

17. Adding Modularity to a Domain-Specific Language with the Reuseware Tool

Prof. Dr. Uwe Aßmann
Technische Universität
Dresden
Institut für Software- und
Multimediatechnik
<http://st.inf.tu-dresden.de>
Version 11-0.2, 01.12.11

- 1) The DSL Taipan
- 2) Reuseware
- 3) Extending the metamodel of Taipan for modularity
- 4) Reuseware tool



SEW, © Prof. Uwe Aßmann

1

17.1 Building Modularisation into Taipan DSL

- ▶ Languages need modularization concepts
 - Reduce complexity
 - Improve reusability
- ▶ Challenges
 - Modularization influences syntax and semantics
 - Requires additional tooling support
- ▶ Reuseware toolkit ^{[1][2]}
 - Does not influence design of DSL syntax or semantics
 - DSL syntax can be extended at the end
 - Composes modularized models to monolithic models
 - DSL semantics do not require extension
 - Generic tooling can be used with arbitrary DSLs

Obligatory Literature

- ▶ [1] Jakob Henriksson, Jendrik Johannes, Steffen Zschaler, and Uwe Aßmann. Reuseware - adding modularity to your language of choice. *Journal of Object Technology*, 6(9):127-146, 2007. On Language-Independent Model Modularisation, *Transactions on Aspect-Oriented Development*, 2008
- ▶ [2] <http://reuseware.org>



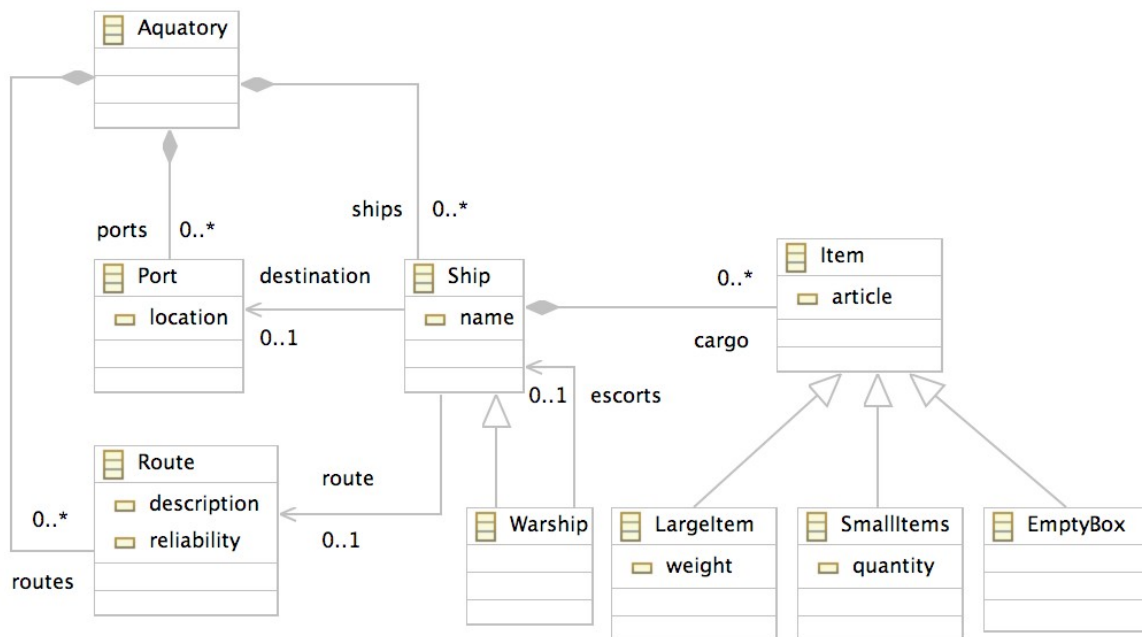
Building Modularisation into a DSL

- ▶ Reuseware approach
 - Define a *composition system* with modularisation concepts (see CBSE course)
 - Composition systems define component model
 - E.g., Modules, Packages, Aspects, etc.
 - Composition techniques
 - E.g., parameterization, extension, weavings
 - And composition languages
 - For the structure in the large
 - Optional: Extend DSL syntax with concepts for variation points
 - Variation points allow definition of templates
 - Define a reuse extension for your DSL
 - Binds the composition system to your DSL
 - E.g., what are the specifics of a module in your DSL, what identifies an aspect, etc.
- Reuseware can handle modularization in your DSL



Building a DSL: Modularisation - Example

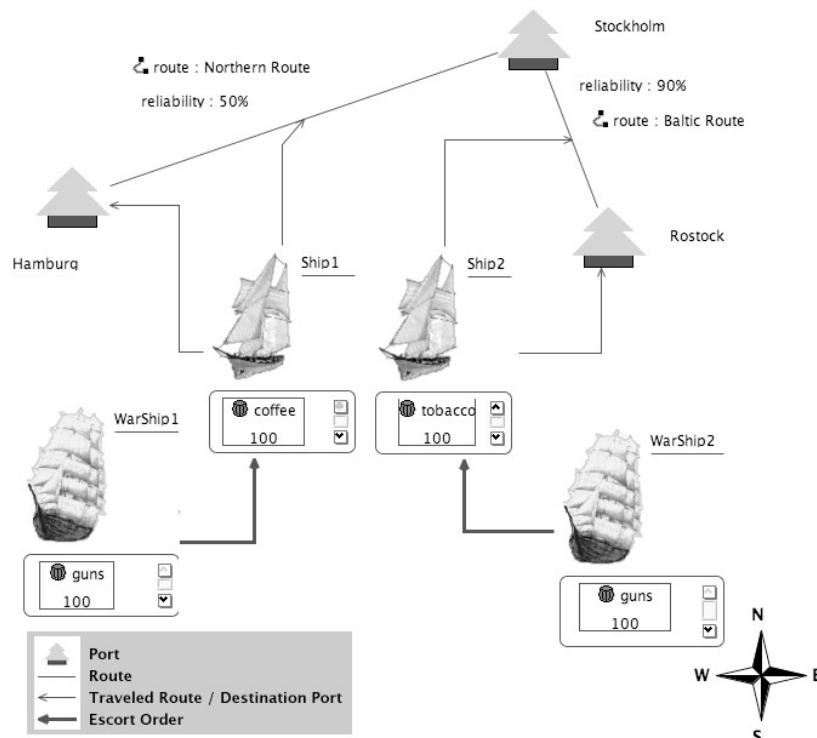
- Taipan DSL^[3] for modeling ship fleets (Metamodel excerpt)



