

HESTIA - SEMI-AUTOMATIC AND ADVERSARY-AWARE RISK ASSESSMENT OF CRITICAL
INFRASTRUCTURE SYSTEMS

A Dissertation

Presented in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

with a

Major in Computer Science

in the

College of Graduate Studies

University of Idaho

by

Ananth A. Jillepalli

Major Professor: Daniel Conte de Leon, Ph.D.

Committee Members: Frederick T. Sheldon, Ph.D.; Jim Alves-Foss, Ph.D.; Michael A. Haney, Ph.D.;

Clinton L. Jeffery, Ph.D.

Department Administrator: Terence Soule, Ph.D.

August 2020

AUTHORIZATION TO SUBMIT DISSERTATION

This dissertation of Ananth A. Jillepalli, submitted for the degree of Doctor of Philosophy with a Major in Computer Science and titled “HESTIA - Semi-Automatic and Adversary-Aware Risk Assessment of Critical Infrastructure Systems,” has been reviewed in final form. Permission, as indicated by the signatures and dates below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor: _____
Daniel Conte de Leon, Ph.D. _____
Date

Committee Members: _____
Frederick T. Sheldon, Ph.D. _____
Date

Jim Alves-Foss, Ph.D. _____
Date

Michael A. Haney, Ph.D. _____
Date

Clinton L. Jeffery, Ph.D. _____
Date

Department Chair: _____
Terence Soule, Ph.D. _____
Date

ABSTRACT

Due to the characteristics and connectivity of today's critical infrastructure systems, cyber-attacks on these systems are currently difficult to prevent in an efficient and sustainable manner. Prevention and mitigation strategies need accurate identification and evaluation of: system vulnerabilities, potential threats and attacks, and applicable hardening measures. Furthermore, the ability to prioritize hardening measures based on accurate assessments of risk is needed. In addition, the consideration of the availability, applicability, and cost of potential mitigation strategies is also needed.

To address this challenge we created HESTIA: High-level and Extensible System for Training and Infrastructure risk Assessment. In this dissertation, we describe the architecture and working principles of HESTIA. We present a formal model of the HESTIA system. We validate the HESTIA system using formal proofs and a case study-based proof tracing. We hope that the HESTIA system model will enable CPS engineers to build software that can iteratively: 1) specify a CPS (system), 2) select applicable attacks and hardening measures from a library (delta), 3) check system and delta specifications for consistency and applicability, and 4) apply deltas on system specifications, forming a new CPS model. We hope that HESTIA will enable the discovery of attack-defend scenarios through simulation and the design of optimal hardening strategies for a given CPS.

The contributions described in this dissertation are: (a) a detailed literature review on adversary-aware critical infrastructure risk assessment; (b) the architecture and workflow of HESTIA: a high-level and extensible system for adversary-aware infrastructure risk assessment; (c) a formalization of the HESTIA fundamental operations; (d) METICS: a realistic case study of a CPS organization; and (e) correctness and termination proofs of HESTIA's fundamental operations using formal verification and proof tracing via the METICS case study.

ACKNOWLEDGEMENTS

I would like to extend my gratitude towards the State of Idaho, for funding this research work under IGEN Cybersecurity grant of 2016. I would like to thank Dr. Daniel Conte de Leon and Dr. Frederick Sheldon for being two sturdy pillars of support for the entirety of my graduate program. I would also like to thank my committee members, Dr. Jim Alves-Foss, Dr. Michael Haney, and Dr. Clint Jeffery for their support and feedback. My thanks to Victor House for providing technological support in setting up the research infrastructure for this, and other related projects. I would also like to thank Susan Branting, Arvilla Daffin, and Arleen Furedy, for providing administrative support. The opinions expressed in this document are neither those of the State of Idaho nor those of the people mentioned herein.

TABLE OF CONTENTS

AUTHORIZATION TO SUBMIT DISSERTATION	ii
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF ACRONYMS	x
LIST OF DEFINITIONS	xi
CHAPTER 1: INTRODUCTION	1
CHAPTER INTRODUCTION	1
PRELIMINARY DEFINITIONS	1
THE CONTRIBUTIONS OF THIS DISSERTATION	1
AUTHOR’S RELATED PUBLICATIONS	2
OVERVIEW OF THIS DISSERTATION	3
CHAPTER 2: BACKGROUND, RELATED WORK, AND INTRODUCTION TO HESTIA	4
CHAPTER INTRODUCTION	4
CHAPTER PROBLEM	4
CHAPTER CONTRIBUTION	5
CHAPTER OUTLINE	5
RELATED WORK ON CYBERSECURITY IN CPS	6
DISCUSSION OF HESTIA’S WORKING PRINCIPLES	10
CONSISTENCY AND APPLICABILITY	12
APPLYING THE HESTIA PROCESS	13
CHAPTER CONCLUSION	14
CHAPTER 3: ARCHITECTURE OF HESTIA	16
CHAPTER INTRODUCTION	16
CHAPTER PROBLEM	16
CHAPTER CONTRIBUTION	16
CONTEXT IN PREVIOUS WORK	17
CHAPTER OUTLINE	18
ARCHITECTURE OF HESTIA	18
SNAPSHOT SPECIFICATION DEVELOPMENT SUBSYSTEM (SUBSYSTEM A)	18
DELTA SPECIFICATION DEVELOPMENT SUBSYSTEM (SUBSYSTEM B)	20

