8. Web-based Embodied Conversational Agents and Older People

Gerard Llorach, Hörzentrum Oldenburg GmbH & Medizinische Physik and Cluster of Excellence 'Hearing4all', Universität Oldenburg, Germany

Javi Agenjo, Universitat Pompeu Fabra, Spain

Josep Blat, Universitat Pompeu Fabra, Spain

Sergio Sayago, Universitat de Barcelona, Spain

Within Human-Computer Interaction, there has recently been an important turn to embodied and voice-based interaction. In this chapter, we discuss our ongoing research on building online Embodied Conversational Agents (ECAs), specifically, their interactive 3D web graphics aspects. We present ECAs based on our technological pipeline, which integrates a number of free online editors, such as Adobe Fuse CC or MakeHuman, and standards, mainly BML (Behaviour Markup Language). We claim that making embodiment available for online ECAs is attainable, and advantageous over current alternatives, mostly desktop-based. In this chapter we also report on initial results of activities aimed to explore the physical appearance of ECAs for older people. A group of them (N=14) designed female ECAs. Designing them was easy and great fun. The perspective on older-adult HCI introduced in this chapter is mostly technological, allowing for rapid online experimentations to address key issues, such as anthropomorphic aspects, in the design of ECAs with, and for, older people.

8.1 Introduction

Embodied and voice-based interactions are increasingly important within Human-Computer Interaction. Speech is widely regarded as the most natural way for humans to communicate. Nowadays, it is possible to interact verbally with Intelligent Personal Assistants (IPAs), which provide assistance or companionship to a wide range of user groups, ranging from children (Druga et al. 2017) to people with disabilities (Pradhan et al. 2018). Examples of IPAs are Apple Siri, Microsoft Cortana, Google Assistant, Amazon Alexa, and social robots, such as Aldebaran Pepper. In the near future, it is expected that important elements of communication, such as emotions and nonverbal behaviors, and context, will be implemented, because "If these assistants communicate naturally with humans, then they essentially disappear as computers and instead appear as partners" (Ebling 2016).

Embodied Conversational Agents (ECAs) can be thought of as IPA variants, which have an anthropomorphic representation, typically a 2D/3D virtual humanoid. Their use has been growing in the last years, due to their potential benefits to society, and thanks to recent technology developments. Recent publications have examined the effectiveness of ECAs as pedagogical agents for learning (Lewis and Lester 2016) and as agents for detecting and preventing suicidal behavior (Martínez-Miranda 2017). Other recent publications deal with them in the context of group-meeting (Shamekhi et al. 2018) or Clinical Psychology (Provoost et al 2017). With older people, previous research on ECAs concentrates mostly on virtual worlds (Carrasco

2017), gesture-based avatar game (self-)representations (Rice et al 2016), and geriatrics (Chi et al. 2017).

Recently, within the KRISTINA project (Wanner et al. 2017), we explored ECAs in two important and increasingly common health scenarios in Europe: a) to provide health support to a migrant with very little knowledge of the local language when the physician or nurse do not speak the migrant's language and b) to provide assistance and basic medical recommendations to older adults.

Building on this work, in this chapter we discuss our ongoing research on building the interactive 3D web graphics aspects of ECAs, showing that the goal of making embodiment available for IPAs, and turning them into online ECAs, is attainable, which makes them more accessible. ECAs are very complex systems integrating different components, and have been mostly developed and deployed in desktop platforms. Based on part of our work, we also present the results of research activities aimed to explore the design of ECAs by older people, using free online character editors, to answer the question of how an ECA should look like for this user group, as well as promoting their transition to content producers (Ferreira et al. 2017, Guo 2017) one step forward: from text, pictures, videos or programs, to 3D agents.

8.2 Creation and design of 3D virtual characters

In the context of the increasing maturity of the 3D interactive web (Evans et al. 2014), this section focuses on the potentialities that a web approach could bring to the wider adoption of ECAs. ECAs as desktop applications present drawbacks for both developers and users: developers must create an ECA for each device and operating system while end-users must install additional software, limiting their widespread use. Llorach and Blat (2016) have demonstrated, through the integration of the *Web Speech* API, *Web Audio* API, *WebGL* and *Web Workers*, together with novel work in the creation and support of embodiment through *WebGLStudio* (Agenjo et al. 2013), the possibility of a simple fully functional web-based 3D ECA accessible from any modern device.

In the following subsections we review a number of open-source tools to create humanoid realistic virtual characters. We also describe the main requirements and modifications for the virtual characters to be ready for the interactive web.

8.2.1. Tools to create 3D virtual characters

Designing virtual characters from scratch is a very time consuming task. There exist tools that allow us to create humanoid characters in less than an hour. We review open-source tools (*MakeHuman*, *Autodesk Character Generator*, *Adobe Fuse CC* with *Mixamo* and *Daz Studio*) that provide some high-level control and flexibility over the physical appearance and behavioral properties of the character - including its expressivity via interactive facial and full body animation. We compare them with respect to basic functionalities, such as geometry control and exporting options, and more advanced aspects for the believability of the characters, such as clothes and hairstyles (see Table 8.1 and Figure 8.1).

