Managing Affective-learning THrough Intelligent atoms and Smart InteractionS

D.5.7 Description of the interactive whiteboards layer

Workpackage	WP5 – Platform Agents
Editor(s):	Enrique Quirós Fernández (ATOS)
Responsible Partner:	ATOS
Quality Reviewers	Lazaros Gymnopoulos (CERTH) Brieuc Posnic (AV) Clément Rodrigues-Viguer (DXT)
Status-Version:	Final –V1.0
Date:	Project Start Date: 01/01/2016; Duration: 36 months Deliverable Due Date: 31/12/2016 Submission Date: 27/01/2017
EC Distribution:	RE
Abstract:	This deliverable contains details about the interactive whiteboards (IWB) as platform agent supporting the interaction of learners with the MaTHiSiS system, especially the design and implementation of applications that will run on it.
Keywords:	IWB, Platform agent, MaTHiSiS
Related Deliverable(s)	D2.2 Full scenarios of all use cases (M9) D2.3 Full system architecture(M6) D5.1 Description of robotic layer (M12) D7.2 MaTHiSiS platform, first release (M12) D5.8 and D5.9 Description of the interactive boards layer (M21, M33) D4.1 MaTHiSiS sensorial component (M12) D3.7 Learners' Profile Repository (M12)

This document is issued within the frame and for the purpose of the *MATHISIS* project. This project has received funding from the European Union's Horizon 2020 Programme (H2020-ICT-2015) under Grant Agreement No. 687772

Document History

Version	Date	Change editors	Changes
0.1	02/12/2016	Enrique Quirós, ATOS	Initial version of the document.
0.2	09/01/2017	Enrique Quirós, ATOS	Revised version according to first round of internal revisions
0.3	16/01/2017	Enrique Quirós, ATOS	Revised version according to second round of internal revisions
1.0	27/01/2017	Enrique Quirós, Carmen Padrón Napoles, ATOS	FINAL VERSION TO BE SUBMITTED

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
AIR	Affect and Intent Recognition
ASC	Autism Spectrum Case
CGDLC	Career Guidance Distance Learning Case
CLS	Cloud-based Learner's Space
DoA	Description of Action
EE	Experience Engine
ES	Experiencing Service
ITC	Industrial Training Case
GUI	Graphical User Interface
IPA	Interaction with Platform Agent
LA	Learning Action
LM	Learning Material
LMC	Learning Material Controller
LRS	Learning Record Stores
MEC	Mainstream Education Case
PA	Platform Agent
PMLDC	Profound and Multiple Learning Disabilities Case
SC	Sensorial Component
TB	TurtleBot robot
xAPI	Experience API
YX	Y1 First yearly working period, Y2 Second yearly working period

Table 1: Definitions, Acronyms and Abbreviations

Project Description

MATHISIS is a 36 month duration project co-funded by the European Commission Horizon 2020 Programme (H2020-ICT-2015) under Grant Agreement No. 687772. It started on 1st January 2016.

One of the core objectives of MaTHiSiS project is to enhance learning environments and make use of computing devices in learning in a more interactive way, which will provide a product-system to be used in formal, non-formal and informal education. An ecosystem for assisting learners/tutors/caregivers for both regular learners and learners with special needs will be introduced and validated in 5 use cases: Autism Spectrum Case (ASC), Profound and Multiple Learning Disabilities Case (PMLDC), Mainstream Education Case (MEC), Industrial Training Case (ITC) and Career Guidance Distance Learning Case (CGDLC).

MaTHiSiS product-system consists of an integrated platform, along with a set of re-usable learning components (educational material, digital educational artefacts, etc.), which will respond to the needs of a future educational framework, and provide capabilities for: i) adaptive learning, ii) automatic feedback, iii) automatic assessment of learner's progress and behavioural state, iv) affective learning and v) game-based learning.

Within MaTHiSiS, an innovative structural tool of learning graphs is going to be introduced to guide the learner through the process of learning in the given scenario. To reach a learning objective, learner will have to "follow the path" of the learning graphs, built up on Smart Learning Atoms, which are certain learning elements that carry defined learning materials.

To ensure barrier free integration in the market, MaTHiSiS makes use of a range of interaction devices, such as specialized robots, mobile devices and interactive whiteboards. The consortium ensures easy-to-use solution with e.g. specialized graphical editor-like tool, allowing to easily create educational materials as well as the reusability within both mainstream education and vocational training setups.

Objectives of the project

A Cloud-based Learner's Space (CLS) will be developed to provide a system for adaptation/personalization in learning, interaction, data acquisition and analysis as well as content creation on the fly. This is a core component of the MaTHiSiS system which includes 3 crucial subsystems which create an innovative smart learning ecosystem: i) the experience engine, a graph-based interactive storytelling engine, that manipulates interactive content that is later sent to a device of tutor's/learner's choice; ii) the learning graph engine, responsible for adaptation of the Learning Graph based on learner's behaviour and interaction; iii) the Decision Support System (DSS) providing directives for personalization and adaption based on the affective and performance information retrieved from learning analytics, and controlling synchronous and asynchronous interaction between devices. To ensure constant educational flow and augmented learner engagement, the emotion recognition and context aware cognitive/behavioural status extraction tools are introduced within the system addressed by the Sensorial Component (CS).

For the purpose of validating MaTHiSiS approaches in learning environment, a set of Smart Learning Atoms (SLA) is going to be created for defined use cases. Such SLAs will adapt to each learner in a different way based on her/his particular needs, profile, cognitive affective state, relevance to specific learning requirements and previous performance. Further, an editor-like tool is introduced to be able to transform educational material into MaTHiSiS Learning Materials usable by SLAs through Learning Actions. The learning graphs then are going to be deployed to interact with the CLS as well as some front-end tools for tutors and caregivers to enable creation, editing and authoring of the learning contents and learning experiences.

MaTHiSiS will support learning across a variety of learning contexts and, with the use of a variety of devices (robots, interactive whiteboards, mobile devices and desktop/laptop computers), with personalized and adaptable, time and location independent learning paths, being transferred between the agents, always taking into consideration best knowledge and practices learnt from the previous device.

By the end of the project, MaTHiSiS will introduce a marketable innovation, aimed at the re-usability of educational and training content and fostering the interactivity between technology and learners/tutors/caregivers.

Executive Summary

This document presents information about the Interactive whiteboards layer requirements and capabilities, and how the MaTHiSiS general platform agent architecture is being implemented for this platform agent. Those are the main results of activities carried out in task T5.3 (Interactive Whiteboards Layer) during Y1.

The document also explores the potential of IWB in terms of learning action materialization capabilities as it is one of the interactive devices (Platform Agents) with more possibilities for creating engaging learning experiences for the learners.

This document describes the main features (components and functionalities) of the first version of MaTHiSiS IWB layer, which are going to be tested on the driver pilots (M13-M16). As result of the first real learners' interaction with IWB agents, some modifications are expected to be considered in the next stage of the IWB developments to adjust its current design to the actual needs of the pilot participants. Consequently, the Conclusion section presents a set of next actions to be conducted in the next working period.

1. Introduction

1.1 Document context

In the MaTHiSiS platform a set of Platform Agents (PA) will be supporting the development of the learning process, namely robots, mobiles and IWB. Even though every PA can be very different to the others in appearance and functionalities, they share common logical architecture and services. So we could see every PA as different modules that share the same core in MaTHiSiS.

