BEMO-COFRA
Brazil-Europe Monitoring and Control Framework

(Project No. 288133)

D5.2.1 Initial Administration Tool

Published by the BEMO-COFRA Consortium

Dissemination Level: Public

Document control page

Document file: D5.2.1 Initial Administration Tool v1_0.docx
Document version: 1.0
Document owner: Markus Taumberger (VTT)

Work package: WP5 – Distributed Control Logic and Enabling Features
Task: T5.2 – Monitoring and control of WSANs
Deliverable type: P

Document status: ☐ approved by the document owner for internal review
                        X approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of Changes made
0.1	Markus Taumberger	2012-10-16	Initial version
0.2	Mikko Ala-Louko	2012-11-26	Implementation section added
0.3	Markus Taumberger	2012-11-27	Architecture added
1.0	Markus Taumberger	2012-11-28	Updated according review comments

Internal review history:

Reviewed by	Date	Summary of comments
Juliana Vilela	2012-11-28	

Index:

1.   Executive summary

This document describes the initial administration tool prototype for the BEMO-COFRA platform. The current prototype is mainly targeted for the M12 demo and supports the visualization of BEMO-COFRA resources in real-time together with additional information. In D5.2.2 it will be extended to support the administration protocol developed in WP4.

The selection of a suitable platform for the tool was supported by a platform survey. Another survey was carried out to study relevant existing network monitoring solutions.

The tool interfaces with the BEMO-COFRA environment through the event manager of the underlying LinkSmart platform. In the visualization, active components are highlighted in green when working within expected parameters and red when an error occurs.

2. Introduction

This document specifies the initial administration and monitoring tool for the BEMO-COFRA system. Chapter 3 presents the background for the work: the requirements, a platform survey and a brief summary of the state-of-the-art of monitoring tools for wireless sensor networks. Chapter 4 presents the implementation of the first version of the tool and describes how it is used in the M12 demo. The final chapter 5 discusses planned future extensions of the administration tool.
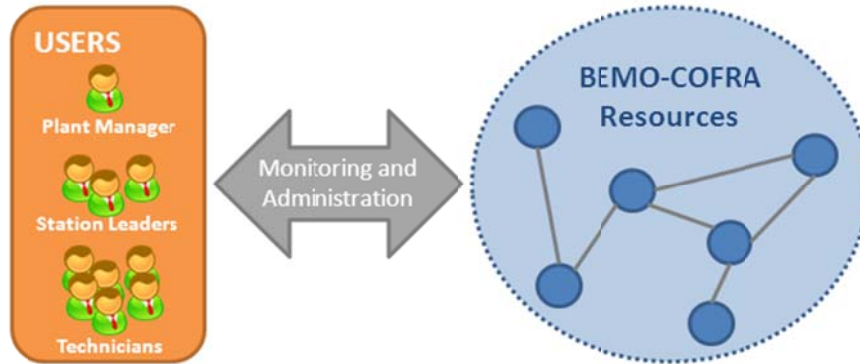
Figure 1: Stakeholders running the BEMO-COFRA system

There are different stakeholders, depicted in *Figure 1*, that need to monitor and administer different parameters of a BEMO-COFRA environment. The ultimate goal of the administration tool is to serve all those users with their various demands and to present only relevant data in their view.

3.   Specification

3.1   Requirements

The requirements identified for the tool are as follows:

- Visualization of the high level state of the system [fulfilled]
- Visualization of network management and resource monitoring data of the WSAN [open]
- Enable notifications to the user [partly fulfilled]
- Allow setting of WSAN parameters [open]
- Running on a portable device (Mobile phone / Tablet) [fulfilled]

3.2   Baseline

3.2.1   Survey of available platforms

In this survey existing platforms have been investigated that could be used for the implementation of the administration tool.

Android

This Linux-based open-source operating system for ARM architectures is the current market leader in mobile platforms. Since Android 4.0 the same version can be used for mobile phones and tablets. Applications are primarily written in a customized version of Java, the SDK is available for Windows, Mac OS X and Linux. [Android]

Android is widely spread, devices exist from many different hardware vendors and offers SDKs for several operating systems.

iOS

The operating system of iPhone and iPad from Apple Inc. is based on OS X and since iOS 4 also multitasking is supported through background APIs [iOS]. Its multi-touch enabling interface features together with the world's largest selection of applications in the App Store helped making iOS devices a huge success.

iOS is licensed only for Apple hardware, the SDK (Xcode) requires Apple devices to develop iOS applications.

