Chapter 8
Towards Truly Autonomous Synthetic Characters with the Sigma Cognitive Architecture

Volkan Ustun
University of Southern California, USA

Paul S. Rosenbloom
University of Southern California, USA

ABSTRACT

Realism is required not only for how synthetic characters look but also for how they behave. Many applications, such as simulations, virtual worlds, and video games, require computational models of intelligence that generate realistic and credible behavior for the participating synthetic characters. Sigma (Σ) is being built as a computational model of general intelligence with a long-term goal of understanding and replicating the architecture of the mind; i.e., the fixed structure underlying intelligent behavior. Sigma leverages probabilistic graphical models towards a uniform grand unification of not only traditional cognitive capabilities but also key non-cognitive aspects, creating unique opportunities for the construction of new kinds of non-modular behavioral models. These ambitions strive for the complete control of synthetic characters that behave as humanly as possible. In this paper, Sigma is introduced along with two disparate proof-of-concept virtual humans – one conversational and the other a pair of ambulatory agents – that demonstrate its diverse capabilities.

DOI: 10.4018/978-1-5225-0454-2.ch008

1. INTRODUCTION

Twenty years ago, Tambe et al. (1995) discussed the generation of human-like synthetic characters that can interact with each other, as well as with humans, within the emerging domain of highly interactive simulations. Many of these simulations strove to create environments that looked realistic, and synthetic characters that looked and behaved as real people to the extent possible. The behavioral models in these simulations extensively utilized *cognitive architectures* (Langley, Laird, & Rogers, 2009) – models of the fixed structure underlying intelligent behavior in natural and/or artificial systems – as the underlying driver for human-like intelligent behavior. Twenty years later, developments in computer graphics and animation have allowed for extremely realistic-looking interactive simulation environments; it is now possible to create almost photo-real synthetic characters with realistic gaits and gestures. However, progress in behavior generation has been more mixed. Mainstream cognitive architectures, including Soar and ACT-R, originated as production systems and are fairly capable of modeling the reactive, knowledge-intensive, and goal-driven aspects of human behavior. For example, Tambe et al.'s (1995) work in the air-combat simulation domain utilized Soar (Laird, 2012) to model the behavior of pilots. These cognitive architectures are also capable of working in real time and, in ACT-R's case, with explicit models of human reaction times and limitations. However, they have not yet been able to successfully incorporate all the capabilities that are required for human-like intelligence.

As Swartout (2010) has pointed out, behaving like real people requires synthetic characters to, among other things: (1) use their perceptual capabilities to observe their environment and other virtual/real humans in it; (2) act autonomously in their environment based on what they know and perceive, e.g. reacting and appropriately responding to the events around them; (3) interact in a natural way with both real and other virtual humans using verbal and nonverbal communication; (4) possess a *Theory of Mind* (*ToM*) to model their own mind and the minds of others; (5) understand and exhibit appropriate emotions and associated behaviors; and (6) adapt their behavior through experience. The Soar and ACT-R communities worked toward addressing these six capabilities for synthetic characters (referred to as the *capability list* hereafter) but some items were simply not feasible within the core architecture. For example, external modules were required for acceptable perceptual and communication capabilities. Likewise, most of the emotion models were also outside the core. More importantly, they haven't been able to fully capture the advances that have been made in recent years in behavioral adaptation, or in other words, learning. A number of aspects of learning were successfully incorporated but generality in statistical machine learning, for example, has eluded them.

Probabilistic graphical models (Koller & Friedman, 2009) provide a general tool that combines graph theory and probability theory to enable efficient probabilistic reasoning and learning in ways that haven't been possible with traditional cognitive architectures based on production systems. The machine learning community employs such models as one of its primary tools, yielding state-of-the-art results for at least four of the listed capabilities that have challenged traditional cognitive architectures: perception, autonomy, interaction and adaptation. However, most of these improvements have been achieved independently, as examples of narrow Artificial Intelligent (AI) systems, with little effort toward cross-integration.

One of the main reasons for the relative lack of integration efforts is the inherent difficulty of general intelligence research: not only is it challenging to integrate all of the requisite capabilities, but it is also hard to measure the resulting incremental progress toward human-like intelligence (Goertzel, 2014). In contrast, many forms of narrow AI systems can easily track incremental progress as they try to improve "intelligent" behaviors in very specific contexts. Therefore, it is easier to assess the merit of these systems through simple comparisons. This strategy has helped narrow AI approaches dominate the research in the last two decades.

The objectives for generally intelligent systems, however, are quite different from narrow AI systems. Generally intelligent systems are not necessarily focused on the extremes of performance in limited contexts but rather on the ability to achieve a wide range of goals and to carry out a variety of tasks in many different contexts and environments, as stated by Goertzel (2014). Such systems require the traits of the capability list to be integrated together and work coherently. As pointed out by Swartout (2010), this integration can be quite difficult, but it can potentially yield more than the sum of its parts. Mainstream cognitive architectures, as natural candidates for the desired integration, have attempted to add some of these capabilities without actually updating their cores (Ji, Gray, Guhe, & Schoelles, 2004). This approach has improved their overall functionality but still has not been enough to capture many critical developments, especially in perception and learning.

This paper introduces a nascent cognitive architecture/system, Sigma (Rosenbloom, 2013), that conflates traditional cognitive architectures and probabilistic graphical models. The long-term goal is to understand and replicate the fixed structure underlying intelligent behavior in *both* natural and artificial systems. Sigma is motivated by the original grand goal of AI, as well as by the more recent reformulation of this goal within artificial general intelligence (AGI) (Goertzel, 2014), in an attempt to create a full cognitive system that can implement the qualities in the capability list and more. As such, it is intended ultimately to support the real-time needs of truly autonomous characters, whether robots or synthetic characters (virtual humans).

Sigma's development is guided by a quartet of desiderata: (1) *grand unification*, uniting the requisite cognitive and non-cognitive aspects of embodied intelligence for intelligent behavior in complex real worlds; (2) *generic cognition*, spanning both natural and artificial cognition at a suitable level of abstraction; (3) *functional elegance*, yielding broad cognitive (and sub-cognitive) functionality from interactions among a small general set of mechanisms; and (4) *sufficient efficiency*, executing rapidly enough for anticipated applications. The first, second and last desiderata are directly relevant to the construction of broadly capable, real-time virtual humans (VHs), while the third implies a rather unique path towards them, where instead of a disparate assembly of modules, all of the required capabilities are constructed and integrated together on a simple elegant base.

