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 3.1.1 -** ***"Detailed sensors system architecture and design"***
Reference	**:**	**D3.1.1 of Task 3.1, WP3**
Availability	**:**	**Confidential**
Date	**:**	**2014-07-31**
Author/s	**:**	**EMPHASIS, IMI, HADATAP, LMS, DAIMLER**
Circulation	**:**	**Consortium**

Summary:

Deliverable D3.1.1 created in Task 3.1 – *"Detailed sensors system architecture and design"*

Contents

1. INTRODUCTION AND AIM OF THE DOCUMENT

One of the main components of the INTERACT platform is the sensor system that monitors human work tasks. As presented in document D1.2.1, several parameters that are related to the human body, objects, assets and tools that participate in a work task can be monitored in order to provide data input for the INTERACT motion synthesis and ergonomic analysis.

This document presents the sensor system architecture and design that has been developed in order to fulfill the INTERACT requirements. It includes both hardware and software components that form the sensor system.

The document is divided in 5 sections as follows:

- Section 2 describes in general the parameters that can be tracked in respect to humans, parts, assets and tools that participate in a human work task.
- Section 3 provides the architecture and design parameters for the INTERACT sensor system. It is divided in three parts which present:
  - the optical sensing subsystem that has been designed for capturing motions
  - the wireless sensor subsystem which provides additional information on motions and other parameters for which monitoring is required
  - the interfacing and data formats between sensor subsystems (optical and wireless sensor subsystem) with the Sensor Data Fusion module and other modules of the INTERACT platform
- Section 4 provides the description of the pilot sensor system implementations for the two industrial use cases, and how they address specific requirements
- Section 5 presents the INTERACT Sensor system setup that will be used in order to capture motion data during the capture of human motion samples as an input to the statistical analysis of the motion synthesis
- Finally, section 6 provides conclusions on the system architecture and design presented in the previous sections.

2. HUMAN WORK TASKS - TRACKABLE PARAMETERS

2.1. Body tracking

As it was already lined out in D1.2.1, different parameters of human body movements can be tracked within the workshop scenario. Within the INTERACT project, the chosen Rocketbox reference skeleton is the standard which determines the parameters that actually need to be tracked.

The RocketBox model consists of 25 bones for the main human body, plus 20 additional joints for the fully articulated hand skeleton (Figure 2.1).

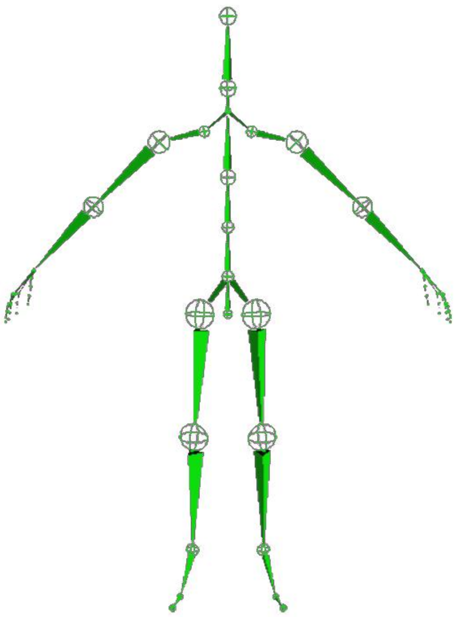

Figure 2.1: RocketBox skeletal structure

The 25 main bones need to be tracked even under difficult conditions, for example in occlusion situations or under poor lighting conditions.

The 20 additional hand bones need to be tracked only in certain situations, for example when grabbing or placing operations are carried out. However, according to the requirements definition in D1.2.1 and D1.1.1, it is not necessary to track a fully articulated hand model, as there are only 7 different and discrete grasp types that need to be supported.

In addition to the skeletal tracking that needs to take place, specific forces also need to be captured, either for cutting support, or for ergonomic evaluations. This mostly includes forces affecting the workers hands and fingers, e.g. when carrying or picking objects or tools, or pushing and pulling objects like trolleys.

During further requirements investigation, it was found that a number of parameters originally mentioned in D1.2.1 do not need to be tracked within INTERACT, namely:

- Visual attention / cognitive state
- Vital data

2.2. Tool and object tracking

As presented in D1.2.1 sensors can be used in order to monitor several parameters of tools that participate in a work task. The tracking can be achieved either by using existing interfaces (e.g. APIs) of the tracked tool that provide operational information, or by integrating specific sensors on the tools that provide the desired parameter data.

The following table presents parameters that have been identified as trackable and the way that they can be tracked.

Table 2.1: Trackable tool parameters

Tool Parameter	Description	Tracking Method
Tool motion	Includes the capturing of the tool's motion and orientation in 3D space	• Integration of motion capturing sensor on tool (e.g. IMU)
Tool Operation	Parameter that states that the tool is currently operating	• Parameter retrieval from tool interface if available (e.g. API) • Integration of external sensor in order to capture tool operation
Tool operational parameter	Any parameter of the tool's operation such as torque, force, etc.	• Parameter retrieval from tool interface if available (e.g. API) • Integration of external sensor in order to capture operational parameter

Tracking of objects like assembly parts or assets that participate in a work task includes only tracking of their motion (position and orientation). This can be achieved by integrating an IMU on the object which can deliver motion and orientation related information.

3. TECHNICAL SOLUTIONS FOR MONITORING MANUAL SHOP-FLOOR OPERATIONS

3.1. Sensor System Architecture overview

The following figure shows the overall sensor system architecture which is planned to be implemented within the INTERACT system.

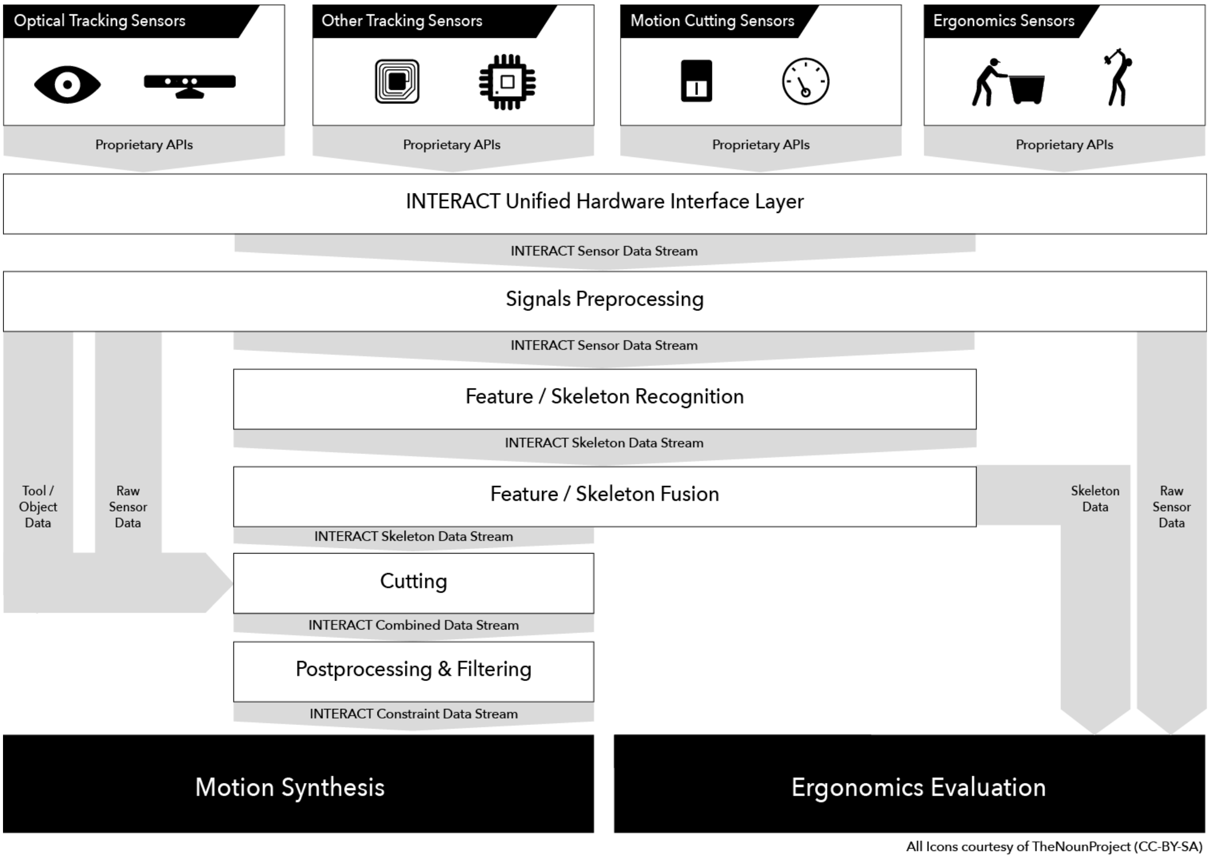

Figure 3.1: INTERACT Sensor system architecture

On the input side, different types and classes of sensors are connected to track parameters of human movements, object movements, tool operation, cutting parameters and ergonomic parameters. The former two comprise full 6-DOF tracking, while the former are made up of data points of different granularity. For example, a cutting support sensor may be only a simple boolean on/off switch, while ergonomic or tool parameters may require a 1D analog scale.

All of the mentioned sensors are accessed via proprietary, vendor-specific APIs, for which a wrapper layer has to be implemented, to allow for transparent sensor access from the INTERACT system. This also allows for swapping of sensors with minimal implementation effort.

Some sensor signals need to be preprocessed before going through the feature detection component, so this is accomplished within a separate step in the data processing pipeline. Again, the unification of the sensor data formats helps here to be able to apply processing algorithms to different sensor data types likewise.

As some of the sensor data are directly mapped to parameters e.g. in the ergonomic evaluation or cutting module, some data is directly forwarded to these components without further processing. Other data, as for example the skeletal tracking data, needs to undergo feature extraction and possible fusion steps, where redundant and / or complementary information is inferred and combined. During this step, it is also necessary to facilitate a skeleton retargeting, as the input skeletons do not necessarily match the Rocketbox reference skeleton model.

For the final output of the sensing system to the motion synthesis subsystems, it is necessary to filter and convert the sensor values to the constraint input format which the motion synthesis component expects.

3.2. Optical sensing system

The optical sensing system forms the base tracking system for human movement tracking within INTERACT. Its purpose is to track the worker movements in the workshop with the feature levels described in ch. 2.1.

3.2.1. Basic system structure

The optical sensing system will be constructed from multiple 2nd gen. Microsoft Kinect for Windows Sensors, which became recently available to the public also for standalone, commercial use.

The 2nd gen. Kinect delivers an improved depth image compared to the 1st gen. Kinect, as well as an extended field of view (see following figure).