Windows Phone

Microsoft's current mobile operating system features the Metro interface and is primarily targeted for smartphones. The most important partner using the Windows Phone operating system for their devices is Nokia. Windows Phone Marketplace offers the infrastructure to purchase applications.

Apps can be developed with Visual Studio 2010 or Visual Studio 2010 Express editions, but have to be programmed with XNA or a WP7 specific version of Silverlight. Application development in Visual Basic is also supported with a software development kit [WP SDK].

The future of Windows Phone is unclear with respect to its successor Windows 8, it is also limited to mobile phones and even there not widely used.

Windows 8 / Windows RT

The new Windows 8 version from Microsoft supports the IA-32, x86-64, and ARM (Windows RT) processor architectures [WIN 8]. As Windows Phone it features the Metro user interface for touchscreen devices. At the moment the Developer Preview and the Customer Preview are available, both expiring in January 2013.

While Windows 8 is targeted to span across whole product categories from desktop PCs to mobile phones, allowing applications to run on a large variety of devices, its release date has not been officially announced, yet. Also the timely availability of Windows 8 devices is not clear right now.

3.2.2   Existing WSN Monitoring tools

In this section a variety of tools available for monitoring wireless sensor networks are introduced to set the baseline for the administration tool.

Z-monitor [ZMonitor]

- Frame Decoding: it shows the details about received frames that pertain to the Physical and MAC layers with respect to the IEEE 802.15.4 standard specification. Two different GUIs were designed for Frame Decoding operation offering more flexibility to users for protocol analysis.

- Topology Visualization: it shows the logical topology of the network (parent-to-child relationship between nodes).

- Traffic Timeline: it shows the sequence of packet received over time.

- Packet Statistics: it provides useful statistics about number and type of packets received by the sniffer. Statistics can be saved in a file and analysed off-line.

- Star-based topology: the current version supports star-based topology as it is related to the Physical and MAC Layers of the IEEE 802.15.4 standard protocol. Multi-hop support is planned for next releases with support of ZigBee and 6LowPAN Network Layer protocols.

- This tool only supports TinyOS, e.g. Contiki support must be implemented from scratch

6PANview [6PANview]

- Supports standard administration tools such as ping6, traceroute6, and SNMP

- Includes a light-weight SNMPv1 agent for the network nodes

- Includes a GUI with various capabilities such as one-click health check for nodes, etc.

- Supports Push and Pulls modes of SNMP operation

- Has flexible configuration options such as : Various monitoring parameters, Poll intervals, Thresholds and Alerts

Nagios [Nagios]

- Browser based (there are alternative native interfaces for windows, osx, linux, mobile)

- Has most features than other framework, but ugly user interface

- Nagios plugins are standalone extensions to Nagios Core that provide low-level intelligence on how to monitor anything and everything with Nagios Core

- The architecture: http://nagios.sourceforge.net/docs/3_0/plugins.html

- Plugins are compiled executables or scripts (Perl scripts, shell scripts, etc.) that can be run from a command line to check the status or a host or service. Nagios uses the results from plugins to determine the current status of hosts and services on your network.

Mango [Mango]

- Browser based

- Mango can receive data from any device for which there is a protocol driver. Currently supported protocols include BACnet I/P, Modbus (ASCII, RTU, TCP, and UDP), OPC DA, 1-wire, SNMP, SQL, HTTP, POP3, NMEA 0183, MBus, DNP3, OpenV, webcams, vmstat, and many proprietary protocols developed by or for hardware vendors.

- User interface is simple. Only an html table.

Zenoss [Zenoss]

- Browser based

- Can be extended for monitoring any protocol: http://community.zenoss.org/community/documentation/official_documentation/zenoss-extended-monitoring/3.0-v04

- Based on Zope (PHP & Ajax)

- Android app available (possibly connected via REST): http://www.android-zenoss.info/

Hyperic HQ [HypericHQ]

- vSphere auto-discovery of all components of virtualized applications
- Automatically discovers, monitors, and manages software and network resources
- Monitors apps on any platform, including Unix, Linux, Windows, Solaris, AIX, HPUX, VMware, and Amazon Web Services
- Built-in support for 75 common components---including databases, application servers, middleware, web servers, network devices and more
- Guide for plugins: http://support.hyperic.com/display/DOC/Plugin+Development+Center#PluginDevelopmentCenter-overview

3.2.3   Conclusions

Based on the existing work, we conclude that we require a possibly simplified browser-based interface as it is widely accepted format used by the existing monitoring tools. In addition, to present e.g. a 3D representation of the high-level system state, standalone version must be supported on the main mobile and desktop platforms. This requirement suggests the usage of a platform-independent framework that introduces minimal code porting overhead.

