



COST

**European Network on New Sensing Technologies for Air
Pollution Control and Environmental Sustainability - *EuNetAir***

COST Action TD1105

**A Holistic Approach in the Development and Deployment of
WSN-based Applications and Systems**

Second Scientific Meeting, Queens' College, University of Cambridge, 18-20 December 2013



Industrial Systems Institute

Prof. George Papadopoulos

MC Member, Sub-WG2.3 Leader

Industrial Systems Institute / Greece

Developing and Deploying WSNs

- Wireless Sensor Networks (WSNs) constitute a rapidly evolving networking area:
 - Environmental, health, security and industrial applications
- Refer to different mechanisms, protocols, algorithms, HW:
 - Scheduling policies, OS, Security mechanisms
 - Routing and MAC protocols
 - Sensors, Actuators, Radio Tx/Rx
- Employ special characteristics:
 - Constrained energy, CPU and memory resources
 - Multi-hop communication

different requirements regarding system performance and QoS

different levels of knowledge

special knowledge

Developing and Deploying WSNs

- Need for a boost to WSN widespread use:

- User friendliness
- Practicality
- Efficiency

A tool integrating all appropriate features for the design of a WSN

- Evaluation, verification and guidance:

- Critical features identification
- Application requirements consideration
- Design indications and suggestions

Need for profiling-modeling of SW-HW

- Long life-cycle, ease of WSN SW development:

- Flexibility, extendibility, reusability

Componentizing SW

Desirable WSN features_Sustainability

- Low power consumption
 - Long operational lifetime for battery-powered unattended WSN nodes
- Joint optimization of connectivity and energy efficiency
 - Best-effort utilization of constrained radios in WSNs & min energy cost
- Self-calibration and self-healing
 - Recovering from failures and errors to which WSNs are prone
- Efficient data aggregation
 - Lessening the traffic load in constrained WSNs
- Programmable and reconfigurable stations
 - Allowing for long life-cycle development
- System security
 - Enabling protection of data and system operation
- Short development time
 - Short time-to-market for WSN systems
- Simple installation and maintenance procedures
 - Widespread use of WSNs

Outline

- ❖ The main contribution of the Industrial Systems Institute and the University of Patras to the ARTEMIS project WSN-DPCM focuses on:
 - *Middleware level*: seamless connectivity and interoperability issues over widely heterogeneous device and communication technologies,
 - *Development Tool level*: efficient hardware-in-the-loop simulation techniques and synthesis model evaluation modules, and
 - *Planning Tool level*: connectivity evaluation, critical node detection and links reduction modules

Middleware (1)

❖ Motivation:

- Great “programming complexity gap” between development of WSN applications and appropriate handling of underlying system operation:
 - dynamic changes of the operational environment
 - different user-application requirements
 - heterogeneity
 - WSN operation specificities, resulting to special knowledge requirement
- Application development turns to be a rather low-level programming procedure, resulting to:
 - relatively decreased energy efficiency and QoS, high resource consumption, time-cost
- Lack of a unified basis of development to promote:
 - interoperability with other systems
 - system extendibility
 - reusability of implementations
 - system adaptation

Middleware (2)

❖ Main Objectives:

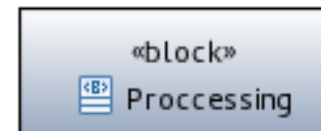
- Implement a middleware architecture which combines the state of the art in disjoint or loosely coupled research directions:
 - Adoption of a uniform MDE approach which maps well and through already existing standard notations and SW modeling constructs
 - Use of generalized high level application programming abstraction definitions
 - Deployment over a generative service/component execution framework, supporting both system and application component reconfiguration/reprogramming
 - Support of design-time and configurable run-time resource consumption awareness
 - Interoperability promotion based on mappings over existing international standards and industry driven specifications

Middleware (3)

❖ High-level Functional Abstractions:

- Appear on top of lower level abstractions, which hide functionalities corresponding to platform and network interfaces:
 - CPU, memory storage, radio, network stack protocols
- Encapsulate functionality of system or application level components:
 - Processing
 - representing application level algorithms and logic
 - Transducer (i.e. Sensors/Actuators), Storage, Communication
 - hiding underlying heterogeneity & corresponding access mechanisms
 - Parameter
 - defining the employed data model

