DEPARTMENT: EDUCATION

WAVE: A Web-Based Platform for Delivering Knowledge-Driven Virtual Experiences

Sachin Shah, Kamran Ali, Lisa Dieker, and Charles Hughes, *University of Central Florida, Orlando, FL, 32816, USA*

Many studies have demonstrated the usefulness of virtual characters in educational settings; however, widespread adoption of such tools is limited by development costs and accessibility. This article describes a novel platform, web automated virtual environment (WAVE), to deliver virtual experiences through the web. The system integrates data acquired from a variety of sources in a manner that allows the virtual characters to exhibit behaviors that are appropriate to the designer's goals, such as providing support for users based on understanding their activities and their emotional states. Our WAVE platform overcomes the challenge of the scalability of the human-in-the-loop model by employing a web-based system and triggering automated character behaviors. To support wide-scale use, we are making WAVE freely accessible (part of the Open Education Resources) and available anytime, anywhere.

For the last decade and a half, our team has been developing learning experiences enabled by context-appropriate behaviors of virtual characters. Initially, our virtual learning environments (VLEs) were focused on teacher preparation, with the virtual characters situated in virtual classroom settings. This use gives rise to the system's name, TeachLivE, standing for teaching and learning in a virtual environment. Here the virtual characters are controlled by a human-in-the loop, called an interactor, who uses gesture-based puppetry and whose performance always stays true to the characters being represented.[1] Our system allows a skilled interactor to rapidly move between inhabiting individual avatars, each of whom is based on one of Long's 2011[2] psychological profiles. The key to a genuine experience is the interactor adopting the backstory, personality, voice (aided by voice morphing), self-regulation, and intellectual strengths and challenges of the character being inhabited.

Figure 1 shows a typical classroom scene with six high school characters. In this context, the interactor inhabiting student Ed is always the caring big brother, the polite young man who wants to be an Air Force pilot, and the student who has some mathematics misconceptions that need to be addressed. While inhabiting CJ, the strongest personality in the class, the interactor displays behaviors that show her to be the least interested in learning. In sharp contrast, Maria is a passive independent with a high IQ, who connects only when directly addressed by a teacher who recognizes Maria's breadth and depth of interests. These characteristics are always the same no matter which human interactor is in control. The standardization and fidelity of the interactions allow for both repeated experiences and standardized practice, much like the standardized patient model in medicine.

With our early focus on classroom teachers, we expanded our VLE to create experiences for teachers working with diverse populations [e.g., second language learners, students who are cognitively challenged, and others exhibiting behaviors associated with autism spectrum disorders (ASD)]. We then developed a wide variety of experiences, including those for interviewers and interviewees, medical residents dealing with children who are exhibiting nonsuicidal self-injury (NSSI) characteristics, and police officers faced with people exhibiting unusual verbal and nonverbal behaviors (e.g., those associated

0272-1716 © 2023 IEEE
Digital Object Identifier 10.1109/MCG.2023.3260599
Date of current version 18 May 2023.

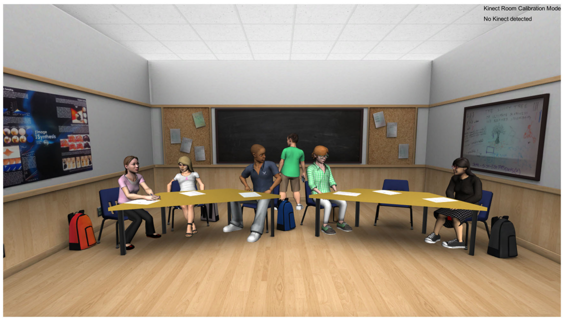

FIGURE 1. TeachLivE High School Scene.

with people on the autism spectrum). For additional information and demonstrations of the TeachLivE system, please visit http://teachlive.org.

Our latest applications of the TeachLivE paradigm have been working directly with young people with ASD.[3] Our learning goals in these projects are to help young children acquire communication and STEM (science, technology, engineering, and math) skills, and to help adolescents deal with challenges in online environments (bullying and access to private data).