MakeHuman is currently a free desktop program. Several general controls for visual features such as ethnicity, age and gender and several exporting options make it suitable for creating custom characters easily and quickly. The geometries of the body and face can be controlled in detail with sliders. There are several options for configuring the skeleton, including complex ones with facial rig, which are necessary to be able to properly animate a character. The resulting character can be exported in several formats. The *.mhx2* format offers more features (rigs, blend shapes) when opened with *Blender*, an open source 3D Computer Graphics (CG) tool, and its specific plugin. One limitation could be the insufficient variety of clothes and hairstyles, which are key aspects in making the character believable, although user community provided non-curated assets, scripts and plugins could help fix it.

Autodesk Character Generator is a web-based editor to create characters. This means that no software has to be downloaded and that characters can be accessed and modified online. A large set of clothes and hairstyles is available as well as different body textures to change their ethnicity. Their geometry is based on the mix of two predefined ones and fine grain control is not possible. The characters' clothes, body and hair are exported as one mesh with one material, great for simple applications, but not good enough for more realistic-looking characters, where different materials and shaders are required for hair, skin and clothes, for example. The textures include color, normal and specular maps for better quality rendering. The exporting options do not cater for as many formats as *MakeHuman* does, but one can choose the quality of the mesh, to include facial blend shapes or facial rig and two different skeleton models. This software is especially useful to create crowd characters or for users with little expertise (no installation required and few exporting options).

Adobe Fuse CC is a desktop application that starts from real human 3D scans to create a model. The geometries of the selected scan can be finely controlled by means of sliders or via the mouse pointer directly on the model, a quite natural geometry modification procedure. The assets provided are quite customizable: the colors and patterns of the clothes, eyes and hair can be changed, offering a wide range of choices. There are many textures for each mesh: (diffuse) color, gloss (roughness of the surface), normals, opacity and specularity, allowing users to customize and improve the quality of the final rendering. A newly created character is exported through *Mixamo*, a web platform where animations from a database containing more than 2,000 can be applied to the characters and pre-visualized. The exporting options are quite simple but enough for web high-quality. Several facial blend shapes are included as well as different skeleton configurations.

Daz Studio offers high quality and realistic virtual characters with a facial rig and detailed textures. Nevertheless, modifying specific visual and geometric features is not allowed, just editing some general features of a predefined character. Most assets and features to create and modify the characters are not free, and the tool becomes expensive if several characters are required. It focuses mainly on the realistic rendering of scenes with virtual characters.

Additionally, instead of modeling a character, a real person can be scanned (based on depth cameras, LIDARs or from simultaneous still images) and transformed into a virtual character. There are current tools, such as *SmartBody* (Feng et al. 2015), whose output includes automatic rigging and facial expressivity. Several issues strongly limit their use for quality interactive web applications, such as their inability to separate the geometries with different materials (such as eyes, hair, skin, clothes, etc.). At their present stage, using them takes more effort and time to achieve web-ready characters than the aforementioned tools.

Table 8.1. Comparative analysis of different tools to create virtual humans

	High-level geometry control	**Detailed geometry control**	**Clothes, hairstyles**	**Textures**	**Facial blend shapes and skeleton**	**Realism**
MakeHuman 1.1.0	++ Ethnicity, age, gender	+ Feature modification through sliders	- Small set	Diffuse, normal map	+ Facial bone rig	+ Medium
Autodesk Character Generator	+ Blending between two predefined geometries	- Blending of predefined geometries	+ Medium-to-large set of hairstyles and clothes	Diffuse, normal map, specular map	+ Facial bone rig or facial blend shapes	- - Low
Adobe Fuse CC 2017.1.0	- 3D scans of real humans	++ Feature modification through sliders or directly on the mesh	++ Highly customizable medium-to-large set of hairstyles and clothes	Diffuse, normal map, specular map, gloss map, opacity map	+ Facial blend shapes	++ Medium-to-high
Daz Studio 4.10 Pro	- - Predefined models	- - Feature modification through sliders	- - Very small set, but dependent on the predefined model	Diffuse, opacity map	+ Facial bone rig	++ High

Figure 8.1. Pictures of characters created (with, from left to right, Makehuman, Autodesk Character Generator, Adobe Fuse CC and Daz Studio) and rendered on the web. The quality of the final rendering is only indicative, as it could be improved using custom shaders and additional textures when possible.

8.2.2 Web requirements

In non-real-time scenarios without time constraints, virtual characters can have highly detailed geometries and photorealistic effects: the rendering of a frame movie can take several hours, whereas real-time applications can afford just some milliseconds. Thus, interactive web characters have specific requirements, and we discus the most important ones next.

File size. The size of a desktop-based ECA matters little as it is stored in a hard drive and quickly retrieved, while for the web, the character file must be transmitted to the browser client and decompressed there in little time (Evans et al. 2018). The size of the surface mesh[1] of the character is directly correlated to the file size, and a highly detailed mesh does not permit interactivity even after having been loaded.