Figure 3.2: Kinect 2 (left) vs. Kinect 1 (right) depth image

The output of those sensors is fused together to create a coherent view of the workshop scene. The Kinect sensors track the scene via TimeOfFlight technology, which delivers a 3D point cloud from multiple angles. On this point clouds, the Kinect skeleton recognizer is able to extract a basic skeleton, which can be further processed within the INTERACT system.

Figure 3.3: The 2nd gen. Kinect for Windows sensor hardware

The 2nd gen Kinect sensors deliver multiple outputs:

- Depth image: 512 x 484 pixels ToF @30fps
- RGB color image: 1920 x 1080 pixels @30fps
- Infrared image: 512 x 484 pixels @30fps
- Audio stream: Directionally coded room microphone
- Skeleton stream: 25-joint skeleton model

Each of this output may provide a REST endpoint, in order to access additional sensing data when necessary. For example, besides the pure skeletal data, it may be useful to also gather the IR image for arbitrary CV tasks, as this image has also very little variance due to external lighting influences.

3.2.2. Input skeleton characteristics

The Kinect SDK skeleton recognizer delivers a 25-joint skeleton, including hand joints and head orientation. However, the skeleton still differs from the Rocketbox skeleton in some details. See the following figures for a comparison of the skeletal detail levels.

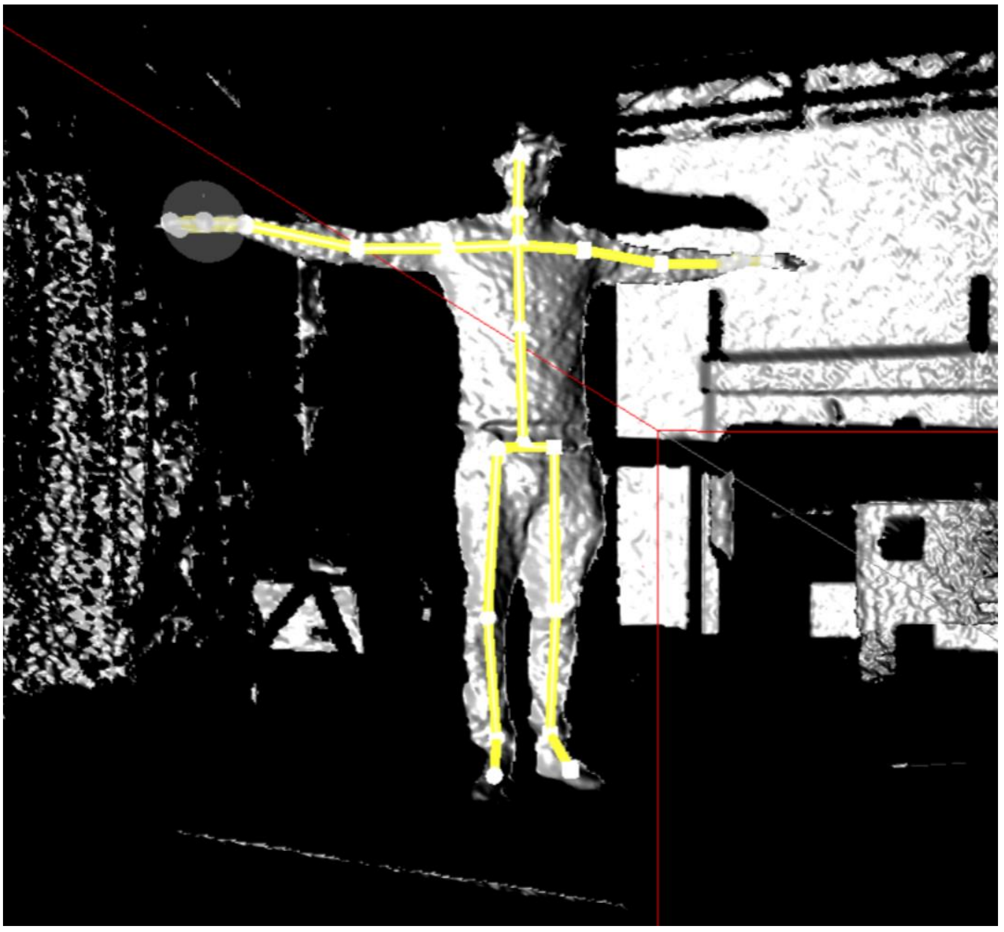

Figure 3.4: Kinect 2 skeletal tracking overlaid to depth image

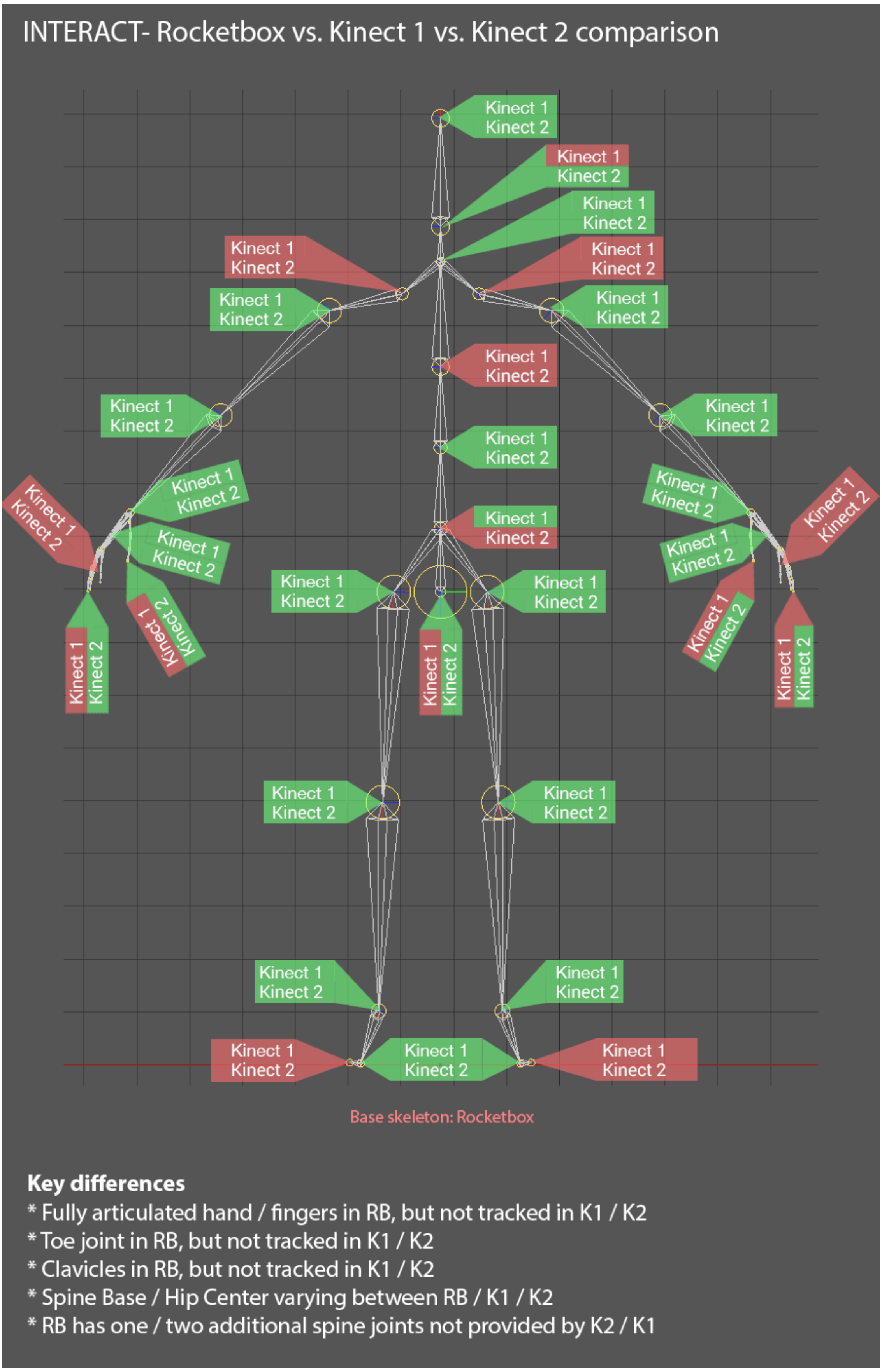

Figure 3.5: Skeletal detail levels of Kinect 1, Kinect 2 and Rocketbox reference skeleton

3.2.3. Multi-camera handling

As already mentioned, it is planned to use multiple sensors in parallel within the INTERACT workshop scenario. The main reasons for this are:

- Occlusion handling
- Extension of tracking area
- Improvement of overall skeleton quality

Depending on the particular goal to achieve, camera positions for such a multi-camera system can lead to variable overlaps between those cameras. This also leads to a higher risk of interference, however previous experiments at Daimler and IMI showed a fair robustness against interference when using multiple 2^{nd} gen. Kinects.

3.2.4. Sensor connection and uplink infrastructure

The Kinect sensors rely on USB 3.0 as a low level data transport protocol. As it is difficult to achieve the amount of cable length necessary for INTERACT with pure USB 3.0, every sensor needs to have its own processing computer located near the sensor. This does not only allow for direct preprocessing of the captured data, but also enables a gigabit LAN-based connection to the other INTERACT infrastructure components.

The minimal system requirements for using a 2^{nd} gen. Kinect for Windows sensor according to Microsoft are:

- **Operating system:** Windows 8, Windows 8.1, Windows Embedded Standard or Windows Embedded Standard 8.1
- **Processor:** 64-bit (x64), dual-core 2.66-Ghz or faster processor
- **Memory:** 2GB RAM
- **Graphics card:** DirectX11 compatible graphics card
- **Port:** USB 3.0

Possible solutions will be not only small industrial PCs with smaller form factors, but also Windows 8 tablet PCs, which provide an additional user interface for controlling the node.

On the processing computer, it is necessary to provide several software layers to enable the integration into the INTERACT system:

- Kinect Sensor Drivers
- Kinect SDK w/ Skeleton Recognition Components
- Middleware, which transforms the Kinect SDK interface to the INTERACT REST interface (and vice versa).

3.2.5. Optical sensing architecture overview

The following figure includes the complete optical sensing architecture including all aforementioned components.

