Position Papers of the Federated Conference on Computer Science and Information Systems pp. 107–114
DOI: 10.15439/2015F414
ACSIS, Vol. 6

Modeling Resiliency and Its Essential Components for Cyberphysical Systems

Janusz Zalewski
Software Engineering Dept.
Florida Gulf Coast Univ.
Ft. Myers, FL 33965
USA
zalewski@fgcu.edu

Steven Drager
William McKeever
Air Force Research Lab
Rome, NY 13441, USA
Steven.Drager@us.af.mil,
William.McKeever.1@us.af.mil

Andrew J. Kornecki
ECSSE Department
Embry-Riddle Aero. Univ.
Daytona Beach, FL 32114
USA
kornecka@erau.edu

Bogdan Czejdo
Dept. of Math. & CS
Fayetteville State Univ.
Fayetteville, NC 28301
USA
bczejdo@uncfsu.edu

***Abstract*—This paper presents an initial approach related to modeling resiliency for cyberphysical systems. It discusses the concept and definitions of resiliency and outlines the process of building a model of resiliency. Through analogies with feedback control and fault tolerance, the Design for Resilience is addressed, where the design of the controller component of a cyberphysical system needs to account for potential safety hazards and security threats, with awareness of its internal faults and vulnerabilities. This model is validated against other approaches to modeling resilience described in the literature, followed by a discussion of the resilience metrics. The paper concludes with presenting the strategy of modeling resiliency, based on the assumption that one cannot guarantee absolute protection against attacks, or failures, but can aim at providing successful recovery after disruptions. With safety and security as essential resiliency components, an extended model is proposed involving an attacker, suggesting appropriate performance metric reflecting the distance between the normal state and the degraded state. A model-based environment Möbius, from the University of Illinois, is considered in helping to evaluate resiliency under various operational scenarios.**

I. Introduction

ALTHOUGH resiliency is essentially a concept adopted in medicine and health science [1] and relates to patient's resistance in response to disease, in common sense, resiliency (or resilience) is often associated with natural or ecological systems demonstrating tolerance to, and respective recovery from, disasters, such as earthquakes, floods, hurricanes, etc. [2]. The concept has been also extended to human-made environments, such as supply chains [3], transportation networks [4], military operations [5], etc., which are called resilient if they can tolerate some major failures or disruptions and smoothly return to normal operational capability. Recent books in systems engineering take note of additional aspects of resiliency, including redundancy [6], adaptability [7], and safety as the ability to succeed under varying conditions [8].

This project has been funded in part by the 2014 Visiting Faculty Research Program at the Air Force Rome Labs. Case Number 88ABW-2015-3306 – 26 June 2015. Distribution unlimited.

In computing, and in cyberphysical systems in particular, the term resilience has been adopted to describe the computer system's ability to restore its original functionality after a loss [9]-[11]. In a contemporary world, it concerns primarily computer networks and cybersecurity, applied in various types of systems, from critical infrastructure [12] to space systems [13], and more.

Resiliency landscape up to early 2011 has been covered in MITRE report [14], which listed approximately 320 articles, divided in eight categories, one of them particularly relevant to modeling, resiliency metrics. In current work, a number of more recent papers (dated 2011 and later) were analyzed, with respect to models of resiliency.

The particular objective of this work is to address resiliency assessment of cyberphysical systems in response to multiple external disturbances and internal parameter fluctuations, as follows:

- Identify critical components of resiliency, beyond security, such as reliability, safety, etc.
- Develop a process for resiliency assessment in cyberphysical systems.
- Apply it to the control-theoretic model of a resilient architecture to assess its resiliency.

The rest of the paper is structured as follows. Section II discusses in detail the concept of resiliency, Section III describes the adopted model of resiliency, followed by its expansion in Section IV and a discussion of metrics and measures in Section V. Section VI presents the modeling strategy, and Section VII constitutes the conclusion.

II. The Concept of Resiliency

To set the stage for serious research on resiliency, some fundamental issues of understanding the concept must be resolved. For example, some authors [12] look at the assessment of resiliency (calling it resilience) from two perspectives: design methods and system operation:

> *Cyber resilience design methods consider primarily how system architecture and activities enhance the resilience of the system to cyber threats. The second*

category, operational resilience assessment methods […] consider physical threats and accidents, in addition to cyber threats.

The architectural view is advocated in [15], stating that "Architectural resiliency is the ability of an architecture – for an enterprise, a mission / business segment, a system-of-systems, a family of systems, or an individual system or component – to enable missions (including cyber defense missions) to anticipate, withstand, recover from, and evolve to address more effectively, cyberdomain attacks."

On the other hand, operational resilience is discussed in detail in [10], as opposed to enterprise resilience, which in fact is consistent with the architectural perspective. Operational resilience, adopting definition from [16], is understood as "the organization's ability to adapt to risk that affects its core operational capacities. […] A subset of enterprise resilience, operational resilience, focuses on the organization's ability to manage operational risk, whereas enterprise resilience encompasses additional areas of risk such as business risk and credit risk".

