

# *Srijan* : Macroprogramming for pervasive systems

Iraklis Leontiadis  
ARLES TEAM  
Iraklis.Leontiadis@inria.fr

29 juin 2010

# Overview

- Pervasive systems
- Macroprogramming
- *Srijan* toolkit
- Future work



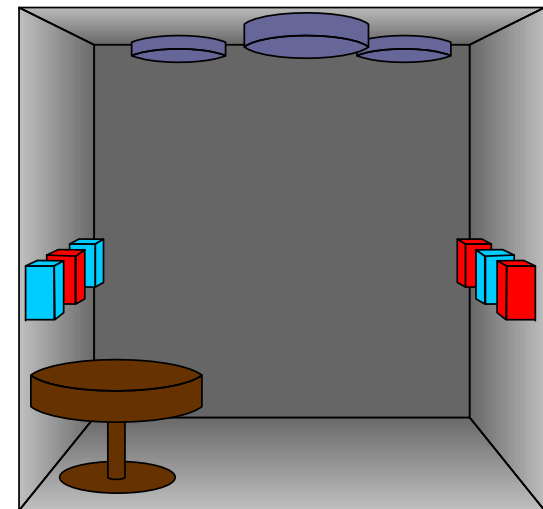
# Pervasive Systems(1)

- A class of mobile devices cooperate together.
- Aware of location , identity and resources.
- The end user does not know about the diversity of devices in resource characteristics (cpu , memory, connectivity).



# Wireless sensor networks applications

- Rapidly emerged as a new type of distributed pervasive systems in different domains.
- Building management systems.
- Traffic management.
- User tracking.
- Health monitoring.



# Macroprogramming (1)

- “Reason at the level of the system, not the nodes”
- The promise: make development of complex sensor network applications easy
- Inputs:
  - High-level application description. E.g. Task Graph
  - Information about target network. E.g. Topology [Tree]
- Output:
  - Customized node-level code for each node in the



# Macroprogramming (2)

- Why “macro”?
  - More and more sensing applications become more mature.
  - Complexity is increased with all these different type of nodes.
  - Nodes are heterogeneous.
  - Easy programming
  - More effective and less timeless development.



# *Srijan* toolkit

- Enables easy programming of wireless sensor nodes
- You define the behavior of the system without thinking the specifications of each node.
- *Srijan* is responsible to create code templates for different types of nodes
- Based on the ATaG macroprogramming language.



