A Distributed Augmented Reality Training Architecture For Distributed Cognitive Intelligent Tutoring Paradigms

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY-NC-SA 4.0

SUBMISSION DATE / POSTED DATE

10-02-2021 / 11-02-2021

CITATION

Herbert, Bradley; Baghaei, Nilufar; Billingham, Mark; Wigley, Grant (2020): A Distributed Augmented Reality Training Architecture For Distributed Cognitive Intelligent Tutoring Paradigms. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.13840151.v1>

DOI

[10.36227/techrxiv.13840151.v1](https://doi.org/10.36227/techrxiv.13840151.v1)

A Distributed Augmented Reality Training Architecture For Distributed Cognitive Intelligent Tutoring Paradigms

Bradley Herbert · Nilufar Baghaei · ·
Grant Wigley · Mark Billinghamurst

Received: 14 October 2020

Abstract Modern training typically incorporates real-world training applications. Augmented Reality (AR) technologies support this by overlaying virtual objects in real-world 3-Dimensional (3D) space. However, integrating instruction into AR is challenging because of technological and educational considerations. One reason is a lack of architecture for supporting Intelligent Tutoring Systems (ITSs) in AR training domains. We present a novel modular agent-based Distributed Augmented Reality Training (DART) architecture for ITSs to address two key AR challenges: (1) a decoupling of the display and tracking components and (2) support for modularity. Modular agents communicate with each other over a network, allowing them to be easily swapped out and replaced to support differing needs. Our motivation is driven by the fact that AR technologies are vary considerably and an ITS architecture would need to be flexible enough to support these requirements. Finally, we believe that our novel architecture will appeal to practical designers of ITSs and to the more theoretical educators who wish to use such systems to simulate and broaden research in the distributed cognitive educational theories.

B. Herbert
University of South Australia, Mawson Lakes, Adelaide
E-mail: bradley.herbert@mymail.unisa.edu.au
ORCID: 0000-0002-8853-1740

N. Baghaei
Massey University, Auckland, New Zealand
E-mail: N.Baghaei@massey.ac.nz
ORCID: 0000-0003-1776-7075

M. Billinghamurst
University of Auckland, Auckland, New Zealand
E-mail: mark.billinghurst@unisa.edu.au
ORCID: 0000-0003-4172-6759

G. Wigley
University of South Australia
E-mail: grant.wigley@unisa.edu.au ORCID: 0000-0003-2036-7320

Keywords Augmented Reality · Intelligent Tutoring Systems · Distributed Cognition · architectures · modularity · training · education

1 Introduction

This paper proposes a novel distributed agent-based architecture for Augmented Reality (AR)-based Intelligent Tutoring Systems (ITSs). AR creates an enhanced representation of the real-world by overlaying virtual 3-Dimensional (3D) objects or annotations on real world space in real-time (Azuma 1997; Milgram et al. 1995). Despite its promising potential in supporting educational and training tasks with a manufacturing or assembly component, few of them incorporate instructional design (Santos et al. 2014; Westerfield et al. 2015). Incorporating instructional design methods into an AR training system can be accomplished by embedding AR technologies into an ITS (Herbert et al. 2018). In practice, however, such tends to be time-consuming and costly (Sottilare et al. 2012). To understand why, it is important to first understand what an ITS is. In simplest terms, an ITS is any system that provides personalised training and task support using task sequencing, error detection/correction support and problem solving support (Latham et al. 2010). Under the hood, however, an ITS is significantly more complex, often programmed to replicate a limited subset of human cognition responsible for learning (Anderson et al. 1995). For example, first-generation ITSs were cognitive tutors designed primarily as research-based tools for simulating cognitive processes, enabling researchers to understand the cognitive mechanisms involved in learning (Anderson et al. 1995; Kim and Nam 2020). They were not intended to be used in conjunction with modern technologies, such as display and tracking technologies. The paradigm shift in technology from a desktop-centric one to one that uses the real-world has potential to radically alter education, and unfortunately, ITS paradigms have not kept up with this radical shift (Billinghurst and Duenser 2012; Herbert et al. 2018). Consequently, ITSs tend to be limited in their capability to be integrated with AR technologies.

However, the more difficult barrier is the major paradigm shift in educational psychology itself. In the 1990s, Perkins (1993) proposed a view of education called person-plus, which also included the surrounding environment. In this new person-plus view of education, a distributed cognitive theory was proposed, which some authors, have likened to a second cognitive revolution called post-cognitivism (Hunt 1999). In this view, the resources rather than being contained within a single cognitive information processing system are distributed across various people, where the learner constitutes just one component of a larger learning system (Xu and Clarke 2012).

Under the cognitive view, an ITS is a standalone system that uses Artificial Intelligence (AI) techniques to model the underlying cognitive processes used for learning (Almurshidi and Abu-Naser 2016). This design pattern is modelled after the cognitive-centric view of education (Anderson 1993). In the post-cognitive view, cognition is a distributed system. So, we argue that a modern

ITS should be designed following the principles of the post-cognitivist view. While this sounds easy and promising, there currently exists no architecture for specifying what a post-cognitivist AR tutor would look like. In this paper, we propose a potential architecture for AR-based ITSs to address this design requirement. This paper is framed around the belief that ITSs have moved from specialised system that aimed to model human cognition to a system that uses various techniques for personalised instruction (Nesbit et al. 2014; Baker 2016). These design shifts are indications not of limitations with the technology but with evolving educational needs.

Our proposed (Distributed Augmented Reality Training (DART)) architecture consists of three components: (1) an Environment component that collects data about the external environment; (2) a Learning component that performs tutoring-related functions and (3) A Backbone component that ties the components together. Distributed agents fit into one of these components and communicate with each other over a computer network. The Environment component consists of one or more Sensor agents that communicate with the external environment (either the real world, a computer program or an AR tracking system). The Learning component models what is understood as the human-cognitive architecture using modelling techniques used by standalone ITSs. Since distributed cognition also presupposes the existence of human cognition but as part of a larger system, it follows that cognitive tutors based on Adaptive Control of Thought-Rational (ACT-R) theory (Anderson 1976) or Ohlsson’s problem solving theories (Ohlsson 1994), model cognition to an accurate degree and these tutors should comprise of the part of the overall system.

The three aims of our designed architecture includes:

1. Accurately replicate the current state of cognition in education as part of a distributed system. So we argue that the individual parts of the ITS should be distributed similarly.
2. Since distributed cognition involves combining of multiple cognitive architectures, we argue that cognitive models should be combined into a distributed ITS. This is evident from the fact that multiple different tutoring systems with different design assumptions have been found to be equally effective (Roll et al. 2010; Kodaganallur et al. 2005).
3. Finally, we extend upon existing distributed architectures by decoupling non-ITS components (such as the tracking and User Interface (UI) components) to create AR-based ITSs.

