Universiteit Antwerpen

FTG+PM for the Model-Driven Development of Wireless Sensor Network based IoT Systems

Burak Karaduman, Sadaf Mustafiz, Moharram Challenger

ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS)

Universiteit Antwerpen

Context

- Wireless sensors can create a large mesh-network to cover wide areas.

- Node which is programmed to collect environmental data named *source node*.
- The node that receives all data and transmits them to a gateway is named *sink node*.
- Gateway transmits this data to Internet level by integrating IoT devices.

- A WSN system can become a major part of an IoT system.
- IoT devices are controlled according the data gathered from WSN.

Complexity

Wireless-Sensor Networks (WSN)

- Can be distributed into a large environment for collecting data via sensors

Internet of Things (IoT)

- Can be used to control various actuators (e.g. solenoid valve)

WSN+IoT

- Various operating systems
- Various components
- Various embedded boards

As a result:

The planned system becomes more complex.

Because of this complexity, implementation of these systems become time-consuming, error-prone, and burdensome.

Motivation

What is needed?

- This complexity can be addressed with MDE techniques.
- Increase the level of abstraction.
- Automatically synthesize the system artifacts.

MPM

- MPM approach can lead to numerous benefits for the development of such complex systems.
- To cope with the complexity of the engineering process of WSN-based IoT systems.
- It is essential to explicitly model the entire process.

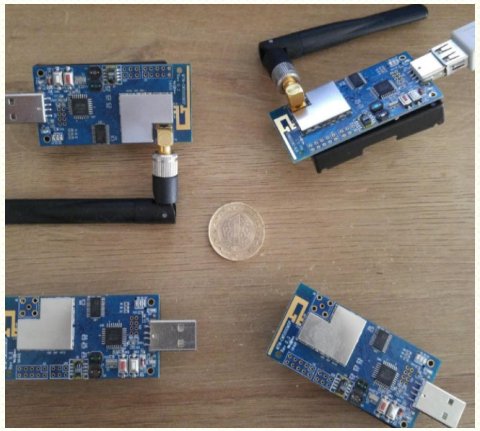
MPM of WSN-based IoT

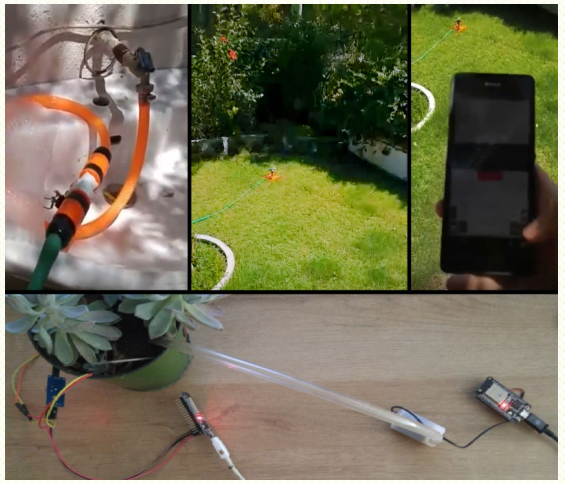
- **Complexity of multi-paradigm systems**
  - Requires description
  - Current methodologies are not enough.
  - Explicitly model

- **FTG+PM**
  - Multi-paradigm based engineering process representation using FTG+PM.
  - It enables the MDE lifecycle of these complex systems to be described.
  - We apply the FTG+PM approach to WSN-based IoT.

Irrigation System Case Study

- Irrigation systems are used to detect and set the level of moisture in the soil.
- System irrigates the environment until the soil reaches the desired moisture level.
- Sustainability of such places require efficient harvest increasing the quality of crops to achieve a profitable year.
- The dryness of the soil can be heterogeneous in the field
- Because of parameters such position of the sun, volatility of wind directions and soil characteristics.
- WSN devices can be distributed to collect moisture data. Then, corresponding solenoid valve can be triggered to irrigate the dry part.