Most of the work to date on Sigma has explored various capabilities individually: learning (Rosenbloom, 2012, 2014; Rosenbloom, Demski, Han, & Ustun, 2013), memory and knowledge (Rosenbloom, 2010, 2014; Ustun, Rosenbloom, Sagae, & Demski, 2014), decision making and problem solving (Chen et al., 2011; Rosenbloom, 2011b), perception (Chen et al., 2011), speech (Joshi, Rosenbloom, & Ustun 2014), ToM (Pynadath, Rosenbloom, Marsella, & Li, 2013; Pynadath, Rosenbloom, & Marsella, 2014), and emotions (Rosenbloom, Gratch, & Ustun 2015). But, even more importantly, *Sigma's* non-modular, *hybrid* (discrete + continuous) *mixed* (symbolic + probabilistic) character also supports attempts at a deep form of integration across the capability list, straddling the traditional boundary between symbolic cognitive processing and numeric sub-cognitive processing. Sigma provides the ability to exhibit this combination of capabilities in a unified manner because of its grounding in a *graphical architecture* that is built from graphical models (Koller & Friedman 2009) (in particular, factor graphs and the summary product algorithm (Kschischang, Frey, & Loeliger 2001)), n-dimensional piecewise linear functions (Rosenbloom, 2011a), and gradient descent learning (Rosenbloom, Demski, Han, & Ustun, 2013). The required VH capabilities emerge in a functionally elegant manner from both interactions among this small but general set of mechanisms and knowledge captured in Sigma models. For example, reinforcement learning (RL) arises from the interactions between gradient descent learning and particular forms of both domain-specific and domain-independent knowledge (Rosenbloom, 2012). Truly autonomous characters would require a computational intelligent behavior model that pushes the boundaries of how the broad range of traits in the capability list may be integrated, with Sigma providing a unique, and potentially powerful, way of doing this.

Still, this is just the beginning of a long journey. The real power of the unique path taken by Sigma is in the emerging synergy between diverse capabilities. Yet the models discussed in this paper are just the first attempts toward developing integrative virtual humans. There is definitely a need to devise more comprehensive virtual humans with more interactions among characters in realistic scenarios and trained across multiple environments in order to assess Sigma's true potential.

Nonetheless, Sigma is maturing day by day, and the diverse progress that has been made so far provides initial evidence that Sigma will be capable of creating truly autonomous VHs as a capstone of its maturation.

There have been a few other recent proposals for the utilization of graphical models in cognitive architectures (Cassimatis, 2010; Danks, 2014; Lee-Urban et al., 2015), but they are mostly just proposals with quite limited implementations. Neural networks also provide a major alternative to graphical models for addressing a number of the items in the capability list that most challenge traditional cognitive architectures, and with the recent focus on deep models provide the current state of the art in statistical machine learning. General intelligence is emerging as a popular research topic in the deep learning community (Mikolov, Joulin, & Baroni 2015; Schmidhuber, 2015; Strannegard, von Haugwitz, Wessberg, & Balkenius 2013), but it remains to be seen what the research in this field will yield. In general, none of the current approaches is yet adequate for comprehensively encompassing the capability list.

In this paper, first a case is made for needing truly autonomous characters that embody human-like intelligence, rather than just characters who may be perceived as intelligent, in interactive simulation environments. Then, Sigma will be introduced and its potential for creating truly autonomous synthetic characters will be discussed along with two disparate proof-of-concept virtual humans – one conversational (Ustun, Rosenbloom, Kim, & Li 2015) and the other a pair of ambulatory agents (Ustun & Rosenbloom, 2015) – that demonstrate its diverse capabilities. Both varieties of virtual human combine cognitive and non-cognitive capabilities, and in the case of the ambulatory agents integrate together diverse forms of learning.

2. TRULY AUTONOMOUS CHARACTERS

Traditional cognitive architectures have controlled virtual characters capably in a number of interactive simulation environments (Tambe et al., 1995; Traum, Marsella, Gratch, Lee, & Hartholt 2008; Zemla et al., 2011). One of the most relevant examples is the use of the Soar cognitive architecture to create a synthetic character with multiple goals and extensive tactics in the computer game Quake II (Laird and van Lent 2001). Laird and van Lent (2001) states: “While the Soar Quakebot explores a level, it creates an internal model of its world and uses this model in its tactics to collect nearby weapons and health, and set ambushes. It also tries to anticipate the actions of human players by putting itself in their shoes (creating an internal model of their situation garnered from its perception of the player) and projecting what it would do if it were the human player” (p. 22). The Soar Quakebot did not have as many rules as a fully developed commercial synthetic character would, nor did it have any probabilistic reasoning or learning capabilities, but it did show some potential to create realistic and believable synthetic characters.

Yet similar efforts never gained much traction due, at least in part, to the reality that major consumers of interactive simulation environments – e.g. training, gaming, and entertainment communities – are not interested in whether these synthetic characters are actually intelligent or not, but only in the perceived intelligence of these characters. This perspective severely limits the resources allocated for efforts in general intelligence. Many of these interactive simulation environments even prefer stricter control on the interactions between synthetic characters and real humans, as such tight control can prevent synthetic characters from exhibiting behaviors that may be potentially perceived as unintelligent. Many simulations use the labor-intensive process of defining hierarchical representations of goals and rules that cover almost all of the simulation's situations. For a higher level of control of these situations, it is a common practice to "cheat" by collecting information on human opponents without actually perceiving them in the gaming environment (Bourg & Seaman 2004; Schaeffer, Bulitki, & Buro, 2008). When these rule-based decision models are combined with very realistic graphics and animations, striking applications can be created with even a relatively simple rule structure (DeVault et al., 2014).

The relative success of applications with extensive rule structures in guiding behavior impedes the penetration of even basic narrow AI approaches into interactive simulation environments. Basically, training by observation is considered to be too time consuming and the resulting behavior unpredictable (Robertson & Watson, 2014; Umarov, Mozgovoy, & Rogers, 2012) when the sole focus is on perceived intelligence. There are even some experts who believe that all that users want is predictability and that it is therefore unnecessary to pursue synthetic characters that learn by experience. On the other hand, there are other experts and researchers who believe that human behavioral models with cognitive capabilities and more human-like intelligence can make simulations more varied and the games more enjoyable (Lucas & Kendall, 2006; Robertson & Watson, 2014). Many interactive simulation models already desire synthetic characters that are not only believable but also indistinguishable from humans – performing coherent sequences of believable behavior sustained across different tasks and environments. Human behavior is still considered a challenge by/for the simulation community (Taylor et al., 2015), where there is a need for creating truly autonomous synthetic characters that actually have generic perception, action and learning capabilities appropriate for a variety of environments. Such characters are almost impossible to generate by strictly controlling their behavior, as this approach would require generating rules for each different task, each different environment, and every possible interaction. These characters need to be trained by observation, and thus they ideally require architectures with general learning capabilities.

Characters that are truly autonomous would be very interesting for interactive simulation environments, as they would be defined by the experiences they have undergone and the environments where they have been, contributing to the personality of each character. Eventually, the hope is that progress in the development of these characters will yield *plug compatibility* between humans and artificial systems (Tambe et al., 1995). Such characters would open doors to uncharted territories and to immense opportunities for the training, gaming, and entertainment communities. One recent example is the deep reinforcement-learning algorithm that has been used to train a synthetic character from high-dimensional sensory inputs to play multiple Atari games (Mnih et al., 2015). The synthetic character in this study received only the game score and pixels as input and was able to achieve a level comparable to humans using the same algorithm, network structure and hyperparameters across games. There was no interaction with other synthetic characters or switching of contexts in this study, but it is still an inspiring achievement in using real perceptions.

