Intl. Journal of Human–Computer Interaction, 30: 250–265, 2014

ISSN: 1044-7318 print / 1532-7590 online
DOI: 10.1080/10447318.2013.849550

The Cognitive Benefit of Dynamic Representations on Procedural Skill Acquisition: A Computational Modeling Approach

Olurotimi Richard Akinlofa, Patrik O'Brian Holt, and Eyad Elyan
IDEAS Research Institute, Robert Gordon University, Aberdeen, UK

Cognitive computational modeling is a viable methodology for further investigation of the hitherto inconclusive findings on the cognitive benefits of dynamic versus static visualization components of instructions. This is more so as contemporary cognitive architectures such as the Adaptive Control of Thought–Rational (ACT–R) 6.0 are increasingly applied to traditional cognitive psychology research problems. The application of this methodology is, however, restricted by the limited capability of existing architectures for implementing detailed atomic motor actions such as those involved in complex skill acquisition and performance. This article presents a 2-component computational modeling methodology for investigating the cognitive processes involved in the acquisition and performance of skilled motor tasks. The approach specifies a novel combination of a sequence-of-point technique with a movement control mechanism to implement variously acquired cognitive mental task representations and their intertwined role in postlearning performance as evident in the atomic control of motor actions. This paradigm is validated for 2 experiments using incrementally developed cognitive models developed in ACT–R 6.0. The model's quantitative outputs correlate significantly with equivalent empirical human data. This has implications for multimedia instructional design, especially where rapid, transferrable skill acquisition is desired on initial exposure.

1. INTRODUCTION

1.1. Visualizations in Instructional Delivery

Despite a plethora of previous research, the cognitive benefits of different visualization components in instructional interfaces are not yet fully understood (see Höffler & Leutner, 2007, for a meta-analysis). Static visualizations like diagrams and pictures are thought to improve cognitive germane processing—the memory capacity required for effective learning (Mayer, Hegarty, Mayer, & Campbell, 2005)—and mental rotation—inferring motion through mental simulation (Hegarty, 2004, 2005), which makes them more effective for long-term retention of acquired skills. Mayer et al. (2005) further suggested that dynamic instructional visualizations, such as videos and animations, may inhibit comprehension by imposing excessive extraneous cognitive loads on the learner. In contrast, other studies have argued that dynamic instructional visualizations are more effective for learning certain skills because they aid the creation of more accurate mental task models by the learner (Akinlofa, Holt, & Elyan, 2013b; Arguel & Jamet, 2009; Wong et al., 2009).

Höffler and Leutner's (2007) meta-analysis of 76 comparative studies found an overall positive advantage of dynamic instructional visualizations over static alternatives with a significant mean effect size of $d = 0.37$, 95% confidence interval (CI) [0.25, 0.49]. More important, the meta-analysis also concluded that several variables moderate the cognitive benefits of instructional visualizations such as the role of the visualization (decorational vs. representational), the learner's prior knowledge and spatial ability and the type of requested knowledge (procedural motor, declarative and problem solving). Further studies have extended from this to investigate the knowledge domain moderator variable focusing on the acquisition of procedural motor skills by novices (Akinlofa et al., 2013b; Wong et al., 2009). Wong et al.'s (2009) approach distinguishes biologically primary and secondary knowledge domains (see Geary, 2007, and proposed a working memory *processor* structure to explain the cognitive benefit of dynamic instructional visualizations over statics. Akinlofa et al. (2013b) extended this model to include interactive dynamic visualizations and control for the learner's spatial abilities with consistent result. Akinlofa, Holt, and Elyan (2013a) further suggested that the benefit of dynamic instructions may be independent of domain expertise to the extent that the current learning tasks are novel. These subsequent studies, in general, are consistent with the findings of Höffler and Leutner's (2007) meta-analysis. More important, however, and of particular relevance to the current study, they also suggest a low-level intertwined role of atomic cognitive processes in postlearning task performance, which is moderated by the visualization component of the instruction.

In this article, we report two experiments that use a cognitive modeling approach to investigate this moderating effect of instructional visualizations on procedural motor skill

Address correspondence to Olurotimi Richard Akinlofa, IDEAS Research Institute, Robert Gordon University, Riverside East, Garthdee Road, Aberdeen AB10 7GJ, United Kingdom. E-mail: o.r.akinlofa@rgu.ac.uk

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/hihc.

acquisition and postlearning task performance. We begin with a review of previous related research that applies computational modeling techniques to human–computer interaction problems in a procedural knowledge domain. We then describe in detail a novel methodology for modeling atomic-level integration of cognitive processing and continuous motor execution based on mental task representations afforded by different instructional visualizations. We validate this method through comparative analysis and application to postlearning procedural motor performance measures of human participants from two previous related studies.

1.2. Modeling Skill Acquisition in a Cognitive Architecture

Computational modeling with cognitive architectures is increasingly becoming a methodology of choice for many human factors studies. Cognitive architectures are general frameworks that afford computational modeling of human behavior and cognitive performance. Some examples of widely accepted cognitive architectures include EPIC (Kieras & Meyer, 1997), SOAR (Laird, Newell, & Rosenbloom, 1987), and Adaptive Control of Thought–Rational (ACT-R; Anderson, 2005; Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004). These architectures have also been combined in other studies for modeling complex multitask execution (Bi, Gan, & Liu, in press; Liu, 2009). Cognitive architectures capture the capabilities and limitations of human cognitive and behavioral performance including perception, memory, and motor processes. By specifying these limitations and capabilities, cognitive architectures afford the implementation of computational behavioral models that are psychologically valid and compare well with actual human performances. The recent upsurge in the use of these architectures may be due to their increasing sophistication and applicability to a wider range of traditional human factors research problems. Comprehensive cognitive modeling architectures have also enabled an integrated theoretical approach to human factors research as opposed to the traditional paradigms that tend to explain separate aspects of human cognition only. The need for such a comprehensive theoretical framework of cognition has long been recognized in cognitive science as expressed succinctly by Newell (1990):

> If a theory covers only one part or component, it flirts with trouble from the start. It goes without saying that there are dissociations, independencies, impenetrabilities, and modularities. These all help to break the web of each bit of behavior being shaped by an unlimited set of antecedents. So they are important to understand and help to make that theory simple enough to use. But they don't remove the necessity of a theory that provides the total picture and explains the role of the parts and why they exist. (pp. 17–18)

Despite the increasing success of applying computational cognitive modeling to several traditional human factors problems, however, the available cognitive architectures still lack functionalities for modeling more complex task performance scenarios such as the acquisition and performance of skilled and continuous human motor action. Existing cognitive architectures, such as ACT–R, have only rudimentary capabilities for modeling motor performance. As such, they are not readily capable of modeling the fine movements involved in skilled human motor performance, because such tasks are difficult. The modeling task is further compounded by the seemingly infinite degrees of movements possible in skilled motor performance coupled with the human ability to execute the required movement almost effortlessly (Viviani & Flash, 1995). Computational modeling using a cognitive architecture has been applied to a wide range of human behavioral tasks in general, but there are relatively few previous studies that have modeled the atomic cognitive processes involved in human motor skill acquisition and performance. Modeling this category of knowledge domain involves not only the integration of percepts to create mental task representations but also specifying in detail the intertwined role of these mental models and the cognitive processes that decompile them in moderating subsequent task performance. An example of a relevant research effort is Kieras, Meyer, Ballas, and Lauber's (2000) computational modeling of Martin-Emerson and Wickens's (1992) manual motor tracking and choice responses in latency tasks using the EPIC architecture.

In a more recent work, Salvucci (2006) modeled automobile driving task using the ACT–R architecture. By leveraging the Embodied cognition, Task and Artefact framework, Salvucci decomposed the driving task to a set of basic tasks (control, monitoring, and decision making) that are subsequently integrated to accomplish the overall driving task. In particular, the control component captures all the motor actions that are associated with safe navigation during driving including manipulative lateral (steering) and longitudinal control (acceleration and braking). Salvucci's implementation of these actions, however, was high level and did not include the detailed integration of the mental task representation with the atomic motor processes. For instance, Salvucci simulated lateral control by integrating feedback from a 2-point shifting visual attention model into a specified control equation that determines the degree of steering correction required to maintain safe navigation. There was no specification, however, of the detailed cognitive processes, which is integrated with low-level motor actions to effect the steering control. As such, Salvucci's driver model did not account for the moderating role of mental task representations on the continuous motor control actions that effect the steering. Furthermore, Salvucci's model does not account for how these mental representations were acquired in the first instance or the effect, if any, of different acquisition paradigms on subsequent motor performance.