While this human-in-the-loop model has been very effective, it does present challenges for scalability. The two most important characteristics of a scalable virtual learning environment are that it be web-based and that the character behaviors be automated. The web-based characteristic is necessary so teachers, trainers, parents, students, and other interested parties can access its capabilities without the need to download and install new applications (that capability in schools is often limited and only available to IT personnel) and ensures a standardized experience across platforms. The automated characteristic removes the need to continue to depend on a human actor to control the avatar behaviors and allows for free access (part of the Open Education Resources) anytime, anywhere.

In preparation for this transition to an automated, web-based platform, we chose to focus on our recent interactions with children and adolescents with ASD as the most demanding and needy applications. To achieve this goal, we spent the last two years using a human-in-the-loop to understand the way a skilled person, with the aid of subject-matter experts, controls virtual companions to interact effectively with participants on the spectrum. In effect, we used the human interactor to train an automated system. That training involved capturing common verbal and nonverbal behaviors, and the situations that trigger these.

Over time, we have shifted human-controlled behaviors to automated ones. As a simple example, when the young children are learning the STEM skills of programming a robot in the visual language Blockly, we have used a single virtual companion, ZB (Figure 2), to help the child stay on task. Just as the human puppeteer can easily see if the child is at the computer, the automated system can use a simple vision-based algorithm to determine when the child loses focus on the programming task. ZB can then encourage the child to return to the computer or tablet, so they achieve a desired goal, (e.g., having the robot move in a manner that outlines a square or some other prescribed figure).

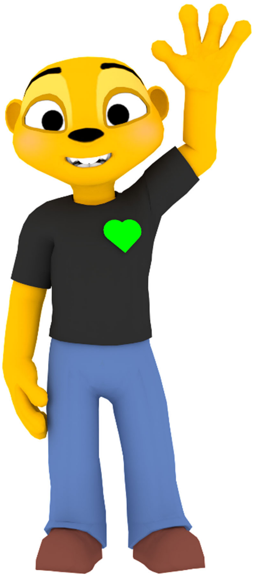

FIGURE 2. ZB virtual companion.

Moreover, just as the human can see if the child is moving toward success in the given task, we developed a very accessible Blockly-based system that the child uses for programming that conveys to the virtual companion the precise nature of the child's activities, even determining if their program will create the desired geometric shape. As there are many ways to accomplish the goal, we recognize the success of any strategy, giving support and encouragement for progress made. Our virtual companion, as it "knows" the steps being carried out when the program runs, can verbally articulate all actions if that level of verbal behavior is helpful.

To make sure the support is appropriate, we have built the Blockly-based visual programming system in a manner that allows an adult guide to configure the system with limited "widgets" (e.g., just forward movements and right turns), or with a full palette of options, including playful ones (such as changing light colors). We can

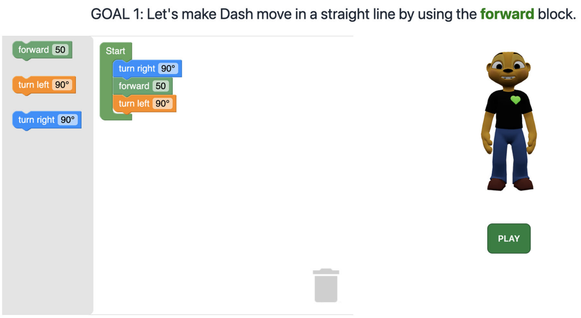
GOAL 1: Let's make Dash move in a straight line by using the **forward** block.

FIGURE 3. Blockly based visual robot programming environment.

also allow the coach (parent, teacher, or other qualified individual) to determine what level of verbal support is appropriate for each individual child. For example, we might always give positive feedback like "That's a great right angle; you're so close to a square!!" for some children but avoid this for those who are too easily frustrated by such interaction (Figure 3).

