Review of the Readout schema

Alessandro, Giovanna

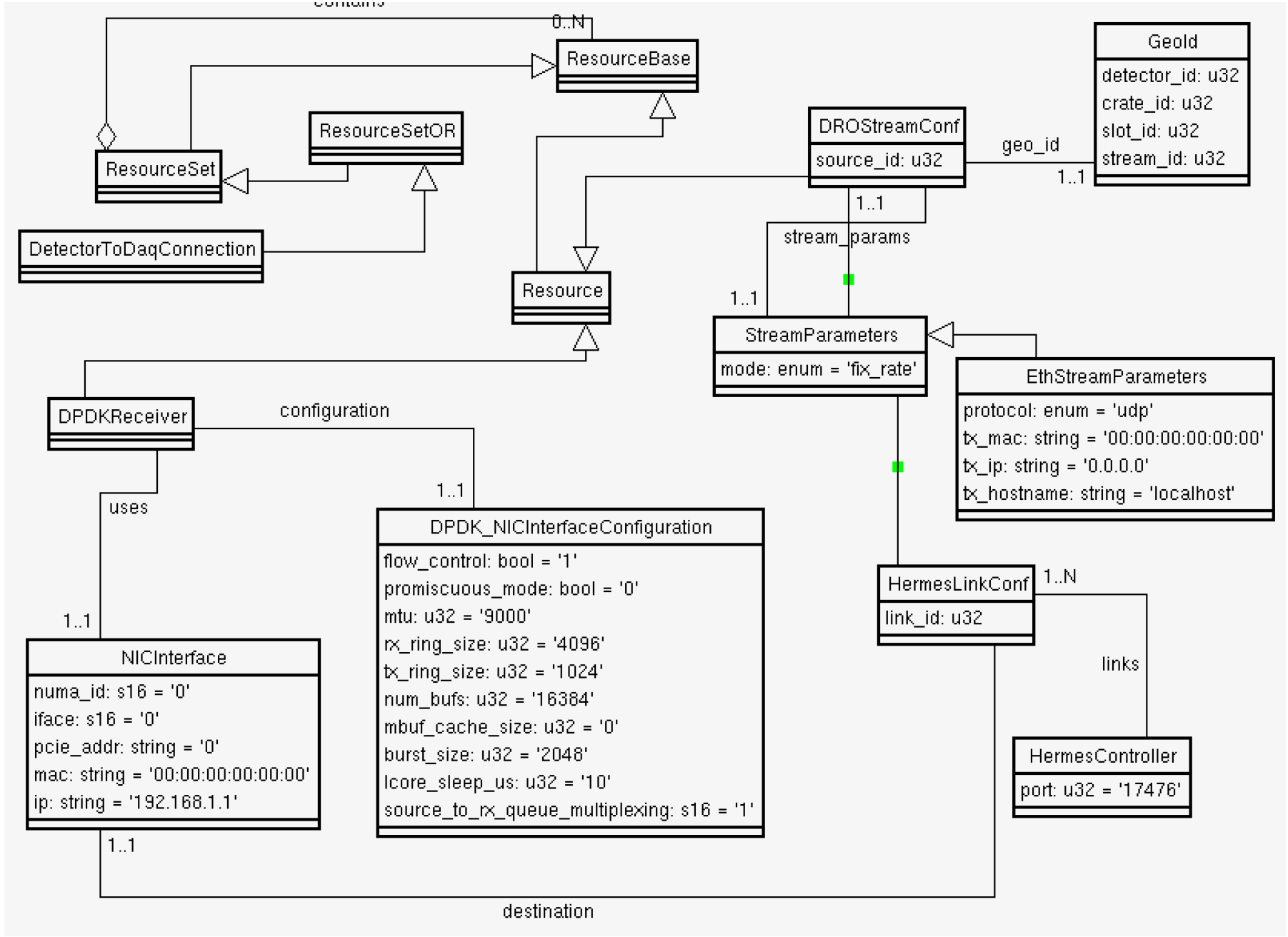
Observations on first version of OKS schema

- Unclean separation of detector streams and receiving endpoint (streams contained in the NICInterface class, streams containing lcore/rx_queue information which are receiver properties….)
- Mix of hardware properties and configuration settings (MAC address, dpdk parameters, …)

We decided to revise the schema “at the whiteboard” to improve modularity and maintainability.

