Chapter 1

ENABLING THE EXPLORATION OF OPERATING PROCEDURES IN CRITICAL INFRASTRUCTURES

Christos Siaterlis, Béla Genge, Marc Hohenadel, and Marco Del Pra

Abstract Modern testbeds for the experimental analysis of CIs either totally ignore the human factor or include real Human Machine Interfaces (HMIs) and software that require the presence of real human operators during an experiment. Although experimentation with human-in-the-loop can provide invaluable experimental data for human decision making and reactions, it would be impossible to do a systematic exploration of the vast parameter space in terms of possible human operator decisions, reasoning and actions. Therefore, in this paper we argue that existing testbeds should include simulated human decision-making capabilities in order to close this important loop that plays a crucial role in the outcome of cyber security experiments involving CIs. Furthermore, we propose an extension of our previously developed experimentation framework with generic *Human Decision* units that enable the integration of HMI and human operator models. The developed prototype was evaluated by assessing the impact of different human operator reactions during an attack against a cyber-physical infrastructure that includes the IEEE 30-bus power grid model.

Keywords: Critical Infrastructures, Operating Procedures, Security, Simulation

1. Introduction

Most investigations focusing on Critical Infrastructures (CIs), e.g., power plants, water plants and smart grids, highlight the fact that human operators play a crucial role in the outcome of cyber security incidents [8]. For instance, simple configuration mistakes that leave systems completely unprotected are most of the times uncovered after major security incidents. On the other hand, human decisions can make the difference between a complete break-down and a survival of the system.

Nowadays, the interaction of human operators with CIs is mostly implemented through Information and Communications Technologies (ICT) since ICT can lead to cost reduction as well as greater efficiency, flexibility and interoperability between components. Consequently, we find several approaches dealing with the design of (graphical) interfaces, also known as Human Machine Interfaces (HMIs), that assist the decision-making process and that reduce the reaction time of human operators [19, 13]. On the other hand, human operators can also interact with the system independently of any HMIs, e.g., by switching ON/OFF devices. Therefore, we can clearly state that testbeds focusing on the analysis of theses systems should also take into account the presence of human operators.

In this paper we argue that the presence of human operators and HMIs changes dramatically the system's behavior and scientists should take them into account when designing testbeds for analyzing CIs. We show that existing testbeds [4, 14, 6, 12] account for real HMI and real human operators, however, they do not include software simulations of these components. Although with real HMI and human operators these testbeds would provide reliable experimental data, they would hardly be able to support exhaustive parameter tests. This is mainly due to the costs involved in acquiring and training human operators, combined with the costs of customizing proprietary HMI software. In order to prove the importance of human operators and particularly of operating procedures in security experiments, we propose an extension of the framework developed in our previous work [7]. The proposed extension includes a *Human Decision* unit (HD unit) that is able to run HMI and operator models in real time. Actions issued by these models are translated into commands that are executed in the cyber and physical realms. This way, the proposed approach enables the recreation of realistic scenarios in which operators do not interact only with the cyber realm, but they also take actions on the physical realm, e.g., switch ON/OFF PCs. Although the extension focuses more on operating procedures and less on human operators, in the context of CIs, human operators must follow well-established procedures in case of contingencies. Nevertheless, the complexity of human operators should not be neglected and we consider this as part of our future work. The developed prototype was evaluated by assessing the impact of different human operator reactions during an attack against a cyber-physical infrastructure including the IEEE 30-bus power grid model.

The paper is structured as follows. After a brief overview of related work in Section 1.2, we discuss the problem of adding *Human Decision* units to experimentation testbeds in Section 1.3. Our proposal is pre-

sented in Section 1.4 and a case study including the IEEE 30-bus power grid model is given in Section 1.5. The paper concludes in Section 1.6.

2. Related Work

Most of the CI experimentation testbeds available today do not account for human operator or HMI models. In contrast, we find several testbeds that involve the interaction of real human operators with the physical process through real HMIs. The most relevant approaches from both categories are discussed within this section.

Chabukswar, *et al.* [3] used the Command and Control WindTunnel [15] multi-model simulation environment, based on the High-Level Architecture IEEE standard [22], to enable the interaction between various simulation engines. The authors used OMNeT++ to simulate the network and Matlab Simulink to build and run the physical process model. In this approach, neither the human, nor the HMI are taken into account, as the main focus of the testbed is the recreation of CIs and hardware control loops. A similar approach is reported in the work of Hopkinson, *et al.* [10], in which the PowerWorld server, "a high-voltage power system simulation and analysis package" [16], provides the simulation environment of Power Systems, while NS2 is used to simulate the other components of the system, e.g., PLCs, malware. Similarly to the work of Chabukswar, *et al.*, this approach does not include human operator or HMI models. Nevertheless, it provides a generic interface for extending the testbed with additional NS2 modules. Although such approach could support the integration of external modules, it was not designed to run more complex mathematical models, as the ones that could be developed in dedicated modeling tools such as Simulink. Furthermore, designers would need to define the "glue" code between NS2 and models in order to enable the integration of a wide range of models. These aspects are the main focus of the present work that proposes a "glue" code consisting of several modules and interfaces to enable the integration of human operator and HMI models into the CI experimentation testbed developed in our previous work.