[1] Surface meshes are most usual representations in 3D graphics; alternatives are voxels, for medical and engineering applications, or unstructured point clouds coming from 3D scanners.

Facial expressions with blend shapes. Blend shapes are commonly used to deform the geometry of the mesh, in our case, the facial geometry to create expressions. Each blend shape has the same number of displacement vectors as vertices of its corresponding mesh, and increases the file size. Thus, web applications can only consider a limited number of blend shapes.

Facial expressions with facial bone rigs. Using bones to deform the face gives much more flexibility than using blend shapes. Each bone can be translated, rotated and scaled affecting its associated vertices with different weights. This flexibility results in more complexity when creating facial expressions. While a single blend shape value is enough to show a happy face, multiple bones with their translations, rotations and scales would need to be adjusted. Moreover, this is very application and character dependent. Blend shapes can be constructed as a combination of bone transformations, but we worked with tools that support directly blend shapes, limiting complexity in the web side. Our real-time web-based 3D engine currently supports only a 4 bones per vertex.

Skeleton and animations. The skeleton or rig of a virtual character is used for animations. Body skeletons can be very varied in complexity. In practical terms, their most important feature is their compatibility with a (large) database of animations, as creating animations that look natural or recording them with motion capture techniques can be time consuming. From the several animations databases available, we started using the CMU Graphics Lab Motion Capture Database and retargeting the animations to *MakeHuman* characters using *Blender*. This required a bit of tweaking for each animation. At a later stage we switched to using *Mixamo*, as the animations can be previsualized and tweaked on the web application, which greatly speeds up the process.

8.2.3 Scene editors and renderers

Scene editors and renderers have become more usable and widespread over the last ten years, in connection to game development. *Unity* and *Unreal Engine*, for instance, save users a lot of time, allowing them to rapidly create, visualize and export 3D scenes and games with little coding and compatibility issues. Exported scenes can be interactively played on the web. Although plugins and/or manual configuration are needed, the maturity of the tools makes them a solid alternative to create playable web content. However, these web scenes do not support key features required for ECAs, such as easy integration with off-the-shelf web APIs, to enable their speech, for instance.

The introduction of *WebGL,* "a cross-platform, royalty-free web standard for a low-level 3D graphics API based on OpenGL ES2.0, exposed through the HTML5 Canvas element as Document Object Model interfaces" (Kronos Group), has contributed to a widespread use of higher quality 3D graphics on the web, because the API addresses directly low-level graphics devices, and web-based 3D graphics applications can become ubiquitous within the web environment (Evans et al. 2014), in particular for ECAs.

In order to provide real-time rendering, scene setup and scripting, a 3D engine and scene editor is required. There are several libraries, such as *SceneJS* and *Three.js* to create 3D scenes on the web without having to struggle with the low-level layer, but demand a good knowledge of 3D graphics programming. Unlike them, *WebGLStudio* (Agenjo et al. 2013), a web based open-source 3D scene editor and rendering engine, requires less knowledge of low-level graphics programming, while following an approach similar to *Unity* and *Unreal Engine*. The scenes created can be accessed through a web browser, without installing plugins. *WebGLStudio* has tools and components to easily integrate virtual characters. It supports animations, blend shape transformation through the GPU, custom shaders, light baking and mesh compression, to mention a few features. *Clara.io* is another browser based scene editor and engine with similar features to *WebGLStudio*, but is more oriented towards 3D modeling (this is also the case of *Sketchfab*).

8.2.4 Virtual Character Integration and Control

With the virtual character imported in *WebGLStudio*, scripts and components can be added on top of it to control its behavior in a fairly straightforward way. ECAs researchers have made efforts to standardize control commands and our work is based on them, specifically, on the SAIBA framework established several years ago (Kopp et al. 2006). This framework "distinguishes three processes in the generation of multi-modal communicative actions: intention planning, behavior planning, and behavior realization" (Heylen et al. 2008), which are interconnected and receive feedback from each other. The communication languages between these processes were also established by this group of researchers: the *Functional Markup Language* (FML) leads from the first to the second process as it "is supposed to be concerned with specifying the intentions of an agent" (Heylen et al. 2008, p. X), while the *Behavior Markup Language* (BML) goes from the second to the third as it "is one kind of specification of the behaviour that results" (Heylen et al. 2008, p. X). We focus on the visual representation, and thus BML is the most relevant for us.

In our work, we have implemented the support of basic BML commands, thus providing a web-based BML Realizer, which relied on a very basic (behavior) planner. For the all-important facial expression, we developed several web-based systems: facial animation based on valence and arousal proposed by (Romeo 2016) for desktop; simple lip-syncing (Llorach et al. 2016); gaze and head nods, part of BML, which are acknowledged as basic communication gestures; and some basic animations to communicate some specific nonverbal messages. Gaze behaviors followed the directions of Ruhland et al. (2015). For more information, Huang and Pelachaud (2012) discuss different animation pipelines based on SAIBA; McTear et al. (2016) provide a more complete review on tools to create desktop ECAs and behavior control standards in their book on conversational interfaces.