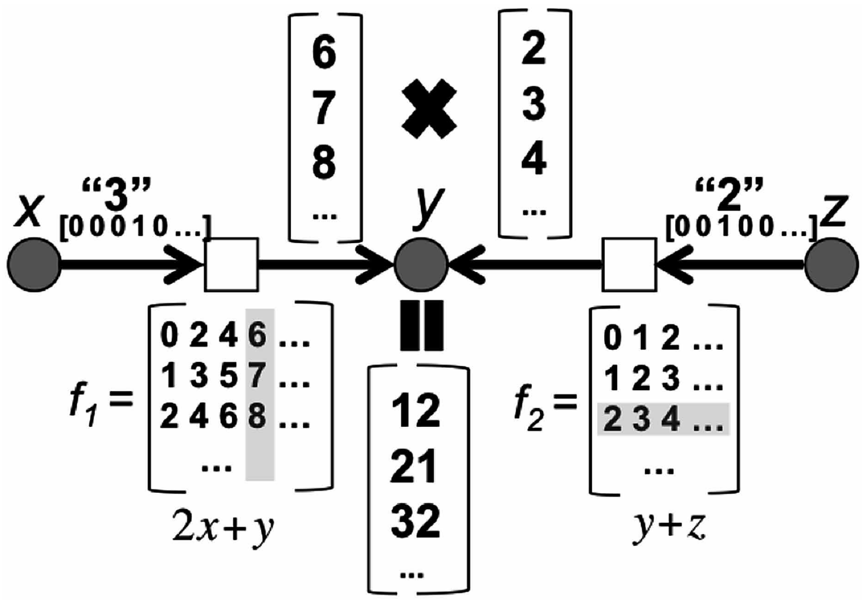
Overall, interest in general intelligence is increasing, and many rich applications with truly autonomous synthetic characters are envisioned for the future. Architectures capable of creating such characters are thus very intriguing.

3. SIGMA

The Sigma cognitive architecture is built on *factor graphs* (Kschischang, Frey, & Loeliger 2001) – undirected graphical models (Koller & Friedman 2009) with variable and factor nodes, and functions that are stored in the factor nodes. Graphical models provide a general computational technique for efficient computation with complex multivariate functions – implemented via hybrid mixed *piecewise-linear functions* (Rosenbloom, 2011a) in Sigma – by leveraging forms of independence to: decompose them into products of simpler functions; map these products onto graphs; and solve the graphs via message passing or sampling methods. The *summary product algorithm* (Kschischang, Frey, & Loeliger 2001) is the general inference algorithm in Sigma (Figure 1). Graphical models are particularly attractive as a basis for broadly functional, yet simple and theoretically elegant, cognitive architectures because they provide a single general representation and inference algorithm for processing symbols, probabilities and signals.

The Sigma architecture defines a high-level language of *predicates* and *conditionals* that compiles down into factor graphs. Predicates specify relations over continuous, discrete and/or symbolic arguments. They are defined via a name and a set of typed arguments, with *working memory* (WM) containing predicate instantiations as functions within a WM sub-graph. Predicates may also have perception and/or long-term memory (LTM) functions. For perceptual predicates, factor nodes

Figure 1. Summary product computation over the factor graph for $f(x,y,z) = y^2+yz+2yx+2xz = (2x+y)(y+z) = f_i(x,y)f_2(y,z)$ *of the marginal on y given evidence concerning x and z*



for *perceptual buffers* are connected to the corresponding WM sub-graphs. For example, Observed(object:object visible:boolean) is a perceptual predicate with two arguments: (1) object of type object; and (2) visible of type boolean. This predicate specifies which objects are visible to the agent at any particular time. For memorial predicates, *function factor nodes* (FFNs) are likewise connected to the corresponding WM sub-graphs. Messages into FFNs provide the gradients for learning the nodes' functions. Gradient calculations require identifying parent-child relationships among predicate arguments; for instance, the predicate function for the Object-Location-X(object:object x:location%) predicate defines a distribution over the x coordinate given an object (x is marked as the child variable in this predicate by %).

Conditionals structure LTM and basic reasoning, compiling into more extended sub-graphs that interconnect with the appropriate WM sub-graphs. Conditionals are defined via a set of *predicate patterns* – in which type specifications are replaced by constants and variables – and an optional *function* over pattern variables. *Conditions* and *actions* are predicate patterns that behave like the respective parts of rules, pushing information in one direction from the conditions to the actions. The example conditional in Figure 2 updates the information about which objects have

Figure 2. Conditional for context information

```
CONDITIONAL Seen
     Conditions: Observed(object:o visible:true)
     Actions: Seen-Objects(object:o)
```

been seen so far, based on the information in the Observed predicate. *Condacts* are predicate patterns that support the bidirectional processing that is key to probabilistic reasoning, partial matching, constraint satisfaction and signal processing. Examples of probabilistic networks based on condacts and functions will be seen in the coming sections on virtual humans. Overall, conditionals provide a deep combination of rule systems and probabilistic networks.

The core processing in Sigma is driven by a cognitive cycle that comprises input, graph solution, decisions, learning, and output. Graph solution yields inference via the summary product algorithm, by *product* of the messages coming into each node – including the node's function when it is a factor node – and then *summarization*, via integration or maximum, of unneeded variables from outgoing messages. Perception and action are modeled as sub-symbolic reasoning in Sigma that occurs predominantly within graph solution, rather than within external modules (Joshi, Rosenbloom, & Ustun, 2014). Specifically, perceptual information collected during the input phase is passed to WM, where it is combined with the current contents of the WM and knowledge available within the LTM – such as might define a hidden Markov model (HMM) or a conditional random field (CRF) – yielding new WM content at the end of the graph solution phase.

Decisions in Sigma, in the classical sense of choosing one among the best operators to execute next, are based on an architecturally distinguished selection predicate – Selected (state:state operator:operator) – with typically the operator associated with the highest value, or utility, in the distribution for a state being selected. The learning phase then modifies the long-term functions in the graph via a general form of gradient-descent learning. Finally, in the output phase, selected actions are performed in the outside world.

A single cognitive cycle yields *reactive processing*; a sequence of them yields *deliberative processing*; and *reflective processing* occurs when no operator can be selected, or a selected operator can't be applied, similar to impasses in Soar (Laird, 2012). Activities such as classifying an object or executing a body of rules can all occur reactively, within a single cognitive cycle. Deliberative processing utilizes reactive processing, and thus individual cognitive cycles, as its inner loop in providing sequential, algorithmic behavior that leads to problem solving and reasoning.

When impasses occur, deliberative processing – either in the same or different problem spaces – can be used reflectively to yield the knowledge that would resolve the impasses.

