APPLICATION OF AGENT-BASED MULTIMETHOD SIMULATION APPROACH TO THE SIMULATION TESTBED PROTOTYPE FOR THE CONCEPT EXPLORATION AND REQUIREMENT ANALYSIS OF UGV

Sang Yeong Choi
The Specialized Research Center
For Future Ground System Analysis
MyongJi University, Cheoin-gu,
Gyeonggi-do, Korea
E-mail: metayoung@gmail.com

Kang Park
Department of Mechanical
Engineering
MyongJi University, Cheoin-gu,
Gyeonggi-do, Korea
E-mail: kxp007@gmail.com

Ji Hyun Yang
Department of Automotive
Engineering
Kookmin University
Seoul, Korea
E-mail: yangjh@kookmin.ac.kr

Hwan Il Kang
Department of Info. & Com.
Engineering
MyongJi University, Cheoin-gu
Gyeonggi-do, Korea
E-mail: hwanilkang@naver.com

KEYWORDS
Agent-based multimethod simulation, concept exploration and requirement analysis, human performan model, simulation testbed, unmanned ground vehicle.

ABSTRACT

This paper presents the application of the agent-based multimethod simulation approach to the development of the Unmanned Ground Vehicle(UGV) simulation testbed prototype to support both concept exploration and the performance-based requirement analysis of future UGVs. We describe the UGV simulation testbed prototype (*UGVSimp*) incorporating 1) UGV mission scenario, 2) the operational concepts of UGVs with the automation level, such as the autonomous unmanned operation, the autonomous manned operation with human operators, and the autonomous autocontrolled operation with Command and Control(C2), 3) UGV functionality, including *move*, *fire*, *rescue*, 4) operator's task processes constrained by the automation level, human resources(visual, auditory, cognition, psychomotor), and the associated human machine interface of the control station. Furthermore, we show some experimental results for the UGV's competing operational concepts for the three cases(autonomous unmanned, autonomous manned, autonomous autocontrolled), using *UGVSimp*.

1. INTRODUCTION

A simulation testbed is needed for the concept exploration and performance-based requirement analysis in the conceptual stage of system acquisition. The term testbed refers to a simulation environment that has all the necessary things to experiment with the system of interest in a representive mission scenario. The simulation testbed of our interest is the UGV simulation testbed. Using the UGV simulation testbed, we will explore the operational concept of the future UGVs reflecting UGV mission scenario, UGV's functions and their automation levels, the UGV operator's tasks processes assigned according to the automation levels, and the human interface of the control station. Traditional Approach to such simulation testbeds is based on Constructive-Virtual(C-V) distributed simulation architecture as shown in Figure 1.

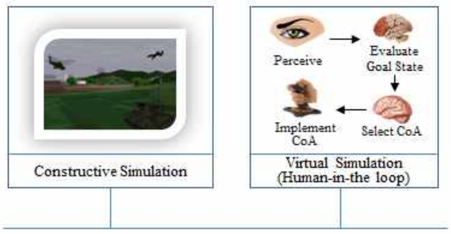

Figure 1: C-V Architecture of Simulation Testbed

In Figure 1, the constructive simulation is to represent the mission scenario and the human-in-the loop(HITL) virtual simulation for the operator's behavior. However, these are both expensive and time consuming because of costly HITL simulator. For this reason, human performance models(HPMs) were proposed as an alternative to traditional HITL virtual simulations(Pew 1998). The HPMs were accepted , for example, in the combat automation requirements testbed(CART)(Brett 2000), which is a well known simulation testbed for the performance-based analysis of crew systems. Additionally, the CART was implemented with the Constructive-Constructive(C-C) distributed simulation architecture. The C-C distributed simulation architecture of CART is cheaper than the legacy architecture and folster reusability and interoperability by using HLA/RTI(IEEE 1516). However, we believe that the C-C distribution simulation architecture is still lengthy and complicated to implement tasks associated with our purpose in regard to the UGV testbed. Our detailed examination of the simulaton testbed has revealed that

Proceedings 29th European Conference on Modelling and Simulation ©ECMS Valeri M. Mladenov, Petia Georgieva, Grisha Spasov, Galidiya Petrova (Editors)
ISBN: 978-0-9932440-0-1 / ISBN: 978-0-9932440-1-8 (CD)

the constructive simulation of mission scenario is implemedted in an event-scheduling world view manner. On the other hand, the human performance model is implemented with processed interaction world view. Although these are definitely taking different world views, nevertheless, it allowed us to attempt to apply the agent-based multimethod simulation for the development of the UGV simulation testbed. This is because the multimethod simulation might be much cheaper than the legacy architecture of the C-V or C-C distributed simulation archticture.

