INTERACT – Interactive Manual Assembly Operations for the Human-Centered Workplaces of the Future

Grant Agreement Number	**:**	**611007**
	:	**INTERACT**
Project Start Date	**:**	**1st October, 2013**
Consortium	**:**	**DAIMLER AG (DAIMLER)- Project Coordinator** **ELECTROLUX ITALIA S.P.A. (ELECTROLUX)** **INTRASOFT INTERNATIONAL SA (INTRASOFT)** **IMK AUTOMOTIVE GMBH (IMK)** **EMPHASIS TELEMATICS AE (EMPHASIS)** **HADATAP SP ZOO (HADATAP)** **UNIVERSITY OF PATRAS (LMS)** **UNIVERSITAET ULM (IMI)** **DEUTSCHES FORSCHUNGSZENTRUM FUER KUENSTLICHE INTELLIGENZ GMBH (DFKI)**

Title	**:**	**Deliverable 3.3.2 -** ***"Sensor management platform"***
Reference	**:**	**D3.3.2 of Task 3.3, WP3**
Availability	**:**	**Public**
Date	**:**	**2015-03-31**
Author/s	**:**	**HADATAP**
Circulation	**:**	**Consortium, Public**

Summary:

Deliverable D3.3.2 created in Task 3.3 – *"Sensor management platform"*

This is the platform that will enable the management of the sensor systems (i.e. add/remove/combine sensor sources). It is the administration component of the integrated INTERACT sensor system and it is delivered as partial output of Task 3.3 by M18.

Table of contents

1. INTRODUCTION

One of the main components of the Interactive Manual Assembly Operations for the Human-Centered Workplaces of the Future (INTRACT) platform is sensors system which core purpose is the most accurate and automated recording and gathering data about operations performed by employees during manual assembly verification workshops. It could be achieved by platform of a set of various sensors, including optical, MEMS and tool sensors.

The Sensor Management Platform (SMP) is the administration component of the system, that allows managing sensors in order to provide full capabilities.

This document presents SMP architecture, design and result of its development. It includes software components that form the platform.

Further information provided as follows:

- Section 2 provides a general description of the platform, identifies its main components and present dependencies between the platform and INTERACT platform work packages. It is divided into the following parts:
  - Overall description of the platform,
  - Precise description of design, architecture and development results,
  - Overall description of closely related parts of INTERACT platform,
  - List of platforms' available use cases.
- Section 3 describes communication interfaces to:
  - Sensor sub-networks,
  - INTERACT platform applications,
  - INTERACT Data Access Layer.
- Section 4 provides conclusions on the platform architecture, design and development results.

The platform impacts not only WP3 but also WP5. Some information in this deliverable could be find as redundant in terms of WP3 but has been included to help understanding the platform as a whole.

2. INTERACT Sensor Management Platform

2.1. Overall platform description

The Sensor Management Platform (SMP) is the administration component of integrated INTERACT sensors system.

Its main responsibility is to enable sensor networks management and configuration by providing sets of methods. It gathers all sensor networks logic and provides an easy, unified way of usage so that other INTERACT platform parts can focus on "what" instead of "how" to get interesting data from sensors. Using the platform, a user will be able to prepare a workstation and sensors for proper data capturing and provide the way to conduct a capture sessions.

The module implements following functionalities:

- Discovery of available, online sensors,
- Data capture (start, stop, retrieve, delete),
- Sensors configuration and calibration,
- Sensors raw data monitor,
- Save/load platform/sensor configuration.