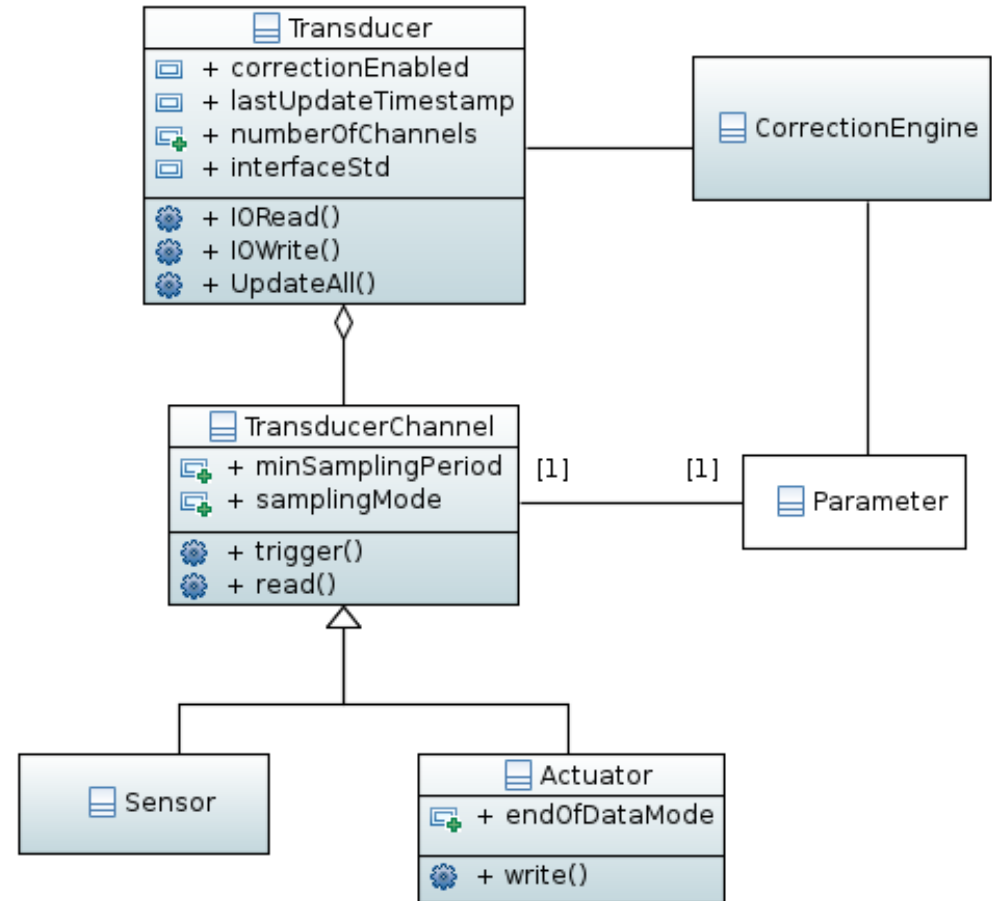
**application-level
abstraction**



Middleware (4)

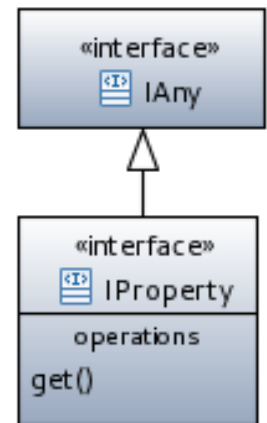
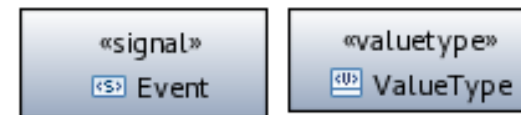
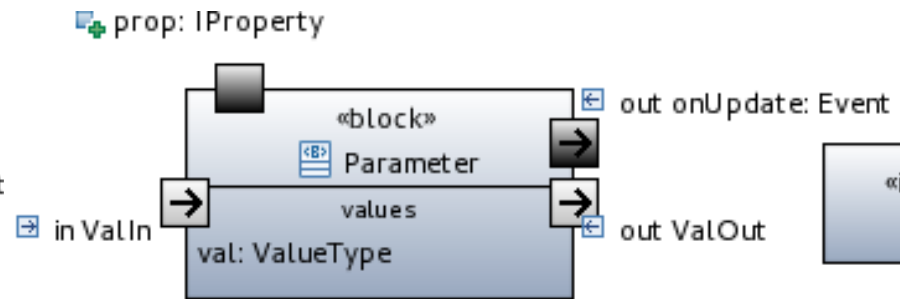
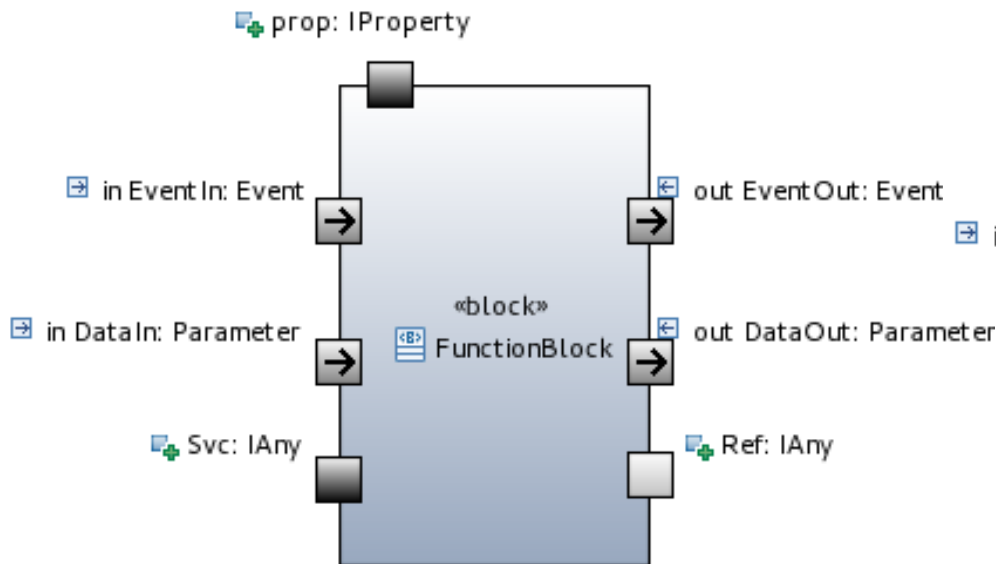
❖ IEEE-1451 compatible Transducer Framework:

