPath : -- GunterMussbacher - 04 Feb 2012

FeaTransformBlockingStub : -- GunterMussbacher - 14 Sep 2011

FeaTransformComponentBinding : -- GunterMussbacher - 10 Jan 2012

FeaTransformDeferredChoice : -- GunterMussbacher - 14 Sep 2011

FeaTransformDynamicStub :

FeaTransformSingletonMap : -- GunterMussbacher - 10 Feb 2012

FeaTransformSynchronizingStub : -- GunterMussbacher - 14 Sep 2011

FeaTransformTimer : -- GunterMussbacher - 14 Sep 2011

FeaTransformWaitingPlace : -- GunterMussbacher - 14 Sep 2011

FeaConditionalStartPoints : -- StephaneLeblanc - 29 Feb 2012

FeaPointcutExpressionValidation : All these validations rely on the definition of a pointcut expression. The

one used in FeaTransformPointcutStub is the following:

A map is a pointcut expression if
- all stubs that bound that map are pointcut stubs from the same concern AND
- all stubs contained in that map have no plugin bindings

However, this definition may be too simplistic:
- It does not allow for using pointcut expression as plugin from another pointcut expression
(Pay attention to recursive maps when dealing with this limitation)
- It allows for regular maps (maps without pointcut stubs) to have a pointcut expression as plugin
- Aspect maps (maps with p ointcut stubs) from different concern cannot reuse a pointcut expressio

Thought a bit more about this. I think the solution is to translate all maps that are assigned to a concern and to not assign pointcut maps to a concern. It is correct that they may be reused by different concerns anyway.

Hence, there would be other checks required such as: it should not happen that a regular map contains a pointcut map, but a pointcut stub may contain regular maps and pointcut maps.

-- GunterMussbacher - 23 Feb 2012

FeaStepPerConcernView : -- StephaneLeblanc - 03 Nov 2011

FeaNameTagForStep : -- GunterMussbacher - 10 Jan 2012

FeaMergeGeneratedWorkflowsWithManuallyWrittenRam : Change in the step name and in the class name that result from change at the UCM level should not be a surprise for the RAM designer in order to avoid a maintenance nightmare.

-- StephaneLeblanc - 01 Nov 2011

FeaMergeMetadataOnStepClassDiagram : -- StephaneLeblanc - 06 Jan 2012

FeaHandleSimplifiedControlFlowInActor : -- GunterMussbacher - 14 Sep 2011

FeaHandleWellNestedControlFlowInActor : -- GunterMussbacher - 14 Sep 2011

FeaTransformStartPointResponsibilityEndPoint : -- GunterMussbacher - 14 Sep 2011

FeaHandleUntaggedResponsibilityInActor : -- GunterMussbacher - 14 Sep 2011

FeaHandleTwoInputsInARow : -- GunterMussbacher - 14 Sep 2011

FeaHandleNoElementInSystem : -- GunterMussbacher - 14 Sep 2011

FeaTransformOrFork : -- GunterMussbacher - 14 Sep 2011

FeaTransformOrJoin : -- GunterMussbacher - 14 Sep 2011

FeaHandleInputBeforeDecisionPoint : -- GunterMussbacher - 14 Sep 2011

FeaNamingofRAMConcern : -- GunterMussbacher - 14 Sep 2011

FeaNamingOfRAMStep : Steps cannot cross concern boundaries

FeaTransformStaticStub : -- GunterMussbacher - 14 Sep 2011

FeaStepView : -- StephaneLeblanc - 03 Nov 2011

FeaStartupTag : - If the step contain only one node and this node is a startPoint, optimize the start point away so that there is no "empty" step

- Three kinds of bound startPoint:
boundSameConcern,boundAcrossConcerns,boundSameConcernAndAcrossConcerns

- May have conditions attached

FeaTransformResponsibilityDefinition : -- StephaneLeblanc - 03 Nov 2011

FeaWorkflowInstantiationWithoutStubs :

FeaWorkflowInstantiationWithStubs : -- StephaneLeblanc - 11 Jan 2012

FeaSystemInstantiation : -- StephaneLeblanc - 11 Jan 2012

FeaTransformAspectMarker : - Entrance aspect markers have only inBindings and exit aspect markers have only outBindings. This only impact FeaWorkflowInstantiationWithoutStubs. However, stub allows to have 0.* inBindings and 0.* outBindings. Thus, this case is "special" but does not require to be handle differently from the normal case.

FeaHandleAspectMarkersAndImplictProcessing :

FeaTransformPointcutStub : A map is a pointcut expression if
- all stubs that bound that map are pointcut stubs from the same concern AND
- all stubs contained in that map have no plugin bindings

Observe that regular map can be used as pointcut expression. For example, if a pointcut stub use a map from a different concern as plugin then the plugin is a regular map used as pointcut expression.

- Pointcut expression shall not be translated to the intermediate workflow model

FeaJucmNavIntegration : - How to propagate exception back:
jUcmNav->kermeta throws an exception: This is log in eclipse console

jUcmNav->kermeta->javaExternalCode throws an exception : This is log in a log file (aoUrnToRam.javaExternalCall.log)

- Need to call Kermeta code from java code

Impact on FeaStepView
- Pass image absolute path as argument of the transformation

FeaKermetaBugCycleInMetamodel :

FeaJucmNavCustomExportDialog : -- StephaneLeblanc - 11 Jan 2012

FeaMakeCustomizableNodeResponsibleForOrFork : -- StephaneLeblanc - 23 Feb 2012

FeaMigrateToLatestRamMetaModel : -- StephaneLeblanc - 23 Feb 2012

FeaInputDataAndOutputDataShallInheritFromBaseClass : -- StephaneLeblanc - 23 Feb 2012

FeaTransformAndFork : -- GunterMussbacher - 14 Sep 2011

FeaTransformAndJoin : -- GunterMussbacher - 14 Sep 2011

FeaDisjunctiveStubEntries : -- StephaneLeblanc - 22 Mar 2012

FeaJucmNavSetup :

FeaDeveloperGuide : -- StephaneLeblanc - 20 Mar 2012

FeaConferencePaper : -- StephaneLeblanc - 20 Mar 2012

FeaDefaultExecuteOperationOnCustomizableNode : -- StephaneLeblanc - 04 Apr 2012

FeaConvertSpaceToUnderscore : Using camel case instead of underscore was also consider as a substitute for space. However, we chose to use underscore since it will facilitate the implementation of Fea Misc Validate Unique Ram Class Name as a jUcmNav OCL constrain.

For example, "MyVar very long name" and "MyVar_very_long_name" should be considered equal.

Number of topics: 65