Architecting Self-Managing Distributed Systems Workshop
ASDS@ECSAW'15

Self-Adaptive Middleware for Wireless Sensor Networks: A Reference Architecture

Flávia C. Delicato

Federal University of Rio de Janeiro

Jesús T. Portocarrero

Paulo F. Pires

Elisa Y. Nakagawa

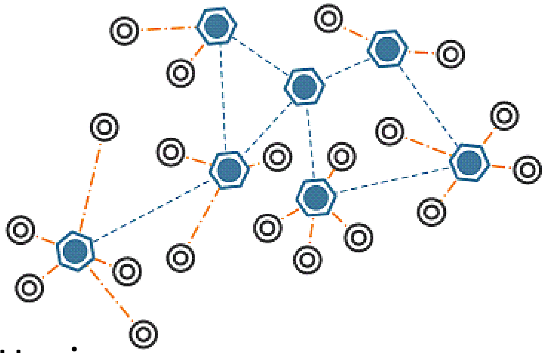
Flavio Oquendo

ECSA 2015

Dubrovnik / Cavtat – Croatia
September 7th, 2015

Introduction

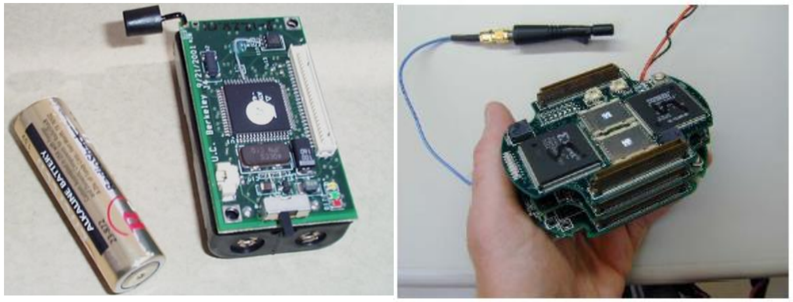
- **Wireless Sensor Networks (WSN)**



– Networks composed of tiny devices equipped with sensing, processing, storage, and wireless communication capabilities

– Limited computing resources, and are usually powered by batteries

– **Energy saving** is a key issue in order to prolong the network operational lifetime.

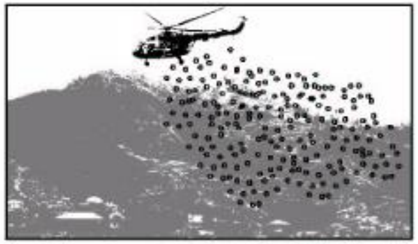


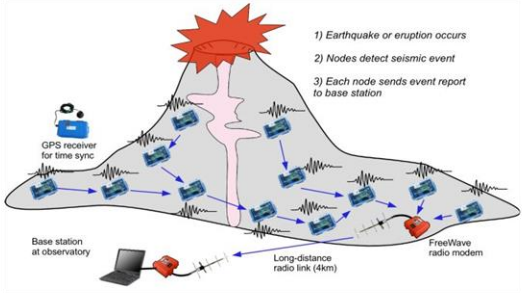
Mica Mote

iMote

Introduction

- **Wireless Sensor Networks (WSN)**
  - WSN are used in highly dynamic, sometimes remote and even hostile environments
    - Sensor nodes must be able to recover autonomously from link and node failures
      - keeping the network coverage and connectivity and meeting application requirements
      - with minimal human intervention




1) Earthquake or eruption occurs
2) Nodes detect seismic event
3) Each node sends event report to base station
GPS receiver for time sync
Base station at observatory
Long-distance radio link (4km)
FreeWave radio modem

- Major requirement of WSNs: capabilities of self-management and context-aware dynamic adaptation

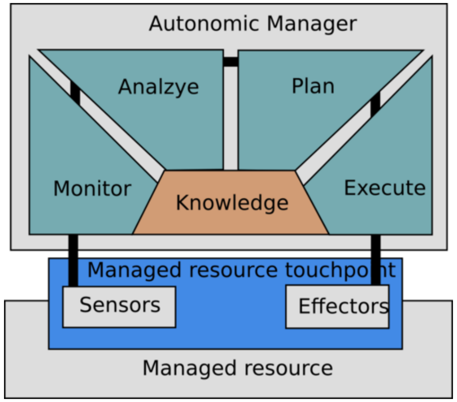
Introduction

- **Middleware for WSN**

  - A WSN middleware is a layered software that lies between application code and the communication infrastructure
    - They support the **development of WSN applications** by providing services and abstractions that hide details about hardware devices, low-level protocols and software
    - WSN middleware platforms developed so far provide good solutions to specify the high-level application logic and to deal with heterogeneity issues
      - Most of them do not provide an explicit way to define the network autonomic behavior **from an architectural perspective**