In a more recent work, Byrne et al. (2010) modeled the fine manual control involved in a motor task. The task involved controlling a coupled disk configuration to hit two targets at the ends of a linear trajectory as described in Huegel, Celik, Israr, and O'Malley (2009). Byrne et al. (2010) made three key modifications to the base ACT–R cognitive modeling architecture to achieve the atomic manual control required for the smooth movements involved in the task. First, they increased the update rate of motor output location from 50 ms to 3 ms. Second, they modified the velocity profile of the movement using the "minimum jerk" paradigm of Hogan (1984). Last, they utilized ACT–R's imaginal module to present intermediate virtual target markers to the motor module along the movement trajectory. These modifications enabled the modeling of the smoother, continuous movements involved in the task than can be afforded by the base ACT–R cognitive architecture. However, Byrne et al.'s (2010) model does not account for the prior acquisition of cognitive mental task representations or its intertwined role in subsequent postlearning motor control/performance. Most notably, their model uses the imaginal module for intermediate virtual target locations along the trajectory but does not specify how these intermediate locations are initially acquired or determined. This is very crucial for trajectory validation processes that are evident in postlearning task performance of acquired motor skills especially in mechanical component manipulation for assembly/disassembly.

Akinlofa et al. (2013b) argued that learners create cognitive mental task representations in the acquisition of motor skills, and these representations are implicated in the subsequent postlearning performance of such motor tasks. Furthermore, they observed that dynamic instructional visualizations afford the creation of more accurate mental task representations and arguably lead to better postlearning task performance than equivalent static visualizations. This cognitive benefit of dynamic instructional visualizations over static equivalents has been shown to be dependent on the knowledge domain (Höffler & Leutner, 2007) and independent of the learner's expertise and spatial abilities (Akinlofa et al., 2013a). In the current study, we propose a novel sequence-of-point computational modeling approach to investigate the atomic cognitive processes involved in learning a novel skill and how these are integrated with subsequent postlearning executive motor actions that drives task performance in a motor knowledge domain. Similar to Byrne et al. (2010), we modify certain aspects of the ACT–R cognitive architecture for our modeling purposes. Our method, however, differentiates between mental task representations acquired from dynamic versus static instructional visualizations. It further specifies a detailed validation process for intermediate points along the movement trajectory that reflects the controlling role of the different cognitive mental task representations in postlearning skilled motor performance.

1.3. The ACT–R Cognitive Architecture

ACT–R 6.0 is the architecture of choice for this study because of its advanced and modular implementation, which is easily extensible. ACT–R is a theory of human cognition that assumes a distinctive categorization of knowledge structures as declarative and procedural (Anderson et al., 2004). Declarative knowledge is composed of logical units or chunks that encode facts such as $1 + 3 = 4$ or target object "a" is at Cartesian coordinate (4, 10) in a reference plane. Procedural knowledge, on the other hand, consists of condition-action rules that manipulate declarative knowledge and external percept. The ACT–R theory is implemented as a hybrid cognitive architecture based on a symbolic central production system influenced by massively parallel subsymbolic processes, which are represented by a set of mathematical equations. The symbolic structure consists of a set of modules for processing different kinds of information, which are interfaced through a central production system by their matching buffers. The modules operate in parallel through internal subsymbolic processes and communicate through the information deposited in their buffers. The central production system coordinates the behavior of these modules by recognizing patterns in their buffers and making requested changes. Our modeling technique leverages the extensibility of the ACT–R architecture by extensive modifications to the motor and imaginal modules. This allows us to implement complex protocols that translate cognitive mental task representations into smoothly executed motor movements that simulate a mechanical assembly task. We have also relied on the versatile chunk activation processes of the declarative module, especially the partial matching retrieval mechanism, to simulate the noise inherent in smooth manipulative movements and enable robust motor performance despite the potentially infinite degrees of movement freedom possible. The selected ACT–R 6.0 version for the modeling effort contains all the components of previous ACT–R versions including ACT–R/PM (ACT–R 6.0 Reference Manual, p. 256).

2. EXPERIMENT 1

2.1. The Task

In Experiment 1, we modeled a subset of the task and data reported in Akinlofa et al. (2013b). The study compared the postlearning task performance of two independent groups that learned a mechanical disassembly task through instructional interfaces with either dynamic (*V-group*) or static (*S-group*) visualization components. Although the disassembly process involves 11 logical and sequential steps, only the fifth step of the process was analyzed for computational modeling. This step involves the rotation of the chassis of the model used in the experiment through π radians to access a component located underneath it as depicted in Figure 1. It was selected for computational modeling because it highlights the differential skills acquisition rate possible via the different instructional interface types. It is also a good example of the abstract and stochastic cognitive processing that results in observable skilled motor action. In addition, it reduces the scope of work for the initial proof of concept modeling and avoids the substantial effort that would be required to model the entire task sequence at an early stage of the work.

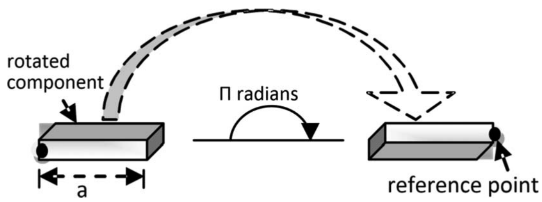

FIG. 1. Trajectory of the manipulated component.

2.2. Methods

Movement analysis and strategies. A kinematic analysis of the video data from Akinlofa et al. (2013b) was conducted in slow motion to extract the time taken by each participant to execute the selected step of disassembly. Based on the biomechanical human movement research of Hamil and Knutzen (2003), a reference point was selected on the rotated component to represent the sum total of manipulations as shown in Figure 1 and the time taken to pass through the *mid*points and *end*points of the ideal semicircular trajectory were recorded. The accuracy of the component manipulation was also recorded as an alignment of the reference point to the required path as it transits through the midpoint of the trajectory. Raw data of the kinematic analysis are detailed in Table 1. As evident from the data, the longest time observed for completion of the rotation was 16 s (Participants 121 & 124). A cutoff time of 17 s was therefore used in the computational modeling of this step as the criterion to determine successful component manipulation. As it is infinitely possible to achieve the component manipulation through stochastic processes, this cutoff time was also adopted for subsequent comparative performance analysis of data from the human participants and equivalent computational model outputs. Discrepancies in the scores were resolved through consensus by three independent assessors.

The movement analysis show that two broad strategies were at play. The first is a stochastic sequence of multidirectional movement observed mostly in the *S-group* participants. This group was presented with only two pictures showing the initial and final states of the manipulated component. They therefore lacked declarative knowledge of all the transitory intermediate states of component manipulation. The second strategy is a combination of the first with a more directed movement along the desired trajectory aided by declarative recall. This hybrid strategy featured prominently in the improved performance of the *V-group* as they had acquired the declarative knowledge of the initial and final component states as well as all intermediate transitory manipulations by watching a video clip of the executed step being performed by a skilled expert. Further detailed analysis shows that different performance protocols were applied at various quadrants of the motor movement as depicted in Figure 2. In the early stages, there is a tendency to initiate a randomly directed movement in the general direction of the perceived end state of the manipulated component. This rapidly changes to a search space in all directions

TABLE 1
Raw Scores of Kinematic Analysis—Experiment 1

ID	Time to 2nd Quadrant	Time to Midpoint	Time to Endpoint	Interface Type	Completed Rotation
045				1	2
049	35			1	2
052	4	9	12	1	1
055				1	2
058				1	2
061				1	2
064				1	2
067		13	15	1	1
070				1	2
073		5	7	1	1
076				1	2
079	19			1	2
082	3			1	2
085				1	2
088		10	12	1	1
091				1	2
094				1	2
097	3	9	12	1	1
100		8	12	1	1
103		12	14	1	1
106	3			1	2
109		3	6	1	1
112				1	2
115		6	8	1	1
118		4	6	1	1
121	2	14	16	1	1
124	7	13	16	1	1
127	11			1	2
130	13			1	2
133		3	5	1	1
047	6	7	9	2	1
048	1	2	4	2	1
051	3	5	6	2	1
054	2	3	5	2	1
057	1	3	5	2	1
060	1	2	4	2	1
063		3	4	2	1
066		1	3	2	1
069	3	5	8	2	1
072	3	4	5	2	1
075	3	4	5	2	1
078		2	4	2	1
081	2	3	5	2	1
084		1	3	2	1
087		1	3	2	1
090		3	4	2	1
093	1	2	3	2	1

(*Continued*)

TABLE 1
(Continued)

ID	Time to 2nd Quadrant	Time to Midpoint	Time to Endpoint	Interface Type	Completed Rotation
096	2	3	5	2	1
099		4	5	2	1
102		3	5	2	1
105		4	6	2	1
108		2	4	2	1
111		3	6	2	1
114	2	3	4	2	1
117		2	3	2	1
120	2	3	5	2	1
123	4	8	9	2	1
126		2	3	2	1
129	1	7	8	2	1

within the second quadrant where most of the failures were recorded. However, once successfully past the midpoint, subsequent movement converges rapidly to the endpoint of the trajectory.

