Project Identification	
Project number	No. 611421
Duration	1st Dec 2013 – 30th Nov 2016
Coordinator	Andreas Stainer-Hochgatterer
Coordinator Organisation	AIT Austrian Institute of Technology GmbH, Austria
Website	www.miraculous-life.eu

Miraculous-Life

Miraculous-Life for Elderly Independent Living

Document Identification	
Deliverable ID:	D3.3b Interface specification for the expressive speech and avatar user interface component
Release number/date	V03/11.12.2015
Checked and released by	Sascha Fagel (Zoobe)
Work Status	Finished
Review Status	Internal review

Key Information from "Description of Work"	
Deliverable Description	The report presents the specification of the interface to the Avatar creation.
Dissemination Level	PU=Public
Deliverable Type	R = Report
Original due date	Project Month 24 / 30. Nov 2015

Authorship& Reviewer Information	
Editor	Sascha Fagel (Zoobe)
Partners contributing	Zoobe
Reviewed by	Maher Ben Moussa (UniGe)

Release History

Release Number	*Date*	*Author(s)*	*Release description /changes made*
V01	10.12.2015	SF/Zoobe	Created based on D3.3a
V02	11.12.2015	MB/Maher	Reviewed Deliverable
V03	11.12.2015	SF/Zoobe	Final Adaptations

Miraculous-Life Consortium

Miraculous-Life (Contract No. 611421) is a project within the 7th Framework Programme. The consortium members are:

Partner 1	*AIT AUSTRIAN INSTITUTE OF TECHNOLOGY GMBH (AIT, Project Coordinator, AT)*
Contact person:	Andreas Stainer-Hochgatterer
Email:	andreas.stainer-hochgatterer@ait.ac.at
Partner 2:	*UNIVERSITY OF GENEVA (UniGe, CH)*
Contact person:	Maher Ben Moussa
Email:	maher.benmoussa@unige.ch
Partner 3:	*UNIVERSITY OF CYPRUS (UCY, CY)*
Contact person:	George Samaras
Email:	cssamara@cs.ucy.ac.cy
Partner 4	*ORBIS MEDISCH EN ZORGCONCERN (ORBIS, NL)*
Contact person:	Cindy Wings
Email:	c.wings@orbisconcern.nl
Partner 5	*FRAUNHOFER IGD (Fh-IGD, DE)*
Contact person:	Carsten Stocklöw
Email:	carsten.stockloew@igd.fraunhofer.de
Partner 6	*Noldus Information Technology BV (Noldus, NL)*
Contact person:	Ben Loke
Email:	b.loke@noldus.nl
Partner 7	*CITARD SERVICES LTD (Citard, CY)*
Contact person:	Eleni Christodoulou
Email:	eleni_christodoulou@cytanet.com.cy
Partner 8	*ZOOBE MESSAGE ENTERTAINMENT GMBH (Zoobe, DE)*
Contact person:	Sascha Fagel
Email:	fagel@zoobe.com
Partner 9	*MAISON DE RETRAITE DU PETIT-SACONNEX (MRPS, CH)*
Contact person:	Donato Cereghetti
Email:	donato.cereghetti@hotmail.com

Table of Contents

Release History *II*
Miraculous-Life Consortium *III*
Table of Contents *IV*
Table of Figures *V*
Abbreviations *VI*
Executive Summary *1*
1 About this Document *2*
1.1 Role of the deliverable 2
1.2 Relationship to other Miraculous-Life deliverables 2
2 Introduction *3*
2.1 Basic design principle 3
3 Animation System *4*
3.1 Animation generator 4
3.2 Text to Speech 4
3.3 Lip-sync 4
3.4 Rendering 4
4 Specifications *5*
4.1 Software presentation 5
4.2 Pre-sets 5
4.3 Avatar Media Converter Tutorial 6
4.4 Avatar Media Converter Output 7
References *8*
Appendix A Module Presentation GUI Interface *9*

Table of Figures

Figure 1: Sample source code for the client 7
Figure 2: GUI interface example 9
Figure 3: Video result example 9

Abbreviations

Abbrev.	*Description*
AAL	Ambient Assisted Living
VSP	Virtual Support Partner
ECA	Embodied Conversational Agent
TTS	Text-To-Speech

Executive Summary

This deliverable contains the description of the expressive speech and avatar user interface component that is delivered by the Miraculous-Life partners in WP3. It refers to the software deliverable D3.4b and specifies the architecture of the software as guideline for the prototype phases of the project. It is based mainly on the result of the questionnaire regarding the avatar appearance and VSP specification functionalities. This deliverable will be split into three parts. First it will summarize the basic design principles for the avatar user interface. Secondly, it will describe the pre-sets of the animation system. Thirdly, it will describe the interface of the software, showing what is available in terms of languages, characters and options.