In this paper, we present the application of the agent-based mulitmethod simulation approach to the development of the UGV simulation testbed prototype called "*UGVSimp*", to support both concept exploration and performance-based requirement analysis of future UGVs. The purpose of this paper is twofold: the first is to address the usefulness of the multimethod simulation approach for the development of UGV simulation testbed, and the second is to identify the technical issues that may occur when the the simulation testbed is developed using the multimethod simulation paradigm. For these purposes, the functionalities of *UGVSimp* will be limited to: 1) a single mission scenario under which UGVs are dispatched to rescue an isolated combat team, 2) three types of operational concepts including the autonomous unmanned operation, the autonomous manned operation with operator, the autonomous autocontrolled operation with Command and Control(C2), 3) UGV mission essential functions which assume three main functions, i.e. *move*, *fire* and *rescue*, and 4) UGV operator task processes which are constrained by the operational concepts(automation level), human resources(visual, auditory, cognition, psychomotor), and the associated human interface of the control station.

We begine by presenting an overview of related studies in Section 2 and continue in Section 3 by addressing the conceptual model of *UGVSimp*. Section 4 explains the agent design of *UGVSimp*. Section 5 describes its implementation and presents some experiments and results using *UGVSimp*. Fnially, we outline the conclusions in Section 6.

2. RELATED WORK

Related published studies are summarized into three catagories: agented-based multimethod simulations, UGV automation, and human performance models.

Agent-Based Multimethod Simulation

Carson II(2004) addressed the traditional simulation worldview as event scheduling, process interaction, and activity scanning. These simulation paradigmes have advantages for modeling the real world. Considering the three world views, object-oriented simulation has provided a rich and lucid paradigm for building computerized simulation models by introducing the notion of object, as mentioned by Rothenberg(1989). However, recently agent-based simulations are drawing increasing attention because the notion of agent is deemed to be more powerful than that of object in the simulation. Objects in the object-oriented simulation typically act upon request only when their methods are called. Agents are more than objects. They are typically dynamic, have internal time delays, and can initiate events as individually autonomous entity(Borshchev 2013). A typical agent-based simulation model has three elements: the agent, the agent relationship, and methods of interaction with agent environment(Macal et al. 2013). Agents in a running agent-based model are more like threads or concurrent processes in a running program, and they live in and interact with their environment as well as with other agents using the strong mechanism of message-passing in an autonomous way(Borshchev 2013). This is where the model's strength lies, in that we can hybridize the different paradigms of simulation world views in a single agent-based simulation model. These types of simulations are referred to as "agent-based mulimethod simulation" in this paper.

UGV Automation

Glster et al.(2007) provide a review of automation and human factor, and the definition of automation that is concerned with the replacement of human functioning by machine functioning. Furthermore, they suggest eight levels of automation for practical application to uninhabited military vehicles by considering the type of automation (Wiener 1988, Sheridan 1988, Rasmussen 1983)and its level(AGARD 1986). The eight levels are no automation, manual augmented, manual augmented and limited, co-operative, automatic pre-select, automatic select, autonomous manned operation, and autonomous unmanned operation. On the other hand, Parasurman et al.(2000) proposed four functional area for automation: information acquisition, information analysis, decision and action selection, and action implementation. They also defined ten levels of automation for each functional area: full auto, auto-human informed in some occasions, auto-human informed, auto-human vetotime limit, auto exercise only if human approves, auto suggests, auto narrows, auto shows with all options, and manual. Even though, there is no universally agreed taxonomy of automation levels. Glster et al.(2007) and Parasurman et al.(2000), provide a reference model to decide on whether the function should be automated and to what extent.

We adopt the definition of Glster et al.(2007) and Parasurman et al.(2000), namely, the autonomous unmanned operation, and the autonomous manned operation that is controlled by an operator in a supervisory way. Aditionaly, for the sake of our purpose, we introduce another level of automation, i.e. autonomous autocontrolled operation, which is the same as the autonomous unmanned operation but is controlled by the automated command and control center. These automation levels constitute operational concepts we use for the concept expoloration and the performance-based requirement analysis in this paper.