- Standard description of transducer's properties in a more generic/structured way
- Definition of common commands and functions to access transducers
- Correction engine performs further processing on measured transducer values
- Leading to short development cycles and easy integration of WSNs



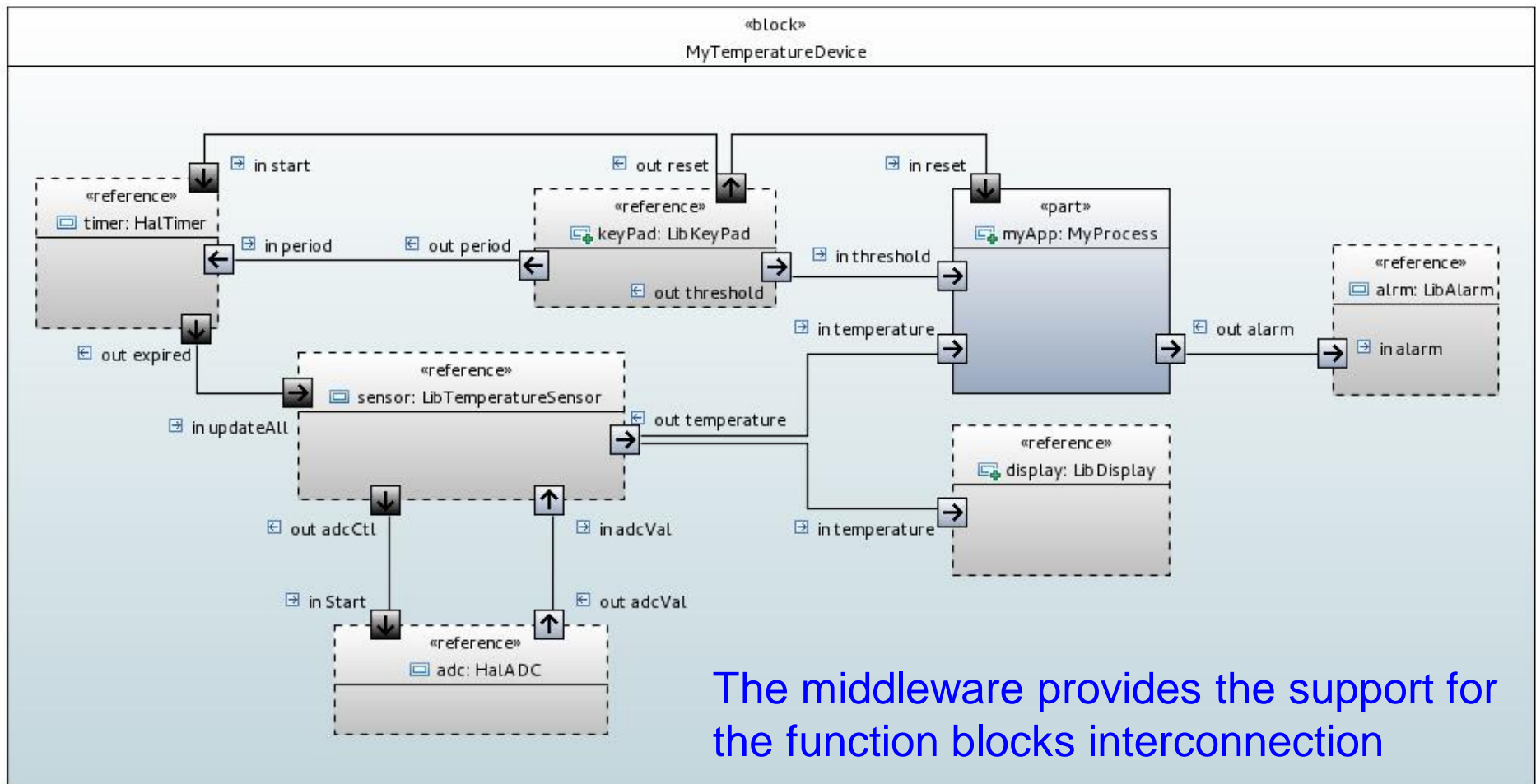
Middleware (5)