Introduction

- **Motivation**
  - Considering WSN basic requirements, Autonomic Computing (AC) is a promising option to design such systems



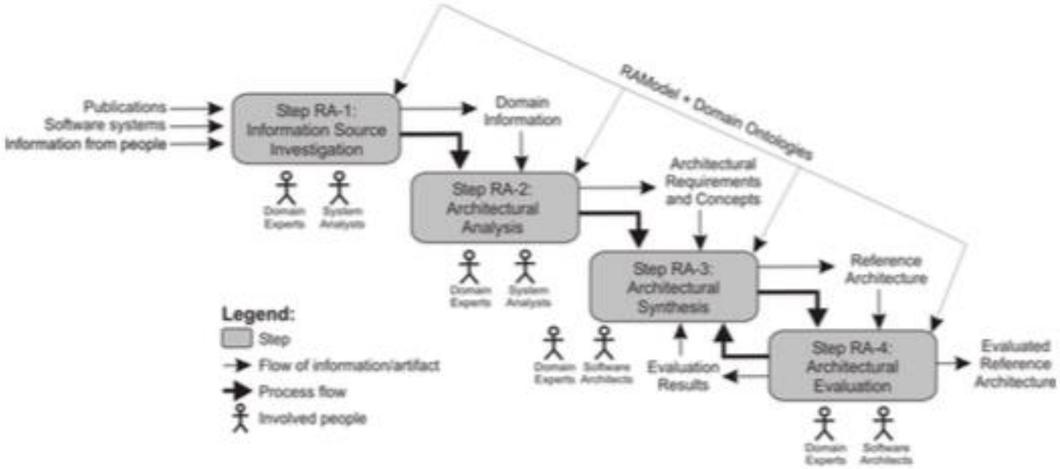
  - AC principles can be applied to optimize WSN resources, facilitate its operation, achieve the desired functionality and enable this type of network to manage itself with minimal human intervention.
  - Applying AC principles into WSN would be facilitated by the development of an architecture at the middleware level.
  - We found different approaches to provide autonomic behavior to WSNs, but they are just in preliminary stages and have several open issues
    - In particular, we noticed a lack of well-defined reference architecture designs to support the autonomy of WSNs

Proposal

- **Objective**
  - The main objective of this paper is to propose a reference architecture for self-adaptive WSN middleware
- **Contributions**
  - **First**: A RA that meets the specific requirements of self- adaptive WSN middleware
    - based on autonomic computing principles
    - maps IBM MAPE-K model to a set of SW components
  - **Second**: A specification of the reference architecture using pi-ADL
    - Pi-ADL provides:
      - A formal and theoretically well-founded language to describe dynamic software architectures under structural and behavioral perspectives
    - Pi-ADL is a suitable language to represent a WSN architecture, considering the highly dynamic environment of WSNs

ProSA-RA

- **Methodology** for building our RA was based in **ProSA-RA**
  - A process that systematizes the design, representation, and evaluation of RAs
  - Encompasses **four steps**:
    - – **Step RA-1** - Information Source Investigation: selection of information about processes and activities that could be supported by software systems of the target domain.
    - – **Step RA-2** - Architectural Analysis: Based on the selected sources, the requirements of the RA are identified.
    - – **Step RA-3** - Architectural Synthesis: The architectural description of the RA is built.
    - – **Step RA-4** - Architectural Evaluation


Publications
Software systems
Information from people
Step RA-1: Information Source Investigation
Domain Experts
System Analysts
Domain Information
RAModel + Domain Ontologies
Step RA-2: Architectural Analysis
Domain Experts
System Analysts
Architectural Requirements and Concepts
Step RA-3: Architectural Synthesis
Domain Experts
Software Architects
Reference Architecture
Evaluation Results
Step RA-4: Architectural Evaluation
Domain Experts
Software Architects
Evaluated Reference Architecture
Legend:
Step
Flow of information/artifact
Process flow
Involved people

ProSA-RA: Step RA-1

- **Information Source Investigation:**
  - Aims to select a set of information sources that allows identifying processes, activities, and tasks to be supported by self- adaptive middleware for WSNs
  - Most of the studies proposed lightweight solutions to implement autonomic behavior for WSN in order to:

    (i) adapt the network to dynamic environments (for instance, node energy depletion and arrival of new applications) and unpredictable events (for instance, node and link failures);

    (ii) save energy, extending the network lifetime;

    (iii) provide network scalability;

    (iv) provide reliability of sensing data and;

    (v) provide independence between the management functions of applications and network configuration.
  - Selected studies addressed the aforementioned concerns by leveraging the self-* properties.

