Eclipse Java Workflow Tooling (JWT)

Transformations from JWT to STP-IM

- **Abstract**: This document describes a set of rules implemented in ATL from the Java Workflow Tooling (JWT) metamodel to the metamodel of the SOA Tools Platform Intermediate Model (STP-IM) as well as concepts of both metamodels

- The following persons have been involved in the specification:
  - Marius Brendle, University of Augsburg, Germany
  - Stephan Malike, University of Augsburg, Germany
  - Christian Pallay, University of Augsburg, Germany
  - Florian Lautenbacher, University of Augsburg, Germany (supervisor)

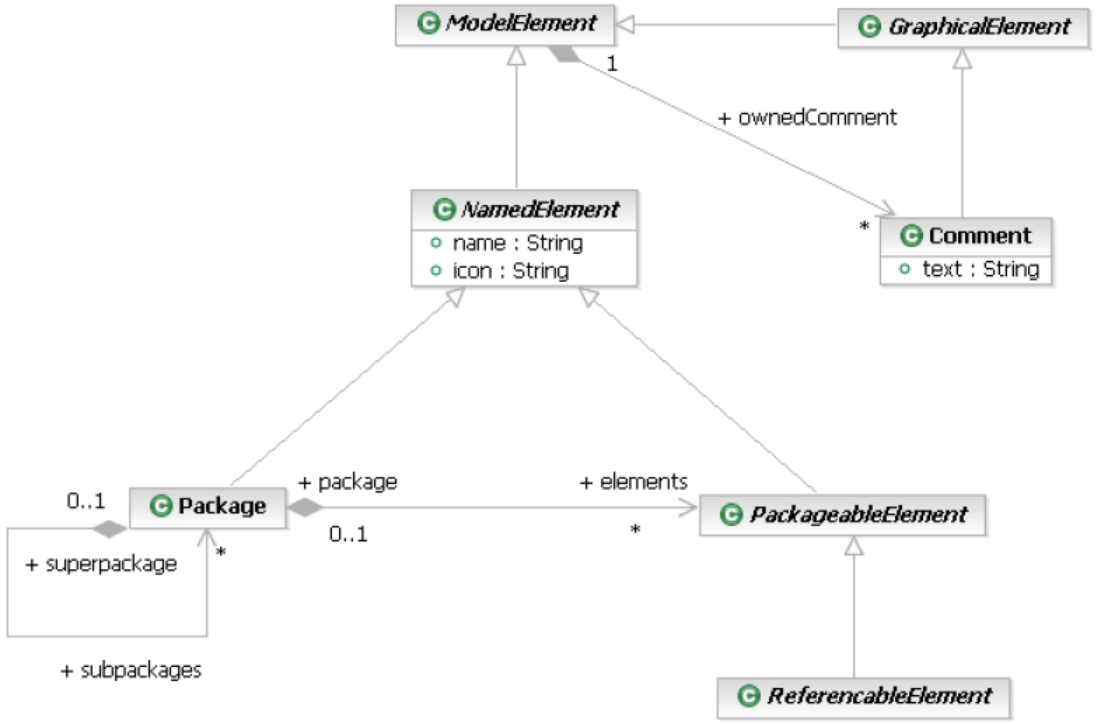
- There exists an exhaustive description of both metamodels and all transformation rules, but alas it is only available in German for the moment. These slides summarize the outcome.

JWT: Current metamodel

- Package View (only used for graphical display)

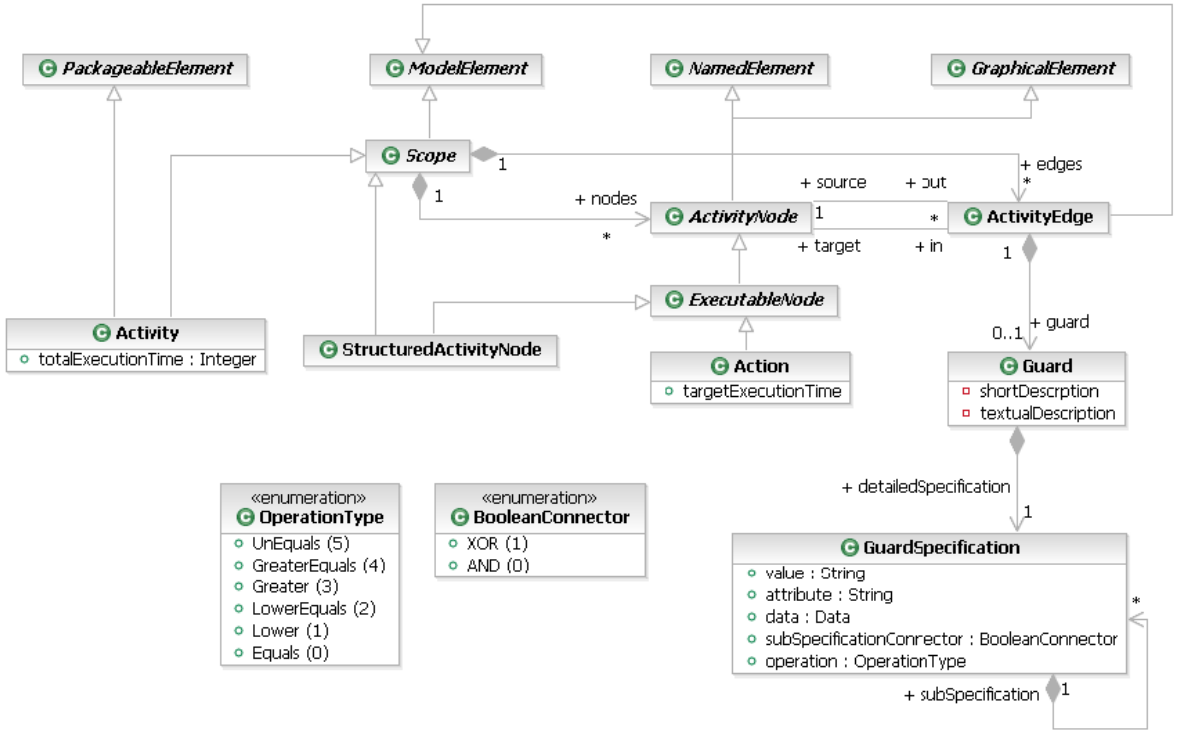
JWT: Current metamodel

- Package Core (allows naming and packaging of model elements)