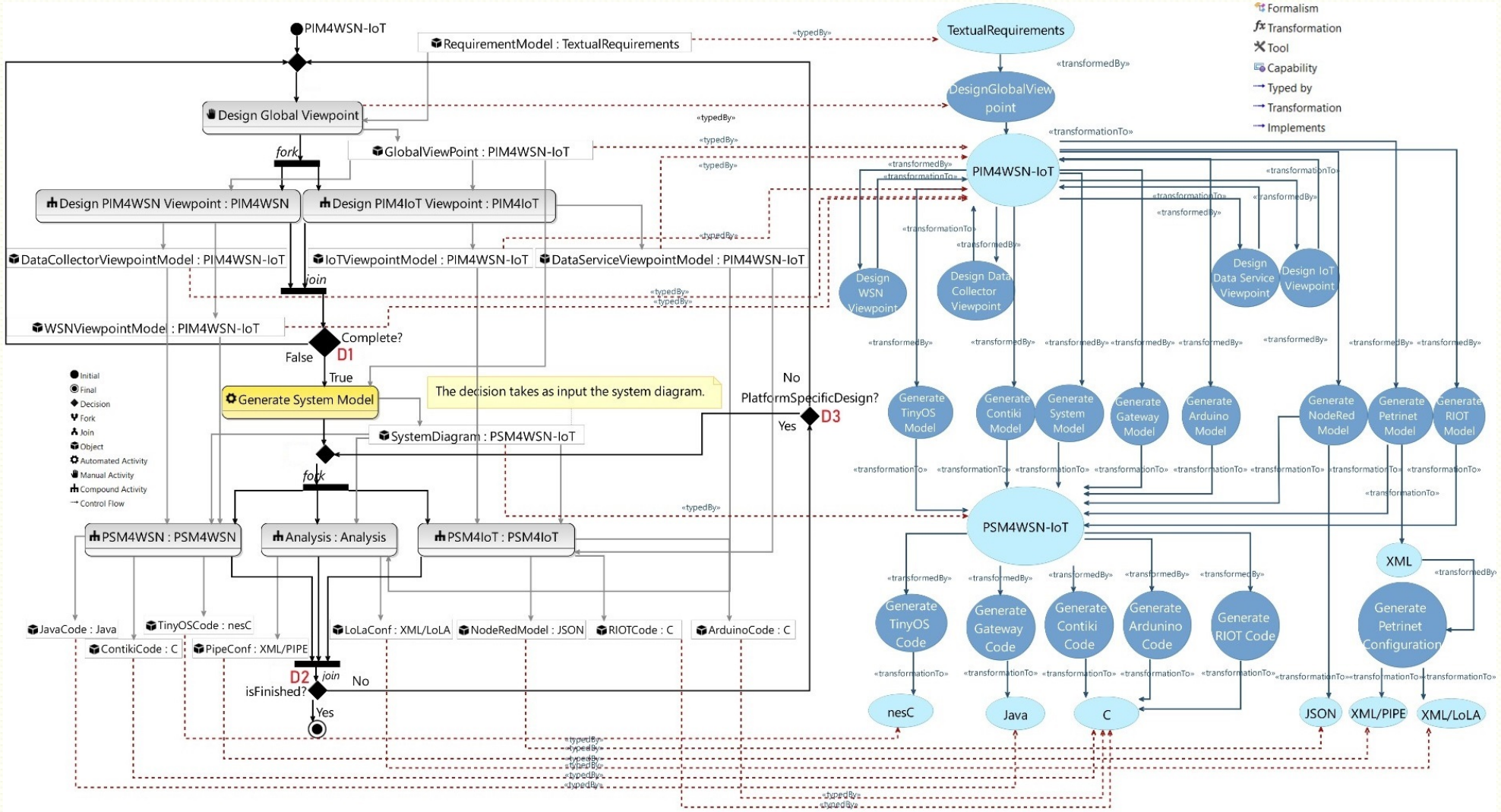




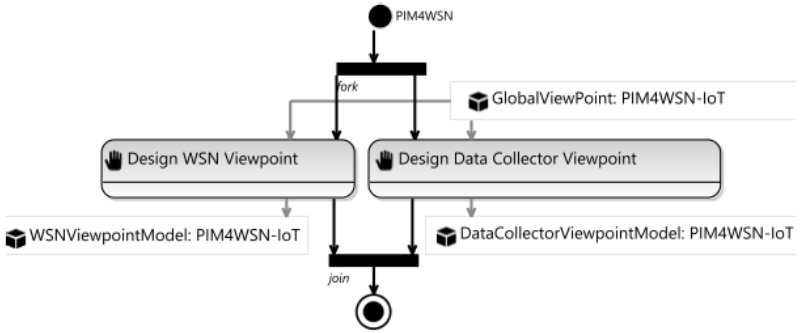
What is proposed?