ProSA-RA: Step RA-1

- **Information Source Investigation:** (Systematic Literature Review – Activity B)
  - Main **development approaches** of AC used to design middleware for autonomous WSNs:
    - Context-based reasoning (CBR) approach considers as meaningful context any information that affects node's operation.
    - Policy-based reasoning (PBR) approaches. A general way of implementing autonomic behavior in distributed systems is through the use of policies.
    - The Feedback Control Loop (FCL) approach is commonly applied in AC systems and most of them use four steps (monitoring, analysis, planning and execution).
    - Mobile agents (MA) approach provides a programming model in which applications consist of evolving communities of agents that share a WSN. Each agent is autonomous, allowing multiple applications to share a same network infrastructure.
    - Model transformation/Code Generation (MT/CG) approaches define an automatic process to derive different middleware configurations depending on the HW/SW of the deployed WSN. This process uses techniques such as MDD and SPL.

Most of the FCL solutions use context-based reasoning for the monitoring process and policy-based reasoning for execution an planning

ProSA-RA: Step RA-1

- **Information Source Investigation** (Reference Architectures - Activity C)

  We identified some relevant works proposing RAs for WSN, such as

  - The reference architecture for Sensor Networks Integration and Management (SeNsIM)
  - The Sensor Network Reference Architecture (SNRA)
    - architectural representation of sensor network entities' functions, activities, and roles
    - SNRA depicts the general operational, functional and technical characteristics of WSN.
  - The e-SENSE Project (Reference Model for Sensor Networks in B3G Mobile Communication Systems)
    - aims to provide the enabling technology required to capture the desired ambient intelligence surrounding the users of services and service related objects through a WSN environment.

  - Most of these architectures were designed using *ad hoc* approaches and represented using informal techniques
  - Furthermore, there is a lack of works specifically dealing with self-management in WSN

ProSA-RA: Step RA-2

- **Architectural Analysis**
  - Based on the outcomes of the previous step, this step aims to establish the requirements of the RA
  - The following activities were performed:

    (A) Definition of **architectural requirements**;

    (B) Definition of **quality attributes** for the RA.

ProSA-RA: Step RA-2

- **Architectural Analysis:** (A) Architectural requirements
  - **Req A**: A WSN contains one or more **sink nodes or base stations**, which must be endowed with a wireless communication interface and an interface with a **Gateway**, in order to integrate the WSN with external systems and other networks.
  - **Req B:** The RA must enable the definition of a set of **sensing-based applications**, responsible for defining **sensing tasks** and **quality attributes** to be attended by the system.
  - **Req C:** The RA must enable the definition of a set of **high-level goals**
    - these goals are composed of **adaptation policies** that will guide decisions and strategies to be used for configuring and adapting action plans in the network.
  - **Req D:** The RA must enable the self-management of the network by defining a set of **software components (middleware** components)
    - such components must be responsible for implementing the **feedback loops** (MAPE-K)
  - **Req E:** The RA must consider a **hierarchical topology** for the WSN organization.

Step 1	Req-A	Req-B	Req-C	Req-D	Req-E
I				X	
II				X	
III	X				X
IV				X	
V		X	X		

ProSA-RA: Step RA-2

- **Architectural Analysis:** (B) Quality attributes for the RA
  - **Security**: Security and privacy are of extreme importance for many WSN applications. Security attribute should provide different security levels according to each application need.
  - **Fault-tolerance and Recoverability**: The consistency of the states and the redundancy of the persistent data are critical software qualities necessary to achieve reliable services even when faults are injected to the system.
  - **Scalability**: As WSNs grow from small residential applications to commercial/industrial applications with thousands of nodes, their processing capacity must be able to grow, to both expand the complexity of the states and the amount of sensor node data the system manages, and maintain acceptable performance levels for real time services.
  - **Analyzability and Changeability**: Changes can include fixes, improvements and adaptations. WSN systems require the ability to allow diagnosis of weaknesses, causes of network failures or identifying parts to be modified.
  - **Adaptability**: Self-adaptive WSN application should have an autonomous behavior in order to adapt its operation and achieve the best network performance.