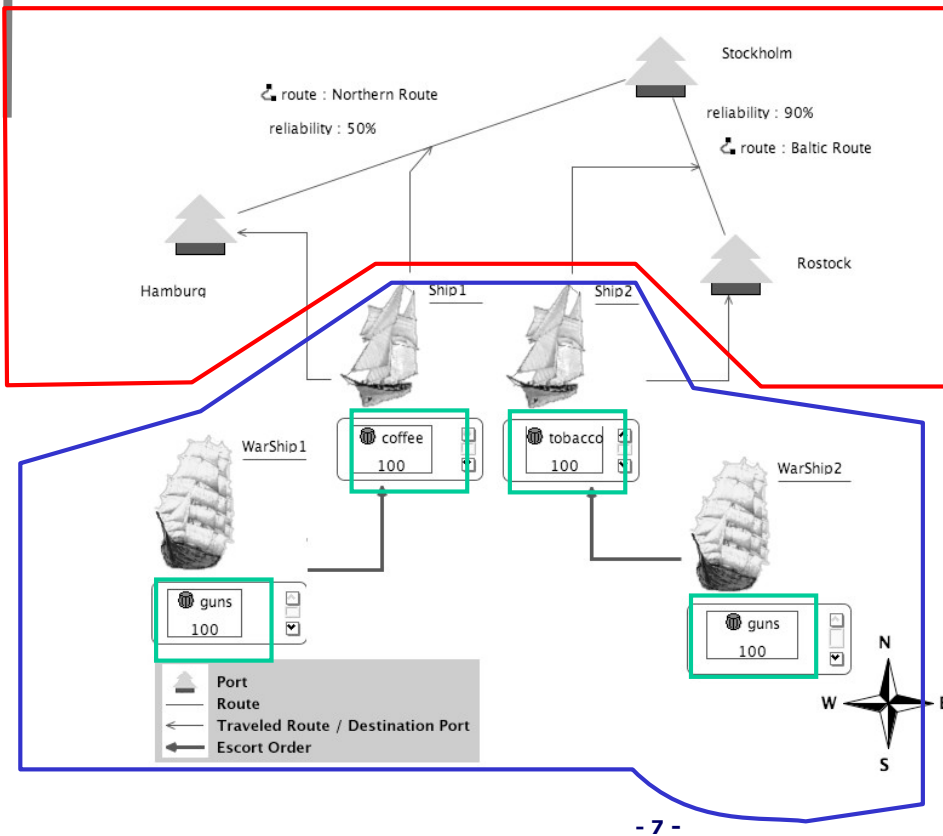
[3] http://wiki.eclipse.org/index.php/GMF_Tutorial#Quick_Start



Building a DSL: Modularisation - Example



Building a DSL: Modularisation - Example



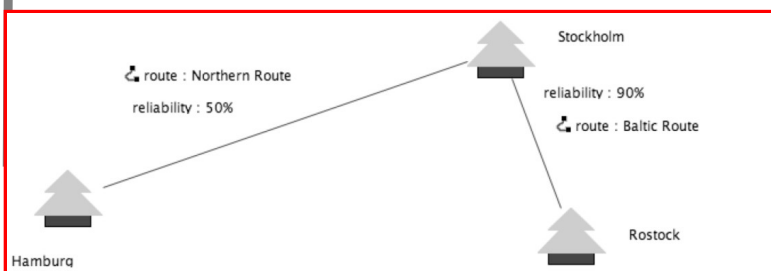
Different concerns should be separated into model fragments

- Port model (configuration of ports and routes)

- Flotilla model (ships and their relations)

- Cargo model (Cargo and its properties)

Building a DSL: Modularisation - Example

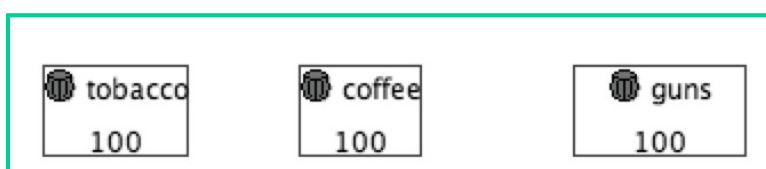
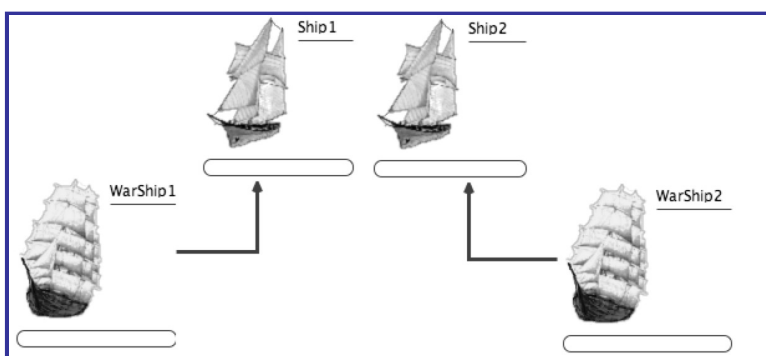


Different concerns should be separated into model fragments

- Port model (configuration of ports and routes)

- Flotilla model (ships and their relations)

- Cargo model (Cargo and its properties)



17.2 Reuseware - Overview

- ▶ **Model fragments** (model snippets) are partial models that may contain variation points
 - Offer a **Composition Interface**
 - **Composition Interface** consists of **Ports**
 - **Ports** point at elements of the model fragment that can be accessed for composition
- ▶ Composition Programs
 - Define **composition links** between Ports
 - Can be executed to produce a composed model where model fragments are merged at the elements pointed out by the linked Ports



Building a DSL: Reuseware - Overview

- ▶ Composition Systems
 - Define modularisation concepts (e.g., Modules, Packages, Aspects)
 - Define relations between modularisation concepts (e.g, an aspect relates to a core)
- ▶ Reuse extensions (for DSLs)
 - Define how modularization concepts defined in a composition system are realized in a concrete DSL
 - Define which ports are related to which model elements of a model fragment



Defining Composition Systems with Reuseware

- ▶ A composition system defines fragment components with
 - Fragment roles
 - Role a model fragment plays in the modularisation (e.g., aspect or core)
 - Fragment roles collaborate through associations between ports
 - Static ports of a fragment component
 - Defined for one fragment role
 - Each fragment playing the role has to offer the port
 - Dynamic ports
 - Defined for one fragment role
 - Each fragment playing the role can offer several of these ports
 - Contribution Associations
 - Defines that two ports are related
 - Executing a composition link between the two ports will trigger the copying of model elements
 - Configuration Associations
 - Defines that two ports are related
 - Executing a composition link between the two ports will NOT trigger the copying of model elements



ReuseTaipan - a Composition System for the Taipan Metamodel

```
compositionsystem reuseTaipan {  
  
    fragment role TravelSpace {  
        static port VehicleContainer;  
        dynamic port Routes;  
        dynamic port Places;  
    }  
  
    fragment role Flotilla {  
        static port Vehicles;  
        dynamic port RouteSlots;  
        dynamic port PlaceSlots;  
    }  
  
    contribution Flotilla.Vehicles --> TravelSpace.VehicleContainer;  
    configuration Flotilla.RouteSlots --> TravelSpace.Routes;  
    configuration Flotilla.PlaceSlots --> TravelSpace.Places;  
  
    fragment role ItemHolder {  
        dynamic port ItemSpaces;  
    }  
  
    fragment role ItemContainer {  
        dynamic port Items;  
    }  
  
    contribution ItemContainer.Items --> ItemHolder.ItemSpaces;  
}
```