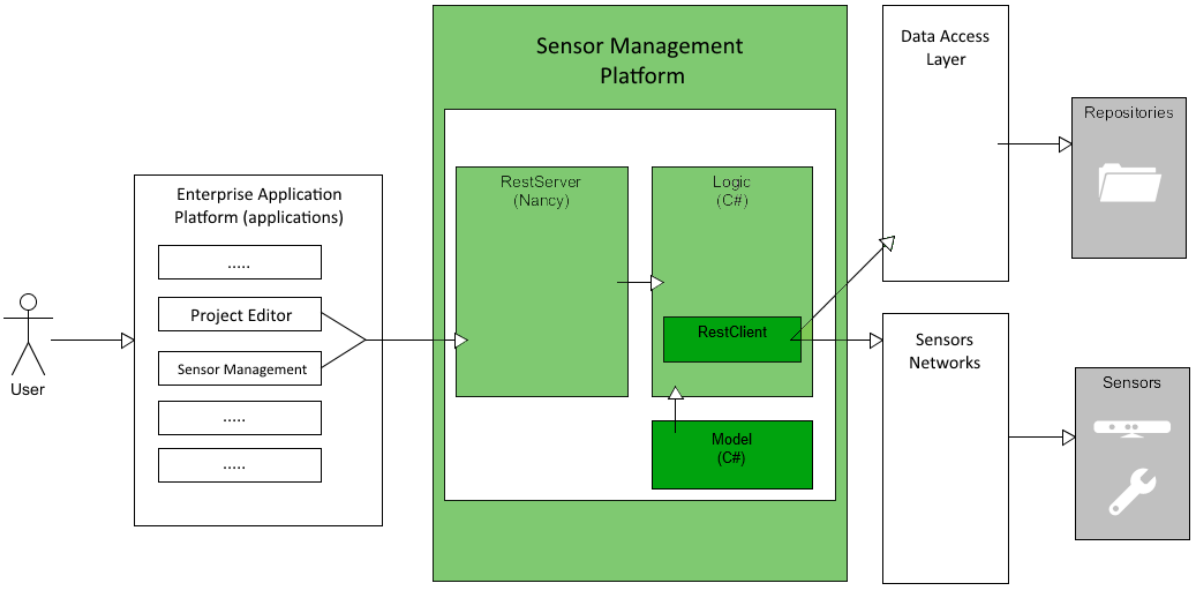

Figure 1: Sensor Management Platform architecture along with interaction with other INTERACT platform parts

The platform is built from two main parts – the Rest Server and the Logic.

The Rest Server provides REST interfaces that deal with requests from other system parts – mainly Enterprise Application Platform applications – Sensor Network Management application and Project Editor application.

The Logic performs heavy work regarding communication, embracing the logic of actions requests and unifying data received from sensor sub-networks. One of the most important logic part is Rest Client that allows application to communicate with DAL and sensors networks – and through them also with data repositories and sensors.

2.2. Implementation

The platform has been implemented using C# programming language. It consists of Rest Server part developed the use of NancyFx [1] and logic part with its model. An overview of the platform parts are shown in Figure 2.

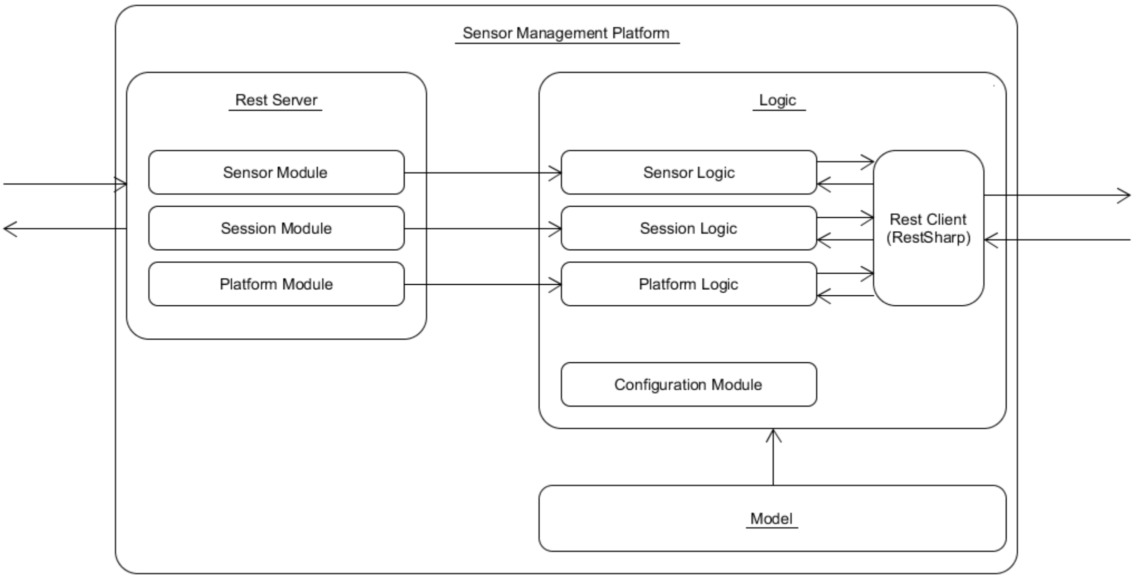

Figure 2: Overview of the platform parts

2.2.1. Rest server

The Rest server has been implemented with usage of NancyFx which is a lightweight, low-ceremony (“Nancy code” needed in application is minimal) framework for building HTTP based services on .NET and Mono. It is licensed under the MIT license [2].

