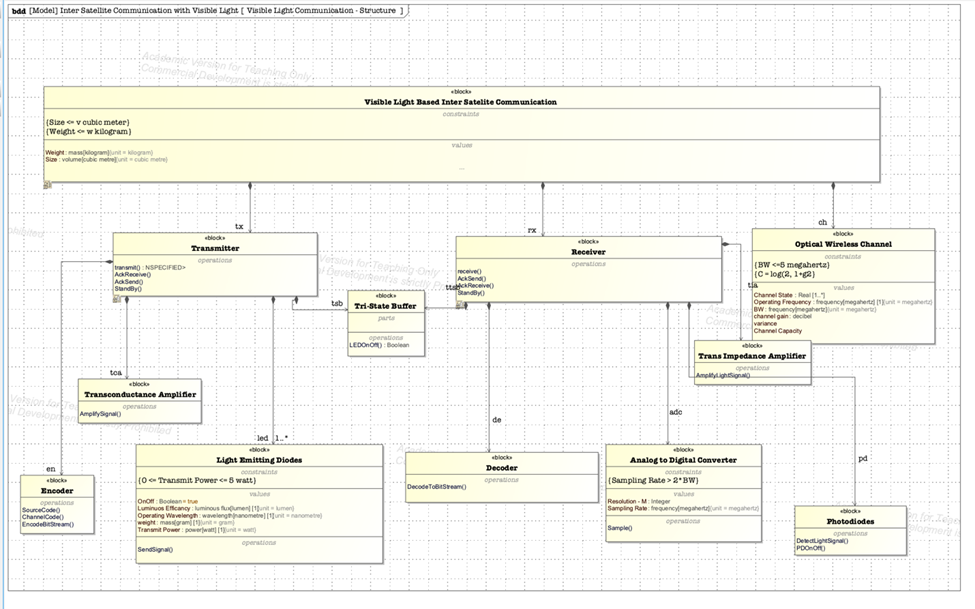
BDD



Missing Items for the IBD:

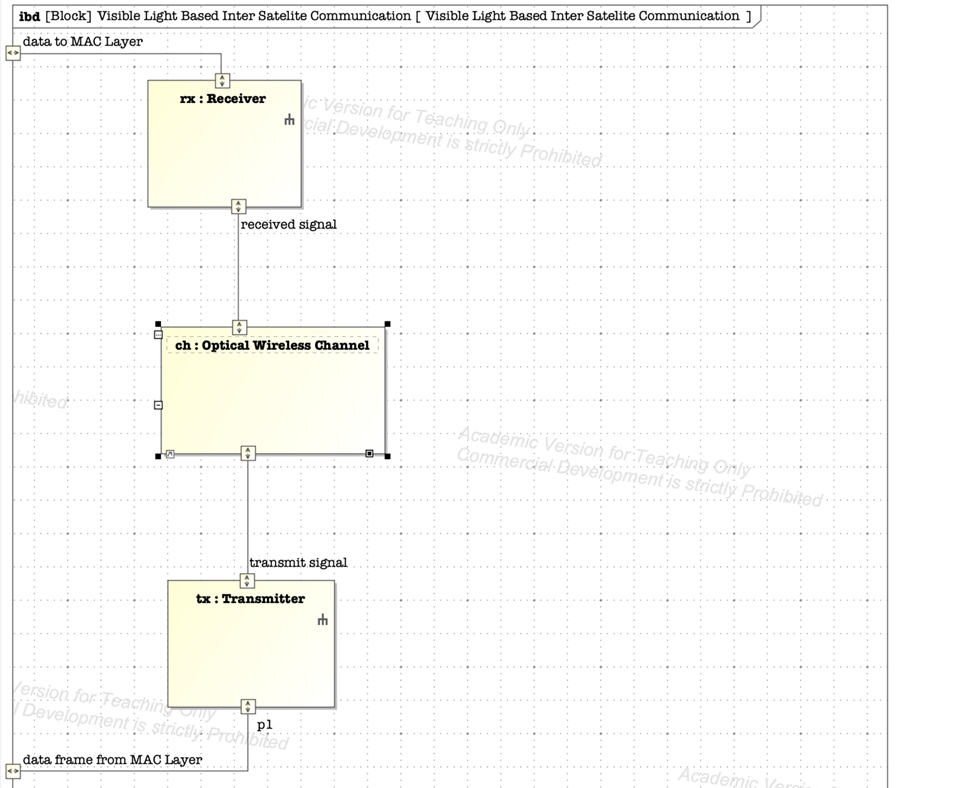
Interface Blocks / Flow Properties





The interfaces are defined using ports. The ports that are attached to each block in the IBD are representing the types of objects that flow in or out of the block. The labels are not for the connectors but for the ports. In addition, it is not required for the ports definition to appear in the block definition diagram. When the port is defined for a block in the IBD, it automatically becomes the port property of the block and if needed can be made visible in the BDD by turning “suppress ports” feature in the symbol properties of the block.