- Reusable, adaptable and extensible FTG+PM.
- Applied as a whole or desired subset of the process for specific purpose.
- Discuss challenges and propose enhancements for FTG+PM.

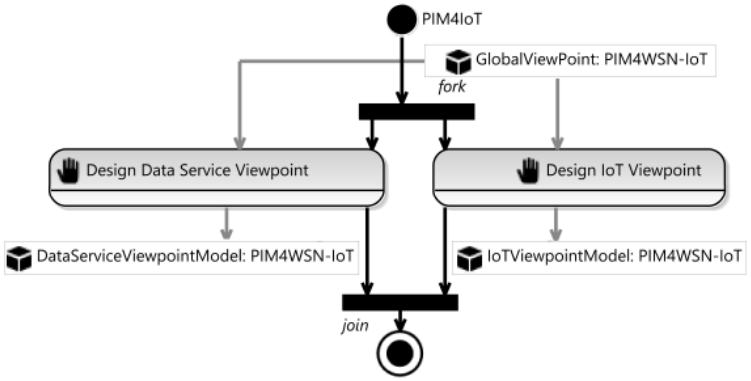
FTG+PM


PIM4WSN-IoT
RequirementModel : TextualRequirements
Design Global Viewpoint
fork
GlobalViewPoint : PIM4WSN-IoT
Design PIM4WSN Viewpoint : PIM4WSN
Design PIM4IoT Viewpoint : PIM4IoT
DataCollectorViewpointModel : PIM4WSN-IoT
IoTViewpointModel : PIM4WSN-IoT
DataServiceViewpointModel : PIM4WSN-IoT
join
WSNViewpointModel : PIM4WSN-IoT
Complete?
False
D1
True
Generate System Model
The decision takes as input the system diagram.
SystemDiagram : PSM4WSN-IoT
fork
PSM4WSN : PSM4WSN
Analysis : Analysis
PSM4IoT : PSM4IoT
JavaCode : Java
TinyOSCode : nesC
ContikiCode : C
PipeConf : XML/PIPE
LoLaConf : XML/LoLA
NodeRedModel : JSON
RIOTCode : C
ArduinoCode : C
D2
join
isFinished?
No
Yes
No
PlatformSpecificDesign?
Yes
D3
Initial
Final
Decision
Fork
Join
Object
Automated Activity
Manual Activity
Compound Activity
Control Flow
«typedBy»
«transformedBy»
«transformationTo»
TextualRequirements
DesignGlobalView point
PIM4WSN-IoT
Design WSN Viewpoint
Design Data Collector Viewpoint
Design Data Service Viewpoint
Design IoT Viewpoint
Generate TinyOS Model
Generate Contiki Model
Generate System Model
Generate Gateway Model
Generate Arduino Model
Generate NodeRed Model
Generate Petrinet Model
Generate RIOT Model
PSM4WSN-IoT
Generate TinyOS Code
Generate Gateway Code
Generate Contiki Code
Generate Arduino Code
Generate RIOT Code
XML
Generate Petrinet Configuration
nesC
Java
C
JSON
XML/PIPE
XML/LoLA
Formalism
Transformation
Tool
Capability
Typed by
Transformation
Implements