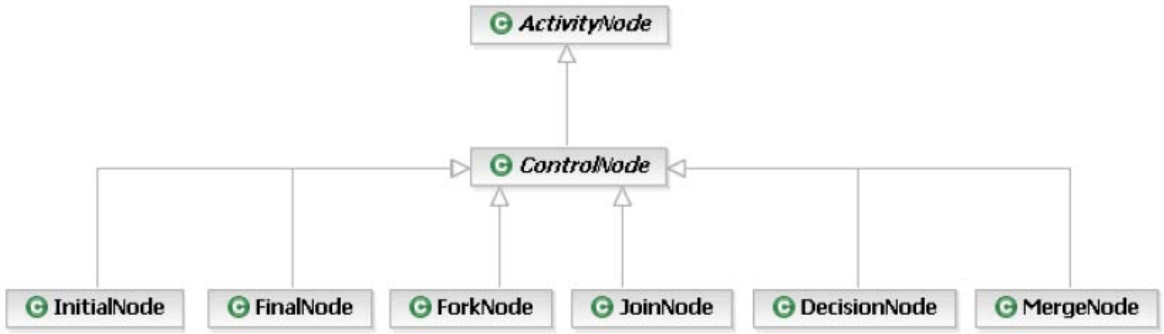
JWT: Current metamodel

- Package Processes (Basic elements to model a process: nodes and edges among others)



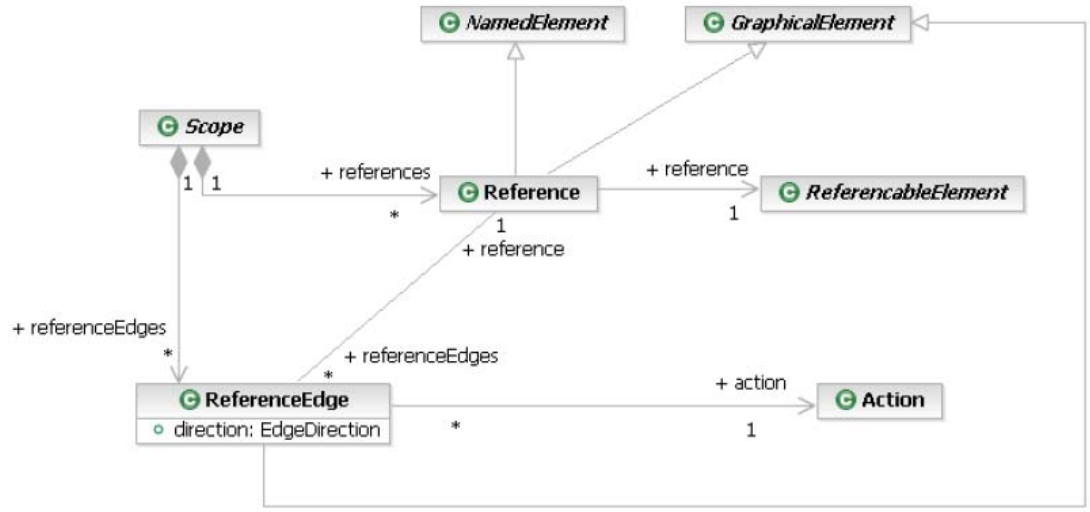
JWT: Current metamodel

- Package ControlNodes (allow alternative or parallel threads in a process model)



