ASEE 2019 Annual Conference
CHARGED UP FOR THE NEXT 125 Years
126TH ANNUAL CONFERENCE & EXPOSITION
ASEE

Paper ID #27457

Using NGOMSL for Formative Feedback Generation in a Virtual Learning Environment

Dr. Yi-hsiang Isaac Chang, Illinois State University

Dr. Yi-hsiang Chang is an assistant professor in the Department of Technology at Illinois State University. He received a MSME degree from Carnegie Mellon University, a MSIE degree and a PhD degree in Technology, both from Purdue University. Dr. Chang has worked with several international corporations including Boeing, DaimlerChrysler, and Dassualt Systemes, as well as multiple small to medium size companies in both US and Asia. His current field of focus is in product innovation, process improvement, and technology diffusion.

Dr. Kevin L. Devine, Illinois State University

Kevin is the Program Coordinator for the Engineering Technology major at Illinois State University. His primary teaching assignments are in engineering graphics, industrial robotics, and CNC programming/machining.

Using NGOMSL for Formative Feedback Generation in a Virtual Learning Environment

Abstract

This research paper presents a feedback generation system using the Natural GOMS Language (NGOMSL) to describe the learner's tasks and performance expectation in a virtual reality-based learning environment. The acronym GOMS stands for Goals, Operators, Methods, and Selection Rules, and is derived from Card, Moran, and Newell's Human Processor Model. It is a representation of a series of "how-to's" required by a system in order to accomplish the intended tasks. NGOMSL is a variation of GOMS that provides an easy-to-read framework that highlights the underlying procedural rules of the task and the assessment of learner mastery of the task. A robust NGOMSL model will be able to provide the essential instructions regarding how the learner's perceptual, cognitive, and motor subsystems should work, and also improve the learner's ability to complete the task with proficiency.

In this paper we first use the domain of industrial robot programming as an example to discuss the difference between the Virtual Reality (VR) based learning environment and the traditional learning environment. Second, using a typical task in industrial robot programming as the point of discussion, our previous work in GOMS and its limitation are discussed. Next the rationale of using NGOMSL to model the procedural knowledge in a VR-based learning environment is presented, along with a comparison of using the NGOMSL model for the same task previously described using the GOMS model. The preliminary results of time duration measurement is presented, along with the discussion of how the proposed feedback system will work. This paper will conclude with a discussion of future work needed to implement the NGOMSL instructional models in a classroom setting.

Introduction

The effectiveness of using simulation-based learning strategy for procedural knowledge or skills has been reported by many researchers [1]–[3]. Such an approach can "replace and amplify real experiences" through proper guidance to "evoke or replicate substantial aspects of the real world" [4]. Simulated environments that can provide the high fidelity immersive experience, such as CAVE [5]–[7] or other forms of visualization representation can help the learners create the necessary cognitive connection [3], [10] between the physical world and the computer-generated instance.

With the technological advancement in visual computing, simulation-based learning through technology such as virtual reality (VR) that provides the essential immersive experience has become more affordable [11]. Nevertheless, a review of literature revealed only a few cases of VR-based STEM learning being reported [12]. Different from the game-based learning strategy [13], [14], learning in immersive VR environments must properly reflect the physical laws or spatial constraints governing our surrounding in order to imitate the real world experience. A typical example is the building activity in the video game Fortnite [15]. While the player has to collect material before actual construction can happen, the structures created were so simplified that they could not exist in the physical world. Video games such as Fortnite do not require an immersive environment, as the focus of the games is on strategy thinking instead of learning about engineering concepts and design processes.

VR-based Learning for Industrial Robot Programming

One factor that may contribute to the lack of adoption of VR-based STEM learning is the design of the virtual content. As indicated by Burdea and Coiffet [16], there are three aspects to be considered in order to create a sound VR-based environment, namely immersion, interaction, and imagination. The level of immersion is objective and mainly determined by the amount of technology incorporated. For example, the resolution of the head-mounted display governing virtual objects' visual presentation, alternative input and output functions enabling the user present or navigate within the virtual world, and so on. In the case of industrial robot programming, the VR add-on in the latest release of ABB Robotics' RobotStudio [17] allows the user to program the robot path by virtually "grabbing" the end effector and moving the robot to the desired location, a task that is traditionally accomplished either through the physical robot teach pedant or specifying points in space within the desktop programming environment.