In this work, we are concerned with extending upon existing distributed ITSs architectures by proposing a particular distributed agent-based architecture that addresses AR requirements for an ITS system. While there are potential existing distributed ITS architectures such as the i-Help architecture (Vassileva et al. 2003), none of these architectures are particularly designed for AR-based tutors (Herbert et al. 2018). Even if such architectures could be reasonably used, they focus more on the technological problems of integration more so that embodying a post-cognitive perspective of education. Our

contributions have potential to provide the basic framework for researchers looking to design and develop AR-based ITSs and for those looking to simulate post-cognitivist theories of education. While the primary focus is on ITS architecture design, our architecture could theoretically be applied to any computer-based training system and because the AR components are decoupled from the other tutoring functions, it can be easily adapted to non-AR ITSs with relative ease. In summary, this paper presents the following contributions:

1. The use of the post-cognitivist idea of distributed cognition as the basis for designing the DART architecture.
2. An adaptation of the distributed architecture to particularly address the requirements of AR-based tutors, which differs from non-AR ITSs (Herbert et al. 2018).

This paper is divided into several parts. The first part reviews different types of ITS architectures and their application to AR training systems. The second part outlines the rationale for designing our architecture. The third part describes our proposed architecture and the design requirements. The fourth part presents the Intelligent Software Training Platform (ISTP), an implementation of a platform using our proposed architecture. The fifth part discusses the lessons learned and lists the potential limitations of our proposed architecture and some potential ways these limitations might be addressed. Finally, the seventh part summarises this work and proposes potential future work.

2 Related Work

2.1 Background

The Adaptive Control of Thought-Rational (ACT-R) cognitive theory, as first proposed in 1976, suggests that within the brain exists a hypothetical information processing system which segments knowledge and large amounts of information into smaller digestible chunks which are then encoded into a series of generalised rules called productions (Anderson 1976). Knowledge acquisition consists of two fundamental phases (Anderson 1982): (1) the declarative phase, the encoding of productions and (2) The procedural phase where external environmental stimuli triggers the cognitive information system into retrieving the productions from a large internal data store. From these productions, more refined productions are created about how to apply the knowledge in a procedural process. Therefore, practice is necessary to stimulate the second phase. Despite this, much about the mechanics of the encoding process and what triggers certain productions was far from clear (Anderson 1993). So, early cognitive psychologists like John Anderson sought to build simulation systems

to replicate these cognitive processes using programming techniques due to perceiving similarities between computer systems and human cognition. In fact, human cognition was viewed as a very complex computer system. Thus, the reasoning was that human cognition could be re-implemented or simulated using a computer (Xu and Clarke 2012). Despite this, success varied. Anderson implemented the ACT-R theory into a series of computer systems which he called cognitive tutors (Anderson et al. 1989). These systems became the first-generation Intelligent Tutoring Systems (ITSs) (Corbett et al. 1999). However, the goal was to simulate ACT-R processes rather than as an education support tool (Anderson 1996). This was because much education despite the emergence of cognitivism was still largely based on the behaviourism theories of Watson (1913), Hull (Hilgard 1948) and Skinner (1950). There was, therefore, very little motivation to use ITSs in a classroom environment to aid with teaching (Philip et al. 2013).

In the 1980s and 1990s, Self (1990) and other researchers proposed the prospect of using cognitive tutors in education after it was demonstrated that they improved knowledge retention in Lisp and mathematical education (Anderson et al. 1989). However, while the human cognitive information processing system could theoretically create many production rules easily, the same could not be said for Artificial Intelligence (AI) systems (Ohlsson 1994). The production rules used by cognitive tutors were for the most part expensive and slow and required expert knowledge (Murray 1999). Often, ITSs had a limited number of production rules that made them considerably less effective than a human tutor in the real world (Murray 1996; Baker 2016). So, other cognitive psychologists were motivated to come up with alternative modelling paradigms that would be easier to implement. One such example was Constraint-Based Modelling (CBM) proposed by Ohlsson (1994). In some instances, the CBM-based tutors showed promising results in problem-solving domains like learning a non-sequential programming language like Structured Query Language (SQL) (Mitrovic and Ohlsson 1999; Mitrovic et al. 2003). The reasoning was that CBM identified imperfections in the solution itself, rather than in cognition. The assumption was that the cognitive process used to infer the answer was irrelevant. Rather, the importance was on the accuracy of the solution and identifying which part(s) of it were incorrect. Ohlsson (1996) reasoned that learners, despite supposedly, having internalised the production rules, sometimes still made errors. Similarly, despite lacking the production rules, learners were sometimes still able to problem-solve and arrive at an equally-satisfying solution. One explanation was that problem solving relied on different cognitive processes (Sweller 2003).

Since the early to mid 2000s, ITSs performed functions beyond simply cognition. For example, ITSs tend to model the affective dimensions as well (Nwana 1990; D'mello and Graesser 2013).

Augmented Reality (AR) uses tracking, display and interaction technologies to superimpose virtual 2-Dimensional (2D) objects or digital information directly onto the real-world (Billinghurst et al. 2015; Azuma 1997; Santos et al. 2014). It has broad applications in education, (Dalim et al. 2016; Billinghurst

2002), maintenance (Henderson and Feiner 2009; Yin et al. 2014; Gilbert 2016) and training (Westerfield et al. 2015; Tang et al. 2003; Werrlich et al. 2017; Sottolare et al. 2017; Sottolare and LaViola 2015). An ITS supports all three of these domains. First, in education, AR helps learners learn the abstract structure of a spoken language (Dalim et al. 2016). Second, in maintenance, it helps guide learners through performing a repair such as connecting cables or wires (Herbert et al. 2017a) and finally, in training, to help develop the skills necessary that would allow one to complete the task without any assistance or mistakes (LaViola et al. 2015). In the early 2010s, ITSs have been used in few AR training domains (Westerfield et al. 2015; Herbert et al. 2017b, 2018).

These diverse uses of ITSs suggest that learning is primarily a distributed system, which encompasses both procedural and problem-solving learning. So it would be evident that multiple tutors would be required. For example, while both production modelling and CBM can be both adapted to support the domain where they are not intend to, this brute-force approach can result in a diminished tutor due to being simpler than intended.

One approach is to combine multiple modelling paradigms together, such as a hybrid system incorporating both CBM and model-tracing approaches (Roll et al. 2010). While effective, ITSs were intended to embody the cognitive theories and to think like a human. So, a more promising approach would be to use two or three tutors: one that specialises in problem solving and one that specialises in procedural knowledge. Each tutor would be coordinated by a third tutor. Such would be possible with a distributed architecture, which we discuss next.

2.2 Architectures

Early cognitive-based ITSs (e.g. Anderson’s LISP Tutor (Anderson et al. 1989)) embody a cognitive-centric view of educational psychology. Like the brain, it was assumed that the ITSs should be composed of several components self-contained within one structure (Anderson et al. 1995). As ITSs emerged as a promising educational tool, the cognitive modeller was combined with a tutoring model which defined how to teach and a student model which replicated the learner’s cognitive states (Major et al. 1997). As Human Computer Interaction (HCI) research evolved, the importance of the User Interface (UI) became stressed and so the existing three-part ITS architecture was combined with a tutoring UI component (Nwana 1990).