Task 5.3 Interactive Whiteboards Layer focuses on the creation of the applications for this PA that manage the materialization of learning actions and gather information to determine the affective and cognitive state of learners according to the MaTHiSiS concept.

Interactive whiteboards are equipped with cameras, microphones, speakers, touch screen and are compatible with Kinect, so are compatible with most of the Sensorial Component modalities managed in MaTHiSiS, except the ones that require sensors as accelerometer or gyroscope.

Because of this, IWB is a Platform Agent has a huge potential to be used in different educational scenarios. Touch screens, cameras, lots of sensors and a powerful processor make IWB flexible interactive devices, capable of working with different types of LM.

1.2 Document structure

This document is the first version of the set of deliverables (D5.7, D5.8 and D5.9) dedicated to describe the results of activities conducted in Task 5.3 and the implementation details of the IWB Platform Agent in MaTHiSiS. D5.7 presents initial results of Task 5.3 during the Y1 of project: first version of IWB layer which will be integrated in the first release of the MaTHiSiS platform. The following sections of this document describe the Interactive White Board Platform Agent design and its first implementation.

- **Chapter 2** presents the IWB layer technical and functional requirements derived from the set of MaTHiSiS objectives defined in the project DoA as well as from the analysis of the requirements extracted from the uses cases defined in D2.2.
- **Chapter 3** includes details about the Architecture of IWB as Platform Agent in MaTHiSiS.
- **Chapter 4** is dedicated to detailed describe the IWB layer specification and main functionalities as well as the current status of those for their integration into the first release of the MaTHiSiS platform.
- **Chapter 5** focuses on presenting the IWB capabilities for the materialization of learning actions considering IWB capabilities for delivering web-based and native app based learning materials.
- And last, the **Chapter 6** presents some conclusions about the work conducted during the first year in Task 5.3 and the set of actions to carry out during the next years.
- Annexes are dedicated to provide more information about the IWB design.

2. IWB layer requirements

This chapter focuses on determining the main requirements a IWB should meet to serve as Platform agent in MaTHiSiS according to the project objectives and the set of initial requirements established in the MaTHiSiS Description of Action. Sections 2.2.1 and 2.2.2 present the elicitation of IWB technical and functional requirements considering the project objectives and initial requirements, and as result of analysing the requirements extracted from the different UC in WP2 as described in D2.2 Full scenarios of all use cases [2] and the user stories included in D7.1 - Integration Strategy and planning [4].

2.1 Objectives as per MaTHiSiS DoA

The IWB layer and its implementations should contribute to the following main objectives of MaTHiSiS according to the DoA[1].

- SO2-1: Emotion recognition tools for image, video and 3D acquisition devices, brought to the needs of formal, informal and non-formal education for both mainstream and special education (T4.1).
- SO2-2: Emotion recognition tools for speech signals brought to the needs of formal, in-formal and non-formal education for both mainstream and special education (T4.1).
- SO2-3: Context aware Cognitive/Behavioural status extraction tools (T4.2).
- SO2-4: To develop a learner cognitive state monitoring framework and system towards constant education flow and augmented learner engagement (T4.3)
- SO2-5: To develop a personalized reaction system able to adapt the platform agents to the learner’s needs based their history, context-based and learner-based predefined information (T4.3) – To be developed in future versions.
- SO5-1: Foster learning adaptability and interoperability in different contexts and through different devices (T6.4) – To be implemented in future versions of the platform.
- SO5-2: Eliminate the need for learning only in specific locations and place individual and learning preferences as central to the design of learning experiences (T6.4) – To be implemented in future versions of the platform.
- SO5-3: Develop novel machine-2-machine (M2M) interaction schemes for knowledge transfer (T6.3, T6.4) – To be implemented in future versions of the platform.
- SO8-2: To provide seamless interoperability of the feedback system with the overall system so as to dictate learning actions to the platform agents (WP5) – To be implemented in future versions of the platform.

2.2 Requirements extracted from the use cases (as per MaTHiSiS DoA)

The main requirements established in the MaTHiSiS Description of Action (DoA) related to the IWB layer are the following:

- Req1: Achievement of the objectives for the innovation “Anthropocentric Interaction Schemes. Affect/Intent-aware machines”.
- Req2: Develop of applications and interfaces for delivering learning material to the right extent, through the right mediator (Agent).
- Req3: Implementation of a common language for information sharing (established in Task 4.2).

[1] DoA MaTHiSiS – Part B, page 7

- Req4: The Platform Agents component is responsible for the different interaction Platform Agents (PAs) that MaTHiSiS will deploy. These are robots (Task 5.1), mobile devices (Task 5.2) and IWBs (Task 5.3), which will provide a broad application potential to the MaTHiSiS ecosystem and will warrant efficient ubiquitous learning across a variety of educational contexts. At the PA level, the platform agents will implement three major sub parts: 1) platform agents' interface and on-board modules (WP5), 2) inter-units collaboration modules (WP6) and 3) PA, CLS information and action communication (T3.3, WP5).
- Req7: Synchronous collaboration among platform agents. MaTHiSiS has to support online, distributed synchronous collaboration, in multi-learner settings. To avoid obstacles related to physical location of the learners, platform agents of heterogeneous nature will be possible to exchange information and collaborate in order to converge, in real time and, based on learners' profiles, to optimized sessions. (T6.3, WP6)
- Req8: Asynchronous collaboration among platform agents: a learning scenario, designed to be executed by a robotic tutor, can execute the same or a similar task with the use of a mobile device without extensive re-training and customizing. (T6.4, WP6)

Previous requirements are of conceptual nature and served as reference for the definition of more concrete technical and functional requirements which their completion will contribute the achievement of the MaTHiSiS objectives.

2.2.1 Technical requirements

The following technical requirements were extracted from Use cases in D2.2 Section 6.3[2]. As result of their analysis all the duplicated requisites have been removed from this list:

ASD Use Case

- The system must work smoothly;
- The system must be multiuser;
- The system must allow the teacher to monitor the sessions and access the feedback;
- The system must work who won't maintain gaze, can't hold the device or be more separated from it than the usual mobile device.

PMLD Use Case

- Hardware must be robust;
- The system must have assistive technologies;
- The system must work without the need to setup.

Mainstream Use Case

- IWBs must be compatible with promethean Active inspire software.

Industrial training Use Case

- System must be compatible with different IWB brands.

CGDL Use Case

- Audio must be able to sound loud.

Next in the section 4 the IWB layer is described in terms of capabilities and implementation taking into consideration the previous enumerated technical requirements as well as the functional ones, which are presented in the next section.

2.2.2 Functional requirements

According to the description of use cases and the Annex II-storyline document of D2.3 [1], the IWB PA (as any PA in MaTHiSiS) is required to support the following user stories, defined for the first release in D7.1 [4]. In the next table components (according to the MaTHiSiS architecture defined in D2.3) responsible for support the defined stories are also presented.

