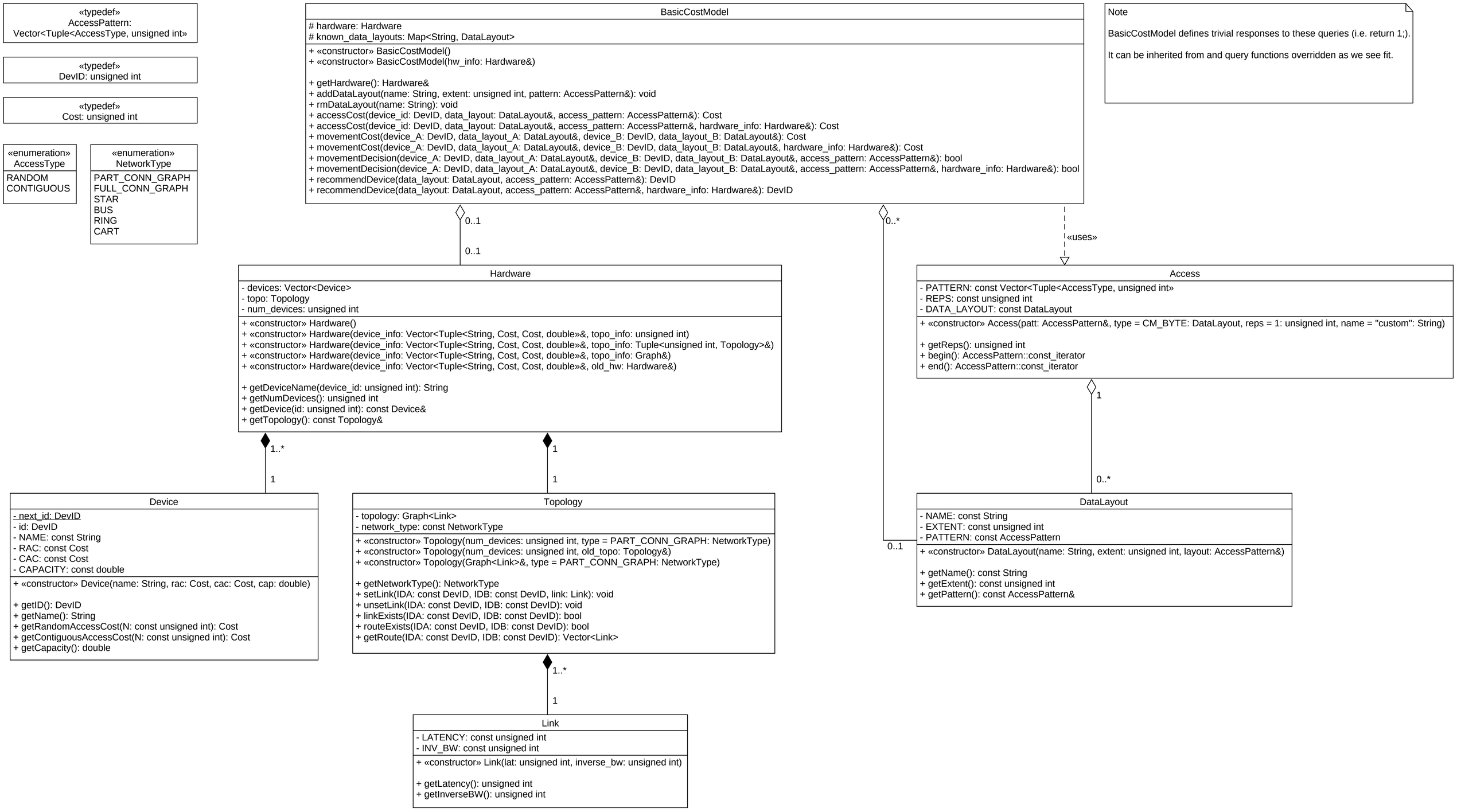
«typedef»
AccessPattern:
Vector<Tuple<AccessType, unsigned int»
«typedef»
DevID: unsigned int
«typedef»
Cost: unsigned int
«enumeration»
AccessType
RANDOM
CONTIGUOUS
«enumeration»
NetworkType
PART_CONN_GRAPH
FULL_CONN_GRAPH
STAR
BUS
RING
CART
BasicCostModel
hardware: Hardware
known_data_layouts: Map<String, DataLayout>
+ «constructor» BasicCostModel()
+ «constructor» BasicCostModel(hw_info: Hardware&)
+ getHardware(): Hardware&
+ addDataLayout(name: String, extent: unsigned int, pattern: AccessPattern&): void
+ rmDataLayout(name: String): void
+ accessCost(device_id: DevID, data_layout: DataLayout&, access_pattern: AccessPattern&): Cost
+ accessCost(device_id: DevID, data_layout: DataLayout&, access_pattern: AccessPattern&, hardware_info: Hardware&): Cost
+ movementCost(device_A: DevID, data_layout_A: DataLayout&, device_B: DevID, data_layout_B: DataLayout&): Cost
+ movementCost(device_A: DevID, data_layout_A: DataLayout&, device_B: DevID, data_layout_B: DataLayout&, hardware_info: Hardware&): Cost
+ movementDecision(device_A: DevID, data_layout_A: DataLayout&, device_B: DevID, data_layout_B: DataLayout&, access_pattern: AccessPattern&): bool
+ movementDecision(device_A: DevID, data_layout_A: DataLayout&, device_B: DevID, data_layout_B: DataLayout&, access_pattern: AccessPattern&, hardware_info: Hardware&): bool
+ recommendDevice(data_layout: DataLayout, access_pattern: AccessPattern&): DevID
+ recommendDevice(data_layout: DataLayout, access_pattern: AccessPattern&, hardware_info: Hardware&): DevID
Note
BasicCostModel defines trivial responses to these queries (i.e. return 1;).