4. THE IMMERSIVE NAVAL OFFICER TRAINING SYSTEM (INOTS)

The first proof-of-concept Sigma Virtual Human discussed in this paper focuses on language and conversation. This Sigma model replicates the mind of a deployed, but cognitively rather simple, virtual human that was originally developed for the Immersive Naval Officer Training System (INOTS) system, a training vehicle for junior leaders in the U.S. Navy in interpersonal communication skills (Campbell et al., 2011). INOTS replaces the live role player in a traditional role-play practice with a virtual human, creating a blended learning environment that merges traditional classroom instruction with a mixed reality training setting. For instance, the virtual human's behavior and dialog are tailored to provide the cues indicative of when to apply specific strategies and skills for the problems on hand. The implementation uses a branching narrative such that each utterance (also known as a challenge in INOTS) by the virtual human is a decision point with a fixed number of possible responses by the trainee. At each challenge, the system displays the possible responses for the trainee to choose. The trainee is expected to speak one of the possible responses that are recognized by the system. Each response is linked to another decision point in the narrative representing the virtual human's reaction and the next set of possible utterances for the trainee. The scripted dialogue of the virtual human and the scripted possible responses for the trainee allow practicing the strategies that an expert leader would use to solve certain types of interpersonal problems (Campbell et al., 2011).

4.1 INOTS Sigma Model

The overall control structure for the INOTS virtual human is based on a branching – directed acyclic – network of states and utterances plus an utterance classifier that is used to determine which choice to make at each point, and thus also what response to make. Two separate tools were combined to make this work, one for the branching structure and the other for the utterance classifier (Campbell et al., 2011). In Sigma these two functionalities were mapped, respectively, onto: (1) deliberative movement in a discourse problem space composed of operators for speaking and listening, and (2) a reactive bag-of-words naïve Bayes utterance classifier (Ustun, Rosenbloom, Kim, & Li 2015).

The basic control structure alternates whose turn it is to talk, with a listen operator selected when it is the human's turn and a speak operator selected when it is the VH's turn. When it is the human's turn, their utterance – which is typed rather than spoken to Sigma – is classified as one from among a small set of standard utterances that have been predefined for the corresponding dialogue state; there are typically 1-3 standard utterances per state. Two distinct types of conditionals realize this naïve Bayes classification: (1) the prior on the utterance being classified (Figure 3), and (2) the conditional probabilities on the words given the utterances, with one conditional per word in the vocabulary (Figure 4). The Prior-Meaning conditional (Figure 3) defines the valid utterances for the current dialog state by having an Utterance(meaning:utterance-number) predicate pattern as a condact, where utterance-number is a discrete numeric type that uniquely identifies the distinct utterances known to the system. The function in the Prior-Meaning conditional, in which only valid utterances for the current dialog state are assigned non-zero probability, captures the prior on the utterances (m is the child variable for this function). In the current implementation, a uniform distribution is used as a prior over the valid utterances for the current dialogue state.

The conditional in Figure 4 demonstrates how conditional probabilities are represented for the word play given the utterance. A conditional probability distribution is defined over the likelihood of the presence of play in a sentence spoken by the trainee for each associated meaning via the function in the Sentence-Word-Play conditional (wp is the child variable for this function). Similarly, for each word in the VH's vocabulary, there is a distinct conditional and function that capture the likelihood of the presence of that word in each utterance. At the end of graph solution, a posterior distribution is yielded for the Utterance predicate and the utterance

Figure 3. Prior on the sentence

```
CONDITIONAL PRIOR-MEANING
     Conditions:  Dialog-State(state-number:sn)
     Condacts:    Utterance(meaning:m)
     Function(sn,m): …
```

Figure 4. Example conditional for the bag-of-words model

```
CONDITIONAL SENTENCE-WORD-PLAY
     Condacts:    Utterance(meaning:m)
                  Word-Play(word-play:wp)
     Function(m,wp): …
```

with the highest probability is selected as the one that is heard. However, if none of the utterances have a probability greater than 0.5, it is assumed that the trainee has entered something unrecognizable, and s/he is asked to reenter the utterance.

The selected standard utterance – as represented in the Heard-Utterance predicate – then feeds the transition conditional (Figure 5), which determines the subsequent dialogue state. When it is the VH's turn to speak, only one possible utterance exists for each dialogue state, so it is simply produced and the next dialogue state is determined as a direct function of the previous one.

This VH mind works in real-time, and is capable of holding a dialogue comparable to the one in the original system. However, it is still rather limited in its capabilities. Even though Sigma is capable of learning the conditional probability distributions for the bag-of-words model, and has done so in a variety of other tasks, for the current implementation these distributions were determined outside of Sigma and then imported into conditional functions. The original goals for implementing this particular VH mind within Sigma were to show that simple such minds could be created simply, and in an integrated fashion, and that they could then be extended to more sophisticated intelligent behavior as required – typically customer requirements in domains like this start simple, but then grow as it is understood what the initial system can and cannot do. The first part of this was demonstrated, but not yet the second part, although such extensions could still be explored in the future.

5. PHYSICAL SECURITY SYSTEM

Chen et al. (2011) discussed the fusion of symbolic and probabilistic reasoning at an earlier stage of the development of Sigma. In that study, initial steps towards grand unification were demonstrated when perception, localization and decision-making were implemented within a single graphical model, with interaction among these capabilities modulated through shared variables. The work here greatly expands on this approach to yield a more significant combination of capabilities, plus a deployment in virtual humans – within a 3D virtual environment (rather than a toy one-dimensional space) – embodied in the SmartBody character animation system

Figure 5. Transition conditional

```
CONDITIONAL Transition
    Conditions:   Heard-Utterance(heard:h)
    Actions:      Dialog-State(state-number:h)
```

(Shapiro, 2011). SmartBody's internal movement, path-finding and collision detection algorithms are used in animating the virtual human's actions, although eventually much of this is to be moved within Sigma. Sigma has no "direct" access to the virtual environment, being limited instead to perceiving and acting on it through a (deliberately) noisy interface.

A number of capabilities within Sigma have been combined to construct an adaptive, interactive virtual human in a virtual environment. The VH is adaptive not only in terms of dynamically deciding what to do based on the immediate circumstances, but also in terms of embodying two distinct forms of relevant learning: (1) the automated acquisition of maps of the environment from experience with it, in the context of the classic robotic capability of Simultaneous Localization and Mapping (SLAM); and (2) reinforcement learning (RL), to improve decision making based on experience with the outcomes of earlier decisions. The VH is interactive both in terms of its (virtual) physical environment – through high-level perception and action – and other participants, although the latter is still quite limited. Speech and language are being investigated in Sigma, but neither is deployed in these virtual humans yet, so social interaction is limited to constructing – actually, learning, with the help of RL – models of the self and others. These forms of adaptivity and interaction are combined together, along with for these initial virtual humans a basic rule-based decision framework, all within Sigma.

5.1 Physical Security System Conceptual Model

Physical security systems are comprised of structures, sensors, protocols, and policies that aim to protect fixed-site facilities against intrusions by external threats, as well as unauthorized acts by insiders. Physical security systems are generally easy to understand but they also allow complex interactions to emerge among the agents. These properties make physical-security-systems simulation a natural candidate as a testbed for developing cognitive models of synthetic characters (Ustun, Yilmaz, & Smith 2006). Similar to the discussion made by Ustun et al. (2006), a physical-security-system scenario in a retail store has been selected as a platform to develop and test Sigma VH models.