In contrast to the previously mentioned approaches we find several testbeds [4, 14, 6, 12] in which both humans and HMIs are real. In other words, in these approaches we find real human operators interacting with real proprietary HMIs. Although from one point of view such testbeds would provide reliable experimental data, they would be hardly able to support exhaustive parameter tests on human operators or HMIs. A similar approach to these is the one of Queiroz, *et al.* [17]. In this case HMIs are simulated as OMNeT++ modules, while human operators are

real humans interacting with the simulated HMIs. Although this is an obvious step forward, compared to the previously mentioned approaches, it still requires real human operators to be present and interact with the system. As already mentioned, the high fidelity of such an approach is counterbalanced by its costs and poor efficiency, as experiments might require the presence of more than one human with repetitive tasks to cover the entire parameters space.

3. Problem Statement

Today, in many fields we see an increasing trend towards replacing the human factor by automated control loops. Nevertheless, human operators continue to play a significant role in the operation of CIs and more importantly in the context of abnormal situations and contingencies.

By inspecting the CI experimentation testbeds available today [4, 14, 6, 12], we see that they account for real human operators and HMIs, and do not include simulators of these. Indeed, by looking at existing human operator models [5–1, 23, 11, 18, 9] we realize just how complex this procedure actually is. In fact, the research on designing human operator models has been around since the beginning of the 20th century [5, 1]. More recent papers have shown the applicability of linear/nonlinear control theories [11] and Belief-Desire-Intention paradigms combined with agent-based platforms [23, 18] in the human operator modeling process. All these approaches clearly show that the human operator modeling problem can be tackled in several ways. Additionally, each approach might come with its own advantages and disadvantages, making the selection process even more difficult. Therefore, the design of a generic experimentation platform with human operator and HMI models that could be applied to a wide range of CIs is not a trivial task.

Taking into account the complexity and diversity of existing CIs, together with the operator models needed for each of these systems, we can clearly state that having a single operator model applicable to all of these systems might not be feasible or even possible. On the other hand, a solution that would require the integration of every operator model from scratch could also be seen as unfeasible. A more acceptable solution could be one that provides designers several interfaces to which models could be adapted and coupled in order to be integrated into the system. These interfaces should be designed in such a manner to enable the integration of a wide variety of operator models. Furthermore, these should enable the integration of HMI models coupled to operator models, as the functionality of HMI software can also have a major impact on the state of the system.

From the perspective of human operators we should also take into account the fact that with larger infrastructures there could be more than one human operator supervising and controlling the system. In fact, a generic representation of the human decision-making process should include both hierarchical and graph-based information flows. Such complex interactions should not be excluded as they could seriously limit the applicability of the entire approach. Another important issue to be considered is the translation of simulated operator decisions to actions on the cyber-physical realm. Such actions could include not only interactions with the physical process, but interactions with the cyber realm as well, e.g., configure firewall, launch an external script, shut down a PC. The translation process should be flexible enough and should rely on generic modules that are easily replaceable with others. This way, the implemented solution would enable experimentation with a wide range of physical processes and Networked Industrial Control Systems (NICS) installations.

To conclude, as shown in the analysis included in this section, extending existing CI experimentation testbeds with human operator and HMI models is not a trivial task. Nevertheless, the advantages of such capabilities for any experimentation environment exceed its disadvantages. Furthermore, by adding human operator and HMI models, the experimentation environment will be able to close an important loop that plays a crucial role in the outcome of cyber security experiments involving CIs.

4. Proposed Approach

In this section we present the proposed approach for integrating human decisions into the experimentation framework developed in our previous work [7]. We start with a short overview of the experimentation framework and we continue with the proposed extension.

