Time to go ONLINE! A Modular Framework for Building Internet-based Socially Interactive Agents

Socially Interactive Agents Track

Paper #196

ABSTRACT

Although socially interactive agents have emerged as a new metaphor for human-computer interaction, they are, to date, absent from the Internet. We describe the design choices, implementation, and challenges in building EEVA, the first fully integrated platform-independent framework for deploying realistic 3D web-based social agents: with real-time multimodal perception of, and adaptation to, user's verbal and non-verbal social cues, EEVA agents are capable of communicating rich customizable content to users in real time, while building and maintaining users' profiles for long-term interactions. The modularity of the EEVA framework enables it to be used as a testbed for agents' social communication model development of increasing performance and sophistication (e.g. building rapport, expressing empathy).

We furthermore discuss a case study in which we show how we used the EEVA framework to 1) create dialog content for a health agent to deliver an online tailored behaviour change health intervention, and 2) integrate a novel model of nonverbal behavior for the agent that can be rendered online, in realtime. Our nonverbal communication model aims at capturing social cues from video recordings of dyadic clinician-patient interactions, by using a temporal generative adversarial network (GAN) conditioned on the user's facial and head signals, along with part of speech tagging of the dialogue.

KEYWORDS

Web-based Embodied Conversational Agent; Health Counselling; Realtime Interaction; Platform Independent Architecture, Framework

1 MOTIVATION

As human-computer interaction (HCI) has become increasingly present in daily life contexts involving socio-emotional content (e.g. medicine, education, entertainment), socially interactive virtual agents – also known as Embodied Conversational Agents (ECA) or as Intelligent Virtual Agents (IVA) – have emerged over the past decade as a new metaphor for HCI to address users' need for natural interfaces simulating human-human conversations.

Building an IVA, however, is no easy feat and presents many interdisciplinary challenges. Whereas having socially appropriate interactions can be challenging even for humans at times, generating artificial social behaviors requires a mix of science, psychology and art. Indeed, social appropriateness during dialogues, includes (apart from choosing an appropriate topic), knowing how to use different channels of communication to establish and maintain rapport via verbal and non-verbal cues [28, 72], such as respectful eye contact [31], motor mimicry and synchoronous postures [6], expression of facial and other nonverbal social cues that are congruent with verbal utterances and emotional states, among others. Adding to the difficulty is that many of these, conscious or unconscious social cues, are often culturally, personality, and context dependent.

In spite of such complexity, IVA researchers have leveraged latest progress in affective computing [3, 14, 75] to build agents with subtle social cues and responses [1, 10, 21, 22, 26, 29, 31, 55, 57, 57, 64, 73, 84]. IVAs are becoming able to establish some rapport [30, 34], express (some) empathy [37, 45, 56].

This progress in social realism has made it possible for IVAs to be introduced in a growing number of application domains (e.g. personal health agents or coaches, intelligent tutors, protagonist computer game characters), and research has shown the benefits that IVAs have on acceptance of, and motivation to, use technologies in these domains [8, 9, 12, 39, 44, 46].

In spite of their success, however, IVA development did not scale with the now ubiquitous connected devices and latest progress on 3D graphics that can be rendered on internet browsers. Whereas a few attempts have been made to build web-based 3D ECAs [47, 62, 69], their implementation is still very rudimentary, and none provide an integrated framework for web-based IVA development, including social cues modeling and dialog generation.

In this article, we describe the design choices, implementation, and challenges in building EEVA, the first fully integrated platform-independent framework for the rapid deployment of realistic 3D social agents on the web, which includes: 1) realistic 3D WebGL graphics for the ECA itselft (with physiologically realistic FACS-validated facial expressions, and full body animations), and its 'work' environment; 2) realtime perception of users' social cues; 3) adaptive non-verbal responses driven by a nonverbal model; 4) speech recognition and synthesis; and 5) configurable dialogue content.

Because software requirements for IVAs have been found context- and role-dependent [7, 65], to examplify how EEVA can be used for deploying IVAs on the web, we chose to build our EEVA case study in a specific context, namely healthcare and medicine, where IVAs are emerging as having a great potential impact to help the general population become or stay healthy [33, 59, 71].

In health care, where human personnel are vastly outnumbered by people who need aid, virtual health agents (also referred to as virtual health coaches) capable of screening or providing empathic support to individuals, anytime anywhere, about their lifestyles (e.g. alcohol, drug or nicotine consumption, exercise or lack of, eating habits) have not only been found promising by healthcare research, but also better accepted by users than text-only computer-based

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 2019, Montreal, Canada

https://doi.org/doi

interventions [45]. Other health-related agents have been emerging [23, 44, 63, 68], but their lack of availability on the web diminishes their potential impact.

In order to be effective, a health agent needs to be easily accessible (via common communication devices, at any time), usable (have an easy to use interface), enjoyable (provide a positive user experience), responsive to user's emotional behaviors (establish and maintain rapport) and scalable (accommodate an increasing number of users without computational overhead). In addition, because the interpersonal relationship between physician and patient involves a highly charged affective component patients' satisfaction, compliance with, and outcome of treatment are usually substantially related to their physicians' ability to establish rapport with patients [24], which, as mentioned, is (at the very least) dependent upon the ability to decode and encode nonverbal messages of affect [15, 24].

To address these requirements, the EEVA framework (described in Section 3) makes it possible to deploy multimodal realtime IVA health agents on the web, multiplying their potential to reach large population in need of access to care. Furthermore, while we built EEVA so that its content is customizable for different dialogs (described in Section 3.2), our case study (in Section 4) discusses how EEVA state of the art 3D graphics full-bodied animated characters, coupled with a nonverbal model of behavior such as the one we developed (described in Section 4.2), aims to enable the agent to establish and maintain rapport with its users, similarly to the physician-patient social relationship.

2 RELATED WORK

As mentioned, IVA research has leveraged latest progress in multimodal automatic recognition and synthesis of social and affective signals [2, 3, 14, 75] to build agents with subtle social cues and responses, such as facial expressions [1, 22, 55, 57], eye gaze [31, 64], gestures [73], pitch and intonation [21, 58], dialog management [84], generation affect-like states [10, 26, 29], among others.

Given their acquired ability to recognize and to synthesize individual social cues, IVAs models of non-verbal communication needed to decide how to control and synchronize these cues for realistic social behavior have been developed. IVAs are becoming able to establish some rapport [30, 34], express (some) empathy [37, 45, 56], portray (aspects of) a specific personality [18, 74], or manage turn-taking [38]. Recent research places focus on the fluidity of the experience with IVAs, aiming to obtain agents with the ability to process unconstrained natural language input (for a systematic review, see [41]).

Further research is needed to establish the contribution of each component in an ECA system [59], and one area that has received little attention in related literature is designing systems that aim for wide-range accessibility, usability, social responsiveness and scalability for web-based deployment of ECAs that can deliver health based interventions. Most of the literature regarding ECAs is, however, focused mainly on the agent itself, and rarely addresses the technical hurdles that need to be overcome to make them actually accessible to the end-user. Our article aims at providing implementation details to the EEVA framework designed to be easily adapted for reuse.