It was further observed that despite the stochasticity of the motor movements at all stages of the trajectory, participants were able to determine when a sequence of random manipulations have sufficiently deviated so as not to satisfy the possible range of configurations for the initial and end positions of the manipulated component. In such instances, they attempt correctional movements to align with the trajectory or if sufficiently deviated, the attempt is aborted and the disassembly tasked terminated.

Modeling continuous motor action—The sequence-of-points technique. Two fundamental problems were posed by the computational modeling of the selected disassembly step. The first was to execute continuous motor actions required to rotate the component from the start to the endpoint of the semicircular ideal trajectory. The second problem was to integrate underlying cognitive processing outputs that adjust the motor movements to align with the participant's mental task model of the task as acquired through the different instructional interfaces.

For the first problem, the ACT–R architecture includes a motor module that specifies default mechanisms for modeling a range of motor movements such as typing and mouse movements. These default mechanisms, however, were not suitable for our modeling purposes for certain reasons. For instance aimed movements, such as pointing with the mouse, are executed by calculating the movement execution time based on Fitts's Law (Fitts, 1954) and updating the cursor location when the simulated duration has elapsed. The computations involved assume that the movement is made toward a target and specify fixed start and end cursor locations. Our model's movement strategy, however, specifies only the cursor start location with the end location dependent on underlying stochastic cognitive processes. To resolve this, we assumed a reference point, as shown in Figure 1, through which all resolved component manipulation forces act (see Hamil & Knutzen, 2003). The default ACT–R motor module was then modified to simulate the movement of this reference point as sequences of fixed magnitude, variable direction unit vectors. The start location of each unit movement vector corresponds to the end location of the previous vector. The end locations, however, are determined through a separate process to reflect the random output of the underlying stochastic cognitive processes. There was still a problem, as the default ACT–R motor module also assumes that aimed movements start and end with zero velocity. In addition, we had fixed the magnitude of the unit movement vectors at approximately 50 ms to be consistent with previous related research (Meyer & Kieras, 1997; Salvucci & Gray, 2004). This resulted in a jerky movement output that was very coarse.

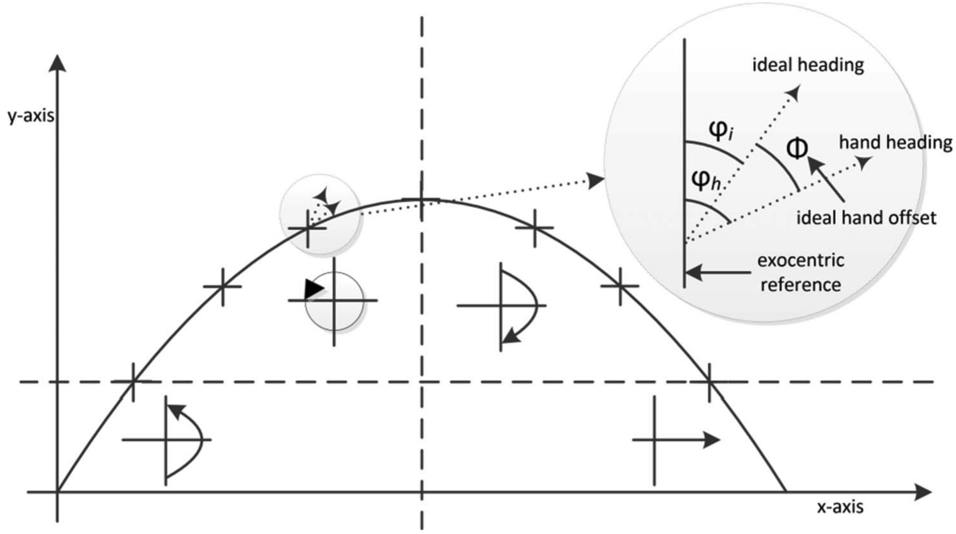

FIG. 2. Kinematic analysis of manipulative motor movements.

Therefore, we adapted the movement velocity profile at the transitional boundaries between the unit movement vectors based on the dynamic cost optimization approach for the mathematical modeling of human hand movements (Flash & Hogan, 1985), using the minimization of the time integral of the square of jerk. According to Flash and Hogan, the location of a reference point at any time t along a straight line trajectory starting and ending with zero velocity is described by Equation 1:

$$x(t) = x_0 + (x_0 - x_f)(15\tau^4 - 6\tau^5 - 10\tau^3)$$

$$y(t) = y_0 + (y_0 - y_f)(15\tau^4 - 6\tau^5 - 10\tau^3)$$

$$\text{where}\,\tau = t/t_f\,,$$

x_o and y_o are initial hand position coordinates ($t = 0$) and

x_o and y_f are final hand position coordinates ($t = t_f$). (1)

For curved point-to-point movement, the equation is redefined to include intermediate points (at times $t_1, t_2, \ldots, t_n$) inserted between the start and end positions as shown in Equation 2. We adapted this equation for curved point-to-point movements by using a shifting boundary technique bound by $t = 0$ and $t = t_f$ across the set of movement vectors transition points to accurately implement continuous acceleration throughout the movement trajectory.

for all times $t \leq t_n$

$$x^-(\tau) = \frac{t_f^5}{720}(\mu_x(\tau_n^4\left(15\tau^4 - 30\tau^3\right) + \tau_n^3(80\tau^3 - 30\tau^4) - 60\tau^3\tau_n^2 + 30\tau^4\tau_n - 6\tau^5) + c_x(15\tau^4 - 10\tau^3 - 6\tau^5)) + x_0$$

$$y^-(\tau) = \frac{t_f^5}{720}(\mu_y(\tau_n^4\left(15\tau^4 - 30\tau^3\right) + \tau_n^3(80\tau^3 - 30\tau^4) - 60\tau^3\tau_n^2 + 30\tau^4\tau_n - 6\tau^5) + c_y(15\tau^4 - 10\tau^3 - 6\tau^5)) + y_0$$

and for all times $t \geq t_n$

$$x^+(\tau) = \frac{t_f^5}{720}(\mu_x(\tau_n^4\left(15\tau^4 - 30\tau^3 + 30\tau - 15\right) + \tau_n^3(-30\tau^4 + 80\tau^3 - 60\tau^2 + 10)) + c_y(-6\tau^5 + 15\tau^4 - 10\tau^3 + 1)) + x_f = x^-(\tau) + \frac{\mu_x t_f^5(\tau - \tau_n)^5}{120}$$

$$y^+(\tau) = \frac{t_f^5}{720}(\mu_y(\tau_n^4\left(15\tau^4 - 30\tau^3 + 30\tau - 15\right) + \tau_n^3(-30\tau^4 + 80\tau^3 - 60\tau^2 + 10)) + c_y(-6\tau^5 + 15\tau^4 - 10\tau^3 + 1)) + y_f = y^-(\tau) + \frac{\mu_y t_f^5(\tau - \tau_n)^5}{120}$$

where $\tau = t/t_f; \tau_n = t_n/t_f$; t_n is a via – point; μ_x, μ_y, c_x,

and c_y are *cons* tan *ts*. (2)

The number of unit movement vectors in a movement sequence as well as their individual directions is, however, stochastically dependent on the current position in the trajectory and the selected productions firing per cycle of cognitive processing. By adopting this synergistic paradigm, we were able to implement the observed ability of the human participants to select and execute a required movement despite the seemingly infinite degrees of possible movement.