4.1 Experimentation Framework Architecture

The experimentation framework developed in our previous work [7] follows a hybrid approach, where the Emulab-based testbed recreates the control and process network of NICS, including PLCs and SCADA servers, and a software simulation reproduces the industrial processes. The architecture, as shown in Figure 1, clearly distinguishes 3 layers: the cyber layer, the physical layer and a link layer in between. The cyber layer includes regular ICT components used in SCADA systems, while the physical layer provides the simulation of physical devices. The

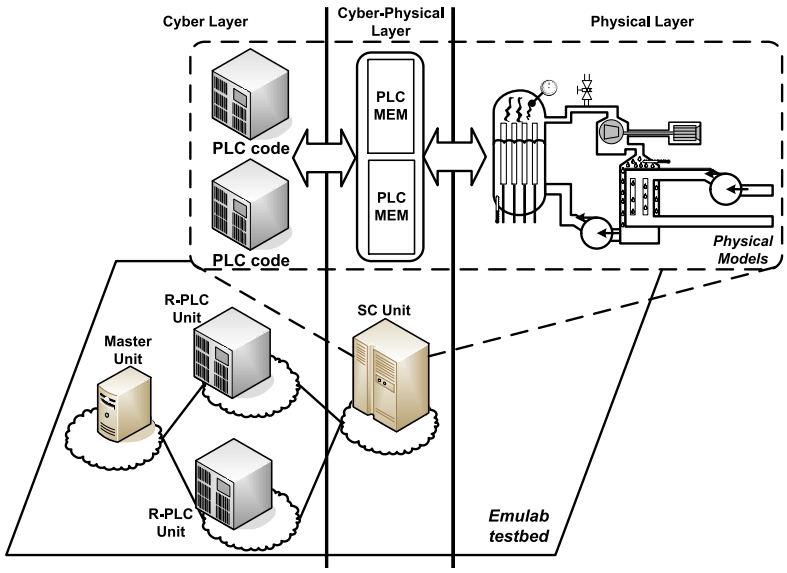


Figure 1: Overview of the experimentation framework developed in our previous work

link layer, i.e., cyber-physical layer, provides the "glue" between the two layers through the use of a shared memory region.

The physical layer is recreated through a soft real-time simulator that runs within the SC (Simulation Core) unit and executes a model of the industrial process. The cyber layer is recreated by an emulation testbed that uses the Emulab architecture and software [21] to automatically and dynamically map physical components, e.g., servers, switches, to a virtual topology. Besides the process network, the cyber layer also includes the control logic code that in the real world is run by PLCs. The control code can be run sequentially or in parallel to the physical model. In the sequential case, a *tightly coupled* code (TCC) is used, i.e., code that is running in the same memory space with the model, within the SC unit. In the parallel case a *loosely coupled* code (LCC) is used, i.e., code that is running in another address space, possibly on another host, within the R-PLC unit (Remote PLC). The main advantage of TCCs is that these do not miss values generated by the model between executions. On the other hand, LCCs allow running PLC code remotely, to inject (malicious) code without stopping the execution of the model, and to run more complex PLC emulators. The unit that implements global decision algorithms based on the sensor values received from the R-PLC units is also present in the experimentation framework as the *Master*

unit. The cyber-physical layer incorporates the PLC memory, seen as a set of registers typical of PLCs, and the communications interfaces that "glue" together the other two layers. Memory registers provide the link to the inputs, e.g., valve position, and outputs, e.g., sensor values, of the physical model.

Prototypes of SC, R-PLC and Master Units have been developed in C# (Windows) and have been ported and tested on Unix-based systems (FreeBSD, Fedora and Ubuntu) with the help of the *Mono* platform. Matlab Simulink was used as the industrial process simulator (physical layer). From Simulink models the corresponding 'C' code is generated using Matlab Real Time Workshop. The communications between SC and R-PLC units is handled by .NET's binary implementation of RPC (called *remoting*) over TCP. For the communications between the R-PLC and Master units, we implemented the Modbus over TCP protocol.

4.2 Architecture Extension

As already mentioned in the previous sections, adding human decisions to experimentation testbeds has several advantages that can not be neglected. However, the large number of available techniques for modeling the complexity of human operators requires an approach that does not limit the testbed to one specific model. Therefore, in this section we propose a generic approach to integrate models into cyber-physical testbeds. In this approach models are seen as "black-boxes" that must implement a standard interface in order to exchange actions with the other components of the system. The required interface includes a set of inputs and a set of outputs, connected to *action scripts* at run-time. Consequently, all actions issued/received by models are sent through *action scripts* that include the necessary code for processing actions and communicating with other software components.

In order to validate the proposed approach we extended the framework developed in our previous work with a generic *Human Decision* (HD) unit. In the remaining of this section we provide a detailed presentation of this prototype and we provide more insight into the implementation specifics of the proposed technique.