ProSA-RA: Step RA-3

- **Architectural Synthesis**
  - This step of ProSA-RA is intended to design the RA
  - For this, following activities were conducted:

    (A) Definition of the architectural styles and design patterns to be used in the RA;

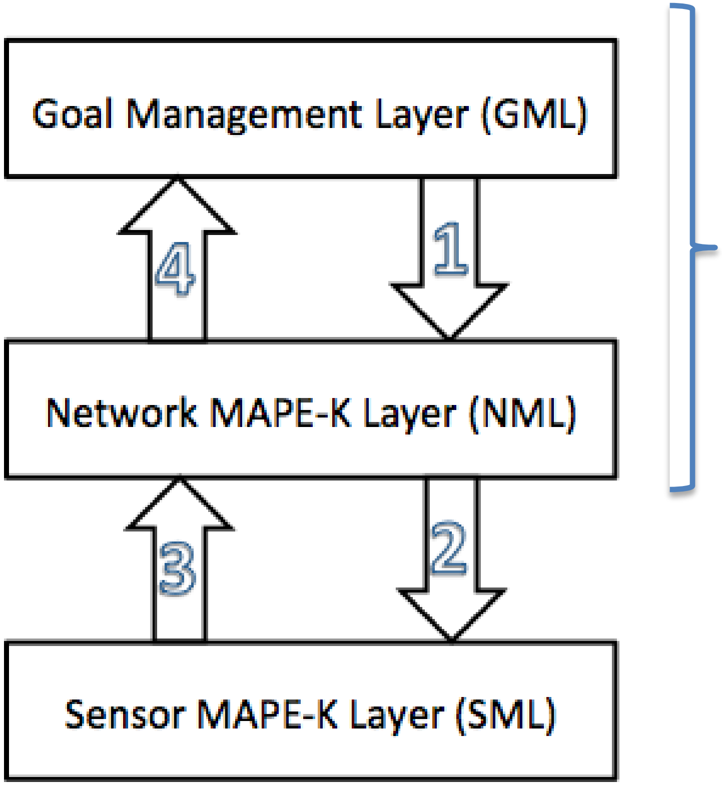
    (B) Description of components of the RA; and

    (C) Specification of the components using pi-ADL language

ProSA-RA: Step RA-3

- **The components of the RA are organized in 3 layers**

- The **Goal Management Layer** contains components that allow human interaction and are deployed in a **Gateway node**


Goal Management Layer (GML)
4
1
Network MAPE-K Layer (NML)
3
2
Sensor MAPE-K Layer (SML)