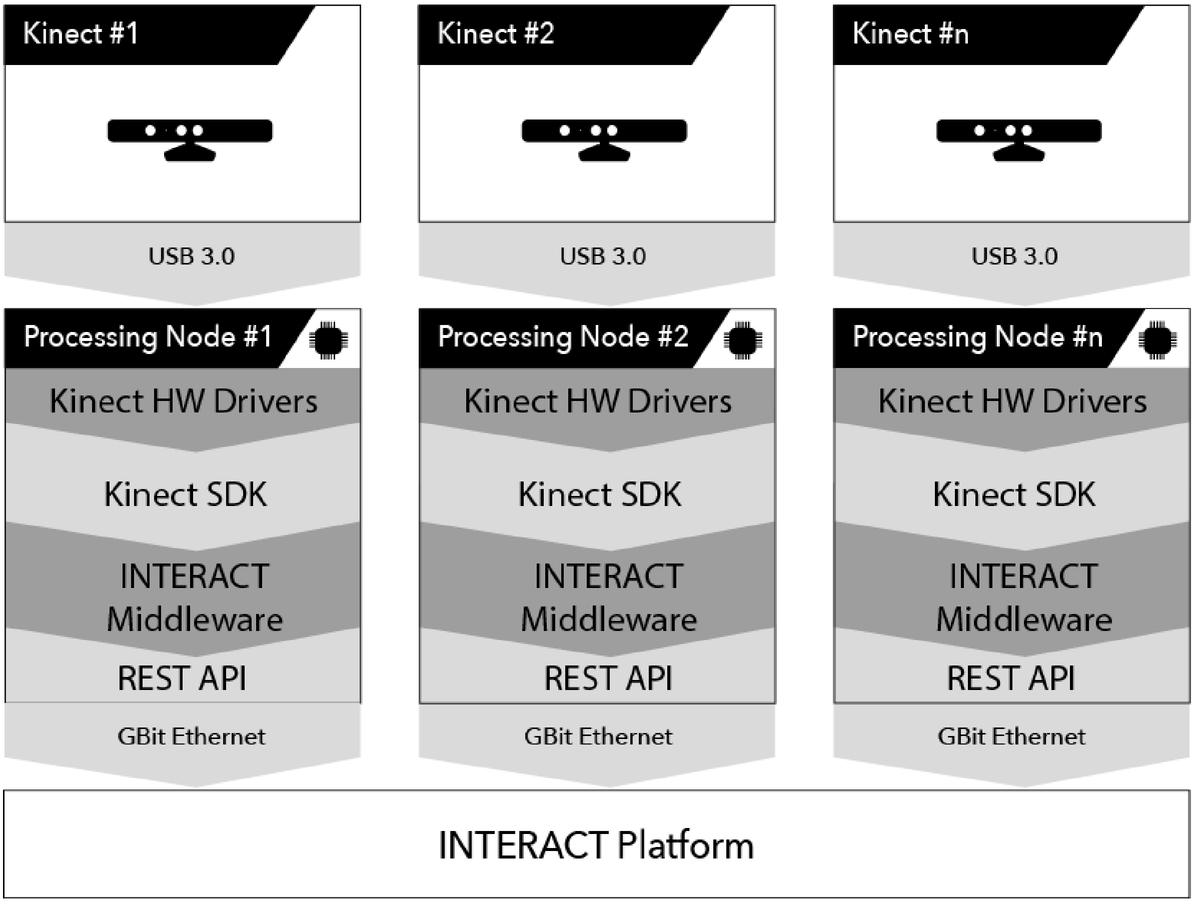

Figure 3.6: Optical sensing architecture

3.3. MEMS and other sensors system

As described on sec 3.1 additional sensors are needed for the INTERACT system in order to supply additional information in complement to the optical sensing system. This includes:

1. Additional sensing information for the retargeting of the optical sensor system skeleton output to the INTERACT skeleton
2. Additional sensing information for the capturing for motion types and detection of movement cuts
3. Additional sensing information for capturing the movement, position and orientation of parts, tools and assets
4. Additional sensing information for the detection of tool operation and other specific tasks like part clipsing
5. Additional sensing information for the ergonomic evaluation of the work tasks

In order to implement those requirements a Wireless Sensor Network (WSNt) has been designed. This network consists of three levels, which are as follows:

1. Set of sensors which measure all needed parameters
2. Set of Wireless Sensor Nodes (WSNs) which integrate all needed sensors, collect sensor data and transmit it to a base station
3. Wireless Sensor Base Station (WSBS) which collects, processes and forwards sensor data to the INTERACT platform.

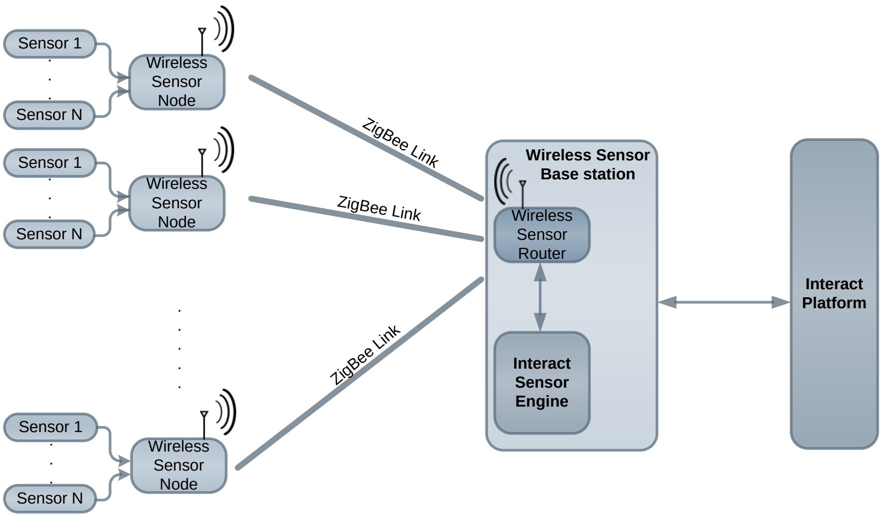

Figure 3.7: INTERACT Wireless Sensor Network

3.3.1. Sensors

In order to fulfill sensor requirements related to monitoring manual operations as described in sec. 2 a set of sensors have been chosen, as described below

1. ***9-axis MEMS accelerometers***, which measure:
    - Linear accelerations on 3 axes
    - Angular accelerations on 3 axes
    - Sensor orientation through magnetic fields measurement on 3 axes

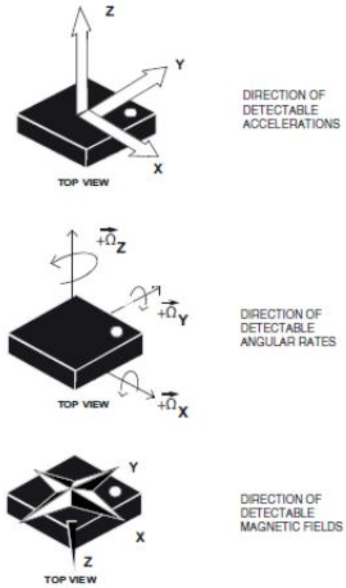

Figure 3.8: 9-axis accelerometer

These sensors will be used in order to implement Inertial Measurement Units (IMUs) for capturing position, movement and orientation of specific body parts, body joints, objects, tools and assets.

2. ***Force/pressure sensors***, which:

    - Detect and measure a relative change of force or applied load
    - Detect and measure the rate of change in force
    - Detect contact and/or touch

Figure 3.9: Force sensor

These sensors will be used within the INTERACT project in order to:

- detect foot to floor contact with the use of two sensors under the worker's feet
- detect application of finger pressure with the use of sensors in the tips of the desired fingers
- detect grasping operation with the placement of sensors on the hands

3. ***Bending sensors***, which detect and measure a change in their bending ratio and produce a proportional analog output. They will be used to measure bending ratios in various points such as toe joints.

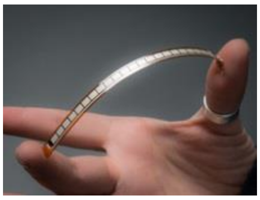

Figure 3.10: Bending sensor

4. ***MEMS microphones***, which capture sound like ordinary microphones. They will be used within INTERACT in order to implement clipsing operation detection. For this purpose two different microphones will be used. One very close to the clipsing point for detecting the clipsing sound, and one in the workshop area for capturing environment sound. Subtracting the captured sound levels in those two points will result in the detection of the clipsing operation.

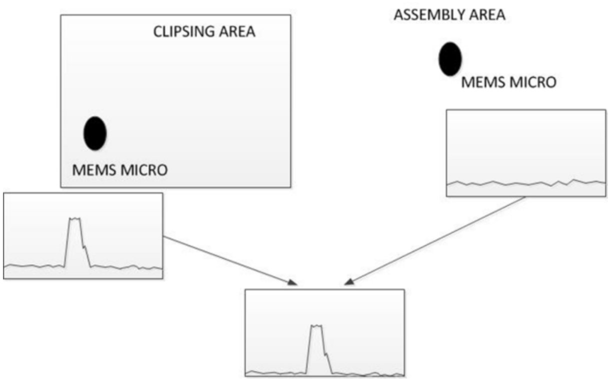

Figure 3.11: Clipsing detecting principle

3.3.2. Wireless Sensor Node

In order to be able to integrate the above presented sensors, or other that may be needed, a wireless platform has been designed which:

1. Interfaces with the sensors and receives sensor information
2. Interfaces with the Wireless Sensor Base Station in order to transmit sensor data and receive sensor management commands

The main design characteristics of the WSN are the following:

- Generic peripheral design in order to support variety of analog and digital sensors
- Reconfigurable sensor operation through ability of changing over the air sensing parameters through a Sensor Management Platform
- Low power consumption for longer battery autonomy
- Compact board size for wearable applications

The main features of the design are as follows:

1. ARM CORTEX-M3 based 180 MIPS Microcontroller with 512kBytes Flash memory, embedded RTC, and RTOS
2. 8 analog channels for sensor interfacing
3. 6 digital inputs and 2 digital outputs for peripheral interfacing
4. SPI interface for up to 3 peripheral connections, mainly for MEMs sensors
5. On board 9-axis accelerometer, in order to use the WSN as IMU without any external sensor interface need
6. On board Micro-SD interface for data storage
7. On board UART and USB interface for external peripheral connections
8. On board ZigBee transceiver for communication with the WSBS
9. 3.8 V battery operated

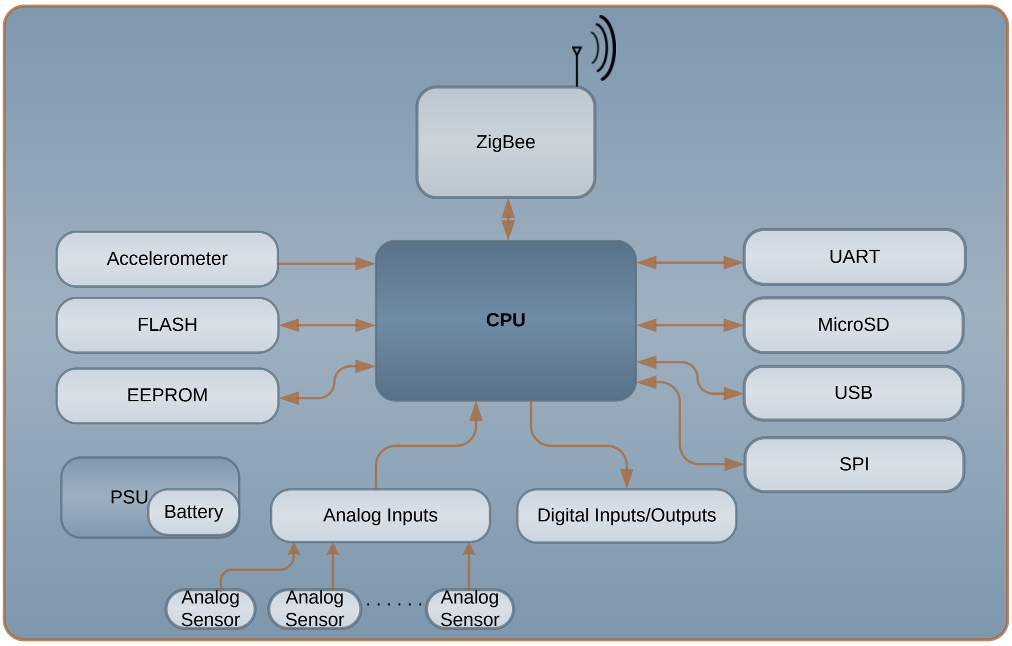

Figure 3.12: INTERACT Wireless sensor node architecture

3.3.3. Wireless Sensor Base Station

The different kinds of sensors presented in sec. 3.3.1 generate raw data streams that have to be efficiently recorded and processed in order to extract information and forward it to the external INTERACT modules.