User story	Description	
PC02	As a Tutor, I want to be able to define my PAs in order to have them ready for learning experiences / sessions.	LES, PC
PC03	As a Tutor, I want to be able to define my already known learning environments with their PAs in order to have them ready for learning experiences / sessions.	LES, PC
LES03	As a Tutor, I want to define the learning context before the beginning of a Learning Session in order to define which learner will use which PA and select the current learning environment.	LES, PC
LES12	As a Learner, I will interact with my preferred PA in order to improve my skills and it will gather all information about my current affective status and progress in order to trigger real -time Learning Graph adaptation.	ES
LES18	As a Learner, I want to practice using touch screens or waving in front of the robot in order to interact with these PAs.	ES

Table 2: User stories related to PA

The functionalities required to be implemented in order to support these user stories are defined in the following table.

ID	Functionality
F1	Establishment of communication with the CLS for the authenticated/authorized users.
F2	Communication of the IWB application and the CLS for the identification and delivery of the learning material to be offered to the learner
F3	Communication of the IWB application and the CLS for the delivery of sensor data to enable affect detection
F4	Communication of the IWB application and the CLS for the delivery of performance data
F5	Synchronous/asynchronous collaboration between platform agents of same/different type

Table 3: Required functionalities to be implemented in the whiteboard agents

3. Architecture

Platform Agents will be used to support learning experiences while providing learners with learning action materializations and extracting their affect state in order to be able to do the adaptation and personalization of the learner's learning experience using MaTHiSiS. One of the aims at defining this layer is to ensure the extensibility of the platform facilitating the introduction of new kind of Platform Agents in the future, without the need of making big modifications in the design of MaTHiSiS platform. The table describing the general IWB design has been included in the Annex IWB design.

Next the common platform architecture is presented along with its main components: The Experiencing services and the Learning Material framework.

The first one is responsible for providing learners with the materials required from the deployment of the learning as well as gathering data sensor information to determine the affective state of the learner while interacting with the learning materials.

Meanwhile the second one, Learning Material framework is in charge of controlling the execution of LM as part of the materialization of the Learning Actions and tracking learning interactions

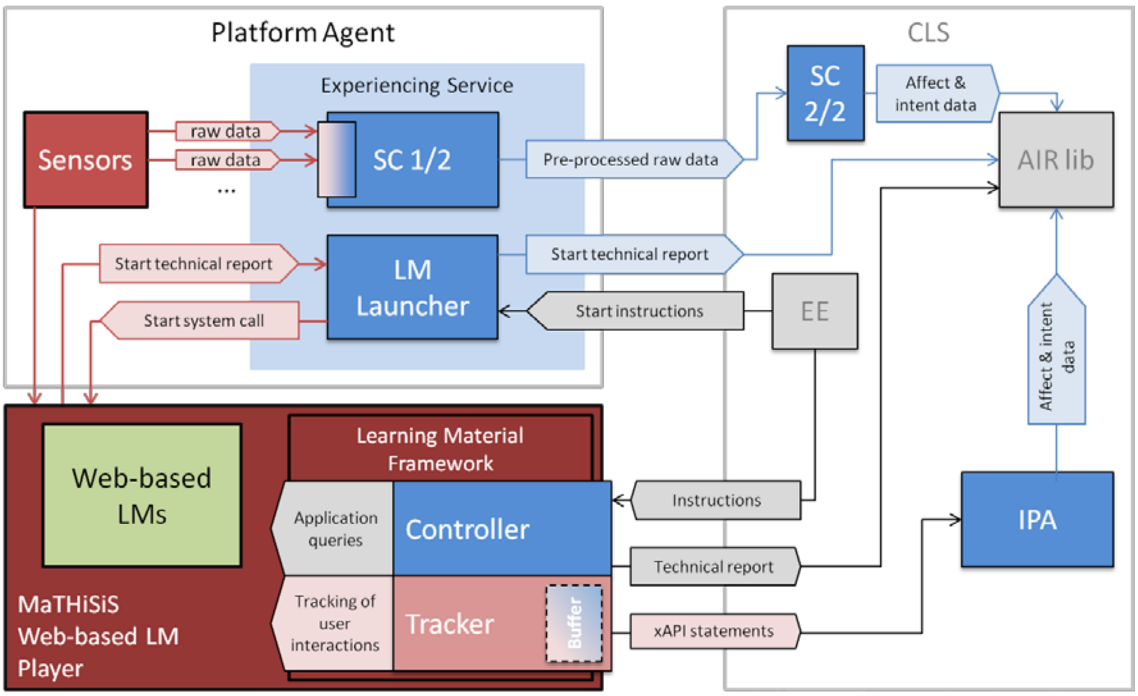

Figure 1: Experiencing Service architecture for IWB: Web-based digital Learning Materials

3.1.1 Experiencing service

The MaTHiSiS Experiencing service of all the Platform Agents communicates the DSS through the AIR Lib Open API on the CLS. This service is implemented in every PA and allows it to exchange information with the cloud system. Other duties of this service is to receive data from the interaction tracker, retrieval of data from the sensors, support the learners' interaction with the LM and finally sending back feedback to the MaTHiSiS backend.

3.1.1.1 Sensorial component

The sensorial component is in charge of the analysis of the data provided by the different sensors available in the platform agent to determine the learner's affective state.

There are 5 types of Sensorial component modalities in MaTHiSiS. From them **Facial expressions analysis** and **Gaze estimation** can be covered by the 2 built in cameras of the IWB to capture images that later would be processed. **Skeleton Motion Analysis** couldn't be done the same way with the built in cameras, but the IWB currently used for MaTHiSiS (i.e. Surface Hub) is fully compatible with Kinect and it could be used as complement to be able to use this Sensorial component modality.

For **Speech recognition and speech-based affect recognition** the IWB counts with a good quality microphone array so tracking sounds from the user wouldn't be a problem the same applies for voice recognition.

In the case of **Mobile device-based emotion recognition** we have a limitation as the IWB doesn't count with accelerometer or gyroscope. That means that tracking any tasks that require inertia sensory data won't be possible. Any tactile requirements will work as it counts with touch screen.

3.1.1.2 Learning Material Launcher (LM launcher)

This component is in charge of starting the application associated with the specified LM. This component has been introduced to do the bridge between the Experience Engine that will send materialization instructions and the concrete LMs, i.e. native applications or web-based for the IWB layer. Also, if there are any problems during the LM execution will retrieve feedback, so the ES can inform the DSS-AIR Lib and act in consequence. In the case of the IWB, the LM launcher will be dealing with web based materials in the first release of the platform. For future releases, native application support will be also considered.

The LM Launcher receives START instructions from the EE, to concretely start a new learning material, i.e. in the IWB the LM launcher is a specific web application that will allow learners to interact with the material. If something went wrong (web resource is not found, application crashed at start, etc.), the LM launcher informs immediately the cloud about the problem, asking to find another materialization of the specified learning action. Even if we ensure by other means that such problems will not occur, this possibility is taken into consideration.

3.1.1.3 Learning Material Framework

This component allows MaTHiSiS to control the different learning materials used for the materialization of specific learning actions through the interpretation of materialization instructions given by the Experience Engine into concrete requests/queries to the learning material container application. It is also in charge of tracking learners' interactions with the LM which will be used to determine the cognitive status (and performance) of the learners. In the case of IWB, the framework is implemented as web-application and the tracker of interactions with materials is a JavaScript Wrapper included in the application.