Gateway

Sensor Node

ProSA-RA: Step RA-3

- **Architectural Synthesis**

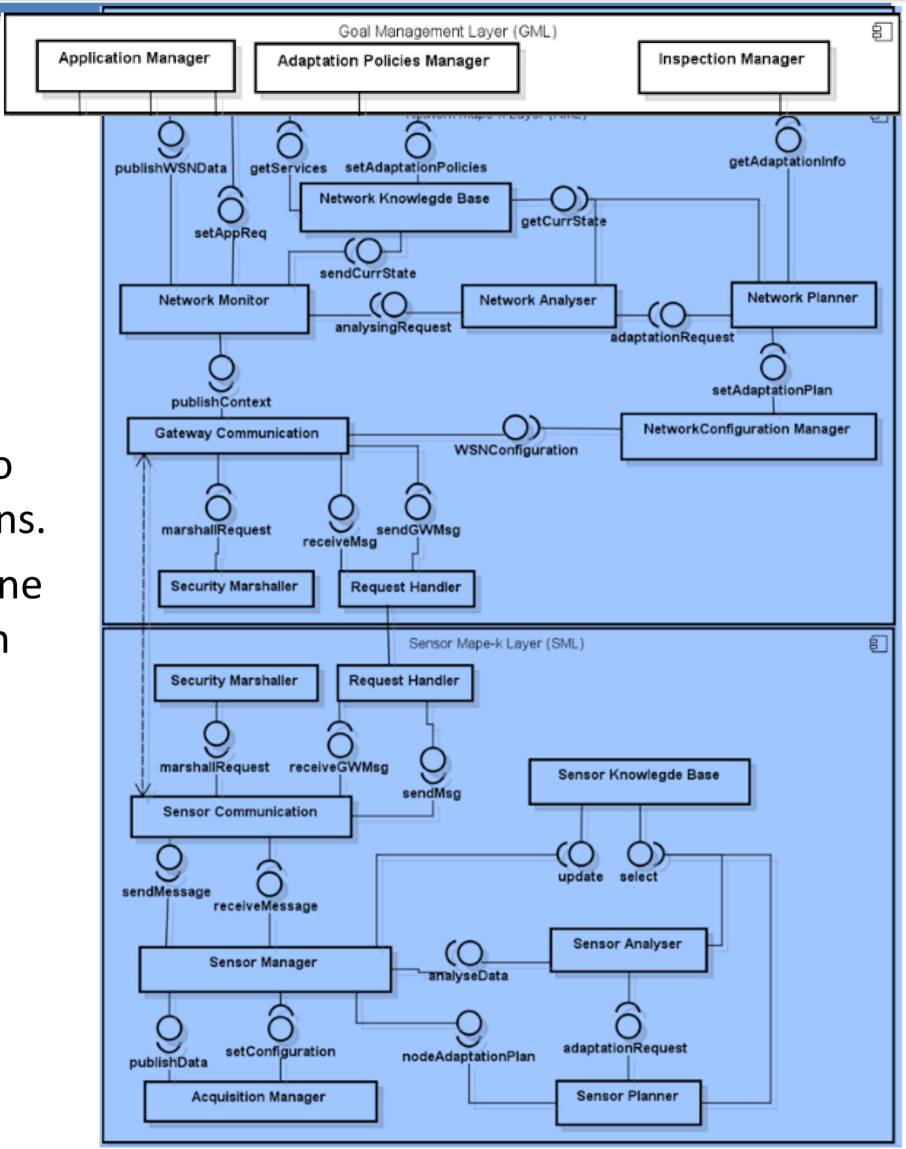
(B) Description of components of the RA:

– <u>**Goal Management Layer (GML):**</u> Contains components that allow human interaction

– <u>**Application Manager**</u> is used by end-users to create and submit sensing-based applications.

– <u>**Adaptation Policies Manager**</u> is used to define policies (by humans) to guide the adaptation actions

– <u>**Inspection Manager**</u> is used to inspect adaptation actions performed by the middleware.