In a different perspective, Madni and Jackson [17] describe resilience as the ability to bounce back after a shock or disturbance, and consider it as a "multi-faceted capability of a complex system that encompasses avoiding, absorbing, adapting to, and recovering from disruptions."

To summarize various understandings of resilience, one can see that some views consider resilience as a state of a system, and some others see it as a system property, calling it ability. These two different, although overlapping, ways of studying resilience are adopted in this work. One notion of resiliency relating to the concept of system state encompasses the system architecture view and operational view, and answers the question:

How to build or operate a computing system to make it resilient?

The second notion of resiliency relates to it as an ability, or system property (attribute), and answers a different question:

To what extent (or to what degree), an existing computing system is resilient?

This dual understanding of a concept of resiliency, one based on studying system state (its architecture and/or operation) and the other based on studying a system attribute or property is adopted in this work, with focus on studying and modeling resiliency as a property.

Such dualism in understanding a system related concept is not that uncommon, as it may look at the first glance, and has further consequences. Since there are essentially two notions, two different definitions of these concepts are needed, and possibly two different terms to denote it:

- After [18], resilience is defined as: the ability to maintain acceptable levels of operation in the presence of abnormal conditions.
- Following this definition, we define resiliency as: the extent to which a computing system is able to maintain acceptable levels of operation in the presence of abnormal conditions.

These definitions fit into multiple others encountered in the literature. For example, Meyer defines resilience as the persistence of performability when facing changes [19]. Bishop claims [20] that "a resilient system is effectively a survivable system that is capable of restoring not only its performance level back to desired levels, but also the capacity of the system itself to recover, maintaining its ability to sustain future attacks or failures." Others define resilience or resiliency as the ability of restoring original operational capabilities (functionality) after a loss [12]. An overall consensus seems to be that resilience characterizes system's ability to conduct recovery from serious disruption.

As a final remark, regarding the subtle distinction between the notions of resilience and resiliency, one has to mention that the split into two separate notions, one based on system state and the other based on system property, is not specific to resiliency. A similar situation exists, although is rarely articulated, with the concept of security, where in addition to security understood as related to system state, there is a concept of security as a system property. The same situation exists with the two concepts of safety. In these cases, it would be proper to coin a different term for one of those close meanings, and talk about *security* and *secureness* and, correspondingly, about *safety* and *safeness*.

III. Building a Model of Resiliency

A. General Considerations

Building a model of resiliency to explore it is not a new topic. Older papers, referred to in the MITRE study [14], seem to discuss it in general terms, at the conceptual level, without even using the term model. There has been also a variety of papers published over the years, how to approach studying resilience (resiliency) and building models from the point of view called resilience engineering, for example [5], [17], [21]. All these models, however, are primarily conceptual and do not facilitate quantitative, or even qualitative, analysis of resiliency. The major shortcoming of all of such attempts and their corresponding models is the lack of mathematical underpinning.

One point that everyone agrees upon, because it is inherent in essentially all definitions of resilience or resiliency, is the illustration of divergence from desired operational conditions due to a sudden disruption, and successful recovery to the desired state, as shown in Figure 1, adopted from [20]. The model is expressed in terms of Quality of Service (QoS) varying over time, and represents a dip in performance, understood as diverging from specified operational conditions, which is caused by a sudden disruption at time A. Value of (B-A) represents the time taken for the system to return to its equilibrium state E. Value of (E-C) represents the maximum disturbance for system marked in blue. Another possible response is shown

for system marked in green. Point F represents a QoS below which the system's mission is compromised [20].

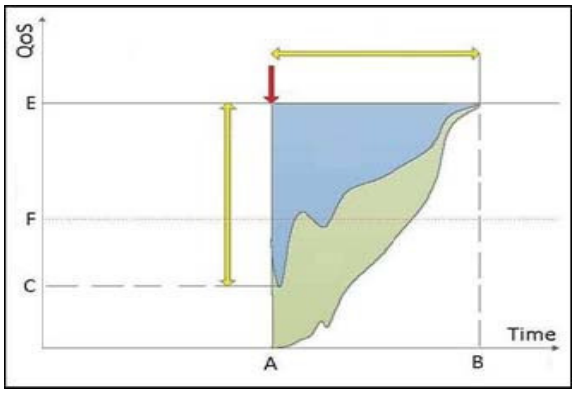

Fig. 1 Illustration of a concept of resilience to a sudden disruption

This model, being relatively widely adopted, is very illustrative for our purposes, because it splits the concept of resilience into its constituting components:

- the system's equilibrium state E
- disturbance or disruption point A
- acceptable service degradation level F
- maximum divergence from the equilibrium, E-C, and
- length of time interval to return to equilibrium, B-A.