Human Performance Model

Human performance is characterizd by human operator's workloads, UGV automation level, and human and machine interface of control station. The introduction of the automation to UGV replaces or supports human operator functioning at different levels. However, to a large extent, operators will interact with UGVs in a way of supervisory control manner where human operator behave like C2. Henry and Farrell(1997) proposed Human Information Process(HIP) model to represent human behavior as four stages consisting of the perception, the evaluation of goal state, the selection of the course of action, the implementation of course of action, on the basis of human information processing and multiple resource theories(Wickens 1984). These constitute human performance models where each stage is comprised of operator task proesses modelled with a series of tasks undertaked by the human operator. The task process resources are described by four components: visual, auditory, cognition, and psychomotor. These are referred to as VACP. McCrasken and Aldrich(1984) developed the rating scales from 0.0 to 7.0, providing a relative rating of the degree with which each resource component is used in order for the operator to carry out his/her task process modeled in the operator's task network model. Keller(2002) used this VACP rating scales to present a methodology for modeling the operator's performance within a discrete event simulation. We will also be using the VACP rating scale to measure the operator's workload in our experimentations.

3. CONCEPTUAL MODEL

For the purpose of this paper, the conceptual model of *UGVSimp* includes the representation of UGV mission scenario, UGV functionalty, UGV operational concepts, and the operator's task processes. Thus, we have six major agents: *BattleSpaceAgent*, *UGVAgent*, *OperatorAgent*, *TaskProcessAgent*, *C2Agent*, *ThereatAgent*. *BattleSpaceAgent* is for a real world representation and consists of a battle field, maneuver path, mission area, artificial and natural obstacles. *UGVAgent* and *OperatorAgent* are for the real UGV and operator, respectively. *TaskProcessAgent* is for the operator's task processes. *C2Agent* is for the command and control of UGVs by the target evaluation and weapon allocation(TEWA). *ThreatAgent* is for the adversary against the UGVs. The detailed decription of the conceptual model is shown in Figure 2. For the graphical depictions, we use oneSAF objective system conceptual modeling language(OOS CML)(Karr 2005). The basic building blocks of CML consists of element(colored green), event(yellow), behavior(blue). Herein, we add an additional notion, namely the process(pink). In Figure 2, elements represent agents. Agents issue events. Events then stimulate the behavior of other agents. In adition, we have three pattern actions. The pattern action *Autonomus Unmanned Operation* is for *UGVAgent* to pursue its mission in an autonomous