The following figure depicts the overall architecture of the wireless sensor base station that receives data, filters it and delegates it to other external modules.

The left-most part of the figure shows the sensor nodes that integrate various sensors that capture motion data. Each wireless node implements a ZigBee-based wireless protocol that interacts with the communication component of the Base Station architecture. The transferred data format is proprietary binary.

The communication component coordinates the communication with the Wireless Sensor Nodes. On one hand, it received sensor data from nodes. Due to the nature of the data that are generated (high frequency and high sensitiveness to rapid changes), it should be filtered in order to satisfy certain criteria of correctness, quality and relevance. Possible sources of inconsistencies could be:

- environmental disturbances like noise, vibrations and high temperatures, or
- sensor faults and hardware problems, or
- transmission corruption and delays caused by (temporarily) unreliable or overloaded networks

Filtering is done by the second component that the communication component forwards data to. Some examples of sensor-dependent filtering are:

- elimination of spikes transmitted by localization sensors by providing delayed identification of movement events after a satisfactory number of candidates has been received
- for the case of sensors with events from a continuous numeric domain (e.g. force or pressure) a range-based filtering is demanded to ignore extreme values

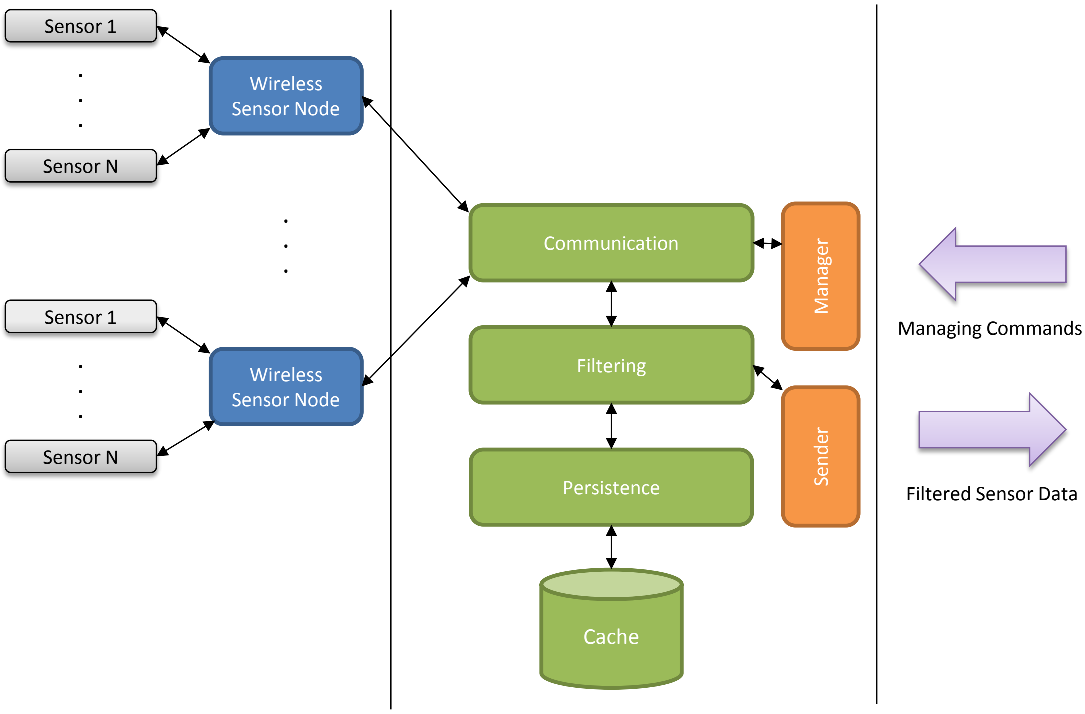

Figure 3.13: INTERACT Wireless Sensor Base Station architecture

Filtered sensor data are directed to the persistence component that saves them to a local cache. The cache is implemented through a relational database system and its role is threefold:

1. It keeps track of sensor topological and addressing information. For example, the available nodes with their addressing IP, the sensors that are attached to them and for each sensor its type and the node channel that data is sent.

2. A relationship between each sensor and the external modules in order to have the flexibility to change the working context of a sensor and send its data to a different external module each time (i.e. data analysis, cutting and ergonomics). Such mappings are configured by the Sensor Management module.

3. A historical record of all sensor data readings in the recent period is kept. Such information is useful for traceability reasons and cases that network with external components is down and data needs to be retransmitted.

Sensor information, after being filtered, is automatically pushed to external data analysis, cutting or ergonomics modules to further contribute to the analysis of captured manual operations. The transmission protocol is based on uniform REST-based HTTP requests that describe sensor ids, timestamps and recorded values.

The INTERACT Sensor Management module configures the sensors landscape sending specific commands being delegated by the sensor engine to the individual nodes. Some commands control the start and end of capture and others configure the participating sensors. For example:

1. Add command: includes a specific sensor to a monitoring session
2. Remove command: excludes a specific sensor from a monitoring session
3. Init command: instructs a specific sensor to calibrate or reset according to its current position or state

4. ChangeParameters command: uploads a new configuration to a sensor (e.g. change sampling frequency or sensitivity
5. StartCapture command: all participating sensors are instructed to start capturing data
6. StopCapture: all participating sensors are instructed to stop capturing data

Management command API is based on HTTP JSON-based requests.

3.4. Data format and communication protocols with aggregation layer

3.4.1. Motion data format specifications

In the INTERACT project the skeletal structure that will be used is the RocketBox skeletal representation (see D1.2.1). Therefore, the motion capturing data provided as described in the previous section must be translated before they are stored to be used. Since the BVH files will be used in many steps across the Best-fit simulation pipeline a standard format has been decided in order to have mutual input and output mechanisms at all steps.

The BVH format as described in [2] contains two parts; a header part where the structure of the skeleton as well as the bone dimensions are defined and the second part where the transportation and rotation values are given. In the first part, beyond the dimensions of the skeleton in the form of joint offsets, the channels that the values from the second part correspond to are provided. Since BVH formats do not contain scaling information only the rotation and translation matrices are considered to construct the local transforms. The channels can be transformation, rotation or both.

In INTERACT the root joint is considered to be the hips of the skeleton for which six channels are provided; transformation on the XYZ axes and rotation on the XYZ axes. The three first channels provide the transformation of the whole skeleton through relative transformations of the rest of the joints. For the rest of the joints only the rotation channels are provided, necessary for the calculation of the skeleton's pose at each frame (Figure 3.14).

```
HIERARCHY
ROOT Hips
{
    OFFSET X Y Z
    CHANNELS 6 Xposition Yposition Zposition Xrotation Yrotation Zrotation
    JOINT Spine
    {
        OFFSET X Y Z
        CHANNELS 3 Xrotation Yrotation Zrotation
        JOINT Spine_1
        {
            OFFSET X Y Z
            CHANNELS 3 Xrotation Yrotation Zrotation
            JOINT Neck
            {
```

Figure 3.14: Header format of the INTERACT BVH files

The sequence of rotation calculations is in the sequence of X-Y-Z and the rotation matrix of each joint is:

$$R = R_X R_Y R_Z$$

Finally, the minimum frame-rate of the BVH files used is 72 fps which corresponds to a frame time of about 0.014 seconds.

The communication of the BVH-structured data is accomplished via RESTful webservices analog to the object and tools data transmission described in the following sections. As BVH relies on the run-time calculation of joint positions depending on the header data, the implementation needs to respect this by providing a header at stream-time if requested by the client (e.g. via an argument included in the GET request that was issued).

A possible first request then might look as follows:

Client request:

```
GET /skeleton/?id=1&bvhheader=true
```

Server response:

```
HIERARCHY
ROOT Hips
{
	OFFSET		-4.79878	52.253353	285.993988
	CHANNELS 6 Xposition Yposition Zposition Xrotation Yrotation Zrotation
	JOINT Spine
	{
		OFFSET		15.3169		0.0	-0.012207
		CHANNELS 3 Xrotation Yrotation Zrotation
		JOINT Spine_1
[…]
```

All subsequent requests do not provide the “bvhheader” flag and thus are answered with the numeric BVH frame data only.

3.4.2. Object tracking data requirements

The object used in the INTERACT cases include both product parts as well as tools that include handheld power tools and finally assets such as trolleys. Different data are required for each type of object.

Table 3.1: Requirements for the tracking of objects

Parts and sub-assemblies	Tools	Assets
• ID • Position • Contact/collision • Grasping type	• ID or type • Position • Orientation • Process parameters	• Position • Orientation • Contact collision

The parts will be tracked based on their ID in order to document the possible sequence changes which will be valuable for the regeneration of the relevant simulation. The part’s ID can be tracked directly from the optical systems or alternatively through other sensors during workshops. Their position will be tracked in the same way (e.g. through the combination of force sensors and optical data), mainly to identify that they are being carried by the operator and that they have been placed/ assembled in their final position (also, using sound sensors). In the same way, the contact or collision identification will provide assembly related information and the grasping type will document the exact way that an operator chooses to grasp a part in order to place it in the assembly or trolley.

The tools position and orientation will be tracked in order to identify the way that the operators work with a certain type of tool and the way the tool is positioned during operation in order to examine relevant ergonomic values. Finally, the process parameters will be tracked in order to identify if there are variances from the planned processes as well as the way the operator controls the tool (e.g. torque/ time distribution). Currently a specific hand-tool model that is examined to be used in INTERACT and can possibly satisfy these requirements, is the cordless angle screwdriver Makita BFL201R which should be able to communicate data regarding:

- tool ID,
- torque value,
- turning angle
- turning obstructions,
- cycle time and timestamps for operations (post operation calculation)
- error code (e.g. exceeding torque as post operation calculation)

Figure 3.15: Makita BFL201R

The information will be provided through Bluetooth in this case directly from the equipment but additional sensors such as MEMs will be used to provide additional or more precise information.