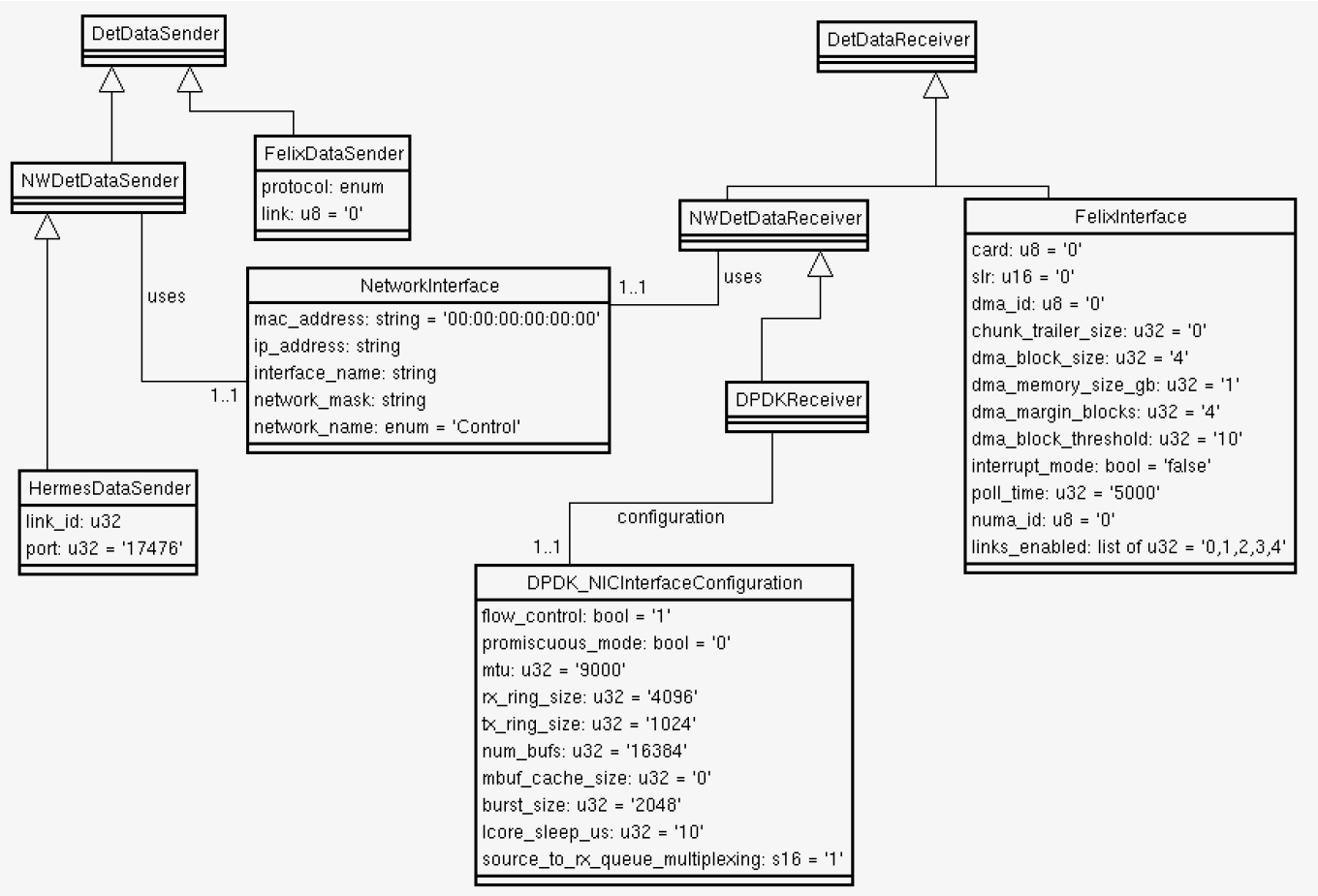
Changes on the readout map

- Introduced a DetectorToDaqConnection class which contains
  - A group of detector streams (only sender properties)
  - One DPDKReceiver
- If one wants to change the receiver for one detector unit (e.g. one APA) it is sufficient to change the DPDKReceiver object used in the connection. 🙂
- What is a DPDKReceiver?
  - It's a class linking a Hardware NICInterface with a set of chosen DPDK specific configuration settings (class DPDK_NICInterfaceConfiguration)
- Why is this a good approach?
  - For the same NICInterface hardware we may want to test different DPDK configurations without affecting the hardware properties that are described "once for good".
  - Several DPDKReceivers may use the same DPDK configuration, running on different NICInterfaces


contains
0..N
ResourceBase
ResourceSet
ResourceSetOR
DetectorToDaqConnection
Resource
DROStreamConf
source_id: u32
geo_id
1..1
GeoId
detector_id: u32
crate_id: u32
slot_id: u32
stream_id: u32
1..1
stream_params
1..1
StreamParameters
mode: enum = 'fix_rate'
EthStreamParameters
protocol: enum = 'udp'
tx_mac: string = '00:00:00:00:00:00'
tx_ip: string = '0.0.0.0'
tx_hostname: string = 'localhost'
DPDKReceiver
configuration
1..1
uses
1..1
DPDK_NICInterfaceConfiguration
flow_control: bool = '1'
promiscuous_mode: bool = '0'
mtu: u32 = '9000'
rx_ring_size: u32 = '4096'
tx_ring_size: u32 = '1024'
num_bufs: u32 = '16384'
mbuf_cache_size: u32 = '0'
burst_size: u32 = '2048'
lcore_sleep_us: u32 = '10'
source_to_rx_queue_multiplexing: s16 = '1'
HermesLinkConf
link_id: u32
1..N
links
HermesController
port: u32 = '17476'
NICInterface
numa_id: s16 = '0'
iface: s16 = '0'
pcie_addr: string = '0'
mac: string = '00:00:00:00:00:00'
ip: string = '192.168.1.1'
1..1
destination