One of the most successful attempts aimed at unifying a multimodal behavior generation framework for ECAs consists of a three stage model (SAIBA) representing intent planning, behavior planning and behavior realization. In short, the SAIBA framework [40] (inspired by the Behavior Expression Animation Toolkit (BEAT) system [16]) is a rule-based system that consists of: firstly, interfacing between intent planning and behavior planning using a Function Markup Language (FML) that describes intent (without any concerns for physical behavior); and secondly, interfacing between behavior planning and behavior realization using a Behavior Markup Language (BML), an XML based language, which can be used to describe behavior blocks with behavior parameters [40]. SAIBA has been used successfully for animating a variety of ECAs (discussed in [73]). However, some challenges remain for acheiving social realism with SAIBA, such as knowing whether to maintain a behavior, knowing what the next behavior should be, or deciding how to synchronize behaviors with timing constraints [73].

Whereas hand-crafted gestures (such as the ones generated with SAIBA) are effective at portraying some social realism, AI-based gesture generation have gained attention because they enable to model real human behaviours from video data. Approaches using hidden markov models (HMM) have been successful at learning a speaker's head nods from gesture video corpora [42, 43], however HMMs assume that the future state is only dependent on the last state, which is not always the case in real data.

Recent machine learning based approaches are also worth exploring, as we have with Generative Adversarial Networks for developing our EEVA case study nonverbal model of behavior, based on a video corpora of real human clinician-patient interactions that we recorded.

Generative Adversarial Networks (GAN) [27] are a recent unsupervised training paradigm that enables neural networks to approximate and efficiently sample from the distribution of training data. The core of the approach consists in two networks that compete in a game against each other, one learning to generate samples while the other discriminates between generated samples and the real data.

Conditional GAN (CGAN) [52] extends the network architecture with a conditional vector that influences the generated samples.

Ideally, the two networks each improve on their task and reach Nash equilibrium, resulting in a generator network that is able to create samples that are indistinguishable from the real data disbribution. However, GANs are notorously difficult to train due the the unstable dynamics of the adversarial training process, which leads to problems such as mode collapse where the generator becomes unable to generate the entire diversity of the data distribution. Nonetheless, much progress has been done since their inception in both network architecture and training stability, resulting in models able to generate sharp high resolution images [13, 61], image-to-image [36] and text-to-image translation [86], video [50] and music generation [85] among many other possible uses.

While most GANs are designed to generate raw images, we currently aim to obtain a model able to control a virtual character through a Facial Action Coding System (FACS) control interface [25]. The FACS is the accepted standard for detecting and measuring visibly different facial movements in terms of anatomically based action units (AU). Our FACS-based approach is feasible because

we designed and validated the facial expression animations of our 3D graphics characters to faithfully reproduce each of FACS action units.

To the best of our knowledge, the only results similar to our approach were proposed by Huang and Khan (2017) [35] and Chu et al. (2018) [20]. Huang et al. were the first to use facial landmark predictions (similar to Constrained Local Models (CLM) used by OpenFace [4]) to condition the generation of static facial expressions in a dyadic interaction context. In contrast, our approach relies on FACS (which can be estimated from facial landmarks) and the aim is to generate the behaviour (sequences of head pose, AUs and eye gaze) of a FACS-enabled virtual character and takes into account information about the dialogue.

Chu et al. use reinforcement learning (RL) and GAN to train a model on a large video dataset preprocessed with OpenFace [4] to generate facial expressions for a virtual character. Our approach differs from that of Chu et al. mainly in that our aim is to capture the real time non-verbal behaviour of one person (the counsellor) based on the interlocutor's (patient) facial expressions and Part of Speech (POS) tagging of the dialogue, while Chu et al. do not condition the generator on the interlocutor and use actual text input. Moreover, the data used in our approach consists in action unit (AU) detection for both dialogue participants simultaneously and also includes eye movement information.

3 EEVA FRAMEWORK

EEVA is a modular framework (shown in Fig. 1) that we developed with the purpose of being easy to configure and to extend to a wide range of scenarios. The framework consists of three main components, each of which is detailed in the following sub-sections. In brief: first, the *application layer* consists in a modular client-side JavaScript mainframe (Fig. 1.a) which has the role of controlling the multi-modal user interface on the client side, *e.g.* WebGL rendering of the ECA (WebGL has recently become the standard technology for 3D interactive web applications), audio/video input, graphical user interface (GUI) interaction and communication with services such as speech recognition and synthesis. Secondly, the JavaScript mainframe handles execution of a scenario (such as a health intervention) which is described in the *logic layer* (Fig. 1.b) – a collection of state-machines that are created by developers using a visual programming interface. Finally, the scenario states may be configured to pull various information (*e.g.* phrases for the ECA to speak, slides to show) from the *data layer* (Fig. 1.c) – a database of content meant to be communicated to the user, created using an additional authoring tool.

3.1 Application layer

The backbone of the client-side application is a JavaScript framework that manages the creation of a collection of modules and the communication routes between them. Each module then implements various functionalities, including:

- obtaining input from the user, such as asking for and acquiring permission to access microphone and camera, processing input information (*e.g.* extracting facial expressions, analysing users' responses);
- deciding how to respond to the user (*e.g.* what verbal utterances should the agent express, and what non-verbal behaviours should it portray; and
- responding to the user through a multi-modal 3D embodied ECA, with speech synthesis, non-verbal behaviours and multimedia content (*e.g.* text, images and videos).

The result is essentially an interactive web application which can run on various platforms from desktop, to mobile phone, to autonomous robotic agent, to potential smart watch integrations (Fig. 2) via a compatible[1] internet browser. In our case study, the user can also choose between a collection of 3D virtual characters to interact with – of different gender, races, and appearances, and change the background view of the 3D office environment. Favourite characters and office views are remembered and displayed after login during the next interaction with the system.

3.1.1 Mainframe. The purpose of the framework is to automate IVA application instantiation given a configuration file containing the list of modules to be used. Once module files are loaded (through asynchronous HTTP requests), the configuration compatibility is checked, for each registered module and resource (discussed in detail in Section 3.1.2), as to whether a providing module exists for each required resource (*e.g.* Speech recognition module requiring WebRTC audio input). If a match is found, a link is made between the modules, with a reference to that particular resource. Otherwise, if the resource is declared optional in the module, the application continues; if the resource is mandatory (not optional) then the process stops with an error message. If a configuration is deemed compatible, each module is initialized and the module execution order is updated (*e.g.* audio input would be connected to speech recognition, the output of which would be sent to other modules).

The framework also provides the ability to unregister a module, which effectively removes it from the application - at runtime - which means that modules can be programmatically loaded and unloaded whenever needed. The runtime of the framework consists in a loop that calls each module in order, which also enables real time behaviour such as triggering ECA verbal and non-verbal backchannels during interaction with the user.

Unlike traditional ECAs, EEVA's design follows common modularity patterns found in robotics platforms (such as ROS [60]), enabling to create collections of modules to cover a variety of application use-cases, such as using a different browser, various internet bandwidth limitations, interaction capabilities, among others. For instance, when using speech recognition, in order to provide the transcript of the user's spoken utterances to be used by the IVA application, based on browser capabilities, a specialized module can be used to either interface with the Web Speech API [81] or to use another service such as the Watson Speech to Text [80]. Another example of module switching is the choice of higher or lower quality graphics for the ECA, based on the available internet connection or hardware capabilities of the end device (*e.g.* desktop vs. smart watch graphics processor). The advantage of this design is the seamless passing from one module implementation to another, including at runtime, without affecting the rest of the application.