Finally, different assets such as trolleys will be tracked, again in order to identify differences from the simulation and real-life (workshop) situations. This information will be used in order to generate more realistic simulations.

3.4.3. Skeleton data formats prior to Fusion

Before skeletal fusion and retargeting to an INTERACT-conforming skeleton takes place, it is necessary to transport skeletal data which may be of arbitrary formatting (e.g. Kinect2-specific skeletons with fewer joints than the Rocketbox skeleton).

For this transport within the data processing pipeline, the framework and conditions specified in section 3.4.1 also apply, but with a slightly changed BVH structure, which represents the particular features of the "raw" skeleton before fusion.

If the skeleton transport takes place within system borders (thus not using the REST / network approach), the respective API object structures can be used.

3.4.4. Raw sensor data for Cutting and Ergonomics analysis

Sensor data is transmitted to the cutting and ergonomics modules using the classification displayed in the following diagram.

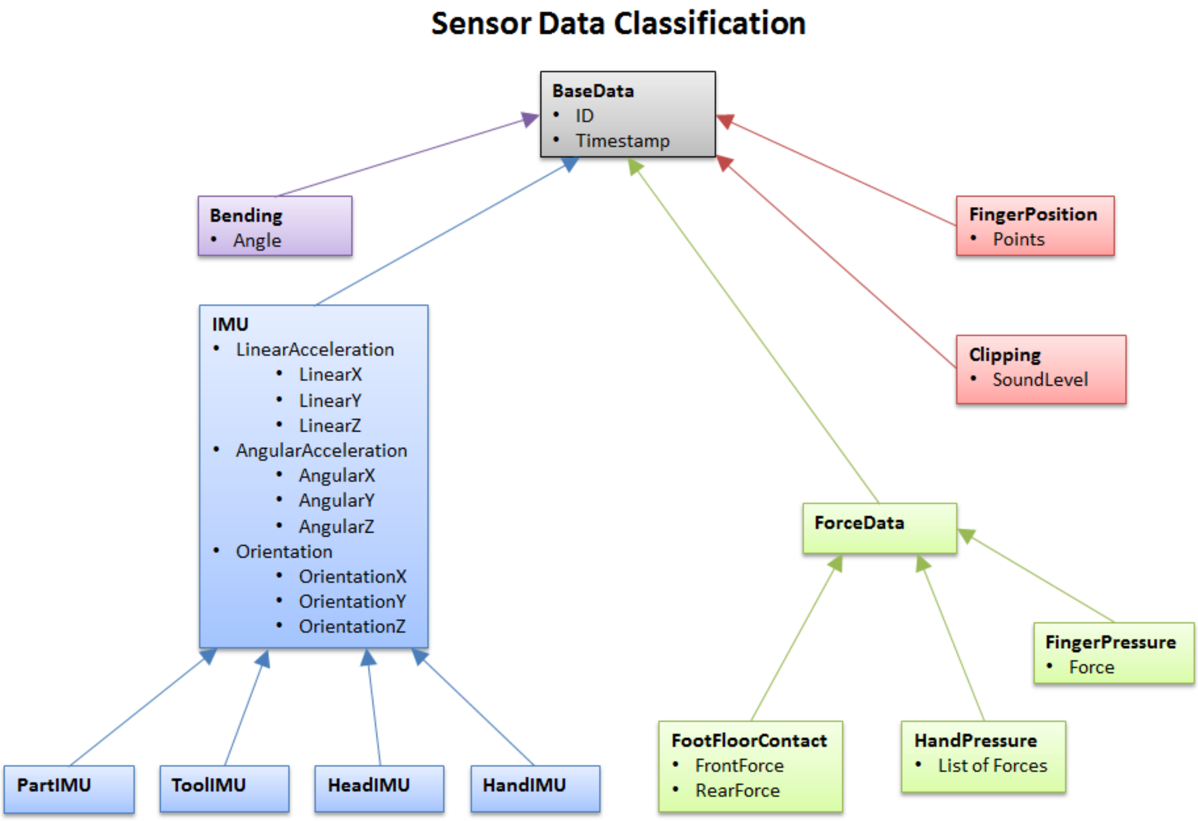

Figure 3.16: Sensor data classification model

Each sensor data inherits the BaseData type that contains the ID of the sensor that generated the event and the timestamp. Due to the high frequency of captures, the timestamp structure has milliseconds resolution.

For each of the sensors described in Sec. 3.3.1 a class is defined with the properties that are being measured. IMU sensors provide three-dimensional linear acceleration in mg (milli G), angular acceleration in milli Degrees / sec and magnetic-field based orientation in milli Gaus.

The force sensors provide data in kg. Special cases are the FootFloorContact that provide front and rear force values, and the HandPressure that provide a list of forces (one for each endpoint attached to hand). The Bending sensor measures angles in degrees and the Clipsing sensor sound level in dB.

Every sensor reading is converted to a unified representation according to the sensor data model and sent to the external cutting and ergonomics module. The transmission is realized through an HTTP REST-based protocol. Below are some examples of such messages.

Table 3.2: INTERACT sensor messages

Sensor Data	HTTP REST request
IMU	event/?**s=**1324&**t=**2014-05-02T14:41:25.430Z&**lx=**2345&**ly=**3234&**lz=**4323 &**ax=**3&**ay=**323&**az=**43&**ox=**2&**oy=**3&**oz=**122
FootFloorContact	event/?**s=**332&**t=**2014-05-02T11:21:02.200Z&**ff=**32&**rf=**44
HandPressure (4 points)	event/?**s=**432&**t=**2014-05-02T09:04:02.125Z&**f1**=1&**f2**=3&**f3**=5&**f4**=5
FingerPressure	event/?**s=**3&**t=**2014-06-02T14:22:51.230Z&**f=**43

After the cutting and similar to the post-processing of MoCap data (see section 5.3), all sensor data after a workshop will be filtered in order to eliminate accidental peaks, smooth jittered signals and fill gaps in the signals coming from all sensors in order to increase the usability of these data and avoid any incorrect information being provided to the end users. This will be a dynamic process as much as possible and will be done automatically.

3.5. Sensor Management Platform

Sensor Management Platform (SMP) is a software component that enables management of the sensor systems as defined by adding, removing, assigning, combining, configuring, calibrating and monitoring individual sensors and sensor software modules. It is the administration component of the integrated INTERACT sensor system.

SMP is comprised of following modules depicted on the below diagram (part of which are overlapping with the Sensor Management App deliverable described in WP5, but are included here for clarity):

- Sensor Management Platform Service
  This is the central part of the platform. It contains all logic needed to manage the INTERACT sensors system.
- Interface to sensor subsystems (left)
  Optical Sensing Processing Nodes and Wireless Sensor Base Station (described in relevant chapters of this deliverable) provide interfaces through INTERACT Unified Hardware Interface Layer through which Sensor Management Platform Service manages all sensors.
- Sensor Network Management App (top)
  GUI and RESTful service of the Sensor Network Management App form the end-user interface to the platform.
- Persistent storage (bottom)
  The CRUD Data Service, used for sensor configuration data storage, is provided by the Enterprise Application Platform.
- Sensor related parameters storage for other INTERACT modules (right)
  Parameter storage for other modules is optionally provided across the boundaries of the INTERACT Platform components. This should pertain only to parameters closely related to the sensors system.

All internal WP3 modules communication, as well as communication with WP5 Sensor Network Management App is done through JSON format based RESTful HTTP APIs.

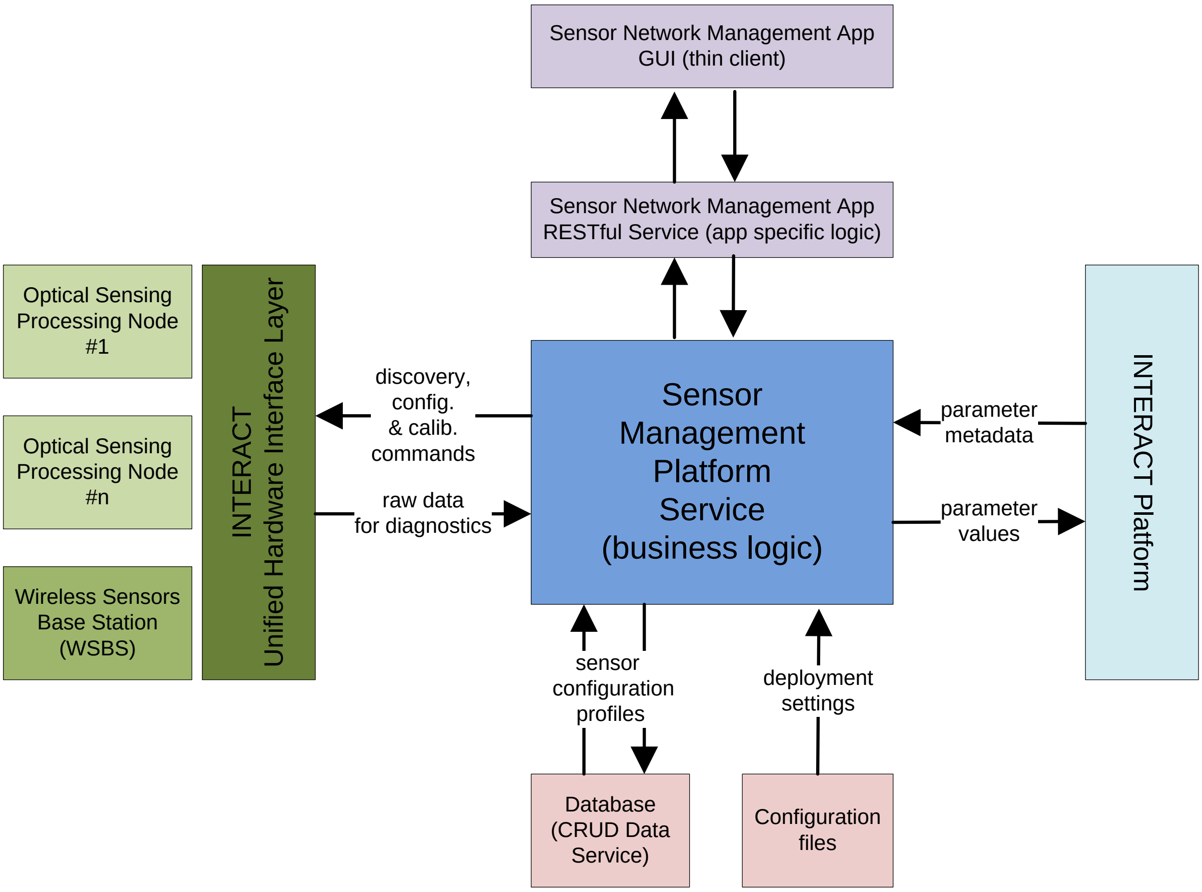

Figure 3.17: Architecture of the Sensor Management Platform

3.5.1. Sensor Management Platform Service

This service performs all the heavy work regarding sensor configuration and management. It provides RESTful interfaces for Sensor Network Management App, through which all the functionality can be accessed by the end user. It uses RESTful interfaces exposed by sensor subsystems to perform sensor discovery, configuration and calibration.