The question is now, how to determine these points and intervals, and – once they are determined – how to develop behavioral policies or operational principles for the system to possibly anticipate the potential disruption and to respond to sudden disruptions and recover from degradation of state to fully operational conditions. The rest of this section outlines the process of building such a model.

B. Analogies with Feedback Control and Fault Tolerance

Feedback Control Analogy. Looking at the illustration in Figure 1, one can immediately find a behavioral analogy with a typical feedback control system, which is shown in Figure 2. For such system, any disruption caused by disturbances results in changes of the Measured Value, which cause its deviation from the Setpoint, represented as ε. The Controller then is responsible for following the Control Law (an algorithm, which determines respective action) and sending an appropriate Control Signal to the Controlled Object to return it to the equilibrium state, as indicated by the Measured Value. Thus, the analogy with the concept of resilience illustrated in Figure 1 can be described as follows:

- the Setpoint (desired value) in Figure 2 corresponds to the system's equilibrium E, in Figure 1
- disturbances in Figure 2 correspond to the disruption at point A in Figure 1
- the deviation from the Setpoint, ε, in Figure 2, corresponds to divergence from the equilibrium, E-C, in Figure 1
- the time constant, τ, for the control system in Figure 2, corresponds to time interval to return to equilibrium, B-A, and
- parameters such as overshoot for the control system in Figure 2 may be viewed as corresponding to acceptable service degradation level F in Figure 2.

This analogy is very instructive not only as a simple illustration of concepts. Its primary result is the conceptual formulation of the Design for Resilience problem.

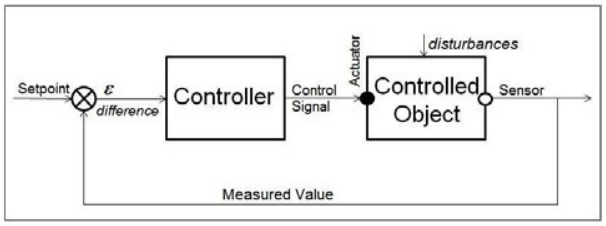

Fig. 2 Illustration of a control system influenced by disturbances

A typical control problem for the system shown in Figure 2 may be articulated as follows. For a given Controlled Object and Disturbances, design a Controller to generate Control Signal that minimizes certain characteristics of the Controlled Object expressed in terms of a Criterion (performance index) usually formulated in terms of the difference, **ε**, between the Setpoint and the Measured Value.

Obviously, strict formulation as mathematical description is needed for both the Controlled Object and Disturbances, as well as for the Criterion used as an indicator of the performance of the Controller. Then, a Control Law can be derived using, e.g., linear feedback control theory [22].

With this in mind, the problem of Design for Resilience (Feedback Control Analogy) can be formulated as follows.

Given (1) the description of the System whose resilience is of concern (analog of the Controlled Object), and (2) the characteristic of the expected Disruptions, develop a Strategy (an analog of a Control Law running on the Controller) to meet a certain Criterion (performance metric) expressed in terms of the distance from the desired state of the System.

There are more analogies between feedback control systems and resilient systems, stemming mostly from the fact that feedback control is very naturally illustrating resilience. For example, to understand effects of sudden disruptions on control systems and draw further analogies with resilient systems, one can talk about step response and impulse response functions [20], tolerating single or multiple upsets, and so on.

Fault Tolerance Analogy. Feedback control deals mostly with response to external disturbances, which are assumed to negatively affect the Controlled Object (Figure 1), be random and well characterized mathematically (for example, described by a Gaussian noise). However, all modern control systems are nowadays implemented digitally, and are

significantly expanded dealing with a User (Operator), are connected to the Network, as well as to a Database, which may be viewed as a logical extension of a single Setpoint data value. This is illustrated in Figure 3. With this complexity of controller interactions, when designing a Controller one has to take into account Controller's internal state, which may be a cause of significant disruptions to the Controlled Object, when a Controller fails. This is the subject of fault tolerance.

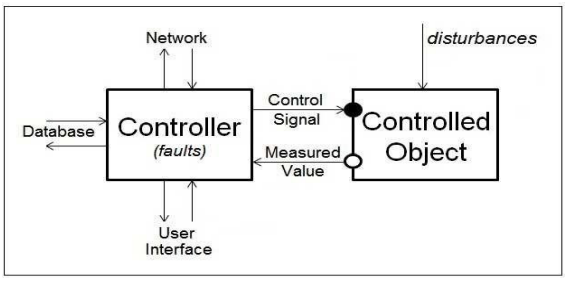

Fig. 3 Illustration of interactions in a modern control system

Fault Tolerance (FT) is a well-developed research domain [23], which has produced numerous methods, techniques and tools to deal with faults and failures. Some of the methods include: graceful degradation, diversity, redundancy, N-version programming, fail safety, and others [20]. In particular, the techniques related to FT are widely applied in dealing with faults to improve safety of cyberphysical systems: Fault Tree Analysis (FTA), Event Tree Analysis (ETA), Failure Mode and Effect Analysis (FMEA), as well as other techniques, such as Markov chains, Petri nets, Hazard and Operability Analysis (HAZOP), etc.