Server is logically divided into three main modules:

- Sensor (supports sensors discovery, configuration, calibration, raw data monitoring),

- Session (supports data capture),
- Platform (supports save/load configuration requests).

The communication between server and other parts of INTERACT platform is enabled through HTTP REST interface. All input and output data is described using JSON notation [3]. Exceptions are handled through HTTP error codes.

Example endpoint in NancyFx notation:

```
Get["/stop"] = parameters =>
{
        ResponseMessage result = _sessionRepository.StopSession();
        string jsonResp = JsonConvert.SerializeObject(result);
        return jsonResp;
};
```

Figure 3: Example NancyFx endpoint code

2.2.2. Logic

Logic with its model has been implemented using C#. It consists of few modules.

Each of service modules has its dedicated logic part which performs all work regarding supported part of sensors management functionalities, and therefore, as in the case of the Rest Server:

- Sensor module (supports sensors discovery, configuration, calibration, raw data monitoring),
- Session module (supports data capture),
- Platform module (supports save/load configuration requests).

One of the most important modules is Rest Client implemented using RestSharp [4] which is simple REST and HTTP API Client for .NET. It is licensed under Apache 2.0 licence [5].

Platform configuration was put in a dedicated module. It provides all platform wide configuration parameters such as, for example, IP of sensor networks' interfaces.

2.2.3. Model

Logic part is strictly related with Model part which supplies module with objects data models.

2.3. Short description of closely related INTERACT platform components

The platform is closely related to few modules of INTERACT platform. Most of them are not a subject of WP3 but here they are briefly described for better understanding of SMP.

2.3.1. Sensor Networks

Sensor Management Platform main responsibility is to manage sensor networks. There are three dedicated sensor network within INTERACT platform:

- Optical Sensor Networks (supplies information gathered from optical sensors – Kinect v2 depth cameras – more information can be found in D3.1.1, D3.2.1 and D3.2.2),
- Wireless Sensor Networks (supplies information gathered from grasp, foot and IMU sensors – more information can be found in D3.1.1, D3.2.1 and D3.2.2),
- Tool Sensor Network (supplies information gathered from sensors categorised as tool sensors, for example screwdriver – more information can be fount in D3.1.1, D3.2.1 and D3.2.2).

Communication with all networks is enabled through HTTP REST interfaces.

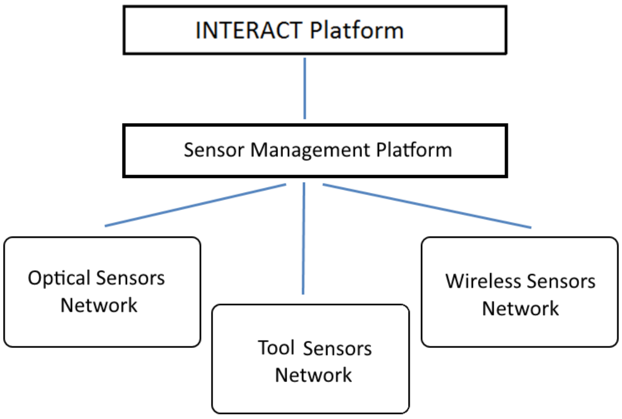

Figure 4: Sensor networks and Sensor Management Platform

2.3.2. Sensor Network Management Application

Sensor Network Management Application (SNMA) is the front-end part of INTERACT platform. It is very closely related to SMP. Main responsibilities of the application are to provide to end-user a GUI that allows preparing data capturing session along with sensors management, configuration, calibration and monitoring. SNMA's communication with SMP is enabled through a HTTP REST interface.

More specific information about application will be provided in D5.3.1.

2.3.3. Project Editor

Project Editor application is the front-end part of INTERACT platform. Its main responsibilities, in context of SMP, are to provide end-user simple, intuitive GUI that allow management of data capture during a workshop. Project Editor's communication with SMP is enabled through a HTTP REST interface.

More specific information about application will be provided in D5.3.1.

2.3.4. Data Access Layer

Data Access Layer (DAL) is one of the Enterprise Application Platform components. It consists of modules that allows to store and access data. SMP communicates with DAL through HTTP REST interface.

More information can be found in D5.2.1. and will be provided in D5.3.1.

3. Platform Communication interfaces

Sensor Management Platform provides a server-side web API in order to build a request-response message system that is used for interaction with other parts of INTERACT Platform.