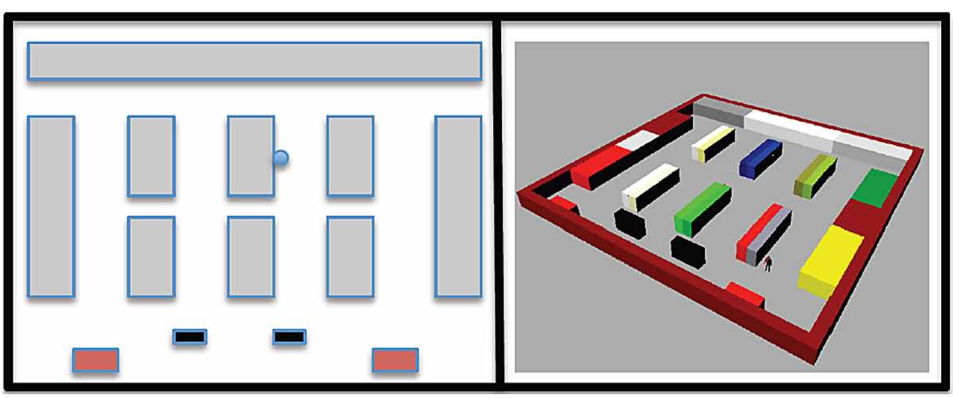
In a typical retail-store shoplifting plot, offenders first pick up merchandise in a retail store and then try to leave without getting caught by any of the store's security measures. A simple grab-and-run scenario is considered in this paper (a large number of different scenarios are possible). In this scenario, the intruder needs to locate the desired item in the store, grab it, and then leave the store. The role of security is to detain the intruder before s/he leaves the store. A basic assumption is that it is not possible to tell what the intruder will do until s/he picks up an item and starts running. The security can immediately detect the activity and start pursuing the intruder once the item is picked up (assuming CCTV). If the intruder makes it to the door, it is considered a success for the intruder.

For the basic setup, it is assumed that the intruder does not know the layout of the store and hence it has to learn a map and be able to use it to localize itself in the store. When the intruder locates the item of interest, it grabs the item and leaves the store via one of the exits. In the hypothetical retail store used here (Figure 6), there are shelves (gray rectangles), the item of interest (the blue circle) and two entry/exit doors (red rectangles). The intruder leaves the store via either (1) the door it used to enter or (2) the door closest to the item of interest. The main task for security is to learn about the exit strategies of intruders and use this to effectively detain them.

SmartBody (Shapiro, 2011), a Behavior Markup Language (BML) (Kopp et al., 2006) realization engine, is used as the character animation platform for this study, with communication between

the Sigma VH model and SmartBody handled via BML messages. In the current setup, locomotion and path finding are delegated to the SmartBody engine. Sigma sends commands and queries to SmartBody to perform these tasks and to return perceptual information. Two basic types of perception are utilized by the Sigma VH model: (1) information about the current location of the agent, mimicking the combination of direction, speed and odometry measurements available to a robot; and (2) objects that are in the visual field of the agent, along with their relative distances, mimicking the perception of the environment for a robot. Location information is conveyed to the Sigma VH model with noise added – perfect location information is not available to the model.

Figure 6. Layout of the store and its SmartBody representation



5.2 Physical Security System Sigma Model

In a typical physical security system setting there are intruders (shoplifters) and security personnel. There may also be neutrals, but they are not modeled in this work. In this initial implementation, an intruder and a security agent are modeled as virtual humans. There are two distinct types of learning (and probabilistic inference) in this scenario: (1) The intruder agent does not know the layout of the store in advance, and so it must learn a map of the store while simultaneously localizing itself in the learned map (SLAM model), and (2) The security personnel agent infers the strategy of the intruder agent – i.e., whether it exits through the entry door or the closest door – by first learning a policy for the intruder agent via RL and then using this policy and the intruder agent's actions to determine on each trial the relative likelihoods of the two strategies being used (Section 5.2.1). The intruder also needs a decision framework to dynamically decide what to do based on its immediate circumstances (Section 5.2.2).

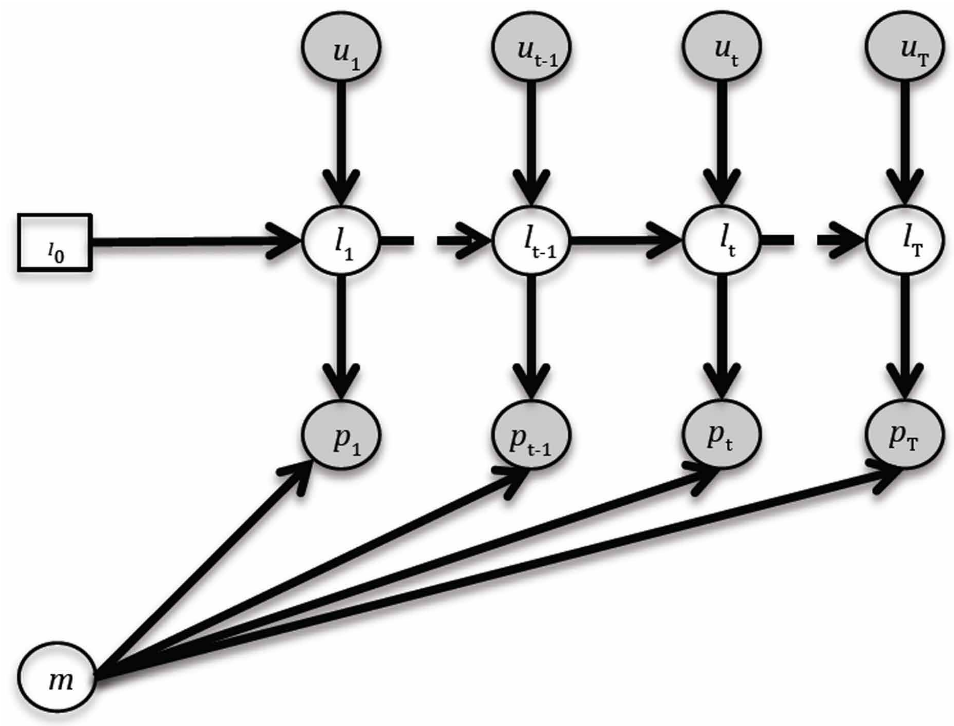
5.2.1 Learning in the Physical Security System VH Models

Sigma can effectively utilize different types of learning, a key capability leading towards adaptive and interactive synthetic characters that are truly autonomous. Here, the agents learn the probability distributions in a Bayesian network and Q functions for an RL algorithm using the same basic set of architectural mechanisms, something that would require multiple distinct modules in other architectures if it could be done at all.

In the SLAM model, the intruder has no a priori knowledge about the layout of the retail store (which is as shown in Figure 6). Therefore, it needs to learn a map of the store while simultaneously using the map to localize itself in the store. A 31×31 grid is imposed on the store for map learning. A virtual human only occupies a single grid cell, whereas objects in the environment – such as shelves – can span multiple cells. While performing SLAM, a Dynamic Bayesian Network (DBN) is utilized (Figure 7) (Grisetti, Kummerle, Stachniss, & Burgard 2010). In this representation, l is the location, u captures the odometry readings, p represents perceptions of the environment, and m is the map of the environment.