Several architectures were proposed to support integrating training systems into ITSs such as: (a) distributed centralised architectures that incorporate separate agents, which communicate information to a central back end system (e.g. KnowledgeTree (Brusilovsky 2004) and Generalized Intelligent Framework for Tutoring (GIFT) (Goldberg and Cannon-Bowers 2013)) and (2) distributed decentralised architectures where each agent builds its own learner and/or domain models (e.g. i-HELP (Vassileva et al. 2003)).

2.2.1 Distributed Centralised

Distributed centralised architectures typically integrate domain and learner modelling into one or more agents with UI and other components distributed across the network. For example, ARWILD combines AR and spatial mapping technologies to create an engaging training experience for military superior officer etiquette (LaViola et al. 2015). The AR application was an agent integrated into the GIFT architecture.

Similar systems for non-AR domains have also been designed. For example, Cheikes (1995) proposed an agent-based architecture for an ITS that consisted of a core module and individual agents. Ritter et. al. (Ritter and Koedinger 1996) proposed a method for integrating a tutoring module that monitors the state of a computer program such as Microsoft Excel and/or sends actions and triggers events. Oscar is a conversational mixed initiative tutor that adapts in real-time based on learner styles (Latham et al. 2010). A user interface agent communicates with a central controller, which communicates with learner model and learner style agents. Information about a learner is stored in a single module.

Similarly, the architecture of Motherboard Assembly Tutor (MAT) decoupled the UI from the other ITSs components. Almiyad et al. (2017) decoupled the UI component from the AR component because the UI and AR are two separate systems that are not necessarily integrated (Azuma 1997). PC-based Open-Architecture Reconfigurable Training Systems (PORTSs) integrates a 3-Dimensional (3D) virtual world application, Second Life, into the PORTS ITS by decoupling the UI from the PORTS back end (Ludwig et al. 2010).

2.2.2 Distributed decentralised

Distributed decentralised architectures, like distributed centralised architectures distribute agents across the network, but learner or domain model construction is performed by each agent independently. For example, i-Help (Vasileva et al. 2003) has numerous agents, each responsible for modelling learner knowledge with respect to their role. Diagnostic agents model learner's deficiencies. Other agents are responsible for modelling other learner aspects. This is the closest representation to a system that distributes cognitive processes across separate entities. However, it does not address the AR requirements. For example, there is no strict separation of the UI and spatial components. Although AR is primarily a user interface, it also consists of tracking and display technological components, which may benefit from being decoupled (Billinghurst et al. 2015).

2.2.3 Combined Architectures

ITS architectures can be combined together to create a unified system that supplements weaknesses in each architecture. For example, the (conceptual) Augmented Reality Adaptive Tutor (ARAT) architecture (Herbert et al. 2018)

combines a decentralised multi-agent architecture with a single centralised architecture. The ITS and UI are decoupled, but the UI also constructs the learner and spatial models using information received from the ITS agent.

2.2.4 Limitations

Distributed centralised architectures, while promising, have several limitations. First, architectures such as the MAT architecture (Westerfield et al. 2015) does not decouple the spatial representation from the domain model. Second, these architectures may result in loss of tutoring functionality because they need to be generalised enough to support agent integration (Brusilovsky et al. 2009; Hampton et al. 2018). Third, in many cases, external applications cannot be simply plugged into the agent-based architecture (Ritter 2016). They may use differing standards that make them hard to integrate.

Distributed decentralised architectures also have several limitations. First, migrating from one tutoring system to another has potential to result in lost functionality (Brusilovsky et al. 2005). For example, an ITS that usually performs all of the modelling must now be expected to offset some of this modelling to external agents. Second, integrating ITSs into such an architecture is difficult because semantic mapping would need to be used to determine which component is responsible for construction (Brusilovsky et al. 2005).

Finally, regardless of which architecture is used, there are still widespread barriers to produce systems that are generalised enough to promote reuse (Sottolare et al. 2012) but not too general to be ineffective (Hampton et al. 2018).

2.3 Intelligent Tutors Uncommon in AR

Despite the potential for ITSs to enhance AR-based maintenance, training and education systems, few AR systems actually use an ITS. For example, a review by Herbert et al. (2018) found a total of three AR training systems that used an ITS (i.e. Westerfield et al. (2015); LaViola et al. (2015); Almiyad et al. (2017)). Since then, a similar motherboard repair guidance system emerged claiming to integrate AI techniques (Iqbal et al. 2019), but which uses simple logic rules and a Soccer rule comprehension training system (Vargas González et al. 2017), which uses an open source ITS.

However, overall, ITSs remain uncommon. One barrier is technological and one theoretical. For instance, although, ITSs appear to be one single system, it is often the case that an ITS is a collection of interconnected subsystems or modules (Beck et al. 1996). These modules, collectively, comprise of an architecture. ITS architectures differ, sometimes considerably, between systems and is one reason for their limited use in AR (Herbert et al. 2018).

The technological barriers are easily addressed, albeit with significant but justifiable costs and resources. For example, networking protocols are commonly used to integrate ITSs with external systems (Westerfield 2012; De Jong

2016; Ludwig et al. 2010; Herbert et al. 2017a,b). A broad range of ITSs, including some older systems, include support for network integration (Westerfield 2012). Gateways or go-between agents can be used to encode or decode messages between incompatible agents (e.g. Almiyad et al. (2017) or the GIFT gateway agent (Sottolare and Brawner 2018)). Different learner models could be accessed and used to overcome conflicts in a decentralised multi-agent architecture and the UI, AR tracker and the tutoring components could be decoupled from each other.

The theoretical barriers are, however, not easily addressable. Perhaps, the reason why is because there are no standardised implementations of AR-based ITSs. For example, while there are distributed architecture suited to embody a post-cognitive education view, they do not address the key issues of AR, such as limited spatial ontologies and real-time tracker support (Westerfield 2012). The ITS needs to communicate with the tracker in real-time to provide ongoing and continuous instruction. In the next section, we outline how our proposed architecture might address some of these challenges.

2.4 Research Gaps

We identified the following research gaps:

1. The requirements and characteristics of an AR architecture integrating ITSs has not been explicitly defined.
2. There is no known architecture that specifically addresses the limitations of AR integration. Some have addressed one or two limitations.
3. While there are many ITSs architectures, few architectures are designed in the context of AR.

3 Rationale for a new architecture

The limited use of AR-based tutors results from two problems: a well-known technological barrier caused by limited standardisation and high costs and (2) a theoretical barrier that is the result of a limited architecture that embodies a cognitive theory of education. Our architecture aims to combine both to overcome both barriers in the hope that AR-based tutors will become more widespread and effective.