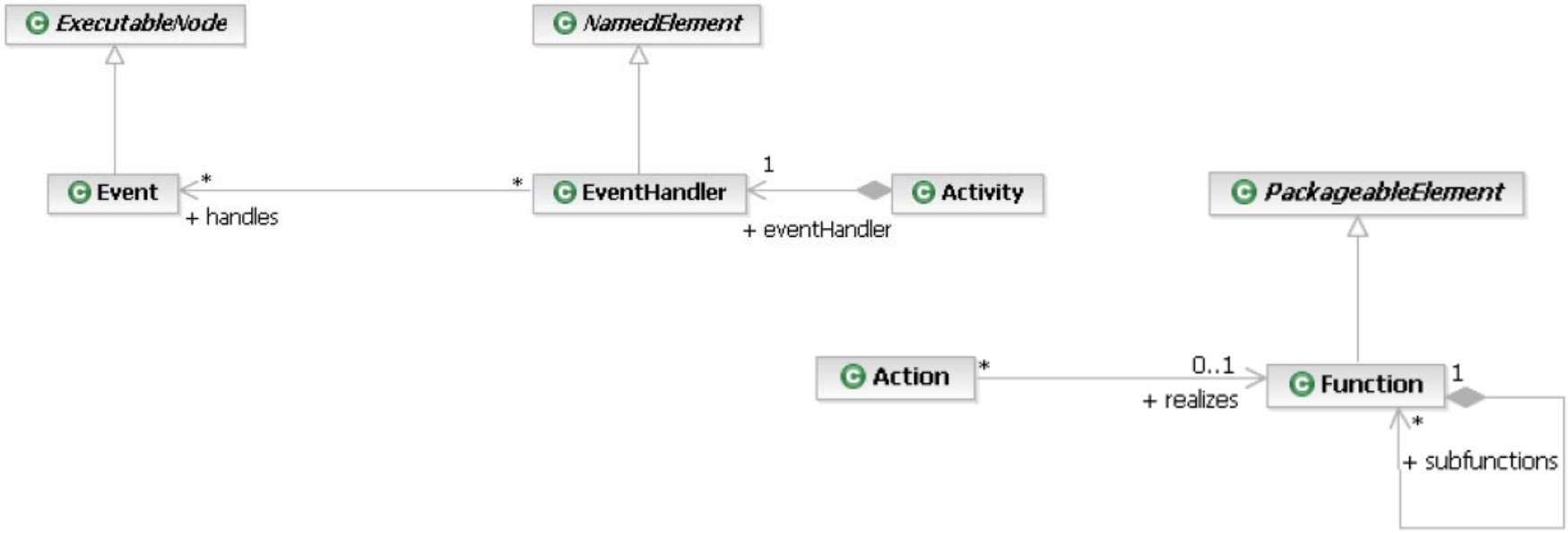
JWT: Current metamodel

- Package References (to connect Actions with ReferenceableElements such as Role, Data or Application)



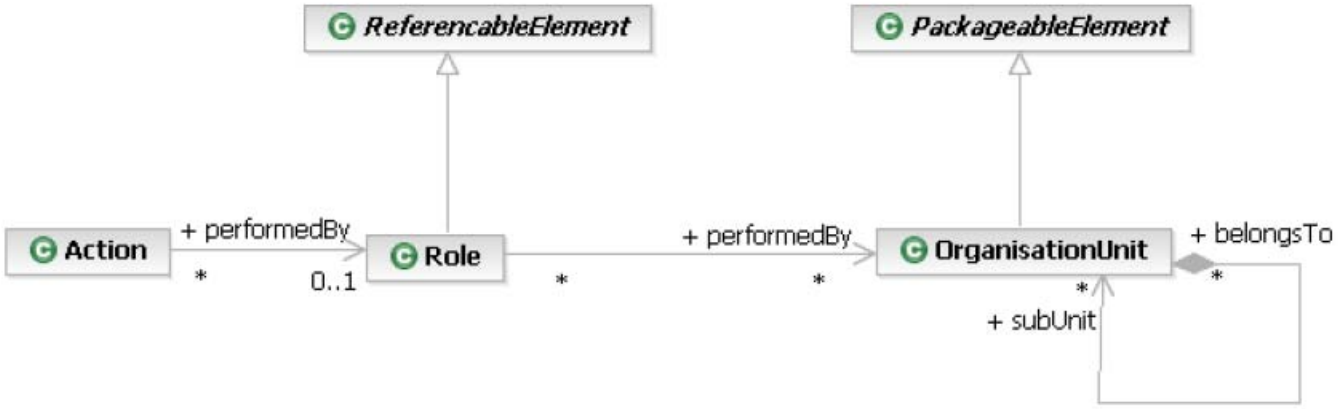
JWT: Current metamodel

- Package Events and Functions (both not really used right now)



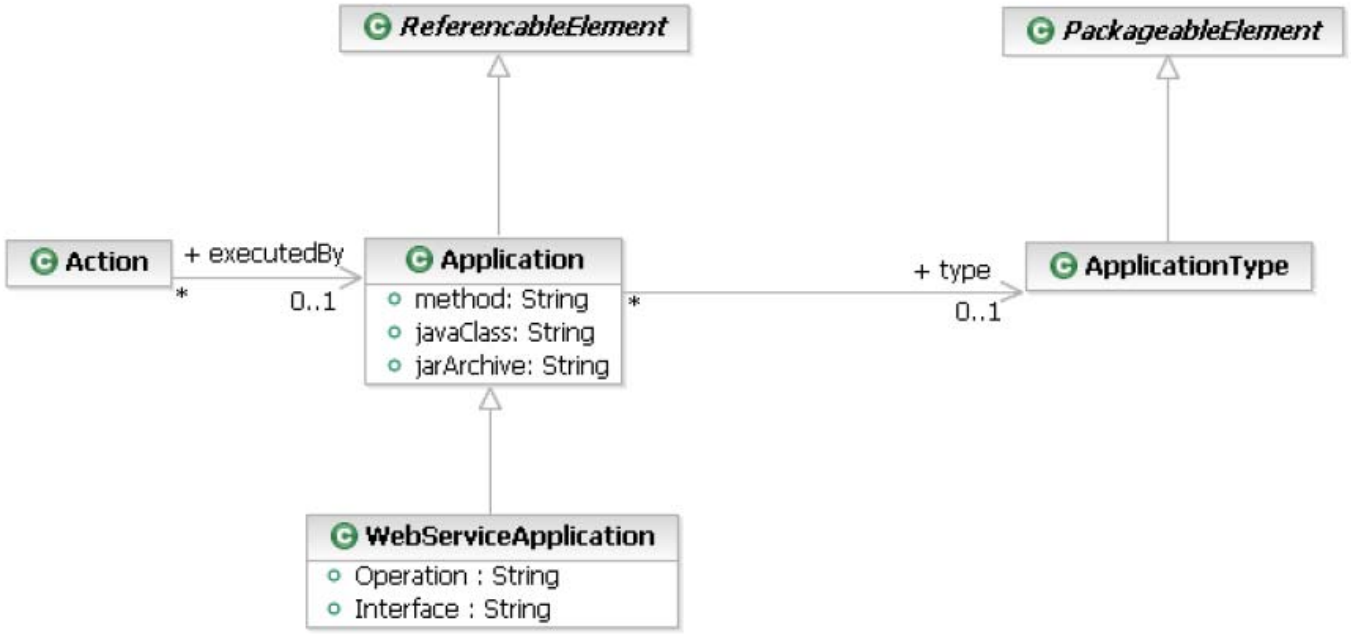
JWT: Current metamodel

- Package Organisations (to specify who is responsible for a task)



