

# Intelligent Augmented Reality Training for Assembly Tasks

Giles Westerfield<sup>1</sup>, Antonija Mitrovic<sup>2</sup> and Mark Billingham<sup>1</sup>

<sup>1</sup>Human Interface Technology Laboratory NZ

<sup>2</sup>Intelligent Computer Tutoring Group

University of Canterbury, Christchurch, New Zealand

[giles.westerfield@gmail.com](mailto:giles.westerfield@gmail.com)

[tanja.mitrovic@canterbury.ac.nz](mailto:tanja.mitrovic@canterbury.ac.nz)

[mark.billinghurst@hitlabnz.org](mailto:mark.billinghurst@hitlabnz.org)

**Abstract:** We investigate the combination of Augmented Reality (AR) with Intelligent Tutoring Systems (ITS) to assist with training for manual assembly tasks. Our approach combines AR graphics with adaptive guidance from the ITS to provide a more effective learning experience. We have developed a modular software framework for intelligent AR training systems, and a prototype based on this framework that teaches novice users how to assemble a computer motherboard. An evaluation found that our intelligent AR system improved test scores by 25% and that task performance was 30% faster compared to the same AR training system without intelligent support. We conclude that using intelligent AR tutor can significantly improve learning compared to traditional AR training.

**Keywords:** augmented reality, intelligent tutoring, assembly skills

## 1 Introduction

Augmented Reality (AR) allows the user's view of reality to be combined with virtual content that appears to be spatially registered in the real world [1]. One area of particular interest is the use of AR to assist with training for manual assembly and maintenance tasks. Whether a person is putting together furniture or repairing a car engine, these types of tasks are inherently spatial in nature, and can be difficult to teach without supervision. Many systems include instruction manuals containing diagrams that detail the necessary steps to be performed, but these can be difficult to interpret. Video tutorials can be more effective, but the user must repeatedly switch between the video and the real-world environment.

AR has the capacity to deliver hands-on training where users receive visual instructions in the context of the real world objects. Instead of reading a paper manual, a person could look at a car engine while the AR display shows the parts that need to be adjusted and the sequence of steps required. Earlier research in this area has largely involved procedural tasks where the user follows visual cues to perform a series of steps, with the focus on maximizing the user's efficiency while using the AR system.

Boeing developed one of the first industrial AR applications [2], which assisted with assembling aircraft wire bundles, with the goal of improving worker efficiency and lowering costs. Henderson and Feiner [3] developed an AR application to support military mechanics conducting routine maintenance tasks inside an armored vehicle turret. They found that the use of AR allowed the subjects to locate components 56% faster than when using traditional untracked head-up displays (HUDs) and 47% faster than using standard computer monitors.

Baird and Barfield [4] studied the assembly of components on a computer motherboard. Participants were asked to perform the task using printed materials, slides presented on a computer monitor, or screen-fixed textual instructions on opaque and see-through HMDs. The test subjects completed the assembly task significantly faster and with fewer errors when using the HMD displays. However, they did not employ spatially-registered AR and users had to follow a rigid series of assembly steps.

While there has been much research into the use of AR to assist with assembly and maintenance, existing systems generally focus on improving user performance while using the AR interface as opposed to teaching the user how to perform the task without assistance. Most systems guide the user through a fixed series of steps and provide minimal feedback when the user makes a mistake, which is not conducive to learning. The learning experience is the same for every user, and there is little regard for whether learning is actually taking place.

In contrast, Intelligent Tutoring Systems (ITSs) provide customized instruction to each student [5]. ITSs have been applied successfully to a variety of topics, such as physics, algebra and database design [6-8]. However, until now, there has been little research investigating how ITSs can be combined with AR technology for training. ITSs have been created for a wide variety of domains, but the interfaces employed are normally text-based or 2D graphical applets, which limit their ability to convey spatial or physical concepts. There have been a few studies investigating the combination of ITSs with Virtual Reality such as [9-11], but very few examining the combination of ITSs with AR. The integration of AR interfaces with ITSs creates new possibilities for both fields and could improve the way we acquire practical skills.