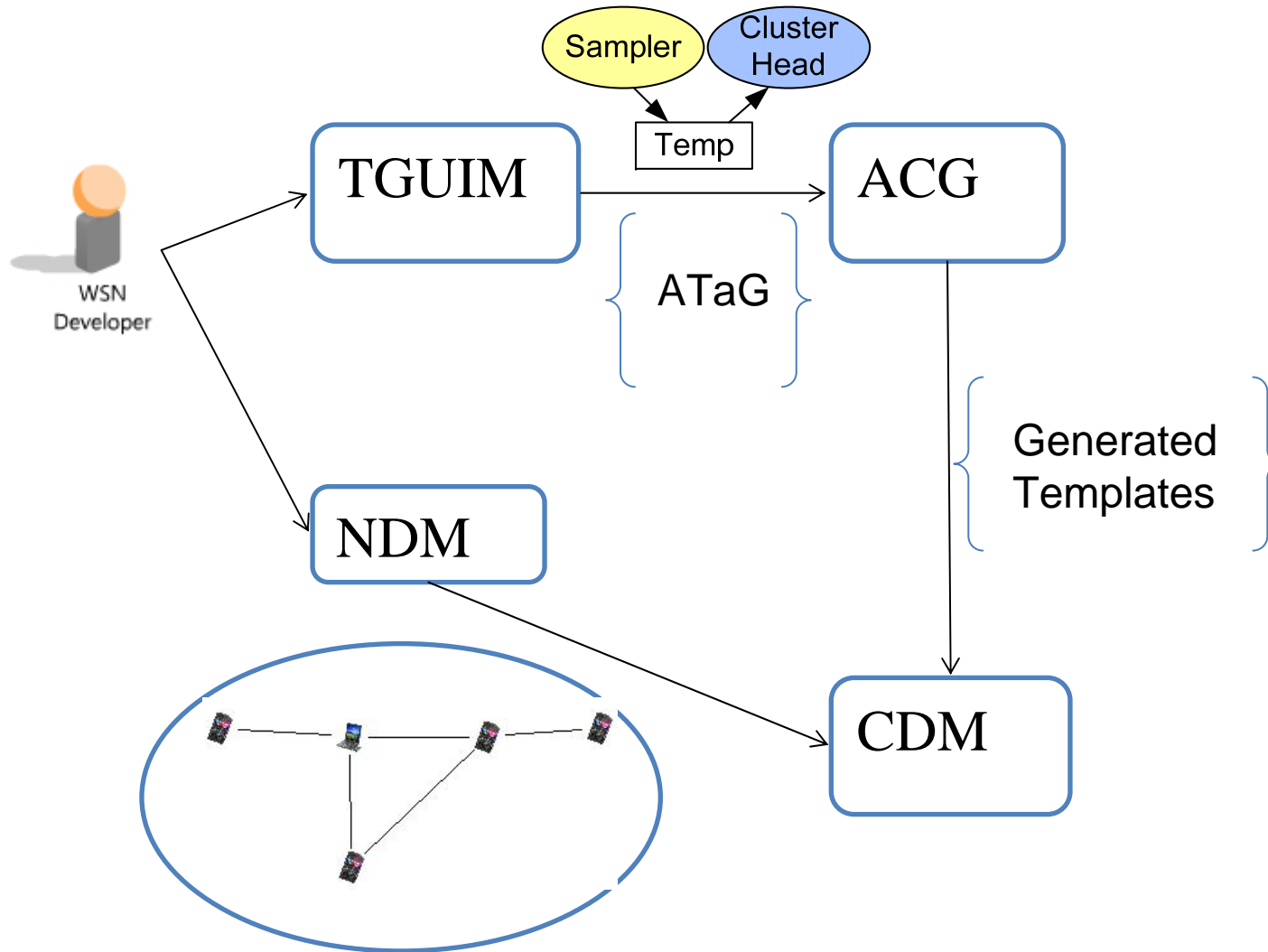
# Srijan components

- Network description module(NDM)
- Task Graph User Interface module(TGUIM)
- Auto Code Generator(ACG)
- Compilation – Deployment module(CDM)