[1] Compatibility consists in WebGL capability and WebRTC [82] compliance; most modern browsers now implement these standards.

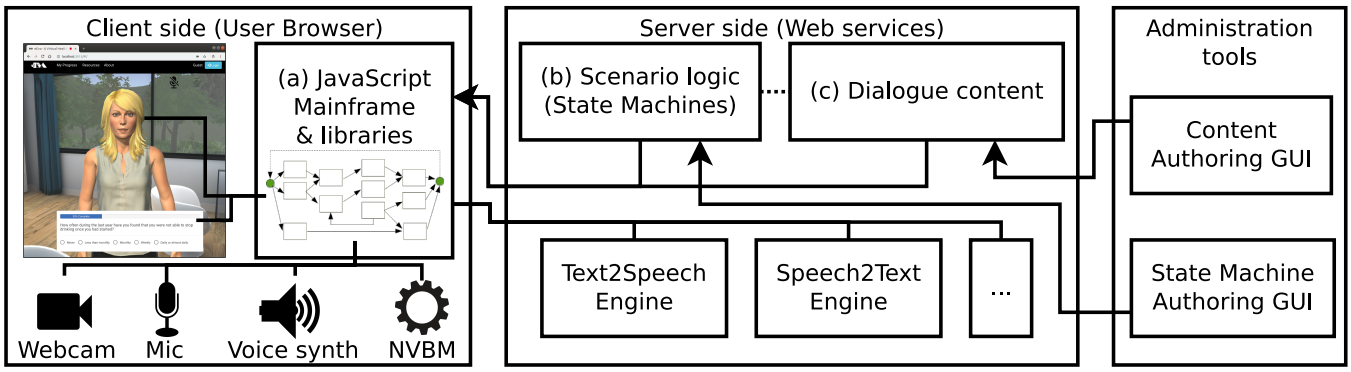

Figure 1: System diagram illustrating main components: the modular JavaScript mainframe (a) which controls the client side application by interconnecting the character input, output and non-verbal behaviour model (NVBM) with the scenario logic (b) and dialogue content (c).

Figure 2: EEVA running on different platforms: desktop (a), mobile phone (b), autonomous robot (c), smartwatch concept (d).

3.1.2 Modules and Resources. We adopted an object-oriented approach [78] to implement modules and resources, to ease development and handling by the mainframe. The main components of a module are its name (its class name), the lists of resources it requires and provides and a "run" method, which are all declared in an abstract module class from which all modules inherit from. Resources consists of their name and a container which may store any data structure (implemented as a JavaScript dictionary). Specialized modules and resources are obtained by deriving from their super-classes as in traditional object-oriented programming. The mainframe also allows a list of optional dependencies to be specified for each module, such as libraries.

Whereas the list of modules that can be used to build a socially interactive agent with EEVA will vary based on the application needs, three main categories of module functionality are necessary for modelling social interaction, namely input/sensing modules (for perceiving social cues from the users in real time); social interaction decision-making modules including the ECA behaviour module, the ECA non-verbal controller, and the scenario controller; and output/actuator modules for actually expressing verbal and non-verbal cues to the user. Other modules such as the *ECA behaviour* and the *Scenario controller* modules are more complex and process resources from multiple sources, as discussed next.

As discussed in Section 2, to generate credible social behaviour for the ECA, the *ECA behavior module* requires a social communication model based on the user's social cues (*e.g.* facial expressions, gestures, part-of-speech (POS) of utterances, intonation), which will involve different multimodal signal processing, depending upon the chosen model. Using the results of social signal processing as input, a number of existing models (discussed earlier in Section 2) can be used to decide what congruent verbal and non-verbal behaviours need to be generated to animate the 3D character. The system can then synthesize the ECA's multimodal behaviors that control the animations of the ECA non-verbal embodiment based on the social communication model.

However, although a number of systems for real-time social signals processing are available [75], most of these were not designed for internet-based interactions, so research is needed for such real time interpretation of social signals that perform over the Internet.

As discussed in our case study (see Section 4), EEVA framework can be used as a testbed for testing social interaction models of increasing sophistication, as these become available.

3.2 Logic Component and Domain Scenario Authoring Components

One of our aims in building EEVA was not only to provide a testbed for building and evaluating a diversity of web-based socially interactive agents, but also to facilitate their deployment for different applications using domain-specific various scenarios.

With EEVA, building scenarios to be interpreted by the application component consists in defining the structure of a state machine (SM) (note that SM should not to be confused with finite state automata, we use a more expressive system as discussed next). The main reasons for this approach for scenario representation are the expressive power and the relative ease of use for a non-expert whom we aid by providing a graphical authoring tool (Fig.3 discussed later). Implementation and conversion into executable code is discussed in the following.

3.2.1 Structure. While we discussed that most modules have the purpose of obtaining user data and giving feedback through various means (*e.g.* ECA animation, text, voice synthesis), the *Scenario*

controller handles the way the actual dialogue between the ECA and the user should progress.

The *Scenario controller* module is essentially an execution environment for a SM with support for external inputs (such as user' responses to prompts), and events (interruption or special commands by the user, or various errors that could occur such as internet connection loss). The controller is able to run a single SM, however, a SM can contain sub-parts (embedded state machines) depending on the application and can thus be used to describe scenarios with arbitrary complexity.

The basic building blocks of each SM consist of states and transitions. To be expressive and modular, this implementation also considers entire SMs as states, enables guards to be set on transitions, allows individual states to execute custom JavaScript code, and provides a global memory that makes it possible to count, save, and load information throughout the runtime of the SM.

The simplest state is the empty state, which conceptually defines a checkpoint in the scenario, and can serve as a starting point for decisions (such as forks). States allow instructions to be executed before and after execution (similar to entry/exit actions in UML state machines), which can be used for example to load various content from the database, or to modify global memory. Encapsulation is built in by design through the ability to call an existing SM as a single state inside a higher level SM, and allowing parameters to be passed to customize it for a particular purpose. Once execution reaches a state which has a sub-SM implementation, the higher level flow halts until the lower level SM finishes its execution (reaches its final state). This way lower level functionality can be encapsulated and reused throughout multiple scenarios.

Transitions define the connections between states and may allow free passage, or contain a guard that routes execution based on a logical condition or an external event (see example in the case study section). Conditional guards have access to the global memory and can thus be used to implement complex decisions based on the current state of the system. Event guards activate when a particular signal is sent from the mainframe itself, such as speech recognition in moments when the user's turn is to listen (user interruption), and can trigger special case behaviour of the system to resolve the interruption, such as the user requesting the agent to repeat or back up on question.

The system also allows the state of the entire hierarchy of SMs to be saved to the database, so that users can resume interacting with the ECA in a later session at the stage in the scenario where they stoped. The execution integrity is conserved by saving the current state at each level in the hierarchy, as well as the global memory, and retrieving the information whenever the application is reloaded by the user, given the user is registered within the system.