- ❖ Basic Component Framework supported by Middleware:
 - Combines CBD principles of accessing functionality implementations in composing WSN systems/applications through interface contracts:
 - IEC-61499: event-driven & data-flow process orientation
 - SysML flow ports representation
 - SCA: typical synchronous method for call semantics
 - SysML ports representation of service-reference-properties features



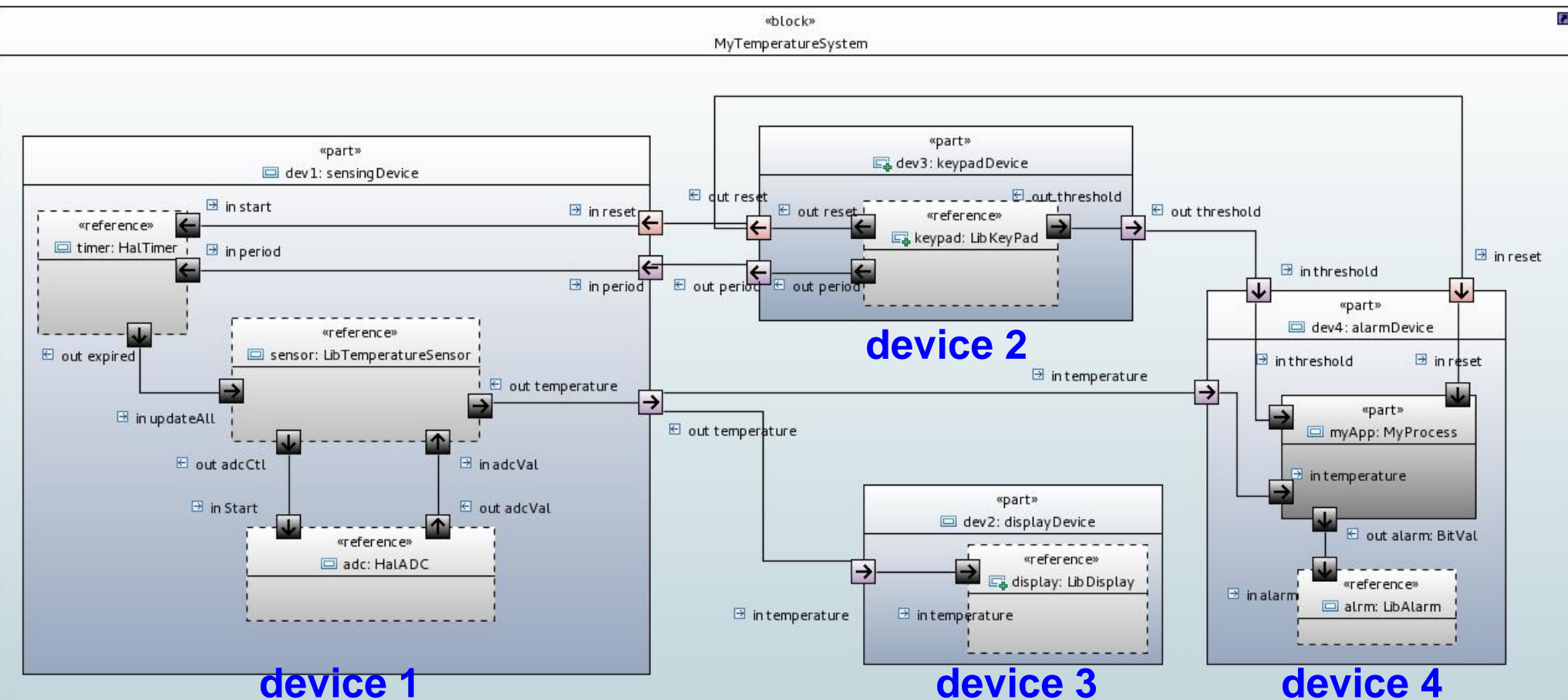
Middleware (6)

- ❖ Application development example (temperature sensing):
 - *Local node-level composition* of a simple application which:
 - Displays the current temperature in a local device
 - Turns-on an alarm when the temperature exceeds a threshold



Middleware (7)

- ❖ Application development example (temperature sensing):
 - *Distributed composition* of the same simple application where:
 - Display, keypad, alarm and sensor parts exist on separate devices and exchange control and data seamlessly over the network



Development Tool (1)

❖ Motivation:

- Development of WSNs has not achieved adequate widespread use because:
 - In-depth knowledge is required at various levels: OSs, protocols, platforms, programming etc.
 - There are no fully integrated environments facilitating all phases of development: Design, Develop, Test, Validate, Maintain, Extend
 - WSN Simulation study suffers from various shortcomings:
 - Lack of accurate models
 - Uncertainties of hardware influence to performance/behavior
- **Result: Inefficient cycles of development and unreliable power consumption estimation**

Development Tool (2)