As additional support for automation, we are using sensors to acquire data about a child's focus (eye gaze), and emotions (facial expression, vocalization, verbalization, heart rate, and respiration). While these data can be easily misinterpreted, even by humans and especially in individuals with ASD, we are using the fusion of all such sensory data to recognize times when a bit of encouragement may be warranted. Even if we provide interactions that are not necessary, that is not a problem so long as all verbal and nonverbal behaviors of the virtual companion are supportive and not of a critical nature.

What follows is a description of how we integrate all these various inputs in a web-based environment that is open to new data sources, new ways to integrate these sources, new virtual character behaviors, and new algorithms to trigger those behaviors that are most effective. Such a system needs to operate without bias to today's available data sources and without preconceptions as to what future devices might emerge.

WAVE PLATFORM

The WAVE platform utilizes web-based technologies to enable broad use by running on the majority of device browsers. The platform employs a service-oriented architecture, with the crux being the synchronization logic (Figure 4). Every service connects to a WebSocket and joins a session room—a digital space that isolates session messages from other sessions. Over the established channel, services can emit "events" to notify other services. For instance, a puppeteering service would

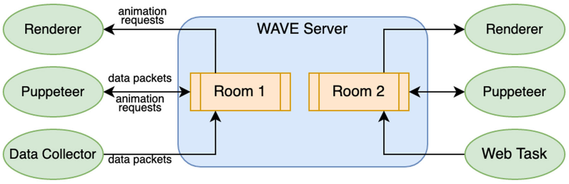

FIGURE 4. WAVE platform service-oriented system architecture. Room separation allows service customization.

emit animation requests for a rendering service to execute. Although similar to remote procedure calls, this system is distinct because each service is isolated since not all emitted events need to be listened to, and all listened-for events do not need to be emitted. This approach allows sessions to customize the experience by changing which services are used. Additionally, multiple same-type services can be used simultaneously in remote locations enabling multiparticipant applications.

The core service is the rendering service responsible for visualizing a 3-D virtual scene modeled in an application such as Blender or Maya. A scene may include audio assets, multiple characters, and skeleton and morph target animations. The WebGL library, three.js, is used to animate the scene in real time. By default, this service renders an idle animation and waits for other services to request actions or sounds to be played. The participant's experience is primarily influenced by this service.

There are three other foundational services important to basic operation. First, a puppeteering service allows a human to control the virtual scene and interact with the participant through virtual characters. This service can be used in highly individualized environments where automated behaviors are not feasible or used as a data collection method to understand how participants interact with the virtual characters so a developer can fine-tune machine learning algorithms. Second, the automated puppeteering service allows researchers to deploy an "engine" to make animation decisions (see the "Automation Engines" section). While engines make different and customizable decisions depending on the setting, they always listen for all data (see the "Knowledge Sources" section) and use some algorithm to decide what the virtual character should do. One important note is this service is not exclusive to the human puppeteer. One may use the automated puppeteer to assist the human by automating or suggesting common actions. Third, the data service type provides information about the participant and environment for both the human and automated puppeteers. The service can exist in multiple forms such as running machine learning models or communicating with external devices such as smartwatches.

KNOWLEDGE SOURCES

Computer vision techniques are an easy method of obtaining significant amounts of data such as the participant's focus because most web-enabled devices include a webcam (laptops, tablets, and phones). A challenge is dealing with participant movement, occlusions, poor lighting, and other environmental factors. Networks not tested on in-the-wild datasets may be unreliable. Additionally, performance issues may arise when trying to make decisions in real time, so it is important to minimize inference time costs. Libraries such as tensorflow.js enable GPU acceleration on the web to improve model execution speed. The service architecture enables running machine learning models on the best available hardware without blocking the virtual environment.

Like computer vision techniques, audio signals from on-device microphones provide a useful data source by enabling analysis of participant vocalization and verbalization. The tone of how the participant interacts with the virtual character may influence how the response should be worded. An increasingly frustrated participant may require a different approach or a slower pace than an excited participant. In addition, natural language processing (NLP) can be employed to determine how well the participant is picking up on the information. Repeating terminology and engaging further in the content can be a key sign of progress.