The subject of fault tolerance has been discussed in association with resilience, beginning as early as in 1990 [24]. One specific approach worth mentioning relates it to safety. The paper [17] states that to achieve resilience "The primary implication of external disruptions is that systems need to be built with adequate safety margins to account for uncertainty." Technically, in safety engineering, external disruptions are representing hazards and in the model from Figure 3 can be viewed as affecting the Controlled Object, as specific disturbances. Formally, a hazard is an intrinsic property or condition that has the potential to cause harm or damage [25]. To assure resilience, the Controller has to be designed to deal with safety hazards, but they are not always easy to capture and are especially difficult to account for in case of hardware or software faults.

Assuming that a fault in the Controller hardware or software, when activated, may cause a failure that will negatively affect the behavior of the Controlled Object, one can reformulate the Design for Resilience (Fault Tolerance Analogy) problem as follows.

Given (1) the description of the System whose resilience is of concern, (2) the characteristic of the expected external Disruptions, including Hazards, and (3) the characteristic of internal Faults, develop a Strategy (an analog of a Control Law running on the Controller) to meet a certain Criterion (performance metric) expressed in terms of the distance from the desired state of the System.

C. Including Security

When dealing with resilience one has to keep in mind that such discussions always involve cybersecurity [25]-[27], which is nowadays considered a primary factor in studying resilience. Nevertheless, any discussion involving security issues and its relationship to resilience is usually self-contained and almost never involves one other important constituting factor of resilience, which is safety.

One has to remember, however, that security and safety are two sides of the same coin, mutually complementary aspects of resilience. According to the International Electrotechnical Commission (IEC) [28], safety is defined as "freedom from unacceptable risk to the outside from the functional and physical units considered" whereas security is defined as "freedom from unacceptable risk to the physical units considered from the outside." Translating this into the language used in the current report:

- Safety is concerned when a Controller failure leads to severe consequences (high risk) to the environment (including Controlled Object);
- Security is concerned when a Controller failure to protect assets (a breach) leads to severe consequences (high risk) to the Controller itself (and potentially to the Controlled Object).

There are numerous definitions of security as a system property, but the one that is the most valuable should include the C+I+A (Confidentiality, Integrity and Availability) factors. In this view, the definition adopted from [29] reads as follows:

Security - the extent to which information and data are protected so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them.

A key element in this definition is "unauthorized access." From the perspective of protecting the system, this unauthorized access is called a threat. A corresponding definition taken from [28] reads as follows:

Threat - a state of the system or system environment which can lead to adverse effect in one or more given risk dimensions.

Assuming that a threat comes from the environment, as in the definition above, one can reflect it in the adjusted diagram of the control system used in the model of resilience (Figure 4). The new diagram shows that multiple Controller interfaces, the one to the Controlled Object, those to the User (Operator), the Network, and the Database, are all subject to security threats, thus forming the attack surface.

More importantly, to take the analogy further, just like control theory assumes that the Controlled Object is subject to Disturbances, security theory, if one is developed for this model, or resilience engineering, could assume that known or unknown Threats play the role of Disturbances to the Controller. Threats can only be effective if they exploit some weaknesses of the Controller called vulnerabilities. In this model, vulnerabilities affecting the controller are endangering the system assets that can be exploited by one or more threats. The formal definition [30] reads as follows:

Vulnerability – a weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source.

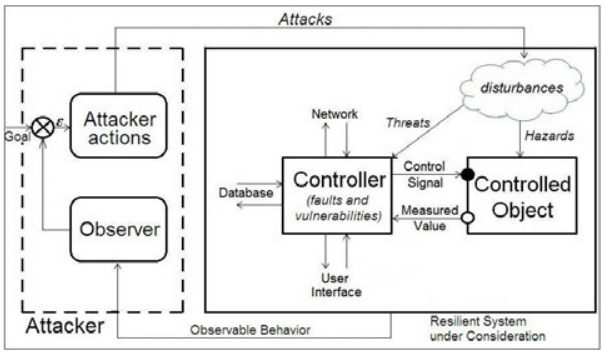

Fig. 4 Modern control system: disturbances and attacks

Pairing this understanding of security related concepts of Threats and Vulnerabilities with safety related concepts of Hazards and Faults, one arrives to the aggregated model suitable for resilience modeling, as shown in Figure 4. Assuming further that existing vulnerabilities in the Controller hardware or software, when exploited, may cause a security breach negatively affecting the behavior of the Controller, one can formulate the Design for Resilience Considering Security problem as follows:

Given (1) the description of the System whose resilience is of concern, (2) the characteristic of the expected external Disruptions, including Hazards and Threats, (3) the characteristics of internal Faults and Vulnerabilities, develop a Strategy (an analog of a Control Law running on the Controller) to meet a certain Criterion (performance metric) expressed in terms of the distance from the desired state of the System.