3.1 Technological Barriers

Perceived or real technological barriers limit integrating existing ARs into ITSs such as: (a) Limited standardisation (Herbert et al. 2018); (b) Limited spatial modelling support (Westerfield 2012; Herbert et al. 2018) and (c) User

Interface decoupling challenges. Addressing these barriers requires extensive costs and time, however, they could be reduced by using a suitable architecture (Sottolare et al. 2012).

3.1.1 Limited Standardisation

One of the recurring issues in ITS research is limited standardisation. Nwana (1990) alone cites four different ITS architectures including: three part architectures (domain, student and tutoring components); four-part architectures (domain, student, tutoring and UI components); O'Shea architectures that decouples teaching strategies, generation and administration. Nwana (1990) concludes that differing architectures result from different assumptions about instructional design. A single perfect architecture for ITSs is unlikely to result. Due to lack of standardisation, time is wasted re-implementing common components between ITSs (Sottolare and Holden 2013).

3.1.2 Limited Spatial Representation

Spatial concepts, such as the relationship between elements in 3D space typically need to be represented in an AR-based ITS (Westerfield 2012; Herbert et al. 2018). Most approaches accomplish this by representing spatial information as declarative knowledge in the domain model. However, such representation is limited and needs to be completed manually. While most spatial data is simply facts or information, they are distinct from the kind of knowledge that would be expected to exist in a domain model.

A more promising approach is a separate spatial ontology for creating a semantic mapping between the spatial relationships and the domain principles or knowledge. However, most architectures do not provide any mechanism for storing a spatial ontology separate from the domain model.

3.1.3 User Interface Decoupling

There is no standardised way for interpreting messages to/from an external application and the ITS back end logic. Due to incompatibilities between systems, migration may be difficult to achieve (Hampton et al. 2018). This is, in part, because ITSs lack the mechanisms to decouple the UI from the back end ITS. There needs to be a mechanism in ITSs to pass messages between the new UI and back end ITS. In some ITSs, decoupling the UI is not possible because tutoring logic is embedded in the UI part. In such cases, successful decoupling would require modification to the ITS back end. On the other hand, if the ITS was designed with UI decoupling in mind (e.g. ASPIRE Tutor (Westerfield et al. 2015)), this process becomes much easier.

3.2 Theoretical Barriers

Theoretical barriers may also limit the ability to implement AR-based ITSs. For instance, ACT-R models used by cognitive tutors might be insufficient for AR-based domains (Westerfield 2012). Such tutors presuppose a cognitivist-centric view of education, so forcing them to work with other educational approaches would not necessarily make learning effective. The modelling paradigms themselves might not be easily able to be decoupled from the other components of the ITS. Similarly, hands-on motor skill support is domain and individual specific and is difficult to generalise across ITSs. This may suggest that many existing ITSs, due to being standalone systems, can only capture one dimension of learning. However, they would need to be integrated with other standalone ITSs that use different modelling paradigms to cover the dimensions of learning more broadly.

3.2.1 Modelling Limitations

Many cognitive tutors, like the Lisp programming tutor (Anderson et al. 1989), were designed based on ACT-R cognitive theory, which asserts that cognition consists of procedural and declarative knowledge (Anderson 1993; Anderson et al. 1997; Mathews 2012). Following this logic, cognitive tutors use production rules to represent this model, since correcting knowledge errors more likely results in greater learning outcomes (Anderson et al. 1995). However, production and model trace tutors tend to only be effective if the entire production space can be modelled (Ohlsson 1994). Since this is rarely the case, CBM tutors can be more effective, especially in problem solving domains (Mitrovic et al. 2003). However, while some ITSs might be integrable into AR, they may not use the desired domain modelling paradigm.

3.2.2 Psychomotor Modelling

Research in sports psychology suggests that mistakes in carrying out physical tasks results from misapplied muscle movements (Moran 2012). The psychomotor domain (Simpson 1966; Ferris 2010) develops muscle skills through physical interaction with the environment. Over time, one learns to perform the task without error by developing their motor skills.

Prior to the 2010s, many ITSs generally ignored the environment (Sottolare et al. 2017). They largely focused on the cognitive aspects of learning and not how the environment played a role in learning. Consequently, many ITS architectures are similarly modelled after this assumption. This is because, according to the cognition view, imparting the right knowledge should be enough for learning to happen. One such researcher, for example, described learning as simply a change in long-term memory schemas (Kirschner et al. 2006). Consequently, the practical experimentation only serves as a facilitator for enhancing skill acquisition and not as a mandatory process (Anderson 1982; Kirschner and Huisman 1998). However, most sporting athletes will tell

you that there is something intuitive about the sense-perception of physically doing something that cannot be taught by knowledge alone (Moran 2012).

These arguments suggest that representing psychomotor skills in an ITS domain model is recommended to support learning. However, unlike cognition which can be generalised to a large group of individuals, psychomotor learning is dependent largely on the individual and the domain, requiring a degree of specialisation. This lack of standardisation would suggest that creating a universal ITS is not appropriate. Given this then, a proposed architecture that follows a modular design would seem more suited.

4 Proposed Architecture

We propose the Distributed Augmented Reality Training (DART) architecture: a distributed agent-based architecture that combines centralisation and decentralisation characteristics. A dispatcher agent allows integration of multiple standalone ITSs that can be combined together in a distributed system. The dispatcher performs either centralised or decentralised modelling using the integrated standalone tutors. The architecture embodies the post-cognitivist view of education by distributing knowledge sources across different agents in the entire system.

4.1 Motivations

The following key motivations drive the design of the novel DART architecture:

- An existing ITS needs to be used but the architecture is not easily modifiable
- The back end ITS needs to be swapped out
- Sensing logic, the UI and internal ITS logic needs to be separated
- Designing and implementing ITSs requires specialised skill. It should be possible to use these existing ITSs with AR.

4.2 Description

The DART architecture integrates three components together (Fig. 1): (1) Environment, which incorporates agents that communicate with the environment; (2) Learning, which incorporates agents that communicate with external ITSs and which perform tutor-related functions like learner modelling but excluding spatial ontologies; and finally, (3) Core, which interconnects all of the other components. Other components connect directly to the Core which coordinates their functions.

Fig. 1 A Conceptual Diagram Showing the Components of the dart Architecture. Every component interacts with an dart Core instance, which in turn, communicates with other components.

4.2.1 Environment

The Environment component consists of agents which interfaces with the environment (such as a flight simulator, a virtual 3D world (Ludwig et al. 2010), a spatial map of the surrounding environment (LaViola et al. 2015), the physical environment or a combination thereof). Agents in the Environment component may interface with machinery by implementing logic and interfaces needed for communication. Information collected by Environment agents is passed to the backbone components. There are two types of agents deployed into the Environment component:

User Interface (UI). The UI module is the user interface application such as a mobile AR application integrated into DART; a flight simulator agent, a Microsoft Excel program or hardware machinery that receives sensor data from the real world.