The design of the HD unit started from the assumption that human operators interact with cyber-physical systems in several ways. First of all, they rely on ICT hardware and software that are typically found in NICS installations, e.g., HMI. However, they can also interact independently of these through installation-specific components, e.g., customized scripts to configure a firewall, or even through physical actions, e.g., shutting down a server. Therefore, the architecture of the Hu-

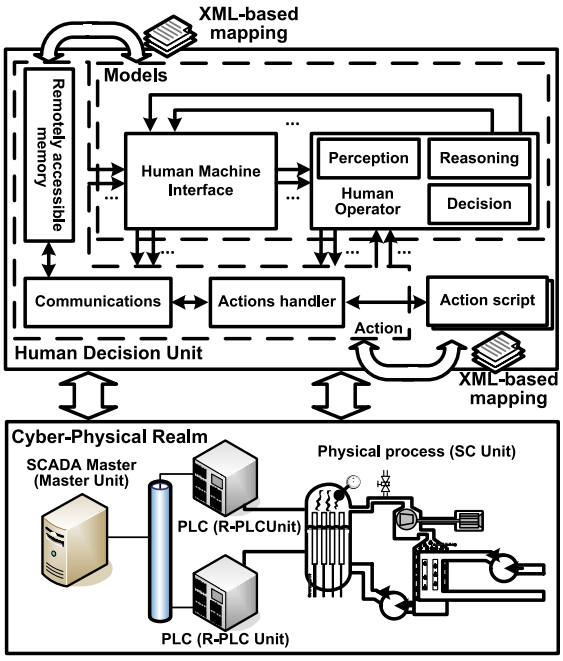


Figure 2: Extended architecture with Human Decision units

man Decision (HD) unit takes all these aspects into account through the *Actions* "glue" layer and through *Action scripts* that implement the specifics of each experiment.

The proposed architecture, including not only the HD unit but other components typical to NICS is depicted in Figure 2. Within the HD unit, the HMI and human operator models interact with the cyber-physical realm through an *Action* module that enables bidirectional communications and execution of commands, i.e., *actions*. The HMI and human operator models depicted in Figure 2 interact in real-time using direct connections included in the model. Although both are constructed according to the specifications of each experiment, for completeness we included three building blocks into the human operator model: *Perception*, *Reasoning* and *Decision*. These can be viewed as basic human operator functionalities that could be taken into account when building similar models. However, they can be replaced with other blocks, depending on the requirements of each experiment.

As shown in Figure 2, human operator models receive events from the *Action* module directly and through HMI models. The first case represents the direct interaction of human operators with the cyber-

physical realm, while the second case represents interactions through the HMI. In both cases the inputs denote the data used by the models in each time step, e.g., measured voltage, while the outputs denote *actions* to be executed. Each action includes an identifier and several parameters, e.g., open/close valve. These are written to the *Remotely accessible memory* module from where they are read by the *Actions handler* module. Then, based on external XML configuration files, the *Actions handler* module runs a specific *Action script* identified by a numeric identifier.

Action scripts are written in the 'C' language and are loaded as external binary libraries. These can be specific to each experiment and can include commands that pass values to other components of our framework, e.g., to the physical model, or for launching additional scripts, e.g., configuring a firewall, turning ON/OFF specific machines. This way, the HD unit provides a flexible approach to translate model-specific outputs to real actions that affect not only the physical realm, but the cyber realm as well.

Finally, we mention that the role of the *Remotely accessible memory* module is not limited to passing values between the internal modules of the HD unit, but it also serves as a way for external components to interact with the HMI and human operator models. By enabling remote access to this memory region, external software components, e.g., other HD units and SCADA Masters, can communicate with the HD unit using simple memory access operations. In these cases requests are received and processed by the *Communications* module that passes the received values to the *Remotely accessible memory* module. From there, they are read by the *Models* module and are provided to the HMI and human operator models. For flexibility reasons we use external XML configuration files to map memory regions to model inputs and to map model outputs to memory regions.

Similarly to our previous work, we implemented a prototype of the HD unit in C# (Windows) and we have ported it to Unix-based systems with the help of the *Mono* platform. At this stage of our work, communications with other HD and Master units are handled by the Modbus protocol, as this protocol was specifically designed for exchanging memory-mapped data between units. Nevertheless, depending on future needs other protocols could be implemented as well by replacing the Modbus handler units.

Human operator and HMI models were implemented in Matlab Simulink, since this is a general simulation environment for dynamic and embedded systems. Its toolboxes, e.g., control systems, neural networks, provide a powerful support for modeling and simulation. From Simulink models the corresponding 'C' code is generated using Matlab RTW. The gener-

ated code is then integrated into the extended framework using an XML configuration file. The model is then able to interact in real-time with the other components of the system.

5. Case Studies

As already stated in this paper, the importance of human operators in the normal operation of CIs can not be neglected neither in real installations, nor in experimentation testbeds. Therefore, the proposed Human Decision unit brings new elements that close an important loop in existing testbeds. In this section we prove with experimental results the applicability of the proposed approach and the importance of human operators in the context of cyber attacks targeting a simulated power grid.

