SAND2013-3616C

The Structure and Use of a Mission Impact Description Language

Philip L. Campbell
Department 5629 Network Systems Survivability & Assurance
Sandia National Laboratories
Albuquerque, USA
plcampb@sandia.gov

***Abstract*— A Mission Impact Description Language (MIDL) enables automatic computation of mission impact in a simulated network. A program in an MIDL consists of three parts: (1) an "Impact Model" that maps changes in the simulated network to mission impact; (2) a set of "Stress Models," where "stress" can be natural and/or malicious; and (3) a "Boundary Model," where "boundary" describes the limit of the allowable stress in any given set of Stress Models. (The Boundary Model is unnecessary and is provided as a convenience to the user.) An MIDL could be an extension of a Network Description Language (NDL) and presumes the use of a network test bed. Just as compiled languages enable optimizations such as function in-lining and loop unrolling, so an MIDL for a network test bed enables investigations, such as comparison of multiple Stress Models or Monte Carlo studies for a given Boundary Model. This paper describes the basics of an MIDL.**

Keywords—command & control; networks; network simulation; mission impact.

I. INTRODUCTION

This paper presents an example of providing a way for the computer to do more of our work for us by describing a type of language—in this case a Mission Impact Description Language (MIDL)—that enables us to describe what we want done. The particular example presented in this paper involves determining whether a given network configuration can provide the communication necessary for a given mission. The same approach applies to any other problem about which we want a computer's help.

II. RELATED WORK

The topic of this paper is bridging the gap between mission and cyber assets to provide mission assurance. (If the gap is to be filled in real time, then the topic is "cyber situation awareness.")

D'Amico et al. bridge the gap via a "relational database that links IA events to the cyber assets (e.g., workstations, datafiles) on which those events occur, and includes relationships about the organization's mission-critical tasks that depend on those assets" ([2] page 1). In a separate paper D'Amico et al. [3] describe two workshops they held that focused on mapping those relationships for both commercial systems and military systems, using entity-relationship-attribute (ERA) diagrams to develop models (Shaw et al. [17] suggest instead the use of the DoD Architecture Framework [4] to map those relationships). The workshops approached the issue from the users' viewpoint (top-down) and from the IT providers' viewpoint (bottom-up). The results from the two workshops were similar. The authors concluded that in addition to information about mission and devices, information is also needed about task, user, capability, service, application, as well as "goals, measures of success, and priority of a specific mission or organization" ([3] page 11) and the equivalent on the IT providers' side.

Jojodia et al. bridge the gap via "topological vulnerability analysis" in the form of their Cauldron tool that combines vulnerabilities "in ways that real attackers might do" ([13], page 1340) to arrive at vulnerability paths. An attack graph simulation then uses those vulnerability paths—along with many other inputs—to enable users to see how attacks might proceed step by step through their network. If users provide a workflow consisting of tasks required for a given mission and also provide a value for each mission and its tasks, then Cauldron can help determine the impact that the compromise of a given asset has on a task or mission.

Goodall et al. bridge the gap via Camus (Cyber Assets to Missions and Users) that uses ERA diagrams along with "commonly available data sources" (such as the enterprise organization chart) and "user logs and network traffic, to infer cyber asset-to-mission relationships" in order to "automatically map cyber assets to the users who depend on them, to the missions they support, and to the services they provide" ([8], pages 1 & 5).

Grimaila et al. bridge the gap via their Cyber Incident Mission Impact Assessment (CIMIA) process, the goal of which is to

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

provide "decision makers with timely notification and relevant impact assessment, in terms of mission objectives, from the time an information incident is declared, until the incident is fully remediated" ([10], page 1). This process models the effects of an attack via simulation, using a business process model [BPM].

Peterson also describes the CIMIA process. Its purpose is to maintain "real-time situational awareness of mission critical resources" ([15], page iv) (see also [7]).

Musman et al. bridge the gap via a "[business] process model [BPM] of the system [in the form of a CIMIA] that can be run as an executable simulation to estimate mission outcomes" ([14]. page 1). The model can compute the Cartesian product of incident, duration, asset, and mission workflow locations.

And Stanley et al. bridge the gap via the IT Infrastructure Library (ITIL) [12] which bundles services and provides a common language for IT providers and end-users. They note that IT providers and end-users "speak different languages" ([18], page 1) so neither party knows which IT services are needed in order to fulfill which missions. The IT providers need to determine what IT components fulfill which services. Meanwhile, the end users need to determine what IT services are required for which missions. With that common language the two parties can work out what is needed and can trace IT service failure to mission failure and mission needs to IT services. IT governance, service level agreements, and databases are also required.