Building a DSL: ReuseTaipan - a Composition System

```
compositionsystem reuseTaipan {  
  fragment role TravelSpace {  
    static port VehicleContainer;  
    dynamic port Routes;  
    dynamic port Places;  
  }  
  
  fragment role Flotilla {  
    static port Vehicles;  
    dynamic port RouteSlots;  
    dynamic port PlaceSlots;  
  }  
  
  contribution Flotilla.Vehicles --> TravelSpace.VehicleContainer;  
  configuration Flotilla.RouteSlots --> TravelSpace.Routes;  
  configuration Flotilla.PlaceSlots --> TravelSpace.Places;  
  
  fragment role ItemHolder {  
    dynamic port ItemSpaces;  
  }  
  
  fragment role ItemContainer {  
    dynamic port Items;  
  }  
  
  contribution ItemContainer.Items --> ItemHolder.ItemSpaces;  
}
```

A **TravelSpace** offers a place where vehicles can be placed (**VehicleContainer**) and a number of **Routes** and **Places**



- 13 -

Prof. U. Aßmann, SEW

13

Building a DSL: ReuseTaipan - a Composition System

```
compositionsystem reuseTaipan {  
  fragment role TravelSpace {  
    static port VehicleContainer;  
    dynamic port Routes;  
    dynamic port Places;  
  }  
  
  fragment role Flotilla {  
    static port Vehicles;  
    dynamic port RouteSlots;  
    dynamic port PlaceSlots;  
  }  
  
  contribution Flotilla.Vehicles --> TravelSpace.VehicleContainer;  
  configuration Flotilla.RouteSlots --> TravelSpace.Routes;  
  configuration Flotilla.PlaceSlots --> TravelSpace.Places;  
  
  fragment role ItemHolder {  
    dynamic port ItemSpaces;  
  }  
  
  fragment role ItemContainer {  
    dynamic port Items;  
  }  
  
  contribution ItemContainer.Items --> ItemHolder.ItemSpaces;  
}
```

A **Flotilla** offers a set of **Vehicles** and has a number of placeholders for routes (**RouteSlots**) and places (**PlaceSlots**)



- 14 -

Prof. U. Aßmann, SEW

14

Building a DSL: ReuseTaipan - a Composition System

```
compositionsyntax reuseTaipan {  
  
  fragment role TravelSpace {  
    static port VehicleContainer;  
    dynamic port Routes;  
    dynamic port Places;  
  }  
  
  fragment role Flotilla {  
    static port Vehicles;  
    dynamic port RouteSlots;  
    dynamic port PlaceSlots;  
  }  
  
  contribution Flotilla.Vehicles --> TravelSpace.VehicleContainer;  
  configuration Flotilla.RouteSlots --> TravelSpace.Routes;  
  configuration Flotilla.PlaceSlots --> TravelSpace.Places;  
  
  fragment role ItemHolder {  
    dynamic port ItemSpaces;  
  }  
  
  fragment role ItemContainer {  
    dynamic port Items;  
  }  
  
  contribution ItemContainer.Items --> ItemHolder.ItemSpaces;  
}
```

A Flotilla contributes Vehicles to a TravelSpace's VehicleContainer; a RouteSlots can be configured with a Route; a PlaceSlots can be configured with a Place



Building a DSL: ReuseTaipan - a Composition System

```
compositionsyntax reuseTaipan {  
  
  fragment role TravelSpace {  
    static port VehicleContainer;  
    dynamic port Routes;  
    dynamic port Places;  
  }  
  
  fragment role Flotilla {  
    static port Vehicles;  
    dynamic port RouteSlots;  
    dynamic port PlaceSlots;  
  }  
  
  contribution Flotilla.Vehicles --> TravelSpace.VehicleContainer;  
  configuration Flotilla.RouteSlots --> TravelSpace.Routes;  
  configuration Flotilla.PlaceSlots --> TravelSpace.Places;  
  
  fragment role ItemHolder {  
    dynamic port ItemSpaces;  
  }  
  
  fragment role ItemContainer {  
    dynamic port Items;  
  }  
  
  contribution ItemContainer.Items --> ItemHolder.ItemSpaces;  
}
```

An ItemHolder offers different ItemSpaces



Building a DSL: ReuseTaipan - a Composition System

```
compositionsystem reuseTaipan {  
  
  fragment role TravelSpace {  
    static port VehicleContainer;  
    dynamic port Routes;  
    dynamic port Places;  
  }  
  
  fragment role Flotilla {  
    static port Vehicles;  
    dynamic port RouteSlots;  
    dynamic port PlaceSlots;  
  }  
  
  contribution Flotilla.Vehicles --> TravelSpace.VehicleContainer;  
  configuration Flotilla.RouteSlots --> TravelSpace.Routes;  
  configuration Flotilla.PlaceSlots --> TravelSpace.Places;  
  
  fragment role ItemHolder {  
    dynamic port ItemSpaces;  
  }  
  
  fragment role ItemContainer {  
    dynamic port Items;  
  }  
  
  contribution ItemContainer.Items --> ItemHolder.ItemSpaces;  
}
```

An **ItemContainer** contains and offers **Items**



- 17 -

Prof. U. Aßmann, SEW

17

Building a DSL: ReuseTaipan - a Composition System

```
compositionsystem reuseTaipan {  
  
  fragment role TravelSpace {  
    static port VehicleContainer;  
    dynamic port Routes;  
    dynamic port Places;  
  }  
  
  fragment role Flotilla {  
    static port Vehicles;  
    dynamic port RouteSlots;  
    dynamic port PlaceSlots;  
  }  
  
  contribution Flotilla.Vehicles --> TravelSpace.VehicleContainer;  
  configuration Flotilla.RouteSlots --> TravelSpace.Routes;  
  configuration Flotilla.PlaceSlots --> TravelSpace.Places;  
  
  fragment role ItemHolder {  
    dynamic port ItemSpaces;  
  }  
  
  fragment role ItemContainer {  
    dynamic port Items;  
  }  
  
  contribution ItemContainer.Items --> ItemHolder.ItemSpaces;  
}
```