A few projects claim to have created intelligent AR applications, but in practice these systems are minimally intelligent and do not employ domain, student and pedagogical models to provide adaptive tutoring. For example, Qiao et al. [12] developed an AR system that teaches users about the instruments in a cockpit. Their system detects which cockpit component the user is looking at and then displays relevant information describing the component's function. This context-based interface is very different from the kind of intelligence that is employed in the ITSs.

Feiner et al. [13] developed a prototype that employed knowledge-based AR. Their system used an intelligent Intent-Based Illustration System to dynamically generate graphics based on the communicative intent of the AR system at any particular moment. While this system is intelligent in how it generates the graphics for the user, it is neither intelligent from a training or tutoring standpoint nor adaptive.

The primary focus of our research is to explore the combination of AR with ITSs for an assembly task. We present the architecture of our intelligent AR system and its components in the next Section. Our research question is whether intelligent AR-based training enables users to learn and retain assembly skills more effectively than

traditional AR training approaches. To address this question, we performed an evaluation study described in Section 3. The results strongly support our conclusion that using an intelligent AR tutor can significantly improve the learning outcome over traditional AR training.

## 2 The Architecture and Development of MAT

We developed the Motherboard Assembly Tutor (MAT), an intelligent AR system for training users how to assemble components on a computer motherboard, including identifying individual components, installing memory, processors, and heat sinks. Figure 1 shows the system's architecture, which is designed to be as modular as possible so that it can be easily adapted for new assembly and maintenance tasks. The display elements and the domain model must be customized for each type of task, but the underlying software architecture, scaffolding algorithms and other back-end processing remains the same.

The communication module relays information between the AR interface and the ITS. The ITS controls what the user sees via the interface, and the AR interface tells the ITS what the user is doing. The AR interface encapsulates the video capture, tracking system, display and keyboard input. It uses 3D graphics, animations, audio and text, which are blended with the student's view of reality via a head-mounted display. The interface uses a camera to observe the student's behaviour, and the communication module sends the necessary data to the ITS via XML remote procedure calls over a TCP/IP network connection. The ITS analyzes the data, provides feedback about student performance and decides what material to present next. We describe the ITS first, followed by the description of the AR interface.

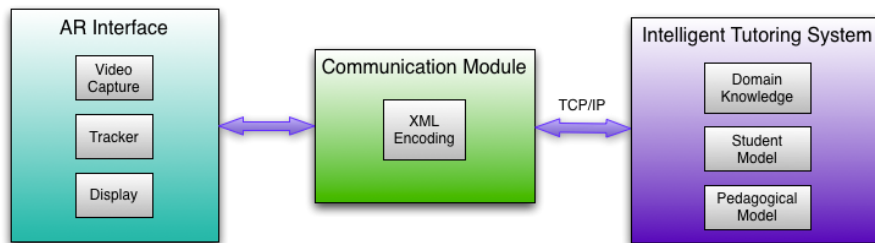


Fig. 1. The architecture of MAT

### 2.1 Developing the Intelligent Support

The intelligent tutoring support was developed in ASPIRE, an authoring system and deployment environment for constraint-based tutors [14]. The first stage of the authoring process involved describing characteristics of the task and composing an ontology of the relevant domain concepts. In the case of the motherboard assembly tutor, the assembly task is procedural in nature consisting of 18 steps to be completed, such as opening the processor enclosure and inserting the processor in the correct orientation. Each concept in the domain ontology has a number of properties and

relationships to other domain concepts. For example, in the case of a memory slot, an important property is an indicator of whether the slot is open or not, since the slot must be opened before the memory can be installed. This property is represented as a Boolean value. There are 14 domain concepts in the ontology.

Next, we specified the solution structure by indicating which ontology concepts are involved with each problem-solving step. For example, installing computer memory involves four steps: (1) Identifying and picking up the memory component, (2) opening the locking levers at the ends of the memory slot, (3) aligning the memory with the slot in the correct orientation, and (4) pushing the memory down into the slot until it locks. Each of these steps has at least one concept associated with it, and each concept has properties that are used to determine whether the student's solution is correct. In the case of the *open locking levers* step, the ITS uses the Boolean *isOpen* property of the *MemorySlot* concept to determine whether the slot has been successfully opened or not. The value of the Boolean property is set via the AR interface, which is described in the next section.