In order to achieve a maximum of supported platforms the administration tool is implemented with Unity3D and can be run on Android, iOS, Windows and as web application.

The actual monitoring protocol operating on the wireless sensor network will be designed in WP4 and the supporting features of the tool will be defined in the final version. In this initial prototype the main focus is on supporting the M12 demo and defining the technical basis for the implementation of the final version.

3.3   Architecture

The administration tool consists of three elements, the *Web Service Client*, the *Event Processing* and the *GUI Engine*, see *Figure 2*.

Figure 2: Administration tool architecture

The *Web Service Client* is used to connect with the BEMO-COFRA environment through the Event Manager of the LinkSmart platform. It subscribes to the events of interest and passes them on to the *Event Processing*.

The *Event Processing* unit creates a new object instance to represent new resources, if an event comes from a resource that is not known by the administration tool yet. In all other cases the current state of the resource representation is updated according the received events.

Finally the visualization of the BEMO-COFRA environment is generated by the *GUI Engine* that is responsible for the interaction with the user.

4.   Implementation

4.1   Overview

The application was implemented using the Unity3D cross-platform game engine [Unity3D]. For scripting, Unity3D uses Mono [Mono], an open source implementation of the .NET framework. Mono has been ported to several operating systems such as Windows, Linux, Android, OS X etc. In addition Unity3D features a Web Player that can be used to run the application inside a web browser. The usage of Mono ensures that the tool is portable with little effort to the aforementioned platforms and more. During the M12 demo at the review in Bonn, the tool was shown running simultaneously on a Windows PC and an Android tablet

The tool communicates with the rest of the BEMO-COFRA components by utilizing the LinkSmart eventing system that allows the tool to subscribe to events of interest and publish administrative events towards the other components. For this purpose, it uses Mono's web service capabilities. The web service –based connectivity also provides remote-monitoring; multiple instances of the application can be run remotely and connect to the BEMO-COFRA installation via the appropriate LinkSmart Event Manager.

4.2   Graphical User Interface

The Graphical User Interface (GUI) of the tool is shown in *Figure 3*. It visualizes the monitoring information gathered from the BEMO-COFRA installation in a continuous-time 3D representation. The screenshot features the main objects used in the M12 demo: the car body, welding robots, and the grippers. Detailed information about each object is available when clicking or touching them on the screen. Active objects are rendered in green colour whereas idle objects are rendered in grey and objects that are in erroneous state are rendered in red.
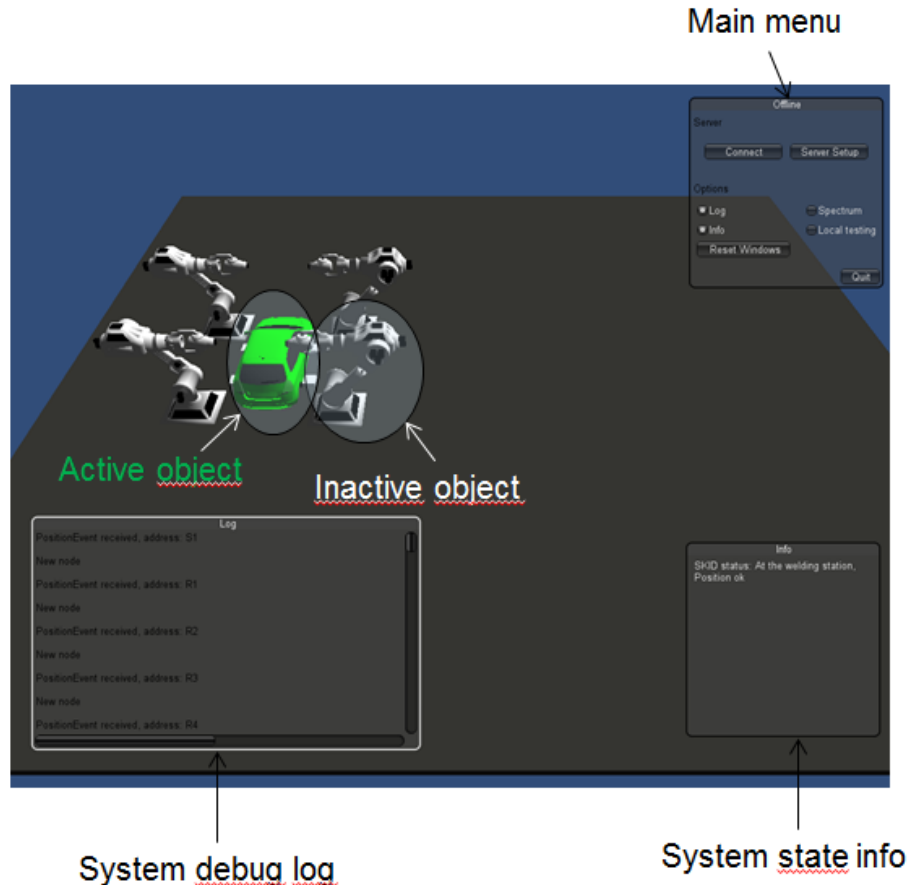