Items can be individually assigned to **ItemSpaces**



- 18 -

Prof. U. Aßmann, SEW

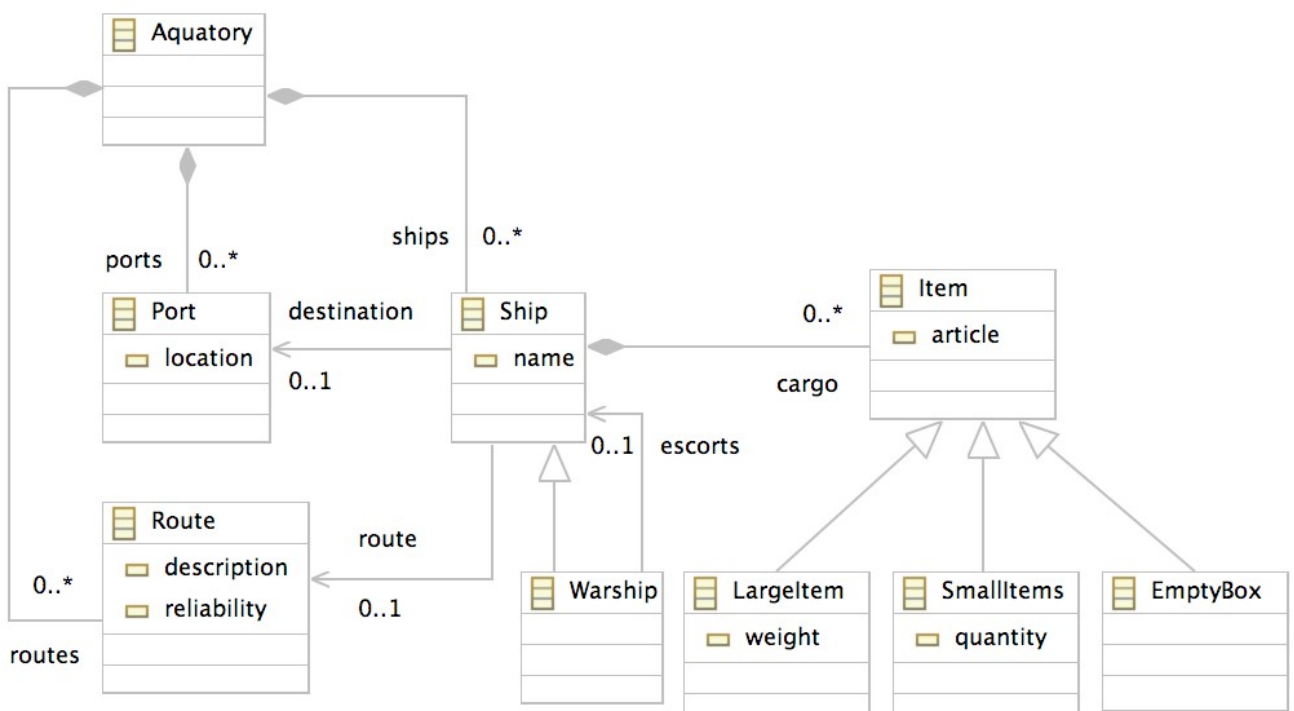
18

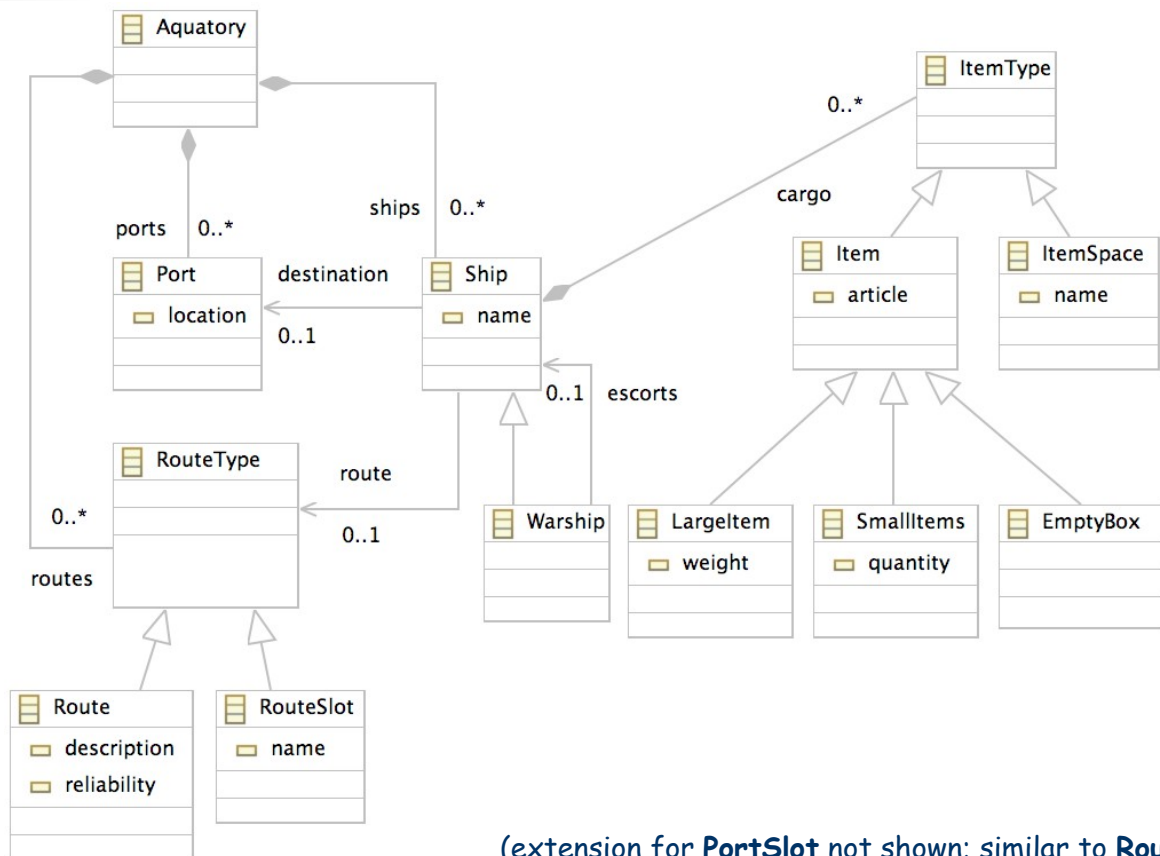
17.3 Building a DSL: Extending a Metamodel for Variation

- ▶ Three kinds of variation points required
 - RouteSlot
 - PortSlot
 - ItemSpace
- ▶ For each kind of variation point we...
 - Introduce a superclass for the metaclass that defines the elements which may replace the variation point
 - e.g., we introduce **RouteType** as a superclass of **Route** in the case of RouteSlot
 - We redirect all references to the metaclass to the new superclass
 - e.g., all references to **Route** are redirected to **RouteType**
 - We introduce a new subclass for the just introduced superclass that represents the variation point. This class needs properties from which a name can be derived.
 - e.g., we introduce **RouteSlot** as a subclass of **RouteType**



The Taipan Metamodel





(extension for **PortSlot** not shown; similar to **RouteSlot**)



Building a DSL: Reuseware - Reuse Extensions

- ▶ A Reuse Extension defines
 - How a composition interface defined by a fragment role (which is defined in a composition system) is linked to the content of a model fragment
 - Each port links to a set of model elements treated as:
 - **Prototype**: Element that can be copied with its contained elements
 - **Anchor**: Element that can be referenced by other elements
 - **Hook**: Variation point where Prototypes can be put
 - **Slot**: Variation point where Anchors can be put



Building a DSL: Binding ReuseTaipan to Taipan DSL

```
reuseextension reuseTaipan implements reuseTaipan
epackages <http://www.eclipse.org/examples/gmf/taipan>
Rootclass TravelSpace {
  fragment role TravelSpace {
    port VehicleContainer {
      Aquatory.ships is hook {}
      Aquatory.ports is hook {}
      Aquatory.routes is hook {}
    }
    port Routes {
      Route is anchor {
        port expr = $self.description$
      }
    }
    port Places {
      Port is anchor {
        port expr = $self.location.concat('Port')$
      }
    }
  }

  fragment role Flotilla {
    port Vehicles {
      Aquatory.ships is prototype {}
      Aquatory.ports is prototype {}
      Aquatory.routes is prototype {}
    }
    port RouteSlots {
      RouteSlot is slot {
        port expr = $self.name$
      }
    }
    port PlaceSlots {
      PortSlot is slot {
        port expr = $self.name$
      }
    }
  }
}
...
}
```

The ReuseTaipan composition system is bound to the Taipan DSL (referred to by the URI of its metamodel)

