**Domain Context for the Hepsy Ecore Metamodel**

**Domain Overview**

This Ecore metamodel is designed to model **system behaviors**, especially in **hardware or embedded systems**. It defines and manages interactions between components like **nodes**, **channels**, **ports**, and **processes**.

The **package name** hepsy and **namespace URI** org.univaq.hepsy indicate a focus on **hierarchical or heterogeneous system specifications**, supporting tasks like **simulation** and **analysis**.

**Key Concepts**

**📦 Classes**

**1. BehaviorSpecification**

* Inherits from: NamedElement
* Role: Root container encapsulating overall system behavior
* Contains: Multiple Node instances

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

* Represents a basic system component
* Can contain: Channel and Port instances
* Facilitates communication

**3. Channel**

* Represents a communication link
* Attributes:
  + queueSize
  + timeout
  + rendezVous (synchronization)
  + direction (message flow)
* Connects: Nodes and Ports

**4. Port**

* Interface for nodes to connect to channels
* Can extend or reference Process

**5. Process**

* Specialized Node
* Attributes: priority, criticality
* Executes system tasks

**6. StructuredNode**

* Composite Node that contains Process instances
* Organizes processes hierarchically

**7. Display / 8. Stimulus**

* Specialized Nodes
  + Likely represent output and input elements respectively

**9. Message**

* Communication unit
* Contains multiple Entry instances

**10. Entry**

* Part of a message
* Attribute: type (required) – defines the role or data type

**🧭 Enumerations**

**DataType**  
Includes types like:  
sc\_bit, sc\_logic, sc\_int, sc\_uint, sc\_bigint, sc\_biguint, sc\_bv, sc\_lv, sc\_fixed, sc\_ufixed, sc\_fix, sc\_ufix  
→ Resemble **SystemC** data types

**Direction**

* Unidirectional
* Bidirectional

**Relationships**

**Containment:**

* BehaviorSpecification → contains Node
* Node → contains Channel, Port
* StructuredNode → contains Process
* Channel → contains Message
* Message → contains Entry

**References:**

* Channel → references Node and Port via nFrom, nTo, pFrom, pTo
* Port → references Channel and Process
* Process → references Port

**Behavioral Semantics**

* Nodes interact through **channels**
* **Ports** serve as connection interfaces
* **Channels** control communication with attributes like queueSize, timeout, and rendezVous
* **Processes** are schedulable units with **priority** and **criticality**, aiding in **task execution** and **resource management**

**External Domain Background**

This metamodel parallels modeling approaches found in:

* **Hardware Description Languages (HDLs)** like **SystemC** or **VHDL**
* **System-Level Design Languages (SLDLs)**

Concepts like **nodes, channels, ports** are fundamental in:

* **Network-on-Chip (NoC)**
* **System-on-Chip (SoC)**  
  → Where communication and computation are tightly integrated

**Tool Context & Suggested Name**

With the package hepsy and namespace URI org.univaq.hepsy, suitable tool names could be:

* **HepsySim**
* **HepsyModeler**

Such tools would focus on **simulation** and **modeling** of **hierarchical or heterogeneous systems**, emphasizing **communication** and **process interaction** in **hardware/embedded** contexts.