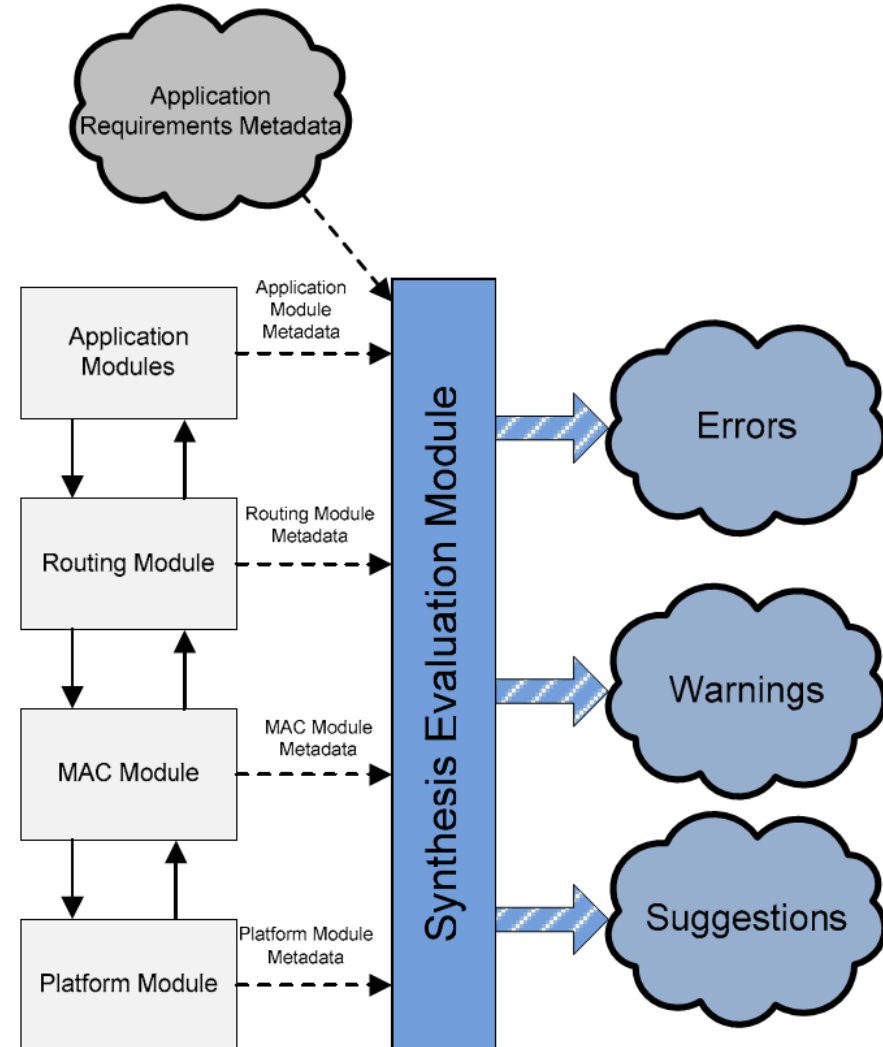
❖ Main Objectives:

- Offer useful and practical tools enhancing the following aspects of DT for both experts and non-experts by means of Synthesis Evaluation Module (SEM)
 - Increase User-friendliness
 - Increase Efficiency
 - Decrease complexity
- Offer Hardware-in-the-Loop techniques and propose approaches enhancing simulation-based features of prominent simulators
 - Increase simulation accuracy
 - Take hardware features into consideration
 - Accurate power consumption estimation

Development Tool (3)

❖ Synthesis Evaluation Module (SEM) Design:

- Application Requirements:
 - Communication traffic specs
 - Communication approach
 - Node density
 - Packet loss
 - Real-Time requirements
- Properties of SW modules in different levels (i.e. network stack)
- Leading to substantial support of development process



Development Tool (4)

❖ Synthesis Evaluation Module Impact:

Based on the User Choices and modules' properties correlation the SEM outputs indicate:

- *Errors:*
 - Composition failures
 - Compatibility issues
 - Inconsistencies
- *Warnings:*
 - Performance degradation
 - Operation failures
- *Suggestions:* Indications leading to optimum SW stack
 - Concerning specific components/protocols configuration
 - Template SW synthesis based suggestion

Development Tool (5)