To effectively recognize emotions in a challenging real-time setup, where the video frames can be occluded, and the audio signal deteriorates due to noise and irrelevant data, we have developed a multimodal emotion recognition algorithm by employing two modalities—visual and audio. Given the impressive performance of attention mechanisms and transformers in the NLP domain, there has been some recent research exploring its effectiveness in multimodel representation learning.[4,5,6] Like Tang et al.[6] we have used a textless vision-language transformer to reduce computational redundancy and inference time suitable for our web-based system. Our transformer-based multimodal emotion recognition network consists of three blocks, an aural block, a visual block, and a visual-aural attention block, as shown in Figure 5.

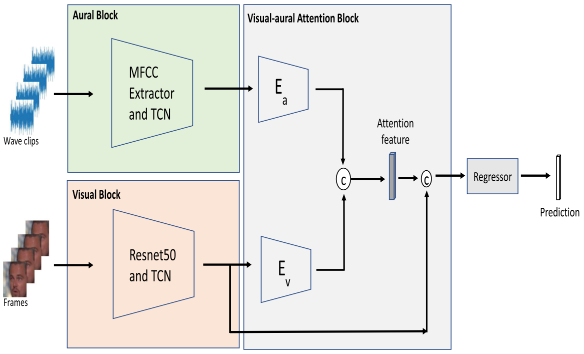

FIGURE 5. Architecture of our multimodel emotion recognition network.

Biometrics are particularly interesting because human puppeteers as well as virtual agents may benefit. Determining a participant's heart rate or respiration is difficult by visual inspection, but external devices such as smartwatches can do so with ease, either through direct measurements or the use of surrogate information. If a participant has such a device, the puppeteer gains otherwise inaccessible data. Some devices such as the polar verity sense provide a Bluetooth protocol that enables direct access through a web-based service using the Web Bluetooth API. However, to improve external device interoperability, we provide an endpoint for on-device applications to which we can post biometric information.

Web tasks can provide metrics about how the participant is doing. By building the task on the web, one can hook up events for important task milestones. This provides knowledge about whether the participant needs more support or is engaged sufficiently in the task. For example, when teaching a participant to code, a researcher might opt to use the Blockly programming environment. Then, as the participant combines blocks in their workspace, the virtual character can provide appropriate support. Another added benefit is the use of the virtual character as a tutor. If the participant is unfamiliar with the task tool, the virtual character can walk them through the basics and even take control of the participant's view to demonstrate functionality.

AUTOMATION ENGINES

There are two modes for automation engines: suggestion and action. When suggesting, the engine examines incoming data streams and suggests to a human-in-the-loop what should occur. This mode is also useful for training a fully automated engine, as the human accepting or rejecting suggestions can be treated as labels in a feedback loop. When taking action, the engine directly signals what actions should occur without human approval. This can be used in tandem with a human puppeteer by automating certain actions to lower the cognitive load on the human or completely in standalone.

The implementation details of an engine are project specific; however, we outline a few key requirements and examples. Because the platform is designed to allow service interchangeability, an engine must either signal missing services or handle incomplete data streams, with the latter being preferred. Engines can be handcrafted decision trees or learned neural networks from previous sessions. Additionally, there may be multiple data providers that are used to trigger the same action: for instance, since the task and eye detection tools both determine the user's level of attention, reconciling conflicts is important. Human actions should always be preferred over computer-generated ones. Between data providers, a majority vote system with lower error data sources having more votes can improve confidence. Because there are only a finite number preset animations, automation engines should be carefully designed around the set of actions available. One trick employed in WAVE is the use of Text-to-Speech functionality to read arbitrary support phrases. Using this automated voice in tandem with the virtual companion can strengthen interaction by referring to specific details.

APPLICATIONS

One project using the WAVE system is RAISE,[3] which introduces the use of a virtual companion named ZB. In this project, ZB assists students with ASD to learn to code a robot with a Blockly-based visual programming tool. The RAISE-UP tool was developed to customize the block types enabled for each phase. Depending on the progress of the student, different blocks can become available.