Once the application is launched a direct bi-directional connection is opened between the web application and the back-end. This allows the EE to send other instructions (e.g. STOP instruction) to the application and to let the application, through the LMF, to send the interactions of the learner to the IPA component. To that purpose, a set of xAPI statements is used, describing the stream of interactions carried out by the learner. All generated xAPI statements follow the structure and design as defined in Annex 8.1.

The first of the trackers of the IWB catches the interactions that the learner does while watching a video LM. It registers when the user starts, stops, pauses and how much time it takes in finish to watch it. This video tracker is already available at the repository. https://gitlab.atosresearch.eu/ari/mathisis/tree/master/Front-end/MaTHiSiS%20Main%20Platform/app/modules/videoTracker

This web based component implemented as JavaScript will run all tasks related to gathering cognitive responses that the learner generates while interacting with the specified LM. That includes interactions information and how was the learner performance (scores, events, etc.) related to the learning activities performed. This component will create xAPI statements for each of the interactions tracked and will send them to IPA.

3.1.1.4 Interaction with PAs tier

The Interaction with PAs tier (IPA) component is responsible of processing the information about the learners' interactions (xAPI statements) in order to facilitate the extraction of learners' performance which will be sent to the DSS-AIR lib to be analysed with the goal of proposing different adaptations of the learning experience which might be required according to the learner's performance. In the initial system architecture described in the D2.3 - Full system architecture M6 [1], the IPA was foreseen to be embedded in the Experiencing Service on each PA. But since this component had no specificities per device and mobiles have strong memory restrictions, it is now part of the MaThiSiS back-end.

3.1.2 Platform agent collaboration tier

This tier will be implemented as part of the T5.3 activities in Y2. It will be responsible for implementing synchronous and asynchronous collaboration scenarios to be defined as part of WP6 activities. They will be described in D6.4.

4. IWB layer description

4.1 IWB Layer specifications

Taking into account the set of technical requirements described in the section 2.2.1, the IWB layer should have a set of minimal features to act as Platform agent in MaTHiSiS. Those minimal features are a camera, microphone, touchscreen, speakers, Internet connection and a web browser.

The Surface Hub is being used as development/testing IWB PA during the project lifetime, it exhibits the previous set of minimal features as well as the following ones.

The Surface Hub has been designed as high quality video conferences device, but for many reasons is perfect for learning purposes. It has a big screen of 55'' and full HD resolution. It is fast as it uses SSD hard drive, counts with 8GB RAM and a good processor.

The specifications relative to sensors are:

- Kinect compatible
- 2 wide angle HD cameras 1080p @ 30fps
- Infrared presence sensors
- Ambient light sensors
- Touch screen
- Stereo speakers
- Microphone array

Those specifications are more than enough to cover any of the requisites listed earlier. The fast processor and RAM will make the system go smooth (Req1 listed in Section 2.2). It is not expected to be any problems to have many users or allowing the teacher to monitor the sessions and consult the Learner's feedback.

Whiteboards are able to let the learner with special requirements use them more separated than other devices, and it's not necessary to hold it, so it complies with Req4.

The hardware is robust (Req5), as it is made with resistant materials.

The system won't have any needs of setup (Req7) as the software will be web based applications and will be transferred to the existing IWB through the MaTHiSiS configuration process. Also, the IWB will be compatible with assistive technologies that will be added to MaTHiSiS front end.

For being compatible with Promethean Active inspire software (section 12.3.2 of D2.2 [2]), the operating system of the IWB must be Windows (7 or later), Mac OS (10.7 or later) or Ubuntu (12.04 or later). As the Surface Hub runs on Windows 10 pro the compatibility should be covered. As for the referred software, we are going to consider it as a native app for LM and we are going to conduct a set of test to verify the compatibility.

The system will be compatible with different IWB brands (Req9), but those IWB will have to cover a set of minimum specifications. For example, to have at least one camera will be needed for most of LMs in order to evaluate the affective state of the learners.

Finally, "the audio must be able to sound loud" (Req10) won't be a problem as the surface count with a pair of big stereo speakers. The device has been created for enterprise videoconferences so the sound system is high quality.

There is the possibility of having several learners per LM, as the IWB counts with 2 wide cameras that will record all the details. This capability won't be implemented for the first platform release and will be evaluated its implementation feasibility for the second release. As presented on section 3.1.2, the IWB is able to play LMs than relies on tactile function. The cameras can support any LM that required Facial expression analysis or Gaze estimation. The IWB array of 4 high performance microphones is able to gather the required information for the Speech recognition and speech-based affect recognition as far as the MaThiSiS SC will be implemented in the IWB According to section 5.3 of D4.1 [7]. Also, the touch screen will provide a lot of freedom for creating LMs that needs tactile information.

4.2 IWB main functionalities

The main technical functionalities that will be supported by this Platform Agent are:

- Video and audio signals recording on the device: IWB can record video and audio signals through webcam and/or Kinect camera. This data will be used by the specific Sensorial Component.
- Video and audio streaming over the network: the signals recorded using webcam and/or Kinect camera can be streamed to another computer.
- Multimedia content: IWB can play different multimedia content which will be used as learning material such video, audio, PowerPoint presentations, web-based resources, etc.
- Touch screen: the learner inputs are simply and intuitively supported by the existing touching screen.

4.2.1 Current development status

Next we explain the main functionalities, considering the set of system roles defined in MaTHiSiS and described in section 3.3 of D2.3 [1], which are currently available for the IWB PA.

IWB Experiencing Service functionalities available for the MaTHiSiS platform first release are:

- Authentication functionalities (Tutor/Learners) (fulfilling requirement F1 in Table 3). It has been implemented using the User API described in D3.7 [8]
- Learning experience supervisor (Tutor)
- Browse (Tutor/Learner)
    - For Tutor: it allows to retrieve all Learning experiences designed or created by the tutor.
    - For the Learner it allows to retrieve all his/her learning experiences (as presented in Figure 1).
- LM launcher allows the Learner to access to the learning experience selected the recommended by the Experience Engine. (F2 in Table 3)

As presented in the next Figure, the Browse functionality of the IWB Experiencing service presents to the Learner all his available learning experiences in the platform, let be on-going experiences or those which has been already completed