The second problem was more important because it is linked directly to the objective of the research, which was to investigate how the different resultant mental task models of the instructional interfaces drive postlearning motor performance. It was observed from the kinematic analysis that despite the stochasticity of the motor actions involved, participants were able to determine when a particular sequence of movements has become so inconsistent with the ideal rotation trajectory that successful manipulation of the component is no longer possible. This tacit ability suggests that participants acquire a mental model of the rotational task during learning, which moderates the subsequent task performance. Furthermore, it is significantly differentiated in the postlearning performances of the compared groups with the *V-group* showing a more robust performance than the *S-group*. We model it as a motor control law that adapts Fajen and Warren's (2003) model of the behavioral dynamics of steering as shown in Equation 3:

$$\varphi_h = -k_i(\varphi_h - \varphi_i) = k_i\Phi \qquad (3)$$

where φ_h is the direction of the heading, φ_i is the direction of the target and

At the end of each unit vector execution of the movement sequence as depicted in Figure 2, the model determines the extent of trajectory deviation by comparing the location of the reference point with its mental task model. The ideal component trajectory is defined by a separate, hidden process and used as a heuristic function to moderate the comparison.

Deviation determination and the magnitude of corrective action required is controlled by setting parameters k_i and Φ, which determines attractiveness of the ideal trajectory heuristic and the actionable threshold for remedial steering, respectively. The motor control law provides the mechanism to execute corrective motor actions for component manipulation only and the same magnitude of the parameters k_i and Φ were set for both the *S-group* (*S-model*) and *V-group* (*V-model*) representations. The

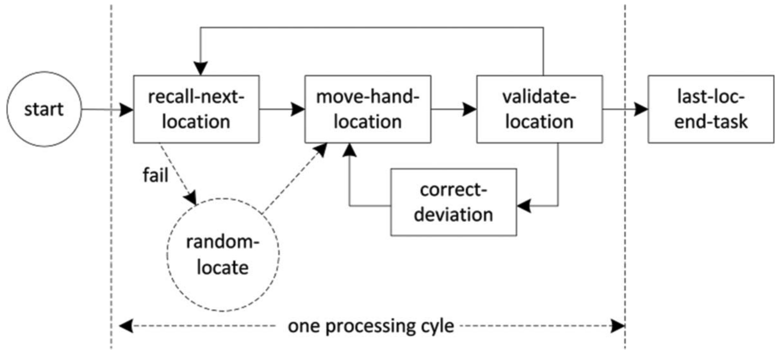

FIG. 3. Schematic model's productions—Experiment 1.

task performance is therefore dependent on the different mental task representations of the compared groups only.

ACT–R implementation. The core model productions are shown in the schematic diagram in Figure 3. The structure of the productions algorithm is essentially the same for the *S-model* and *V-model* implementation. The only differences are in the implementation of the declarative mental task representations and how this moderates subsequent task performance. The movement trajectory and spatial locations of the rotated component are defined in the 2-D Cartesian coordinate "where system" specified in the ACT–R's vision module. The *S-model* starts with a declarative knowledge of only the initial and final positions of the rotated component as corresponding to viewing static visualizations of these stages of the assembly. The *V-model*'s declarative knowledge structure, however, includes both the initial and final component positions as well as all the intermediate locations of the movement reference points along the rotation trajectory, which corresponds to interacting with dynamic instruction visualizations such as a video playback of expert execution of the procedure.

In the *S-model*, a top-level goal attempts to retrieve the next movement location for the rotated component's reference point after the start position. The retrieval fails as its declarative knowledge does not include this location and it reverts to a random location determination strategy. This random location determination is limited by Equation 2 as well as the restriction of the search space specifications of the trajectory quadrant (see Figure 2). When a random location is returned, the move-hand-to-location production fires to move the selected reference point to that location and simulate hand movement. The location is then validated against the model's internal representation of the task acquired during the learning phase. If the spatial location is validated, the cycle is repeated by firing subsequent productions that attempts further failed retrievals and reversion to the random location determination strategy. However, if the location is determined to have sufficiently deviated, a corrective process is activated to restrict the search space for further random location determination. The actionable deviation threshold and search space restriction is controlled by the parameter Φ whereas the magnitude of the correctional movement is determined by the parameter k_i. The corrective process terminates once the trajectory deviation is reduced below the minimal threshold Φ and the model reverts back to the retrieve-fail/random-locate strategy with further location validations. The productions cycle repeats until a specified cutoff time is exceeded or the last-loc-end-task production is fired to report a validated spatial location within a specified range of the end-position of the rotated component.

The internal task representation of the *V-model* is different from that of the *S-model* because it includes additional knowledge of the intermediate spatial locations between the start and endpoints of the component rotation. Its top-level goal retrieval attempt is therefore more likely to be successful and the rotated component's reference point is moved directly to the retrieved spatial location. The partial matching mechanism of ACT–R's retrieval module was utilized to simulate the inaccuracy of recalling component's intermediate positions along the trajectory of rotation. As the model movement is implemented in 2D Cartesian space, a sim-hook function was implemented to define matching inaccuracies on the x-coordinates. An extension of the activation equation was further used to define matching inaccuracies on the y-coordinate and a summation of the matching functions outputs was computed as the overall match score of a specific location in the movement space. This design, as depicted in Figure 4, is very flexible and could be a starting point for extending to 3D spatial movements. If the retrieval is successful, a production is fired to move the hand to the recalled location followed by a validation process similar to that for the *S-model* as just outlined. If the retrieval fails, the model reverts to the random-locate strategy used by the *S-model*. The *V-model* therefore implements the hybrid strategy of task performance as determined from the kinematic analysis of the human participant's movements. A validated spatial location could trigger the correct-deviation process to bring it within the minimum deviation threshold before another retrieval attempt is fired. The production cycle of the *V-model* is also terminated if a specified cutoff time is exceeded or when the end of the trajectory is reported.

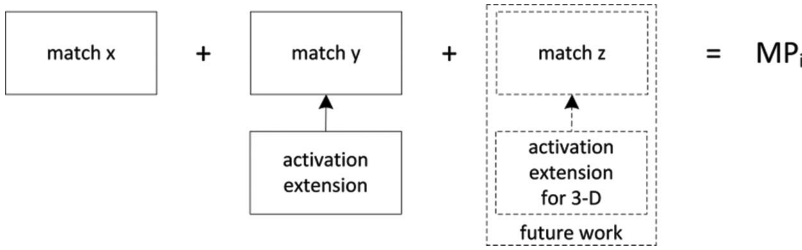

FIG. 4. Random spatial location matching design.

2.3. Model Validation

Model strategies and performance were validated by comparative analysis with empirical test data from Akinlofa et al. (2013b). Model and human data were analyzed in the same manner to generate directly comparable and more reliable performance measures. The human data was split into Development ($n = 28$) and Test ($n = 59$) for analysis. The model's parameters were refined with development data and validated with the test data. Most of the ACT–R architecture parameters were kept at their default settings with the exception of the base-level constant, which was set to 5.0 to reflect the recency of acquisition of the declarative knowledge through interaction with the task instructions. The *transient noise* and *mismatch penalty* parameters were also activated with values 0.2 and 1.0, respectively. The domain-specific parameters k_i and Φ were initially set to reasonable values and then refined for qualitative and quantitative fit to the development data. Similar final values were estimated for the two models as detailed in Table 2.

TABLE 2
Domain-Specific Parameters of the Model

Parameter	Description	Value
k_i	Ideal trajectory attractiveness	1.0
Φ	Actionable deviation threshold	2.0
cutoff	Model run-time limit (seconds)	17.0

The mean task execution time and trajectory alignment rate for the human data and 500 runs of the ACT–R models are reported in Table 3. The model's quantitative predictions were very accurate on the performance measures of time to mid-trajectory ($R^2 = .98$, RMSE $= .52$), end-trajectory ($R^2 = .98$, RMSE $= .56$) and trajectory tracking (see Table 2). Independent-samples t tests were also conducted for paired comparison of human and model data. The results, as detailed in Table 4, replicated the significant differences observed between the *S-group* and *V-group* in the empirical data. Furthermore, no significant differences were found in within-group comparison of human and model performance measures.