The aspect of imagination can be very subjective and personal, relying heavily on the user's previous experience and mental creativity to utilize the full potential of the VR system. During simulation-based robot program verification, for example, the desktop user must use the computer mouse to change the viewpoint in the three-dimensional space to evaluate the robot path. On the other hand, the VR add-on of RobotStudio allows the user to navigate the space by physically moving around the virtual robot work cell or using the controller to change the viewpoint in a similar manner of using the computer mouse. Without some imagination of how the system could work, an expert user (instead of novice users) might choose the controller instead of walking to the viewing position desired.

The aspect of interaction is built upon human sensations such as touch and hearing. One example interaction is the use of haptic gloves to provide the sensation of touch in the virtual space. The stimuli needed to provide sensory registration for the user, and thus trigger deep learning, can be provided by a few key interactions in the VR environment. An inappropriate amount of interaction, however, can result in information shortage or overload, which might confuse the user. Until a suitable balance among immersion, interaction, and imagination is presented, the virtual content might not make sense to the user. It is essential for those who design the VR environments to consider the core cognitive activities needed during the learning of procedural knowledge or skills.

While being able to provide the immersive experiences needed by the encapsulation of the user's view through the head-mounted displays, modern VR technology also presents some challenges. Conventionally the learning of procedural knowledge is done by physically performing the desired tasks where the instructor is able to give formative assessment and provide feedback after observing the learner's behavior. However, there is only a limited amount of information provided to the external observer in a two-dimensional display for most of today's commercial VR environments. The instructor outside of the loop (e.g. virtual world) will only see either partial content within two circular areas side by side, or a square or rectangular window on the screen. Without knowing the learner's focal point or seeing the learner's facial expression, the instructor has little evidence to determine the learner's cognitive status or ability to complete the tasks in the real environment. To address this concern, an intelligent VR-based learning environment that is able to determine the learner's cognitive progress and prompt hints for problem solving is warranted.

Modeling Cognitive Activities: An Example

Chang and Devine [18] proposed to use the GOMS framework to model the user's cognitive activities, including navigation, inspection, and manipulation in the VR environment. GOMS stands for Goals, Operators, Methods, and Selection Rules, and could be used to describe procedural knowledge, e.g. the how-to's. First proposed by Card, Moran, and Newell [19], GOMS and its variants have been used to model the human computer interaction and assess the usability of various systems. Chang and Devine illustrated the use of GOMS to describe the work required to define the Cartesian Coordinate system for a workpiece, a typical task in industrial robot programming [20]. By selecting three points sequentially on the edges of a workpiece in the CAD-like environment (Figure 1), an X-Y-Z system corresponding to the workpiece can be defined so that the robot's end effector could approach the workpiece correctly.