SMP is interfaced to

- Sensor networks (Optical Sensor Network, Wireless Sensor Network and Tool Sensor Network),
- EAP applications (Sensor Network Management Application, Project Editor),
- Data Access Layer (data storage and access).

All communication in platform is based on HTTP based REST services which send and receive data in JSON format.

Exceptions are handled through HTTP Error Codes, for example:

- 200 OK,
- 500 Internal Server Error.

3.1. Platform endpoints

Communication is implemented as HTTP GET/POST requests.

Syntax:

```
http://<URL>:<PORT>/sensor/add/{sensorId}/{sessionId}
```

Category	Request	Result	Functionality
Sensor	GET /sensor/discover/	[{ Id: “0”, DevId: “01011”, Name: “Screwdriver 01001”, Type: “Screwdriver”, Address: “COM3” }, ...]	Raw data monitor
	GET /sensor/config/{id}/{type}/{paramName}/{paramValue}	---	Sensors' configuration

	GET /sensor/calibrate/ {id}/{type}/{calType}	---	Sensors' calibration
	GET /sensor/data/{id}/ {type}	Each sensor type provides different data format – all are described in D3.3.1.	Raw data monitor
	GET /sensor/stopdata/{id}/{type}	---	Raw data monitor
	POST /sensor/saveconfig/ Data: <sensorParams>	---	Save/load configuration
	GET /sensor/loadconfig/{id}	Each sensor type provides different data format.	Save/load configuration
Session	POST /session/start Data: <sessionParams>	7d147c2a-196f-405b-a7ce-c3413b693425 (SessionId)	Data capture
	GET /session/stop/{sessionId}	---	Data capture
	GET /session/data/ {sessionId}	Each sensor type provides different data format. All are described in D3.3.1.	Data capture
	GET /session/removedata	---	Data capture
Platform	POST /platform/saveconfig/ Data: <platformConfiguration>	None	Save/load configuration
	GET /platform/loadconfig/	<platformConfiguration>	Save/load configuration
	GET /platform/sync/	None	Time synchronization

Table 1: Sensor Management Platform endpoints

3.2. Interface to sensor systems

Each sensor network has its own communication interface. The differences are due to the way the methods are enabled but all of them provide the same functionality. Communication is based on HTTP based REST services which receive/send data in JSON format. Exceptions are always handled through HTTP error codes.

Further information can be found in D3.2.2.

3.3. Interface to data repositories (DAL)

Data Access Layer is interfaced to Sensor Management Platform to provide methods for data storage and retrieval. Endpoints presented below represents the current state of work on the subject. It may (or may not) be changed during future development progress.

Endpoint	Request parameter	Result
GET /platform/loadconfig/	PlatformConfigurationId	Described in 3.3.1.
POST /platform/saveconfig/	Described in 3.3.1.	---
GET /sensor/loadconfig/	SensorConfigurationId	Specific parameters list depends on sensor type. Overall description in 3.3.2.
POST /sensor/saveconfig/	Specific parameters list depends on sensor type. Overall description in 3.3.2.	---
GET /sensor/remove/	PlatformConfigurationId, SensorId	---
GET /sensor/add/	PlatformConfigurationId, SensorId	---

Table 2: Interface to Data Access Layer

3.3.1. Platform configuration data

Below is a list of platform configuration data which should be saved in data repository:

- Platform Configuration Id (PK)
- Scene Id (FK), file, version
- CNL task list Id (FK)
- Platform Configuration Name
- Platform Configuration Description
- List of selected sensors
- System-wide parameters including:
- List of triples: IMU Sensor (Id) + part (Id) + configuration (Id)
- Capture volume parameters (4 cords) + Optical sensors (Id) + scene (file, version) + configuration (Id)

3.3.2. Sensor configuration data

Below is a list of sensor configuration data which should be saved in data repository:

- Sensor Configuration Id (PK)
- Sensor Id (FK)
- List of pairs: Parameter Type (Id) + Parameter Value

3.4. Communication example (based on sensors discovery)

Example of communication within platform based on sensors discovery is depicted in Figure 2.