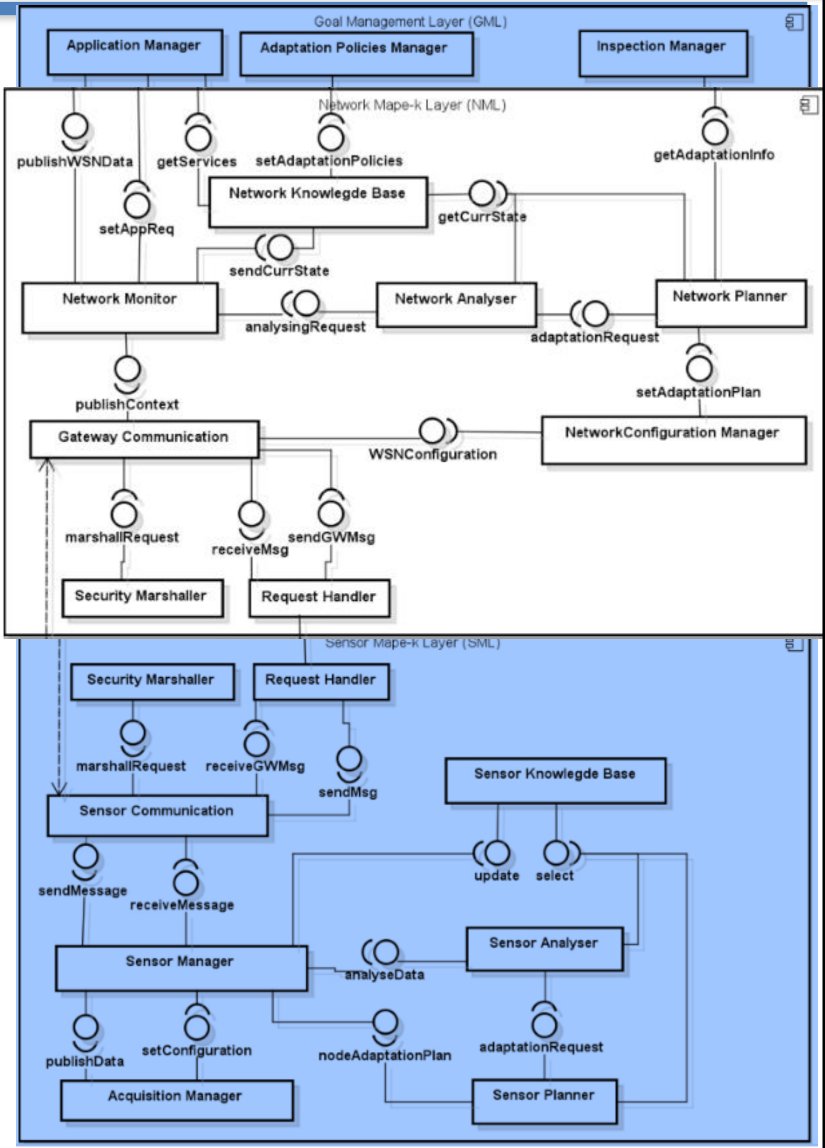


ProSA-RA: Step RA-3

- **Architectural Synthesis**

(B) Description of components of the RA:

- **Network MAPE-K Layer (NML):**
  - Performs Mape-k process for the whole WSN
  - Performs adaptation actions
    - For configuring the network
    - Using contextual information provided by the whole network
  - Main components:
    - Gateway Communication
    - Network Monitor
    - Network Analyser
    - Network Planner
    - Network Configuration Manager
    - Network Knowledge Base

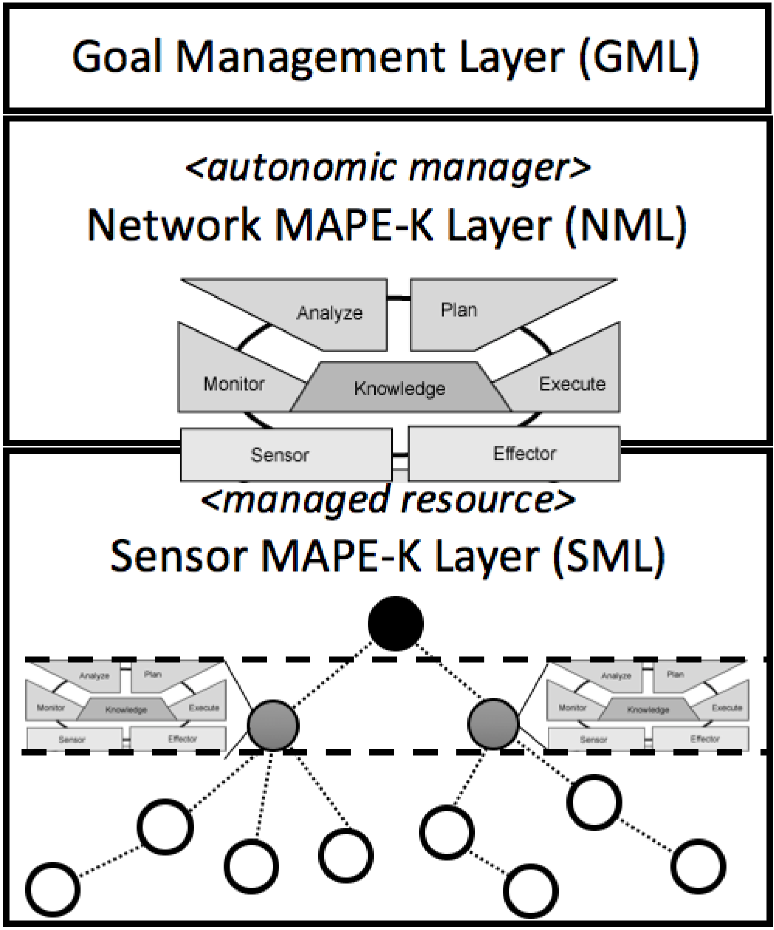


ProSA-RA: Step RA-3

- **Architectural Synthesis**

(B) Description of components of the RA:

- **Sensor MAPE-K Layer (SML):** allows adapting a **node configuration** according to the node context and adaptation policies
- Our RA considers a WSN as a set of heterogeneous nodes grouped in Clusters
  - Cluster heads are **Node managers**:
    - manage their own clusters
    - act as **autonomic manager**s of ordinary nodes
    - perform Mape-k process for the cluster
  - **Managed nodes** (ordinary node):
    - receive adaptation messages from CHs
    - act accordingly



Goal Management Layer (GML)

<autonomic manager>

Network MAPE-K Layer (NML)

<managed resource>

Sensor MAPE-K Layer (SML)