8.3 Our pipeline in more detail

In 8.3.1, we summarise our pipeline for designing, rendering, and controlling web-based ECAs. In 8.3.2 – 8.3.6, we focus on specific and key aspects of our pipeline, such as gaze and gestures. In 8.3.7 we offer an example of implementation of the pipeline.

8.3.1 Overall process

Virtual character creation

1. Among the alternatives discussed in 2.1 we regard *Adobe Fuse CC* as the best current option because of its realism, customization of clothes, small file size, number of textures for the materials, number of facial blend shapes and direct connection to the animation database *Mixamo*. Nevertheless we also discuss the issues and steps required with the other virtual character creation tools.

Blender: improving the model

2. Once the character is created with one of the virtual character creation tools (*Makehuman, Adobe Fuse CC, Autodesk Character Generator or Daz Studio)*, our next step is to import it into *Blender*. With *MakeHuman* we use the *Blender Exchange* format (*.mhx2*) together with the *MakeHuman* plugin for *Blender*; with the rest of the tools we import the virtual character as an *FBX* file.
3. In Blender we separate the head from the body (which appears as a single mesh in most cases) and remove blend shapes from body parts that are actually not deformed. A 60% size reduction results, and 20% more if merging symmetric blend shapes (e.g., for eyes) and removing unused blend shapes.
4. We create other necessary blend shapes by mixing existing blend shapes/facial bones or manually. In our facial expression system and lip-sync, only 9 blend shapes are used, four for the upper part of the face and five for the lips and mouth: eyebrows up, eyebrows down, inner eyebrows up and eyelids down; lips funneled, smile, lip corners down, lips pressed, mouth open. In terms of the Facial Action Coding System (FACS) (Ekman and Rosenberg 1997): AU2, AU4, AU1, AU43; AU22, AU12, AU15, AU24, AU27 respectively.
5. Usually small corrections are needed to fix errors of the model or to improve the character. Adding a bit of weight to the skin surrounding each eye adds a lot of realism: when the character looks away from the center, the eyelids and skin around the eyes will follow slightly. Some fixes depend on the software: in *MakeHuman* adding bone weights for the tongue and shoes and adjusting the teeth position; in *Autodesk Character Generator* unpacking the textures from the *.FBX* format to separate files; in *Adobe Fuse CC* renaming the objects and skeleton.
6. Further file size reduction for better loading process performance can be achieved by lowering the resolution of the textures, simplifying or deleting geometries that are not going to be seen or do not take part in the final scene.

Mixamo / Blender with Makehuman plugin for animations

7. We perform the preprocessing of the animations in *Blender* and/or in *Mixamo*. With *Mixamo* the character can be uploaded on the platform and the animations previsualized and adjusted in a very user-friendly way (e.g., via sliders that can even control motion style). *Blender* permits finer control (e.g., modifying weight bones), but loading, visualizing and modifying animations is slower. As an example of adjustment needed, if a character has very small hips and torso, it is likely that an animation from a standard database will place the arms too separated from the thin body. Within *Mixamo* the animations are automatically adapted to the character's size, whereas *Blender* animations would need to be adapted manually to each character. In our pipeline, animations can be imported later on *WebGLStudio*, so new animations can be continuously added to the web-based characters.

WebGLStudio for integration and control

8. We import in *WebGLStudio*, which supports two 3D standard formats, *OBJ* and *COLLADA*. *OBJ* does not support animations, thus we use *COLLADA* to import the characters with animations, skeleton, blend shapes, materials, lights and cameras.
9. To create materials, textures can be dragged directly on the 3D object in the scene, but some configuration of each material is still required, such as adding reflections to the cornea and specifying which texture applies to what (diffuse, specular, opacity…).

10. We use *WebGLStudio* components and scripts to configure the scene with the set (cameras, lights, background scenario) and add the scripts and components to control the virtual character. The scripts and components will be described in detail in the following sections.
11. As *WebGLStudio* works in the (browser) client, the character and scene needs to be either stored in a server (through a *WebGLStudio* account for example) or downloaded. *WebGLStudio* can download the whole scene in a custom format (*.WBIN*), as a folder ready to deploy in a HTML server, or to store a new character on a remote disk.

8.3.2 Behavior scripting and control

In an ECA application, the character is controlled via instructions generated by the system that replies to user input. There are several standards to control virtual characters (Gibet et al. 2016) and some have been proposed and used over the years specifically for ECAs, within the SAIBA framework mentioned above. We use BML, as it is currently the most widespread format; EMBR (Heloir et al. 2009) has been proposed as an alternative for finer animation control. As our character is web-based, it can be controlled by a third party application hosted remotely, which is indeed the case in the KRISTINA project.