Figure 1. Set up the coordinate system on the prismatic workpiece

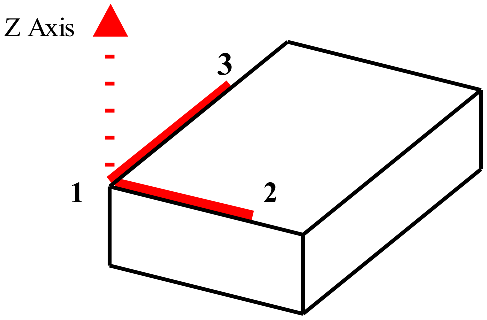


Figure 2. The flow chart for creating the coordinate system

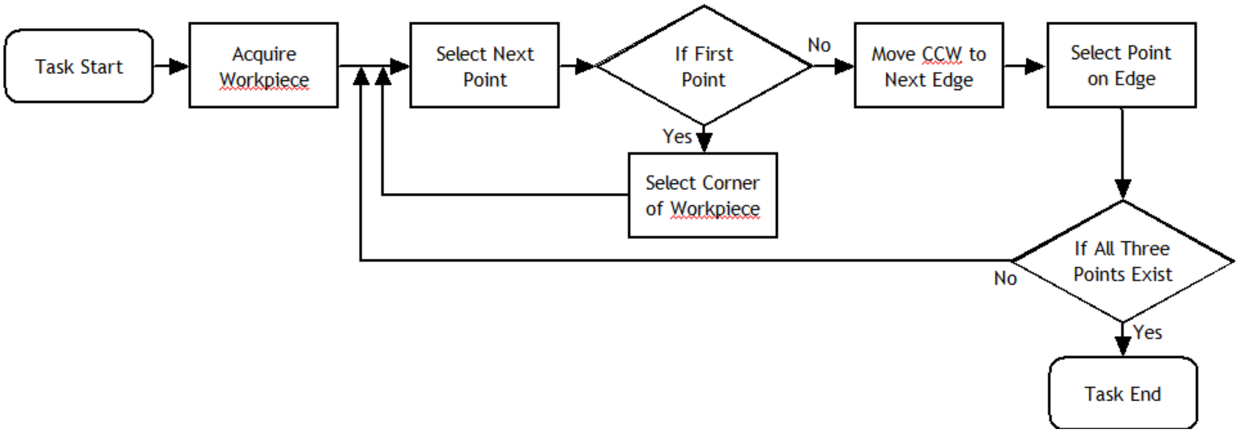


Figure 2 illustrates the cognitive process to create a Cartesian Coordinate system on the prismatic workpiece. After acquiring a workpiece, the user will choose Point 1 at the corner and Point 2 on one edge to create the X axis. A third point, Point 3, will be allocated on an edge in the counter clock-wise direction in order to create the Y axis; from Point 1 to Point 3 comes the positive Y direction so that the positive Z axis will point upward (right hand coordinate system rules apply). As shown in Figure 2 there are two decision making points in this process, the moments when the user engage in problem solving activities, and in our experience the most time-consuming parts of the whole exercise.

Figure 3. The GOMS model for creating the workpiece's coordinate system [18]

```
GOAL: CREATE-COORDINATE
.        GOAL: CHOOSE-POINT … repeat until all three points selected
.        .        GOAL: ACQUIRE WORKPIECE … if workpiece exists
.        .        GOAL: MOVE-CURSOR-TO-EDGE … choose edge
.        .        GOAL: CHOOSE-POINT-EDGE … choose a point along one edge
.        GOAL: VERIFY-Z-DIRECTION … verify if the z axis is in the right direction
```

The GOMS model proposed by Chang and Devine [15] for this task is shown in Figure 3, consisting of a hierarchical structure and loops (e.g. CHOOSE-POINT), if certain tasks had to be repeated until all requirements are met. The selection rules, or the decision-making process, are implicit in the GOMS model. Once the process is modeled, the time needed for expert users to execute each goal is measured. Instead of reporting the average time needed and its standard deviation, a GOMS model presented a sequence of "goals" required for completing the task, along with a range of time for each goal. Since procedural knowledge or skill is practicable, we can determine whether or not a specific learner is proficient if he or she can complete the goals within the given amount of time established by the expert users.

Table 1. The preliminary result of time needed for the procedure to set up a Cartesian Coordinate system on a workpiece within a VR-based industrial robot programming environment

Step	Task Description	Time Needed (seconds)	Type of Cognition Activity
1	Locate the object of which will be utilized as workobject	2 – 5	N
2	Travel to a location next to the object ideal for the following tasks	5 – 15	N
3	Determine the alignment of Cartesian Coordinate System (CCS) on the object	5 – 10	I
4	Determine the proper alignment for creating CCS on the object	5 – 10	I
5	Choose the first point based on step 4	2 – 5	M
6	Choose the second point based on step 4	2 – 5	M
7	Choose the third point based on step 4	2 – 5	M
8	Click on the "Create" button to establish CCS based on steps 4-7	2 – 5	M
9	Step back, reorient the view, and locate the object	5 – 10	N
10	Determine if the desired CCS is correctly created	5 – 15	I
	Sum Total of Time	31 – 85	seconds

Table 1 depicts the preliminary result of time measurement for expert users to execute the procedure to set up a Cartesian Coordinate system in ABB's RobotStudio with the VR add-on. The goal of each step is described, and the amount of time needed is reported in seconds. The letters in the last column indicate specific types of cognitive activities that the user is performing; N, I, and M represents Navigation, Inspection, and Manipulation respectively [15]. As being used in most research to assess human task performance, the sum total of time needed for expert users to perform the procedure is between 31 to 85 seconds. Time needed for individual steps, nevertheless, are critical as they give the system a baseline to determine whether or not an individual may need help or formative feedback at specific steps, if he or she takes longer than what expert users take.

The range of time needed is presented instead of average values, because an individual's cognitive processing speed may vary, even for the same expert user to perform the same step at different times in our preliminary study. Besides, the mental status of the user (e.g. tiredness, level of concentration, mood, motivation, etc.) could also affect his or her cognitive performance. The average of time needed may change if more expert users participate in the time measurement, but the total number of participants will only affect the range slightly.