● - Sink Node

◍ - Node Manager *(cluster head)*

○ - Managed Node *(ordinary node)*

ProSA-RA: Step RA-3

- **Architectural Synthesis**

(B) Description of components of the RA:

– **Sensor MAPE-K Layer (SML**

Our RA currently considers as **context information**:

– battery level of sensor nodes,

– data delivery model,

– data sending rate,

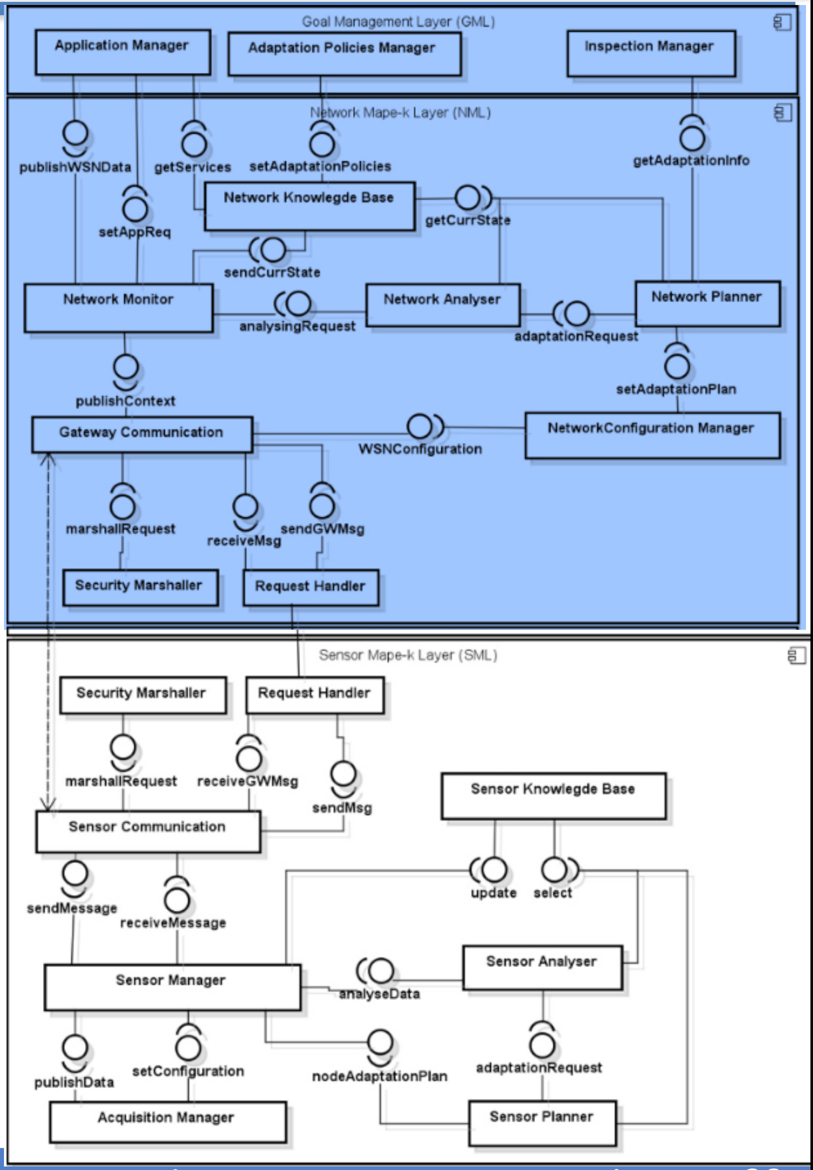
– state of nodes (active/inactive/idle),

– type of node (node manager/managed node)

– role of node (routing/sensing/storing),

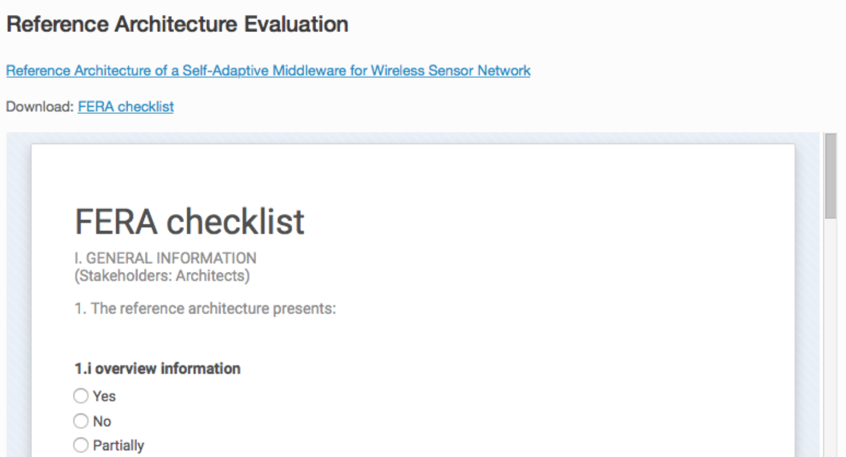
– signal strength, and

– localization.