2.4. Discussion

We model a single step of the experimental task from Akinlofa et al. (2013b) in the ACT–R cognitive architecture by using a novel sequence-of-points technique. The computational model implements similar productions structure for the two independent groups compared—static pictures versus video task instructions. The declarative knowledge structures were different, however, to reflect interaction with the respective static and dynamic visualizations components of the instructional interfaces. The model's quantitative predictions on postlearning task performance were accurate and replicated the significant differences observed in the human data from the original study. This reinforces the argument that dynamic instructional visualizations may be more cognitively beneficial than static equivalents for the acquisition of procedural motor knowledge. Our results are limited, however, as only a single

TABLE 3
Descriptive Statistics for Human and Model Performance Measures—Experiment 1

		Midpoint		Endpoint		Trajectory (%)	
Category	n	M	SD	M	SD	Completed	Aligned
S-human	30	8.39	3.93	10.77	3.96	43.3	33.33
S-model	30	8.56	3.48	10.55	3.6	40.0	23.33
S-model (500)	500	8.85	3.35	10.89	3.36	43.6	40.6
V-human	29	3.28	1.75	4.93	1.73	100	100
V-model	29	3.2	.49	5.35	.7	100	100
V-model (500)	500	2.85	.44	4.8	.61	100	100

TABLE 4
Comparative Analysis of Human and Model Data—Experiment 1

	Time to Midpoint						Time to Endpoint					
Paired Categories	*t (df)*	*p (two-tailed)*	η^2	*M Difference*	*95%*	*CI*	*t (df)*	*p (two-tailed)*	η^2	*M Difference*	*95%*	*CI*
S-model V-model	5.31 (11.18)	<.01	.42	5.36	3.14	7.57	5.0 (11.34)	<.01	.39	5.19	2.89	7.49
S-human S-model	−.12 (23)	.91	<.01	−.17	−3.26	2.91	.15 (23)	.88	<.01	.22	−2.92	3.37
V-human V-model	.22 (32.38)	.83	<.01	.07	−.61	.76	−1.22 (37)	.23	<.01	−.42	−1.13	.28

step of an entire assembly sequence was modeled. A more complete comparison will include the entire assembly sequence of procedural motor tasks. This limitation is addressed in a follow-up experiment based on the sequence-of-point modeling paradigm in the ACT–R architecture.

3. EXPERIMENT 2

3.1. The Task

The objective of Experiment 2 was to extend the sequence-of-point modeling methodology to an entire sequence of procedural-motor task. We apply the method to model the experimental task of Watson, Butterfield, Curran, and Craig (2010), which compares the effectiveness of dynamic and static computer multimedia instructions for learning a mechanical assembly task. The task was to assemble a device comprising 49 separate parts, which must be put together in a particular sequence. It consists of four progressive stages—central gear assembly, frame, propeller and crank arm. Participants were independently grouped by three instructional interfaces—animated video, static diagrams and text—and completed one postlearning assembly task per day for 5 consecutive days. Task performance of the independent groups was compared on the factors of device assembly time and errors. Watson et al. (2010) data describes the immediate postlearning performance effect on the first build as well as long-term retention and performance convergence for the three compared groups over five builds. Our modeling effort however is limited to the early stages of performance for the animated video (*dynamic or V-group*) and static diagram (*static or S-group*) instruction groups only. The performance of the text group was not modeled as it is not relevant to the objective of the experiment. Furthermore, only the first postlearning build for the *V-group* and *S-group* were modeled as our objective was to compare the performance effect of the mental task representations afforded by the different instructional visualizations and not long term retention or performance convergence. The methodology of Watson et al. also allowed for continuous reference to the instructions during the task execution and their subsequent data analysis separated the reference time from the actual build time. In contrast, our modeling technique assumes a single interaction with the instructions with no further references during the task execution. Last, due to the restrictions imposed by the 2D visual reference framework of the ACT–R architecture, the assembly of nine components whose trajectories were orthogonal to the main plane of assembly was not modeled (see Figure 5).

3.2. Methods

Movement analysis and sequencing. The trajectories of the assembled components were analyzed as linear movements between specific start and end points in a 2D Cartesian reference plane (see Figure 5). The trajectories were grouped into four categories based on the direction of movement from the start to the endpoints—right, left, up, or down within the Cartesian reference framework. The assembly starts with the central rod in place and the components are progressively attached in the sequence specified in Table 5 (component C2 to E27) until the task is completed.

Extending the sequence-of-points technique. The model's production system, as shown in Figure 6, is essentially the same as that for Experiment 1 with additional mechanisms to switch to the next component in the sequence or reset a failed assembly attempt. The next-component production is fired when the reference point of the component being assembled is within specified limits of its trajectory endpoint. A component's assembly attempt may also be reset if the movements have substantially deviated from the ideal assembly trajectory that successful coupling is no longer possible. The reset mechanism allows the model to retry the assembly of such components in the same manner as observed in the analysis of equivalent human performance data. The main differences between the representative *S-model* and *V-model* was in the declarative mental task knowledge structures as applicable in Experiment 1.

The *S-model*'s mental task representation includes only the start and end spatial locations of each component's assembly trajectory, which corresponds to viewing static pictures of the components in such configurations. It utilizes the same retrieve-fail/random-locate strategy as its equivalent representation in Experiment 1 and uses the same control process to correct deviations to the assembly trajectory. The *V-model*'s mental

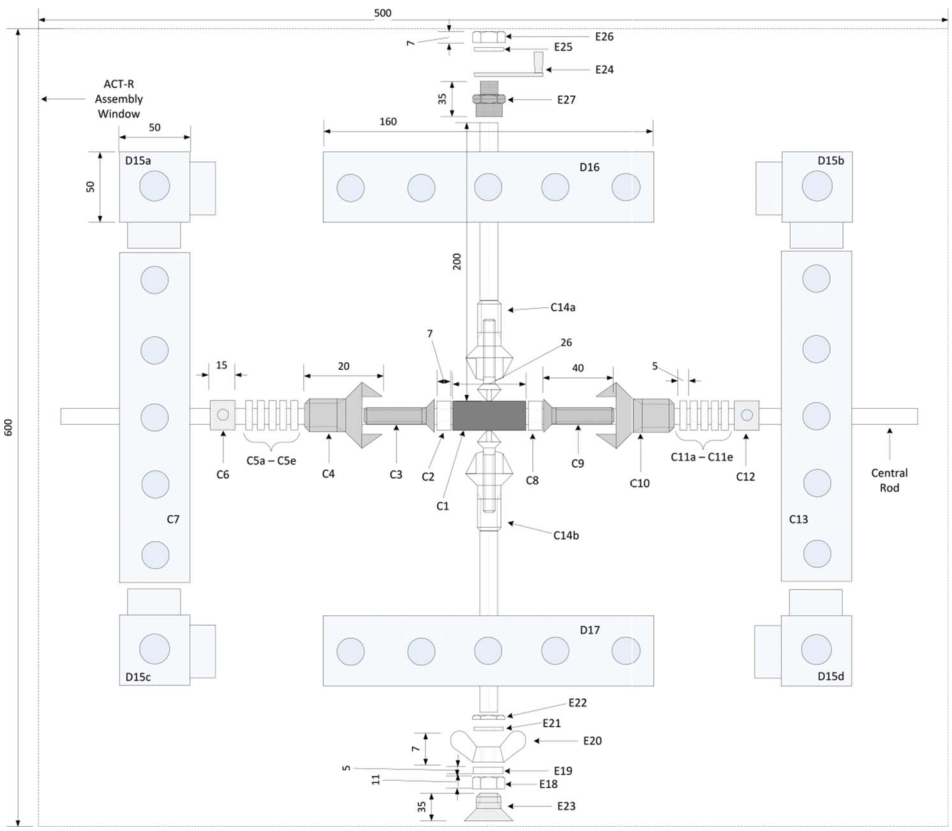

FIG. 5. Kinematic analysis of assembly motor actions.

task representation includes knowledge of the start and end locations as well as all intermediate spatial locations of the assembly sequence corresponding to learning from dynamic instructional visualizations. It utilizes the hybrid strategy as described in Experiment 1, which combines intermediate location retrieval attempts with the random-locate mechanism when retrieval fails. ACT–R's partial matching mechanism is also used to simulate retrieved location inaccuracies as described in Experiment 1 (see Figure 4).