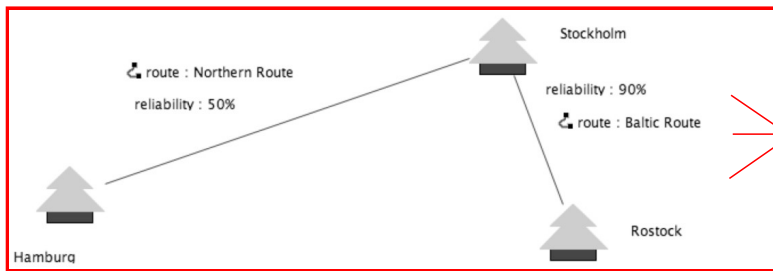
Building a DSL: Binding ReuseTaipan to Taipan DSL

```
reuseextension reuseTaipan implements reuseTaipan
epackages <http://www.eclipse.org/examples/gmf/taipan>
Rootclass TravelSpace {
  fragment role TravelSpace {
    port VehicleContainer {
      Aquatory.ships is hook {}
      Aquatory.ports is hook {}
      Aquatory.routes is hook {}
    }
    port Routes {
      Route is anchor {
        port expr = $self.description$
      }
    }
    port Places {
      Port is anchor {
        port expr = $self.location.concat('Port')$
      }
    }
  }

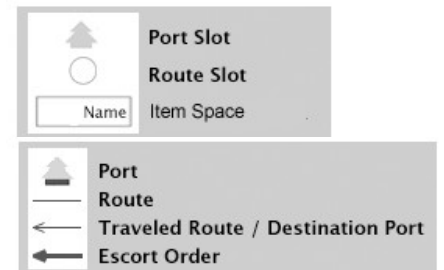
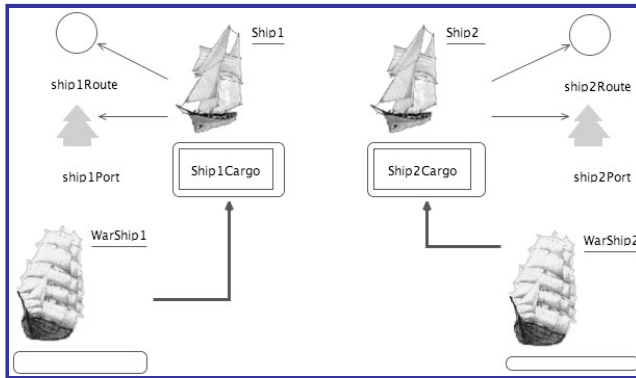
  fragment role Flotilla {
    port Vehicles {
      Aquatory.ships is prototype {}
      Aquatory.ports is prototype {}
      Aquatory.routes is prototype {}
    }
    port RouteSlots {
      RouteSlot is slot {
        port expr = $self.name$
      }
    }
    port PlaceSlots {
      PortSlot is slot {
        port expr = $self.name$
      }
    }
  }
}
...
}
```

The references **ships**, **ports** and **routes** of the metaclass **Aquatory** all act as hooks accessible through the **VehicleContainer** port

Building a DSL: Binding ReuseTaipan to Taipan DSL



VehicleContainer



Building a DSL: Binding ReuseTaipan to Taipan DSL

```

reuseextension reuseTaipan implements reuseTaipan
epackages <http://www.eclipse.org/examples/gmf/taipan>
Rootclass TravelSpace {
  fragment role TravelSpace {
    port VehicleContainer {
      Aquatory.ships is hook {}
      Aquatory.ports is hook {}
      Aquatory.routes is hook {}
    }
    port Routes {
      Route is anchor {
        port expr = $self.description$
      }
    }
    port Places {
      Port is anchor {
        port expr = $self.location.concat('Port')$
      }
    }
  }

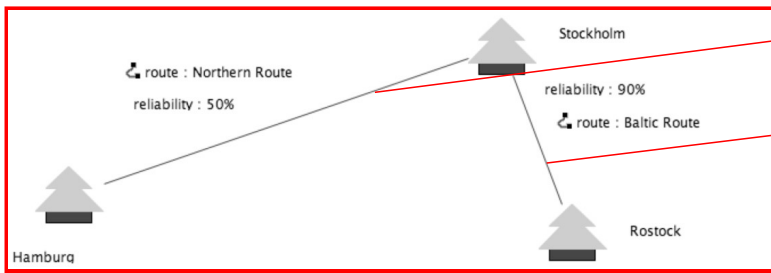
  fragment role Flotilla {
    port Vehicles {
      Aquatory.ships is prototype {}
      Aquatory.ports is prototype {}
      Aquatory.routes is prototype {}
    }
    port RouteSlots {
      RouteSlot is slot {
        port expr = $self.name$
      }
    }
    port PlaceSlots {
      PortSlot is slot {
        port expr = $self.name$
      }
    }
  }
  ...
}

```

Each Route is an anchor accessible through individual ports; the ports are named using the description attribute of the Route metaclass (OCL Expression: self.description)