# Srijan flow process



# Network description module

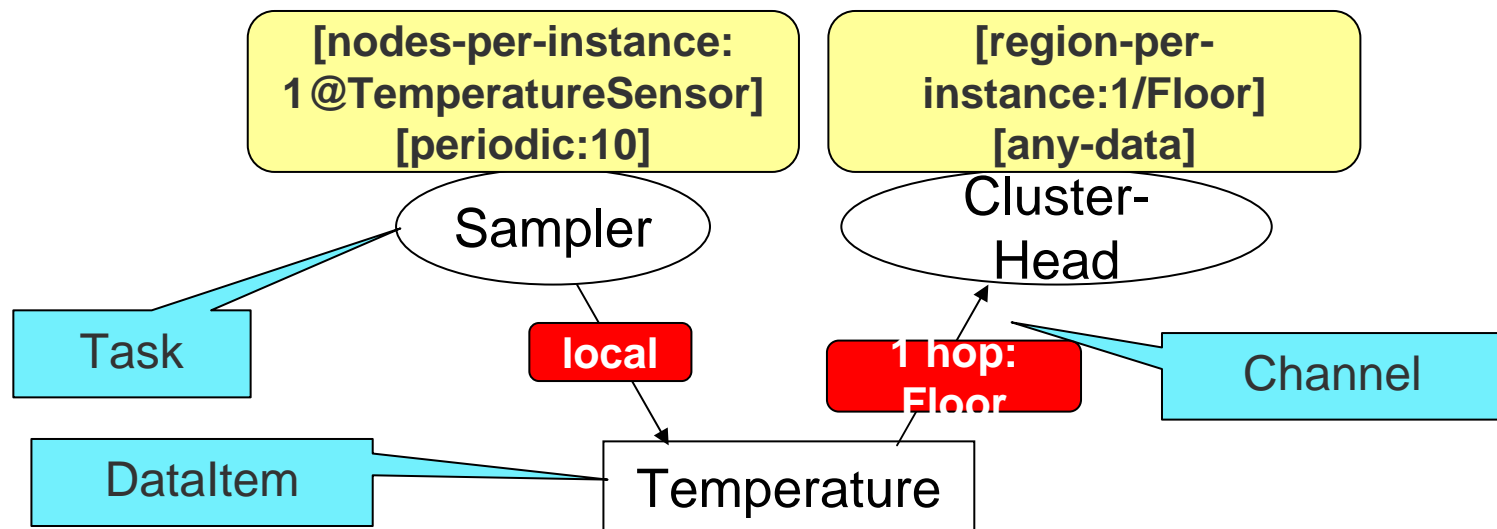
- Responsible for the definition of the network topology
- Addressing details and capabilities for each node
- Example in a simple text file

```
0014.4F01.0000.0642 room:0 attachedSensors:TemperatureSensor freespot  
0014.4F01.0000.02E0 room:0 attachedActuators:TemperatureActuator freespot
```



# Task Graph User Interface module(1)

- ATaG : Abstract Task Graph
  - Tasks interact with each other by producing/consuming Data Items
  - Annotations define the rules for placement, firing and interest



# Task Graph User Interface module(2)

- Firing Rules
  - `periodic`: `<period>` – fires periodically
  - `anydata` – fires when any incoming data item arrives
  - `alldata` – fires only when all incoming data items arrive



# Task Graph User Interface module(3)

- Instantiation Rules
  - `nodes-per-instance : <num> [ @NodeType ]`
    - one copy of the task for every `<num>` nodes [of type `NodeType`]
    - “One sampling task on each node with a temperature sensor”
  - `region-per-instance : <num> / RegionLabel`
    - `<num>` copies of the task in each set of nodes with the same `RegionLabel`
    - “One collecting task per Floor”



# Auto code Generator

- For each different type of nodes *Srijan* will produce code templates depending on the task graph.
- The developer writes the bussiness model of the application on these templates.
- Finally for each node of the WSN *Srijan* will produce and deploy code segments.



# Compilation – Deployment module

- The developer opens the imperative code templates and fills them in with specific code details.
- *Srijan* creates the final code for each node.
- Compiles the code.
- Deploy on nodes.



# Implementation

- *Srijan* is implemented in Java.
- A plugin for Eclipse was created to allow the user specify the ATaG and then to load the *Srijan* toolkit.
- The ATaG was created with the EMF plugin of Eclipse, which lets you define metamodels.
- The metamodel was created with GEMS