Sensor Sensor agents parse messages from the UI agent and uses that information to inform the other agents of key events and for building of the spatial model. The Sensor combines the functionality of the gateway and AR tracker agents. It is also responsible for interfacing with other physical hardware such as a machine operated by the learner. Typically, events from a UI are passed to the Sensor module, which in turn, passes them to the Dispatcher modules.

4.2.2 Learning

The Learning Component consists of the dispatcher agent and one or more external standalone ITSs systems or modelling engines that perform learner and domain knowledge modelling.

Dispatcher Agents. The Dispatcher agents coordinate communication between the different modelling agents and, in the event, of a conflict determines which one has priority. It also raises assessments for events, manipulates training logic and creates a unified learner model that can be shared with agents otherwise incompatible with the external standalone ITSs. The Dispatcher agent sends and receives information from the external ITS using an interface format compatible with the chosen external ITS.

External Intelligent Tutoring System. An external standalone ITS or modelling engine is a system, or series of systems that are integrated with the system. The Dispatcher agent implements an interface for communicating with the Core agent and the external systems. The external ITS does nothing with the other agents directly. Rather, its requests are interpreted by the Dispatcher agent in the manner that it deems pedagogically appropriate. It could even ignore requests from external ITSs. The Dispatcher agent can enhance an ITS by implementing functionality missing from the external ITS, without needing to reproduce the entire domain and learner models. For example, the Dispatcher agent may incorporate one or more internal modelling engines that it uses in lieu of an external system.

4.2.3 Back Bone

The back bone consists of one or more Core agents that handles the intercommunication between three types of agents: (1) Dispatcher agents; (2) Sensor agents and (3) Core agents. These three types of agents share a common communication language to enable intercommunication between them. These three agents together form the back bone. The back bone intercommunicates with external systems using two types of agents: (a) Dispatcher agents and (b) Sensor agents. Dispatcher agents communicate with other external tutoring agents and Sensor agents communicate with UI agents and hardware systems. Each Backbone agent is described below:

Core. The central module of the architecture. All other backbone modules are tied together by communicating via the Core module. It performs remote procedural calls, sends messages and arguments to requested modules.

Dispatcher. Dispatcher agents send data to external ITSs, modeller engines or internal modelling engines to model cognitive and similar learning dimensions. The Dispatcher may build its learner or domain model from information it gets from external ITSs, creating a unified domain and learner model for use by all the agents in the distributed system.

Sensor. Sensor agents interconnect the backbone to the virtual (or real) environment. Typically, the Sensor agents listen for information from the real-world, or from an external computer program (such as a mouse click). It then passes this information to the UI agent or the Dispatcher agents.

5 Intelligent Software Training Platform

We designed and implemented the Intelligent Software Training Platform (ISTP) using our DART architecture. Our system was developed in C# using NET framework 4.7. Remote Procedural Calls (RPCs) was implemented using JavaScript Object Notation (JSON) NET Library (JSON.net n.d) and C# reflection. The models created by each of the agents was encoded in an Structured Markup Language (XML) data structure to support interoperability between the agents.

5.1 Network Communication

The networking stack is a useful mechanism for passing information between agents residing on the same or different computers. The Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) protocols can work over various technologies and across different configurations, making it an ideal choice.

5.1.1 Agent Discovery Protocol

The Core, Sensor and Dispatcher agents communicate with each other using the TCP protocol. However, TCP requires for each agent to be assigned an Internet Protocol (IP) address and port, which can be cumbersome to configure for each environment. Therefore, we implemented the DART discovery protocol to automate this step. It uses the connectionless UDP protocol to send a broadcast request to potential agents on the network. These agents acknowledge that they are an agent by sending back a JSON-formatted string containing the Message Digest 5 (MD5) hash matching the agent that initiated the discovery request and a list of IP addresses it is listening on. The agent performing the discovery uses the IP address information to connect to the

agent. All agents are interconnected to each other by using the discovery protocol to locate a suitable Core agent. Since all modules connect to the Core, there is no need for the other modules to connect to each other.

5.1.2 Redundancy

One or more Backbone agents of the same type may be integrated, but only one same-typed agent plays an active role. The other same-typed agents play a passive role and only become active when an already-active module fails. The architecture is designed to provide redundancy. We incorporated redundancy on the Core agents only, so that another Core agent would take over if the main Core agent failed or stopped responding for some reason.

5.2 Dispatcher Agents

Many existing ITSs, including GIFT, are integrable with external applications using existing features in those systems. In our implementation, GIFT connects with our Dispatcher agent using an Extensible Mark-up Language over Remote Procedural Call (XML-RPC) connection (LaViola et al. 2015). The Dispatcher agent receives feedback strings from GIFT over this connection and forwards them to the other agents. By formatting the GIFT feedback messages in a certain way (such as prepending them with a dollar sign) represents a command that tells the Dispatcher and/or its internal modelling engines to do something else, other than forward them to the training applications.

5.2.1 Intelligent Tutoring System Document Model

The Intelligent Tutoring System Document Model (ITSDM) is an XML data structure that defines the semantic structure of the Dispatcher. The structure and content of the various learner/domain models and spatial ontologies are represented within this XML data structure.

This data structure can be shared between external training applications, such as the mobile AR training application or with external ITSs. This allows for client-side parsing of the ITS model by these applications without having to construct the model of its own.

5.2.2 Election Mechanism

When the Dispatcher processes a condition in the UI agent, it determines which external ITSs is going to handle the request, either: GIFT or our internal modelling engine. Our Dispatcher agent uses two mechanisms to make this decision: (a) Explicit election or (b) ITS cost.

1. *Explicit Election.* The Dispatcher will make a determination as to which external standalone ITS should be used to handle a learner event (e.g. a

mouse click). Using explicit rules authored in the ITSDM course file, the Dispatcher agent elects the most suitable external system for an event (e.g. mouse clicks result in message being sent to GIFT). If there are no explicit rules matching the criteria, our Dispatcher agent resorts to using ITS cost.

2. *ITS Cost.* The Dispatcher selects the ITS with the lowest cost. In our implementation, this is the GIFT framework followed by our internal engine. If it cannot infer the ITS by cost, no action is carried out and the learner would receive no feedback. This is desirable in cases where you do not want any ITS monitoring the learner e.g. studies comparing ITS to non-ITS conditions. The selected ITS may send back an unhandled message. If so, the Dispatcher agent would then try the ITS with the second lowest cost until eventually, one of the listed ITSs responds. If the list is exhausted, then nothing happens.

6 Limitations

Our proposed architecture has several technical, design and research limitations. We also discuss some ways that we addressed or mitigated some of these limitations.

6.1 Design Limitations

Conflicting Agents. Agents or ITSs may conflict with one another in terms of what learning support they should offer. One solution involves the use of semantic mapping to convert data from one format to another (Brusilovsky et al. 2009) and creating a unified learner model. However, many ITSs lack this functionality.