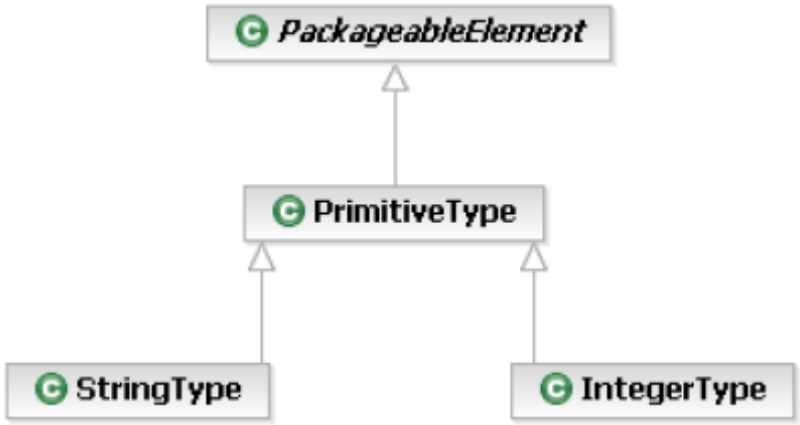
JWT: Current metamodel

- Package Applications (what shall be invoked)



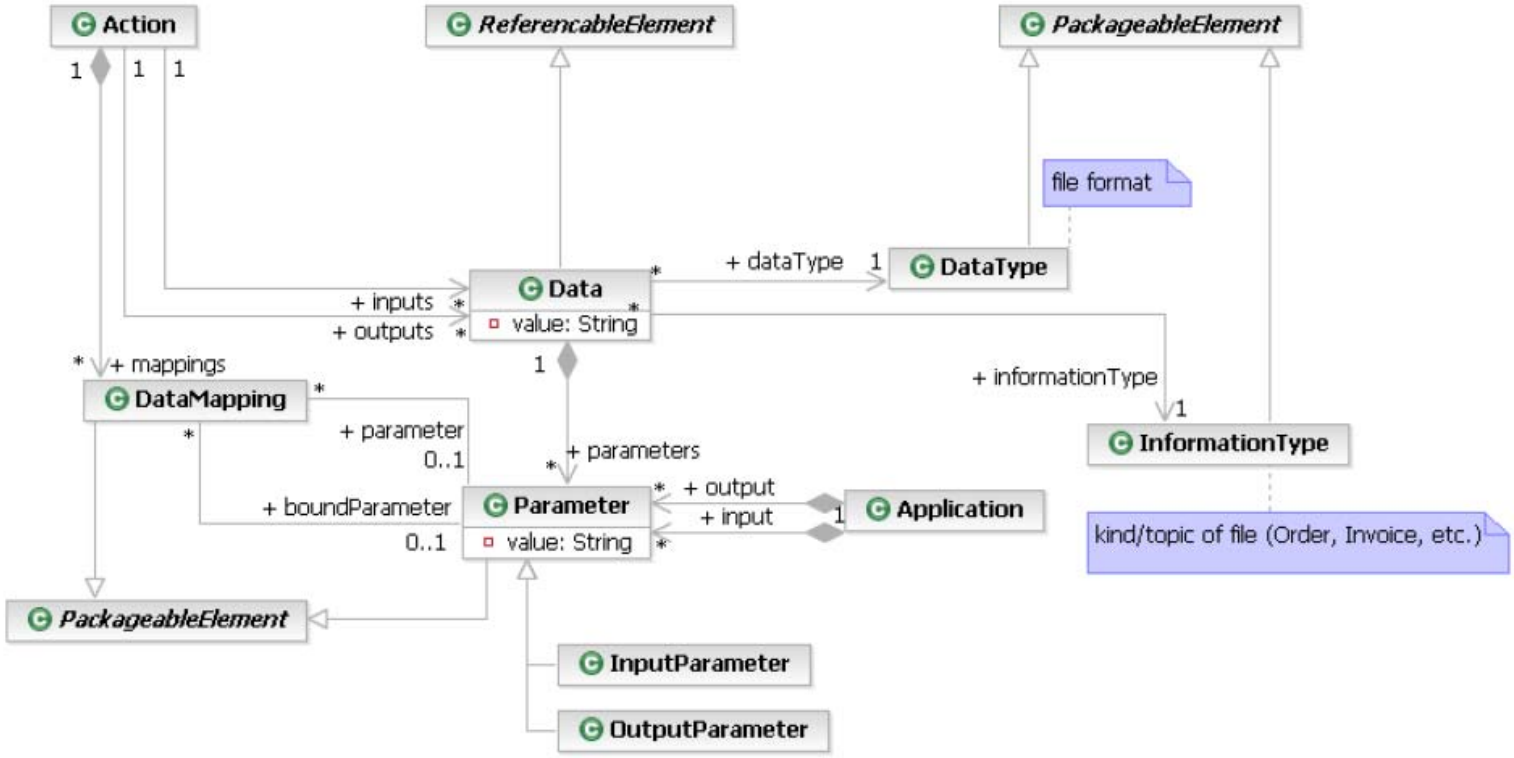
JWT: Current metamodel

- Package PrimitiveTypes (to allow the usage of data types, not used)



JWT: Current metamodel

- Package Data (the files and values that are need in an application)