The Sigma VH model defines two perceptual predicates – Location-X(x:location) and Location-Z(z:location) – to represent the location of the virtual human on the grid; here, the location type is discrete numeric, with a span of 31. Together these two predicates represent the space of 2D cell locations in the grid. They are perceptual predicates, and hence induce perceptual buffers that emulate odometry readings. In particular, the current location of the agent is perceived with noise – the default noise model assumes that any neighboring cell of the correct cell may be perceived

Figure 7. Dynamic Bayesian Network of the SLAM process



as the agent's current location. In addition, the objects in the visual field are also perceived along with the relative distances of the center of these objects to the agent's location.

The DBN is captured in two almost identical conditionals, one for x (Figure 8) and a similar one for z. These conditionals convert relative locations of objects given the agent to absolute locations in the map. They use Sigma's general capability for affine transforms in visual imagery (Rosenbloom, 2012) to offset the agent's current location by the distance to the object being updated. In Figure 8, the Object-Location-X predicate is a memorial predicate so it has a function that learns the x coordinates in the map via gradient descent. Since both the Location-X and Object-Location-X patterns are condacts, the processing is bidirectional between them; both perception of the VH's location and perception of the object locations have an impact on the posterior for the VH's location. This bidirectional processing forms the basis for SLAM, where the map is learned while it is simultaneously used for localization.

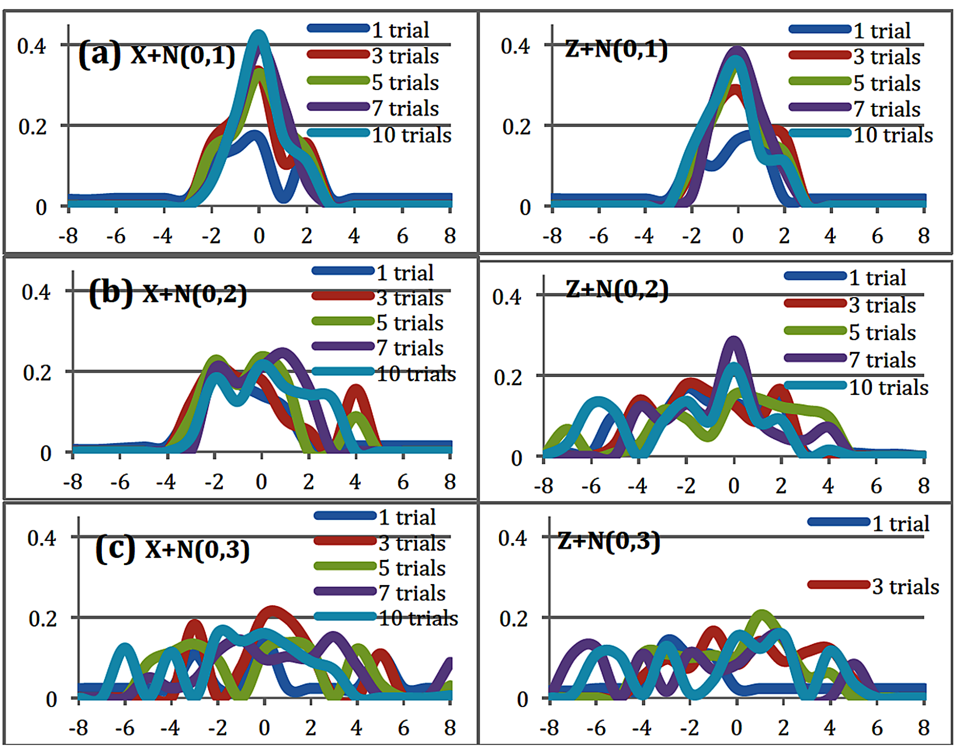
Once it was verified that without noise this implementation of SLAM could acquire appropriate maps, an additional experiment was run on this VH to explore more probabilistic learning under noise, and thus uncertainty. This experiment in-

Figure 8. SLAM conditional for the x coordinate

```
CONDITIONAL SLAM-X
     Conditions:   Observed(object:o visible:true)
                   Object-Distance-X(dist-x:dx object:o)
     Condacts:     Location-X(x:lx)
                   Object-Location-X(object:o x:(lx-dx))
```

vestigated the effect of different noise models on the number of trials required to learn a correct map. A normally distributed noise with mean 0 and varying standard deviations was added to the location perceptions of the VH in this experiment, with the results shown in Figure 9. As expected, the learned map is sensitive to noise, but small amounts of it can be tolerated, even given just a small number of trials (Figure 9 (a)). However, high-variation noise models make it hard to learn correct maps. In Figure 9(c), the correct location of the item of interest is not effectively

Figure 9. Learned distributions over the displacements of the x (left) and z (right) coordinates from the correct values for the item of interest after varying number of trials, under three different noise models



learned even after 10 trials. These experiments provide evidence that general probability distributions are effectively learned in the Sigma cognitive architecture.

In addition to mapping and localizing itself like the intruder, the security agent needs to reason about the intruder agent's actions for effective detainment, and hence it needs to learn about the policies used by the intruder agent. One basic assumption made in this paper is that it is easy to recognize that a grab-and-go scenario has been initiated, by observing the pick-up behavior of the intruder.[1] However, even though security can easily recognize when such a scenario has been initiated, it still needs to intercept the intruder before it leaves the retail store. As there are two exit doors, early anticipation of the intruder's choice increases the chances of a successful detention.

Theory of Mind (ToM) involves formation of models of others and generation of expectations about their behavior based on these models to enable effective decisions in social settings (Whiten, 1991). In decision theoretic approaches to ToM, such models can be represented as reward functions. For the intruder in our scenario there are two possible models, distinguished by whether a reward is received when the agent returns to its door of entry or when it reaches the nearest door from the item of interest. This corresponds to Bayesian approaches to multi-agent modeling that use a distribution over a set of policies to specify the beliefs that one agent has about another (Pynadath & Marsella, 2005).

Here, as in the work by Pynadath et al. (2014), RL is leveraged in selecting among models of other agents; in particular it is used so that the security agent can learn a model of the intruder. First a form of multiagent RL is used to learn a distinct *policy*, or *Q function*, for the intruder under each possible model, and then these policies are used in combination with the perception of the intruder's actions to yield a gradient over the two models that is proportional to the models' Q values for the performed actions. For example, the model for which the observed action has higher Q values will achieve increased likelihood in the posterior distribution. This very quickly enables the security VH to determine the correct door in the experiments run, with the Q value learned by RL substituting for what would otherwise be a need to explicitly extrapolate and compare the paths the intruder might take to the two doors.

The conditional that compares the Q values and generates a posterior distribution for the models is shown in Figure 10. It multiplies the Q values for the observed action – specified by the location of the intruder agent and the direction of movement from that location – in each policy by 0.1, to scale utilities in [0,10] to values for selection in [0,1], and then projects these values onto the model predicate to generate a posterior distribution on the model currently being used by the intruder.