3.3. Model Validation

The mean assembly times (in seconds) for 100 runs each of the *S-model* and *V-model* and the corresponding data from human participants (Watson et al., 2010) are shown in Table 6. The table also shows data for 10 runs each of the cognitive models groups (*S-model [10]* &*V-model [10]*) for direct comparison with the equivalent sample size of human participants from Watson et al. (2010) study. The human data do not include timings for the substages of the assembly, and comparison with model data was therefore limited to the final build times only. In addition, the sample size for human participants (Watson et al., 2010) was small, which may account for the large deviations reported in that study. Despite this, the reported human data clearly show the trend of learning differences and interface effectiveness between the compared groups. The Animation group (*dynamic*) recorded considerably lower deviation than the Diagram (*static*) group, indicating more consistent superior performance. This decreasing trend in performance time was also replicated in the models' data. Of interest, correspondingly large standard deviations were observed in only the *S-model [10]* and *V-model [10]* group's data with more consistent deviations recorded for the 100-runs of model data. This may imply that the larger sample size of the 100-runs model groups afforded a more consistent measurement of task performance. The *S-model [10]* and *V-model [10]* group's data were excluded from subsequent analysis as the equivalent raw data of human participants from Watson et al.'s (2010) study were not provided when requested.

The ACT–R architecture and task domain parameters settings from Experiment 1 were retained with the exception that no cutoff time was set for the task. The cutoff criterion was not required as the task was to complete the entire assembly and not a substep. The model's quantitative data were analyzed with similar parametric statistical tests to those used in the original study by Watson et al. (2010). An independent samples t test revealed that the *V-model*'s mean task performance time ($M = 515.5$, $SD = 75.0$) was significantly faster than the *S-model* ($M = 682.8$, $SD = 33.8$), $t(198) = 20.4$, $p = .0$, two-tailed. The magnitude of the differences in the means was very large

TABLE 5
Decomposition of Assembly Movements Within a 2D Cartesian Reference Framework

Serial	Code	Component	Thickness (Units)	Start	End	Trajectory
1.	C1	Spacer Ring (on long central rod)	26	0,300	250,300	Right
2.	C2	Left Metal Washer	7	0,300	237,300	Right
3.	C3	Left Gripping Screw	40	0,300	230,300	Right
4.	C4	Left Beveled Gear	20	0,300	230,300	Right
5.	C5a	Left Thin Washer	5	0,300	210,300	Right
6.	C5b	Left Thin Washer	5	0,300	205,300	Right
7.	C5c	Left Thin Washer	5	0,300	200,300	Right
8.	C5d	Left Thin Washer	5	0,300	195,300	Right
9.	C5e	Left Thin Washer	5	0,300	190,300	Right
10.	C6	Left Collar	15	0,300	185,300	Right
11.	C7	Left Beam	50	0,300	170,300	Right
12.	C8	Right Metal Washer	7	500,300	263,300	Left
13.	C9	Right Gripping Screw	40	500,300	270,300	Left
14.	C10	Right Beveled Gear	20	500,300	270,300	Left
15.	C11a	Right Thin Washer	5	500,300	290,300	Left
16.	C11b	Right Thin Washer	5	500,300	295,300	Left
17.	C11c	Right Thin Washer	5	500,300	300,300	Left
18.	C11d	Right Thin Washer	5	500,300	305,300	Left
19.	C11e	Right Thin Washer	5	500,300	310,300	Left
20.	C12	Right Collar	15	500,300	315,300	Left
21.	C13	Right Beam	50	500,300	330,300	Left
22.	C14a	Upper Central Gear Assembly	200	250,0	250,300	Down
23.	C14b	Lower Central Gear Assembly	200	250,600	250,300	Up
24.	D15a	Upper Left Corner Piece	50	0,0	170,0	Right
25.	D15b	Upper Right Corner Piece	50	500,0	330,0	Left
26.	D15c	Lower Left Corner Piece	50	0,600	170,600	Right
27.	D15d	Lower Right Corner Piece	50	500,600	330,600	Left
28.	D16	Upper Beam	50	250,0	250,220	Down
29.	D17	Lower Beam	50	250,600	250,380	Up
30.	E18	Thick Washer	11	250,500	250,600	Down
31.	E19	Thin Washer	5	250,500	250,589	Down
32.	E20	Propeller	7	250,500	250,584	Down
33.	E21	Thin Washer	5	250,500	250,577	Down
34.	E22	Outer Nut	7	250,500	250,572	Down
35.	E23	Gripping Screw	35	250,565	250,465	Up
36.	E24	Crank Arm	8	250,100	250,15	Up
37.	E25	Washer	5	250,100	250,23	Up
38.	E26	Nut	7	250,100	250,28	Up
39.	E27	Part-threaded Nut	35	250,35	250,135	Up
40.	N1	Tightening Screws (not modeled)				
41.	N2	Tightening Screws (not modeled)				
42.	N3	Tightening Screws (not modeled)				
43.	N4	Tightening Screws (not modeled)				
44.	N5	Tightening Screws (not modeled)				
45.	N6	Tightening Screws (not modeled)				
46.	N7	Tightening Screws (not modeled)				
47.	N8	Tightening Screws (not modeled)				
48.	N9	Tightening Screws (not modeled)				
49.	L1	Long Central Rod	Fixed	Fixed	Fixed	Fixed

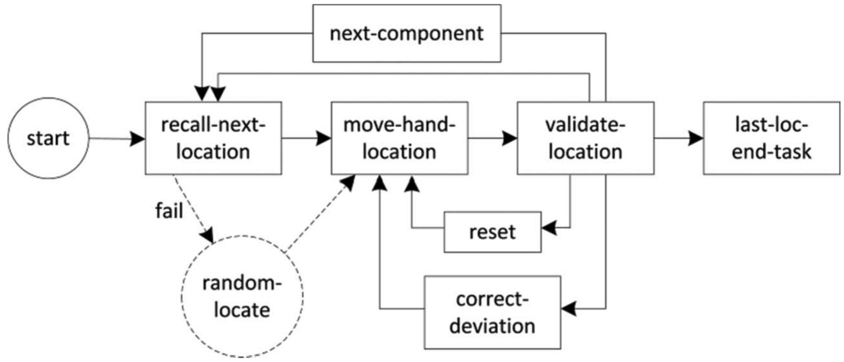

FIG. 6. Schematic model's productions—Experiment 2.

TABLE 6
Mean Assembly Times and Error Counts for Human and Model Performance—Experiment 2

		Central Gear		Frame		Propeller		Crank Arm/Total		Error Counts	
Group	*n*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Diagram (static)	10	—	—	—	—	—	—	710.9	329.0	7 (total)	
S-model	100	524.5	30.1	620.0	32.3	650.6	32.7	682.8	33.8	84.7	13.4
Animation (dynamic)	10	—	—	—	—	—	—	522.6	92.9	1 (total)	
V-model	100	388.6	67.8	484.6	73.8	499.6	74.3	515.5	75.0	1.4	0.6

(M difference $= 167.4$), 95% CI [151.1, 183.6], $\eta^2 = 0.7$. This is partially consistent with the results of Watson et al. (2010), which found a significant effect of instructional group on overall build times with the animation group observed to be 28% faster than the diagram group. Curiously, however, no significant effect of the instructional group was observed for net build times. Watson et al.'s further analysis shows that only the difference between the animation and text instruction groups overall build times was significant (which was not modeled in this study), whereas that for the animation versus diagram group did not reach statistical significance. Only one assembly error was reported in the assembly performance of the animation (*dynamic*) group at Build 1, whereas seven errors were observed for the diagram (*static*) group. The mean error counts for the models, however, were much higher. An independent samples t test revealed that the *S-model* had significantly higher mean error count ($M = 84.7$, $SD = 13.4$) than the *V-model* ($M = 1.4$, $SD = 0.6$), $t(198) = 61.9$, $p = .0$, two-tailed. The magnitude of the differences in the means was very large (M difference $=$ 83.3), 95% CI [80.6, 86.0], $\eta^2 = 0.9$.