IV. Verification and Expansion of the Model

The Verification Problem can be formulated as follows:

Given (1) the description of the System whose resilience is of concern, (2) the characteristic of the expected external Disruptions, including Hazards and Threats, (3) the characteristics of internal Faults and Vulnerabilities, develop a Strategy (an analog of a Control Law running on the Controller) to meet a certain Criterion (performance metric) expressed in terms of the distance from the desired state of the System.

In this view, we review a number of recent papers on assessment of resiliency, with two objectives in mind:

- First, to see whether or not the structural components of resiliency discussed in other papers fit into our model, which would serve the purpose of model validation.
- Second, to see whether or not a Performance Metric can be developed that would be useful in assisting in the assessment of resiliency.

The MITRE resilience review report [14] does not build any specific model of resilience, but introduces an interesting taxonomy composed of eight resilience categories. The categories differ regarding the ways how the resilience is implemented and their relation to the system components and events as presented in Figure 4. The taxonomy of resilience categories include: Adaptive Response, Deception, Detection/Monitoring, Dynamic Variations, Resilience Integrity, Isolation/Containment, Metrics/Assessment, and Cross-Area.

Rieger et al. [11] state that "resilience describes how systems operate at an acceptable level of normalcy despite disturbances or threats" and define explicitly a Resilient Control System as the one "that maintains state awareness and an accepted level of operational normalcy in response to disturbances, including threats of an unexpected and malicious nature." Thus, the definition is strictly consistent with the view presented earlier in this section. Among the specific issues to be considered when addressing the notion of resilience, the authors listed: latency, physical degradation, cyber security, and human performance.

Strigini [21] presents an interesting perspective on resilience, deriving the word from the Latin verb *resilire* (*re-salire*: to jump back), which literally means "the tendency or ability to spring back, and thus the ability of a body to recover its normal size and shape after being pushed or pulled out of shape, and therefore figuratively any ability to recover to normality after a disturbance." He confirms the technical meaning of the term referring "to materials recovering elastically after being compressed, and also in a variety of disciplines to designate properties related to being able to withstand shocks and deviations from the intended state and go back to a pre-existing, or a desirable or acceptable, state." The paper also confirms the approach presented here to building a model of resiliency, referring to feedback control and stability, as well as to fault tolerance and redundancy.

To summarize, the papers by Rieger et al. [11] and Strigini [21] address concepts directly compatible with those proposed when building the model of a resilient system in Figure 4. It contains a control system as an example of a cyberphysical system and includes all related components: threats that can exploit vulnerabilities in the controller, hazards/threats that may activate controller's faults, and a

hypothetical attack surface that consists of four interfaces through which the controller interacts with the world.

Thus, the model of a resilient system is nearly complete and can be viewed as validated. As indicated in the analysis of the MITRE report [14], an extension of the initial model is proposed, which includes an abstraction of an Attacker, capturing the essence of his actions, which is also illustrated in Figure 4.

V. Resiliency Metrics and Measures

A number of authors discuss various aspects of assessing resiliency, presenting numerous metrics and measures, using these terms interchangeably. For example, Strigini [21] discusses the entire array of measures related to quantitative reasoning about resilience (they should be in fact called metrics), including the following:

- measures of dependability in the presence of disturbances, which may be estimated empirically in operation or in a laboratory, or through probabilistic models (as functions of measures at component level)
- measures of the amount of disturbances that a system can tolerate, typically obtained from analyzing a system's design
- measures of probability of correct service given that a disturbance occurred ("coverage factors"), typically estimated empirically, often in a laboratory.

Additional measures for less technical categories of systems listed in [21] include:

- buffering capacity, which is essentially an "extent of tolerable disturbances";
- flexibility versus stiffness: the system's ability to restructure itself in response to external changes of pressure;
- margin: how closely or how precarious the system is operating relative to one or another kind of performance boundary;
- tolerance: how a system behaves near a boundary – whether the system gracefully degrades as stress/pressure increase or collapses quickly when pressure exceeds adaptive capacity;

In [17], the authors state that the "framework for resilience engineering is based on four key pillars: disruptions, system attributes, methods, and metrics," but do not create a more formal model of resilience beyond listing a number of components for each "pillar." The most interesting from our perspective are the Metrics, which include the following: time/cost to restore operation, time/cost to restore configuration (reconfigure), time/cost to restore functionality/performance, degree to which pre-disruption state is restored, potential disruption circumvented, and successful adaptations with time and cost constraints.