3.2.2 Authoring & code generation. While a fair level of understanding of the system is necessary to develop scenarios for interactions with the social agent, scenario developers should not require in-depth programming skills to create sequences of interactions between the ECA and its end-users. At the same time, declaring the structure of SMs in pure JavaScript quickly becomes tedious, redundant and difficult to follow even for experienced programmers.

Once created with the authoring tool (see example in case study section), the collection of SMs that make up the scenarios are saved in JSON format to the database, and can be modified or extended at a later time. In order to play the scenario within the actual application, a file is dynamically generated by the system which contains the translation of the JSON SMs into pure JavaScript code that can be executed on the client-side. Data storage is described in more detail in the following subsection.

3.3 Data layer

As mentioned, one of our design goals is to allow the content of the interaction with the social agent to be manipulated based on the needs of the domain application. Therefore the system requires a flexible data layer that can support virtually any type of interaction. Non-relational (NoSQL) databases (such as MongoDB which is used in the current implementation) offer the ability to store content with variable structure, while maintaining resource efficiency and scalability to large numbers of online users (for an overview see [17]).

Data storage in MongoDB is performed in the JavaScript Object Notation (JSON) format (in fact, a binary encoding of JSON is used internally for efficiency purposes), and allows the application to request rich content in an object-oriented way. For instance, a entire SM, user profiles, or interaction episodes can be saved as complete entities in the database, making their retrieval computationally efficient and easily interpretable for data analysis.

The data layer is designed as a list of scenarios, each containing a set of content elements. A content element can be one of the implemented generic types, including question/answer single-choice (radio button) and multiple-choice (check box), text area, tabular input, simple feedback or HTML content. It is important to note that this does not prevent the creation of new generic types, such as flexible ones for free speech dialog (see more details in future work section). Each type is interpretable in a generic way by the ECA system and can be used to author a wide range of dialogue interactions including direct questions, requests to fill forms, follow slides, or simply discussing various topics. The entire hierarchy of scenario folders and content elements is also saved as JSON documents in the database.

These design choices make it possible to link the logic and data layers together by referencing complete content documents in the SMs that are designed to process them. For example, a generic form state machine (see Fig. 3) can implement the rules to communicate an entire form to the user, and, depending on which form reference is passed as parameter, it can be reused in various moments during a scenario.

4 CASE STUDY: SOCIALLY INTERACTIVE HEALTH AGENT

In this section, we discuss how the EEVA framework can be used to develop a web-based ECA capable of delivering behaviour change health interventions. Using the mainframe discussed in Section 3, we selected a collection of modules necessary to design the health agent.

The functionalities of the main modules used in the current version of EEVA are listed in Table 1. Most modules have simple functions to retrieve or display information from and to the user or call functions from libraries (3rdparty or in-house) or services. For

example, we use the ClmTrackr JavaScript library [76] to perform facial expression recognition. Clmtrackr operates entirely on the client-side, which has the advantage of never requiring the user's webcam feed to ever exit the device, and thus maximizing privacy and trust – sine qua non basic ingredients when aiming to build rapport between the ECA and the user.

Table 1: Listing of most significant modules used in EEVA for our Health Agent.

Module function description
Input/Sensing modules
User camera interface using WebRTC API
Facial expression recognition using [67, 76]
User microphone interface using WebRTC API
Speech recognition using Google Chrome API
Speech recognition using IBM Watson API
Interface with CoreNLP [49]
GUI for direct user input (text, buttons)
GUI controller (toggle on-screen information)
Social Interaction Decision-making modules
Vocal command interpretation
ECA's non-verbal model (gesture and facial animations)
Scenario controller (state machine execution)
Output/Actuator modules
Speech synthesis using Windows SAPI
3D scene rendering for Oculus Rift
User camera display (with recognition overlay)
WebGL EEVA 3D ethnically diverse characters (Fig. 2)
Lightweight 3D scene rendering (low-end devices, Fig. 2.b,d)

4.1 Structure and Authoring

The current case study consists in delivering a brief motivational interviewing (BMI) intervention for at-risk behaviors such as alcohol consumption, over eating, smoking, lack of exercise. As detailed in [51], the content of any BMI is clearly structured into a sequence of four steps, in addition to an initial greeting and a closing statement (with potential referral of resources for healthy lifestyles) [51]: 1) screening the person's lifestyle with a series of questionnaires; 2) providing normative feedback about the person's lifestyle; 3) if the person is found to have lifestyle patterns placing them at risk (as determined in steps 1-2), assessing what level or readiness to change the at-risk behaviour(s) the person is experiencing (from not at all, to unsure, to ready); and 4) collaborating with the person to create a behaviour change plan that is aligned with the level of readiness determined in step 3.

Each step has a number of questions that prompt the user to input one answer, multiple choices, or typed or spoken natural language. The system output consists in the feedback given by the virtual character, along with visual content such as text, images and videos.

In our current health agent case study example, the scenarios determine the content and flow of the intervention (*e.g.* what questions and answers about lifestyle patterns are sought from the individual), so that the social agent can deliver the designed health intervention. Figure 3 shows a sample of the authoring tool for designing scenarios for the health agent intervention, with which content designers can author a particular interaction by specifying the steps of the procedure without necessity of in-depth programming skills.

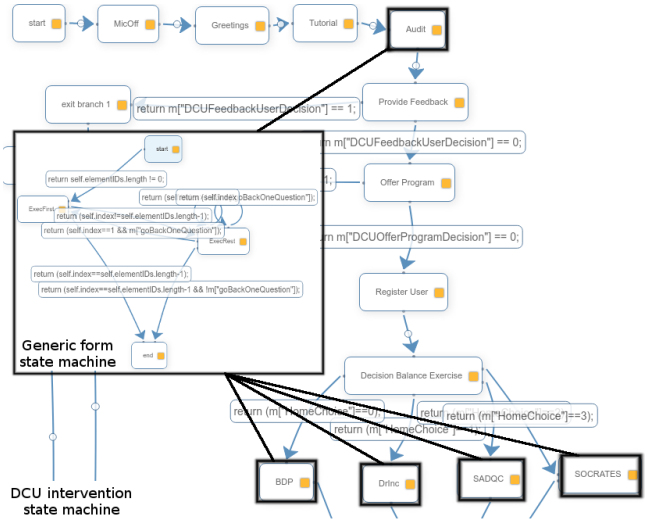

Figure 3: Designing the DCU health intevention using the state machine authoring tool. States, transitions and guards can be created with the help of a graphical user interface (GUI uses [77]). Generic SMs (black contour box) can be reused within higher level scenarios.

Figure 4 illustrates the system's data layer, a set of tools created to facilitate the creation of content for diverse use cases. The interface allows the content creator to input multimedia (visual and audio content) relevant to a particular scenario that is to be executed by the virtual character.

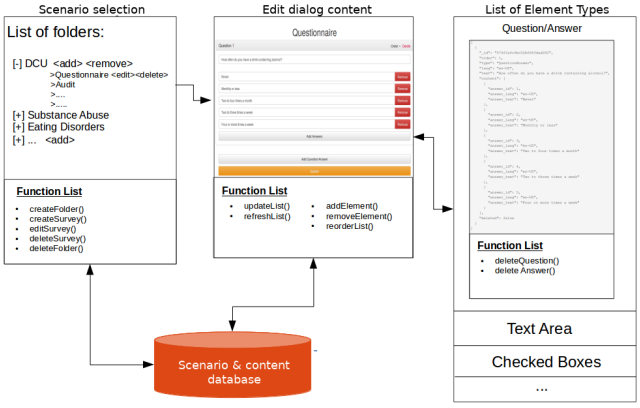

Figure 4: Structure of content authoring graphical user interface (GUI) to create dialogues and other scenario constituents using predefined generic element types.