The following step was to create the interface that the students would use. ASPIRE supports text-based and graphical interfaces, and also communicates over a network via a remote procedure call (RPC) protocol, which allows it to communicate with an external AR interface. In the case of the motherboard assembly tutor, the AR front-end communicates with ASPIRE directly over a network.

We then specified a set of problems with their solutions. The problem structure describes steps that apply to all motherboards, while a particular problem and associated solutions apply to a specific brand and model of motherboard. ASPIRE allows multiple solutions to be specified for each problem. In the case of motherboard assembly, there is often only one way to correctly install each component, but this is not always the case. For example, a memory module can be inserted into one of several slots, and a heat sink can sometimes be installed in more than one orientation. Accepting these different configurations as correct solutions gives the student more flexibility when solving the problem and enhances learning.

Using the information provided in the domain ontology, problem/solution structures and the set of problems with solutions, ASPIRE generated the domain model consisting of 275 constraints. We tailored the constraints by changing the feedback messages that ASPIRE generates automatically, so that the feedback is more useful for the students.

## 2.2 AR Interface Design

The AR interface presents problems and other information from the ITS to the student. The tracking module calculates the pose of the computer motherboard and its components relative to the camera affixed to the head-mounted display. This serves two fundamental purposes: (1) It allows the display module to render 3D graphics in an AR view of the real world, and (2) the tracker sends information about the relative positions of the motherboard components to the ITS, which allows it to analyze the user's behavior, provide feedback and make changes to the teaching approach as necessary. The bulk of the work performed in the tracking module is handled by the un-

derlying osgART software library [15], which uses the used the ARToolkit marker tracking approach [16].

All of the graphics are generated by the OpenSceneGraph<sup>1</sup> computer graphics library (OSG), which has been integrated into the osgART software package. OSG is based on the standard OpenGL<sup>2</sup> API, and provides a robust scene graph structure. In addition to built-in support for materials, textures, lighting and shaders, OSG has a set of plug-ins that allow it to handle a wide variety of file formats for images, 3D models and sound. We created accurate 3D models of the components to be installed on the computer motherboard, including memory, processor, graphics card, TV tuner card and heatsink. Models were also produced for relevant parts of the motherboard itself, such as the processor enclosure and memory securing mechanisms. Other 3D models, such as arrows, were created to guide the user through the tutoring process.

Figure 2.a shows a first-person view of the display for the TV tuner installation task. The insertion animation is not visible in the picture. The models were then animated to illustrate the proper installation procedures. For example, the graphics card is visibly pushed downward into the PCI express slot, and the processor enclosure is opened before the processor is inserted. The animations were embedded into the exported 3D model files, which can be loaded directly into the display module by the appropriate plug-in in the OpenSceneGraph software library.



**Fig. 2.** a) First-person view of the AR display for part of the TV tuner installation task. The red-colored 3D model indicates where the component should go. b) A participant using the tutor

In addition to the spatially-registered 3D models that are anchored to a position within the scene, we developed a screen-aligned head-up display (HUD) for displaying text messages from the ITS. As the user looks around, the HUD components always stay in the same place on the screen. The ITS messages consist of instructions and positive/negative feedback. The text is displayed across the top of the screen and is highlighted with a semi-transparent background that changes color based on the message type. Instructions are blue (such as in Fig. 2a), positive feedback is green and

---

<sup>1</sup> [www.openscenegraph.org](http://www.openscenegraph.org)

<sup>2</sup> [www.opengl.org](http://www.opengl.org)

negative feedback is red. The HUD also utilizes text-to-speech technology to read the messages to the user, via the Microsoft Speech API.

The hardware setup for the AR interface consists of a head-mounted display, a camera, a MS Windows computer and the ARToolkit fiducial markers used for tracking (Fig. 2b). An Intel motherboard was selected for use with the computer assembly, as well as five generic hardware components to be installed: memory, processor, graphics card, TV tuner card and heatsink. At least one unique marker was attached to each component to enable the system to identify and track its position. The motherboard itself was mounted on a sturdy wooden surface and surrounded with a configuration of eight separate markers. This group of markers works together with the tracking system to limit the effects of marker occlusion as users look around and move their arms during the installation procedures. As long as the camera can see at least one of the eight markers, the tracking system is able to determine the relative position and orientation of the motherboard.