Grimaila & Fortson describe the gap as a gap between infrastructure and information. They note that cyber defense tends to focus on the "infrastructure rather than the information" ([9] page 13) and of course the commander focuses on the information rather than the infrastructure. It is not surprising, then, that D'Amico et al. could find "no readily available techniques for automatically determining mission impact" ([1] page 1) and that there is "no systematic method for defining the complex mapping between cyber assets (hardware, software, data), missions and users" ([3] page 1).

This paper bridges the gap by describing a type of language that enables a commander to describe the mission and attacks of interest in terms that allow a network simulator to determine the effect of those attacks on that mission. (In general, the authors cited above are concerned with actual, running systems, not simulations of running systems.)

III. A Computer Language

There are an unbounded number of missions and an unbounded number of network configurations. At first glance, then, it can seem that the task of computing how a given configuration supports or detracts from a given mission is hopeless. However, if we shift our view and focus instead on providing a way to describe each of those missions and configurations, then we see the way forward. In this paper the way we provide "a way for the computer to do more of our work for us," as mentioned in the Introduction above, is to develop a way that the work can be described so the computer itself can perform the work. That is, we develop a computer language. Down near the hardware level it is all a representation of zeros and ones but up at our level it means, as Humpty Dumpty would put it, what we "choose it to mean—neither more nor less."

Consider a military command & control network. What the commander wants (i.e., the mission of the network) is for the network to provide sufficient communication. The network is subject to all of the natural stress that physical computer and communication devices are subject to, as well as malicious stress due to attacks. It is the network configuration operating under that stress that interests us. So we simulate the network in a network test bed and then we simulate the natural and malicious stress. What we need is a way to describe the effect that this stress has on the mission of the network. This "way" is what we call in this paper a "Mission Impact Description Language" (MIDL). Hopefully this strikes the reader as both a natural and a simple way to proceed.

IV. Extending a Current Language

A commander (or at least one of his lieutenants) can use what we could call a Network Description Language (NDL)[1] to describe how a test bed should construct and simulate the operation of a representation[2] of a network. We could extend an NDL to describe how to compute the impact to the mission that stress has on that network, arriving at an MIDL. The fundamental part of an MIDL program (i.e., a program written in an MIDL) is a description of how changes in the network map to mission impact. This fundamental part is an "Impact Model" and is describable in functional terms as follows:

Impact Model: changes → mission impact

where "changes" are reflected in measurements.

The "mission" embedded in an Impact Model could be defined (or at least approved) by the commander and could be expressed as an enum or an integer range, for example (using pseudo-code),

[1] An example NDL is what Emulab uses, namely ns2 [6]. Faber et al. refer to a topology description language "topdl" ([5] page 3) which presumably only describes a topology and does not describe a simulation run using that topology and thus would be a proper subset of an NDL. A second example is the Georgia Tech Network Simulator (GTNetS) [16].

[2] Faber et al. call this an "apparatus" [5].

- enum { functional, email and chat only, email only, chat only, nonfunctional } mission;
- int mission 0..1000;

The expression of a mission is not confined to a single enum or a single integer range, of course. The expression could have any amount of complexity.

Statements in the Impact Model could describe the impact that particular measurements have on the mission, such as shown in the following pseudo-code:

- If throughput on node A goes below rate B for more than C amount of time, then the mission degrades to at least "email only" for D amount of time;

- If connectivity between node A and node B is not available between times t1 and t2, then mission = mission – 100;

The measurements could include

- throughput,
- connectivity,
- up/down status, and
- (many others).

These primitives can be the foundation for higher-level routines (not yet services or even libraries) such as

- the mean & standard deviation packet loss during a specified window of time and sampled at a particular rate on a specified set of links, or
- the number of times that information of type A ever touches any nodes of type B.

The particular measurement primitives and higher-level library routines for a given MIDL would depend upon the intersection of

- what can be measured in the particular network test bed,
- what the MIDL user community wants, and
- what the MIDL developers are funded (and willing) to develop.