The present document is based on the previous version of the deliverable and contains minor editorial changes. The Text-To-Speech system and the rendering server were maintained and adjusted according to requests concerning pronunciation of specific words, speech tempo, camera settings, colours of the clothes of the avatar, background (scene) details, and video format. Furthermore, the female avatar was extended by six sets of animations to enable the emotional expressions defined in D1.2b, and the according pre-sets were added which is documented in this deliverable. The basic functioning of the avatar interface itself does not differ from the one in version D3.3a.

1 About this Document

1.1 Role of the deliverable

This deliverable contains the description of the expressive speech and avatar user interface that is delivered by the Miraculous Life partners in WP3. The avatar interface is integrated in the prototypes to generate avatar output.

1.2 Relationship to other Miraculous-Life deliverables

The deliverable is related to the following Miraculous-Life deliverables:

Deliv:	*Relation*
D1.2	Specification of use case scenarios and User Interface: This document presents the use case scenarios and also an analysis of the interaction requirements needed to specify the Human-Computer interface. The avatar interface specified in the present deliverable defines the necessary expressions to cover the use case scenarios.
D3.4	The avatar software: This deliverable documents the software that creates avatar output according to the properties described in the present deliverable.

2 Introduction

2.1 Basic design principle

Modeling and designing a daily activities support system for elderly raises several research questions including the interaction between the user and the system, the computation and selection of verbal and non-verbal ways and their synchronization and representation resulting in emotional understanding. The objective is to simulate human-like interaction of the elder with a computer system . The elder should be given the feeling of communicating with a real human who understands both his/her emotional state and behavior and provides appropriate and familiar responses.

For the development of such an empathic support system, related interesting initiatives in the area of Human-Computer interaction have to be considered. Such initiatives address, among others, the development of Intelligent User Interfaces, like the Embodied Conversational Agent (ECA) (also called Synthetic Character or Virtual Character). An ECA provides an avatar based interface represented on the screen by a human, or cartoon like, face or body, on the audio channel via speech output, and aiming at being conversational in a human-like behavior for the generation of verbal and non-verbal output and also recognition and response to verbal and non-verbal input. Empirical studies (Ortiz, A et el., 2002) with elder people (i.e. normal aging, mild cognitive impairment) reveal strong evidence that ECAs could improve the interaction between elder people and machines: elder people, both with and without cognitive impairment, are capable of recognizing emotions in the facial expressions of the avatar. The elderly follows instructions much better when interacting with an avatar and finds the experience of having an emotional avatar as an interface a pleasant one. In agreement with the above findings (Nijholt, A, 2003) concluded that embodied agents allow the development of affinitive relationships with their human partners and can therefore help to fulfill the need of affiliation in ambient assisted living.

We present a system for automated animation of text messages suitable for avatar user interfaces. Input to the system is a text message. An avatar video is rendered and converted into an h.264 video by a back-end server system. The interface to this system and its integration into the multimodal dialogue manager are described in this document.

3 Animation System

This part consists of a global description of the Zoobe avatar media converter. The system is not provided as software but its description is useful for the client side interface we provide. Our approach is to define on the server side, some pre-sets (character, backgrounds, languages, optional settings) that could be called from the client to generate a video. Here, we describe first what are these pre-sets and then describe how we fuse video and audio part for dependent feature such as lip synchronizing.

3.1 Animation generator

The animation system gets its settings from pre-sets containing the following elements:

- The character model
- An animation style for the character
- The scenery
- The light set
- The camera settings
- The resolution of the resulting video file

Given a specific animation style that expresses a certain emotion, the system selects all suitable available animations and creates an animation sequence from it. This sequence is then used to animate the character in the given scene using light and camera settings into the specified video file format.