Using NGOMSL for Feedback Generation

While the GOMS modeling strategy presented by Chang and Devine seemed promising, its concise statement for each goal did not convey task content clearly, and therefore was not suitable for the purpose of providing formative feedback. Kieras [21] proposed the use of NGOMSL for instructional purposes. NGOMSL, abbreviation of Natural GOMS Language, is based on the cognitive modeling of human computer interaction by Kieras and Polson [22]. As pointed by John and Kieras [23], NGOMSL is suitable for many computing situations in which the user's procedures are hierarchical and sequential. The explicit nature of its syntax enables NGOMSL to be a strong candidate for generating instructions or feedback.

For the example task described in Figure 2, its NGOMSL model for the cognitive activity is shown in Figure 4. The procedural knowledge is explicitly described, and the decision-making points are clearly identified. The benefit of this modeling method is that if the learner spends more time in one goal (e.g. exceed the expert's time range), the system can prompt with hints without external intervention. The hints can be extracted directly from the NGOMSL model, in the form of task description.

Figure 4. The NGOMSL model for creating the workpiece's coordinate system

```
Method for goal: choose next point to create the coordinate system
        Step 1. Get next unit task information from the workpiece.
        Step 2. Decide: If all three points have been selected, then return with goal accomplished.
        Step 3. Accomplish goal: move to the unit task location.
        Step 4. Accomplish goal: perform the unit task.
        Step 5. Goto 1.
```

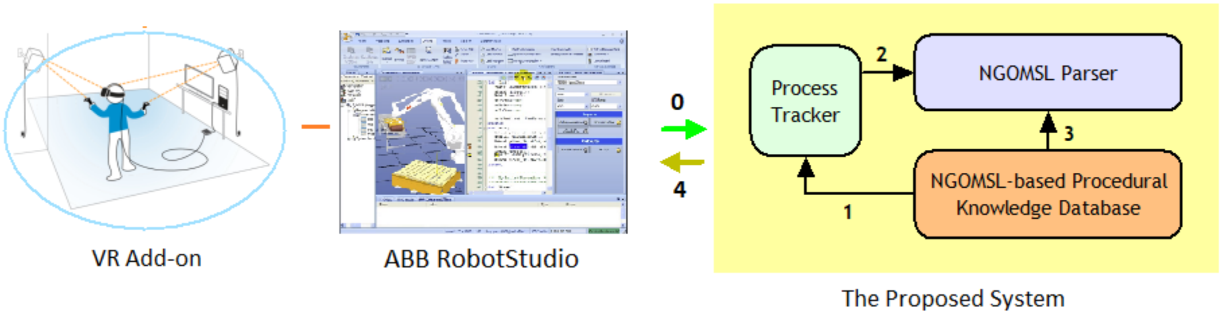
(Figure 4. The NGOMSL model - continued)

```
Selection rule set for goal: perform the unit task
    If the task is selecting the first point, then accomplish goal: assign corner point.
    If the task is selecting the second or third point, then accomplish goal: assign edge point.
    Return with goal accomplished.

Method for goal: assign corner point
    Step 1. Get location information of desired corner.
    Step 2. Decide: If a point exists at corner, then return with goal accomplished.
    Step 3. Accomplish goal: select point at corner.
    Step 4. Return with goal accomplished.

Method for goal: assign edge point
    Step 1. Get location information of desired edge.
    Step 2. Decide: If a point exist on edge, then return with goal accomplished.
    Step 3. Accomplish goal: select point on edge.
    Step 4. Return with goal accomplished.
    Step 5. Goto 2.

Method for goal: select point at corner
    Step 1. Locate corner.
    Step 2. Move cursor to corner.
    Step 3. Click mouse button.
    Step 4. Verify correct corner is selected.
    Step 5. Return with goal accomplished.

Method for goal: select point on edge
    Step 1. Locate edge.
    Step 2. Move cursor to edge.
    Step 3. Click mouse button.
    Step 4. Verify correct edge is selected.
    Step 5. Return with goal accomplished.
```

Conceptualization of the Feedback System

Figure 5 illustrates how the proposed NGOMSL-based feedback system can be implemented into ABB's RobotStudio. There are three components in the system, namely the Process Tracker used to determine the user's progress and time used, NGOMSL Parser to generate formative feedback based on the situation, and NGOMSL-based Procedural Knowledge Database. Once the user enters into the ABB RobotStudio VR environment, the information of the procedure to be executed will be sent to Process Tracker (e.g. the flow "0" from ABB's RobotStudio toward the proposed system), which will retrieve the information of the task's time duration suggested by expert users, from the Procedural Database (e.g. the flow "1") and start monitoring the user's progress.