Almeida et al. [31] use a model similar to ours, but much less detailed, to reason about resilience of self-adaptive systems, calling the assessment process "benchmarking." They use several service related metrics, including:

- Performance: the number of operations the system is able to perform per unit time.
- Uptime: measure of the time the system is available during the benchmark procedure.
- Robustness: requires assessing the relative number of perturbations the system deals with gracefully, while maintaining system attributes values close to the desired specifications.

To better characterize self-adaptation capability, they consider other metrics that include:

- Time to react: the time elapsed from the exposure of the system to a perturbation until its recognition and decision to act upon.
- Time to adapt: the time necessary to execute the decided adaptation.
- Time to stabilize: the time the system takes to stabilize its operation.

Finally, stating that "as a system's ability to successfully adapt to perturbations depends on correctly deciding which perturbations to act upon, and doing it in a timely fashion," they include two additional metrics:

- Sensitivity: represents the ratio of adaptations performed to the number of perturbations submitted to the system.
- Degree of autonomy: portrays the system dependency on human operators.

On the other hand, Ramuhalli et al. [26] have a critical view of this approach to resilience metrics and state the following: "The bulk of these metrics are focused on system-level quantities (such as time to recover from an attack, percentage of available services, etc.). While these are important and help characterize the system performance, these are difficult to use for dynamic reconstitution, as computing such metrics in real-time (as the system is being reconstituted) from knowledge of only the configuration and/or connectivity is difficult." What they propose to use instead are indirect metrics and including graph metrics, "such as diameter, algebraic connectivity, average path length, clustering coefficient, although other graph statistics may be relevant and computable in real-time."

In an extensive report, Bodeau et al. [32] distinguish between two broad types of metrics relevant to cyber resiliency:

- Technical metrics, which evaluate the behavior of technologies and of technology dependent mission/business processes (particularly cyber defense processes);
- Organizational metrics, which evaluate organizational processes for resilience (in which cyber resiliency is – or should be – a consideration).

Both categories are, however, related to a much higher level of resiliency than that concerned in cyberphysical systems.

VI. Modeling Strategy

The essence of resilience is not to guarantee absolute protection against attacks or failures, but to provide successful recovery after disruptions. For example, Ramuhalli and his group at Pacific Northwest National Laboratory understand resilience as the degree of stability of the system at or near any operational state [33]. Similar views have been expressed by researchers at the Idaho National Laboratory [11] and others. Consequently, Vugrin et al., at SANDIA [34] state that "the cybersecurity community has voiced the opinion that cybersecurity strategies must expand beyond the protection-centric focus to incorporate cyber resilience principles." This has been advocated even earlier, by a national panel of researchers [35], stating that in case of a disruption such as an imminent security breach, what "cyberphysical systems require is either reconfiguration to reacquire the needed resources automatically or graceful degradation if they are not available."

In previous research, the authors have addressed this problem with respect to security [36]. An essential assumption in this approach was that a security breach may not necessarily cause complete system failure but just degradation of system services. The effects of a security breach were analyzed with respect to changes of system behavior in the following states: normal state, several degraded states (depending on the system or application), and failure state. The results led to a better understanding of consequences of such breaches and improvement of security policies.

The same strategy is applied in case of modeling resiliency. First, based on the model of resiliency developed in Section III, involving safety and security as essential resiliency components, an extended model is proposed involving an Attacker. Then, the Performance Metric can be used, which adequately reflects the distance between the Normal and Degraded states. Finally, a simulation tool is applied to evaluate resiliency under various scenarios.

The modeling process involves the Model-Based Environment, Möbius [37] which includes a number of modeling formalisms assisting in system performance and dependability modeling.

One of these formalisms involves Fault Trees that are widely used for modeling system safety property. An illustrative example of a car engine and wheels control, as a case of a cyberphysical system, is shown in Figure 5 [37], as an AND tree for potential engine failure, and can be enhanced by an OR tree for wheel failure. Running the simulator for a specific set of parameters constitutes an experiment, which results in calculating means and variances confirming specific hypotheses that can be related to safety evolving over time as a component of resiliency.

A newer modeling formalism, the Adversary VIew Security Evaluation (ADVISE) was developed recently to enhance Möbius and provide means for quantitative, state-based analysis of system security [38]. Building the ADVISE model relies on constructing an attack execution graph describing steps that an attacker might attempt to achieve specific goals. In addition, various attributes of the attacker are defined in his profile.