3.2 Text to Speech

As the interface provides a conversion from plain text to video that shows an avatar uttering this text, the language and the gender need also to be specified. The following parameters are implemented according to the chosen character:

- The language (English, Dutch, German or French)
- The gender (male or female; only female used in the present prototype)
- The speed of speech
- The voice activation (low, medium, or high) to modify the speech melody according to the arousal of the requested emotion

3.3 Lip-sync

Animation parameters for jaw opening, lip opening and lip spreading are calculated from a phonetic transcription of the given text. This transcription is provided by the text-to-speech module along with synthesized audio speech.

3.4 Rendering

The pre-set, the speech audio and the lip-sync parameters are used to render the avatar sequence. The sequence is then converted to an MPEG-4 video file with the given resolution.

4 Specifications

4.1 Software presentation

The software we deliver as D3.4b consists of a Java project library handling the communication between the Zoobe server and this client side application. This software is provided with a “Hello World” interface that firstly allows to show the software in an easy way, but mostly allows partners to know which are the main methods to use which will be described below. This section focuses first on the pre-sets available for this deliverable, then it introduces a tutorial showing how to use the library in few steps, and finally gives the output specifications.

4.2 Pre-sets

First of all the main method of this library needs to be called with a certain pre-set. This pre-set is represented by the parameters listed below:

- Bundle → identifies a set of stories (see below)
- Lang → language for the Txet-To-Speech synthesizer (TTS)
- Locale → regional variant of the language
- Gender → gender for the TTS
- Character → appearance of the VSP
- Story → animation style / expressive emotion of the VSP
- Stage → scene or background for the VSP
- Activation → Controls the level of arousal of the output speech (inherited from the story)

In the previous version of the present deliverable, two characters – one male and one female – in neutral animation style with their own background were supported, as shown in Table 1. These presets are dedicated for development and system integration purposes. Table 2 shows the supported languages, locales and activations for male or female characters. Table 3 shows the pre-sets added for the female character in order to enable the emotion expression defined in D1.2b.

Table 1: Initial VSP pre-sets specifications

Settings	**Old Man**	**Young Woman**
Bundle	5001	5001
Character	Albert	Alina
Story	50003 (neutral male)	50004 (neutral female)
Stage	61 (consultation room)	62 (living room)

Table 2: Voice pre-set specifications

Settings	**Male voice**	**Female voice**

Gender	male	female
Lang	en, de	en, nl, de, fr
Locale	de_DE , en_US	de_DE, en_US, fr_FR. nl_NL
Activation	low, medium, high	low, medium, high

Table 1: Initial VSP pre-sets specifications

Story	**Emotional state of the avatar**
50008	Happy
50011	Worried
50010	Relieved
50006	Compassionate
50007	Directive Behaviour
50009	Neutral

4.3 Avatar Media Converter Tutorial

With the parameters above all set the Miraculous-Life system requests via secure websocket communication to render an avatar video with a given text according to these parameters. Figure 1 shows an example how to embed the interface, define a pre-set, and call the avatar rendering.

```
//import requested packages
import java.io.UnsupportedEncodingException;
import eu.miraculouslife.core.media.conversion.ZoobeMediaConverter;
import eu.miraculouslife.core.media.conversion.impl.AvatarMediaConverter;

public class main {
    public static void main(String[] args) {
        //create a new media converter object
        ZoobeMediaConverter m_xZoobeServices= new AvatarMediaConverter();
        // set parameters
        m_xZoobeServices.setBundle("5001");
        m_xZoobeServices.setLang("de");
        m_xZoobeServices.setLocale("de_DE");
        m_xZoobeServices.setGender("male");
        m_xZoobeServices.setCharacter("Albert");
        m_xZoobeServices.setStory("50003");
        m_xZoobeServices.setStage("61");
        m_xZoobeServices.setActivation("medium");
        // set input - output
        String sInputText = "Super Geil!";
        String sResult = null;
        //do the conversion
        try {
            sResult = m_xZoobeServices.convert(
                    eu.miraculouslife.core.media.conversion.MediaConverter.VIDEO,
                    sInputText.getBytes("UTF-8"), "video/mp4").toASCIIString();
        } catch (UnsupportedEncodingException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        //print result
        System.out.print(sResult);
    }
}
```

Figure 1: Sample source code for the client

4.4 Avatar Media Converter Output

The method AvatarMediaConverter.convert takes as input:

- The output format file : Video or Audio
- The text message converted in bytes
- The output extension of the output file (mp3 or mp4)

As soon as the method is executed it returns a URL to retrieve the resulting video. The video can then be played via pseudo-streaming or saved on the client device.