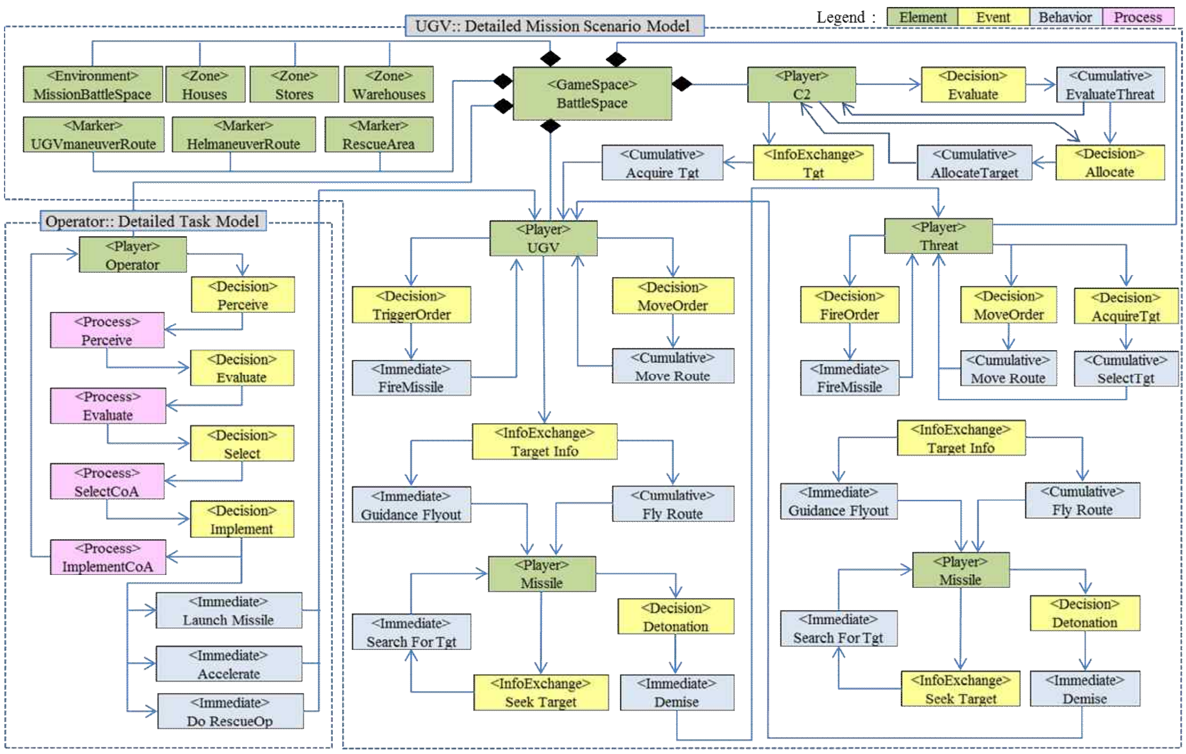

Figure 2: Conceptual Model of *UGVSimp*

way. The pattern action *Autonomous Manned Operation* is to pursue the mission using the supervisory control of an operator. The pattern action *Autonomous Autocontrolled Operation* is to persue the mission under a control of automated C2 with TEWA. Figure 3 shows one of the graphical depictions of the pattern action(*Autonomous Manned Operation*).

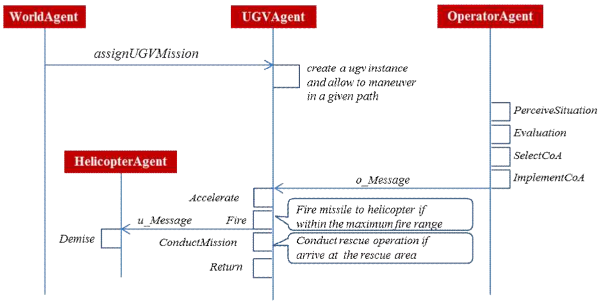

Figure3: Pattern Action *Autonomous Manned Operation*

4. AGENT DESIGN

We have six major agents in the conceptual model. However, we will be limited to describing three cases of agent design such as *UGVAgent*, *OperatorAgent*, *TaskProcessAgent*, because these agents are sufficiently enough to explain different simulation world views in the multimethod simulation.

UGVAgent

UGVAgent is the agent for the UGV which has three types of state machines: Maneuver(*AutoManeaver*, *MovementByOpCtrl*), Fire(*FireByAuto*, *FireByC2*, *FireByOperator*), and *Liveness*. Figure 4 shows one of the state machine types, namely, *FireByOperator*.

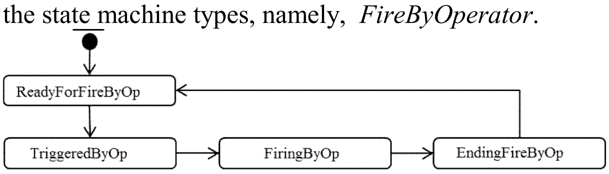

Figure 4: State Machine of *FireByOperator*

The definitions of the state machine and action type are shown in Tables 1 and 2.

Table 1: State Machine Definition of *FireByOperator*

State name	Exit condition	Exit Action	NextState
ReadyForFireByOp	*"TriggerFire"* received From TaskProcessAgent	Send *"StartFire"* to itself	TriggeredByOp
TriggeredByOp	*"StartFire"* received from itself	Do *fireMissile()*	FiringByOp
FiringByOp	*"EndOfFire"* received from MissileAgent		EndingFireByOp
EndingFireByOp	Timeout		ReadyForFireByOp

Table 2: Action Type Definition of *FireByOperator*

Action Name	Return Value	Description
fireMission()	void	Fire missile

OpertorAgent

OperatorAgent is the agent for the human operator who controls the UGV in a supvisory way. The state machine of *OperatorAgent* is shown in Figure 5. In this figure, CoA is short for course of action.

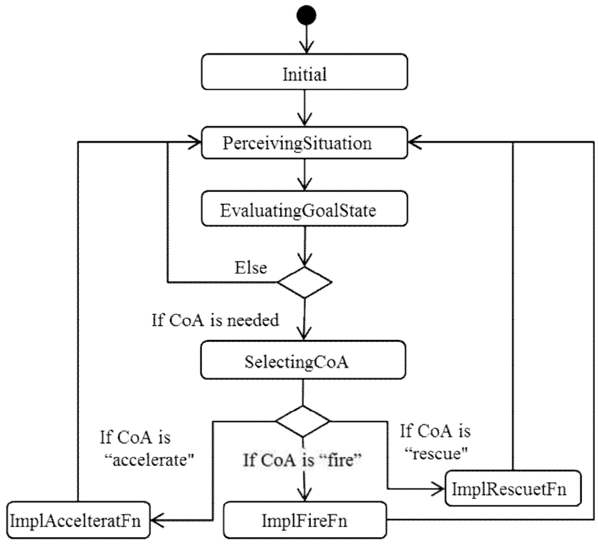

Figure 5: State Machine of *OperatorAgent*

The definition of the state machine is shown in Table 3, and the action carried out by the operator is defined in Table 4.