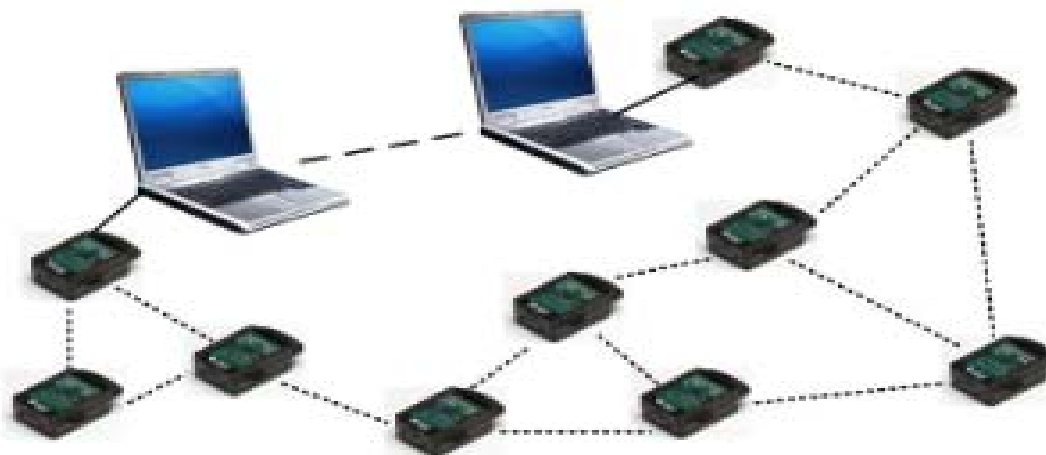
# Testing nodes(1)

- Sun SPOTs
  - 32bit ARM9 compliant microprocessor
  - Squawk virtual machine
  - IEEE 802.15.4
  - TI CC240 radio
  - Accelerometer , temperature and light sensor
  - Deployment over the air through gateways



# Testing nodes(2)

- Any pc with Java Runtime Environment
- Sun SPOT hosts , acting as gateways between PCs and nodes.



# Web

- <https://gforge.inria.fr/projects/srijan/>
  - Bug tracker
  - Forum
  - SVN
  - Task list
  - Wiki
  - Documentation



# Eclipse ATaG plugin

ATaG plugin  
button

Java - test/src/ATAG1.atg - Eclipse SDK

File Edit View Navigate Search Project Run Srijan Window Help

start client start server

Package Explorer Hierarchy \*ATAG1.atg

> ATAG123 [https://scm.gforge.inria.fr/svn/srijan, True]  
test  
src  
(default package)  
ATAG1.atg  
ATAG1.atg.gemx  
JRE System Library [JavaSE-1.6]

TempCollector [Task]  
name [OutputChannel]  
TempCollector [Task]  
TempCollector [InputChannel]  
name [DataItem]

Palette  
Select  
Marquee  
Connection  
Components  
OutputCha...  
Task  
ATAGPlan  
InputChan...  
DataItem