The module implements following functionality:

- It provides default parameters for known types of sensors.
- It contains functionality for sensor assignment to different topological locations and logical purposes in the INTERACT system.
- It provides mechanisms for interaction with the user during individual sensors calibration, and calibration of whole subsystems (this could be for example a set of Kinect nodes, possibly along with assigned MEMSs) or the system as a whole.
- It contains functionality to prepare and switch configuration profiles for the whole sensor system. A configuration profile contains information about a set of sensors which are to be used along with their configuration and calibration parameters. This provides for quick switching between different configurations, e.g. to test them side by side, or to have ready

made configurations when frequently moving physical components of the system between predefined physical locations.

- It contains functionality for monitoring raw data of individual sensors for diagnostic purposes.

3.5.2. Interface to sensor subsystems

SMP is interfaced to sensor subsystems (Optical Sensing Processing Nodes and Wireless Sensor Base Station described in relevant chapters of this deliverable) through INTERACT Unified Hardware Interface Layer.

SMP sends following messages to the hardware layer, part of which was already mentioned as examples in relevant chapters. Command parameters are shown in parentheses:

1. Discover ()
   Gets available sensors' IP addresses, IDs, types.
2. Add (sensor ID, session, sensor function)
   Includes a specific sensor in a monitoring session. The session is assigned to specific external module accepting sensor data, with assigned physical sensor location and logical function.
3. Remove command (sensor ID, session)
   Excludes a specific sensor from a monitoring session.
4. Init (sensor ID)
   Instructs a specific sensor to calibrate or reset according to its current position or state.
5. StartCalibration (module)
   Puts a software module into calibration mode.
6. CommitCalibrationStep (module)
   Commits current calibration step. Advances or ends calibration (depends on specific software module and its current calibration state). Software module must reply with a text message describing next calibration step for the end-user.
7. ChangeParameters (sensor ID, parameter list, ...)
   Uploads a new configuration to a sensor (e.g. change sampling frequency or sensitivity).
8. ChangeParameters (module, parameter list, ...)
   Uploads a new configuration to a software module (e.g. extrinsic calibration parameters).
9. StartCapture ()
   All participating sensors are instructed to start capturing data.
10. StopCapture ()
    All participating sensors are instructed to stop capturing data.
11. StartRawMonitoring (sensor id)
    Puts specified sensor into raw data transmission mode for diagnostic purposes. Hardware layer starts sending RawData messages to SMP.
12. StopRawMonitoring (sensor id)
    Ends raw data transmission to SMP for given sensor.

Following messages are sent by the hardware layer to the SMP:

1. GetParameters (sensor id)
   Obtain currently set parameters for a given sensor. Needed for initialization after a reset.
2. GetParameters (module)
   Obtain currently set parameters for a given software module. Needed for initialization after a reset.
3. RawData (sensor id, data)
   Raw data stream used by SMP for diagnostic purposes.

3.5.3. Sensor Network Management App

For architectural flexibility, Sensor Network Management App was split into two components: a RESTful App service containing all the screen navigation logic (Controllers or Presenters following the MVC/MVP architecture), and thin client GUI part which is solely a graphical presentation layer (Views part of the MVC/MVP) and can be implemented in any UI technology including HTML and AJAX.

Examples of messages passed between the App service and UI are:

- UI asks App service to display certain screen
- App service gets business data from SMP, and passes as formatted screen data (UI controls' captions, values, actively selected tabs, etc.) to UI
- UI tells App service that user has clicked a button to perform some function
- App service performs communication with SMP to invoke relevant business logic, formats all resulting data for display, performs necessary screen transition logic, then passes a message describing a screen for display to the UI

Main benefit from using above architecture is that the app client contains plain UI, without any business logic and with minimal presentation logic so it can be implemented on very limited devices, such as smartphones.

3.5.4. Persistent storage

SMP uses the CRUD Data Service provided by the Enterprise Application Platform to store all dynamic data used for sensor and WP3 software modules configuration, calibration and sensor assignment. All configuration profiles created by the end-users are saved here. Also parameters for other INTERACT modules can be stored, as mentioned in the below chapter.

SMP also uses configuration files for some less frequently changed overall deployment settings, such as descriptions of sensor types and available sensor physical locations and functions of different sensor types in context of software modules (e.g. list of available joints and bones on a skeleton to place a sensor).

3.5.5. Sensor related parameters storage for other INTERACT modules

SMP can optionally provide GUI sections for configuration of algorithms and modules functioning in other parts of the INTERACT Platform, which don't have their own GUI for configuration and calibration or want to have GUI for sensor-specific options in one place with all other sensor configuration (the Sensor Network Management App). This is depicted on the right hand part of the diagram. For example an Ergonomics Evaluation module might want to put its sensor-specific options in the GUI of Sensor Network Management App (for whatever reason). To achieve this, the module should be registered with SMP in its configuration files and provide implementation of Parameter metadata interface (through a RESTful API). Parameter metadata contain simple definitions of all parameters that need configuration like: parameter name, parameter type. SMP would then ask Ergonomics Evaluation module about all Parameter metadata, and display those parameters for configuration in Sensor Network Management App GUI. Parameters set by the end-user in this app would then be sent to Ergonomics Evaluation module whenever changed or needed during system initialization.

4. ADDRESSING OF HUMAN WORK TASK MONITORING REQUIREMENTS FOR THE INDUSTRIAL USE CASES

In the following sections, the requirements satisfaction for both INTERACT industrial cases is examined, in order to verify that all work tasks will be tracked during the development of the platform as well as the tracking method that will be used.

Each motion type is presented in the table below with the relevant equipment that will be utilized for its tracking. Regarding the tasks of both use cases, they are all described based on the motion types that are required to complete them. The motion types that have currently been decided are:

Table 4.1: INTERACT Motion Types

No.	Motion type	Motion Description	Motion Primitives	Tracking equipment
1	Walk	Comprises right and left stances incl. Starting and Stopping, only comprises way lengths > 1 m	Walk Right Stance Walk Left Stance	• Optical system for motion capturing • Force sensors for foot-floor contact detection
2	Carry	Same as walk, however only with velocity of 1m/s and an object being attached to one or two hands.	Carried Object Position Carry Right Stance Carry Left Stance	• Optical system for motion capturing • Force sensors for foot-floor contact detection • IMU on carried object for motion capturing • Force sensors on hands for carrying detection
3	Pick	Action grapsing an object while standing next to it. The action starts when the hand starts moving away from the body while the worker is standing on the ground or sitting. The action ends when the object is leaving its previous position.	Pick Reach Pick Grasp	• Optical system for motion capturing • Force sensors for foot-floor contact detection • IMU on carried object for motion capturing • Force sensors on hands for carrying detection
4	Place	Action putting down an object. The action starts when the hand starts moving away from the body while the worker is standing on the ground or sitting. The action ends when the hand	Place Put Place Release	• Optical system for motion capturing • Force sensors for foot-floor contact detection • IMU on carried object for motion capturing • Force sensors on

		has left the object.		hands for carrying detection
5	Side Step	Non continuous side steps with way lengths < 1m and an object being attached to one or two hands.	Side Right Stance Side Left Stance	• Optical system for motion capturing • Force sensors for foot-floor contact detection • IMU on carried object for motion capturing • Force sensors on hands for carrying detection
6	Press with Finger	Instance of Application of pressure with finger	Finger motion towards the pressure point. Pressure application. Finger motion away from pressure point	• Optical system for motion capturing • Force sensor on finger for pressing detection
7	Bolt Connect/Disconnect	Attaching tool to bolt. Pressing tool button in order to start process. Releasing button to stop process. Removing tool from bolt	Tool movement to bolt, Tool operation, Tool movement from bolt	• Optical system for motion capturing • IMU on tool for motion capturing • Sensors on tool for operation detection
8	Turn head to look around			• Optical system for motion capturing • IMU on head for motion capturing (if needed)
9	Hand screwing	The movement rotates the hand, wrist, and forearm about the long axis of the forearm (left/right).	Hand attachment. Hand Turning. Disattachment of hand	• Optical system for motion capturing • IMU on hand for motion capturing
10	Push/pull object (extend / retract arms)	Same as walk, however only with velocity of 1m/s and an object being pushed or pulled.	Body posture and hand attachment for pushing. Body posture and hand attachment for pulling. Hand pressure for pushing. Hand pressure for pulling, Disattachment of hand	• Optical system for motion capturing • IMU on carried object for motion capturing • Force sensors on hands for carrying detection
11	Press/Lift (move object down/up)	Same as carrying, without walking	Press/Lift Object Position, Press/Lift	• Optical system for motion capturing • IMU on carried object for motion

				capturing • Force sensors on hands for carrying detection
12	Regrasp (Change grasp without relinquishing control)	Changing of hand postures and pressure	Hand movement	• Optical system for motion capturing • Force sensors on hands for carrying detection
13	Transfer (control transfer grasp from one hand to other)	The movement rotates the hand, wrist, and forearm about the long axis of the forearm (left/right).	Hand attachment. Hand Turning. Disattachment of hand	• Optical system for motion capturing • Force sensors on hands for carrying detection
14	Clipsing	Attaching a part on another by pressing it till clip is in final position		• Optical system for motion capturing • MEMs microphones in clipsing area • Force sensors on hands for clipsing detection

In the following sections, each Motion type is linked to a specific task of the use cases and in some cases different alternatives of these tasks are provided.

4.1. Daimler

Daimler's industrial use case is divided into two scenarios; one involves the pre-assembly of a car's console while the other involves the assembly of a rear light to the car (main assembly line). The tables below present the correlation between the work tasks that form the scenarios and the motion types that these tasks consist of.