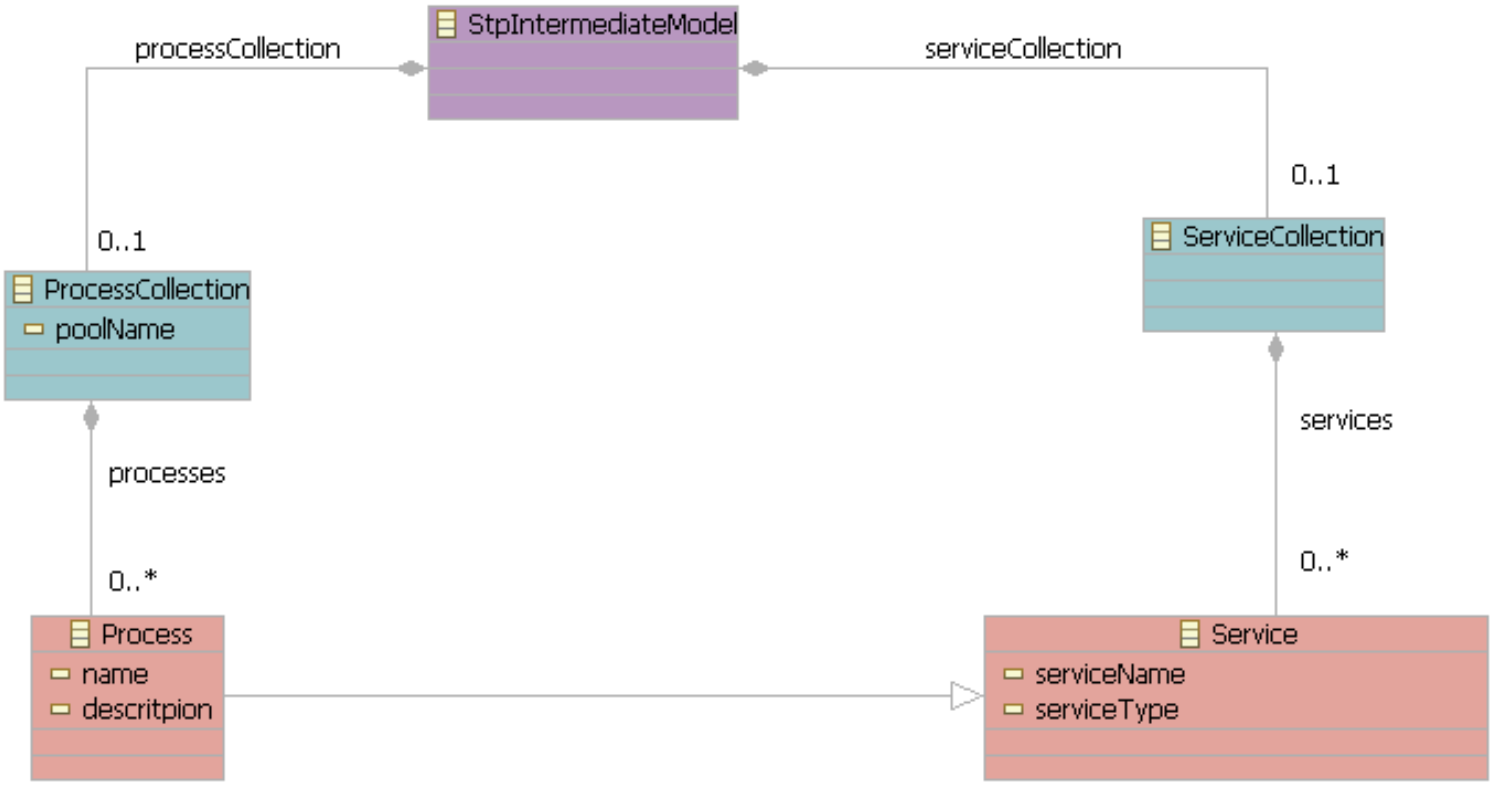


JWT: Current metamodel

- During the work on the transformation rules, it came clear that some parts were added in the last weeks/month that are not graphically displayed right now:
  - ActivityLinkNode: to invoke one Activity from another (similar to CallBehaviorAction in UML)
  - Model: includes information about who the model created, when, some description, etc.