The power grid model we employed for this experiment is the well-known IEEE 30-bus test system that includes 6 power generators and a total of 20 loads distributed on 20 buses. For this model we defined four regions, each controlled by a different human operator. One of the main goals of the implemented scenario is to show that disturbances can be balanced by countermeasures taken by power grid operators. Furthermore, the results also confirm the fact that with such interconnected power grids, operators from different regions need to collaborate in order to restore the normal operation of the entire grid.

5.1 Experimentation Scenarios

As shown in Figure 3, within the employed IEEE 30-bus power grid model we defined four regions that minimize the number of connections, i.e., transmission lines, between regions. In real scenarios operators might use other criteria in the definition of these regions, e.g., balanced power and loads. Nevertheless, the main focus of this paper is not the definition of regions, but the recreation of a possibly realistic scenario with multiple operators that need to cooperate in order to keep the power grid within its normal operating limits.

In the implemented scenario we assume that the adversary is able to compromise the entire ICT infrastructure in region 3 by social engineering and by exploiting the vulnerabilities of SCADA protocols [2]. Then, the same adversary employs a coordinated worm-based attack to trigger synchronized events in control devices. These turn ON large consumers at the same time within two substations (a total of 120MW), i.e., 12 and 20, and cause a disturbance that decreases voltages below their operating limits, i.e., below 0.95 p.u.. If this scenario would happen in reality, it could lead to large-scale black-outs and malfunctioning hardware, but

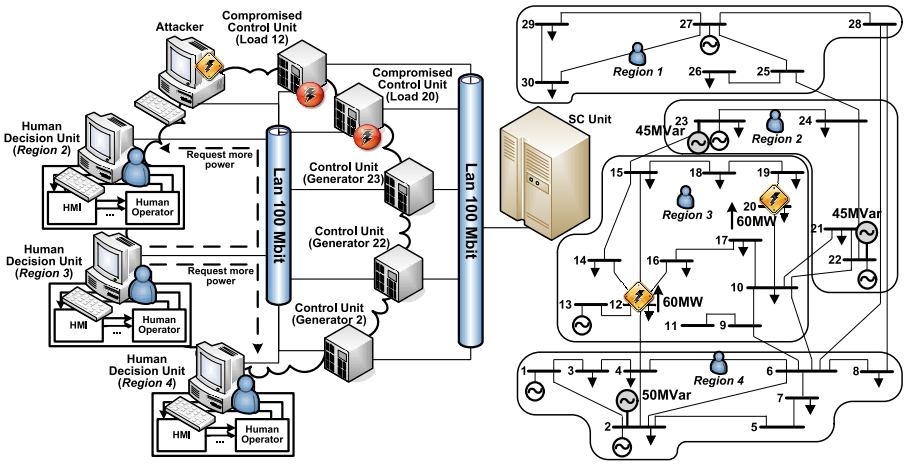


Figure 3: Experiment setup and defined regions for the IEEE 30-bus test system

it could also have other cascading effects that could spread to the entire grid.

As operators in region 3 loose control over their infrastructure, we further assume that they might request the assistance of operators from neighboring regions. Based on this request, operators in regions 2 and 4 inject additional power into the system and, as shown by experimental results, they manage to stabilize the voltages in region 3. Next, we provide a brief description of operator reaction cases implemented within the described scenario.

Case 1: Operators in region 3 loose control and seek no assistance. In this case operators loose control over their infrastructure and do not request any assistance from neighboring operators. As shown by the results, in such cases voltages decrease well below their normal operating limits and operators would need to react quickly in order to prevent serious damages to physical devices.

Case 2: Operators in region 3 loose control and request assistance from operators in region 4. This case shows that the disturbance originating in region 3 can be counterbalanced by measures taken in other regions, i.e., region 4. In this case, operators in region 3 contact operators in region 4 in order to increase the production of energy and finally to increase the power injected into the grid. Consequently, operators in region 4 start back-up generators and increase production by 50 MVar. Although the effects are limited, this case shows the importance of collaborations between operators.

Case 3: Operators in region 3 loose control and request assistance from operators in region 4 & 2. As the actions taken in the previous case had limited effects, operators need to request additional assistance from operators in region 2. Therefore, operators increase the production of energy by 90 MVar in region 2. This finally stabilizes the bus voltages in region 3.

5.2 Experiment Setup and Models

The experimental scenario was implemented in the Joint Research Centre's (JRC) Experimental Platform for Internet Contingencies (EPIC) laboratory [20]. The Emulab testbed included nodes with the following configuration: FreeBSD OS 8, AMD Athlon Dual Core CPU at 2.3GHz and 4GB of RAM. The experiment setup is depicted in Figure 3.

Models have been developed using the "black-box" approach described in the previous sections. For each operator we developed a Simulink model with inputs and outputs connected to action scripts.