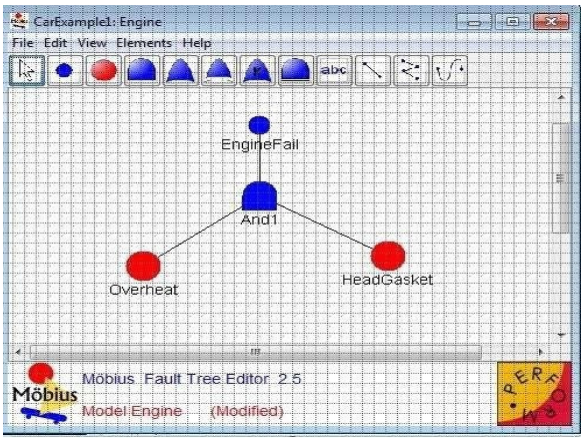

Fig. 5 Building Fault Trees in Möbius

Essential in the ADVISE model, the attack execution graphs (AEG) consist of attack step nodes, state variable nodes, and directed arcs between both types of nodes. State variable nodes store the state of a model during execution. During the run of an ADVISE model, the attacker (called an adversary) evaluates the state of the system, determines the most attractive attack step and attempts it. This decision process is repeated throughout the entire run of simulation.

ADVISE takes advantage of the Abstract Functional Interface (AFI) that facilitates the addition of new modeling formalism modules and new solver modules. Thanks to this feature, ADVISE models are designed to be composable with other Möbius models. It is anticipated that this capability can be used to combine security analysis in ADVISE with safety analysis using Möbius fault tree models for joint assessment of resiliency. Metrics for the assessment can be defined using the standard performance model available in Möbius, such as reward code expressions and impulse rewards. Specific metrics can be constructed to assess accomplishment of the goals by an attacker and risks associated with safety violations, to draw conclusions about resiliency levels.

VII. Conclusion

This paper addressed the assessment of resiliency of cyberphysical system in response to external disturbances, understood as hazards and threats causing safety and security violations, respectively, and related internal defects known as faults and vulnerabilities. A combined model for resiliency modeling and assessment was built, based on the view of feedback control theory enhanced with principles of

fault tolerance. This model was validated against the recent literature and enhanced with the view of potential attackers.

Resilience metrics were reviewed and analyzed by analogy with performance measures of the control system to assist in Design for Resilience. With the multitude of different approaches to resilience metrics and measures, it is suggested that those measures be selected, which best address the distance between the desired state of a system and the disrupted state level. The modeling strategy was proposed, based on using the Möbius modeling tool, which can address both security and safety issues as components of resiliency. Future work will involve combined simulations of fault-tree based (safety) and ADVISE models (security).

REFERENCES

[1] Zimmerman M.A., Resiliency Theory: A Strengths-Based Approach to Research and Practice for Adolescent Health, Health Education and Behavior, Vol. 40, No. 4, pp. 381– 383, August 2013.

[2] Holling C., Resilience and stability of ecological systems. Annual Review of Ecology and Systematics, Vol. 4, pp. 1-23, 1973.

[3] Christopher M., H. Peck, Building the Resilient Supply Chain. International Journal of Logistics Management, Vol. 15, No. 2, pp. 1-14, 2004.

[4] Adjetey-Bahun K. et al., A simulation-based approach to quantifying resilience indicators in a mass transportation system, Proc. ISCRAM2014, 11th Int'l Conference on Information Systems for Crisis Response and Management, University Park, Penn., May 18-21 2014.

[5] Goerger S.R., A.M. Madni, O.J. Eslinger, Engineered Resilient Systems: A DoD Perspective, Procedia Computer Science, Vol. 28, pp. 865-872, 2014.

[6] Castano V., I. Schagaev, Resilient Computer System Design, Springer-Verlag, Heidelberg, 2015.

[7] Suri N., G. Cabri (eds.), Adaptive, Dynamic, and Resilient Systems. CRC Press, Boca Raton, Fla., 2014.

[8] Hollnagel, E. Pariès, J. Woods, D. D. & Wreathall, J. (eds.). Resilience Engineering Perspectives. Vol. 3: Resilience Engineering in Practice. Ashgate, Farnham, UK, 2011.

[9] Ellison R. J. et al., Survivable network systems: An emerging discipline. Technical Report CMU/SEI-97-TR-013, Software Engineering Institute, Pittsburgh, Penn., 1997.

[10] Allen J., N. Davis, Measuring Operational Resilience Using the CERT® Resilience Management Model, Technical Note CMU/SEI-2010-TN-030. Software Engineering Institute, Pittsburgh, Penn., September 2010.

[11] Rieger C.G., K.L. Moore, T.L. Baldwin, Resilient Control Systems: A Multi-Agent Dynamic Systems Perspective. Proc. EIT 2013, IEEE International Conference on Electro/Information Technology, Rapid City, SD, May 9-11, 2013.

[12] Vugrin E.D., J. Turgeon, Advancing Cyber Resilience Analysis with Performance-based Metrics from Infrastructure Assessment. Int'l Journal of Secure Software Engineering, Vol. 4, No. 1, 2013.

[13] Alexander J.S., Achieving Mission Resilience for Space Systems. Spring 2012. URL: http://www.aerospace.org/2013/07/29/achieving-mission-resilience-for-space-systems/

[14] Pietravalle R., D. Lanz, Resiliency Research Snapshot. The MITRE Corporation, Bedford, Mass., June 2011.