State name	Exit condition	Exit Action	Next State
Initial		Send *"ExecPerceive"* to TaskProcessAgent	PerceivingSituation
Perceiving Situation	*"RdyEvaluate"* received from TaskProcessAgent	Do *evaluateGoalState()* Send *"ExecEvaluate"* to TaskProcessAgent	EvaluatingGoalState
Evaluating GoalState	*"RdySelect"* received from TaskProcessAgent and CoA is needed	Do *selectCourseOfAction()* Send *"ExecSelect"* to TaskProcessAgent	SelectingCoA
	"RdySelect" received from TaskProcessAgent and CoA is not needed	Send *"ExecPerceive"* to TaskProcessAgent	PerceivingSituation
SelectingCoA	*"RdyImplement"* received from TaskProcessAgent and CoA is *accelerate*	Send *"ExecAccelerate"* to TaskProcessAgent	ImplementingAcc
	"RdyImplement" received from TaskProcessAgent and CoA is *fire*	Send *"ExecFire"* to TaskProcessAgent	ImplementingFire
	"RdyImplement" received from TaskProcessAgent and CoA is *rescue*	Send *"ExecRescue"* to TaskProcessAgent	ImplementingRescue
Implementing Acc	*"RdyPerceive"* received from TaskProcessAgent	Send *"ExecPerceive"* to TaskProcessAgent	PerceivingSituation
Implementing Fire	*"RdyPerceive"* received from TaskProcessAgent	Send *"ExecPerceive"* to TaskProcessAgent	PerceivingSituation
Implementing Rescue	*"RdyPerceive"* received from TaskProcessAgent	Send *"ExecPerceive"* to TaskProcessAgent	PerceivingSituation

Table 3: State Machine Definition of *OperatorAgent*

Table 4: Action Type Definition of *OperatorAgent*

Action Name	Return Value	Description
evaluateGoalState()	void	Evaluate the situation
selectCourseOfAction()	Goal Function Name	Select the goal function on the basis of the evaluation

TaskProcessAgent

TaskProcessAgent is for the human performance model representing the operator's task processes and the human resources of VACP. Figure 6 shows the simple task processes embodies in the *TaskProcessAgent*.

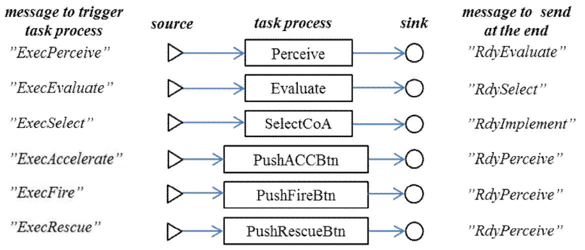

Figure 6: Task Process Model of *TaskProcessAgent*

The definition of the task process is shown in Table 5.

Table 5: Task Process Definition of *TaskProcessAgent*

Task Process Name	Entry Condition	Resource				Exit Condi-tion	Exit Action
		V	A	C	P		
Perceive	*"ExecPerceive"* received from OperatorAgent	√		√		Timeout	Send *"RdyEvaluate"* to OperatorAgent
Evaluate	*"ExecEvaluate"* received from OperatorAgent	√		√		Timeout	Send *"RdySelect"* to OperatorAgent
SelectCoA	*"ExecSelecte"* received from OperatorAgent	√		√		Timeout	Send *"RdyImplement"* to OperatorAgent
PushACCBtn	*"ExecAccelerate"* received from OperatorAgent	√		√	√	Timeout	Send *"TriggerAccelerate"* to UGVAgent Send *"RdyPerceive"* to OperatorAgent
PushFireBtn	*"ExecFire"* received from OperatorAgent	√		√	√	Timeout	Send *"TriggerFire"* to UGVAgent Send *"RdyPerceive"* to OperatorAgent
PushRescueBtn	*"ExecRescue"* received from OperatorAgent	√		√	√	Timeout	Send *"TriggerRescue"* to UGVAgent Send *"RdyPerceive"* to OperatorAgent