For region 3 the operator model has one input, i.e., the status of the link, and three outputs, i.e., the action identifier and two parameters denoting the request for assistance. In this case the input action script continuously tests the link between the HD unit and the physical process, i.e., the SC unit. In case the link is ON, the action script provides a value of '1' to the model, and '0' otherwise. The model takes this input and forwards its negated value to the output, together with the action identifier. The output action script takes these two values and forwards them to the HD units running in region 2 and 4. In our implementation a value of '1' means that there is a need of assistance for operator in region 3. The functions we implemented are different for each case: $f_1(x) = (ID, 0, 0)$ (*Case 1*), $f_2(x) = (ID, !x, 0)$ (*Case 2*), and $f_3(x) = (ID, !x, !x)$ (*Case 3*), where x is the model input and ID is the action identifier.

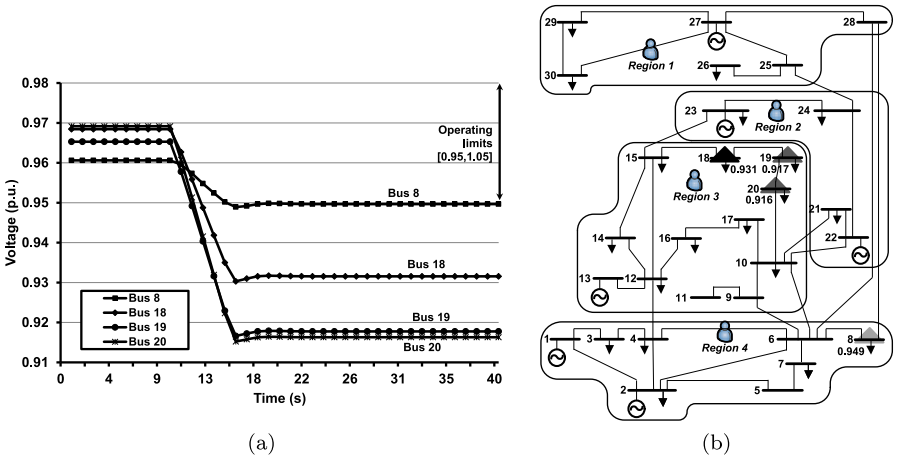
In region 4 the operator model has one input, i.e., the status of assistance request (0/1), and two outputs, i.e., the action identifier and one parameter denoting the power injected by back-up generators, i.e., 50 MVar. The model receives its input value from the HD unit in region 3 and produces an output that is sent to the SC unit and finally to the physical process. The mathematical function for this model is $f(x) = (ID, (x = 1)?\ 50 : 0)$, where x is the model input, and ID is the action identifier.

The model for the operator in region 2 is similar to the operator model in region 4 in the sense that it takes the same input, but it produces one additional output. The first output is the action identifier, while

the second and the third outputs are MVar-s produced by two back-up generators. In our scenario we used two back-up generators in region 2 to produce a total of 90 MVar, i.e., 2x45 MVar. The mathematical function for this model is $f(x) = (ID, (x = 1)?\ 45 : 0, (x = 1)?\ 45 : 0)$, where x is the model input, and ID is the action identifier.

5.3 Experimental Results

Following the description of the experimental scenario and setup, in this subsection we present the results obtained for each of the three cases.

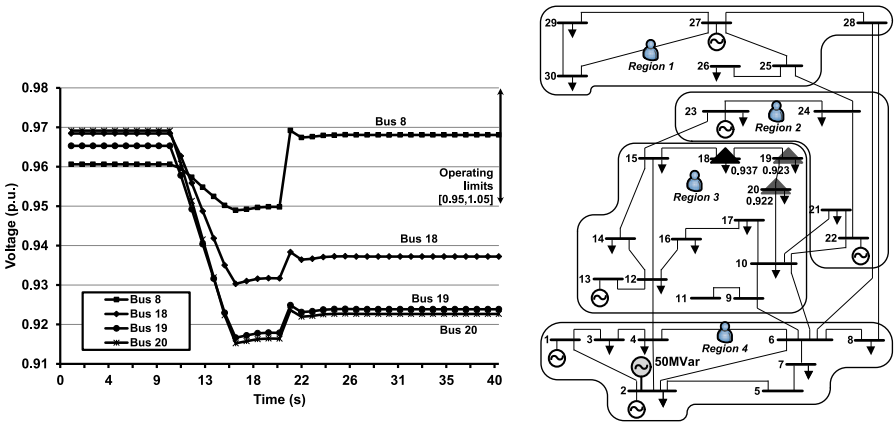


(a) (b)

Figure 4: Effect of the attack in *Case 1*: (a) time series of affected bus voltages (b) regional view

Case 1. Immediately after the attack is started on buses 12 and 20, voltages begin to drop. For 3 buses (18, 19, and 20) voltages fall well below the operating limit of 0.95 p.u. to almost 0.91 p.u.. The disturbance propagates to region 4 as well, where it causes a drop of the voltage on bus 8 slightly below the operating limit (to 0.949 p.u.). This effect is also shown in Figure 4 (a) and (b), where we can see that without any intervention, the disturbance causes severe changes in the voltages, mostly located in region 3.