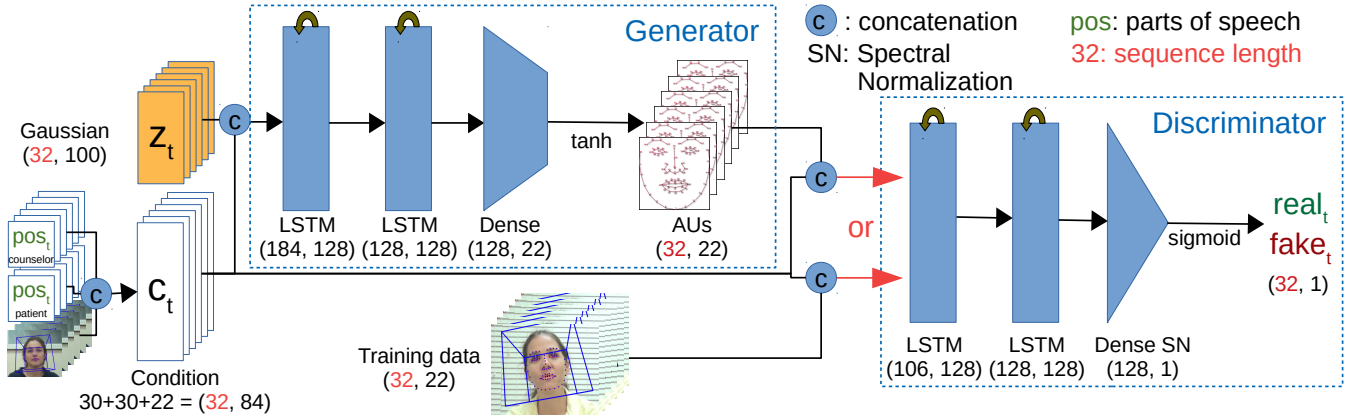

Figure 5: Behaviour model diagram. The generator (G) takes a sequence of Gaussian noise (zero mean, unit standard deviation) concatenated with the condition vector sequence as input and produces a behaviour sequence (head orientation, facial AUs and eye gaze). The discriminator (D) classifies between generated and real counsellor behaviour sequences, each concatenated with the same condition vector sequence. We used Dropout (p=0.5) [70] after each hidden layer and Spectral Normalization [53] on the last layer of D.

The EEVA framework also enables the creation and retrieval of user models that can be used to tailor and personalize the interaction with the ECA. As previously discussed, the user's answers to the various questionnaires are saved to provide personalized feedback – in our case, normative feedback about their [un/]healthy life-style. The user model consists in storing the user's answers to the agents' utterances, along with a set of scores that are calculated based on these results. These scores are used to select the branching of the scenarios. Figure 3 shows how conditional guards are taken into account during transitions between states to select the interaction path.

4.2 Non-Verbal Behaviour Model

To complete our approach, in this section we discuss the implementation of the character's non-verbal model. We based our approach on video recordings of dyadic real clinician-patient counseling sessions, in order to replicate, as closely as possible, the nonverbal behaviors of a real counselor.

4.2.1 Dataset creation. We organized and recorded sessions of behavioural change interventions between a certified clinical psychologist and student volunteers, on the topic of alcohol consumption. Synchronized video/audio recordings were captured of both interlocutors with faces clearly visible.

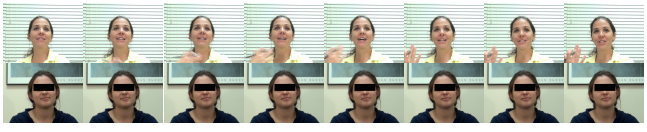

Figure 6: Time-synchronized video frame sequence from the dataset. Sampling rate: 15FPS.

Twenty-four minutes of the sesssion (Fig. 6) were semi-automatically tagged with POS information. For words spanning multiple frames, the corresponding POS was repeated. The videos of both counsellor and patient were processed with OpenFace 2.0 [4, 5, 83] to extract head orientation, gaze direction and AUs 1, 2, 4, 5, 6, 7, 9, 10, 12, 14, 15, 17, 20, 23, 25, 26 and 45, at a frequency of 15 frames per second. Counsellor and patient POS were converted to one-hot vectors, patient AUs were normalized to $[0, 1]$, and were concatenated together to form the Condition vector. Counsellor AUs were normalized in the same manner and used as real samples training the GAN. No additional preprocessing was performed. The resulting dataset was split into ~ 6 continuous minutes for validation and the rest for training.

4.2.2 Temporal Conditional GAN. We trained a temporal generative adversarial network conditioned on dialogue parts of speech and on the patient's social cues, that generates AUs, head orientation and eye gaze for the counsellor (Fig. 5). Both generator (G) and discriminator (D) consist of 2 LSTM [32] layers with hidden size of 128, and a dense layer. We used the hyperbolic tangent and sigmoid for the output of G and D respectively, and we found that adding a LeakyReLU in addition to the *tanh* output (similar to rectified-tanh [11]) of each LSTM layer was critical to obtaining realistic behaviour (leak rate 0.01 was used). Our experiments with using only *tanh* (standard LSTM), only *(Leaky)ReLU* [48, 54] and even using the same approach but with larger leak rates (0.2), all resulted in the inability to correctly capture some correlations in the data such as opening the mouth only when speaking (based on POS information), and frequent uncanny effects were observed.

Sequence length affects how much past information the model takes into account to generate the current behaviour frame. Given that AU detection was performed at 15 FPS, setting the sequence length to 15 results in a model with a memory of roughly one second. We tested with lengths of 8, 16, 32 and 64; smaller lengths resulted purely reactive (and somewhat repetitive) behaviour, while larger lengths tend to produce more dynamic and fluent patterns. However, we noted that longer sequences make the model much harder to train as becomes easier for D to detect inconsistencies,

resulting in partial (intricate gestures that take longer than they should) or total mode collapse (generator becomes stuck in an usually uncanny configuration). A memory length of ~2 seconds (sequence of 32 frames) was found to be an acceptable trade-off.

For stabilizing training we added uniform noise to the generated and real input of D with decay over time: $n_i = n_0 * 0.9999^i$, where $n_0 = 0.1$ and i is the iteration number. The model was trained for 40000 iterations with a batch size of 256; training was less stable with lower sizes. In D we used Dropout ($p = 0.5$) [70] after each hidden layer and Spectral Normalization [53] on the last layer. We also used soft noisy labels [66] for training D: label flip with 10% probability, uniform noise between $[0, 1]$ for "real" and $[0.9, 1]$ for "fake". A learning rate of 0.0001 and 0.5 momentum (similar to [61]) was used for the Adam optimizer for both G and D .