The HMD and camera combination chosen for the project is the Wrap 920AR model produced by Vuzix<sup>3</sup>, which has a resolution of 1024x768 pixels with a 31-degree horizontal field of view. It supports stereoscopic viewing, and the front of the display is outfitted with two cameras for stereo video capture at 640x480 at 30 frames per second. The device connects to a computer via the standard VGA interface and also delivers audio via earbud headphones.

### 3 Study

We conducted a study in which we compared the intelligent AR system with a traditional AR tutor. The goal of the study was to determine the difference in knowledge retention between the two approaches. The evaluation was split into two phases: a training phase and a testing phase (without the tutor) that measured the extent to which the participants retained the knowledge they acquired.

The traditional AR training proceeds linearly through the assembly steps like slides in a slideshow. It does not customize the experience to each individual student: it simply shows the student what needs to be done for each step. In contrast, the intelligent AR system controls the ordering of the assembly steps and can make decisions about what material to present next based on the student's performance. Both tutors have the same interface and provide the same visual and oral instructions for each step, so the only differences lie in the features directly related to the ITS. Whether using the intelligent or traditional tutor, the student indicates that he/she is finished with the current step by pressing a button. If the solution is incorrect, the intelligent tutor prevents the student from proceeding to the next step and provides a specific feedback message, while the traditional tutor always proceeds regardless.

There were 16 participants who were randomly allocated to one of the conditions. The experimental group used the intelligent AR tutor, while the control group used the traditional AR tutor. Great care was taken to select participants with minimal experience with computer hardware assembly. To measure this, all participants were

---

<sup>3</sup> <http://www.vuzix.com/consumer/produces/wrap920ar.html>

given a written pre-test asking them to identify the five hardware components and their position on the motherboard. The participants also rated their prior hardware experience on a scale from one (not experienced) to seven (very experienced). All of the participants were university students aged 18-45 (11 males and 5 females).

Following the pre-test, the participants were given an orientation to the AR tutor (intelligent or traditional) and its operation procedures. After they put on the head-mounted display, the tutor guided them through the process of identifying and installing five motherboard components: memory, processor, graphics card, TV tuner card and heatsink. After all of the components were assembled, the tutoring phase was complete and the participants were given a written post-test that was similar to the pre-test to measure how well they learned from the tutor. The two written tests covered the same material, but were not identical.

Immediately after the written post-test, the participants were asked to perform a physical post-test in which they attempted to assemble the motherboard components once more, this time without the help of the tutor. The aim of the physical post-test was to measure how well the participants retained the physical assembly knowledge gained from the tutoring process. Given only the name of each component, the participants had to correctly identify and install them one by one. In addition to qualitative observations, a number of quantitative measures were taken during this process, including task completion time and error counts.

Finally, the participants completed a questionnaire, which prompted them to provide detailed feedback regarding their experience with the tutor. In addition to asking about prior hardware experience, the questionnaire contained a variety of questions with Likert-scale ratings. These asked the participants to indicate whether they thought the tutor was effective, whether they were satisfied with the 3D AR content, whether they thought the AR training system was more effective than other types of media such as videos or paper manuals, and whether they felt physically or mentally stressed during the tutoring process. Participants also had the opportunity to provide additional written feedback.

## 4 Results

Table 1 summarizes the written pre-test and post-test scores for the two groups. The maximum score on each tests was 10 marks. There was no significant difference between the two groups on the pre-test performance. There was also no significant difference in the times both groups spent on working with the tutoring systems. The performance of both groups increased significantly between the pre- and the post-test, yielding  $t(7) = 7.165$ ,  $p < .0002$  for the experimental group, and  $t(7) = 5.291$ ,  $p < .002$  for the control group. Both of these values are significantly less than the Bonferroni-corrected  $\alpha$  value of .0083 (.05/6), which makes a very strong case for the effectiveness of both AR tutors.

The post-test performance of the experimental group is significantly higher than that of the control group ( $t(14) = 3.374$ ,  $p < .005$ ). This is less than the Bonferroni-corrected value of .0083 (.05/6), so the intelligent AR tutor produced a significantly better learning outcome than the non-intelligent AR tutor. There is also a significant

difference between the normalized learning gains of the two groups ( $t(14) = 2.198$ ,  $p < .05$ ). The effect size (Cohen's  $d$ ) is 0.981, which is a significant improvement.