Nested Activities

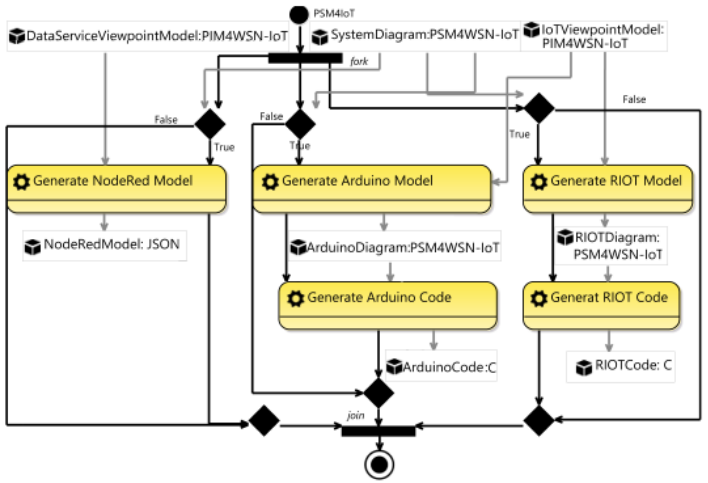


(a) Activity: PIM for WSN.

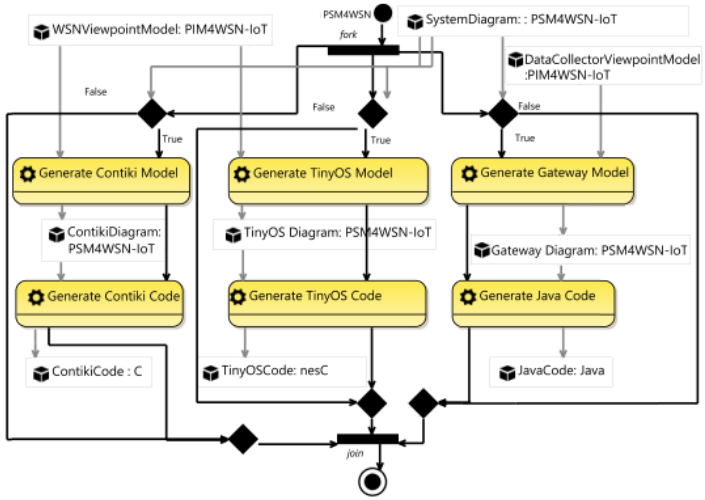


(b) Activity: PIM for IoT.

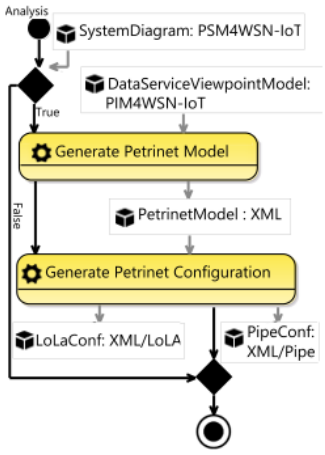
Fig. 9: Platform-independent modelling activities.



(a) Activity: PSM for IoT.



(b) Activity: PSM for WSN.



(c) Activity: Analysis.

Fig. 10: Platform-specific modelling activities.

Multi-paradigm Modelling

1 .Multi-Abstraction

The development process is associated with multiple abstraction levels:

- Platform-independent level
- Platform-specific level
- System analysis level
- Target platform (code) level

To support these abstraction levels, MDE approach is composed of

- A platform-independent modelling language (PIM4WSN-IoT)
- A platform-specific modelling language (PSM4WSN-IoT).
- Each of these languages has several viewpoints of the system.

Five artifacts as executable code, three artifacts as configuration files.

Multi-paradigm Modelling

2 .Multi-Formalism

In many domains, engineers benefit from various disciplines using multiple formalisms to achieve a complex task.

- DSMLs.
- Complexity.
- Right level of abstraction.

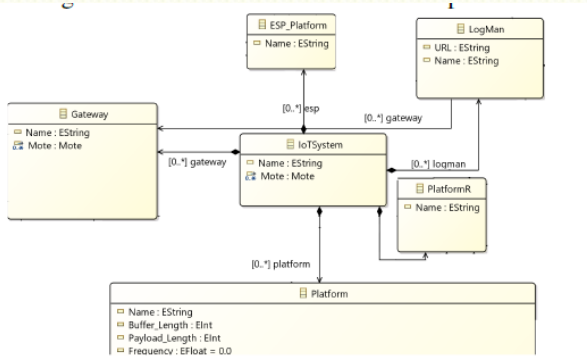


Fig. 4: Metamodel subset: System Viewpoint.

As mentioned;

- PIM4WSN-IoT has four different viewpoints.

WSN, IoT, Data service and middleware services for WSN-based IoT Systems.