We addressed this by using a single Dispatcher agent that interprets the data from each of the external ITSs and uses an election system to determine which ITS (or agent) should be used.

6.2 Technical Limitations

Dispatcher Communication. Our architecture assumed that the external ITS was capable of communicating with the Dispatcher agent instance. However, this may not be the case. Even if networking integration was possible, the Dispatcher agent would need to support the networking protocols used and the interface to encode/decode the messages from the external ITS.

Moreover, the ITS needs to be able to represent enough information to the Dispatcher agent. In some cases, significant development of the Dispatcher agent interface mechanism would be required.

Despite these limitations, we argue that developing an interface and communication protocol for the Dispatcher agent is much easier and quicker than building the entire tutoring framework from scratch.

6.3 Research Limitations

Evaluation. Evaluating ITS architectures differs from evaluating the learning effectiveness of a system integrating the architecture (Brusilovsky et al. 2005). One of the challenges for evaluating architectural effectiveness is that an architecture is an abstraction of interconnected components (Bosch and Molin 1999). The validity of DART has not been tested empirically.

6.4 Lessons Learned

Implementing our DART architecture revealed some important lessons that were not apparent through design alone such as:

- *Agent Integration.* Unity applications, such as AR tracking applications, require the UI agent to be embedded within the Unity application for it to communicate with the Sensor and Core agents. In our implementation, the agent was embedded as a separate threaded application, but this resulted in inconsistent performance. Communicating between the threaded agent and Unity was a considerable challenge. Unity does not native support our networking protocol, such as the DART discovery protocol. So we embedded it as a separate, threaded agent which sometimes caused inconsistent behaviour due to race conditions.

Vuforia Integration. In our case, the tracker was not decoupled from the Unity application, requiring duplication of tracker logic across tutors. Since the development, newer versions of Unity addressed this by integrating Vuforia into the Unity engine itself.

7 Conclusions and Future Work

Combining ITSs with AR training systems potentially enhances training and should ideally be accomplished with minimal architectural changes to the ITS components. Fortunately, some useful architectures facilitate this design aim such as: centralised architectures, distributed agent-based architectures, centralised agent-based architectures and hybrid architectures. However, these existing architectures tend to be overly generalised to apply specifically to AR training systems and do not embody a theory of distributed cognition as their design basis. So, we proposed the DART architecture to specifically address these two limitations. It is a distributed agent-based architecture that separates Sensor and UI and modelling functionality into separate distributed

agents to support interfacing with various external ITSs such as GIFT and custom built ITSs. This would help overcome limitations with existing AR-based ITS architectures such as the MAT architectures which does not decouple the tracking and UI components, requiring the domain model to be used to represent spatial ontologies.

Our architecture is only the first step in trying to reduce the technological and theoretical barriers but we do not propose that our architecture completely overcomes these barriers. For example, the interface between the external ITS and the modelling agents would need to be created and there are many successful ITSs that cannot be modified to suit this requirement (Westerfield 2012). Some ITSs are decades old and cannot be easily re-implemented or modified to run on modern hardware. Interpreting messages from the ITS is also a challenge and not all ITSs support this requirement. Nevertheless, if it can be shown, that it is a step in the right direction, the architecture could be further refined to further reduce these barriers.

More work is needed to further validate the proposed architecture. One potential way that this could be achieved is to evaluate the perceptions of our architecture, or by measuring the time taken to implement tutors using DART as opposed to traditional approaches. While these approaches would show that development is subjectively easier using DART, it would not necessarily show that the architecture itself is effective. Future work could explore potential ways of evaluating novel ITS architectures.

References

- Almiyad MA, Oakden-Rayner L, Weerasinghe A, Billingham M (2017) Intelligent Augmented Reality Tutoring for Physical Tasks with Medical Professionals. In: Artificial Intelligence in Education, Springer, Cham, Lecture Notes in Computer Science, pp 450–454, DOI 10.1007/978-3-319-61425-0_38
- Almurshidi SH, Abu-Naser SS (2016) Design and development of diabetes intelligent tutoring system. *European Academic Research* IV(9):8117–8128
- Anderson JR (1976) Preliminary Considerations. In: *Language, Memory, and Thought*, Psychology Press
- Anderson JR (1982) Acquisition of cognitive skill. *Psychological Review* 89(4):369–406, DOI 10.1037/0033-295X.89.4.369
- Anderson JR (1993) *Rules of the Mind*. Lawrence Erlbaum Associates, Inc, 365 Broadway, Hillsdale, NJ 07642
- Anderson JR (1996) ACT: A simple theory of complex cognition. *American Psychologist* 51(4):355–365, DOI 10.1037/0003-066X.51.4.355
- Anderson JR, Conrad FG, Corbett AT (1989) Skill acquisition and the LISP tutor. *Cognitive Science* 13(4):467–505, DOI 10.1016/0364-0213(89)90021-9
- Anderson JR, Corbett AT, Koedinger KR, Pelletier R (1995) Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences* 4(2):167–207, DOI 10.1207/s15327809jls0402_2

- Anderson JR, Matessa M, Lebiere C (1997) ACT-R: A Theory of Higher Level Cognition and Its Relation to Visual Attention. *Human-Computer Interaction* 12(4):439–462, DOI 10.1207/s15327051hci1204_5
- Azuma RT (1997) A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments* 6(4):355–385
- Baker RS (2016) Stupid Tutoring Systems, Intelligent Humans. *International Journal of Artificial Intelligence in Education* 26(2):600–614, DOI 10.1007/s40593-016-0105-0
- Beck J, Stern M, Haugsjaa E (1996) Applications of AI in education. *XRDS: Crossroads, The ACM Magazine for Students* 3(1):11–15, DOI 10.1145/332148.332153
- Billinghurst M (2002) Augmented reality in education. *New horizons for learning* 12
- Billinghurst M, Duenser A (2012) Augmented Reality in the Classroom. *Computer* 45(7):56–63, DOI 10.1109/MC.2012.111
- Billinghurst M, Clark A, Lee G (2015) A Survey of Augmented Reality. *Found Trends Hum-Comput Interact* 8(2-3):73–272, DOI 10.1561/11000000049
- Bosch J, Molin P (1999) Software architecture design: Evaluation and transformation. In: *Proceedings ECBS'99. IEEE Conference and Workshop on Engineering of Computer-Based Systems*, pp 4–10, DOI 10.1109/ECBS.1999.755855
- Brusilovsky P (2004) KnowledgeTree: A distributed architecture for adaptive e-learning. In: *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*, Association for Computing Machinery, New York, NY, USA, WWW Alt. '04, pp 104–113, DOI 10.1145/1013367.1013386
- Brusilovsky P, Sosnovsky S, Shcherbinina O (2005) User Modeling in a Distributed E-Learning Architecture. In: *Ardissono L, Brna P, Mitrovic A (eds) User Modeling 2005*, Springer, Berlin, Heidelberg, *Lecture Notes in Computer Science*, pp 387–391, DOI 10.1007/11527886_50
- Brusilovsky P, Mitrovic A, Sosnovsky S, Mathews M, Yudelsohn M, Lee DH, Zadorozhny V (2009) Database Exploratorium: A Semantically Integrated Adaptive Educational System. In: *Proceedings of the 7th International Workshop on Ubiquitous User Modelling*, Trento, Italy, pp 24–32
- Cheikes BA (1995) GIA: An agent-based architecture for Intelligent Tutoring Systems. In: *Proceedings of the CIKM'95 Workshop on Intelligent Information Agents*, Citeseer
- Corbett A, Anderson J, Graesser A, Koedinger K, VanLehn K (1999) Third Generation Computer Tutors: Learn from or Ignore Human Tutors? In: *CHI '99 Extended Abstracts on Human Factors in Computing Systems*, ACM, New York, NY, USA, *CHI EA '99*, pp 85–86, DOI 10.1145/632716.632769
- Dalim CSC, Dey A, Piumsomboon T, Billinghurst M, Sunar S (2016) TeachAR: An Interactive Augmented Reality Tool for Teaching Basic English to Non-Native Children. In: *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pp 82–86, DOI 10.1109/ISMAR-Adjunct.2016.0046