In general, RL enables agents to learn effective policies for task performance based on rewards received over a sequence of trials (Sutton & Barto, 1998). In

Figure 10. Model prediction conditional

```
CONDITIONAL PREDICT-MODEL
    Conditions:  Previous-RL-Loc(location:loc)
                 RL-Direction(direction:d)
                 Q(model:m location:loc direction:d
                   value:[0.1*q])
    Actions:     Model(model:m)
```

Sigma, RL is not a separate architectural learning algorithm, but occurs through a combination of gradient-descent learning and the appropriate knowledge expressed in predicates and conditionals. This also means that, similar to the work by Pynadath et al. (2014), it is possible to move from single-agent to multi-agent RL, and from RL given a single reward function to RL given a range of possible reward functions – i.e., models – by appropriately changing the predicates and conditionals.

5.2.2 Behavior Rules in the Physical Security System VH Models

The objective of the intruder is to grab the item of interest (Figure 6) and to leave the store through one of the exits without being detained. In the current implementation, four basic behaviors, with corresponding Sigma operators, are available to the intruder: (1) *walk towards target object*; (2) *run towards target object*; (3) *pick up target object*; and (4) *walk towards random object*. Target objects play a role in the intruder achieving its goals, while random objects drive exploration of the store.

The intruder is initialized with a sequence of target objects that it needs either to walk towards or to pick-up. Given that the intruder does not have a priori knowledge of the store, the location of the current target object may be unknown to it. In this case it needs to explore the store, mapping it in the process, to locate the target object. The basic operator used for exploration is *walk towards random object*. The intent is that doing this will help the agent discover new objects, and eventually the target object.

This exploratory operator is always available; however, if other more task-relevant behaviors are available, they take precedence. For example, Figure 11 shows a conditional in which the operator *walk towards target object* is suggested for selection with a utility of 0.5 when the virtual human has seen the target object (and hence, there is an estimate of its location). Exploration has a lower utility, so walking to a target object takes precedence if both operators are available.

When the virtual human is within a threshold distance of the target object, a new operator – *pick-up target object* – is selected. The model terminates when the intruder reaches its preferred exit door, which acts as the target object for the *run towards target object* operator.

Figure 11. Conditional suggesting the walk towards target object operator

```
CONDITIONAL WALK-TOWARDS-TARGET
     Conditions:   Target-Object(object:o)
                   Seen-Objects(object:o visible:true)
     Actions:      Selected(operator:walk-target)
     Function:     0.5
```

Combining the short-term adaptivity provided by this rule-based behavior with the long-term adaptivity provided by map and ToM learning yields embodied Sigma-based VHs that exhibit effective social and environmental interaction in the physical security model.

6. CONCLUSION

The two models presented here are important steps in the maturation of Sigma as they are the first uses of Sigma to control virtual characters, demonstrating small but diverse combinations of symbolic and sub-symbolic processing. Each embodies an important subset of the capabilities that virtual humans should support. It is hoped that over the long term such combinations of the capabilities of traditional symbolic cognitive architectures and probabilistic graphical models may assist in the generation of high-fidelity computational behavior models. There are short-term plans for extending these particular models: (1) the INOTS model is to incorporate a Sigma-based continuous speech recognizer, and (2) the physical security model is to incorporate natural language, speech, and affect. Furthermore, the variety of participants involved – intruders, security, and neutrals – makes the physical security model in particular extremely flexible for the generation of scenarios that encompass many different interactions among VHs and between VHs and humans. As the scenarios get more complex, the expectation is that the forms of cognition exhibited will keep up, going beyond simple rule-based reasoning to more involved combinations of reactive, deliberative and reflective processing. Further experiments with these models will help to test the limits and assess Sigma as an architecture for general intelligence.

Broadly capable real-time virtual humans provide ideal test beds for demonstrating and evaluating progress on Sigma's four desiderata, with "broadly capable" challenging the extent of its grand unification; "real-time" challenging its sufficient efficiency (Rosenbloom, 2012); and "virtual humans" with their need to exhibit human-like behavior in artificial systems, challenging its generic cognition. The virtual humans here also leverage Sigma's functionally elegant approach to providing and combining the requisite capabilities, such as rule-based reasoning, language, SLAM, ToM, and RL.

ACKNOWLEDGMENT

This effort has been sponsored by the Office of Naval Research and the U.S. Army. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred. We would also like to thank Ari Shapiro for his overall support with SmartBody.

REFERENCES

Bourg, D. M., & Seaman, G. (2004). *AI for Game Developers*. O'Reilly Media.

Campbell, J. (2011). Developing INOTS to Support Interpersonal Skills Practice. In *Proceedings of the Thirty-second Annual IEEE Aerospace Conference (IEEEAC)*. Washington, DC: IEEE doi:10.1109/AERO.2011.5747535

Cassimatis, N. (2010). An architecture for adaptive algorithmic hybrids. *Systems, Man, and Cybernetics, Part B: Cybernetics*. *IEEE Transactions on*, *40*(3), 903–914.

Chen, J., Demski, A., Han, T., Morency, L. P., Pynadath, D. V., Rafidi, N., & Rosenbloom, P. S. (2011). Fusing Symbolic and Decision-Theoretic Problem Solving+ Perception in a Graphical Cognitive Architecture. In Biologically Inspired Cognitive Architectures.

Danks, D. (2014). *Unifying the mind: Cognitive representations as graphical models*. MIT Press.

DeVault, D. (2014). SimSensei Kiosk: A virtual human interviewer for healthcare decision support. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent systems*.

Goertzel, B. (2014). Artificial General Intelligence: Concept, state of the art, and future prospects. *Journal of Artificial General Intelligence.*, *5*, 1–46.

Grisetti, G., Kummerle, R., Stachniss, C., & Burgard, W. (2010). A tutorial on graph-based SLAM. *Intelligent Transportation Systems Magazine, IEEE*, *2*(4), 31–43. doi:10.1109/MITS.2010.939925

Ji, Q., Gray, W. D., Guhe, M., & Schoelles, M. J. (2004). Towards an integrated cognitive architecture for modeling and recognizing user affect. In *AAAI Spring Symposium on Architectures for Modeling emotion: cross-disciplinary foundations*.

Joshi, H., Rosenbloom, P. S., & Ustun, V. (2014). Isolated word recognition in the Sigma cognitive architecture. In Biologically Inspired Cognitive Architectures. doi:10.1016/j.bica.2014.11.001

Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Kopp, S., Krenn, B., Marsella, S., Marshall, A. N., Pelachaud, C., Pirker, H., & Vilhjálmsson, H. (2006). Towards a common framework for multimodal generation: The behavior markup language. In Intelligent Virtual Agents.

Kschischang, F. R., Frey, B. J., & Loeliger, H. A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, *47*(2), 498–519. doi:10.1109/18.910572

Laird, J., & van Lent, M. (2001). Human-level AI's killer application: Interactive computer games. *AI Magazine*, *22*(2), 15.