Figure 3: GUI implementation on Unity3D

The menus feature a server setup, where one can define the Uniform Resource Identifier (URI) of the LinkSmart Event Manager to connect to, and several subwindow toggles. In the screenshot, a debug log window, showing all the received and sent events is shown in the lower left corner. This information is mainly targeted for a technician providing means for detailed troubleshooting. The lower right corner features a system state info window that summarizes the high-level state of the system (car body location, robots and grippers active / idle etc.). Finally, the GUI allows for the plotting of the channel occupancy of all the 16 IEEE 802.15.4 channels. This information is received from wireless sensor motes and is part of the WP4 work.

4.3   Event specifications

Table 1 lists the events that were subscribed to in the M12 demo by the tool. In addition the table specifies the actions that the tool will take upon the reception of each event.

Table 1: List of subscribed events and corresponding actions

Name	Description	Action
PowerConsumption	The power consumption of the last welding cycle in Wh.	Value shown in the welding robot info window.
bemocofra/event/WSN	Spectrum occupancy values and car body acceleration from the wireless sensors.	Occupancy plotted in spectrum window. Acceleration shown in the car body info window.
SkidWeldingPosition	Car body in the welding station.	Car body model moved to the welding station position and stopped.
MoveSkidBack	Car body leaves the welding station.	Car body model starts to move towards the start position. Car body object state is set to idle.
SkidStartPosition	Car body at the starting position.	Car body model moved to the starting position and stopped.
MoveSkid	Car body leaves the starting position.	Car body model starts to move towards the welding station position.
SkidWeldingPositionOk	Car body position correct at the welding station.	Car body object state is set to active if the car is at the welding station.
SkidWeldingPositionError	Car body position error at the welding station.	Car body object state is set to error if the car is at the welding station.
ConfirmClamps	Grippers closed.	Gripper objects' states are set to active.
ConfirmClampsOpen	Grippers open.	Gripper objects' states are set to idle.
ClampsError	Grippers in error state.	Gripper objects' states are set to error.
StartWelding	Welding robots start welding.	Welding robot objects' states are set to active.
StopWelding	Welding robots stop welding.	Welding robot objects' states are set to idle.

5. Future work

Upcoming work for the administration tool will focus on two aspects, which are (1) enabling viewing of information at different levels of detail for different users and (2) integrating the monitoring and administration protocol into the tool. This will allow providing the relevant stakeholders with additional in-depth data of the Wireless Sensor and Actuator Network components of the BEMO-COFRA environment.

References

(6PANview)   http://ece.iisc.ernet.in/6panview/

(Android)   Android homepage, http://www.android.com/

(HypericHQ)   http://sourceforge.net/projects/hyperic-hq/

(iOS)   Develop for iOS, https://developer.apple.com/technologies/ios/

(Mango)   http://mango.serotoninsoftware.com

(Mono)   http://www.mono-project.com

(Nagios)   http://www.nagios.org/projects/nagioscore

(Unity3D)   http://www.unity3d.com

(WIN 8)   Wikipedia article on Windows 8, http://en.wikipedia.org/wiki/Windows_8

(WP SDK)   Windows Phone SDK, http://create.msdn.com/en-us/home/getting_started

(Zenoss)   http://www.zenoss.com/

(ZMonitor)   http://www.z-monitor.org