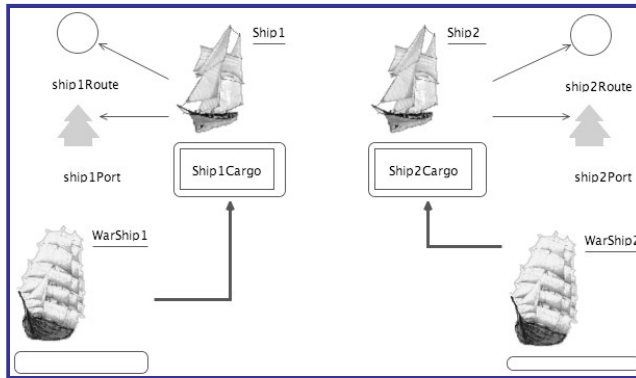


Building a DSL: Binding ReuseTaipan to Taipan DSL



Northern Route

Baltic Route



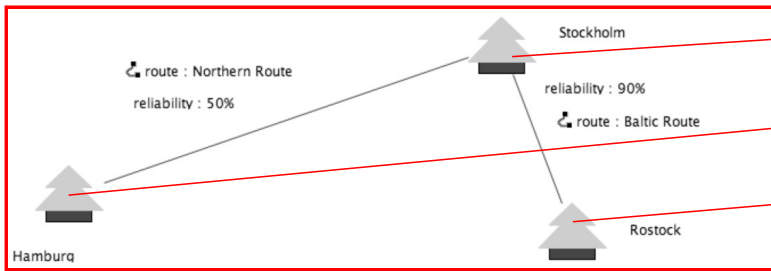
Building a DSL: Binding ReuseTaipan to Taipan DSL

```
reuseextension reuseTaipan implements reuseTaipan
epackages <http://www.eclipse.org/examples/gmf/taipan>
Rootclass TravelSpace {
  fragment role TravelSpace {
    port VehicleContainer {
      Aquatory.ships is hook {}
      Aquatory.ports is hook {}
      Aquatory.routes is hook {}
    }
    port Routes {
      Route is anchor {
        port expr = $self.description$
      }
    }
    port Places {
      Port is anchor {
        port expr = $self.location.concat('Port')$
      }
    }
  }
}

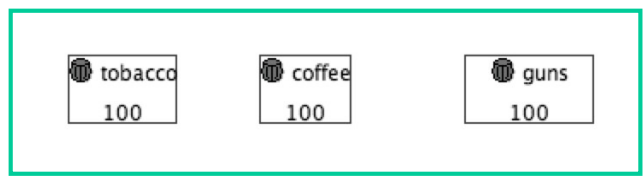
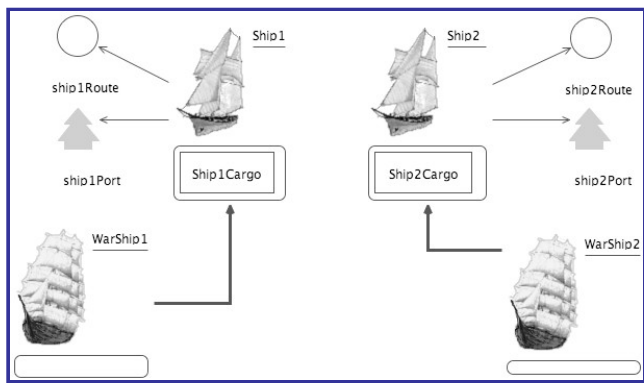
fragment role Flotilla {
  port Vehicles {
    Aquatory.ships is prototype {}
    Aquatory.ports is prototype {}
    Aquatory.routes is prototype {}
  }
  port RouteSlots {
    RouteSlot is slot {
      port expr = $self.name$
    }
  }
  port PlaceSlots {
    PortSlot is slot {
      port expr = $self.name$
    }
  }
}
...
}
```

Each Port is an anchor accessible through individual ports; the ports are named using the location attribute of the Port metaclass

Building a DSL: Binding ReuseTaipan to Taipan DSL



- StockholmPort
- HamburgPort
- RostockPort



	Port Slot
	Route Slot
<input type="text"/>	Name
<input type="text"/>	Item Space
	Port
	Route
	Traveled Route / Destination Port
	Escort Order



Building a DSL: Binding ReuseTaipan to Taipan DSL

```

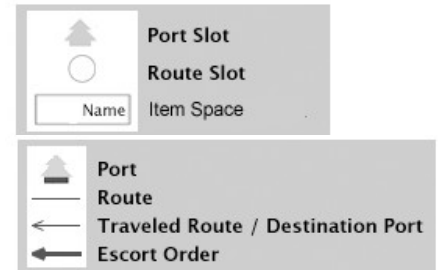
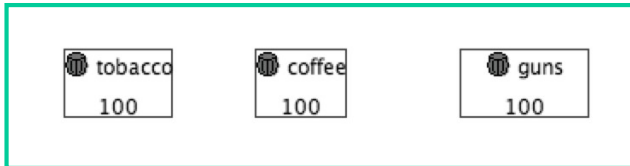
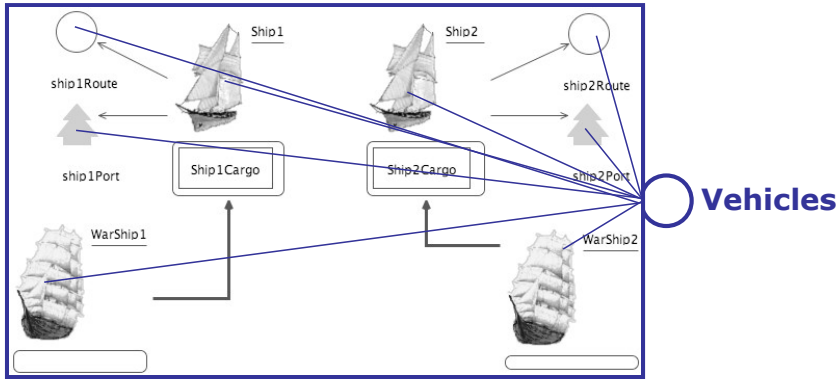
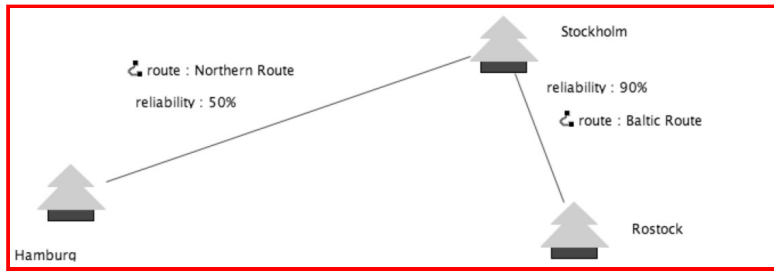
reuseextension reuseTaipan implements reuseTaipan
epackages <http://www.eclipse.org/examples/gmf/taipan>
Rootclass TravelSpace {
  fragment role TravelSpace {
    port VehicleContainer {
      Aquatory.ships is hook {}
      Aquatory.ports is hook {}
      Aquatory.routes is hook {}
    }
    port Routes {
      Route is anchor {
        port expr = $self.description$
      }
    }
    port Places {
      Port is anchor {
        port expr = $self.location.concat('Port')$
      }
    }
  }

  fragment role Flotilla {
    port Vehicles {
      Aquatory.ships is prototype {}
      Aquatory.ports is prototype {}
      Aquatory.routes is prototype {}
    }
    port RouteSlots {
      RouteSlot is slot {
        port expr = $self.name$
      }
    }
    port PlaceSlots {
      PortSlot is slot {
        port expr = $self.name$
      }
    }
  }
}

```

All elements of the references ships, ports and routes of the metaclass Aquatory act as prototypes accessible through the Vehicles port

Building a DSL: Binding ReuseTaipan to Taipan DSL



Prof. U. Aßmann, SEW

Building a DSL: Binding ReuseTaipan to Taipan DSL

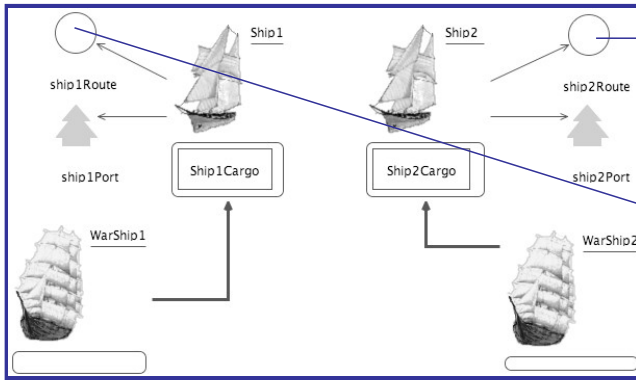
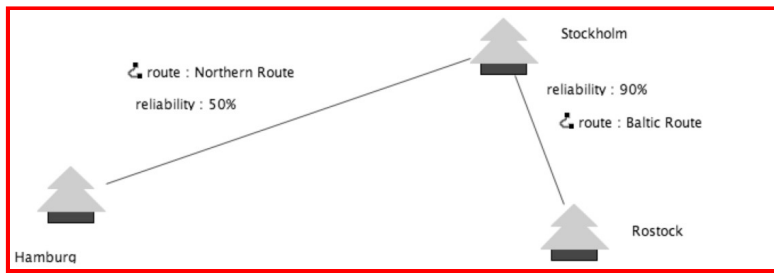
```
reuseextension reuseTaipan implements reuseTaipan
epackages <http://www.eclipse.org/examples/gmf/taipan>
Rootclass TravelSpace {
  fragment role TravelSpace {
    port VehicleContainer {
      Aquatory.ships is hook {}
      Aquatory.ports is hook {}
      Aquatory.routes is hook {}
    }
    port Routes {
      Route is anchor {
        port expr = $self.description$
      }
    }
    port Places {
      Port is anchor {
        port expr = $self.location.concat('Port')$
      }
    }
  }

  fragment role Flotilla {
    port Vehicles {
      Aquatory.ships is prototype {}
      Aquatory.ports is prototype {}
      Aquatory.routes is prototype {}
    }
    port RouteSlots {
      RouteSlot is slot {
        port expr = $self.name$
      }
    }
    port PlaceSlots {
      PortSlot is slot {
        port expr = $self.name$
      }
    }
  }
}
```