4.2.3 Qualitative Evaluation. The only post-processing consisted in ignoring negative values from the *tanh* activation for AUs, and head movement smoothing modelled as leaky integration over each coordinate: $x'_t \leftarrow 0.8x'_{t-1} + 0.2x_t$. We ran the trained model on the validation sequence of length 5000 (*sim*6 minutes) and empirically evaluated the model focusing on the following factors: acceptance (reasonable or clearly uncanny), gesture diversity (varied or mode collapse) and ability to synchronize mouth movements with speech based on POS (indicator that different modes can be captured). The model presented above satisfies, to a certain degree, all factors that we selected for.

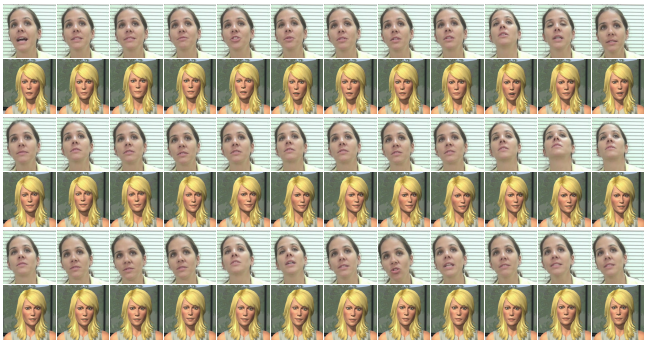

Figure 7: Synchronized sequence of real (counsellor) and generated (character) and non-verbal behaviour. Interval selected to contain change between speaker and listener roles. Sampling rate of 3FPS for visibility.

Figure 7 illustrates a sample of the generated behaviour. A video demonstration[2] of the result is also provided to allow the reader to judge the quality of the obtained behaviour.

4.2.4 Limitations. While the resulting behaviour is arguably acceptable, *i.e.* the model is able to capture the different modes for speaker and listener behaviour and produces gestures that are similar to the training data in both style and timing, we notice the following limitations of the behaviour model. First, due to the lack of information of what word is to be spoken (only POS information available), lip syncing is not realistic. However, in reality, mouth movements would be generated directly from the visemes produced by the text to speech engine. Secondly, the amount of data used to train the model is small relative to other state-of-the-art GANs, partly because of the *semi*-automatic way that was used to tag POS and that the aim was to capture the behaviour of a single person (in contrast to data from multiple people as used in other approaches). Model performance when trained on more data was not explored in this paper. Finally, quantitative evaluation of GANs is still an open problem in the field and recent improvements in this direction focus primarily on image data, such as the Inception Score [66]. We limited our work to a qualitative evaluation which focused on acceptable (in contrast to uncanny) character behaviour, which remains to be improved.

[2]Temporary anonymous link to demonstration video: https://drive.google.com/open?id=14iNkEEJCMoUZ075t2YzHziBHaj2G0Id9

5 FUTURE WORK ADDING EMPATHIC CUES

Future work will involve carrying out evaluations of the resulting nonverbal model of behavior by end-users of the health agent system, in terms of the realism of the IVA behaviors, as well as the end-users' perceived sense of rapport with the IVA delivering the health intervention.

The nonverbal behavior model was trained with Keras [19] which provides an easy way to deploy it using Tensorflow.js [79] library directly in a client-side environment, to be integrated with the EEVA mainframe. One of the advantages of this approach lies in the fact that the end-users' facial images would not need to exit the user's personal device for the system to function, thereby removing any potential privacy concerns about sharing identifiable facial images over the network.

6 CONCLUSIONS

This article described a recipe to build a complete system for developing and deploying ECAs with accessibility, responsiveness and scalability in mind. These constraints were met through a modular, platform agnostic architecture designed to be easily adaptable to various technologies, run in real time and allow end users to access its functionality with only a compatible internet browser – a common built-in feature of most modern devices. We also discussed a data-driven neural model able to capture the multimodal communicative signals from a human health counsellor, based on only video data annotated with POS, that is able to express acceptable conversational behaviors. While the main use case discussed herein stems from healthcare, since avoiding programme interruption and relapse depends on achieving the aforementioned challenges, the approach can be easily adapted for other purposes including more natural human-robot interaction, interactive storytelling, virtual training environments or e-learning.

,

REFERENCES

[1] R. Amini, C. Lisetti, and G. Ruiz. 2015. HapFACS 3.0: FACS-based facial expression generator for 3D speaking virtual characters. *IEEE Transactions on Affective Computing* 6, 4 (2015). https://doi.org/10.1109/TAFFC.2015.2432794

[2] Elisabeth André and Catherine Pelachaud. 2010. Interacting with embodied conversational agents. In *Speech technology*. Springer, 123–149.

[3] Maryam Ashoori, Chunyan Miao, Majid Nili, and Mehdi Amoui. 2008. Economically inspired self-healing model for Multi-Agent Systems. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2007*, Vol. 2. 261–264. https://doi.org/10.1109/IAT.2007.80

[4] Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson. 2015. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, Vol. 6. IEEE, 1–6.

[5] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. 2018. OpenFace 2.0: Facial Behavior Analysis Toolkit. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*. IEEE, 59–66.

[6] Janet B. Bavelas, Alex Black, Charles R. Lemery, and Jennifer Mullett. 1986. "I show how you feel": Motor mimicry as a communicative act. *Journal of Personality and Social Psychology* 50, 2 (1986), 322–329. https://doi.org/10.1037/0022-3514.50.2.322

[7] Al Baylor and Yanghee Kim. 2004. Pedagogical agent design: The impact of agent realism, gender, ethnicity, and instructional role. *Intelligent Tutoring Systems* 1997 (2004), 592–603. https://doi.org/10.1007/978-3-540-30139-4_56

[8] Amy L. Baylor. 2009. Promoting motivation with virtual agents and avatars: Role of visual presence and appearance. *Philosophical Transactions of the Royal Society B: Biological Sciences* 364, 1535 (2009), 3559–3565. https://doi.org/10.1098/rstb.2009.0148

[9] Amy L Baylor. 2011. The design of motivational agents and avatars. *Educational Technology Research and Development* 59, 2 (2011), 291–300.

[10] Christian Becker-Asano. 2008. *WASABI: Affect simulation for agents with believable interactivity*. Ph.D. Dissertation. University of Bielefeld, Faculty of Technology. http://www.becker-asano.de/Becker-Asano

[11] Xavier Bouthillier, Kishore Konda, Pascal Vincent, and Roland Memisevic. 2015. Dropout as data augmentation. *arXiv preprint arXiv:1506.08700* (2015).

[12] A. Breso, J. Martinez-Miranda, C. Botella, R. M. Banos, and J. M. Garcia-Gomez. 2016. Usability and acceptability assessment of an empathic virtual agent to prevent major depression. *Expert Systems* 33, 4 (2016), 297–312. https://doi.org/10.1111/exsy.12151

[13] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018).

[14] Rafael A Calvo and Sidney D'Mello. 2010. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing* 1, 1 (2010), 18–37. https://doi.org/10.1109/T-AFFC.2010.1

[15] K.S. Campbell. 2005. The rapport management model: how physicians build relationships with patients. *IPCC 2005. Proceedings. International Professional Communication Conference, 2005.* (2005), 422–432. https://doi.org/10.1109/IPCC.2005.1494206

[16] Justine Cassell, HH Vilhjálmsson, and Timothy W. Bickmore. 2001. BEAT: the behavior expression animation toolkit. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01)*. ACM, 477–486. https://doi.org/10.1145/383259.383315

[17] Rick Cattell. 2011. Scalable SQL and NoSQL data stores. *Acm Sigmod Record* 39, 4 (2011), 12–27.