Within BML standardized behaviors, we support time synchronization, face, gaze, gesture, head and speech. We found especially challenging the simultaneous combination of several behaviors as far as they affect the same parts of the body. For example, both the actions "look to the left" and "head nod" modify the rotation of the head in different ways. We combined the rotations of the two, so that the agent can look towards the left while nodding. Both facial expressions and lip syncing (speech behavior) also control the lower part of the face. In our implementation we opted to override the facial expression of the lower part of the mouth with the lip syncing while the agent is speaking, although there are alternatives to combine facial expressions and lip syncing (Karras et al. 2017).

8.3.3 Look at

Gaze is a very important feature in the communication process. It indicates where attention is directed, plays an important role in turn-taking and even thoughts and emotions can be inferred from it (Ruhland et al. 2015). In our pipeline, we implemented a script component to make the character look at a desired 3D point in the scene by rotating the bones of the eyes, head and neck with different intensities through inverse kinematics. We adjust the intensity of the component to achieve natural relationships between the eye movement and head rotations.

8.3.4 Lip-syncing

Lip-syncing is still an active area of research, where knowledge and expertise in speech processing, computer graphics and human animation is required. Although many solutions provide real-time lip-sync, commonly their parameters are generated using pre-recorded speech and then lip-sync is reproduced in real-time. Few solutions generate lip-sync based on live input in real-time (Wei and Deng 2015)(Liu et al. 2011) and they have not been implemented in a web browser.

We developed a simple lip-syncing (Llorach et al. 2016) which only uses the web resources, but can analyze the audio signal and extract the lip parameters with live input in real-time. The algorithm computes the short-term power spectrum and extracts the energies of different frequency bands, which are then mapped to three blend shapes through a set of equations based on the formant location and other speech features of the spectrum. The blend shapes used for this lip-sync are lips funneled, lips pressed, mouth open, which in terms of the FACS are AU22, AU24 and AU27 respectively.

8.3.5 Facial expressions

It is essential that facial expressions are accurate in order to properly communicate, as shown by Beale and Creed (2009). Non-real-time applications can spend as long as needed to create facial expressions. On the contrary, the ECAs facial expression needs to be almost real-time without the possibility of small fixes. It can be controlled at different abstraction levels: high level labels such as happy, angry and displeasure; low-level instructions such as those provided in MPEG-4 (Tekalp and Ostermann 2000) or the FACS, each with its own advantages and limitations. High level labels limit the encoding capabilities to a set of predefined expressions, and low-level instructions need detailed specific control, which probably require another level of abstraction for its definition.

We have adapted to the web an intermediate approach proposed by Romeo (2016), where a 2D emotional space was used to generate facial expressions combining several facial actions from the FACS. The approach uses a valence-arousal (also known as activation – evaluation) space, which has been recently become more widespread in the literature. For instance, Hyde et al. (2014, 2015) use a valence-arousal based system to understand how to animate the faces of realistic characters to improve perceptions of emotions and naturalness with the goal of creating more compelling interactive applications. In Romeo's model there are eleven predefined facial expressions in different positions of the valence and arousal wheel (Figure 8.2). The system interpolates between the nearest predefined facial expressions in the 2D space.

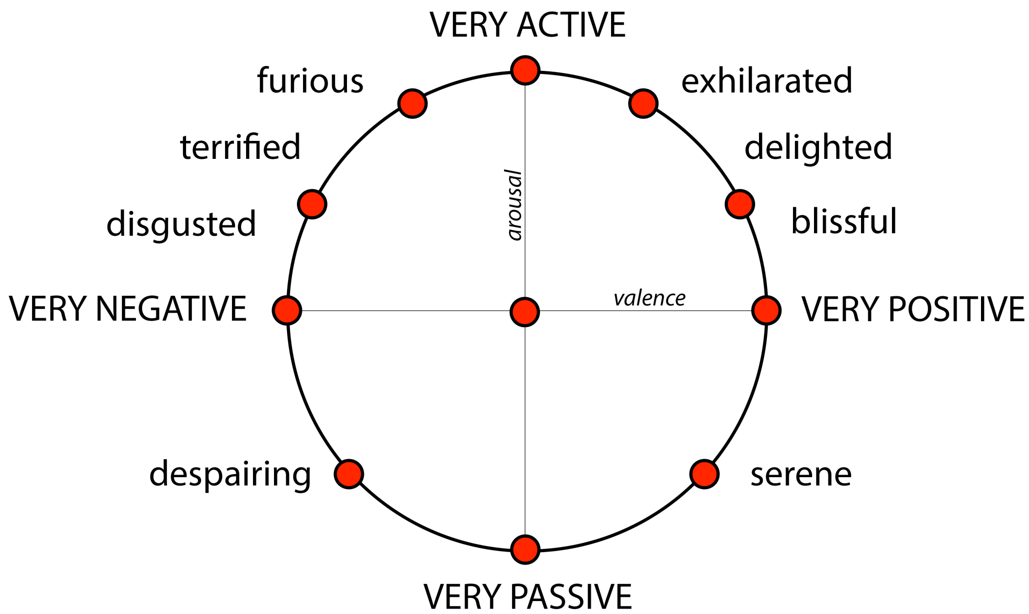

Figure 8.2: Positioning of the predefined facial expressions in the valence (horizontal axis) and arousal (vertical axis) space.