Outline

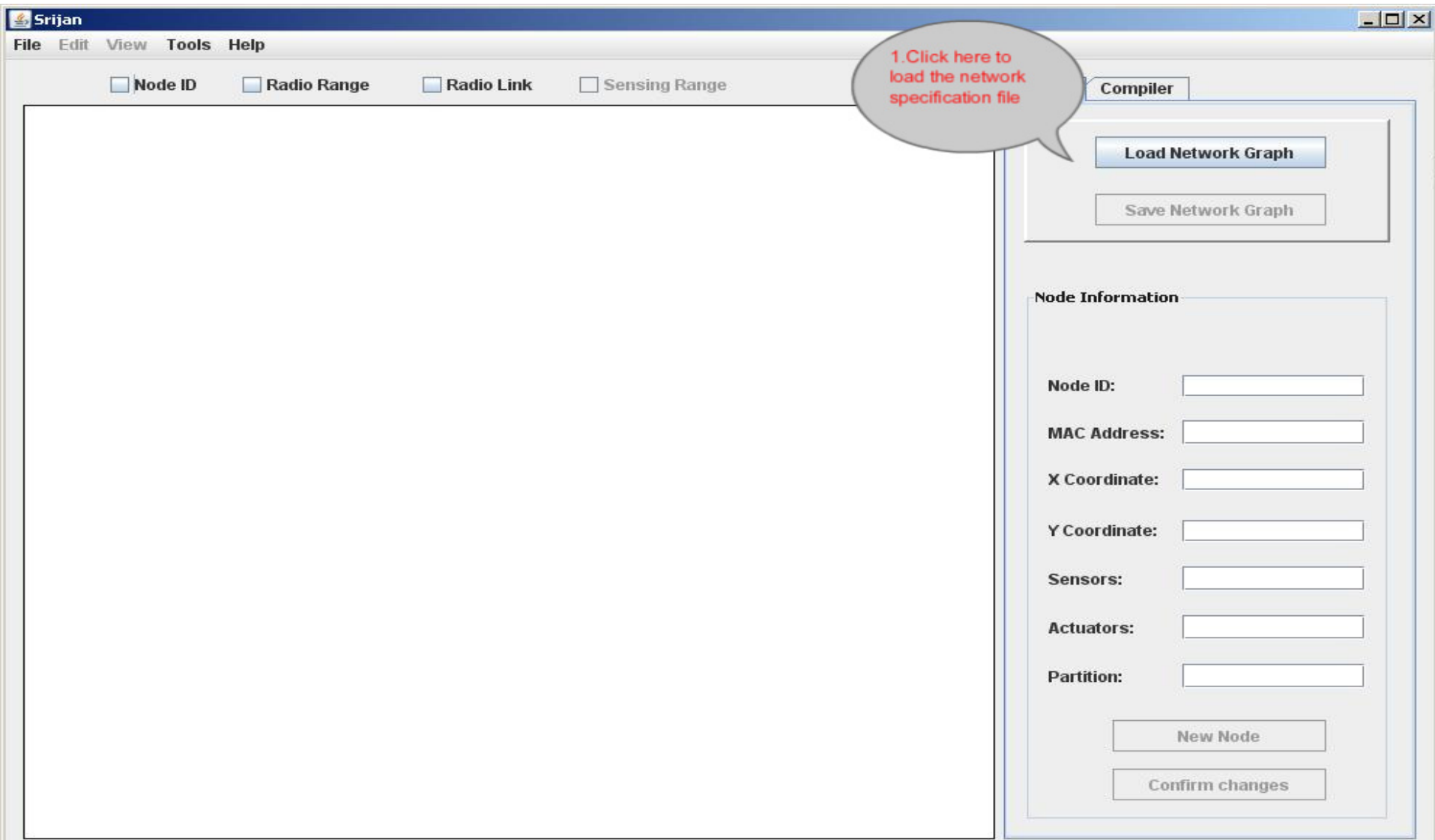
Problems Javadoc Declaration Progress Error Log Properties