It can be inherited from and query functions overridden as we see fit.
0..1
0..1
0..*
«uses»
Hardware
- devices: Vector<Device>
- topo: Topology
- num_devices: unsigned int
+ «constructor» Hardware()
+ «constructor» Hardware(device_info: Vector<Tuple<String, Cost, Cost, double»&, topo_info: unsigned int)
+ «constructor» Hardware(device_info: Vector<Tuple<String, Cost, Cost, double»&, topo_info: Tuple<unsigned int, Topology>&)
+ «constructor» Hardware(device_info: Vector<Tuple<String, Cost, Cost, double»&, topo_info: Graph&)
+ «constructor» Hardware(device_info: Vector<Tuple<String, Cost, Cost, double»&, old_hw: Hardware&)
+ getDeviceName(device_id: unsigned int): String
+ getNumDevices(): unsigned int
+ getDevice(id: unsigned int): const Device&
+ getTopology(): const Topology&
Access
- PATTERN: const Vector<Tuple<AccessType, unsigned int»
- REPS: const unsigned int
- DATA_LAYOUT: const DataLayout
+ «constructor» Access(patt: AccessPattern&, type = CM_BYTE: DataLayout, reps = 1: unsigned int, name = "custom": String)
+ getReps(): unsigned int
+ begin(): AccessPattern::const_iterator
+ end(): AccessPattern::const_iterator
1..*
1
1
1
1
0..*
Device
- next_id: DevID
- id: DevID
- NAME: const String
- RAC: const Cost
- CAC: const Cost
- CAPACITY: const double
+ «constructor» Device(name: String, rac: Cost, cac: Cost, cap: double)
+ getID(): DevID
+ getName(): String
+ getRandomAccessCost(N: const unsigned int): Cost
+ getContiguousAccessCost(N: const unsigned int): Cost
+ getCapacity(): double
Topology
- topology: Graph<Link>
- network_type: const NetworkType
+ «constructor» Topology(num_devices: unsigned int, type = PART_CONN_GRAPH: NetworkType)
+ «constructor» Topology(num_devices: unsigned int, old_topo: Topology&)
+ «constructor» Topology(Graph<Link>&, type = PART_CONN_GRAPH: NetworkType)
+ getNetworkType(): NetworkType
+ setLink(IDA: const DevID, IDB: const DevID, link: Link): void
+ unsetLink(IDA: const DevID, IDB: const DevID): void
+ linkExists(IDA: const DevID, IDB: const DevID): bool
+ routeExists(IDA: const DevID, IDB: const DevID): bool
+ getRoute(IDA: const DevID, IDB: const DevID): Vector<Link>
DataLayout
- NAME: const String
- EXTENT: const unsigned int
- PATTERN: const AccessPattern
+ «constructor» DataLayout(name: String, extent: unsigned int, layout: AccessPattern&)
+ getName(): const String
+ getExtent(): const unsigned int
+ getPattern(): const AccessPattern&
0..1
1..*
1
Link
- LATENCY: const unsigned int
- INV_BW: const unsigned int
+ «constructor» Link(lat: unsigned int, inverse_bw: unsigned int)
+ getLatency(): unsigned int
+ getInverseBW(): unsigned int