Figure 5. The proposed NGOMSL-based system of automatic generation of formative feedback



If a specific step takes the user longer than the suggested time to complete, the Process Tracker will assume that the user needs help. It will send a flag of the specific step to the NGOMSL Parser (e.g. the flow "2"), which will retrieve the detailed information of the tasks from the Procedural Knowledge Database (e.g. the flow "3") and generate corresponding feedback and send the feedback to the user in-the-loop (e.g. the flow "4").

For example, if the user spends more than 15 seconds for the second step in Table 1, the feedback provided to the user by the proposed system will be in the form of a pop-up window with a message, "Please move closer to the object". Once the user acknowledges the feedback, the timer for step 2 will be reset to zero and the Process Tracker will keep monitoring this step until it is done. If the user did not move to a new position, or still spent more time to complete step 2, the system will assume the user is not proficient and needs help on the VR add-on's interface. A new feedback will be provided to the user, stating "Use either the touch controller or physically walk to move to a new position."

Discussion and Future Work

In this paper the authors have presented the rationale and the preliminary effort to utilize the NGOMSL framework for modeling the cognitive activities required to perform a task in a virtual world. An example in industrial robot programming was provided to highlight the potential of NGOMSL. The cognitive tasks for creating the coordinate system is the same for both the conventional desktop environment and VR-enabled environment. The difference is on how the user navigates in space, inspects the workpiece, and manipulates the object to complete the task. The VR-enabled environment, while providing the immersive experience to ease the user's cognitive workload from hand-eye coordination and mental rotation [24], greatly reduces the amount of information available for people out of the loop such as the instructor or colleagues. Consequently, it is difficult for an observer to provide feedback to improve the learning of procedural knowledge or skills. Thus, the introduction of NGOMSL into the VR-based environment may bridge the gap, helping the instructor assess the learner's cognitive stage to form intervention strategies. Nevertheless, more work is still needed in order to incorporate this cognitive model into the production environment or for class usage.

Future research will include evaluating procedural learning material for a variety of robot programming tasks so that the cognitive activities required to accomplish specific tasks can be identified. NGOMSL models will then be developed based for the tasks using a yet to be developed template. Using special software such as Micro Saint Sharp Simulation [25] for scenario scripting and task monitoring will be used to ease the instructor's load further.

Additionally, the three-dimensional homing function of the VR environment needs to be improved. Due to the limitation of hardware and tracking algorithms used, the current VR controllers such as those for HTC VIVE or Oculus Rift can only estimate the point or direction in space. Niehorster, Li, and Lappe [26] studied the accuracy and precision of HTC VIVE's position and orientation tracking and conclude that it was not yet suitable for scientific research due to the varying offsets existing between the virtual and physical tracking space. Our experience of these systems also confirmed that it was not trivial to select a virtual entity such as points or edges with these controllers. Additional calculation is necessary to figure out where the user points at, and the virtual object can then be snapped by the user through visual feedback such as a small circle or highlighted items when the user is at the proximity of the entity. Additionally, spatial references such as grids or rulers along the axes will also improve the system's usability.

Finally, there is a need to inform the people out of the loop where the user's focal point is in the virtual world. A simple circle or cross on the external screen could help the instructor to determine what the user is focusing on, so that intervention can be rendered. The eye tracking ability embedded the recently released VIVE PRO EYE system [27] might be just the perfect solution for this need.

Acknowledgement

The authors would like to express their sincere gratitude for the financial support from the College of Applied Science and Technology and from the SoTL Center, both at Illinois State University, as well as the help from Mr. Gunnar Klitzing and Mr. Manan Shah.

Reference

[1] G. K. Chung, T. C. Harmon, and E. L. Baker, “The impact of a simulation-based learning design project on student learning,” *IEEE Trans. Educ.*, vol. 44, no. 4, pp. 390–398, 2001.

[2] L. Davidovitch, A. Parush, and A. Shtub, “Simulation-based learning in engineering education: Performance and transfer in learning project management,” *J. Eng. Educ.*, vol. 95, no. 4, pp. 289–299, 2006.

[3] M. Ronen and M. Eliahu, “Simulation — a bridge between theory and reality: the case of electric circuits,” *J. Comput. Assist. Learn.*, vol. 16, no. 1, pp. 14–26, Mar. 2000.