5. IMPLEMENTATION AND EXPERIMENTS

The implementation of *UGVSimp* is done using the commercial tool AnyLogic 6(XJ Technologies Company Ltd. 2013). This software package provides capabilities to implement multimethod simulation paradigms. In the *UGVSimp*, we implement *BattleSpaceAgent*, *UGVAgent*, *OperatorAgent*, *TaskProcessAgent*, *C2Agent*, *ThereatAgent* with the event scheduling world view. On the other hand, *TaskProcessAgent* is implemented with the process interaction world view. These are combined with the message-passing mechanism in the AnyLogic simulation engine as shown in Figure 7.

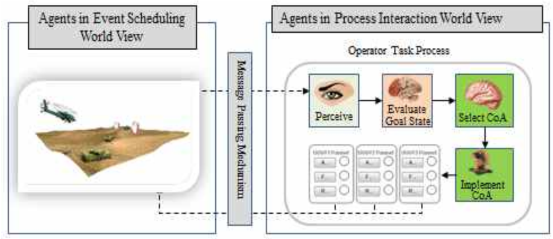

Figure 7: Message Passing Mechanism of *UGVSimp*

For illustrative purposes, we assume the following input data set, as shown in Table 6. We investigate the outcomes by the experimentation of the three case of the operational concepts(the autonomous unmanned, the autonomous manned, the autonomous autocontrolled) in terms of the amount of ammunitions consumed and operator's workload by changing the number of UGVs dispatched.

Table 6: Input Data

Type	Input Name			Scenario1
Battle Area	Mission Area			800*600: (80Km*60Km), 10pixels/Km
	Number of rescue area			1~3
	Distance to the rescue area			770: (77Km)
Threat Helicopter	Helicopter velocity			50: (300Km/h)
	Flight interval			7
	Maximum firing range			90: (9Km)
	Missile velocity			500: (Mach 2.5)
	Missile detonation range			0.3: (30m)
	Missile SSKP			0.5
Mission Vehicle	Number of vehicles			1~3
	Average velocity			10: (60Km)
	Maximum velocity			15: (90Km/h)
	Maximum firing range			90: (9Km)
	Missile velocity			500: (마하2.5)
	Missile detonation range			0.3: (30m)
	Missile SSKP			0.9
Operator	Perceive/Evaluate/Select Course of action		Performance time	Triangle(0.017,0.025,0.033): (1sec, 1.5sec, 2sec)
			WorkLoad value	V:5.0, C:6.8
	Implement course of action	Push Accelerate_button	Performance time	Tringle(0.008,0.017,0.025): (0.5sec, 1.sec, 1.5sec)
			WorkLoad value	V:3.7 C:1.2 P:2.2
		Push Fire_button	Performance time	Tringle(0.008,0.017,0.025): (0.5sec, 1sec, 1.5sec)
			WorkLoad value	V:3.7 C:1.2 P:2.2
		Push Rescue_button	Performance time	Uniform(3,5): (3min, 5min)
			WorkLoad value	V:3.7 C:1.2 P:2.2