This project utilizes the core service to render the ZB character to the participant. By observing interactions with a human-in-the-loop, the project learned common behaviors and critical data points that can inform decisions. A common phrase ZB would use with easily distracted children is "come back to the screen" or with stuck students is "try the blue blocks." These action types are easily automated as task-centric data and provide information about the student's current engagement. If a student is not selecting the correct block types for the given task, a suggestion might be appropriate. Other techniques such as eye-tracking may improve the ability to detect off-task actions.

Additionally, this project uses face videos and heart rate information to provide knowledge about the emotional response of the participant. Many challenges exist in gathering this data, particularly given the unique ways through which children with ASD may express themselves. Because of misclassification dangers, reenactment has been found to be more effective.[7] For human-in-the-loop systems, heart rate data can provide the puppeteer with previously unknown information. Students may appear calm, but internally feel frustrated. Physiological sources can provide the puppeteer with key knowledge to change approaches. For more information about this project and how WAVE is used, please visit https://www.ucpcfl.org/project-raise.

Another project that is supportive of and supported by RAISE is one associated with online bullying. The first phase of this project involves participants watching virtual situations transpire and recognizing instances of bullying and attempts to access private information. Multicharacter automation can be challenging in general; however, because the situation is predetermined, the first phase simply plays through preset animations. During the second phase, when the virtual characters involve the participant in the scenario, the automated behaviors are triggered primarily based on vocalization. The participant may need additional prompting or clarification of the situation or may want to skip over a part of the experience entirely. Timing when to move on or press further is important in order to have a respectful and helpful session with the participant.

PRELIMINARY RESULTS

Educators (three classroom teachers, a self-identified paraprofessional serving as a classroom teacher, and a paraprofessional) participated in an interview and survey conducted by an external evaluator.[8] Sample results include three educators agreeing or strongly agreeing that the project and its tools were 1) conceptually sound and 2) based on evidence; two agreed that the project capabilities were clearly communicated. Three educators agreed or strongly agreed the project addresses a critical need. Four educators agreed the project 1) applies to diverse groups within the target population and 2) is relevant. Only the paraprofessional in a tech support role was asked to rate the usefulness of the toolkit, strongly agreeing the toolkit was useful and user-friendly. Interviews with five parents of the students revealed all had noticed a positive difference in their child's attitude or behavior at home and provided examples. Three of the five parents had observed a positive difference in how their child approaches schoolwork. All five expressed appreciation for having their students involved.

Two students rated their experiences in 55 sessions on a Likert-scaled instrument. Students reported positive experiences in over 90% of the sessions. Other early findings supported the effectiveness in engaging students. Time on task was consistently above 90% across participants while programming a robot with the assistance of a supportive virtual companion.

Recordings of classroom sessions showed that students bonded with the virtual companion and its presence improved their willingness to interact in class. One example of this transformation and change came from a participant who originally did not want to talk with the socially assisted avatar, ZB. However, after being successful in coding a robot, Dash, in a square with support from avatar ZB, she entered the next phase of the study with a peer. The goal was for her to introduce ZB and teach the peer how to code the robot. Surprisingly, she began to sing "I love ZB," "ZB you are my best friend." The interesting part is this same social connection with the AI agent continued when ZB entered the classroom to provide simple reaffirming executive functioning types of statements during a mathematics class. This same student made certain to "impress" her new friend ZB, sang again "I love ZB" and "ZB you are my best friend" while increasing time on task in the classroom. The level of stimming behaviors and off-topic comments also decreased with the presence of her AI friend.

Over and over the team has seen evidence of students identified as those on the spectrum connecting with ZB and even trying to show ZB their best behavior. In one instance, a student did not want anything to do with ZB, as this student struggled with any change of routine while another chose to bully the AI agent. These two students were the exception instead of the norm of what the team observed and coded but does show that AI agents are not the answers for all students. These outliers have helped the team consider the best students to target AI support in both coding and increasing time-on-task and communication skills in classroom settings.