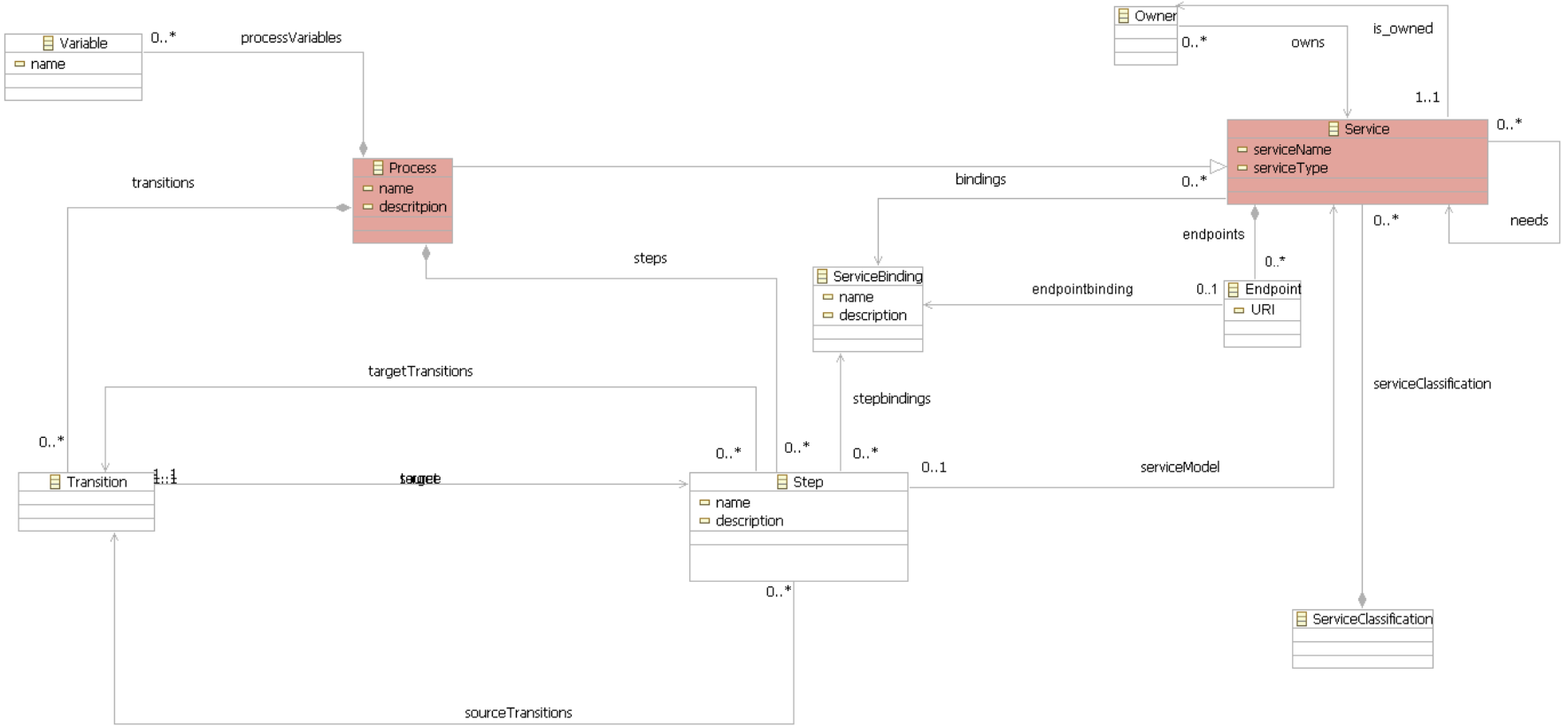
STP-IM: Current metamodel

- Package Root (STP-IM includes Processes and Services)



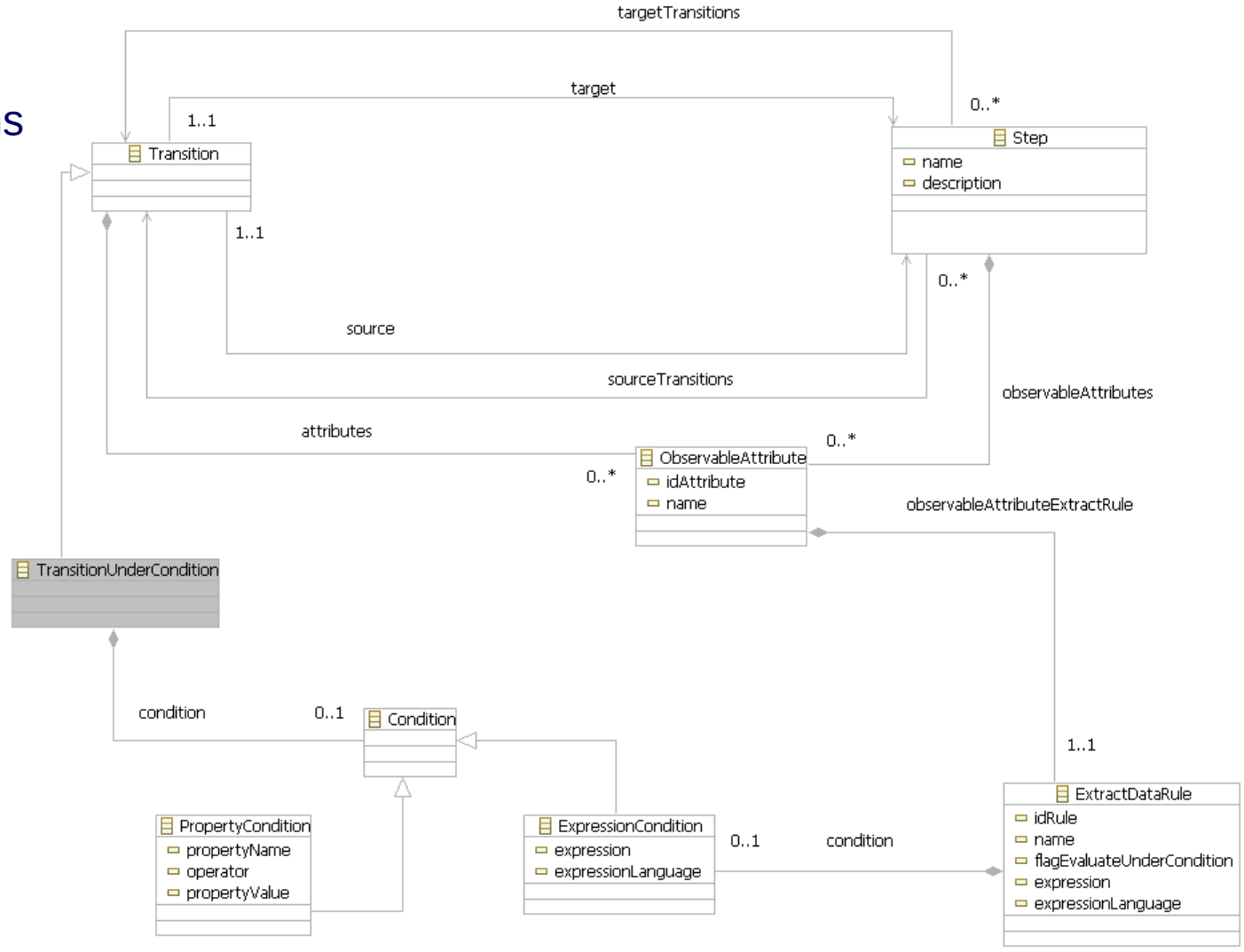
STP-IM: Current metamodel

- Package Core (how services and processes are connected)