IBD



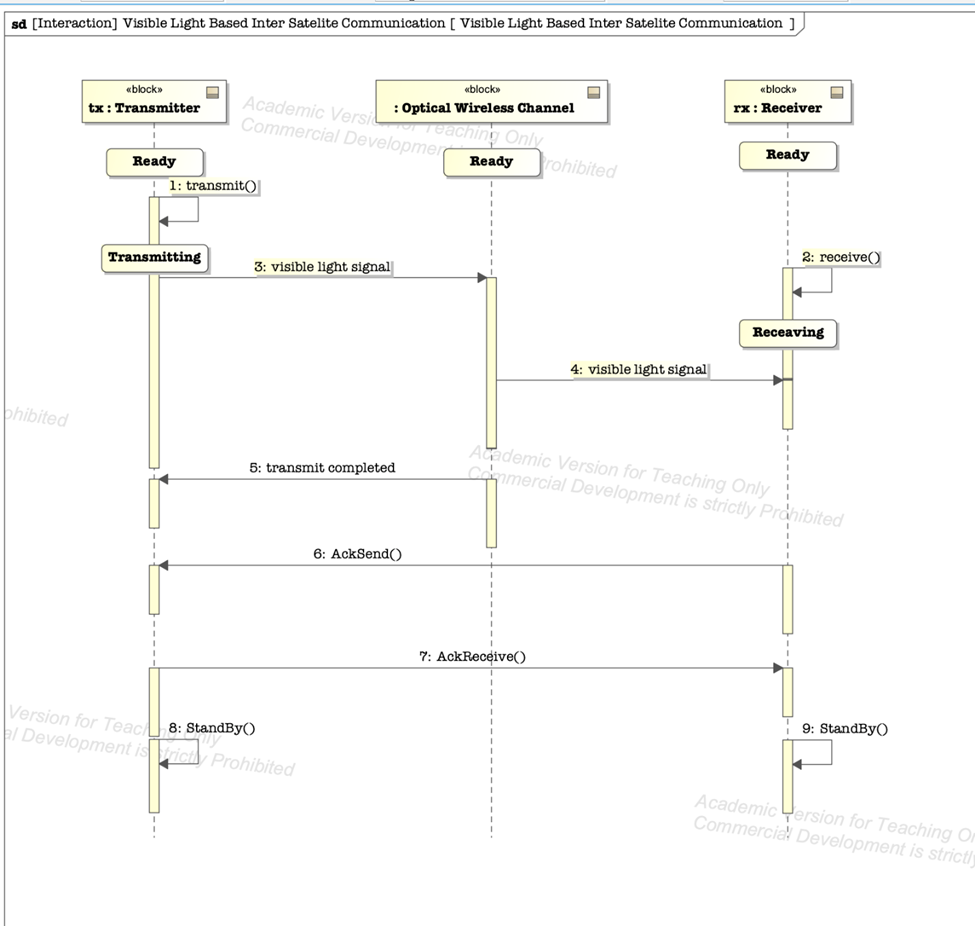
Incorrect:

The connector doesn’t have to have a name, but the interface block on both ends of the connector should be defined with the type. This corresponds to the interface block in the BDD.



The bidirectional ports are modified to correctly represent the direction of the flow for each IBD diagram. The names are not for the connectors. They are the names for the ports.

SD



Incorrect:

All messages must match the name of the operation that is invoked by the message:

3: visible light spectrum, 4: visible light spectrum and 5: transmit complete do not meet this requirement.

Updated based on the suggestion.

3: and 4: visible light spectrum should be defined in the BDD as the flow property for the connector between these physical elements.

Please see updated diagram.

Proper representation of message types and “Executions” (yellow vertical bars) on lifelines needs attentions.

An “execution” is usually activated by a synchronous message (as you did) and ended with a reply message, which is represented by a dashed line with open arrowhead. (Book [Practical Guide] 10.5.4 Executions, and book [Distilled] 7.5.2. A reply message can be implicit; it would be appreciated to make it explicit.

Synchronous message (the sender will wait for reply to it) -- solid line with solid close arrowhead.

Asynchronous message -- solid line with open arrowhead

Reply message -- dashed line with open arrowhead

Please see updated diagram.

Developing State Machine Diagram

We believe developing a state machine diagram might be a good idea at this point. Mapping a state machine diagram to functional reactive programs (FRP) might be easier than the other types of SysML structural or behavioral diagrams. The reason is that transitioning from state machine to functional programming seams simple and natural as state machines are defined mathematically as a set of states and a transition function and could be a better place to start the mapping process.