V. Creating an Impact Model

The average commander probably considers the supporting network as a tool, as a "black box," and thus of only peripheral interest. The commander would probably not be able to describe the supporting network in sufficient detail to create an adequate NDL program. A graphical user interface would be of little help. In a similar vein, even though the commander has intense interest in the mission of the network, the commander may not be able to describe that mission in sufficient detail to create an Impact Model. However, the commander can certainly describe the mission operationally. So one approach to creating an Impact Model is to present to the commander (or at least to his lieutenants) a simulation of the network and then to stress the network in various ways, asking the commander to identify the resulting impact. The state of the network when the commander identifies impact should then be translatable, admittedly with effort, into the measurements described by an Impact Model. This is similar to the way that people construct investment utility functions, namely repeatedly balancing risk versus rate of return to identify one's "investment boundary" ([11] page 212).

VI. Executing an Impact Model

A test bed can construct an apparatus (i.e., a representation of a network with a given topology) but, like all simulations, it needs an initial list of events, with an indication of when the events occur in simulation time. An event could start a data flow, say, or degrade a link or bring up a node that has been down. Each event needs to be described in terms of its effect on the network and that effect must be measurable and may conceivably have an impact on the mission. If an event has no effect reflected in any measurement, then it cannot have an impact on the mission.

The simulation begins with the events scheduled for time 0. As the simulation progresses additional events may be added to the list. The simulation ends when the event list is empty or a when special HALT event is encountered.

With the above in hand, a test bed, F, can provide the following mapping:

> F: network topology x initial time-event list x Impact Model → mission impact

The "initial time-event list" initiates what we could call the normal or unstressed (i.e., baseline) activity of the network.

VII. Modeling Stress

The whole purpose of an MIDL program is to explore the effect of stress on the mission. The Impact Model provides the relationship between changes and mission impact and it simulates what is intended to be the baseline activity. In order to explore stress we need a "Stress Model," where stress is natural and/or malicious.[3]

---

[3] A proper superset of a Stress Model would be what we could call a "War Game Model" that could describe, in addition to the offensive activity of the

A Stress Model maps time to network changes. It is describable in functional terms as follows:

Stress Model: time → changes

where "time" is simulated time, of course.

VIII. Modeling the Boundary

A "Boundary Model" is provided as a convenience to the user. A Boundary Model describes the limits of the cumulative stress described by a set of Stress Models (where a set can be a singleton). For example, what happens if both Adversary A and Adversary B attack at the same time? A Boundary Model could specify that the cumulative stress cannot take down more than half of the nodes at once or during the entire simulation run and/or it could specify that the stress cannot degrade a particular link by more than 30%. A Boundary Model thus maps changes suggested by one or more Stress Models to an acceptable interval and is describable in functional terms as follows:

Boundary Model: suggested changes → acceptable interval

A Boundary Model is a convenience because Stress Models could impose their own limits.

Of course, an MIDL program could be run without a Boundary Model. That would be the equivalent of running the program with a Boundary Model that only has one statement that is the semantic equivalent of "anything goes."

IX. An MIDL Program

An MIDL program then consists of three parts:

1. an Impact Model,
2. a set of Stress Models, and
3. an optional Boundary Model.

For development purposes an MIDL program can be run without any Stress Models or a Boundary Model. However, what the commander wants to know, and hence the reason for building and running the program in the first place, is to include a set of Stress Models. An MIDL, extending an NDL, thus provides for programs to express the following mapping

MIDL program: network topology x initial time-event list x Impact Model x Stress Model(s) x Boundary Model → mission impact

X. Parameter Studies

Using an MIDL program the test bed can run parameter studies, testing, for example, each of a set of Impact Models against each of a set of Stress Models using a set of Boundary Models, as well as running Monte Carlo studies.

XI. Moving to a Higher Level

The above can be thought of as the beginning of an assembly language level description of an MIDL. To be useful, the above beginning must first be expanded to the point that it enables the expression of complete programs that compilers can recognize and translate down into assembly. That initial version would then need to be refined and moved "higher" as the language developers work with the users of the network test bed, finding out from those users the answers to questions such as

- Why are the users using the test bed?
- What is it that they want to know?
- How is what they want to know reflected in the network?
- What constructs in the MIDL would best serve those users?

These questions should enable an MIDL to move closer to the needs of the test bed users, bundling higher-level routines into libraries and then bundling those libraries into services, as Stanley et al. [18] suggest.

XII. Conclusion And Future Work

This paper describes the basics of a type of language referred to as a "Mission Impact Description Language" (MIDL). An MIDL could extend an NDL and assumes a network test bed. An MIDL enables the writing of a program that can automatically compute the impact of stress on the operation of a network to the mission that the network serves. An MIDL program consists of at least an Impact Model that maps changes in network measurements to mission impact. Such a program should also include a set of Stress Models and may include a Boundary Model.