Property	Value
Attributes	
Annotation	
External Link Target	
Name	name

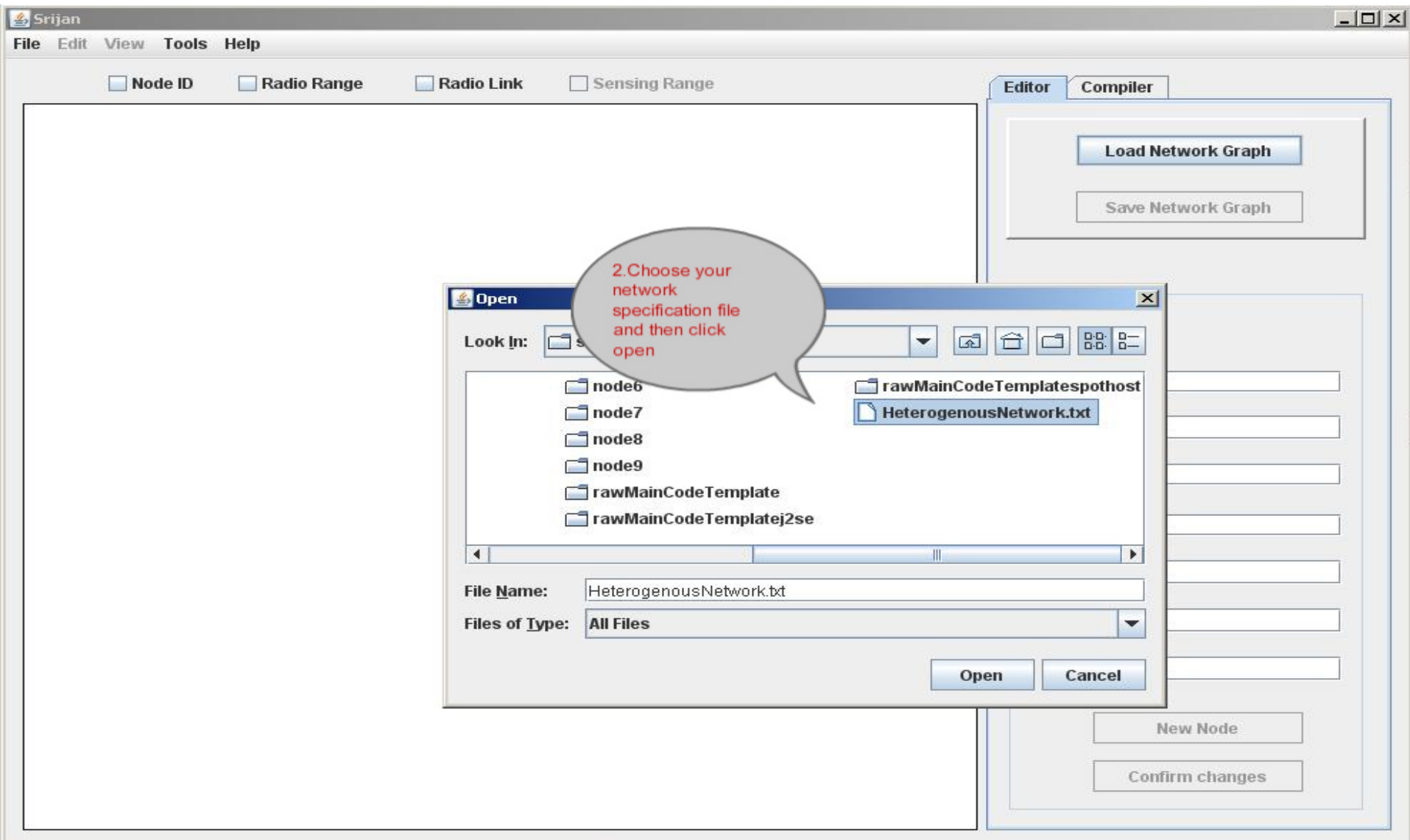
ATaG Graph

Properties view.

# Screenshots(1)



# Screenshots(2)



# Screenshots(3)

The screenshot displays the Srijan network simulation software interface. The main window shows a network topology with five nodes (represented by mobile phone icons) connected by lines. A green callout box points to the topology with the text "Network topology". The interface includes a menu bar (File, Edit, View, Tools, Help) and a toolbar with checkboxes for "Node ID", "Radio Range", "Radio Link" (checked), and "Sensing Range". On the right, there is a panel with "Editor" and "Compiler" tabs. The "Editor" tab contains buttons for "Load Network Graph" and "Save Network Graph". Below these is a "Node Information" section with fields for Node ID (3), MAC Address (0014.4F01.0000.1462), X Coordinate (311), Y Coordinate (107), Sensors (none), Actuators (none), and Partition (1). At the bottom of this panel are buttons for "New Node" and "Confirm changes".