Laird, J. E. (2012). *The Soar Cognitive Architecture*. Cambridge, MA: MIT Press.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*(2), 141–160. doi:10.1016/j.cogsys.2006.07.004

Lee-Urban, S., Trewhitt, E., Bieder, I., Odom, J., Boone, T., & Whitaker, E. (2015). CORA: A Flexible Hybrid Approach to Building Cognitive Systems. In *Proceedings of the Third Annual Conference on Advances in Cognitive Systems Poster Collection* (p. 28).

Lucas, S. M., & Kendall, G. (2006). Evolutionary Computation and Games. *IEEE Computational Intelligence Magazine*, *1*(1), 10–19. doi:10.1109/MCI.2006.1597057

Mikolov, T., Joulin, A., & Baroni, M. (2015). *A roadmap towards machine intelligence*. Retrieved from http://arxiv.org/abs/1511.08130

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., & Hassabis, D. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. doi:10.1038/nature14236 PMID:25719670

Pynadath, D. V., & Marsella, S. C. (2005). PsychSim: Modeling theory of mind with decision-theoretic agents. IJCAI.

Pynadath, D. V., Rosenbloom, P. S., & Marsella, S. C. (2014). Reinforcement Learning for Adaptive Theory of Mind in the Sigma Cognitive Architecture. In Artificial General Intelligence.

Pynadath, D. V., Rosenbloom, P. S., Marsella, S. C., & Li, L. (2013). Modeling two-player games in the Sigma graphical cognitive architecture. In Artificial General Intelligence. doi:10.1007/978-3-642-39521-5_11

Robertson, G., & Watson, I. (2014). A review of real-time strategy game AI. *AI Magazine*, *35*(4), 75–204.

Rosenbloom, P. S. (2010). Combining procedural and declarative knowledge in a graphical architecture. In *Proceedings of the 10th International Conference on Cognitive Modeling*.

Rosenbloom, P. S. (2011a). Bridging dichotomies in cognitive architectures for virtual humans. In *Proceedings of the AAAI Fall Symposium on Advances in Cognitive Systems*.

Rosenbloom, P. S. (2011b). From memory to problem solving: Mechanism reuse in a graphical cognitive architecture. In Artificial General Intelligence.

Rosenbloom, P. S. (2012). Deconstructing reinforcement learning in Sigma. In Artificial General Intelligence. doi:10.1007/978-3-642-35506-6_27

Rosenbloom, P. S. (2012). Towards a 50 msec cognitive cycle in a graphical architecture. In *Proceedings of the 11th international conference on cognitive modeling*.

Rosenbloom, P. S. (2012). Extending mental imagery in Sigma. In *Artificial General Intelligence* (pp. 272–281). Springer Berlin Heidelberg. doi:10.1007/978-3-642-35506-6_28

Rosenbloom, P. S. (2013). The Sigma cognitive architecture and system. *AISB Quarterly, 136*.

Rosenbloom, P. S. (2014). Deconstructing episodic memory and learning in Sigma. In *Proceedings of the 36th Annual Conference of Cognitive Science Society*.

Rosenbloom, P. S., Demski, A., Han, T., & Ustun, V. (2013). Learning via gradient descent in Sigma. In *Proceedings of the 12th International Conference on Cognitive Modeling*.

Rosenbloom, P. S., Gratch, J., & Ustun, V. (2015). *Towards Emotion in Sigma: From Appraisal to Attention*. Artificial General Intelligence.

Schaeffer, J., Bulitki, V., & Buro, M. (2008). Bots Get Smart: Can video games breathe new life into AI research? *IEEE Spectrum*, *45*(12), 48–56. doi:10.1109/MSPEC.2008.4688952

Schmidhuber, J. (2015). *On Learning to Think: Algorithmic Information Theory for Novel Combinations of Reinforcement Learning Controllers and Recurrent Neural World Models*. Retrieved from http://arxiv.org/abs/1511.09249

Shapiro, A. (2011). Building a character animation system. In Motion in Games. doi:10.1007/978-3-642-25090-3_9

Strannegård, C., von Haugwitz, R., Wessberg, J., & Balkenius, C. (2013). A cognitive architecture based on dual process theory. In *Artificial General Intelligence* (pp. 140–149). Springer Berlin Heidelberg. doi:10.1007/978-3-642-39521-5_15

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Swartout, W. (2010). Lessons learned from virtual humans. *AI Magazine*, *31*(1), 9–20.

Tambe, M., Johnson, W. L., Jones, R. M., Koss, F., Laird, J. E., Rosenbloom, P. S., & Schwamb, K. (1995). Intelligent agents for interactive simulation environments. *AI Magazine*, *16*(1), 15.

Taylor, S. J., Khan, A., Morse, K. L., Tolk, A., Yilmaz, L., Zander, J., & Mosterman, P. J. (2015). Grand challenges for modeling and simulation: simulation everywhere—from cyberinfrastructure to clouds to citizens. *Simulation: Transactions of the Society for Modeling and Simulation*.

Traum, D., Marsella, S. C., Gratch, J., Lee, J., & Hartholt, A. (2008). Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *Intelligent Virtual Agents* (pp. 117–130). Springer Berlin Heidelberg. doi:10.1007/978-3-540-85483-8_12

Umarov, I., Mozgovoy, M., & Rogers, P. C. (2012). Believable and effective AI agents in virtual worlds: Current state and future perspectives. *International Journal of Gaming and Computer-Mediated Simulations*, *4*(2), 37–59. doi:10.4018/jgcms.2012040103

Ustun, V., & Rosenbloom, P. S. (2015). Towards adaptive, interactive virtual humans in Sigma. In *Intelligent Virtual Agents* (pp. 98–108). Springer International Publishing. doi:10.1007/978-3-319-21996-7_10

Ustun, V., Rosenbloom, P. S., Kim, J., & Li, L. (2015). Building High Fidelity Human Behavior Model in the Sigma Cognitive Architecture. In *Proceedings of the 2015 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

Ustun, V., Rosenbloom, P. S., Sagae, K., & Demski, A. (2014). Distributed vector representations of words in the Sigma cognitive architecture. In Artificial General Intelligence. doi:10.1007/978-3-319-09274-4_19

Ustun, V., Yilmaz, L., & Smith, J. S. (2006). A conceptual model for agent-based simulation of physical security systems. In *Proceedings of the 44th annual Southeast regional conference*. ACM. doi:10.1145/1185448.1185530

Whiten, A. (Ed.). (1991). *Natural Theories of Mind*. Oxford, UK: Basil Blackwell.

Zemla, J. C., Ustun, V., Byrne, M. D., Kirlik, A., Riddle, K., & Alexander, A. L. (2011). An ACT-R model of commercial jetliner taxiing. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 55, No. 1, pp. 831-835). Sage Publications. doi:10.1177/1071181311551173

ENDNOTE

[1] The details of this recognition process are beyond the scope of this paper but there are a variety of behavioral cues (e.g. posture changes while concealing an item, gait changes under stress etc.) that could be revealing. Exhibiting and detecting such cues is one of a number of intriguing future directions for this work.