The eleven predefined facial expressions of the model can be created with only 9 blend shapes, four for the upper part of the face and five for the lips and mouth: eyebrows up, eyebrows down, inner eyebrows up and eyelids; lips funneled, smile, lip corners down, lips pressed, mouth open, which are FACS AU2, AU4, AU1, AU43; AU22, AU12, AU15, AU24, AU27 respectively (Figure 8.3).

Figure 8.3: Blend shapes used in our pipeline. From left to right and from top to bottom: eyebrows up (AU2), eyebrows down (AU4), inner eyebrows up (AU1), eyelids down (AU43); lips funneled (AU22), smile (AU12), lip corners down (AU15), lips pressed (AU24) and mouth open (AU27).

In our web-based implementation the facial expression of the virtual character can be controlled by either moving a pointer in the valence-arousal space or setting numerically the valence and arousal values. These different layers give a lot of flexibility to the model, as it can be controlled at very high levels (valence-arousal and emotional tags) or with more detail (blend shapes/facial action units).

8.3.6 Gestures

Gestures can convey a lot of non-verbal information. They can also modify the meaning of the verbal message. The BML standard divides body motions into head gestures, upper body gestures (arms/hands) and body posture. As discussed earlier, conflicts can appear when several BML instructions control the same parts of the body. How to solve these conflicts is not standardized, and different BML realizers can produce characters that behave differently for the same BML instructions. We implemented different postures, so that changes of mind or state can be shown.

8.3.7 An example of a fully web-based ECA

To demonstrate the possibilities of web browser based ECAs and test our BML Realizer, we developed a complete ECA that can interact verbally in real time (Llorach and Blat 2017) with a user. The ECA uses the aforementioned BML Realizer, interfaces with the speech recognition and synthesis of the Web Audio and Speech APIs and a basic artificial conversational entity, ELIZA. Additionally, the face of the user was tracked by means of jsfeat (integrated through Web Workers) and the gaze character followed it. The whole application works in the client but for the speech recognition and synthesis services, which use Google ones by default.

8.4 ECAs and older people

Much research on ECAs with older people focuses on health-related issues. One of the first significant works on ECAs with older adults (Bickmore et al. 2005) concentrated on a relational agent that plays the role of an exercise advisor. More recently, in (McTear et al. 2016), one of the future directions of (embodied) conversational interfaces is health and ageing. The same future direction – namely, the Healthcare Personal Agent - is suggested in a relatively recent review of open challenges in modelling, analysing and synthesising human behaviour in human-machine interactions (Vinciarelli et al. 2015).

While health is a very important topic, especially in light of a growing ageing population, and much HCI research with older people also concentrates on health (Rogers and Marsden 2013), we consider that ECAs could be of great benefit to older people in other facets of their everyday lives, ranging from learning and leisure to well-being and tourism. To this end, a fundamental question for us is the physical appearance, along with the emotional component and personality, of ECAs. For humans, it is almost inevitable to impute human qualities to animals, things, and objects which are not human, in order to understand and deal with them (Lakoff and Jonhson 2003).

Working towards this end, we conducted a workshop (Maña et al. 2018) intended to explore how a group of older people (N=14) interested in technologies think ECAs look like. We first talked to them about ECAs and showed them some examples in an MS PowerPoint presentation. We then asked them to create their own ECA for two scenarios, which we thought could be of interest to them: a travel guide, a technology expert / assistant. To create the ECAs, participants used *Adobe Fuse CC* and *Mixamo*. This activity also allowed us to test the usability of these editors for older people. The activity seemed to elicit fun for them and was positively perceived. Overall, the participants created mostly female characters, with white skin, and were clothed informally (Figure 8.4). Participants were able to create their agents mostly on their own, with our support.

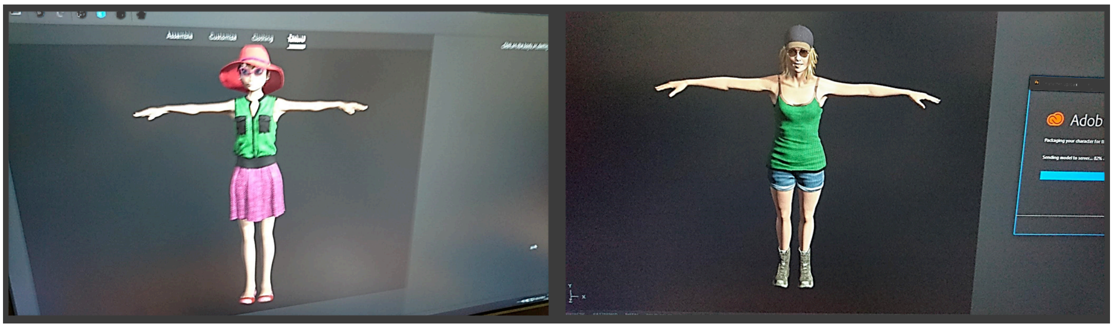

Figure 8.4: Two of the female characters created by the older adults.