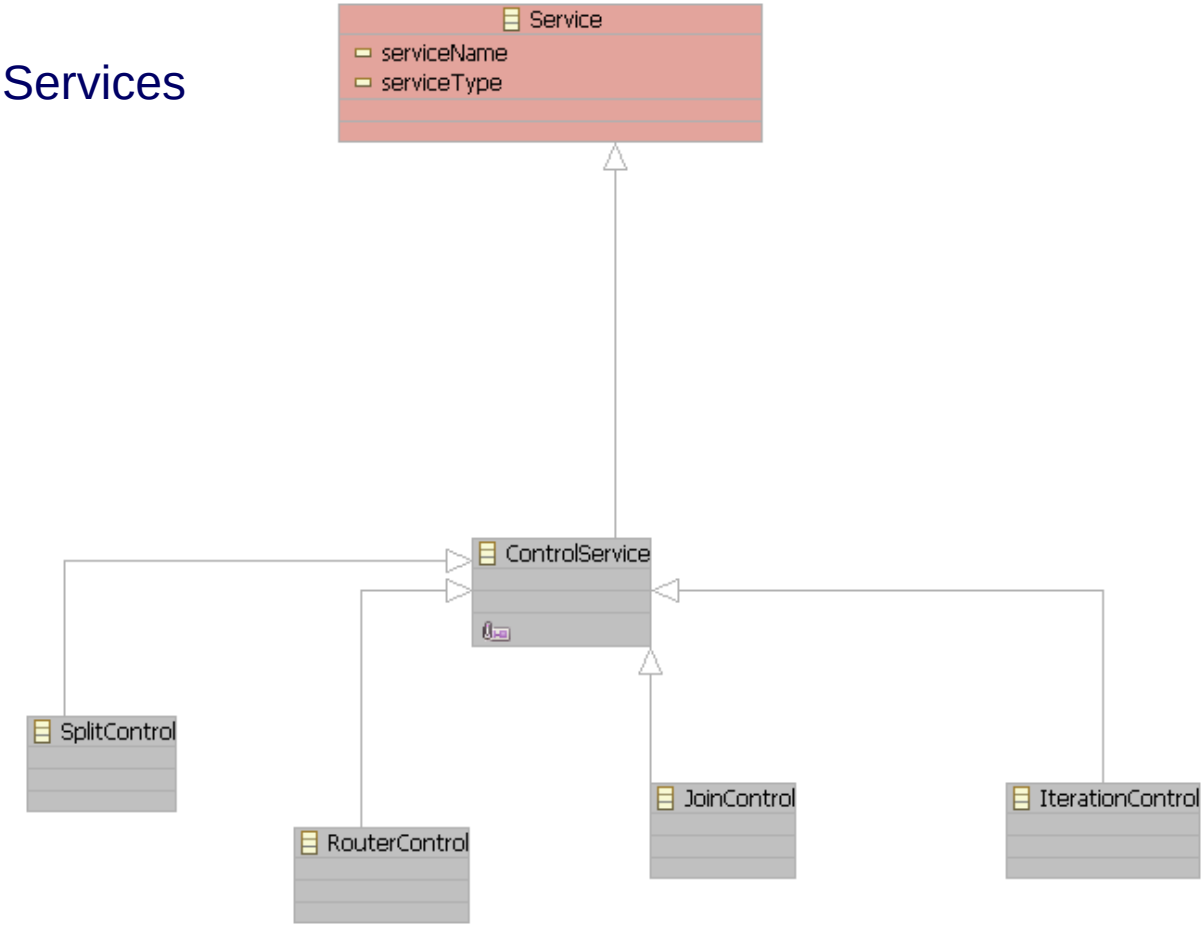
STP-IM: Current metamodel

- Transitions



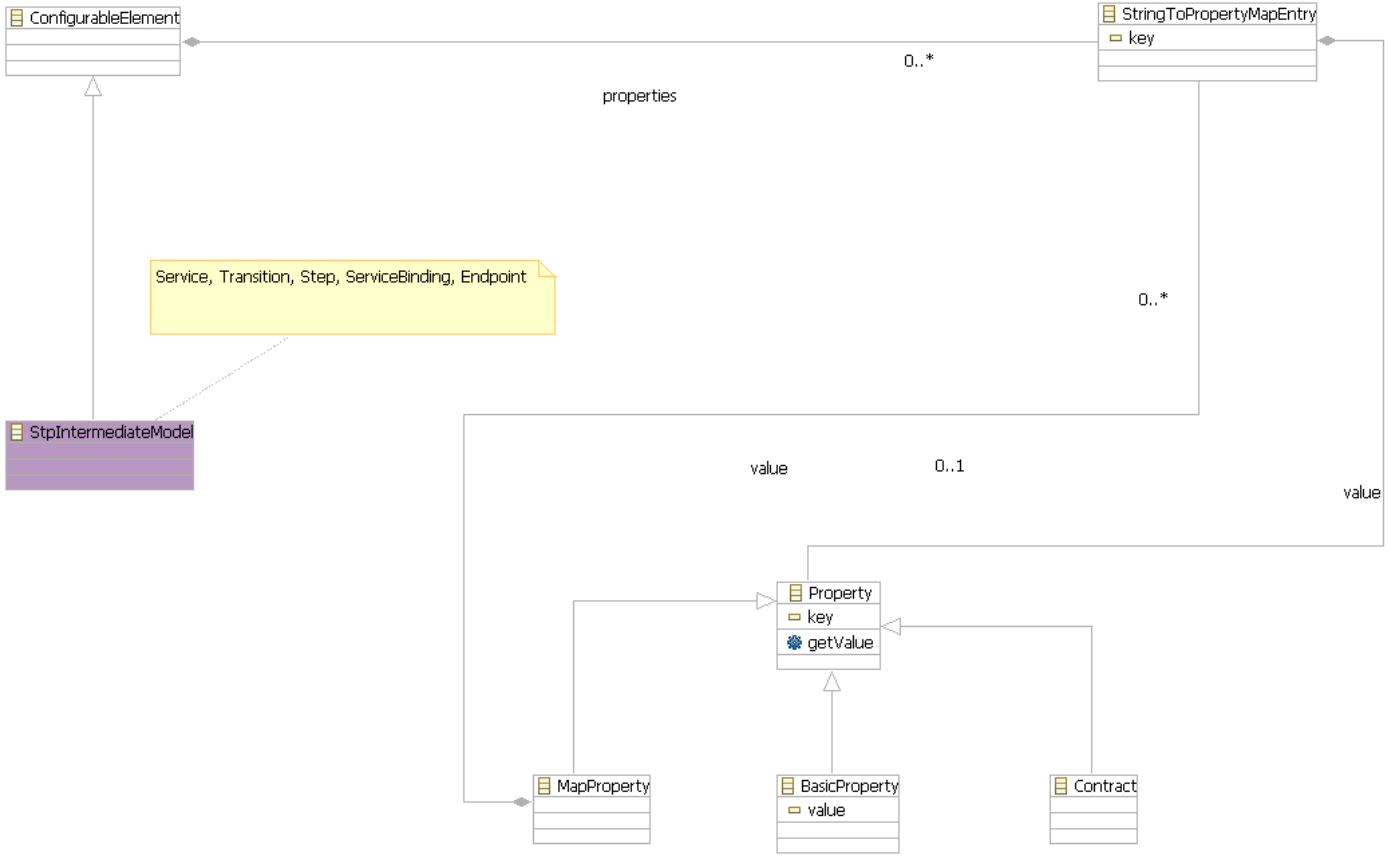
STP-IM: Current metamodel

- ControlServices



STP-IM: Current metamodel

- ConfigurableElement



Transformation between JWT and STP-IM

- Currently only one way: from JWT to STP-IM
- First we will describe which concepts from JWT have a similar concept in STP-IM and which don‘t
- A table is available in the German description where one can see at a glance which concepts are mapped where.

- The transformations have been implemented with ATL.
- Several examples have been tested and will be committed also.

Transformable concepts

- JWT:Activity gets to STP:Process (STP:Service would have been another solution, but then there would be no transitions and steps)
- JWT:ActivityEdge becomes a STP:Transition
- JWT:StructuredActivityNode is similar to a JWT:Activity and therefore another kind of STP:Process. On the other side, JWT:StructuredActivityNode is also an JWT:ExecutableNode which can be solved in creating a new STP:Step that is connected through a STP:Service with a STP:Process
- JWT:ActivityLinkNode points to an JWT:Activity, so a STP:Step is introduced which links to another STP:Process (similar described above).
- JWT:Action is a STP:Step.
- JWT:Guard is a condition and hence part of a STP:TransitionUnderCondition.

Transformable concepts (cont.)

- JWT:GuardSpecification is similar to a STP:PropertyCondition
- All kind of JWT:ControlNodes such as InitialNode, FinalNode, ForkNode, JoinNode, DecisionNode and MergeNode become STP:Steps.
- JWT:Data becomes a STP:ServiceBinding and JWT:Role a STP:Owner.
- JWT:OrganisationUnit becomes a property of STP:Owner (as soon as STP:Owner becomes a ConfigurableElement)
- JWT:Events become STP:Steps (does not fit completely, the other idea would be to create a property on a STP:Process somewhere)