Each RouteSlot is a slot accessible through individual ports; the ports are named using the name attribute of the RouteSlot metaclass

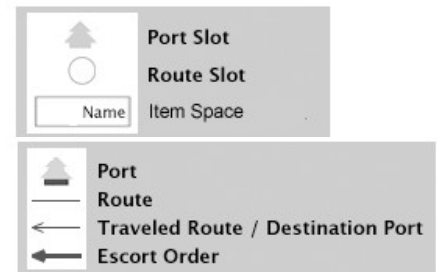
Prof. U. Aßmann, SEW

Building a DSL: Binding ReuseTaipan to Taipan DSL



ship2Route

ship1Route



Building a DSL: Binding ReuseTaipan to Taipan DSL

```

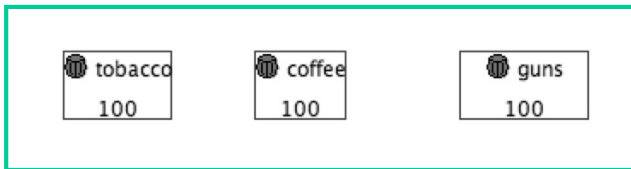
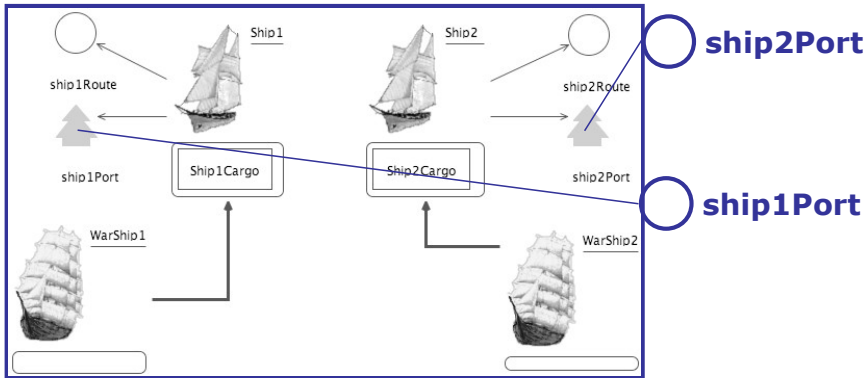
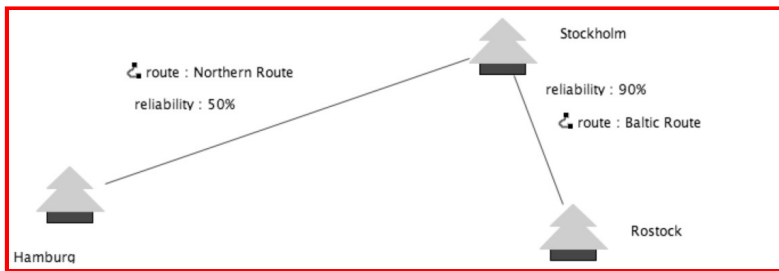
reuseextension reuseTaipan implements reuseTaipan
epackages <http://www.eclipse.org/examples/gmf/taipan>
Rootclass TravelSpace {
  fragment role TravelSpace {
    port VehicleContainer {
      Aquatory.ships is hook {}
      Aquatory.ports is hook {}
      Aquatory.routes is hook {}
    }
    port Routes {
      Route is anchor {
        port expr = $self.description$
      }
    }
    port Places {
      Port is anchor {
        port expr = $self.location.concat('Port')$
      }
    }
  }

  fragment role Flotilla {
    port Vehicles {
      Aquatory.ships is prototype {}
      Aquatory.ports is prototype {}
      Aquatory.routes is prototype {}
    }
    port RouteSlots {
      RouteSlot is slot {
        port expr = $self.name$
      }
    }
    port PlaceSlots {
      PortSlot is slot {
        port expr = $self.name$
      }
    }
  }
}

```

Each PortSlot is a slot accessible through individual ports; the ports are named using the name attribute of the RouteSlot metaclass

Building a DSL: Binding ReuseTaipan to Taipan DSL



Prof. U. Aßmann, SEW

35

Building a DSL: Binding ReuseTaipan to Taipan DSL

```

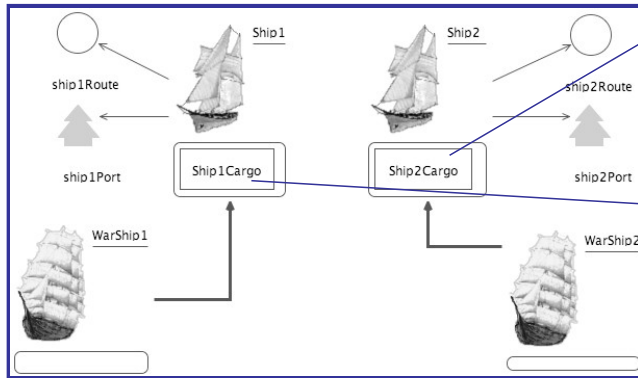
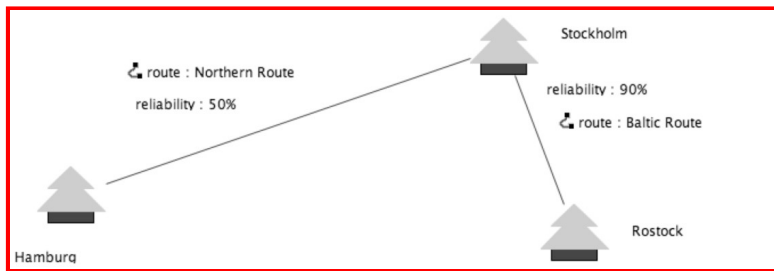
...
binding ItemHolder {
  binding ItemSpaces {
    ItemSpace is hook {
      port expr = $self.name$
    }
  }
}

binding ItemContainer {
  binding Items {
    Item is prototype {
      port expr = $self.article$
    }
  }
}
}

```

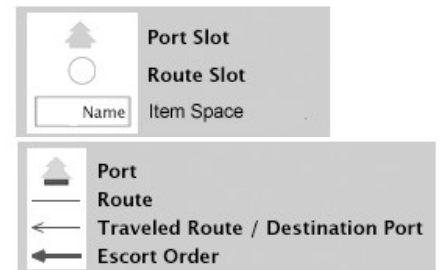
Each **ItemSpace** is a hook accessible through individual ports; the ports are named using the **name** attribute of the **ItemSpace** metaclass