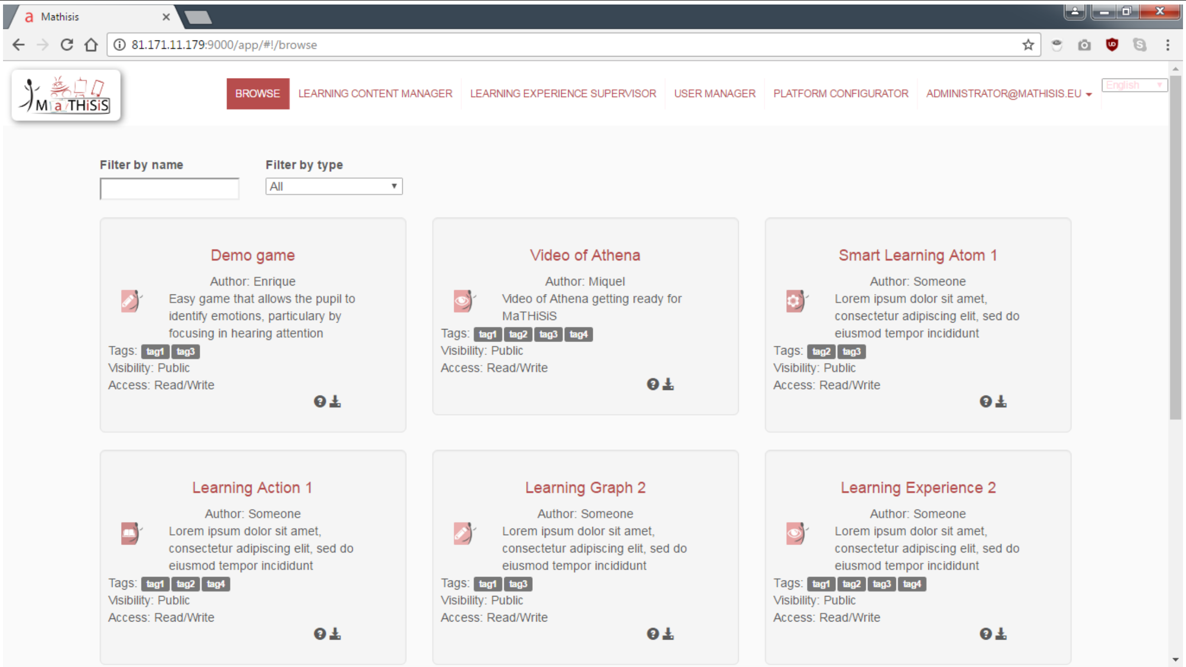

Figure 1 - IWB Experiencing service initial UI: visualizing a list of learning experiences

In the case of IWB the LM launcher and the Learning Material framework are implemented as web application which has been developed with Angular. It launches the learning material recommended by the Experience Engine, informs on the current status of the application to the back-end (AIRLib); and allows learners to access and interact with the learning materials while tracking his/her interactions.

Next figure shows the final stage of a game (LM) accessible through the LM launcher:

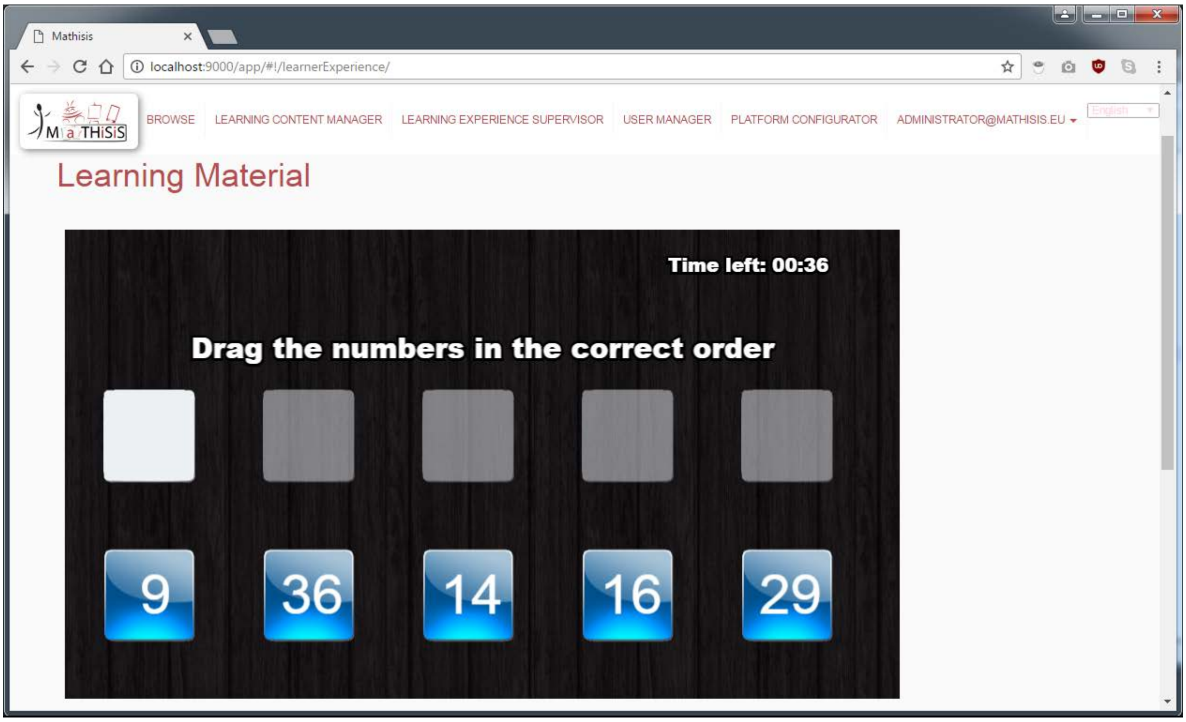

Figure 2 – IWB LM launcher: Presentation of "Feelings quiz" LM

Currently the Sensorial component is based on the implementations provided by WP4 (task 4.1). it is responsible for data gathering to ensure Facial expressions analysis, Gaze estimation, Speech recognition and speech-based affect recognition. The Skeleton Motion Analysis is not currently supported but the first version of the SC is expected to be completed by the end of M13.

All information about the IWB layer code is accessible through the GitLab as referenced in D7.2 section 2.3.1.6 [5].

5. Learning Action materialization capabilities

5.1 Examples of web-based learning materials

Next we present examples of web-based learning materials implemented for the Mainstream UC to support the learning experiences designed in section 13.3.2 of D2.2[2] for the Learning goal: Mathematical competences: numbering and the SLA (learning objective) Discrimination of greater than / less than.

All code information of these materials is accessible through the GitLab as referenced in D7.2 [5] section 2.3.2.4.

5.1.1 Game: Which is the biggest number?

This simple game is a web-based learning material implemented for a learning experience of the UC Mainstream which learning goal is "Mathematical competences and numbering" and the specific SLA: "Discrimination of greater than / less – Identify the largest / smallest number between two numbers" according to the design presented in the above mentioned section of D2.2[2]

The system displays two different numbers and asks the pupil to indicate which is the biggest/smallest. The pupil taps on which he/she thinks is the correct number.

Mechanics

- 2 big numbers appear in the middle of the screen
- The screen asks "which is the BIGGEST number?" or "which is the SMALLEST number?"
- The learner taps the number he thinks it's correct, then the system will show feedback to the learner
- There are 2 types of sounds "correct" and "wrong", which are played to enforce the feedback.
- As suggested in the Technical meeting held in Maastricht by educators and pedagogic partners, the game will show a happy or sad face to reinforce the feedback provided to the learner.