- PSM4WSN-IoT includes seven different viewpoints.

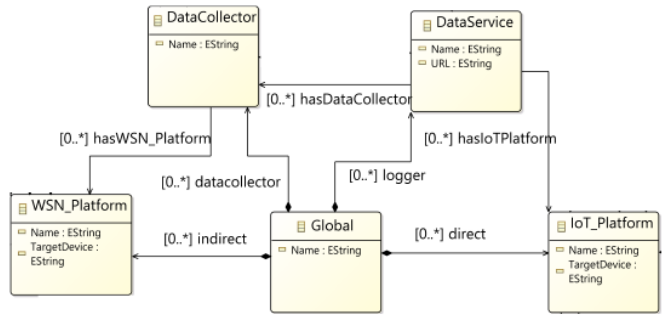


Fig. 2: Metamodel subset: Global Viewpoint.

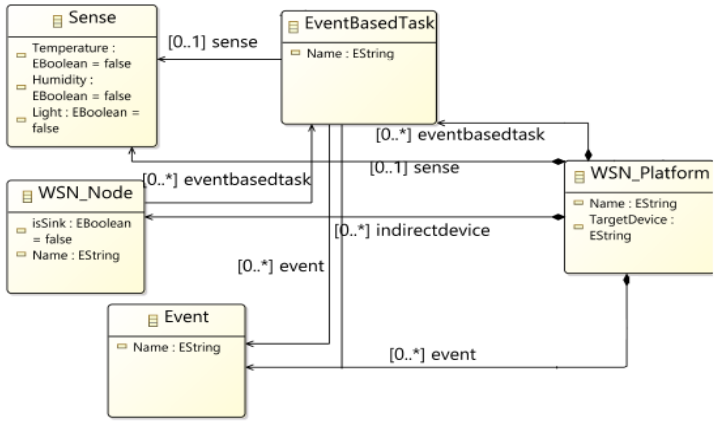


Fig. 3: Metamodel subset: WSN Viewpoint.

Multi-paradigm Modelling

3. Multi-View

As the framework follows MDA principles, various viewpoints are created for the formalisms at each of the PIM and PSM-levels.

Each PIM viewpoint is transformed to another PSM viewpoint

- The DataCollector viewpoint serves as an architectural model for data aggregation. PSM -> Gateway.
- The DataService viewpoint has two roles, generation of Petri-net models and Node-Red models. Mainly used for network topology.
- IoT Viewpoint
- WSN Viewpoint

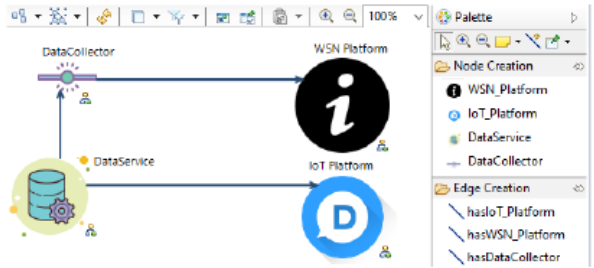


Fig. 6: Platform-independent global model of the irrigation system case study.

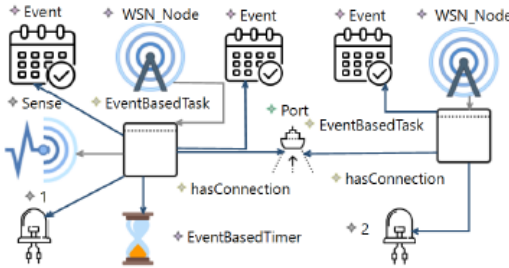
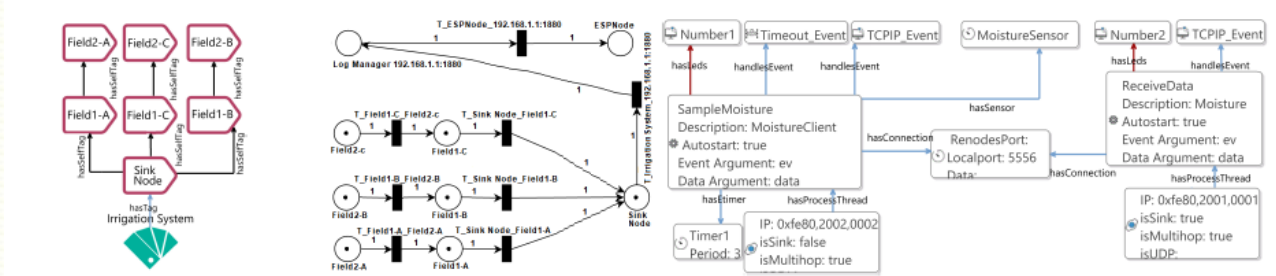