[18] Aleksandra Cerekovic, Oya Aran, and Daniel Gatica-Perez. 2017. Rapport with Virtual Agents: What Do Human Social Cues and Personality Explain? *IEEE Transactions on Affective Computing* 8, 3 (2017), 382–395. https://doi.org/10.1109/TAFFC.2016.2545650

[19] François Chollet et al. 2018. Keras: The python deep learning library. *Astrophysics Source Code Library* (2018).

[20] Hang Chu, Daiqing Li, and Sanja Fidler. 2018. A Face-to-Face Neural Conversation Model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7113–7121.

[21] Norman D. Cook, Takashi X. Fujisawa, and Kazuaki Takami. 2006. Evaluation of the affective valence of speech using pitch substructure. *IEEE Transactions on Audio, Speech and Language Processing* 14, 1 (2006), 142–151. https://doi.org/10.1109/TSA.2005.854115

[22] M. Courgeon and C. Clavel. 2013. MARC: A framework that features emotion models for facial animation during human-computer interaction. *Journal on Multimodal User Interfaces* 7, 4 (2013), 311–319. https://doi.org/10.1007/s12193-013-0124-1

[23] Fiorella de Rosis, Nicole Novielli, Valeria Carofiglio, Addolorata Cavalluzzi, and Berardina De Carolis. 2006. User modeling and adaptation in health promotion dialogs with an animated character. *Journal of Biomedical Informatics* 39, 5 (2006), 514–531. https://doi.org/10.1016/j.jbi.2006.01.001

[24] M Robin DiMatteo. 1979. A Social-Psychological Analysis of Physician-Patient Rapport: Toward a Science of the Art of Medicine. *Journal of Social Issues* 35, 1 (1979), 12–33. https://doi.org/10.1111/j.1540-4560.1979.tb00787.x

[25] Paul Ekman and Wallace V. Freisen. 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press.

[26] Patrick Gebhard. 2005. ALMA – A Layered Model of Affect. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. 29–36.

[27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[28] JE Grahe. 1999. The importance of nonverbal cues in judging rapport. *Journal of Nonverbal behavior* 23, 4 (1999), 253–269. http://www.springerlink.com/index/V8U30855W38M4673.pdf

[29] Jonathan Gratch and Stacy Marsella. 2004. A domain-independent framework for modeling emotion. *Cognitive Systems Research* 5, 4 (dec 2004), 269–306. https://doi.org/10.1016/j.cogsys.2004.02.002

[30] Jonathan Gratch, Ning Wang, Jillian Gerten, Edward Fast, and Robin Duffy. 2007. Creating Rapport with Virtual Agents. *International Workshop on Intelligent Virtual Agents* 4722, September (2007), 125–138. https://doi.org/10.1007/978-3-540-74997-4

[31] Ouriel Grynszpan, Jean Claude Martin, and Philippe Fossati. 2017. Gaze leading is associated with liking. *Acta Psychologica* 173 (2017), 66–72. https://doi.org/10.1016/j.actpsy.2016.12.006

[32] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[33] Anita Honka, Kirsikka Kaipainen, Henri Hietala, and Niilo Saranummi. 2011. Rethinking health: ICT-enabled services to empower people to manage their health. *IEEE Reviews in Biomedical Engineering* 4 (2011), 119–139. https://doi.org/10.1109/RBME.2011.2174217

[34] Lixing Huang, Louis-philippe Morency, and Jonathan Gratch. 2011. Virtual Rapport 2.0. In *International Conference on Intelligent Virtual Agents, Intelligence, Lecture Notes in Artificial Intelligence*. Springer-Verlag Berlin Heidelberg, 68–79.

[35] Yuchi Huang and Saad M Khan. 2017. Dyadgan: Generating facial expressions in dyadic interactions. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2259–2266.

[36] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. *arXiv preprint* (2017).

[37] Joris H. Janssen. 2012. A three-component framework for empathic technologies to augment human interaction. *Journal on Multimodal User Interfaces* 6, 3-4 (2012), 143–161. https://doi.org/10.1007/s12193-012-0097-5

[38] Mathieu Jégou and Pierre Chevaillier. 2018. A computational model for the emergence of turn-taking behaviors in user-agent interactions. *Journal on Multimodal User Interfaces* 12, 3 (2018), 199–223. https://doi.org/10.1007/s12193-018-0265-3

[39] Yanghee Kim and Amy L. Baylor. 2016. Research-Based Design of Pedagogical Agent Roles: A Review, Progress, and Recommendations. *International Journal of Artificial Intelligence in Education* 26, 1 (2016), 160–169. https://doi.org/10.1007/s40593-015-0055-y

[40] Stefan Kopp, Brigitte Krenn, Stacy Marsella, A Marshall, C Pelachaud, H Pirker, K Thórisson, and H Vilhjálmsson. 2006. Towards a common framework for multimodal generation: The behavior markup language. In *Intelligent Virtual Agents (Lecture Notes in Computer Science)*, Jonathan Gratch, Michael Young, Ruth Aylett, Daniel Ballin, and Patrick Olivier (Eds.), Vol. 4133. Springer, 205–217.

[41] Liliana Laranjo, Adam G Dunn, Huong Ly Tong, Ahmet Baki Kocaballi, Jessica Chen, Rabia Bashir, Didi Surian, Blanca Gallego, Farah Magrabi, Annie Lau, et al. [n. d.]. Conversational agents in healthcare: a systematic review. *Journal of the American Medical Informatics Association* ([n. d.]).

[42] Jina Lee and Stacy C Marsella. 2009. Learning a Model of Speaker Head Nods using Gesture Corpora. In *8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra, and Castelfranchi (Eds.). International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org), Budapest, Hungary.

[43] Jina Lee and H Prendinger. 2009. Learning models of speaker head nods with affective information. *Affective Computing* (2009). http://ieeexplore.ieee.org/xpls/abs

[44] C. LeRouge, K. Dickhut, C. Lisetti, S. Sangameswaran, and T. Malasanos. 2016. Engaging adolescents in a computer-based weight management program: Avatars and virtual coaches could help. *Journal of the American Medical Informatics Association* 23, 1 (2016). https://doi.org/10.1093/jamia/ocv078

[45] Christine Lisetti, Reza Amini, Ugan Yasavur, and Naphtali Rishe. 2013. I can help you change! an empathic virtual agent delivers behavior change health interventions. *ACM Transactions on Management Information Systems (TMIS)* 4, 4 (2013), 19.

[46] C. Lisetti, R. Amini, U. Yasavur, and N. Rishe. 2013. I can help you change! An empathic virtual agent delivers behavior change health interventions. *ACM Transactions on Management Information Systems* 4, 4 (2013). https://doi.org/10.1145/2544103

[47] Gerard Llorach and Josep Blat. 2017. Say Hi to Eliza. In *International Conference on Intelligent Virtual Agents*. 255–258. https://doi.org/10.1007/978-3-319-67401-8_34

[48] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, Vol. 30. 3.

[49] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60.

[50] Michael Mathieu, Camille Couprie, and Yann LeCun. 2015. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440* (2015).