Table 1 also reports the number of errors made and the total completion time to install all five motherboard components during the physical post-test. The errors generally fit into two categories: failing to match a name with the correct component, or incorrectly performing an installation procedure. There was no significant difference on the number of errors made, but the experimental group participants completed the task significantly faster than their peers ( $t(14) = 2.9$ ,  $p < .02$ ).

**Table 1.** Mean and standard deviations for two groups

Group	Pre-test	Post-test	Normalized Gain	Time (s)	Errors
Exper.	2.50 (2.27)	9.13 (1.13)	0.66(0.26)	56.56 (11.31)	0.50 (0.93)
Control	2.63 (1.92)	6.63 (1.77)	0.40 (0.21)	81.13 (21.11)	1.00 (0.93)

The questionnaire feedback was positive for both tutors. Most participants felt that the visual step-by-step instructions were very helpful, allowing them to proceed at their own pace. The immersive first-person experience provided by the head-mounted display was engaging, and the system as a whole was interesting and fun to use. Some of these responses can be attributed to the novelty factor associated with AR, but the fact remains that the participants generally found the tutors to be both effective and entertaining. Many of the experimental group participants found the ITS feedback very helpful. One criticism stemmed from the fact that the textual instructions were screen-aligned in typical HUD fashion. Reading the text required the participants to shift their focus from looking into the scene to looking at the text displayed on the surface of the screen. It may have been more natural to use spatially-registered text that appeared within the scene to keep the students immersed in the AR environment. Other criticisms addressed the tracking performance. The virtual content would sometimes jiggle or disappear entirely when the tracking system was unable to obtain enough information about the markers. These issues could be addressed with a more robust tracking approach, perhaps one that utilizes multiple cameras and tracks the natural features of the motherboard components without markers.

While the participants found determining the correct position of the components to be relatively easy, determining the proper orientation was more difficult. This was partially due to a lack of orientation cues in some of the virtual content shown. The memory and processor are essentially symmetrical in shape, and it can be difficult to determine which direction the virtual rendering is facing when there are no distinguishing features. In these cases, it would be helpful to have some additional AR cues to help the student infer the correct orientation. One idea would be to attach virtual arrows to the motherboard slot as well as the actual component to be inserted, prompting the student to line up the arrows with each other. When this type of orientation mistake occurred, the intelligent AR tutor was able to detect the error and inform the student that the orientation was incorrect. The participant was required to correct the mistake before being allowed to proceed. The traditional tutor was unable to observe or correct errors, and they often went unnoticed by the student. In these cases, the student typically made similar mistakes during the post-test. This supports the claim that the ITS feedback improved the learning outcome over the traditional AR training approach, particularly where it was easy to make a mistake.



The results of the study confirm the overarching hypothesis that the use of ITSs with AR training for assembly tasks significantly improves the learning outcome over traditional AR approaches.

## 5 Conclusions

Augmented Reality has been repeatedly shown to improve education and training through visualization and interactivity, but most AR training systems are not intelligent. In this paper we have shown how to combine an AR interface with an Intelligent Tutoring System to provide a robust and customized learning experience for each user. To demonstrate this we created a prototype application that teaches users how to assemble hardware components on a computer motherboard. An evaluation found that our intelligent AR system improved test scores by 25% and that task performance was 30% faster compared to the same AR training system without intelligent support. From these results, we conclude that using the ITS approach can significantly improve the learning outcome over traditional AR training.

There are many future research directions that could be explored. For example, the intelligent AR tutor could be extended by integrating a virtual character into the tutoring environment. Research has shown that virtual characters can be beneficial in tutoring situations as they increase student motivation [17-19]. A 3D virtual character would allow the ITS to inhabit the world with the user, where it could give verbal instructions, make gestures and demonstrate installation procedures.

Tracking is another area in which the intelligent AR tutor can be improved. The current solution uses a fiducial marker-based approach, which has limited accuracy, poor resistance to occlusion and obtrusive markers. There are a number of better tracking approaches such as natural feature tracking or using multiple cameras to reduce the effect of occlusion. Stereoscopic cameras and depth mapping could be used to determine the three-dimensional shapes of objects. This would allow the system to generate a model of the environment on the fly, and adapt to new scenarios such as different brands of computer motherboards and components. It could also enable more complex training tasks that require more robust tracking.