- JWT:Function becomes a property of STP:Step.
- JWT:Application gets a STP:Service.
- JWT:WebServiceApplication gets also a STP:Service. JWT:ApplicationType becomes property of STP:Service.

Transformable concepts (cont.)

- JWT:Data becomes STP:ServiceBinding (does not fit completely!)
- JWT:DataType,InformationType, Parameter, etc. become properties of STP:ServiceBinding
- JWT:DataMapping becomes a property of STP:ServiceBinding
- JWT:Model and its properties become STP:StpIntermediateModel.

Concepts that are not transformable

- JWT:GraphicalElement, Point, Dimension, EdgeDirection: since they are only necessary for graphical display.
- JWT – Abstract classes in the metamodel such as ModelElement, NamedElement, Scope, ActivityNode, etc. will never be modeled.
- JWT:Comment is not used right now since the support for modeling a comment is still missing in the workflow editor and there is no useful concept in STP-IM similar to it.
- JWT:Package could be a STP:ProcessCollection, but packages can be nested whereas process collections can‘t.
- JWT:EventHandler is not visible and only part of a JWT:Activity.

Problems we had to cope with & Outlook

- During the implementation phase several problems were detected:
  - Description of the JWT metamodel is not actual (missing Model, no ActivityLinkNode, problems with GuardSpecifications)
  - In STP-IM: Owner is not of type ConfigurableElement and has no attributes; hence, no names of a role can be transformed
  - In STP-IM: ServiceClassification is not of type ConfigurableElement
  - Both metamodels are still in development and will be changed sooner or later; hence, the transformations will need to be adapted as soon as something has changed.

- For problems in using the transformation, please report on Bug #244825 or ask the jwt-dev@eclipse.org – mailing list.