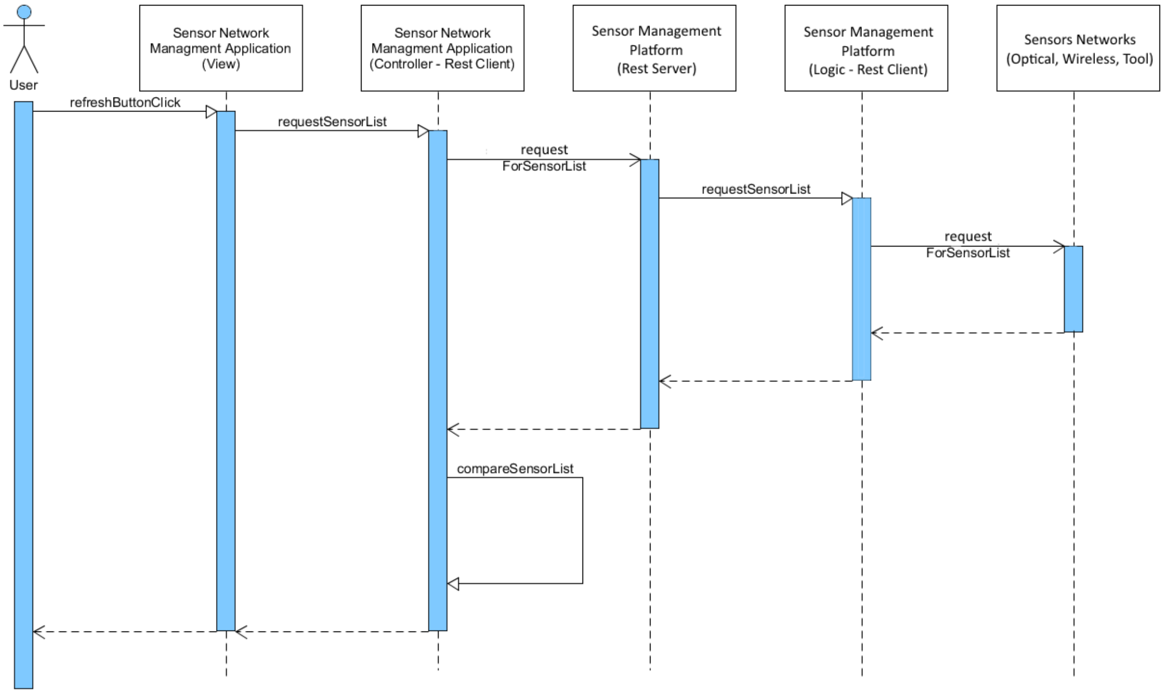
User
Sensor Network Managment Application (View)
Sensor Network Managment Application (Controller - Rest Client)
Sensor Management Platform (Rest Server)
Sensor Management Platform (Logic - Rest Client)
Sensors Networks (Optical, Wireless, Tool)
refreshButtonClick
requestSensorList
request ForSensorList
requestSensorList
request ForSensorList
compareSensorList

Figure 5: Platform communication schema (example)

4. CONCLUSIONS

In this deliverable, the full description of a final version for Sensor Management Platform of INTERACT project has been shown. The design and development follows the requirements presented in D3.1.1.:

- Platform implements all stated functionalities (sensors discovery, data capture, sensors configuration and calibration, data monitor, save/load sensors configuration),
- Platform enables sensors management and configuration in easy, unified way,
- No raw data captured from sensors is stored in the platform. Data acquired from sensor networks is immediately send to suitable INTERACT platform part and deleted from platform and sensor subsystems,
- REST approach allows for easy expansion of the platform capabilities,
- Set of HTTP REST endpoints enables communication with other INTERACT platform components – applications and DAL.

The integration tests showed that the platform cooperates well with sensor networks.

GLOSSARY

API	Application Programming Interface
DAL	Data Access Layer
EAP	Enterprise Application Platform
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
REST	Representational State Transfer
SMP	Sensor Management Platform
SNMA	Sensor Network Management Application

APPENDIX

Third party software used in the sensor management implementation:

Name	Version	Developer	Licence
NancyFx	0.21.1	Andreas Hakansson, Steven Robbins and contributors	MIT licence
Json.NET	6.0.7	James Newton-King	MIT licence
RestSharp	105.0.1	John Sheehan, RestSharp Community	Apache 2.0 Licence

REFERENCES

[1] NancyFx documentation: https://github.com/NancyFx/Nancy/wiki/Documentation

[2] MIT licence: http://opensource.org/licenses/MIT

[3] ECMA-404 The JSON Data Interchange Standard: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

[4] RestSharp documentation: https://github.com/restsharp/RestSharp/wiki

[5] Apache 2.0 Licence: https://www.apache.org/licenses/LICENSE-2.0