Table 4.2: Console assembly

Task	Motion Types
1. Pick center console/storage tray out of roller-dare and place in assembly fixture	3-1-4
	3-4
2. Lock holding fixture (of assembly fixture)	11
3. Open lid, center armrest/stowage compartment, front	11-4
4. Insert bracket, media interface control unit into center console/storage tray	3-4
5. Lay cable (of bracket, media interface control unit) into center console	3-4
	3-5-4
6. Tighten bracket, media interface control unit with screw 1x to center console with cordless screwdriver	3-7-4
7. Tighten insert, stowage compartment with screw 1x with center console/storage	7

tray with cordless screwdriver	5-7
8. Unlock Work table	6
9. Turn Work table	6
10. Lock Work table	6
11. Pick wiring harness (of center console) out of set box	5-3
12. Plug in wiring harness (of center console) electrically	7
13. Lay branch-off line K/Z*, socket, center console (left) into center console/storage tray	5-3-13-5-4
	5-3-5-4
	3-4
14. Hook K/Z*, socket, center console (left) 3x into hook (3 pieces)	3-4-7
15. Lay branch-off line K/Z*, socket, center console (right) into center console/storage tray	5-3-13-5-4
	5-3-5-4
	3-4
16. Hook K/Z*, socket, center console (right) 3x into hook (3 pieces)	3-4-7
17. Pick lock storage tray out of set box and place at lid, center armrest/stowage compartment, front	1-3-2-4
	3-2-4
	1-3-4
18. Snap lock storage tray to lid, center armrest/stowage compartment, front	6
	5-6
19. Pick blower motor, rear out of set box and place at center console/storage tray	1-3-2-4
	3-4
	1-3-4
	5-3-5-4
20. Tighten blower motor, rear 4x with center console with cordless screwdriver	5-7
21. Fasten branch-off line K/Z* Blower motor, rear to blower motor, rear with Clip 1x	9
22. Plug in branch-off line K/Z* Blower motor, rear electrically to blower motor, rear	7
23. Position and click section point (of branch-off line K/Z* Blower motor) to center console	5-6
24. Lay branch-off line K/Z* Blower motor, rear (left) on center console	5-3-13-5-4
	5-3-5-4
	3-4
25. Lay branch-off line K/Z* Blower motor, rear (right) on center console	5-3-5-13-4
	5-3-5-4
	3-4
26. Lay branch-off line K/Z* Central audio component on central audio component	5-3-13-5-4
	5-3-5-4

	3-4
27. Clip lighting system, stowage compartment, armrest (top) into center armrest/stowage compartment, front	4-14
	5-4-14
28. Clip lighting system, stowage compartment, armrest into center armrest/stowage compartment, front	4-14
	5-4-14
29. Lay branch-off line K/Z*, lighting system, stowage compartment, armrest on center armrest/stowage compartment, front	5-3-13-5-4
	5-3-5-4
	3-4
30. VM Miko: Code 427: Leitungssatz zu Schalter UBF nach unten durchstecken (Push through the line set to switch UBF down)	6
31. Unlock Work table	6
32. Turn Work table	6
33. Lock Work table	6
34. Thread branch-off line K/Z*, switch group, lower control panel through switch group, lower control panel (right)	6
	5-6
35. Lay branch-off line K/Z*, lighting system, cup holder, front/stowage compartment, front on ashtray/oddments tray	5-3-13-5-4
	5-3-5-4
	3-4
36. Thread branch-off line K/Z*, switch group, lower control panel through switch group, lower control panel (left)	6
	5-6
	1-6
37. Plug in central audio component electrically	7
38. Insert central audio component into center console	4
39. Tighten central audio component to center console with cordless screwdriver	9
	5-9
40. Pick ashtray/oddments tray out of set box and place in center console	1-3-2-4
	3-4
	1-3-4
41. Pick cover, cup holder, center console, rear	5-3-5-4
42. Clip cover, cup holder, center console, rear into ashtray/oddments tray	14
43. Pick cover, ashtray/stowage box out of Charge	5-3-5-4
44. Clip cover, ashtray/stowage box into ashtray/oddments tray	14
45. Plug in center console/storage tray 2x electrically	7
46. Lay branch-off line K/Z*, socket, 12 V on center console/storage tray	5-3-13-5-4
	5-3-5-4

	3-4
47. Tighten center console/storage tray with screw 4x with center console with cordless screwdriver	7
48. Pick cover, stowage compartment/central control and place at center console	5-3-5-4
49. Snap 2x cover, stowage compartment/central control to center console	6
50. Plug in lower control panel electrically	7
51. Pick cup holder, front from set box and place in center console/storage tray	5-3-4
52. Tighten central audio component with screw 2x to cover, stowage compartment/central control with cordless screwdriver	7
53. Pick trim, center console, left and place at center console/storage tray	5-3-5-4
54. Clip trim, center console, left into center console/storage tray	14
55. Pick trim, center console, right and place at center console/storage tray	5-3-5-4
	3-4
56. Clip trim, center console, right into center console/storage tray	14
57. Close lid, center armrest/stowage compartment, front	1-6
	6
58. Open holding fixture	1-9
	5-9
59. Pick center console/storage tray from Work table and place at Picking / roller-dare	1-3-2-4
	1-3-4

Table 4.3: Tail Light assembly

Task	Motion Types
1. Push/pull assembly trolley	10
2. Provide taillamp right on assembly trolley	3-4
	5-3-5-4
3. Provide taillamp, trunk lid/liftgate, right on assembly trolley	3-4
	5-3-5-4
4. Stamp stamp card, card 1	TBD
5. Position taillamp, trunk lid/liftgate, right in trunk lid	3-1-4
	3-5-4
6. Tighten taillamp, trunk lid/liftgate, right with nut 1x to trunk lid with cordless screwdriver	9
7. Plug in taillamp, trunk lid/liftgate, right electrically	7

8. Test taillamp, trunk lid/liftgate, right for tight fit	6
9. Insert taillamp right into bodywork	5-6
10. Plug in taillamp right electrically	7
11. Test taillamp right for tight fit	6
12. Push/pull assembly trolley	10
13. Provide taillamp left on assembly trolley	3-4 5-3-5-4
14. Provide taillamp, trunk lid/liftgate, left on assembly trolley	3-4 5-3-5-4
15. Stamp stamp card, card 1	TBD
16. Position taillamp, trunk lid/liftgate, left in trunk lid	3-1-4 3-5-4
17. Tighten taillamp, trunk lid/liftgate, left with nut 1x with trunk lid with cordless screwdriver	9
18. Plug in taillamp, trunk lid/liftgate, left electrically	7
19. Test taillamp, trunk lid/liftgate, left for tight fit	6
20. Insert taillamp left into bodywork	5-6
21. Plug in taillamp left electrically	7
22. Test taillamp left for tight fit	6
23. Position and click 3x cover, taillamp, trunk lid, left into trunk lid	1-6 5-6

Besides the sensor usage needs presented above also bending sensors on both toe joints of the worker will be used in order to enhance the motion animation.

4.2. Electrolux

The Electrolux case comprises different logistics tasks which are repeated for a single run. The following table presents the necessary motion types next to the relevant tasks.

Table 4.4: Loading trolley case

Task	Motion Types
1. Bring empty trolley in front of shelf	10

2. Read picking-list sequence with items to be loaded	8
3. Move to the virtual cell where the first item is stocked	1
4. Pick the item	3
5. Bring the item in front of trolley	2
6. Load the item on trolley	4
7. Repeat the sequence from point 2 to point 6 for all the items to be loaded on trolley	-

Besides the sensor usage needs presented above also bending sensors on both toe joints of the worker will be used in order to enhance the motion animation.

5. CAPTURING HUMAN MOVEMENT SAMPLES FOR MOTION SYNTHESIS

5.1. Motion capture system

The sample collection motion capturing system is used to capture human motion samples as an input to the statistical analysis of the motion synthesis. Thus, it is necessary to capture motions in a consistently high quality, mainly because the synthesis algorithms are very sensitive to common motion capturing shortcomings like gapped marker tracking (e.g. through self-occlusion of markers) or noise and marker jitter.

As current markerless MoCap technologies are not able to deliver data in this quality range, it is necessary to use a professional MoCap system for this purpose. However, this is only necessary for the sample collection step within the INTERACT process, while there will be used markerless capturing methods later in the workshop environment.

The setup that was installed at IMI for this task consists of a NaturalPoint OptiTrack motion capturing system, which is equipped with 16 NaturalPoint Flex13 MoCap cameras (120fps, 1280x1024 each). The system is marker-based, which means that the capture subjects are required to wear a specialized marker suit. The cameras are equipped with an infrared illumination unit, which is attached concentrically around the camera lens, to illuminate the retroreflective markers on the MoCap suit. Further, the cameras are able to do preprocessing and segmentation internally, which allows to connect the desired amount of MoCap cameras to a single workstation computer for capturing and processing in real-time. For this, a software named “Motive”, provided by the MoCap system manufacturer, is used. The process following the capturing takes is lined out below in section 5.3.

Figure 5.1: Optitrack Flex13 cameras mounted in the capture lab

For the setup, a separate laboratory was installed at IMI in Ulm, with approx dimensions of 7 x 6 x 2,7 meters. For an easier and more flexible setup, a truss construction was set up, which allows to place the cameras at nearly every position around the capture volume, to accommodate for different capture situations.

Figure 5.2: One half of the MoCap lab at IMI

The markerset used on the suit for full body MoCap was taken from the Motive software, and consists of 49 markers including hand markers. See the following figure for an overview of the marker placement.