3.4. Discussion

We developed a computational model in the ACT–R architecture that replicated the performance of dynamic versus static groups of human participants in a sequential assembly task (Watson et al., 2010). The model utilized the sequence-of-point technique from Experiment 1 for individual component rotation and extended this with further productions to switch to the next component in the sequence when the subassembly was completed. It also included additional mechanisms that simulate component manipulation retrials for failed assembly attempts. The performance of human participants that learned the assembly task through static instructional visualizations was simulated by the model's declarative knowledge that includes chunks of the start and end trajectory positions for each manipulated component (*S-model*). The declarative knowledge of the representative model for participants learning through dynamic instructional visualizations (*V-model*), however, included chunks of the start and end component positions as well as all the intermediate spatial locations along the trajectory of manipulation.

In general, the model's quantitative predictions replicated the trends observed in the equivalent analysis of human data from Watson et al. (2010). However, the analysis of the model's data revealed small differences between the compared groups in contrast to the findings of Watson and his colleagues. An explanation for this could be that the methodology of Watson et al. (2010) was not powerful enough to detect statistically significant differences between the compared groups due to the low samples sizes used. Their data, however, clearly shows the trend of learning differences and interface effectiveness between the compared groups. In contrast, the sample sizes for the model data were much larger (100 model runs for each group), and the subsequent data analysis was powerful enough to replicate the trend in human data as well as detect the differences in the performances of the compared groups.

4. GENERAL DISCUSSION

In a series of two experiments, we applied a novel sequence-of-points method to model the acquisition and execution of skilled, procedural-motor movements in ACT–R 6.0 cognitive architecture. The first experiment of the series was essentially a proof of concept that applies the sequence-of-point approach to a selected single step of the sequential procedural task from Akinlofa et al. (2013b). The modeled step was selected because its performance was significantly moderated by the level of dynamic visualizations components of the instructions for learning it. The second, follow-up experiment extends the modeling methodology to an entire task sequence from Watson et al. (2010) to overcome the limitation of the first experiment. Model data from both experiments were validated with equivalent empirical human data from the related studies with significantly accurate quantitative prediction outcomes.

The sequence-of-points method successfully addresses two key problems associated with modeling the acquisition of skilled human motor performance—the smooth execution of continuous movements along curved and linear trajectories and the simulation of the cognitive roles of different mental task representations in postlearning task performance. The first problem is a long-recognized constraint in the computational cognitive modeling of human motor performance. Most modern cognitive architectures have only rudimentary mechanisms for simulating motor performance and the modeling of smooth continuous movement trajectories is especially difficult (Byrne et al., 2010; Flash & Hogan, 1985). Our sequence-of-point method addresses this problem by decomposing continuous motor movement trajectories into unit vectors of fixed magnitude and variable direction. Our approach also specifies a continuous velocity profile across the transitional boundaries sequential unit vectors based on Flash and Hogan's (1985) dynamic cost optimization method for the mathematical modeling of human movements. This method was also utilized in a related previous study by Byrne et al. (2010) but was restricted in that study to simple linear movements only. In addition, Byrne et al.'s approach relies solely on the imaginal module of the ACT–R cognitive architecture for virtual visual targets for motor movement termination. In contrast, our approach affords modeling of curved as well as linear motor movements by specifying different parametric equations for various segments of the trajectory. Furthermore, our method specifies a separate abstract process that integrates the task declarative knowledge with the mechanisms of the imaginal module to determine spatial locations for unit movement termination. This allows flexible, robust, and on-the-fly determination of movement trajectory that simulates the effect of different instructional approaches on postlearning task performance.

The second problem is more important and relates directly to the overall objective of the study, which is to investigate the integrated, intertwined role of cognitive mental task representations acquired from different levels of dynamic instructional visualizations on postlearning procedural-motor task performance (see Akinlofa et al., 2013b; Wong et al., 2009). We model this through different declarative knowledge structures of the mental task models acquired through instructions with different levels of dynamic visualizations component. Furthermore, our approach abstracts the underlying cognitive processing and trajectory computations from the ACT–R manual module, which executes the actual motor movements. The abstraction process relies on a process control law similar to Salvucci's (2006) 2-points model for modeling lateral steering control in highway driver behavior (see also Salvucci & Gray, 2004). Salvucci's method, however, does not address prior learning and acquisition of mental task models through different instructional formats and the subsequent effect of this on postlearning performance. Our method's control law is a novel application of Fajen and Warren's (2003) steering model in which the ideal movement trajectory becomes the heuristic for an abstract process for integrating participant's mental task model with actual motor execution.

The sequence-of-point modeling method combines the partial matching mechanism of the ACT–R retrieval module with an extension of the activation equation to simulate the stochasticity of spatial location recall during the motor task execution. This afforded the fairly accurate simulation of human's ability to select and execute a specific movement trajectory from the large degrees of freedom inherent in skilled procedural-motor performance (see Viviani & Flash, 1995). Such extensions of the ACT–R architecture could be further developed to modeling more natural 3D spatial movements. One possible method could be the further extension of the activation equation to simulate spatial locations recall inaccuracies in a third "z" coordinate for a 3D reference framework. However, such an approach would require an upgrade of the visual system of the ACT–R architecture to support 3D visual location chunks, which is not possible in the current version 6.0.

The comparative analysis of the model's data with equivalent empirical data was more consistent in Experiment 1 than in Experiment 2. The inconsistencies with human data observed in Experiment 2 could be attributed to slight differences in the methodologies adopted, sample sizes, and data analysis techniques. Watson et al.'s (2010) sample sizes were quite small (10 participants per group), and the subsequent analysis is arguably not powerful enough to detect statistically significant differences in the performances of the independent groups. In contrast, sufficient runs of the computational models were conducted (100 runs per group), which afforded statistically significant differences to be observed in the postlearning task performance measures. In general, however, the computational model's predictions were closely accurate for comparative human data in the two experiments conducted. Our results provided evidence that dynamic instructional visualizations may be more effective for learning procedural-motor skills than their static equivalents. This is also consistent with the view that postlearning performance is moderated by the type of requested knowledge (Höffler & Leutner, 2007), the level of dynamism of the instructional interface (Akinlofa et al., 2013a, 2013b; Höffler & Leutner, 2011) and dedicated processing of dynamic instructional percept through a separate working memory *motor processor* construct (Wong et al., 2009).

5. LIMITATIONS

The computational models developed in this study were implemented in the ACT–R 6.0 cognitive architecture version. Accordingly, the simulations were constrained to the 2D spatial reference framework of the ACT–R visual system. The corresponding human performance data, however, involved natural 3D spatial movement. We minimize this limitation by integrating well-established mathematical models of human movement from previous related research in the design. In addition, we modeled only a subset of procedural-motor movements that lie in a 2D reference framework and excluded all other with trajectories orthogonal to this primary plane. An extension of the ACT–R activation equation could be a possible methodology for future work to extend the modeling to 3D spatial movements. This would however require substantial upgrade to the visual reference system of the base ACT–R architecture.

Furthermore, it has been argued that the design of computer based learning interfaces may interact with the learner's cognitive characteristics (Davis & Bostrom, 1992). In particular, the participant's spatial ability and domain expertise has been established as a moderating factor for postlearning procedural-motor performance by previous related research (Gegenfurtner, Lehtinen, & Saljo, 2011; Höffler, 2010; Höffler & Leutner, 2011). In contrast to the corresponding human data however, our models did not control for this factor, which limits the generalizability of our findings.

6. CONCLUSION AND FURTHER WORK

We utilized a novel computational modeling methodology to argue for a central cognitive role of acquired mental task representations in the postlearning performance of skilled motor tasks. The methodology distinguished mental task representations acquired from instructions with dynamic visualization contents as opposed to those with static alternatives and demonstrated their comparative moderating effects on efficient transfer to actual motor performance. There were two components of the methodology, each addressing separate aspects of problems associated with detailed modeling of fine, human motor performance in contemporary cognitive architectures like the ACT–R 6.0. The first part is a sequence-of-point technique for the specification of task-related spatial knowledge in declarative working memory. This technique is based on the application of well-accepted mathematical models to generate list structures that simulate variously acquired mental task models in the declarative knowledge module of the base cognitive architecture. These structures are later integrated with the subsequent execution of the procedural motor task to simulate differences in performance corresponding to the different initial instruction formats. The second component of the methodology is a movement control mechanism for the integration of the mental task models to actual task execution. This is implemented as a motor control law based also on established mathematical models of human motor control. The motor control law affords the translation of variously acquired mental task representations into smooth, continuous human movement in the execution of the task. It also specifies a process for simulating the stochastic but effective selection of a desired movement trajectory from an infinite range of alternatives that is inherent in human motor performance. The combination of the sequence-of-points technique and the movement control mechanism constitutes the methodology that affords the simulation of the atomic motor actions evident in skill acquisition and performance. To the best of our knowledge, this is a novel paradigm for the computational modeling of skilled human motor performance, which overcomes the limitation of coarse motor output inherent in the default implementation of contemporary cognitive modeling architectures such as the ACT–R 6.0.