While creating their preferred agents, we also had an opportunity to gather their opinions on this new type of user interface for them. There were diverse views: while some of them disliked the idea of talking with a cardboard, others pointed out that interacting with ECAs could help them skip a lot of steps while organizing a trip or buying a new computer online, for instance.

8.5 Conclusion

This chapter has presented an innovative way of building ECAs, in the context of future more widespread embodied conversational interfaces. The innovation lies in designing, controlling and deploying ECAs online. Web-based ECAs pose a few challenges, and this chapter has discussed a number of them, proposing and demonstrating solutions. This chapter has also presented results of research activities aimed to explore the design of ECAs by older people, using free online character editors. The aim of this exploration was to start to understand how an ECA should look like for this user group, and to identify the usability of online editors.

Unlike other chapters in this book, the perspective on HCI research with older people introduced in this chapter is mostly technical. The new perspective lies in a technological pipeline that allows for rapid or "quick-and-dirty" online experimentations, which are important to examine a wide range of aspects involved in the interaction between ECAs and older people. A large number of aspects (physical appearance, gestures, emotion, empathy…) remain to be understood with older people and ECAs. Exploring them online allows us to conduct a number of experiments relatively quickly. Also, older people can become "designers of assistants", using online free tools, taking one step forward the transition from passive users to active creators of digital content (and agents).

8.6 Acknowledgements

This work was funded by the EU's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 675324 (ENRICH) and under the contract number H2020-645012-RIA (KRISTINA).

8.7 References

Adobe, 2018. Adobe Fuse CC (beta). Available at: https://www.adobe.com/products/fuse.html/ Accessed 15 October 2018

Adobe, 2018. Mixamo. Available at: https://www.mixamo.com/ Accessed 15 October 2018

Agenjo J, Evans A, and Blat J (2013) WebGLStudio: a pipeline for WebGL scene creation. In Proceedings of the 18th International Conference on 3D Web Technology (pp. 79-82), ACM, NY, USA.

Autodesk, 2014. Autodesk Character Generator. Available at: https://charactergenerator.autodesk.com/ Accessed 15 October 2018

Beale R, Creed C (2009) Affective interaction: How emotional agents affect users. International journal of human-computer studies, 67(9), pp.755-776.

Bickmore, T W, Caruso L, Clough-gorr K et al (2005) 'It's just like you talk to a friend' relational agents for older adults. Interacting with Computers, 17: 711–735

Carrasco R (2017) Designing Virtual Avatars to Empower Social Participation among Older Adults. In Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (pp. 259-262), ACM, NY, USA.

Chi, N, Sparks, O, Lin S-Y et al (2017) Pilot testing a digital pet avatar for older adults. Geriatric Nursing 38:542-547

Daz Productions, 2018. Daz Studio. Available at: https://www.daz3d.com/daz_studio Accessed 15 October 2018

Druga S, Breazeal C, Williams R et al (2017) Hey Google is it OK if I eat you? Initial Explorations in Child-Agent Interaction. In: IDC '17 Proceedings of the 2017 Conference on Interaction Design and Children, ACM, NY, USA: 595- 600 doi 10.1145/3078072.3084330

Ebling M R (2016) Can Cognitive Assistants Disappear? IEEE Pervasive Computing, 15(3): 4-6, doi: 10.1109/MPRV.2016.41

Ekman P, Rosenberg EL (eds) (1997) What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS). Oxford University Press, USA.

Epic Games, 2018. Unreal Engine 4. Available at: https://www.unrealengine.com/

Evans A, Agenjo J and Blat J (2018). A pipeline for the creation of progressively rendered web 3D scenes. Multimedia Tools and Applications 77: 20355-20383

Evans A, Romeo M, Bahrehmand A, et al (2014) 3D graphics on the web: A survey. Computers & Graphics, 41: 43-61.

Feng, A, Casas D, Shapiro A (2015) Avatar reshaping and automatic rigging using a deformable model. In: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, Place, p. 57-64

Ferreira SM, Sayago S, Blat J (2017) Older people’s production and appropriation of digital videos: an ethnographic study. Behaviour and Information Technology, 36(6): 557-574 https://doi.org/10.1080/0144929X.2016.1265150

Gibet S, Carreno-Medrano P, Marteau PF (2016) Challenges for the Animation of Expressive Virtual Characters: The Standpoint of Sign Language and Theatrical Gestures. In Dance Notations and Robot Motion (pp. 169-186). Springer,

Guo P J (2017) Older Adults Learning Computer Programming: Motivations, Frustrations, and Design Opportunities. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM, NY, USA: 7070–7083. https://doi.org/10.1145/3025453.3025945

Heloir A, Kipp M (2009) EMBR: A realtime animation engine for interactive embodied agents, In: 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, Amsterdam, , pp. 1-2. doi: 10.1109/ACII.2009.5349524

Heylen D, Kopp S, Marsella SC et al (2008) The Next Step towards a Function Markup Language. In: Prendinger H, Lester J, Ishizuka M (eds) Intelligent Virtual Agents. IVA 2008. Lecture Notes in Computer Science, vol 5208. Springer, Berlin, Heidelberg

Huang J, Pelachaud C (2012) September. Expressive body animation pipeline for virtual agent. In International Conference on Intelligent Virtual Agents (pp. 355-362). Springer, Berlin, Heidelberg.