Building a DSL: Binding ReuseTaipan to Taipan DSL



Ship2Cargo

Ship1Cargo



Prof. U. Alßmann, SEW

37

Building a DSL: Binding ReuseTaipan to Taipan DSL

```

...
fragment role ItemHolder {
  port ItemSpaces {
    ItemSpace is hook {
      port expr = $self.name$
    }
  }
}

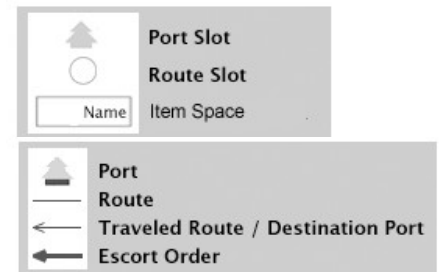
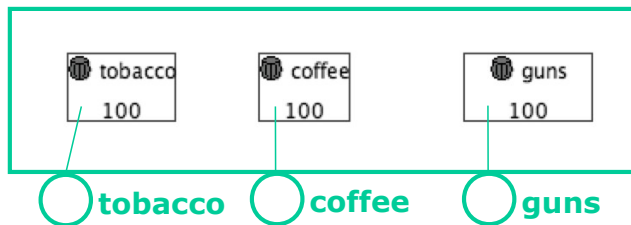
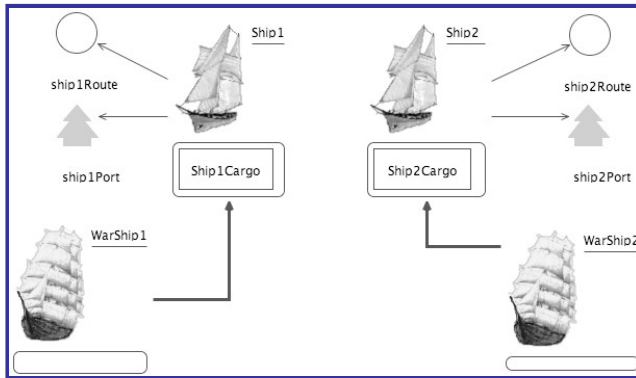
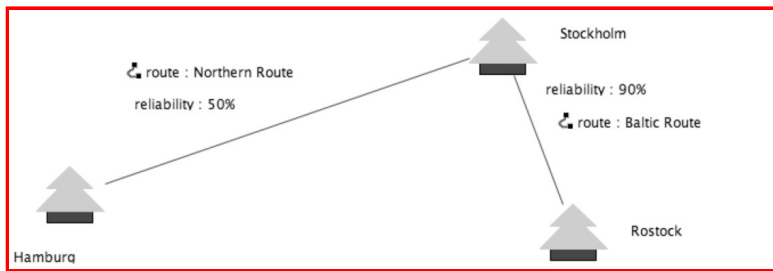
fragment role ItemContainer {
  port Items {
    Item is prototype {
      port expr = $self.article$
    }
  }
}

```

Each **Item** is a prototype accessible through individual ports; the ports are named using the **article** attribute of the **Items** metaclass



Building a DSL: Binding ReuseTaipan to Taipan DSL



Prof. U. Aßmann, SEW

39

17.4 Using Reuseware Tooling with a DSL

- ▶ Fragment Repository
 - Light-weight repository to manage and find reusable model fragments
 - Can instantly be used to build libraries of model fragments designed in a DSL
- ▶ Composition Program Editor
 - Independent of composition systems and reuse extensions
 - Can instantly be used to define compositions for the DSL
 - Layout can be customized if desired

Building a DSL: Using Reuseware Tooling with a DSL

The screenshot shows the Eclipse IDE with a DSL model diagram. The diagram consists of three fragments: **EuropeanSea.taipan**, **MyFlotilla.taipan**, and **MyCargo.taipan2**. **EuropeanSea.taipan** is connected to **MyFlotilla.taipan** via several lines, and **MyFlotilla.taipan** is connected to **MyCargo.taipan2** via three lines. The left sidebar shows a project tree with various reusable components like `reuseTaipan.Flotilla`, `reuseTaipan.ItemContainer`, and `reuseTaipan.TravelSpace`. The bottom panel shows the **Properties** view for the **Fragment Instance MyFlotilla.taipan**.

Property	Value
Core	
Composition	
Cs Fragment Roles	reuseTaipan.Flotilla, reuseTaipan.ItemHolder
Name	MyFlotilla.taipan

41

Building a DSL: Using Reuseware Tooling with a DSL

This screenshot is similar to the previous one, but with a red box highlighting the project tree on the left and a yellow callout box over the diagram. The callout box contains the following text:

The fragment repository shows model fragments, the fragment roles they can play and the details of the corresponding composition interfaces

42

Building a DSL: Using Reuseware Tooling with a DSL

Java - taipan.taosd/taipan_example/models/TravelPlan.fcdi - Eclipse Platform

Lucida Grande

9 B I A

100%

Fragment Instance MyFlotilla.taipan

Property	Value
Composition	
Cs Fragment Roles	reuseTaipan.Flottilla, reuseTaipan.ItemHolder
Name	MyFlotilla.taipan

43

Building a DSL: Using Reuseware Tooling with a DSL

Java - taipan.taosd/taipan_example/models/TravelPlan.fcdi - Eclipse Platform

Lucida Grande

9 B I A

100%

Fragment Instance MyFlotilla.taipan

Property	Value
Composition	
Cs Fragment Roles	reuseTaipan.Flottilla, reuseTaipan.ItemHolder
Name	MyFlotilla.taipan

44

Building a DSL: Using Reuseware Tooling with a DSL

Java - taipan.taosd/taipan_example/models_reference/TravelPlan.taipan_diagram - Eclipse Platform

Tahoma

TravelPlan.fcdi TravelPlan.taipan_diagram

EuropeanSea.taipan

- reuseTaipan.Flotilla
- reuseTaipan.ItemContainer
- reuseTaipan.ItemHolder
- reuseTaipan.TravelSpace
- BalticRoute (Routes)
- HamburgPort (Places)
- NorthernRoute (Routes)
- RostockPort (Places)
- StockholmPort (Places)
- VehicleContainer

MyCargo.taipan

- reuseTaipan.Flotilla
- reuseTaipan.ItemContainer
- coffee (Items)
- guns (Items)
- tobacco (Items)
- reuseTaipan.ItemHolder
- reuseTaipan.TravelSpace

MyFlotilla.taipan

- reuseTaipan.Flotilla
- ship1Port (PlaceSlots)
- ship1Route (RouteSlots)
- ship2Port (PlaceSlots)
- ship2Route (RouteSlots)
- Vehicles
- reuseTaipan.ItemContainer
- reuseTaipan.ItemHolder
- reuseTaipan.TravelSpace

Stockholm

reliability : 80%

route : BalticRoute

Hamburg

route : NorthernRoute

reliability : 80%

Rostock

Ship1

tobacco

Ship2

coffee

WarShip1

WarShip2

Aquatory

Property	Value
Domain Model	
Rulers & Grid	

- 45 -

Prof. U. Aßmann, SEW

45

The End

- 46 -

Prof. U. Aßmann, SEW

46