Update since last week

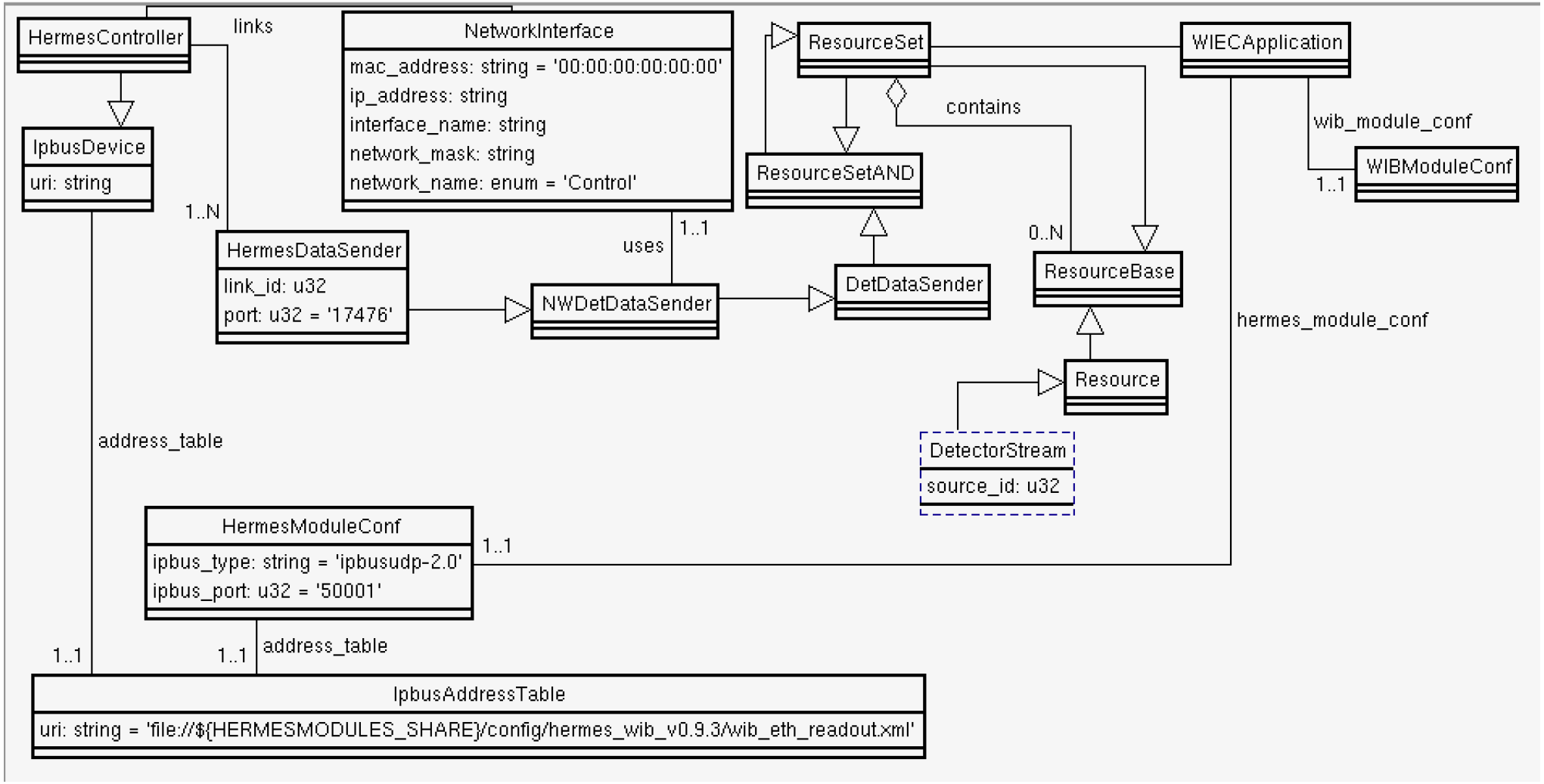
- Abstracted and generalised the concept of DetDataSender and DetDataReceiver
- A DetToDaqConnection typically contains one DetDataReceiver and a set of DetDataSenders
- Adapted for FELIX and should be on track for ND



Changes to the ReadoutApplication

- Instead of pointing to the old NICInterface(s) the ReadoutApplication now points to one or more DetectorToDaqConnection(s)
- The generate_modules() method for the ReadoutApplication needs to be updated accordingly
- Other algorithms (for other smart apps) need minor adjustments.
- The Hermes schema was updated to used the smart application concept and have a clean split of hardware/configuration properties

Hermes Schema



Status and next steps

- New schema in coredal and appdal, glm/readout branch
  - Need to update example confs still
- Adjust daq modules code that is affected by this change (should be very minor modifications; I can do *readout*, someone needed for hermes, dfmodules, maybe trigger?)
- Decide today on date to merge all branches into nightly
  - Prepare a wiki with the list of packages & branches to work on (who?)
  - Assign all affected packages to developers