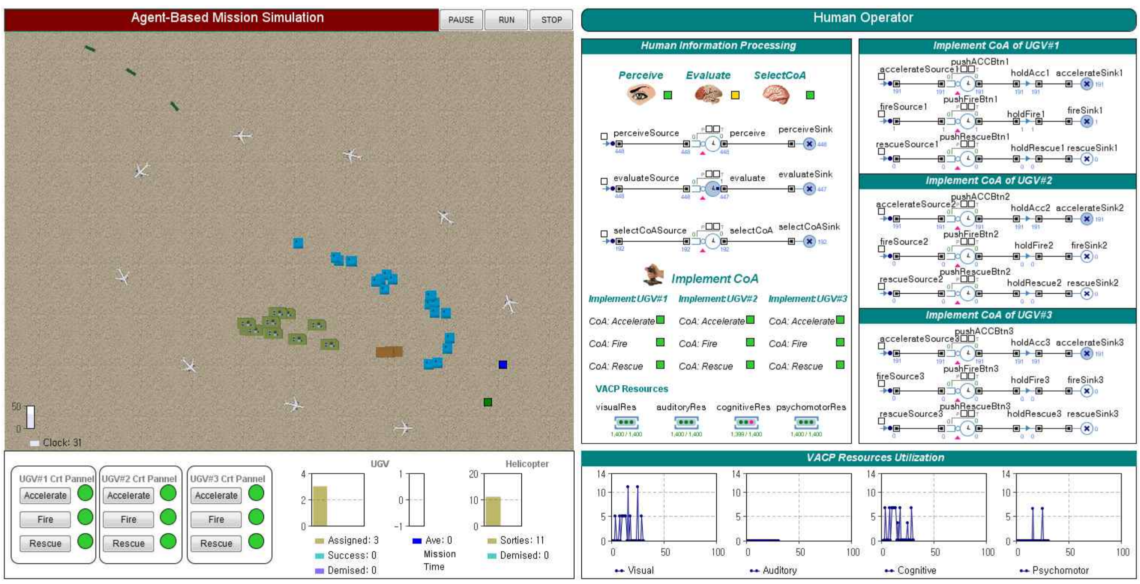

Figure 8: Snapshot of the Execution of *UGVSimp*

Figure 8 shows the snapshot of the execution of *UGVSimp*. Figure 9 shows the outcomes of average amount of ammunitions consumed by a single UGV for the three cases of operational concepts: the autonomous unmanned operation(AUTO), the autonomous manned operation(OP), the autonomous autocontrolled operation(C2). In these cases, two UGVs are employed, and the kill probability increases from 0.5 to 0.9 in intervals of 0.2. The simulation period is 3000 min.

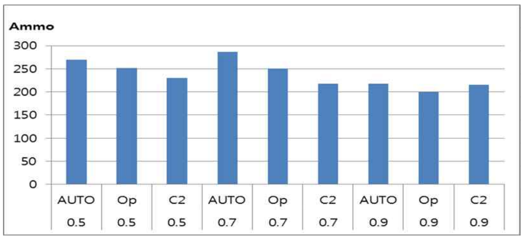

Figure 9: Experimental Results

Figures 10,11,12 show the sample of the operator's workload when the number of UGVs employed is 1,2, or 3 with the input data shown in Table 6. It is noted that the maximum workload for a single operator is 7 level.

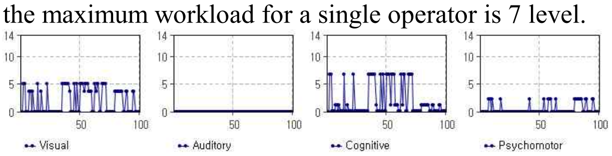

Figure 10: Case1: an operator controls a single UGV

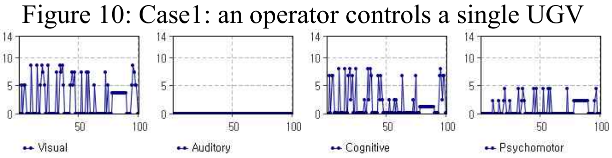

Figure 11: Case2: an operator controls two UGVs

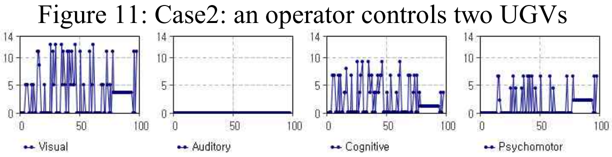

Figure 12: Case3: an operator controls three UGVs

6. CONCLUSIONS

We described the application of the agent-based mulitmethod simulation approach to the development of the UGV(Unmanned Ground Vehicle) simulation testbed prototype(*UGVSimp*) and have shown some experimental results for three cases of operational concepts: the autonomous unmanned operation, the autonomous manned operation, the autonomous autocontrolled operation. Throughout our attempts, we believe that multimethod simulation approach is very powerful to implement our UGV simulation testbed. However, some issues arise, such as the lack of standard visual modeling method and deadlock issue, because agents are deemed to be concurrent processes in the running simulation. The second issue might be trivial, but when the size of agents increases and the interactions between the agents implemented with different world views become heavy, the deadlock can be of significance and might need a resolvable algorithm. We will contine to extend the *UGVSimp* and address the issues that have been encountered throughout this work.

Acknowledgement

This work was supported by a research program(The Specialized Research Center For Future Ground System Analysis) funded by the Agency of Defense Development of Korea and we appreciate it.

REFERENCES

AGARD . 1986. " Improved guidance and control automation at the man-machine interface." In: Hollister, W.M. (Ed.). AGARD advisory report AR-228. Neuilly-sur-Seine: NATO.