Figure 3 - First version of learning material Which is the biggest number game

Key moments of interaction to be tracked of

- game start
- wrong answer
- learner wins
- time is out

Scenarios + xAPI

Game starts: The teacher presses the “Start” button and the game starts according to the level that was selected by the platform.

xAPI statement
*{"**actor**": { "type": "learner", "name": "Codified name", "mbox": "mailto:learner@example.com" }, "**verb**": { "id": "http://adlnet.gov/expapi/verbs/initialized", "display": { "en-US": "initialized" } }, "**object**": { "id": "http://example.com/activities/bigger-number", "definition": { "name": { "en-US": "Find the bigger number!" }, "activity type": "assessment" } }, "context": { "location": "Classroom id", "SLA": {"id": "SLA id", "name": "Discrimination of greater than / less than", "SLA weight": "0.3" }, "LA": {"id": "LA id", "name": "Identify the biggest/smallest number between two numbers"}, "LG": {"id": "LG id", "name": "Find the bigger number!" } }, "**Result**": {"Completion": false, "success": false}, "**timestamp**": "2016-11-14T21:15:30+01:00"}*

Figure 4 - Which is the biggest number game?- Main screen

User wins: The student answers correctly and receives positive feedback.

xAPI statement
*{"**actor**": { "type": "learner", "name": "Codified name", "mbox": "mailto:sally@example.com" }, "**verb**": { "id": "http://adlnet.gov/expapi/verbs/passed", "display": { "en-US": "passed" } },*
*"**object**": { "id": "http://example.com/activities/which-is-the-biggest-number", "definition": { "name": { "en-US": "Which is the biggest number?" }, "activity type": "assessment" } },*
*"**context**": { "location": "Classroom id", "SLA": {"id": "SLA id", "name": " Discrimination of greater than / less than ", "SLA weight": "1" }, "LA": {"id": "LA id", "name": " Identify the biggest/smallest number between two numbers " "}, "LG": {"id": "LG id", "name": " Which is the biggest number?" " } },*
*"**Result**": {"Completion": true, "success": true, "score": { "time-needed":”7s”}},*
*"**timestamp**": "2016-11-14T21:15:30+01:00 "}*

Figure 5 - Which is the biggest number game?- Victory screen

Wrong answer: The student answers wrong and receives feedback.

xAPI statement
{"**actor**": {"type": "learner", "name": "Codified name", "mbox": "mailto:sally@example.com" }, "**verb**": { "id": "http://adlnet.gov/expapi/verbs/failed", "display": { "en-US": "failed" } },
"**object**": { "id": "http://example.com/activities/which-is-the-biggest-number", "definition": { "name": { "en-US": "Which is the biggest number?" }, "activity type": "assessment" } },
"**context**": { "location": "Classroom id", "SLA": {"id": "SLA id", "name": " Discrimination of greater than / less than ", "SLA weight": "1" }, "LA": {"id": "LA id", "name": " Identify the biggest/smallest number between two numbers " "}, "LG": {"id": "LG id", "name": " Which is the biggest number?" " } },
"**Result**": {"Completion": false, "success": false, "score": { "time-needed":"7s" },
"**timestamp**": "2016-11-14T21:15:30+01:00 "}

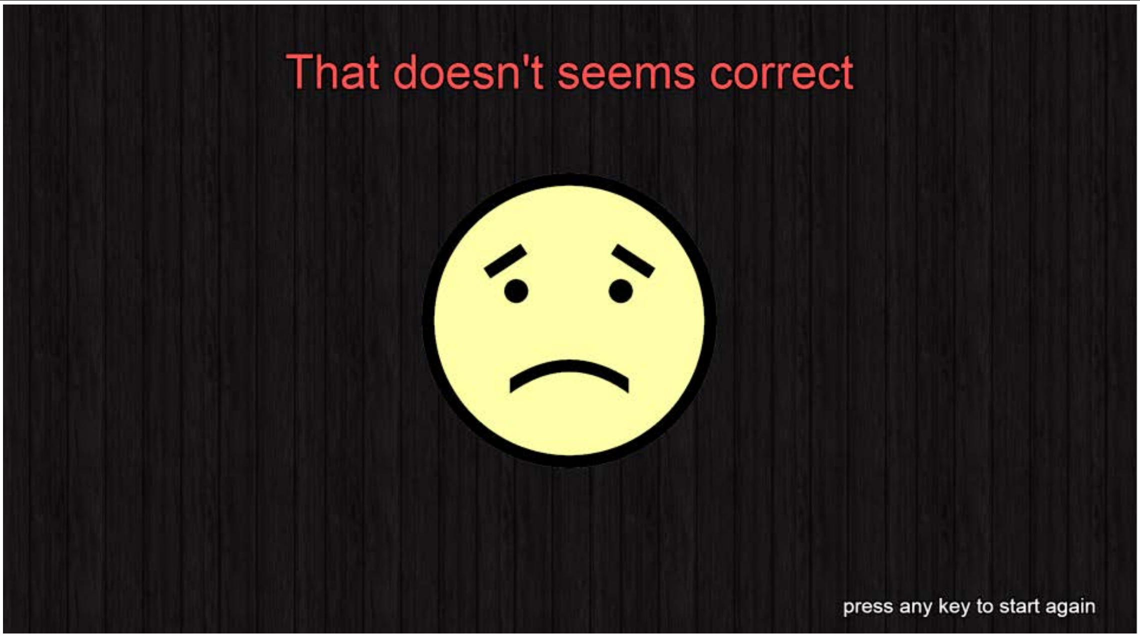

Figure 6 - Which is the biggest number game?- Wrong answer screen

Time is out: The student didn’t answer in a long time.

xAPI statement
*{"**actor**": {"type": "learner", "name": "Codified name", "mbox": "mailto:sally@example.com" }, "**verb**": { "id": "http://adlnet.gov/expapi/verbs/abandoned", "display": { "en-US": "abandoned" } },* *"**object**": { "id": "http://example.com/activities/which-is-the-biggest-number", "definition": { "name": { "en-US": "Which is the biggest number?" }, "activity type": "assessment" } },* *"**context**": { "location": "Classroom id", "SLA": {"id": "SLA id", "name": " Discrimination of greater than / less than ", "SLA weight": "1" }, "LA": {"id": "LA id", "name": " Identify the biggest/smallest number between two numbers " "}, "LG": {"id": "LG id", "name": " Which is the biggest number?" " } },* *"**Result**": {"Completion": true, "success": false, "score": { "time-needed":”7s” },* *"**timestamp**": "2016-11-14T21:15:30+01:00 "}*

Figure 7 - Which is the biggest number game?- Time is out screen

5.1.2 Ordering numbers

This simple game is a web-based learning material implemented for a learning experience of the UC Mainstream which learning goal is Mathematical competences and numbering and the specific SLA: “Discrimination of greater than /less - Put the numbers in order“ according to the design presented in D2.2[2]

- On the screen there are numbers in a random order. The pupil has to order these numbers correctly.

Mechanics

- Random unsorted numbers on the screen. In the top there is a row with as many empty cells as numbers are in the screen.
- The screen asks "Drag the numbers in the correct order"
- The learner drags the numbers to the row.
    - If the learner got every number in the correct order the game ends
    - Every time the learner put a number in a wrong place there is a “wrong” sound and the number will move to a new random spot. This wrong try will be noted.
    - Every time the learner put a number in the right place will sound a “correct” sound.
    - If the time runs out, the game ends

When the game ends the system will show feedback to the learner.