[51] William R Miller, R Gayle Sovereign, and Barbara Krege. 1988. Motivational interviewing with problem drinkers: II. The Drinker's Check-up as a preventive intervention. *Behavioural and Cognitive Psychotherapy* 16, 4 (1988), 251–268.

[52] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[53] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* (2018).

[54] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

[55] Magalie Ochs, Catherine Pelachaud, and Gary McKeown. 2017. A User-Perception Based Approach to Create Smiling Embodied Conversational Agents. *ACM Transactions on Interactive Intelligent Systems* 7, 1 (2017), 1–33. https://doi.org/10.1145/2925993

[56] Ana Paiva, Iolanda Leite, Hana Boukricha, and Ipke Wachsmuth. 2017. Empathy in Virtual Agents and Robots. *ACM Transactions on Interactive Intelligent Systems* 7, 3 (2017). https://doi.org/10.1145/2912150

[57] Catherine Pelachaud. 2009. Modelling multimodal expression of emotion in a virtual agent. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 364, 1535 (dec 2009), 3539–48. https://doi.org/10.1098/rstb.2009.0186

[58] Catherine Pelachaud, Norman I Badler, and Mark Steedman. 1996. Generating Facial Expressions for Speech. *Cognitive Science* 20, 1 (1996), 1–46. https://doi.org/10.1207/s15516709cog2001_1

[59] Simon Provoost, Ho Ming Lau, Jeroen Ruwaard, and Heleen Riper. 2017. Embodied conversational agents in clinical psychology: a scoping review. *Journal of medical Internet research* 19, 5 (2017).

[60] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, Vol. 3. Kobe, Japan, 5.

[61] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[62] Vikram Ramanarayanan, David Pautler, Patrick Lange, and David Suendermann-Oeft. 2018. *Interview with an avatar: A real-time cloud-based virtual dialog agent for educational and job training applications.* Technical Report Research Memorandum No. RM-18-02. Princeton, NJ: Educational Testing Service. 1–8 pages.

[63] Albert Rizzo, Russell Shilling, Eric Forbell, Stefan Scherer, Jonathan Gratch, and Louis-Philippe Morency. 2016. Chapter 3 – Autonomous Virtual Human Agents for Healthcare Information Support and Clinical Interviewing. *Artificial Intelligence in Behavioral and Mental Health Care* (2016), 53–79. https://doi.org/10.1016/B978-0-12-420248-1.00003-9

[64] K. Ruhland, C. E. Peters, S. Andrist, J. B. Badler, N. I. Badler, M. Gleicher, B. Mutlu, and R. McDonnell. 2015. A Review of Eye Gaze in Virtual Agents, Social Robotics and HCI: Behaviour Generation, User Interaction and Perception. *Computer Graphics Forum* 34, 6 (2015), 299–326. https://doi.org/10.1111/cgf.12603

[65] Jennifer Sabourin, Bradford Mott, and James Lester. 2011. Computational Models of Affect and Empathy for Pedagogical Virtual Agents. *Standards in Emotion Modeling* (2011). http://www.lorentzcenter.nl/lc/web/2011/464/presentations/Sabourin.pdf

[66] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*. 2234–2242.

[67] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. 2011. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision* 91, 2 (2011), 200–215.

[68] Mark R Scholten, Saskia M Kelders, and Julia EWC Van Gemert-Pijnen. 2017. Self-guided web-based interventions: Scoping review on user needs and the potential of embodied conversational agents to address them. *Journal of medical Internet research* 19, 11 (2017).

[69] Jessica Schroeder, Chelsey Wilks, Kael Rowan, Arturo Toledo, Ann Paradiso, Mary Czerwinski, Gloria Mark, and Marsha M. Linehan. 2018. Pocket Skills: A Conversational Mobile Web App To Support Dialectical Behavioral Therapy. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2018)* (2018), 1–15. https://doi.org/10.1145/3173574.3173972

[70] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[71] J. Graham Thomas and Dale S. Bond. 2014. Review of innovations in digital health technology to promote weight control. *Current Diabetes Reports* 14, 5 (2014). https://doi.org/10.1007/s11892-014-0485-1

[72] L. Tickle-Degnen and Robert Rosenthal. 1990. The nature of rapport and its nonverbal correlates. *Psychological Inquiry* 1, 4 (1990), 285–293. http://www.tandfonline.com/doi/abs/10.1207/s15327965pli0104

[73] Hannes Vilhjalmsson, Nathan Cantelmo, Justine Cassell, Nicolas E Chafai, Michael Kipp, Stefan Kopp, Maurizio Mancini, Stacy Marsella, Andrew N. Marshall, Catherine Pelachaud, Zsofi Ruttkay, Kristinn R Thórisson, Herwin Van Welbergen, and Rick J. Van Der Werf. 2007. The Behavior Markup Language: Recent Developments and Challenges. In *International Conference in Intelligent Virtual Agents. Lecture Notes in Artificial Intelligence*, Pelachaud C.; Martin JC.; André E.; Chollet G.; Karpouzis K.; Pelé D. (Ed.). Vol. 4722. Springer, Berlin, Heidelberg, 99–111. https://doi.org/10.1007/978-3-642-40415-3

[74] Alessandro Vinciarelli and Gelareh Mohammadi. 2014. A survey of personality computing. *IEEE Transactions on Affective Computing* 5, 3 (2014), 273–291. https://doi.org/10.1109/TAFFC.2014.2330816 arXiv:arXiv:1011.1669v3

[75] Johannes Wagner, Florian Lingenfelser, Tobias Baur, Ionut Damian, Felix Kistler, and Elisabeth André. 2013. The Social Signal Interpretation (SSI) Framework Multimodal Signal Processing and Recognition in Real-Time. In *Proceedings of the ACM Multimedia Conference*. 1–4.

[76] Website. [n. d.]. Clmtrackr javascript library for fitting facial models. https://github.com/auduno/clmtrackr. ([n. d.]). Accessed Nov 2018.

[77] Website. [n. d.]. jsPlumb Community Edition. https://github.com/jsplumb/jsplumb. ([n. d.]). Accessed Nov 2018.

[78] Website. [n. d.]. Prototype.js. http://prototypejs.org/. ([n. d.]). Accessed Nov 2018.

[79] Website. [n. d.]. TensorFlow.js. https://js.tensorflow.org/. ([n. d.]). Accessed Nov 2018.

[80] Website. [n. d.]. Watson Speech to Text. https://www.ibm.com/watson/services/speech-to-text/. ([n. d.]). Accessed Nov 2018.

[81] Website. [n. d.]. Web Speech API. https://w3c.github.io/speech-api/. ([n. d.]). Accessed Nov 2018.

[82] Website. [n. d.]. WebRTC project. https://webrtc.org. ([n. d.]). Accessed Nov 2018.

[83] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. 2015. Rendering of eyes for eye-shape registration and gaze estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. 3756–3764.

[84] U. Yasavur, C. Lisetti, and N. Rishe. 2014. Let's talk! speaking virtual counselor offers you a brief intervention. *Journal on Multimodal User Interfaces* 8, 4 (2014). https://doi.org/10.1007/s12193-014-0169-9

[85] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient.. In *AAAI*. 2852–2858.

[86] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint* (2017).