[15] Bodeau D., R. Graubart, Cyber Resiliency Assessment: Enabling Architectural Improvement, Technical Report MTR120407, The MITRE Corporation. Bedford, Mass., May 2013.

[16] Caralli R.A. et al., CERT® Resilience Management Model, v1.0. Technical Report CMU/SEI-2010-TR-012. Software Engineering Institute, Pittsburgh, Penn., 2010.

[17] Madni A.M., S. Jackson, Towards a Conceptual Framework for Resilience Engineering, IEEE Systems Journal, Vol. 3, No. 2, pp. 181-191, June 2009.

[18] Teixeira A., Toward Cyber-Secure and Resilient Networked Control Systems. PhD Thesis, KTH Royal Institute of Technology, Stockholm, November 2014.

[19] Meyer, J. F. Defining and evaluating resilience: A performability perspective. Proc. PMCCC, Int'l Workshop on Performability Modeling of Computer and Communication Systems, Eger, Hungary, September 17-18, 2009.

[20] Bishop M. et al., Resilience Is More than Availability. Proc. NSPW'11, New Security Paradigms Workshop, Marin County, Calif., September 12-15, 2011, pp. 95–104.

[21] Strigini L., Fault Tolerance and Resilience: Meanings, Measures and Assessment, Resilience Assessment and Evaluation of Computing Systems, K. Wolter et al. (eds.), Springer-Verlag, Berlin, 2012.

[22] Athans M., P. Falb, Optimal Control. An Introduction to the Theory and Its Applications. McGraw-Hill, New York, 1966.

[23] Randell B. et al. (eds.), Predictably Dependable Computing Systems, Springer-Verlag, Berlin, 1995.

[24] Najjar W., J. Gaudiot, Network resilience: A measure of fault tolerance, IEEE Trans. Computers, Vol. 39, No. 2, pp. 174–181, February 1990.

[25] Axelrod W., Investing in Software Resiliency, CrossTalk: The Journal of Defense Software Engineering, Vol. 22, No. 6, pp. 20-25, September/October 2009.

[26] Ramuhalli P. et al., Towards a Theory of Autonomous Reconstitution of Compromised Cyber-Systems. Proc. HST2013, IEEE International Conference on Technologies for Homeland Security, Waltham, Mass. November 12-14, 2013.

[27] Ross R., J.C. Oren, M. McEvilley, Systems Security Engineering: An Integrated Approach to Building Trustworthy Resilient Systems. NIST Special Publication 800-160. National Institute of Standards and Technology, Gaithersburg, MD, May 2014.

[28] International Electrotechnical Vocabulary (IEV), International Electrotechnical Commission (IEC), Geneva, Switzerland. URL: http://www.electropedia.org/

[29] IEEE Software and Systems Engineering Vocabulary. IEEE Computer Society, Washington, DC, URL: http://computer.org/sevocab

[30] National Information Assurance (IA) Glossary. CNSS Instruction No. 4009. Committee on National Security Systems, 26 April 2010.

[31] Almeida R., H. Madeira, M. Vieira, Benchmarking the Resilience of Self-Adaptive Systems: A New Research Challenge. Proc. 29th IEEE Int'l Symposium on Reliable Distributed Systems, New Dehli, October 31 - November 3, 2010.

[32] Bodeau D., R. Graubart, L. LaPadula, P. Kertzner, A. Rosenthal, J. Brennan, Cyber Resiliency Metrics. Version 1.0, Rev. 1. Technical Report MP120053, The MITRE Corporation, Bedford, Mass. April 2012.

[33] Ramuhalli P., Theory of Resilience: A Framework for Resilient Design and Reconstitution of Cyber Systems, Project Flyer, Pacific Northwest National Laboratory, Richland, Wash., 2014. URL: http://cybersecurity.pnnl.gov/documents/projects/Theory_Flyer.pdf

[34] Vugrin E.D., R.C. Camphouse, Infrastructure resilience assessment through control design. International Journal of Critical Infrastructures, Vol. 7, No. 3, pp. 243-260, 2011.

[35] National Research Council, Committee for Advancing Software-Intensive Systems, Producibility Critical Code: Software Producibility for Defense, National Academies Press, Washington, DC, 2010.

[36] Kornecki A., J. Zalewski, W. Stevenson, Availability Assessment of Embedded Systems with Security Vulnerabilities, Proc. SEW-2011, 34th IEEE Software Engineering Workshop, Limerick, Ireland, June 20-21, 2011, pp. 42-47.

[37] Möbius: Model-Based Environment for Validation of System Reliability, Availability, Security and Performance. Performability Engineering Research Group, University of Illinois, Urbana-Champaign, Ill., 2014. URL: https://www.mobius.illinois.edu/

[38] Ford M.D. et al., Implementing the ADVISE Security Modeling Formalism in Möbius. Proc. DSN '13, 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Budapest, Hungary, June 24-27, 2013.G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.