Hyde J, Carter EJ, Kiesler S et al (2014) Assessing naturalness and emotional intensity: a perceptual study of animated facial motion. In Proceedings of the ACM Symposium on Applied Perception (pp. 15-22), ACM, NY, USA.

Hyde J, Carter EJ, Kiesler S et al (2015) Using an interactive avatar's facial expressiveness to increase persuasiveness and socialness. In Proceedings of the 33rd

Annual ACM Conference on Human Factors in Computing Systems (pp. 1719-1728), ACM, NY, USA.

Karras T, Aila T, Laine S, et al (2017) Audio-driven facial animation by joint end-to-end learning of pose and emotion. ACM Transactions on Graphics, 36(4): 94.

Kopp S, Krenn B, Marsella SC et al(2006) Towards a Common Framework for Multi-modal Generation: The Behavior Markup Language. In: Gratch J, Young M, Aylett RS et al (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 205–217. Springer, Heidelberg

Lakoff G, Johnson M (2003) Metaphors we live by. The University of Chicago Press, London

Lewis, W, Lester, C (2016) Face-to-Face interaction with pedagogical agents, twenty years later. International Journal of Artificial Intelligence and Education 26: 25-36

Liu J, You M, Chen C et al (2011) Real-time speech-driven animation of expressive talking faces. International Journal of General Systems, 40(04), pp.439-455.

Llorach G, Blat J (2017) Say Hi to Eliza. In: International Conference on Intelligent Virtual Agents (pp. 255-258). Springer

Llorach G, Evans A, Blat J et al (2016) Web-based live speech-driven lip-sync. In Games and Virtual Worlds for Serious Applications (VS-Games), 2016 8th International Conference on (pp. 1-4). IEEE.

MakeHuman_Team, 2016. MakeHuman. Available at: http://www.makehuman.org/ Accessed 15 October 2018

Maña F, Toro I, Sayago S et al (2018). Older people's interactive experiences through a citizen science lens: a research report. Funded by ACT (Ageing-Communication-Technologies).

Martínez-Miranda, J (2017) Embodied Conversational Agents for the detection and prevention of suicidal behaviour: current applications and open challenges. Journal of Medical Systems, 41:135.

McTear M, Callejas Z, Griol D (2016) The Conversational Interface. Talking to Smart Devices. Springer

Pradhan A, Mehta K and Findlater L (2018) Accessibility Came by Accident: Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (pp. 459). ACM, NY, USA

Provoost, S, Ming, H, Reward, J et al (2017) Embodied Conversational Agents in Clinical Psychology: A scoping review. Journal of medical internet research, 19(5): 1-17

Rice, M, Koh, R Y I, Ng, J (2016) Investigating gesture-based avatar game representations in teenagers, younger and older adults. Entertainment Computing, 12:40–50

Ring L, Utami D and Bickmore T (2014) The right agent for the job?. In International Conference on Intelligent Virtual Agents (pp. 374-384). Springer, Cham. NOT CITED.

Rogers Y, Marsden G (2013) Does He Take Sugar? Moving Beyond the Rhetoric of Compassion. Interactions, 20(4): 48–57

Romeo M (2016) Automated processes and intelligent tools in CG media production. PhD dissertation. Available at: http://hdl.handle.net/10803/373915

Roosendaal T., 1995. Blender. Available at: https://www.blender.org/ Accessed 15 October 2018

Ruhland K, Peters CE, Andrist S, et al (2015) A review of eye gaze in virtual agents, social robotics and hci: Behaviour generation, user interaction and perception. In Computer Graphics Forum (Vol. 34, No. 6, pp. 299-326).

Shamekhi, A, Liao, Q, Wang, D et al (2018) Face Value? Exploring the effects of embodiment for a group facilitation agent. CHI 2018, Canada, Paper 391

Tekalp AM, Ostermann J (2000) Face and 2-D mesh animation in MPEG-4. Signal Processing: Image Communication, 15(4-5): 387-421.

Unity Technologies, 2018. UNITY. Available at: https://unity3d.com/.

Vinciarelli A, Esposito A, André E et al (2015) Open challenges in modelling, analysis and synthesis of human behaviour in human-human and human-machine interactions. Cognitive Computing, 7:397-413

Wanner L et al (2017) KRISTINA: A Knowledge-Based Virtual Conversation Agent. In: Demazeau Y, Davidsson P, Bajo J et al (eds) Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection. PAAMS 2017. Lecture Notes in Computer Science, vol 10349. Springer, Cham, https://doi.org/10.1007/978-3-319-59930-4_23

Wei L, Deng Z (2015) A practical model for live speech-driven lip-sync. IEEE computer graphics and applications, 35(2), pp.70-78.