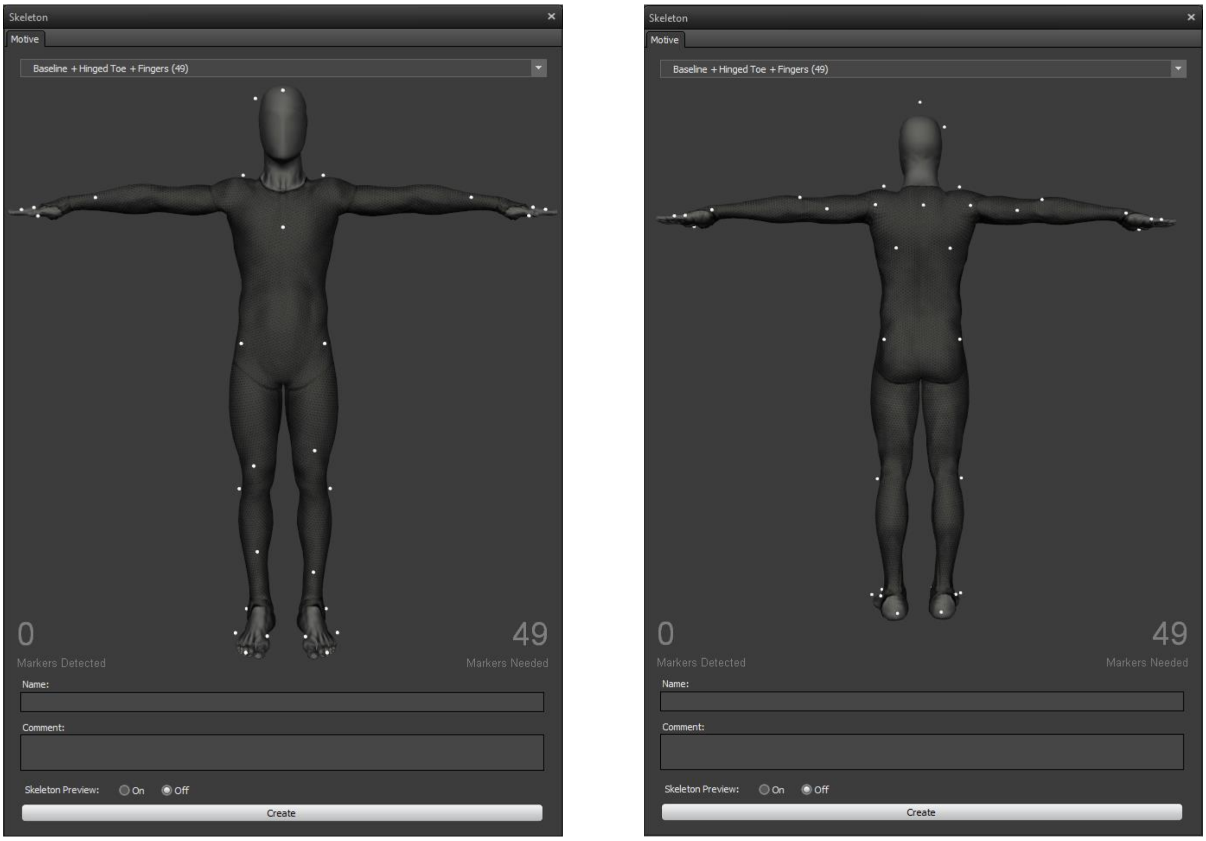

Figure 5.3: Sample collection marker setup

5.2. Motion sample collection takes

As a result of the first motion synthesis module developments and the application scenario, a list of different motion types was created (see D3.1.1 and D 1.1.1). For each of those motions, it is necessary to capture multiple samples, with the exact number and parameters depending on the type of motion.

For this, the given list is expanded to a take list for every take that has to be captured, and which can be imported directly into the MoCap software, to ensure completeness and consistent naming of the resulting MoCap sample files.

Each of the sample recordings generates a bvh and c3d file, to which postprocessing is applied afterwards.

Motion type	Description	Motion Primitives (Proposal)	D1.1.1 Reference	Parameters	Sensing WORKSHOP	Sensing LAB	Range or set	Distribution	No. samples per proband	No. Probands	No. samples total
Walk	Comprises right and left stances incl. Starting and Stopping, only comprises way lengths > 1 m	Walk Right Stance Walk Left Stance	22	Curvature [°/m] Start angle (iff curvature=0) [°] Stop angle (iff curvature=0) [°] Velocity [m/s]	Optical, foot-floor contact	Optical	[-0.09] [0;90] [0;90] [1;2]	NegExp Uniform Uniform Uniform	28	3	84
Carry	Same as walk, however only with velocity of 1m/s and an object being attached to one or two hands.	Carry Object Position Carrying Carry Right Stance Carry Left Stance	10	Curvature [°/m] Start angle (iff curvature=0) [°] Stop angle (iff curvature=0) [°] Object size & form	Optical, foot-floor contact, Object IMU, hand pressure for carrying	Optical	[-0.09] [0;90] [0;90] {screw, small single hand, large both hands, sheet, cylinder}	NegExp Uniform Uniform Uniform	63	3	189
Pick	Action grapsing an object while standing next to it. The action starts when ...	Pick Reach Pick Grasp	1	Object height [m] Distance worker object grip points [m]	Optical, foot-floor contact, Object ...	Optical	[0;2] [0, 0.7]	Uniform Uniform	120	3	360

Figure 5.4: Motion primitives-based capture planning list

Capture Subject	Number	Curvature	Start angle	Stop angle	Take Name	Completed?
Subject1	1	0	0	0	Subject1_1_0_0_0	No
Subject1	2	0	0	0	Subject1_2_0_0_0	No
Subject1	3	0	0	0	Subject1_3_0_0_0	No
Subject1	4	0	0	0	Subject1_4_0_0_0	No
Subject1	5	0	0	0	Subject1_5_0_0_0	No
Subject1	6	0	0	0	Subject1_6_0_0_0	No
Subject1	7	0	0	0	Subject1_7_0_0_0	No
Subject1	8	0	0	0	Subject1_8_0_0_0	No
Subject1	9	0	0	0	Subject1_9_0_0_0	No
Subject1	10	0	0	0	Subject1_10_0_0_0	No
Subject1	11	10	10	30	Subject1_11_10_10_30	No
Subject1	12	20	10	70	Subject1_12_20_10_70	No
Subject1	13	20	90	60	Subject1_13_20_90_60	No
Subject1	14	30	90	0	Subject1_14_30_90_0	No
Subject1	15	30	0	40	Subject1_15_30_0_40	No
Subject1	16	0	30	80	Subject1_16_0_30_80	No
Subject1	17	10	20	70	Subject1_17_10_20_70	No
Subject1	18	0	50	80	Subject1_18_0_50_80	No
Subject1	19	0	0	20	Subject1_19_0_0_20	No

Figure 5.5: Planning list expanded to take list

5.3. Motion capture postprocessing

The postprocessing of the motion data is necessary to achieve the best possible skeletal tracking quality out of the raw captured marker data. It consists of multiple different steps (**Figure 5.6**):

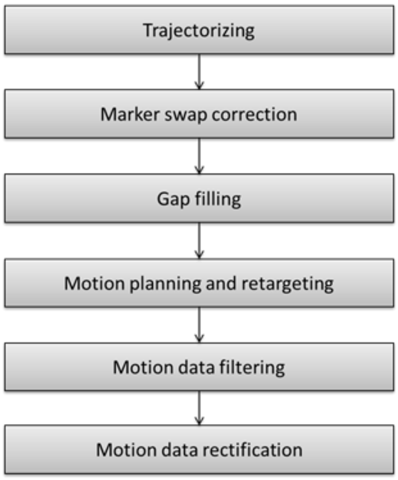

Figure 5.6: Steps of MoCap data translation to Interact BVH data

- Trajectorizing

Inferring a skeletal configuration out of the raw marker cloud. This is done in the Motive tracking software which is part of the Optitrack mocap system. Here, a frame with optimal view of all necessary markers is selected, which is then used to compute the follow-up positions of all markers, and their affiliation with the different body parts.

- Marker swap correction

Often, markers become interchanged during phases of occlusion. This step helps to find and correct such errors.

- Gap filling

During occlusions, naturally no data is recorded for certain markers. If these periods of occlusion are short enough, it is possible to estimate or extrapolate the movement during the occlusion phases, either through simple mathematical approaches like spline interpolation, or through more complex methods like model-based gap filling.

- Motion planning and retargeting

In this step, the motion is planned on skeleton used in the previous steps (the translations of the different end effectors are calculated) and then the translations of the relevant joints are retargeted to the skeletal structure used in INTERACT (RocketBox skeleton). After this, the motion is re-planned in order to generate the relevant BVH motion data corresponding to the RocketBox skeleton.

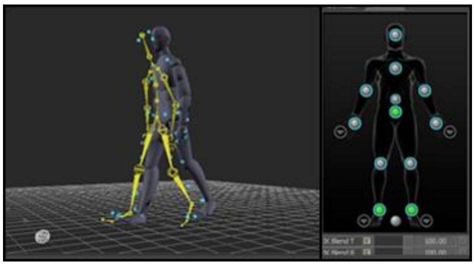

Figure 5.7: Retargeting of motion to the RocketBox skeleton

- Motion data filtering

Sometimes due to sensor data inaccuracy, the motions are generated with jitters which cause unnatural motions and have to be filtered out. In order to do that the jitters are located in the generated data and filtered out of the relevant joints. Low pass and smoothing filters are used to remove them but the relevant joints/ bones have to be selected since the application of these filters in the whole skeleton would result in the exclusion of “normal” motions from the rest of the skeleton.

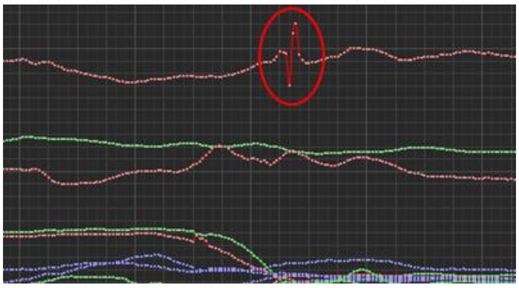

Figure 5.8: Motion data jitter

- Motion data rectification

Finally, the motion data file (BVH) is configured using the specifications described in the previous section. A motion file editor is used where the channels initially generated can be translated into the relevant ones specified for INTERACT.

6. CONCLUSIONS

In this deliverable, the sensor system design for INTERACT has been laid out and the sensor data interchange format has been described. Since preliminary tests had shown that premium motion capture quality is crucial for the chosen Morphable Graph approach. Requirements have proven to exceed current capabilities of low cost sensing systems even when combining them in a multi sensor fusion approach. Therefore, a separate marker based Motion Capture system has been set up for motion synthesis. In-workshop sensing has been described in section 4 and 5 while motion synthesis sensing has been described in section 5. An overview of trackable parameters has been provided. These parameters have been categorized into body tracking as well as tool and generic object tracking.

For the parameters, technical monitoring solutions have been presented and discussed. The solutions have been mapped onto the motion types that had been presented in deliverable D1.1.1 Requirements on efficient manual assembly model generation and interaction because sensor information is required for cutting these motion types for the best fit simulation in WP2 (task 2.2),

- Ergonomic assessment in WP2 (task 2.3),
- Movement type classification in WP4 (task 4.3) and
- Coping with occlusions that occur when motion capturing in the vicinity of obstacle in WP6 (Task 6.2).

It has been shown that a multi sensor approach as envisioned in the beginning of the project is still considered the best solution for each movement type. The reason for this is both spatial and temporal accuracy that is required for sensor data fusion and synchronization as well as optimizing the best fit simulation without manual intervention by an expert.

It is worth noting that the presented approach focuses on information that is stored inside production verification workshops. Tracking of human motions takes place only in a special area in the workshop room that is visualized with lines on the floor. Therefore, no hidden monitoring of workers is expected to take place.

GLOSSARY

EMG	Electromyography
MEMS	Microelectromechanical Sensor
WSNt	Wireless Sensor Network
WSN	Wireless Sensor Node
WSBS	Wireless Sensor Base Station
IMU	Inertial Measurement Unit
API	Application programming interface

REFERENCES

[1] P. K. Atrey, M. A. Hossain, A. E. Saddik, and M. S. Kankanhalli, “Multimodal fusion for multimedia analysis: a survey,” *Multimedia Systems*, vol. 16, no. 6, pp. 345–379, Nov. 2010.

[2] M. Meredith, S.Maddock, “Motion Capture File Formats Explained”, 2001.