File Edit View Tools Help

☐ Node ID ☐ Radio Range ☒ Radio Link ☐ Sensing Range

Network topology

Editor Compiler

Load Network Graph

Save Network Graph

Node Information

Node ID: 3

MAC Address: 0014.4F01.0000.1462

X Coordinate: 311

Y Coordinate: 107

Sensors: none

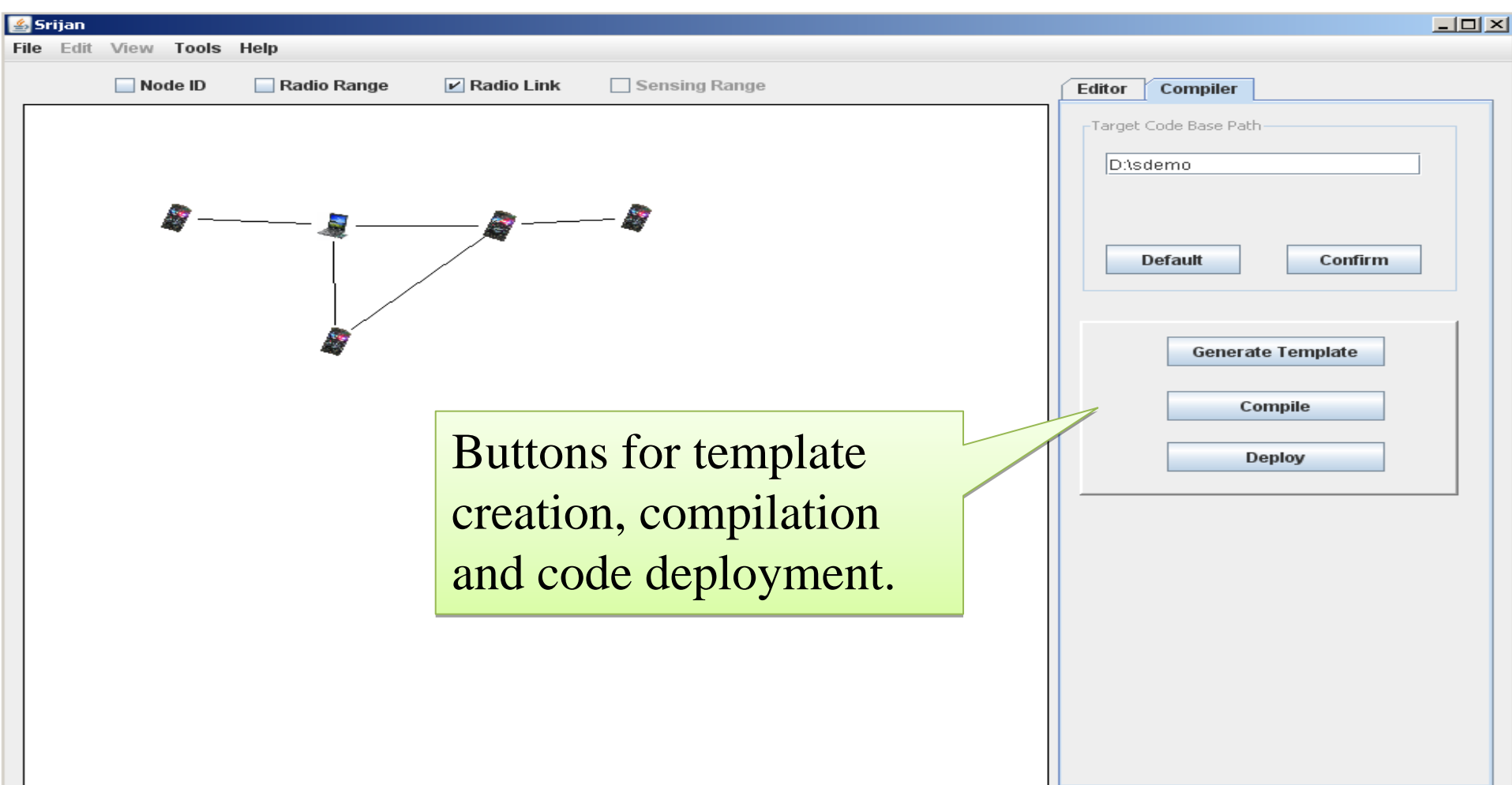
Actuators: none

Partition: 1

New Node

Confirm changes

# Screenshots(4)





# Future Work

- Extend *Srijan* to support to more resource constrained nodes :
  - Contiki / TinyOS nodes
  - RFID cards
  - Android mobile devices
- Supporting dynamic configuration and re-configuration of the pervasive system



Merci de votre attention

Questions?