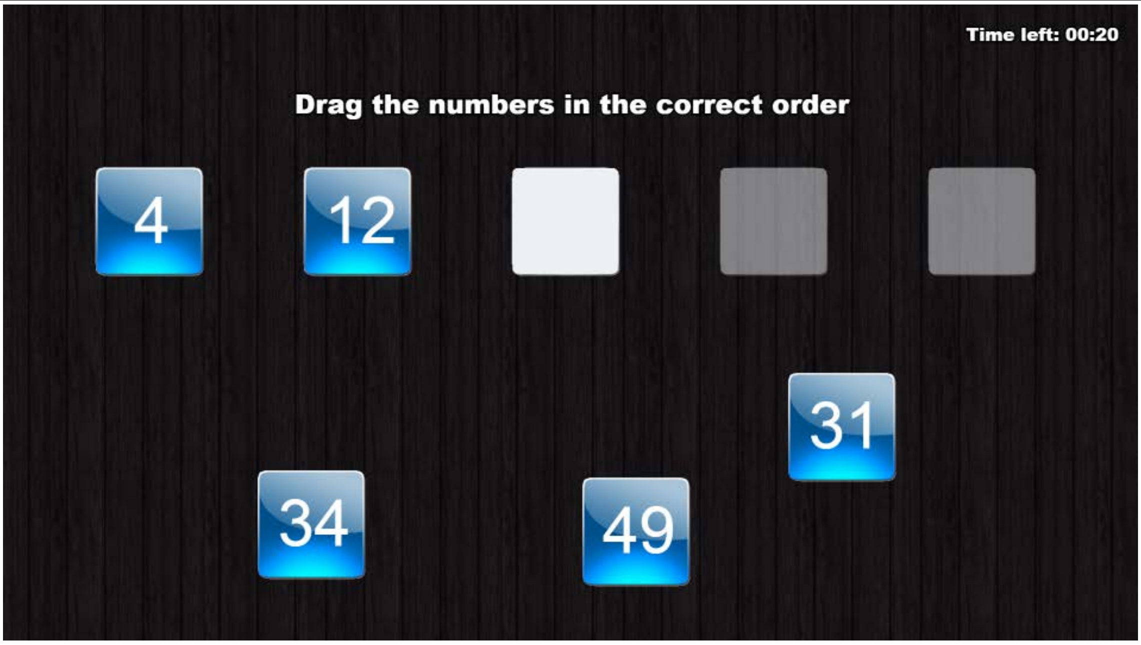

Figure 8 - First version of learning material Ordering numbers

Key moments of interaction to be tracked

- LM initialises
- game starts
- learner gives a wrong answer
- time is out (gameover)
- learner wins

Scenarios + xAPI

Game starts: The teacher presses the "Start" button and the game starts according to the level that was selected by the platform.

xAPI statement
*{"**actor**": { "type": "learner", "name": "Codified name", "mbox": "mailto:learner@example.com" }, "**verb**": { "id": "http://adlnet.gov/expapi/verbs/initialized", "display": { "en-US": "initialized" } }, "**object**": { "id": "http://example.com/activities/bigger-number", "definition": { "name": { "en-US": "Find the bigger number!" }, "activity type": "assessment" } }, "context": { "location": "Classroom id", "SLA": {"id": "SLA id", "name": "Discrimination of greater than / less than", "SLA weight": "0.3" }, "LA": {"id": "LA id", "name": "Ordering numbers"}, "LG": {"id": "LG id", "name": " Ordering numbers" } }, "**Result**": {"Completion": true, "success": true}, "**timestamp**": "2016-11-14T21:15:30+01:00"}*

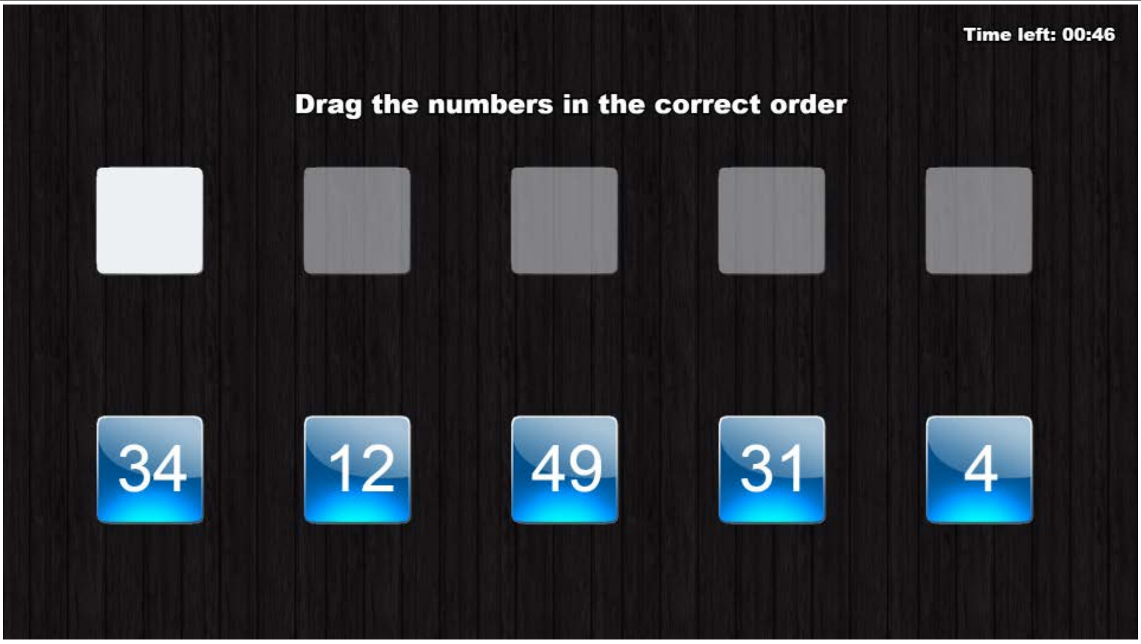

Figure 9 - Ordering numbers game -Main screen

User wins: The student answers correctly and receives positive feedback.

xAPI statement
*{"**actor**": { "type": "learner", "name": "Codified name", "mbox": "mailto:sally@example.com" }, "**verb**": { "id": "http://adlnet.gov/expapi/verbs/passed", "display": { "en-US": "passed" } },* *"**object**": { "id": "http://example.com/activities/ordering-numbers", "definition": { "name": { "en-US": "Ordering numbers" }, "activity type": "assessment" } },* *"**context**": { "location": "Classroom id", "SLA": {"id": "SLA id", "name": " Discrimination of greater than / less than ", "SLA weight": "1" }, "LA": {"id": "LA id", "name": " Identify the biggest/smallest number between two numbers " "}, "LG": {"id": "LG id", "name": " Ordering numbers" " } },* *"**Result**": {"Completion": true, "success": true, "score": { "time-needed":"7s"}},* *"**timestamp**": "2016-11-14T21:15:30+01:00 "}*

Figure 10 - Ordering numbers game -Victory screen

Wrong answer: The student answers wrong and receives feedback.

xAPI statement
*{"**actor**": {"type": "learner", "name": "Codified name", "mbox": "mailto:sally@example.com" }, "**verb**": { "id": "http://adlnet.gov/expapi/verbs/failed", "display": { "en-US": "failed" } },* *"**object**": { "id": "http://example.com/activities/ordering-numbers", "definition": { "name": { "en-US": "Ordering numbers }, "activity type": "assessment" } },* *"**context**": { "location": "Classroom id", "SLA": {"id": "SLA id", "name": " Discrimination of greater than / less than ", "SLA weight": "1" }, "LA": {"id": "LA id", "name": " Identify the biggest/smallest number between two numbers " "}, "LG": {"id": "LG id", "name": "Ordering numbers" " } },* *"**Result**": {"Completion": true, "success": false, "score": { "time-needed":"7s" },* *"**timestamp**": "2016-11-14T21:15:30+01:00 "}*