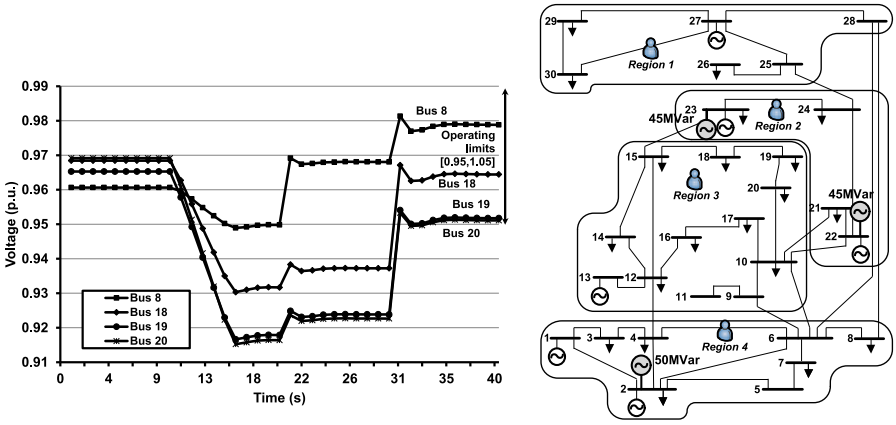
Case 2. Next, we assume that operators are able to get assistance from region 4, where an additional back-up generator injects 50 MVar into the grid. This action has a significant effect on bus 8, where the voltage increases above the operating limit. However, the effect is less-



(a) (b)

Figure 5: Effect of the attack in *Case 2*: (a) time series of affected bus voltages (b) regional view

significant on the other buses, where voltages still remain below 0.95 p.u. (Figure 5).



(a) (b)

Figure 6: Effect of the attack in *Case 3*: (a) time series of affected bus voltages (b) regional view

Case 3. Finally, in the last case operators request the assistance of operators in regions 4 & 2. While operators in region 4 inject an

additional power of 50 MVar using one back-up generator, operators in region 2 start 2 back-up generators and inject a total of 90 MVar. As we can see in Figure 6, this causes an immediate increase of voltages above the operating limit. Consequently, although operators in region 3 have loosed the control over their region, they still manage to ensure the stability of their region by cooperating with neighboring regions.

The previous results confirmed the fact that human operators are indispensable elements of security studies involving CIs. Furthermore, by comparing results from case 3 to the ones from case 1 and 2, we can clearly state that with today's highly interconnected power grids operators need to collaborate in order to balance regional disturbances and to ensure the voltage stability of the entire grid. On the other hand, the results also confirmed the fact that the proposed approach can be used to recreate complex scenarios involving not only the cyber & physical realms but also human operators. This closes an important loop that can finally have a significant impact on cyber-physical security experiments.

6. Conclusions

In this paper we showed that human operators continue to play an important role in the operation of today's Critical Infrastructures (CIs). Although the industry is moving rapidly towards fully automated control loops, human operators are still the most important actors in abnormal situations and contingencies. Therefore, the goal of this paper was to show the importance of adding operating procedures to existing cyber-security experimentation testbeds and to propose a new method to integrate existing human operator models. Our main contribution is the design of a new decision unit that integrates models based on a "black-box" approach. More specifically, as long as models implement a well-defined interface consisting of a set of input and output signals, the proposed unit can incorporate generic models irrespective of their content. The paper also showed through several case studies involving a prototype of the proposed decision unit and a model of the IEEE 30-bus test system, that the approach is suitable for cyber-security experiments. Furthermore, the experimental results confirmed an already well-known fact that in today's large-scale interconnected CIs involving multiple operators, it is crucial that operators communicate and exchange information in order to ensure the global stability of the physical process. Therefore, existing testbeds will need to take into account not only single isolated human operators, but several operator networks that need to communicate with each other in order to have a global view of the

system. Consequently, as future work we intend to explore the complexity of operator networks and the vulnerability of the CIs they rely on to exchange information, e.g., public internet, DNS service. This, unavoidably, will lead to the need to incorporate more realms into the same system and to recreate more complex scenarios in which one realm might have a cascading effect on other realms, e.g., the effect of contingencies in mobile communications on the operation of the power grid.