[4] F. Lateef, “Simulation-based learning: Just like the real thing,” *J. Emerg. Trauma Shock*, vol. 3, no. 4, p. 348, 2010.

[5] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, “The CAVE: audio visual experience automatic virtual environment,” *Commun. ACM*, vol. 35, no. 6, pp. 64–73, 1992.

[6] H. Ohtani and R. Horiuchi, “Scientific visualization of magnetic reconnection simulation data by the CAVE virtual reality system,” *Plasma Fusion Res.*, vol. 3, pp. 054–054, 2008.

[7] S. K. Tang *et al.*, “How to Simulate and Realise a Disappeared City and City Life?-A VR Cave Simulation,” 2002.

[8] D. A. Bowman, D. Koller, and L. F. Hodges, “Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques,” in *Virtual Reality Annual International Symposium, 1997., IEEE 1997*, 1997, pp. 45–52.

[9] G. Robertson, M. Czerwinski, and M. Van Dantzich, “Immersion in desktop virtual reality,” in *Proceedings of the 10th annual ACM symposium on User interface software and technology*, 1997, pp. 11–19.

[10] T. Griffiths and D. Guile, “A connective model of learning: The implications for work process knowledge,” *Eur. Educ. Res. J.*, vol. 2, no. 1, pp. 56–73, 2003.

[11] B. Dalgarno and M. J. Lee, “What are the learning affordances of 3-D virtual environments?,” *Br. J. Educ. Technol.*, vol. 41, no. 1, pp. 10–32, 2010.

[12] H.-M. Huang, U. Rauch, and S.-S. Liaw, “Investigating learners’ attitudes toward virtual reality learning environments: Based on a constructivist approach,” *Comput. Educ.*, vol. 55, no. 3, pp. 1171–1182, 2010.

[13] S. E. Kirkley and J. R. Kirkley, “Creating next generation blended learning environments using mixed reality, video games and simulations,” *TechTrends*, vol. 49, no. 3, pp. 42–53, 2005.

[14] M. J. Mayo, “Video games: A route to large-scale STEM education?,” *Science*, vol. 323, no. 5910, pp. 79–82, 2009.

[15] M. B. Read, “Fortnite building guide: How to build with materials and traps in Fortnite,” *Eurogamer*, 24-Jul-2018. .

[16] G. C. Burdea and P. Coiffet, “Virtual Reality Technology,” *Presence Teleoperators Virtual Environ.*, vol. 12, pp. 663–664, 2003.

[17] “Release Notes RobotStudio 6.07,” p. 32, 2018.

[18] Y.-H. Chang and K. L. Devine, “Modeling Cognitive Activities in a Virtual Reality-assisted Industrial Robot Programming Environment,” in *Proceedings of ASEE EDGD 73rd Midyear Conference.*, Berkeley, CA, 2019, pp. 99–104.

[19] S. K. Card, T. P. Moran, and A. Newell, “The keystroke-level model for user performance time with interactive systems,” *Commun. ACM*, vol. 23, no. 7, pp. 396–410, 1980.

[20] K. Devine, “Integrating robot simulation and off-line programming into an industrial robotics course,” in *ASEE Annual conference*, 2009.

[21] D. Kieras, “GOMS modeling of user interfaces using NGOMSL,” in *Conference companion on human factors in computing systems*, 1994, pp. 371–372.

[22] D. Kieras and P. G. Polson, “An approach to the formal analysis of user complexity,” *Int. J. Hum.-Comput. Stud.*, vol. 51, no. 2, pp. 405–434, 1999.

[23] B. E. John and D. E. Kieras, “The GOMS family of analysis techniques: Tools for design and evaluation,” CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1994.

[24] Y. I. Chang and K. L. Devine, “Board 83: A Tale of the Robot: Will Virtual Reality Enhance Student Learning of Industrial Robotics?,” in *2018 ASEE Annual Conference & Exposition*, 2018.

[25] N. Parameshwara, J. H. Kim, W. Guo, and K. S. Pasupathy, “NGOMSL simulation model in an Emergency Department,” in *2016 Winter Simulation Conference (WSC)*, 2016, pp. 1938–1949.

[26] D. C. Niehorster, L. Li, and M. Lappe, “The accuracy and precision of position and orientation tracking in the HTC vive virtual reality system for scientific research,” -*Percept.*, vol. 8, no. 3, p. 2041669517708205, 2017.

[27] “The New VIVE Pro.” [Online]. Available: https://www.vive.com/eu/pro-eye/. [Accessed: 05-Feb-2019].