Fig. 7: Platform-independent WSN model of the irrigation system case study.



(a) Data Service Viewpoint. (b) Petrinet model. (c) Contiki Model.

Fig. 8: Smart irrigation system: model instances of different viewpoints.

Discussion

Potential FTG+PM Enhancements

Addressing viewpoint-based language engineering:

- Complex languages with viewpoints cannot be indicated in the FTG+PM.
- The multiple viewpoints of a language along with the link between a model and its viewpoint is not currently evident in the FTG.
- Each formalism in the FTG would be associated with any corresponding viewpoints with a <<representedBy>> link.
- Each input/output model in the PM would be associated with the appropriate viewpoint in the FTG with a <<viewedAs>>

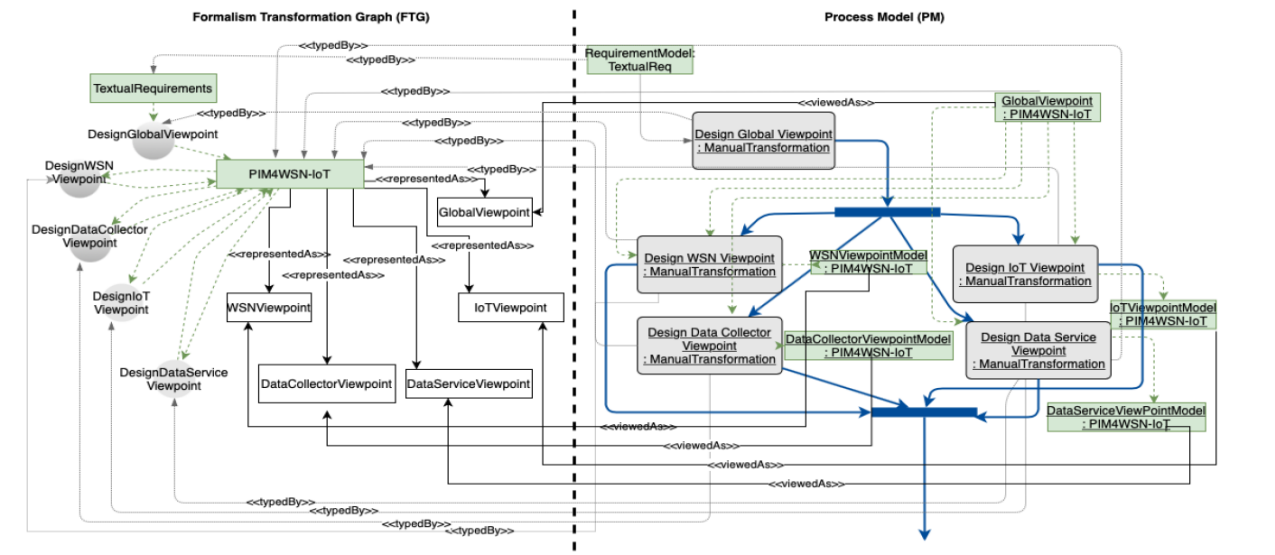


Fig. 11: FTG+PM slice with viewpoints.

Discussion

Generation of FTG:

- Creating FTG less cumbersome, FTG+PM tools could support automated discovery of the (partial) FTG based on the PM.

Orchestration and Traceability Analysis of the Process Model

- Automated execution of FTG+PM makes possible to generate and retain useful trace information including conformance and compatibility checks between the various models and transformations.

- In the WSN-based IoT systems, this would be beneficial on various fronts.
- As an example, incorrect IP that needs to be corrected at the code-level, it can be traced back to find the models and elements that include the incorrect IP.

Thank you for your attention