Brett, B.E. et al. 2000. "The Combat Automation Requirements Testbed Task1 Final Report: Implementation Concepts and an Example" Interim Report. Human Effectiveness Directorate Crew System Interface Division 2255 H Street Wright-Patterson AFB, OH 45433-7022.

Borshchev A. 2013. "Agent based modeling". In *The Big Book Simulation Modeling*, Anylogic North America, 60-63.

Carson II J.S. 2004. "Introduction to Modeling and Simulation." In *Proceedings of the 2004 Winter Simulation Conference*. 9-1.

Galster,S. et al. 2007. "Human Automation Integration." In Uninhabited Military Vehicle(UMVs): Human Factor Issues in Augmenting the Forces, The Research and Technology Orgnisation(RTO) of NATO, 7-1~7-11.

Hendy, K. C. and P. S. E. Farrell. 1997. "Implementing a Model of Human Information Processing in a Task Network Simulation Environment." Canadian Department of National Defence, Defence and Civil Institute of Environmental Medicine Technical Report DCIEM No. 97-R-71. North York, Ontario, Canada M3M 3B9.

Macal C.M. and North M.J. 2013. "Introductory Tutorial: Agent-Based Modeling and Simulation." In *Proceedings of the 2013 Winter Simulation Conference*. 362-376.

McCrasken,J.H. and Aldrich,T.B. 1984. "Analysis of selected LHX mission fuctions: Implications for operator workload and system automation goals." Technical Note ASI479-024-84. Fort Rucker, AL: Army Research Institute Aviation Research and Development Activity.

Parasuraman, R. et al. 2000. "A model for types and levels of human interaction with automation." *IEEE Transactions on Systems, Man and Cybernetics*, 30(3), 286-297.

Pew, R. W. and Mavor. 1998. *Modeling human and organizational behavior.* (Eds.) National Research Council: Washington, DC. National Academy Press.

Rasmussen, J. 1983. "Skills, rules and knowledge; signals, signs, and symbols, and other distinctions in human performance models." *IEEE Transactions on Systems, Man, and Cybernetics*, 13, 257-266.

Rothenberg J. 1989. "Object-Oriented Simulation: Where Do We Go from Here ?." A RAND NOTE MDA 903-85-C-0030. The RAND Corporation, Santa Monica, USA.

Sheridan, T.B. 1988. " The system perspective." In: Wiener, E.L. and Nagel, D.C. (Eds.) Human factors in aviation. San Diego: Academic Press.

Wickens, C.D. 1984. "Processing Resources in Attention." In Varieties of Attention. Parasuraman R. and Davies D.R. (Eds.) New York: Academic Press, 63-102.

Wickens et al. 2013. Engineering Psychology and Human Performance. New York: Pearson Education, Inc.
Wiener, E.L. 1988. "Cockpit automation." In: E.L. Wiener and D.C. Nagels (Eds.) Human factors in aviation . San Diego: Academic, 433-461.

AUTHOR BIOGRAPHIES

SANG YEONG CHOI was born in Busan, Korea and went to Korean Military Academy, Korean Natinoal Defense University, Cranfield University of UK(Royal Military College of Science), where he studied defense science and obtained Ph.D. in 1989. He is a research professor in The Specialized Research Center For Future Ground System Analysis, MyongJi University. Previously, he was a professor at Korean National Defense University. His e-mail address is : metayoung@gmail.com.

KANG PARK was born in Busan, Korea and went to Seoul National University, Pennsylvania State University, where he studied mechanical engineering and obtained Ph.D. in 1996. He is a professor in the department of mechanical engineering, MyongJi University. Currently, he is a head of The Specialized Research Center For Future Ground System Analysis. His e-mail address is : kxp007@gmail.com.

JIHYUN YANG was born in Seoul, Korea and went to Seoul National University, Massachusetts Institute of Technology, where she studied Aeronautics and Astronautics and obtained Ph.D. in 2007. She is an assistant professor in the department of automotive engineering, Kookmin University. Her e-mail address is : yangjh@kookmin.ac.kr

HWAN IL KANG was born in Incheon, South Korea and graduated from the Seoul National University, Korea Advanced Institute of Science & Technology and University of Wisconsin-Madison in USA where he studied electrical and computer engineering and obtained his BS, MS and Ph. D. degrees in 1980, 1982 and 1990, respectively. He has been a professor at dept. of Information & Communications eng. in Myongji University, South Korea since 1996. His research interest includes modelling & simulation, military sciences and control applications. His e-mail address is : hwanilkang@naver.com and his Web-page can be found at http://www.mju.ac.kr/~hwan.