Finally, more user studies need to be conducted in a wider range of training domains. Our results have shown the value of using an intelligent AR tutor in training for motherboard assembly, but it would be good to examine the educational benefits in other assembly or maintenance tasks.

## References

1. Azuma, R.T.: A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, pp. 355–385, 1997.
2. Caudell, T., Mizell, D.: Augmented reality: an application of heads-up display technology to manual manufacturing processes. *Proc. 25<sup>th</sup> Hawaii Int. Conf. System Sciences*, vol. 2, pp. 659 – 669, 1992.
3. Henderson, S. J., Feiner, S.: Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. *Proc. 8<sup>th</sup> Int. Symp. Mixed and Augmented Reality*, pp. 135-144. IEEE, 2009.

4. Baird, K.M., Barfield, W.: Evaluating the effectiveness of augmented reality displays for a manual assembly task, *Virtual Reality*, 4(4), 250–259, 1999.
5. Psozka, J., Mutter, S.A.: *Intelligent Tutoring Systems: Lessons Learned*. Lawrence Erlbaum Associates, 1988.
6. VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., Wintersgill, M.: The Andes Physics Tutoring System: Lessons Learned'. *Artificial Intelligence in Education*, 15, 147-204, 2005.
7. Koedinger, K. R., Anderson, J. R., Hadley, W. H., Mark, M.A.: Intelligent tutoring goes to the big city. *Artificial Intelligence in Education*, 8, 30-43, 1997.
8. Mitrovic, A. Fifteen years of Constraint-Based Tutors: What we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, 22(1-2), 39-72, 2012.
9. Mendez, G., Herrero, P., de Antonio, A.: Intelligent virtual environments for training in nuclear power plants. Proc.6<sup>th</sup> Int. Conf. Enterprise Information Systems, 2004.
10. Evers, M., Nijholt, A.: Jacob - An animated instruction agent in virtual reality," *Advances in Multimodal Interfaces*, pp. 526–533, Springer, 2000.
11. Fournier-Viger, P., Nkambou, R., Nguifo, E. Exploiting partial problem spaces learned from users' interactions to provide key tutoring services in procedural and ill-defined domains. In Dimitrova, V., Mizoguchi, R., du Boulay, B., Graesser, A. (eds.) Proc. 14<sup>th</sup> Int. Conf. Artificial Intelligence in Education, pp. 383-390, 2009.
12. Qiao, Y., Xie, X., Sun, T.: Design for the cockpit intelligent tutoring system based on augmented reality. Proc. Int. Symp. Computational Intelligence and Design, vol 2, pp. 224-227, 2008.
13. Feiner, S., Macintyre, B., Seligmann, D.: Knowledge-based augmented reality. *Communications of ACM*, 36, 53–62, 1993.
14. Mitrovic, A., Martin, B. Suraweera, P., Zakharov, K., Milik, N., Holland, J., McGuigan, N. ASPIRE: an authoring system and deployment environment for constraint-based tutors. *Artificial Intelligence in Education*, 19(2), 155-188, 2009.
15. Looser, J., Grasset, R., Seichter, H., Billinghamurst, M.: OSGART-A pragmatic approach to MR, 5<sup>th</sup> IEEE and ACM Int. Symp.on Mixed and Augmented Reality, pp. 22–25, 2006.
16. Kato, H., Billinghamurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. Proc. 2<sup>nd</sup> IEEE and ACM Int.Workshop on Augmented Reality, pp. 85 – 94, 1999.
17. Johnson, W. L., Rickel, J.W., Lester, J.C.: Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *Artificial intelligence in education* 11(1), 47-78, 2000.
18. Liu, Z., Pan, Z.: An emotion model of 3d virtual characters in intelligent virtual environment. In J. Tao, T. Tan, and R. Picard (eds.), *Affective Computing and Intelligent Interaction*, vol. 3784, Lecture Notes in Computer Science, pp. 629–636, Springer Berlin / Heidelberg, 2005.
19. Gulz, A., Haake, M.: Design of animated pedagogical agents—A look at their look. *Human-Computer Studies*, 64(4), 322-339, 2006.