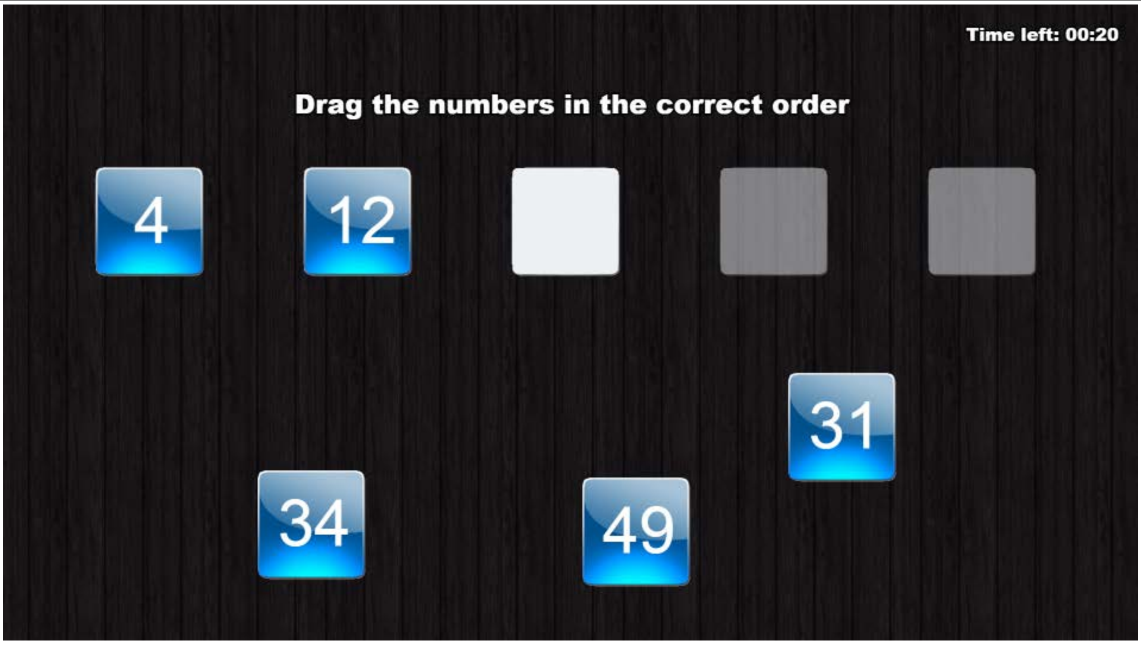

Figure 11 - Ordering numbers game- Wrong answer

Time is out: The student didn’t answer in a long time.

xAPI statement
*{"**actor**": {"type": "learner", "name": "Codified name", "mbox": "mailto:sally@example.com" }, "**verb**": { "id": "http://adlnet.gov/expapi/verbs/abandoned", "display": { "en-US": "time out" } },*
*"**object**": { "id": "http://example.com/activities/ordering-numbers", "definition": { "name": { "en-US": "Ordering numbers" }, "activity type": "assessment" } },*
*"**context**": { "location": "Classroom id", "SLA": {"id": "SLA id", "name": " Discrimination of greater than / less than ", "SLA weight": "1" }, "LA": {"id": "LA id", "name": " Identify the biggest/smallest number between two numbers " "}, "LG": {"id": "LG id", "name": Ordering numbers" " } },*
*"**Result**": {"Completion": true, "success": false, "score": { "time-needed":”7s” },*
*"**timestamp**": "2016-11-14T21:15:30+01:00 "}*

Figure 12 - Ordering numbers game -Time is out screen

6. Conclusion

This deliverable describes how components of the MaTHiSiS IWB platform agent are implemented according to the objectives of the project and the set of requirements extracted from D2.2. The implementations include the development of the Experiencing Service, the service that will allow the communication between the Platform Agents and the cloud with the rest of the components. Also, presents the types and indicative instances of the learning materials supported by IWB, including information and screenshots about the LM implementations.

Furthermore, it also described the technologies used for development, provided screenshots of the prototype and presented indicative print-screens of the developed User Interfaces following the guidelines prescribed by task 3.5. In the near future components presented in section 4 will be improved with more functionalities according to the MaTHiSiS platform second release plan and new LMs will be developed to support the learning activities to be conducted during the pilots.

Next actions include the improvement or inclusion of some components in subsequent versions. The list of future components to be implemented (but not limited to) is included below:

- Incorporation of new sensorial components;
- Incorporation of new functionalities of the Learning experiencing service and Learning material framework;
- Enhancements for the “interaction with PAs tier” in the Experiencing Service;
- Implementation of Platform Agents collaboration tier.

This deliverable corresponds to the first of a set of three deliverables resulting from the work carried out in task 5.3. Next versions, namely D5.8 and D5.9 will update the information included in this document, providing additional information about the new functionalities and Learning Action materialization capabilities will be available on M21 and M33, respectively.

7. References

[1] DIGINEXT (ed.): D2.3 Full system architecture M6. Deliverable of the MaTHiSiS project, 2016

[2] NTU (eds.): D2.2 Full scenarios of all use cases. Deliverable of the MaTHiSiS project, 2016

[3] UM (eds.): D5.1 Description of the robotic layer. Deliverable of the MaTHiSiS project, 2016

[4] Nadia Politou (ed.), D7.1 Integration strategy and planning, Deliverable of the MATHISIS project, 2016.

[5] DXT (eds.) D7.2 MaTHiSiS platform, first release (M12), Deliverable of the MaTHiSiS project, 2016

[6] ATOS (eds.) D5.8 and D5.9 Description of the interactive boards layer (M21, M33), Deliverable of the MaTHiSiS project, 2017 (due in M21 and M33)

[7] CERTH (eds.) D4.1 MaTHiSiS sensorial component, Deliverable of the MaTHiSiS project, 2016

[8] ATOS (eds.) D3.7 Learners' Profile Repository, Deliverable of the MaTHiSiS project, 2016

8. Annex

8.1 Information about xAPI

All information about the xAPI and its use in MaTHiSiS has been included in the Annex 8.1 of D5.1 [3]

8.2 Experiencing service and architecture

All information about the Experiencing service and the general PA architecture included in the Annex 8.2 of D5.1 [3]

8.3 IWB design

Tier Name	IWB - LED screen
Layer	Platform Agent
Inputs	
1	Source: EE Data: Learning Actions and transmedia content Schema: JSON
2	Source: Learners Data: Gestures, sounds, specific interaction with the PA
3	Source: MaTHiSiS frontend Data: learner profile data (demographics, special needs, etc.) Schema: JSON,XML
Outputs	
1	Destination: Learners Data: Specific behaviour through the PA interface
2	Destination: IPA Data: Raw signals from sensors Schema: JSON,XML
3	Destination: Experiencing service Data: Device identification and capabilities Schema: JSON,XML
Sensors / Hardware	
1	Camera
2	Microphone

3	Speakers	
4	Touchscreen	
5	USB or another type of ports	
6	Network connections	
Software		
1	Windows 10 (We will ensure compatibility with other OS)	
Technical abilities		
1	Low durability (touchscreen can be damaged)	
2	Heavy	
3	Can be used as a common computer (another PA)	
Constraints		
1	Reflection is a problem because the screen emits light by itself	
High – Level Information		Computed on
1	Classic interactions through the screen	Device
2	Affect State through motion-gesture detection	Sensorial Component

Table 4 IWB design