A subsequent step will be to describe how an MIDL could work on actual, running systems, not simulations of running systems.

Stress Model, defensive activity, such as upgrading a communication line or routers or adding new communication lines and servers. A War Game Model would require that the test bed be able to change the network topology during a simulation run, which presumably is outside of the current scope.

[1] <2891> Anita D'Amico, Mark Larkin, "Methods of visualizing temporal patterns in and mission impact of computer security breaches," DARPA

Information Survivability Conference & Exposition II, 2001. DISCEX '03 Proceedings, pp. 343-351.

[2] <2892> Anita D'Amico, Stephen Salas, "Visualization as an aid for assessming the mission impact of information security breaches," DARPA Information Survivability Conference & Exposition, 2003, DISCEX '03 Proceedings, pp. 190-5

[3] <2583> Anita D'Amico, Laurin Buchanan, John Goodall, Paul Walczak, "Mission impact of cyber events: scenarios and ontology to express the relationship between cyber assets, missions, and uses," pages 388-397 in Proceedings of the 5th International Conference on Information Warfare and Security, edited by E. L. Armistead 2010.

[4] <DoDAF> DoD Architecture Framework (DoDAF) www.prim.osd.mil/Documents/**DoDAF**_2-0_web.pdf

[5] <2567> Ted Faber, Mike Ryan, John Wroclawski, "Building apparatus for multi-resolution networking experiments using containers," (unpublished).

[6] <2577> The Flux Research Group, "An evaluation of Emulab software and its evolution for the National Cyber Range," Document Version 0.6, September 3, 2009, University of Utah School of Computing.

[7] <2581> Larry W. Fortson, Jr., "Towards the development of a defensive cyber damage and mission impact methodology," Thesis, March 2007, AFIT/GIR/ENV/07-M9, Department of the Air Force, Air Force Institute of Technology (AFIT), Wright-Patterson Air Force Base, Ohio.

[8] <2586> John R. Goodall, Anita D. D'Amico, and Jason Kopylec, "Camus: automatically mapping cyber assets to missions and users," Military Communications Conference (MILCOM), 2009.

[9] <2585> Michael Grimaila, Larry Fortson, "Improving the cyber Incident damage and mission impact assessment," IAnewsletter, Vol. 11, No. 1, Spring 2008, pp. 10-5, http://iac.dtic.mil/iatac

[10] <2894> Michael R. Grimaila, Larry W. Fortson, Janet L. Sutton, "Design Considerations for a Cyber Incident Mission Impact Assessment (CIMIA) Process," 2009 International Conference on Security and Management (SAM09).

[11] <Hubbard> Douglas W. Hubbard, How to measure anything: finding the value of "intangibles" in business (2010), Second Edition, John Wiley & Sons, Hoboken, NJ.

[12] <ITIL> The ITIL Infrastructure Library. Office of Government Commerce (OCG).

[13] <2592> Sushil Jajodia, Steven Noel, Pramod Kalapa, Massimiliano Albanese, John Williams, "Cauldron: mission-centric cyber situational awareness with defense in depth," Military Communications Conference (MILCOM) 2011.

[14] <2893> Scott Musman, Mike Tanner, Aaron Temin, Evan Elsaesser, and Lewis Loren, "A systems engineering approach for crown jewels estimation and mission assurance decision making," 2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), pp. 210-6, April 11-5, 2011.

[15] <2584> Christy L. Peterson, "Measuring the utility of a Cyber Incident Mission Impact Assessment (CIMIA) process for mission assurance," Thesis, March 2011, AFIT/GIR/ENV/11-M9, Department of the Air Force, Air Force Institute of Technology (AFIT), Wright-Patterson Air Force Base, Ohio.

[16] <2590> George F. Riley, "The Georgia Tech Network Simulator," Proceedings of the ACM SIGCOMM 2003 Workshops, August 2003, pp. 5-13.

[17] <2582> A. Shaw, R. Mills, B. Mullins, K. Hopkinson, "A multilayer graph approach to correlating network events with operational mission impact," 75th Military Operations Research Society (MORS), June 12-4, 2007, US Naval Academy, Annapolis, MD.

[18] <2591> Jeffrey E. Stanley, Robert F. Mills, Richard A. Raines, Rusty O. Baldwin, "Correlating network services with operational mission impact," Military Communications Conference (MILCOM) 2005.