**Domain Context for the Hepsy Metamodel**

**Domain Overview**

The Hepsy metamodel defines a structured framework for modeling behavior specifications in a system, focusing on **nodes, channels, ports, and processes**. It is suitable for domains such as:

* Hardware modeling
* System behavior simulation
* Message passing in distributed systems

**Namespace URI:** org.univaq.hepsy  
**Package:** hepsy

**Key Concepts**

**📦 Main Classes**

**1. BehaviorSpecification**

* Supertype: NamedElement
* References:
  + nodes (Type: Node, Multiplicity: many, Containment: true)

**2. Node** *(Abstract)*

* Supertype: NamedElement
* References:
  + nChannels (Channel, many, Containment: true)
  + ports (Port, many, Containment: true)

**3. Channel**

* Supertype: NamedElement
* Attributes:
  + queueSize: EInt
  + timeout: EInt
  + rendezVous: EBoolean
  + direction: EString
* References:
  + nFrom, nTo: Node
  + pFrom, pTo: Port
  + message: Message (Containment: true)

**4. NamedElement**

* Attributes:
  + name: EString

**5. StructuredNode**

* Supertype: Node
* References:
  + processes: Process (many, Containment: true)

**6. Port**

* Supertype: NamedElement
* References:
  + pChannels: Channel (many, Containment: true)
  + portExtension: Process

**7. Process**

* Supertype: Node
* Attributes:
  + priority: EInt (Default: 1)
  + criticality: EInt
* References:
  + processExtension: Port

**8. Display / 9. Stimulus**

* Supertype: Node
* Attributes: None
* References: None

**10. Message**

* Supertype: NamedElement
* References:
  + entry: Entry (many, Containment: true)

**11. Entry**

* Supertype: NamedElement
* Attributes:
  + type: EString (Required)

**🧭 Enumerations**

**DataType:**  
sc\_bit, sc\_logic, sc\_int, sc\_uint, sc\_bigint, sc\_biguint, sc\_bv, sc\_lv, sc\_fixed, sc\_ufixed, sc\_fix, sc\_ufix

**Direction:**  
Unidirectional, Bidirectional

**Relationships**

**Containment:**

* BehaviorSpecification contains Nodes
* Node contains Channels and Ports
* StructuredNode contains Processes
* Channel contains a Message
* Message contains Entries
* Port contains Channels

**Association:**

* Channel is associated with Node via nFrom, nTo
* Channel is associated with Port via pFrom, pTo
* Port is associated with Process via portExtension
* Process is associated with Port via processExtension

**Behavioral Semantics**

* **BehaviorSpecification** acts as the root container of nodes
* **Nodes** can have multiple channels and ports for communication
* **Channels** define communication paths with behavior attributes
* **Ports** are node interfaces to connect to channels and processes
* **Processes** are computational units with priority and criticality
* **Messages** are data structures within channels, composed of Entries

**Domain Applications**

* **Hardware Modeling** – components, buses, interfaces, processing units
* **System Behavior Simulation** – component interaction, message flow
* **Distributed Systems** – channel-based communication, message structures

**Terminology**

* **Node:** Fundamental system unit
* **Channel:** Communication pathway
* **Port:** Node interface
* **Process:** Computational entity
* **Message:** Transmitted data structure
* **Entry:** Typed element within a message

**Metadata**

* **Package Name:** hepsy
* **Namespace URI:** org.univaq.hepsy
* **Namespace Prefix:** hml