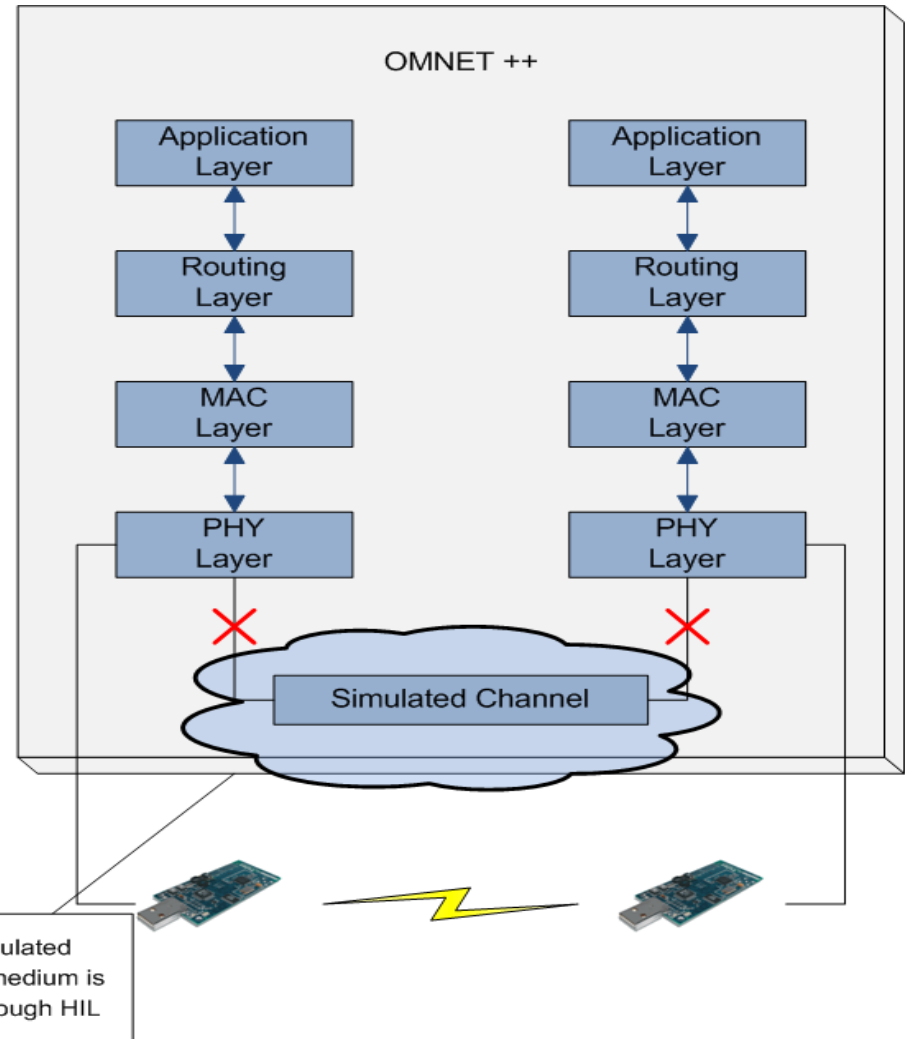
❖ Hardware-in-the-Loop Rationale:

- Integrate aspects of the a real WSN operation into dominant WSN simulation environments
- Omnet++ is selected as one of the most prominent network simulator providing:
 - High modularity and flexibility
 - Component based architecture
 - Various WSN oriented frameworks focusing on different aspects
 - ✓ Castalia : Focusing on PHY layer mostly
 - ✓ MiXiM : Focusing communication protocols

Development Tool (6)

❖ Hardware-in-the-Loop Rationale: 1st Aspect

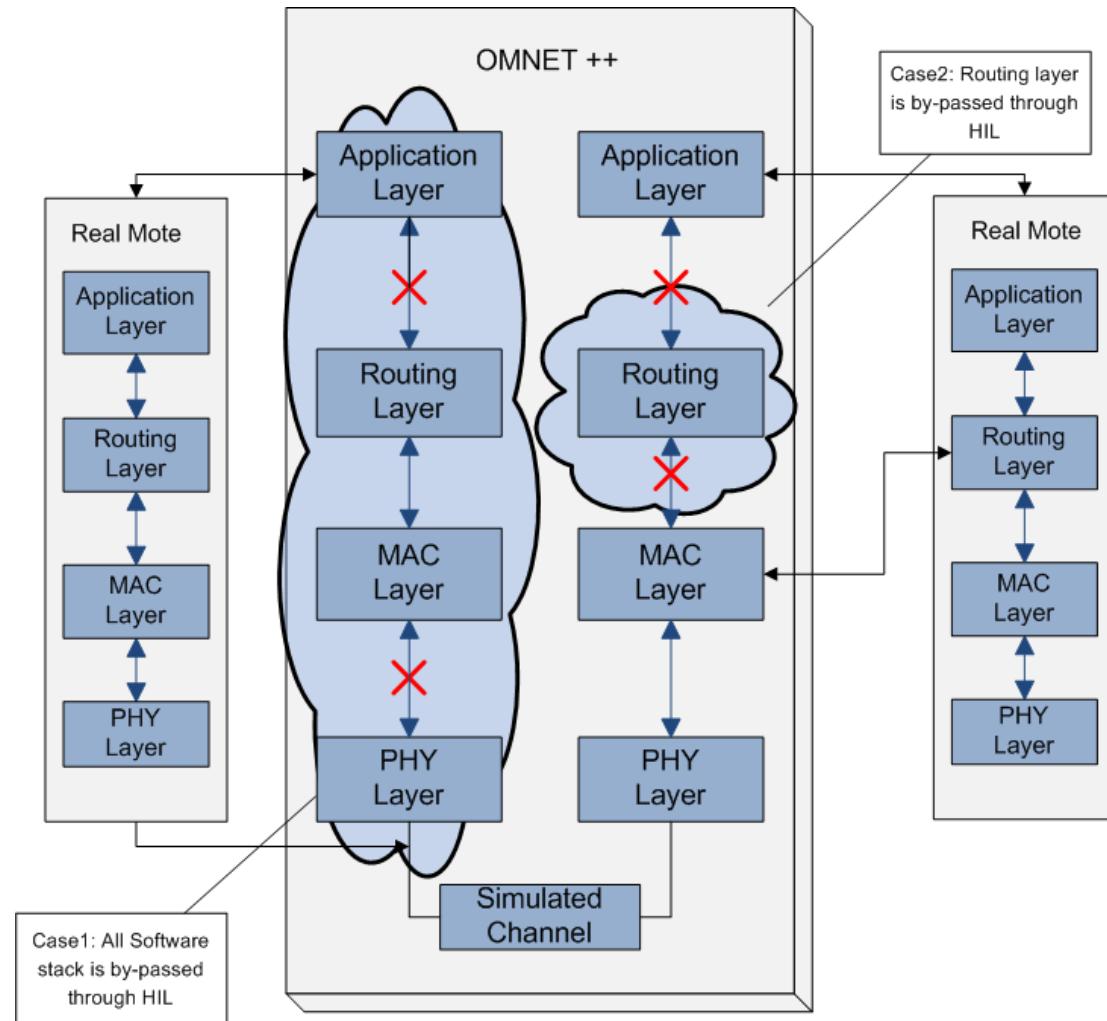
- Simulation taking into consideration real wireless transmission characteristics