References

[1] G.A. Bekey and C.B. Neal, Identification of sampling intervals in sampled-data models of human operators, *IEEE Transactions on Man-Machine Systems*, pp. 138–142, 1968.

[2] E. Bompard, P. Cuccia, M. Masera, and I. Nai Fovino, Cyber vulnerability in power systems operation and control, *Critical Infrastructure Protection, Lecture Notes in Computer Science*, pp. 197–234, 2012.

[3] R. Chabukswar, B. Sinopoli, B. Karsai, A. Giani, H. Neema, and A. Davis, Simulation of network attacks on SCADA Systems, *1st Workshop on Secure Control Systems, Cyber Physical Systems Week*, 2010.

[4] W. Chunlei, F. Lan, and D. Yiqi, A simulation environment for SCADA security analysis and assessment, *In Proc. of the 2010 International Conference on Measuring Technology and Mechatronics Automation*, pp. 342–347, 2010.

[5] K.J.W. Craik, Theory of the human operator in control systems, *British Journal of Psychology*, pp. 142-148, 1947.

[6] C.M. Davis, J.E. Tate, H. Okhravi, C. Grier, T.J. Overbye, and D. Nicol, SCADA cyber security testbed development, *38th North American Power Symposium*, pp. 483–488, 2006.

[7] B. Genge, I. Nai Fovino, C. Siaterlis, and M. Masera, Analyzing cyber-physical attacks on networked industrial control systems, *Critical Infrastructure Protection*, pp. 167–183, 2011.

[8] A.V. Gheorghe, M. Masera, M. Weijnen, and L. De vries, *Critical Infrastructures at Risk: Securing the European Electric Power System*, Dordrecht, The Netherlands, 2006.

[9] R. Harris, J.D. Kaplan, C. Bare, H. Iavecchia, L. Ross, D. Scolaro, and D. Wright, Human Operator Simulator (HOS). IV User's Guide, 1989.

[10] K.M. Hopkinson, K.P. Birman, R. Giovanini, D.V. Coury, X. Wang, and J.S. Thorp, EPOCHS: integrated commercial off-the-shelf soft-

ware for agent-based electric power and communication simulation, *In Proc. of the 2003 Winter Simulation Conference*, pp. 1158–1166, 2003.

[11] T. Ivancevic and B. Jovanovic, Human operator modeling and lie-derivative based control (http://arxiv.org/pdf/0907.1206.pdf), 2009.

[12] M.J. McDonald, G.N. Conrad, T.C. Service, and R.H. Cassidy, Cyber effects analysis using VCSE, *Technical Report, SAND2008-5954, Sandia National Laboratories*, USA, 2008.

[13] F. Moussa, C. Kolski, and M Riahi, A model based approach to semi-automated user interface generation for process control interactive applications, *Interacting with Computers*, vol. 12(3), pp. 245–279, 2000.

[14] I. Nai Fovino, M. Masera, L. Guidi, and G. Carpi, A simulation environment for SCADA security analysis and assessment, *In Proc. of the 3rd International Conference on Human System Interaction*, pp. 679–686, 2010.

[15] S. Neema, T. Bapty, X. Koutsoukos, H. Neema, J. Sztipanovits, and G. Karsai, Model based integration and experimentation of information fusion and C2 systems, *In Proc. of the 12th International Conference on Information Fusion*, pp. 1958–1965, 2009.

[16] PowerWorld server (http://www.powerworld.com), 2012.

[17] C. Queiroz, A. Mahmood, J. Hu, Z. Tari, and X. Yu, Building a SCADA security testbed, *In Proc. of the 2009 Third International Conference on Network and System Security*, pp. 357–364, 2010.

[18] A. Rao and M. Georgeff, BDI agents: from theory to practice, *In Proc. of the first international conference on multiagent systems ICMAS95*, pp. 312-319, 1995.

[19] R.W. Reeder and R.A. Maxion, User interface defect detection by hesitation analysis, *in Proc. of the International Conference on Dependable Systems and Networks*, pp. 61–72, 2006.

[20] C. Siaterlis, A.P. Garcia, and B. Genge, On the use of Emulab testbeds for scientifically rigorous experiments, *IEEE Communications Surveys and Tutorials*, Accepted, 2012.

[21] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, An integrated experimental environment for distributed systems and networks, *In Proc. of the 5th Symposium on Operating Systems Design and Implementation*, pp. 255–270, 2002.

[22] S. Xiaoxia and Z. Qiuhai, The introduction on high level architecture (HLA) and run-time infrastructure (RTI), *SICE 2003 Annual Conference*, pp. 1136–1139, 2003.

[23] X. Zhao, J. Venkateswaran, and Y.J. Son, Modeling human operator decision-making in manufacturing systems using BDI agent paradigm, *IIE Annual Conference and Exposition*, 2005.