References

[1] Fagel S, Hilbert A, Morandell M, et al. The virtual counselor —automated character animation for ambient assisted living. In:ACHI 2013, The Sixth International Conference on Advances inComputer-Human Interactions. 2013. p. 184—7.

[2] Schröder, M. and Trouvain, J., “The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching”, Intl. Journal of speech Technology 6, 365-377, 2003.

[3] CereProc, cServer Text-to-Speech Server http://www.cereproc.com/en/products/server [online resource, retrieved 18.10.2012]

[4] Löfqvist, A. “Speech as audible gestures”, In W. J. Hardcastle and A.Marchal (Eds.): Speech Production and Speech Modeling, NATO ASI Series, 55, Kluwer, Dordrecht, 289–322, 1990.

[5] Fagel, S. and Clemens, C., “An Articulation Model for Audiovisual Speech Synthesis - Determination, Adjustment, Evaluation”, Speech Communication 44, 141-154, 2004

[6] The Nebula Device. http://sourceforge.net/projects/nebuladevice

[7] FFmpeg. ffmpeg.org [online resource, retrieved 19.10.2012]

[8] ActionScript 3.0 Reference for the Adobe Flash Platform. http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/ [online resource, retrieved 02.01.2013]

Appendix A Module Presentation GUI Interface

As an example how to use the avatar interface we provide here screenshots of the GUI interface provided to present the module:

The GUI is separated into 4 panels. The first one allows a user to enter the text message that will be translated into voice. The second and the third one allow the user to select the language of the speech and the character to be used. The fourth one is the result panel. The buttons “get video” take all input and provide a video saved on the client device whereas “play media” play the video obtained. See just below a screenshot of the GUI and a screenshot of the resulting video.

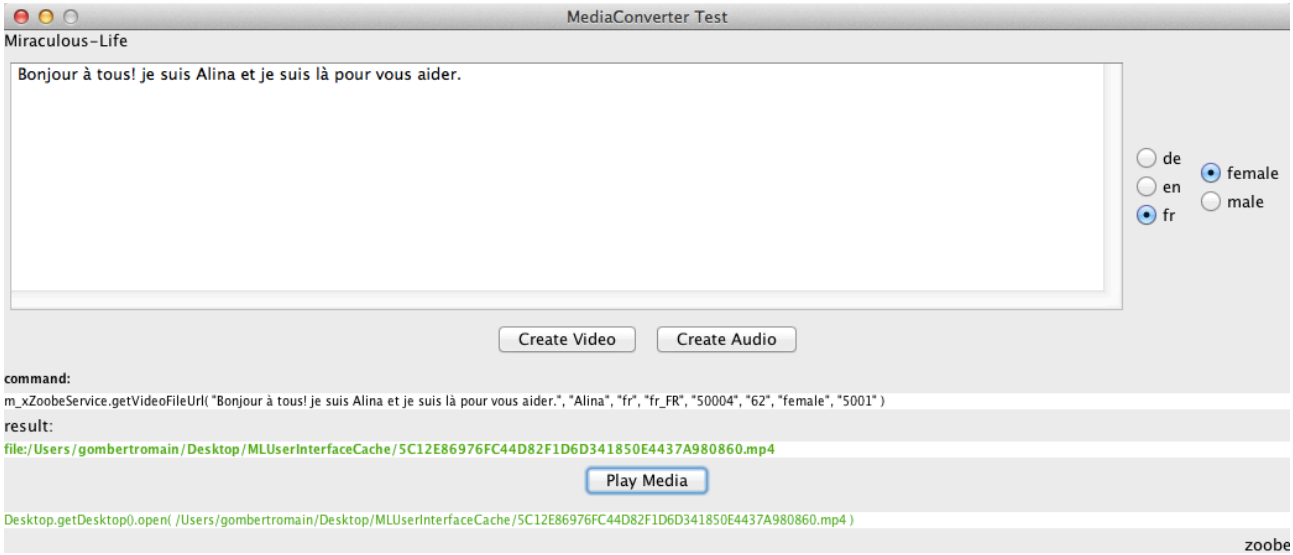

Figure 2: GUI interface example

Figure 3: Video result example