- De Jong CJ (2016) Embedding an Intelligent Tutor into existing Business Software to provide On-the-Job Training. Masters, University of Canterbury, Christchurch, New Zealand
- D'mello S, Graesser A (2013) AutoTutor and Affective Autotutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers That Talk Back. *ACM Trans Interact Intell Syst* 2(4):23:1–23:39, DOI 10.1145/2395123.2395128
- Ferris T (2010) Bloom's taxonomy of educational objectives: A psychomotor skills extension for engineering and science education. *International Journal of Engineering Education* 26(3):699–707
- Gilbert A (2016) Augmented Reality for the US Air Force. In: *Virtual, Augmented and Mixed Reality*, Springer, Cham, pp 375–385, DOI 10.1007/978-3-319-39907-2_36
- Goldberg B, Cannon-Bowers J (2013) Experimentation with the generalized intelligent framework for tutoring (GIFT): A testbed use case. In: *AIED 2013 Workshops Proceedings Volume*, vol 7, pp 27–36
- Hampton AJ, Hu X, Graesser AC, Cai Z, Tackett AC (2018) Potential to Migrate ElectronixTutor to GIFT. In: *Proceedings of the Sixth Annual GIFT Users Symposium*, US Army Research Laboratory, vol 6, p 27
- Henderson SJ, Feiner S (2009) Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In: *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pp 135–144, DOI 10.1109/ISMAR.2009.5336486
- Herbert B, Ens B, Billinghamurst M (2017a) An Adaptive AR Tutor For Cabling a Network Topology. In: *Eurographics Symposium on Virtual Environments (2017)*, The Eurographics Association, Adelaide, Australia, DOI 10.2312/egve.20171369
- Herbert B, Ens B, Weerasinghe A, Billinghamurst M, Wigley G (2018) Design considerations for combining augmented reality with intelligent tutors. *Computers & Graphics* 77:166–182
- Herbert BM, Weerasinghe A, Ens B, Billinghamurst M, Wigley G (2017b) An AR Network Cabling Tutoring System for Wiring a Rack. *The Eurographics Association*, DOI 10.2312/egve.20171381
- Hilgard ER (1948) *Theories of Learning*. New York : Appleton-Century Crofts
- Hunt E (1999) What is a theory of thought? In: Sternberg RJ (ed) *The Nature of Cognition*, MIT Press
- Iqbal MZ, Mangina E, Campbell AG (2019) Exploring the use of Augmented Reality in a Kinesthetic Learning Application Integrated with an Intelligent Virtual Embodied Agent. In: *IEEE International Symposium on Mixed and Augmented Reality Adjunct*, IEEE Computer Society, Beijing, China, pp 12–16, DOI 10.1109/ISMAR-Adjunct.2019.00018
- JSONnet (n.d) Introduction. <https://www.newtonsoft.com/json/help/html/Introduction.htm>
- Kim N, Nam CS (2020) Adaptive Control of Thought-Rational (ACT-R): Applying a Cognitive Architecture to Neuroergonomics. In: Nam CS (ed) *Neuroergonomics: Principles and Practice*, Cognitive Science and Technology, Springer International Publishing, Cham, pp 105–114, DOI 10.1007/978-3-

- 030-34784-0-6
- Kirschner P, Huisman W (1998) 'Dry laboratories' in science education; computer-based practical work. *International Journal of Science Education* 20(6):665–682, DOI 10.1080/0950069980200605
- Kirschner P, Sweller J, Clark R (2006) Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist* 41(2):75–86, DOI 10.1207/s15326985ep4102_1
- Kodaganallur V, Weitz RR, Rosenthal D (2005) A Comparison of Model-Tracing and Constraint-Based Intelligent Tutoring Paradigms. *International Journal of Artificial Intelligence in Education* (IOS Press) 15(2):117–144
- Latham AM, Crockett KA, McLean DA, Edmonds B, O'Shea K (2010) Oscar: An intelligent conversational agent tutor to estimate learning styles. In: *International Conference on Fuzzy Systems*, pp 1–8, DOI 10.1109/FUZZY.2010.5584064
- LaViola J, Williamson B, Brooks C, Veazanchin S, Sottolare R, Garrity P (2015) Using augmented reality to tutor military tasks in the wild. In: *Inter-service/Industry Training Simulation and Education Conference*, Orlando, FL, USA, vol 15050, pp 1–10
- Ludwig J, Sincoff E, Remolina E, Stottler R, Nguyen A (2010) Integrating an Intelligent Tutoring System for TAOs with Second Life. In: *Training Centric... Readiness Focused*, Naval Undersea Warfare Center, Newport ,RI, 02841, vol Vol. 2010, p 11
- Major N, Ainsworth S, Wood D (1997) REDEEM: Exploiting symbiosis between psychology and authoring environments. *International Journal of Artificial Intelligence in Education* pp 317–340
- Mathews MM (2012) A Framework for Multiple Adaptable Pedagogical Strategies in Intelligent Tutoring Systems. PhD, University of Canterbury, Christchurch, New Zealand
- Milgram P, Takemura H, Utsumi A, Kishino F (1995) Augmented reality: A class of displays on the reality-virtuality continuum. In: Das H (ed) *Proceedings of SPIE, The International Society for Optical Engineering*, vol 2351, pp 282–292, DOI 10.1117/12.197321
- Mitrovic A, Ohlsson S (1999) Evaluation of a Constraint-Based Tutor for a Database Language. *International Journal of Artificial Intelligence in Education* 10(6):238–256
- Mitrovic A, Koedinger K, Martin B (2003) A comparative analysis of cognitive tutoring and constraint-based modeling. *User Modeling 2003* pp 147–147
- Moran A (2012) Thinking in action: Some insights from cognitive sport psychology. *Thinking Skills and Creativity* 7(2):85–92, DOI 10.1016/j.tsc.2012.03.005
- Murray T (1996) Having it all, maybe: Design tradeoffs in ITS authoring tools. In: *Intelligent Tutoring Systems*, Springer, Berlin, Heidelberg, *Lecture Notes in Computer Science*, pp 93–101, DOI 10.1007/3-540-61327-7_105
- Murray T (1999) Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*