Development Tool (7)

❖ Hardware-in-the-Loop Rationale: 2nd Aspect

- Simulation taking into consideration processing capabilities



Deployment: Planning Tool (1)

❖ Motivation:

- WSN deployment provides answers to fundamental issues:
 - Number of nodes needed to meet overall system objectives
 - Deployment of these nodes to optimize network performance
 - Adjustment of network topology and sensor node placement in case of changes in data sources and network malfunction
- WSN simulation suffers from various shortcomings:
 - Insufficient propagation models in terms of environmental parameters and WSN hardware variations
 - Leading to inaccurate positioning and link quality estimation
 - Lack of accurate energy models
 - Leading to inaccurate energy-waste-per link and overall network lifetime estimation.

Deployment: Planning Tool (2)

❖ Main Objectives:

- Dominant deployment objectives:
 - Optimal sensing coverage of an area
 - Realistic representation of signal propagation considering all environmental and HW particularities
 - Optimal and energy efficient connectivity
 - Construction of network topology based on application-driven network requirements
 - Simple installation and maintenance
- Of paramount importance are the aspects of:
 - *Connectivity*: seamless capability of getting data from the source to the appropriate destination
 - *Topology*: nodes and links that allow direct communication

Deployment: Planning Tool (3)

- Connectivity evaluation and identification of critical nodes:
 - *Critical nodes* determine the reliable operation of the network, since possible malfunction or removal leads to partition
 - Connectivity evaluation algorithms encompassing:
 - ✓ the Algebraic Graph Theory (AGT) approach: representing network topologies as graphs and using linear algebra and matrix theory
 - ✓ the Depth First Search (DFS) approach
- Topology control:
 - Generate initial network topology based on application requirements
 - Evaluate reduced topology alternatives, preserving the connectivity and taking into account the communication overhead
 - Topology control algorithms based on: Minimum Spanning Tree (MST), transmission power and neighborhood exploitation

Deployment: Planning Tool (4)

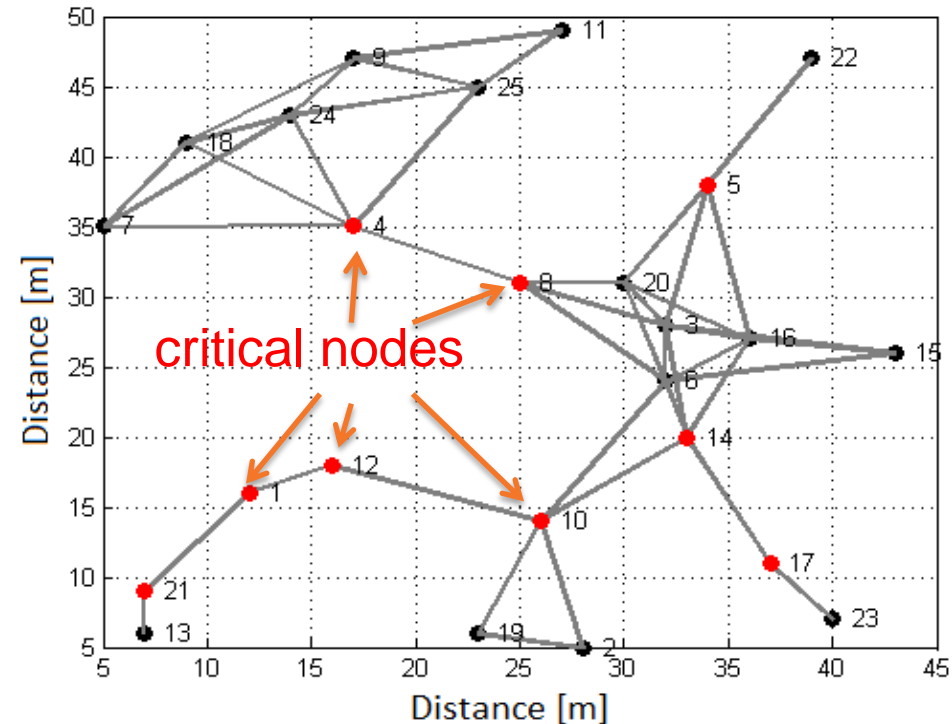
❖ Connectivity evaluation example:

● Input:

- Area size: 50m x 50m
- Number of nodes: 25
- Tx power: 0dBm
- Height from the ground: 0.1m
- Propagation model:
2-ray ground based

● Output:

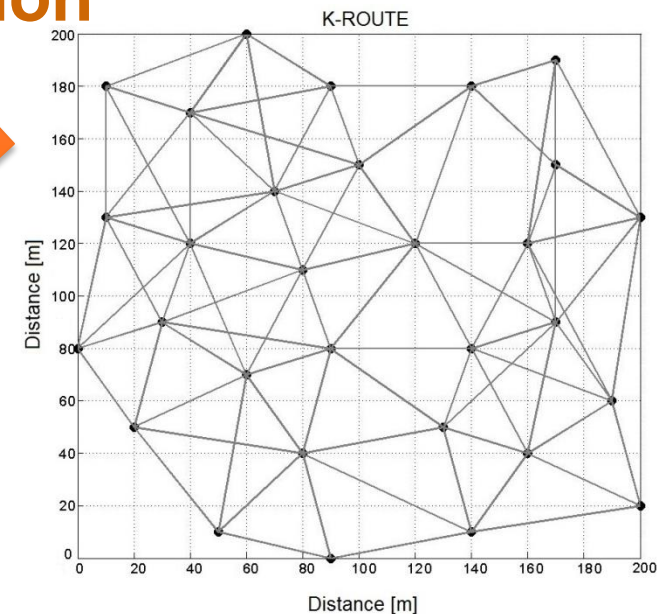
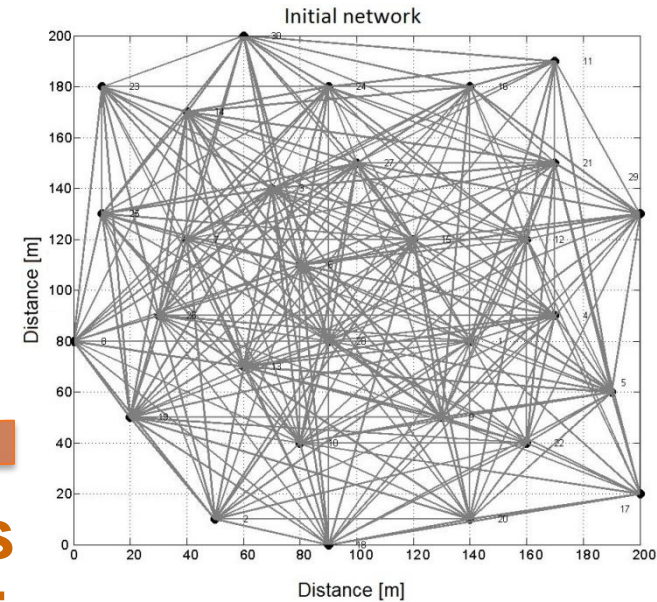
- Status of the network: connected
- Number of critical nodes: 9
 - Node IDs: 1, 4, 5, 8, 10, 12, 14, 17, 21



Deployment: Planning Tool (5)

❖ Topology control example:

- Initial topology:
 - Number of nodes: 30
 - Number of links: 582
 - Avg number of neighbors: 19
- Reduced topology:
 - Number of nodes: 30
 - Number of links: 174
 - Avg number of neighbors: 6
- System performance benefits:
 - Less contention, collisions and power waste



Deployment: Planning Tool (6)

Initial network

nodes: 167 gateways: 1
links: 4271 max hops: 3

ected!
work is at least 2-connected

Exit

Deployment: Planning Tool (7)

Google Earth interface showing a network topology overlay on an aerial view of a campus. The network consists of yellow nodes and cyan links. A 'BS' (Base Station) is marked with a red pin, and a 'GW' (Gateway) is marked with a red pin. A mouse cursor is hovering over a node. A statistics window is open on the right, displaying network metrics.

With links reduction
nodes: 167 gateways: 1
links: 1512 max hops: 7

Statistics window content:
ected!
work is at least 2-connected
logy - Statistics:
ed parts: 1
ighbors: 22
ighbors: 78
ighbors: 51
hops to the GW: 1
unction topology: Done



Thank you for your attention.

Prof. George Papadopoulos
MC Member, Sub-WG2.3 Leader
Industrial Systems Institute / Patras
papadopoulos@ece.upatras.gr



Abbreviations

- *MDE* : Model-Driven Engineering
- *SCA* : Service Component Architecture
- *CBD* : Component-Based Development