CONCLUSION

In this article, we present the WAVE platform for real-time web-based delivery of virtual characters. By providing both in-the-web data providers and hooks for external devices, the platform enables multimodal data collection and automation models to become widespread. We see three primary areas for future work. First, further analysis of real-time performance implications of delivering machine learning-based data providers over the web should be performed to understand the viability of such techniques on low-grade consumer devices. Second, a user study should be conducted to evaluate the ease of use for participants. Third, solutions for nontechnical individuals to construct new scenes should be developed to support future research endeavors.

ACKNOWLEDGMENTS

The authors would like to recognize the contributions of Eric Imperiale (artist), Kathryn Ingraham (interactor), and John Murphy (visual programming developer) to the evolution of the experiences discussed here. The authors would also like to thank the entire Project RAISE team with a special thanks to the project PI, Ilene Wilkins, and the UCP Charter Schools team for their assistance and hospitality. This research was supported in part through cooperative agreements with the U.S. Department of Education, Office of Special Education Programs (OSEP), under Grants H327S210005 and H327S200009H, and in part by the National Science Foundation under Grant 2114808. The products under development do not necessarily represent the policy of the U.S. Department of Education or the National Science Foundation, and you should not assume endorsement by the Federal Government.

REFERENCES

1. L. A. Dieker, J. Rodriguez, B. Lingnugaris-Kraft, M. Hynes, and C. E. Hughes, "The future of simulated environments in teacher education: Current potential and future possibilities," *Teacher Educ. Special Educ.*, vol. 37, no. 1, pp. 21–33, 2014.
2. W. A. Long Jr., *Your Predictable Adolescent*. Charleston, SC, USA: BookSurge, 2011.
3. C. E. Hughes et al., "RAISE: Robotics & AI to improve STEM and social skills for elementary school students," in *Proc. Front. Virtual Reality: Virtual Reality Hum. Behav.*, 2022, Art. no. 155.
4. H. Akbariet al., "VATT: Transformers for multimodal self-supervised learning from raw video, audio and text," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, 34, pp. 24206–24221.
5. S. Zhang, Y. Ding, Z. Wei, and C. Guan, "Continuous emotion recognition with audio-visual leader-follower attentive fusion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3567–3574.

6. Z. Tang, J. Cho, Y. Nie, and M. Bansal, "TVLT: Textless vision-language transformer," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022.
7. M. Del Coco et al., "A computer vision based approach for understanding emotional involvements in children with autism spectrum disorders," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 1401–1407.
8. B. Swan, T.-A. Trail, N. Narkiewicz, and L. Carone, "PROJECT RAISE: Robots and artificial intelligence to improve social skills for elementary students: Year 2 stakeholder interview and questionnaire results," Program Eval. Educ. Res. Group, Univ. Central Florida, Orlando, FL, USA, Tech. Rep. 68UCPEDOSEP2020.Y2F, 2022.

SACHIN SHAH is a graduate student in computer science at the University of Maryland, College Park, MD, 20742, USA. He is also a researcher with the University of Central Florida Synthetic Reality Lab. His research interests include computational imaging, machine learning, and computer graphics. He is the corresponding author of this article. Contact him at sachin.shah@knights.ucf.edu.

KAMRAN ALI is a postdoctoral research associate in the computer science department at the University of Central Florida, Orlando, FL, 32816, USA. His research interests include computer vision and machine learning. Contact him at kamran.ali@ucf.edu.

LISA DIEKER is a pegasus professor and lockheed martin eminent scholars in the College of Community Innovation and Education at University of Central Florida (UCF) Orlando, FL, 32816, USA. Her research focuses on harnessing the power of teachers working across disciplines in inclusive settings in teacher education, special education, and simulation. Contact her at lisa.dieker@ucf.edu.

CHARLES HUGHES is a pegasus professor of Computer Science at the University of Central Florida, Orlando, FL, 32816, USA. His research interests include virtual learning environments, computer graphics, machine learning, and visual programming systems. Contact him at charles.hughes@ucf.edu.

Contact department editor Beatriz Sousa Santos at bss@ua.pt, department editor Alejandra J. Magana at admagana@purdue.edu, or department editor Jiri Zara at zara@fel.cvut.cz.