- (IJAIED) 10:98–129
- Nesbit JC, Adesope OO, Liu Q, Ma W (2014) How Effective are Intelligent Tutoring Systems in Computer Science Education? In: 2014 IEEE 14th International Conference on Advanced Learning Technologies, pp 99–103, DOI 10.1109/ICALT.2014.38
- Nwana HS (1990) Intelligent tutoring systems: An overview. *Artificial Intelligence Review* 4(4):251–277
- Ohlsson S (1994) Constraint-Based Student Modeling. In: Greer JE, McCalla GI (eds) *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, Springer Berlin Heidelberg, NATO ASI Series, pp 167–189
- Ohlsson S (1996) Learning from error and the design of task environments. *International Journal of Educational Research* 25(5):419–448, DOI 10.1016/S0883-0355(97)81236-0
- Perkins DN (1993) Person-plus: A distributed view of thinking and learning. In: *Distributed cognitions: Psychological and educational considerations*, Cambridge University Press, pp 88–110
- Philip PJ, Brawner K, Olney A, Mitrovic A (2013) A Review of Student Models Used in Intelligent Tutoring Systems. In: *Design Recommendations for Intelligent Tutoring Systems: Volume 1-Learner Modeling*, vol 1, US Army Research Laboratory, Orlando, FL, USA, pp 39–67
- Ritter S (2016) Plug-in Tutor Agents: Still Pluggin'. *International Journal of Artificial Intelligence in Education* 26(1):405–415, DOI 10.1007/s40593-015-0062-z
- Ritter S, Koedinger KR (1996) An Architecture for Plug-In Tutor Agents. *Journal of Artificial Intelligence in Education; Charlottesville* 7(3):315–347
- Roll I, Aleven V, Koedinger KR (2010) The Invention Lab: Using a Hybrid of Model Tracing and Constraint-Based Modeling to Offer Intelligent Support in Inquiry Environments. In: *Intelligent Tutoring Systems*, Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, pp 115–124, DOI 10.1007/978-3-642-13388-6_16
- Santos MEC, Chen A, Taketomi T, Yamamoto G, Miyazaki J, Kato H (2014) Augmented Reality Learning Experiences: Survey of Prototype Design and Evaluation. *IEEE Transactions on Learning Technologies* 7(1):38–56, DOI 10.1109/TLT.2013.37
- Self J (1990) Theoretical Foundations for Intelligent Tutoring Systems. *Journal of Artificial Intelligence in Education* 4:3–14
- Simpson EJ (1966) The Classification of Educational Objectives (in the) Psychomotor Domain. Tech. Rep. BR-5-0090, University of Illinois, Urbana, Urbana, Illinois
- Skinner BF (1950) Are theories of learning necessary? *Psychological Review* 57(4):193–216, DOI 10.1037/h0054367
- Sottolare R, Brawner K (2018) Component interaction within the Generalized Intelligent Framework for Tutoring (GIFT) as a model for adaptive instructional system standards. In: *Proceedings of the 14th International Conference on Intelligent Tutoring Systems 2018 Workshops, CEUR Workshop Proceedings*, Montreal, QC, Canada, vol 2121, pp 55–62

- Sottolare R, LaViola J (2015) Extending intelligent tutoring beyond the desktop to the psychomotor domain: A Survey of smart glass technologies. In: Proceedings of the 2015 Inter-Service/Industry Training, Simulation and Education Conference, National Training and Simulation Association, Arlington, VA, pp 1–11
- Sottolare R, Hackett M, Pike W, LaViola J (2017) Adaptive instruction for medical training in the psychomotor domain. *The Journal of Defense Modeling and Simulation* 14(4):331–343, DOI 10.1177/1548512916668680
- Sottolare RA, Holden HK (2013) Motivations for a generalized intelligent framework for tutoring (GIFT) for authoring, instruction and analysis. In: AIED 2013 Workshops Proceedings, vol 7, p 1
- Sottolare RA, Goldberg BS, Brawner KW, Holden HK (2012) A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems (CBTS). In: Proceedings of the Interservice/Industry Training, Simulation, and Education Conference, pp 1–13
- Sweller J (2003) Evolution of human cognitive architecture. In: *Psychology of Learning and Motivation*, vol 43, Academic Press, pp 215–266, DOI 10.1016/S0079-7421(03)01015-6, URL <http://www.sciencedirect.com/science/article/pii/S0079742103010156>
- Tang A, Owen C, Biocca F, Mou W (2003) Comparative Effectiveness of Augmented Reality in Object Assembly. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, Fort Lauderdale, Florida, USA, CHI '03, pp 73–80, DOI 10.1145/642611.642626
- Vargas González AN, Kapalo K, Koh SL, LaViola JJ (2017) Exploring the Virtuality Continuum for Complex Rule-Set Education in the Context of Soccer Rule Comprehension. *Multimodal Technologies and Interaction* 1(4):30, DOI 10.3390/mti1040030
- Vassileva J, McCalla G, Greer J (2003) Multi-Agent Multi-User Modeling in I-Help. *User Modeling and User-Adapted Interaction* 13(1):179–210, DOI 10.1023/A:1024072706526
- Watson JB (1913) Psychology as the behaviorist views it. *Psychological Review* 20(2):158–177, DOI 10.1037/h0074428
- Werrlich S, Eichstetter E, Nitsche K, Notni G (2017) An Overview of Evaluations Using Augmented Reality for Assembly Training Tasks. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering* 11(10):1054–1060
- Westerfield G (2012) *Intelligent Augmented Reality Training for Assembly and Maintenance*. Masters, University of Canterbury, New Zealand
- Westerfield G, Mitrovic A, Billingham M (2015) Intelligent Augmented Reality Training for Motherboard Assembly. *International Journal of Artificial Intelligence in Education* 25(1):157–172, DOI 10.1007/s40593-014-0032-x
- Xu L, Clarke D (2012) What Does Distributed Cognition Tell Us about Student Learning of Science? *Research in Science Education* 42(3):491–510, DOI 10.1007/s11165-011-9207-8

Yin X, Gu Y, Qiu S, Fan X (2014) VR&AR Combined Manual Operation Instruction System on Industry Products: A Case Study. In: 2014 International Conference on Virtual Reality and Visualization, pp 65–72, DOI 10.1109/ICVRV.2014.55