ProSA-RA: Step RA-4

- **Architectural Evaluation**
  - In ProSA-RA, evaluation refers to the task of checking the architectural description of the RA along with diverse stakeholders aiming to detect defects in the description
  - ProSA-RA suggests using a checklist-based inspection approach named FERA (Framework for Evaluation of Reference Architectures) in order to verify:
    - (i) completeness of information related to the construction and content of the architectural views;
    - (ii) the adequacy of the provided RA documentation for the public usage; and
    - (iii) how ease is to instantiate the RA and change its components if necessary.



Reference Architecture Evaluation

Reference Architecture of a Self-Adaptive Middleware for Wireless Sensor Network

Download: FERA checklist

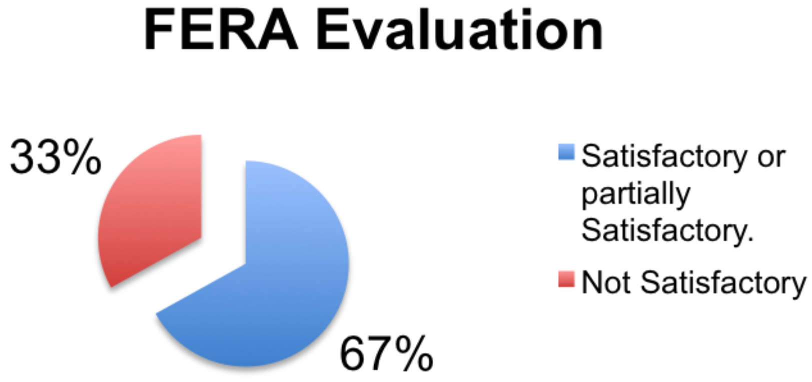
FERA checklist

I. GENERAL INFORMATION
(Stakeholders: Architects)

1. The reference architecture presents:

1.i overview information

○ Yes
○ No
○ Partially



FERA Evaluation

ProSA-RA: Step RA-4

- **Architectual Evaluation**
  - Four evaluators (two experts in RA and two experts in WSN) answered 93 questions
    - Available at: http://146.164.247.214/wordpress/rawsn/ra-evaluation/
  - Summarizing
  - **Positive aspects:**
    - Results showed that the provided abstraction level is suitable for the RA purposes
    - The concepts underlying the RA are clearly explained and its detail level favors the RA understanding
    - All modules of the RA are clearly identified, the relationships between these modules can be determined, and the runtime dependencies of these modules can be identified
    - The required hardware elements can be identified
    - RA is in conformance and complete regarding domain requirements, and it addresses the key issues of the WSN domain.

ProSA-RA: Step RA-4

- **Architectural Evaluation**
  - Summarizing
  - Negative aspects:
  - The main problems reported in the conducted inspection were the lacking of:
    - explicit definition of stakeholders,
    - specification of the RA variability mechanisms,
    - documentation regarding the threats for introducing the RA.
  - We intend to complete our architectural description in order to overcome the pointed drawbacks and leverage the RA adoption.

Conclusions and Future Directions

- Autonomic Computing is a promising option to meet basic requirements in WSN design
- **Ongoing work:** definition of a process to instantiate the RA
  - With support of a Model-driven tool for automatically generation of code
- **Future works** include the full assessment of the RA through the implementation of a prototype and a scenario-based evaluation
- We also plan to perform formal analysis of RA properties, related to completeness, consistency, and correctness
  - Using Pi-ADL validation tools

Architecting Self-Managing Distributed Systems Workshop
ASDS@ECSAW'15

Self-Adaptive Middleware for Wireless Sensor Networks: A Reference Architecture

UFRJ

PPGI PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

ubicomp Laboratório de Computação Ubíqua

CAPES

CNPq

FAPERJ Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro

nuffic

Flávia C. Delicato

Federal University of Rio de Janeiro

fdelicato@gmail.com

Thank You!!

Dubrovnik / Cavtat – Croatia
September 7th, 2015