We validate our methodology through incremental development of ACT–R 6.0 models in two experiments and the comparative analysis of the model's outputs with equivalent empirical human data from previous studies. The first experiment's model provided a proof of concept but was limited to a single step of a procedural task sequence. The second experiment's model extended the methodology to the entire task sequence to overcome this limitation. The two models' quantitative performance measures were fairly accurate and correlate significantly with the equivalent human data. This provides further evidence that dynamic instructional visualizations are more effective that their static alternatives for capturing the latent transitory information that are intrinsic and key to the efficient execution of skilled procedural motor tasks (Akinlofa et al., 2013b; Wong et al., 2009). The results are, however, limited as the model movements were implemented in 2D space as opposed to the more natural 3D human movements used in the comparative studies. This limitation is dictated by the underlying restrictions of the ACT–R 6.0 default visual module used for implementation and may be overcome in further studies by an extension of the sequence-of-point technique as we have specified. Future studies would also be required to evaluate the established effect of other performance moderating factors, such as the learner's spatial ability, which was not accounted for in the implementation of our cognitive models.

Our results validate the methodology as a candidate reference framework that is based on rigorous mathematical models, which could be applied for the rapid simulation and testing of computer-based multimedia instructional designs. The results also have implications for the design and development of instructions where rapid skill acquisition is desired for immediate task performance. Examples of such instances may be in the conceptual training of maintenance engineering apprentices, the rapid briefing of enroute firefighting personnel on building layout or the transmission of mission-critical information to military commanders in active operations.

FUNDING

This work was supported by Petroleum Technology Development Fund, Nigeria grant PTDF/TR/PhD/AOR/206/09.

ACKNOWLEDGEMENTS

We thank Gareth Watson, Joe Butterfield, Ricky Curran, and Cathy Craig for granting access to the experimental materials and empirical data referenced in Experiment 2 of this study.

REFERENCES

Akinlofa, O., Holt, P. O., & Elyan, E. (2013a). Domain expertise and the effectiveness of dynamic simulator interfaces in the acquisition of procedural motor skills. *British Journal of Educational Technology*, *44*, 810–820.

Akinlofa, O., Holt, P. O., & Elyan, E. (2013b). Effect of interface dynamism on learning procedural motor skills. *Interacting with Computers*, *25*, 259–269.

Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science*, *29*, 313–341.

Anderson, J. R., Bothell, D., Bryne., Douglass, S., Lebiere, C. & Qin, Y. (2004). An Integrated theory of the mind. *Physcological Review III*, 1036–1060.

Arguel, A., & Jamet, E. (2009). Using video and static pictures to improve learning of procedural contents. *Computers in Human Behavior*, *25*, 354–359.

Bi, L., Gan, G., & Liu, Y. (in press). Using queuing network and logistic regression to model driving with a visual distraction task. *International Journal of Human-Computer Interaction*.

Byrne, M. D., O'Malley, M. K., Gallagher, M. A., Purkayastha, S. N., Howie, N., & Huegel, J. C. (2010). A preliminary ACT-R model of a continuous motor task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *54*, 1037–1041.

Davis, S., & Bostrom, R. (1992). An experimental investigation of the roles of the computer interface and individual characteristics in the learning of computer systems. *International Journal of Human-Computer Interaction*, *4*, 143–172.

Fajen, B. R., & Warren, W. H. (2003). Behavioral dynamics of steering, obstacle avoidance, and route selection. *Journal of Experimental Psychology: Human Perception and Performance*, *29*, 343–362.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, *47*, 381–391.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, *5*, 1688–1703.

Geary, D. (2007). Educating the evolved mind: Conceptual foundations for an evolutionary educational psychology. In J. Carlson & J. Levin (Eds.), *Psychological perspectives on contemporary educational issues* (pp. 1–99). Greenwich, CT: Information Age.

Gegenfurtner, A., Lehtinen, E., & Saljo, R. (2011). Expertise differences in the comprehension of visualizations: A meta-analysis of eye-tracking research in professional domains. *Educational Psychology Review 23*, 523–552.

Hamil, J., & Knutzen, K. (2003). *Biomechanical basis of human movement*. Baltimore, MD: Lippincott Williams & Wilkins.

Hegarty, M. (2004). Mechanical reasoning by mental simulation. *Trends in Cognitive Sciences*, *8*, 280–285.

Hegarty, M. (2005). Multimedia learning about physical systems. In R. E. Mayer (Ed.), *The Cambridge handbook of multimedia learning* (pp. 447–465). New York, NY: Cambridge University Press.

Höffler, T. N. (2010). Spatial ability: Its influence on learning with visualizations—A meta-analytic review. *Educational Psychology Review*, *22*, 245–269.

Höffler, T. N., & Leutner, D. (2007). Instructional animation versus static pictures: A meta-analysis. *Learning and Instruction*, *17*, 722–738.

Höffler, T. N., & Leutner, D. (2011). The role of spatial ability in learning from instructional animations—Evidence for an ability-as-compensator hypothesis. *Computers in Human Behavior*, *27*, 209–216.

Hogan, N. (1984). An organizing principle for a class of voluntary movements. *Journal of Neuroscience*, *4*, 2745–2754.

Huegel, J. C., Celik, O., Israr, A., & O'Malley, M. K. (2009). Expertise-based performance measures in a virtual training environment. *Presence: Teleoperators and Virtual Environments*, *18*, 449–467.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the epic architecture for cognition and performance with application to human-computer interaction. *Human–Computer Interaction*, *12*, 391–438.

Kieras, D., Meyer, D. E., Ballas, J., & Lauber, E. (2000). Modern computational perspectives on executive mental processes and cognitive control: Where to from here? In S. Monsell & J. Driver (Eds.), *Control of cognitive processes, attention and performance* (pp. 681–712). Cambridge, MA: MIT Press.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1–64.

Liu, Y. (2009). QN-ACES: Integrating queuing network and ACT-R, CAPS, EPIC, and Soar architectures for multitask cognitive modeling. *International Journal of Human–Computer Interaction*, *25*, 554–581.

Martin-Emerson, R., & Wickens, C. (1992). The vertical visual field and implications for the head-up display. In *Proceedings of the Thirty-sixth Annual Symposium of the Human Factors Society* (pp. 1408–1412). Santa Monica, CA: Human Factors Society.

Mayer, R., Hegarty, M., Mayer, S., & Campbell, J. (2005). When static media promote active learning: Annotated illustrations versus narrated animations in multimedia instruction. *Journal of Experimental Psychology: Applied*, *11*, 256–265.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review*, *104*, 3–65.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *48*, 362–380.

Salvucci, D. D., & Gray, R. (2004). A two-point visual control model of steering. *Perception*, *33*, 1233–1248.

Viviani, P., & Flash, T. (1995). Minimum-jerk, two-thirds power law, and isochrony: Converging approaches to movement planning. *Journal of Experimental Psychology: Human Perception and Performance*, *21*, 32–53.

Watson, G., Butterfield, J., Curran, R., & Craig, C. (2010). Do dynamic work instructions provide an advantage over static instructions in a small scale assembly task? *Learning and Instruction*, *20*, 84–93.

Wong, A., Marcus, N., Ayres, P., Smith, L., Cooper, G. A., Paas, F., & Sweller, J. (2009). Instructional animations can be superior to statics when learning human motor skills. *Computers in Human Behavior*, *25*, 339–347.

ABOUT THE AUTHORS

Olurotimi Richard Akinlofa is a doctoral student in the School of Computing. His research is on the computational modelling of the cognitive effects of different interface visualizations in engineering training simulators.

Patrik O'Brian Holt is a Research Professor and Director of the Computational Systems Joint Research Institute, which is part of the Northern Research Partnership. His current research includes augmented reality and cognitive modeling.

Eyad Elyan is a lecturer in the School of Computing. His principal research interests are in the areas of 3D object modeling and recognition.