SPECIFICATION APPLICATION SUBSYSTEM (SUBSYSTEM C)	21
DEVICE SPECIFICATION TEMPLATES	23
APPLICABILITY PROBLEMS AND SOLUTIONS	23
RELATED WORK	24
CHAPTER CONCLUSION	24
CHAPTER 4: A FORMAL MODEL FOR THE HESTIA PROCESS	25
CHAPTER INTRODUCTION	25
CHAPTER PROBLEM	25
CHAPTER CONTRIBUTION	25
CHAPTER OUTLINE	26
A SEMANTIC FORMAL MODEL	26
UNIVERSES, CARDINALS, AND INDEXES	26
SPECIFICATIONS AND CONSTITUENT COMPONENTS	27
SPECIFICATIONS	27
ENTITIES	27
FIELDS	28
TEMPLATES	29
ALPHANUMERIC STRING	29
DOTTED STRINGS	29
FIELD REGISTER TYPES: DICTIONARIES	30
FIELD REGISTER TYPES: NAMED SETS	30
ALPHANUMERIC STRING EQUALITY OPERATION	32
FIELD REGISTER TYPES: VALUES	33
FIELD REGISTER SIZE DETERMINATION OPERATION	33
FIELD REGISTER SIZE COMPARISON OPERATION	34
DOTTED STRING EQUALITY OPERATION	35
DOTTED STRING INTERSECTION IDENTIFICATION OPERATION	35
ENTITY ID COMPARISON OPERATION	36
ENTITY TYPE COMPARISON OPERATION	37
FIELD NAME COMPARISON OPERATION	38
FIELD REGISTER TYPE DETERMINATION OPERATION	38
FIELD REGISTER TYPE COMPARISON OPERATION	39

FIELD REGISTER DICTIONARY EQUALITY OPERATION	40
FIELD REGISTER DICTIONARY KEY INTERSECTION IDENTIFICATION OPERATION	41
FIELD REGISTER NAMED SET EQUALITY OPERATION	42
FIELD REGISTER NAMED SET INTERSECTION IDENTIFICATION OPERATION	42
FIELD REGISTER COMPARISON OPERATION	43
FIELD REGISTER RELAXED COMPARISON OPERATION	46
FIELD REGISTER APPLICATION OPERATION	48
THE SPECIFICATION- AND ADVERSARY-BASED RISK ASSESSMENT PROCESS: HESTIA	50
CHECKING CONSISTENCY	51
CHECKING CONFLICTS	53
CHECKING APPLICABILITY	54
DELTA APPLICATION	55
CHAPTER CONCLUSION	57
CHAPTER 5: A HOLISTIC CYBER PHYSICAL SYSTEM CASE STUDY	59
CHAPTER INTRODUCTION	59
CHAPTER PROBLEM	59
CHAPTER CONTRIBUTION	59
CHAPTER OUTLINE	60
THE FICTIONAL-YET-REALISTIC POWER UTILITY: ACME CORPORATION	60
ACME CORP.'S SUBSTATIONS	61
ACME CORP.'S SUBSTATION CONTROLS	62
ACME CORP.'S OPERATIONS CENTER	63
ACME CORP.'S ENTERPRISE CENTER	64
UTILIZATION OF ACME CORP. MODEL FOR CYBER SECURITY TESTING	65
RELATED WORK	66
CHAPTER CONCLUSION	66
CHAPTER 6: FORMAL VERIFICATION AND PROOF TRACING FOR THE HESTIA PROCESS	71
CHAPTER CONTRIBUTION	71
CHAPTER OUTLINE	71
FORMALLY VERIFYING LEMMAS AND THEOREMS FOR HESTIA	71
LEMMA OF ALPHANUMERIC STRING EQUALITY	72
LEMMA OF SIZE DETERMINATION	72

LEMMA OF SIZE COMPARISON	73
LEMMA OF DOTTED STRING EQUALITY	74
LEMMA OF DOTTED STRING INTERSECTION IDENTIFICATION	75
LEMMA OF ENTITY ID COMPARISON	75
LEMMA OF ENTITY TYPE COMPARISON	76
LEMMA OF FIELD NAME COMPARISON	77
LEMMA OF FIELD REGISTER TYPE DETERMINATION	78
LEMMA OF FIELD REGISTER TYPE COMPARISON	78
LEMMA OF DICTIONARY EQUALITY	79
LEMMA OF DICTIONARY KEY INTERSECTION IDENTIFICATION	80
LEMMA OF NAMED SET EQUALITY	81
LEMMA OF NAMED SET INTERSECTION IDENTIFICATION	81
LEMMA OF FIELD REGISTER COMPARISON	82
LEMMA OF FIELD REGISTER RELAXED COMPARISON	83
LEMMA OF FIELD REGISTER APPLICATION	84
LEMMA OF ENTITY CONSISTENCY CHECKING	85
LEMMA OF ENTITY CONFLICT CHECKING	85
LEMMA OF ENTITY APPLICABILITY CHECKING	87
LEMMA OF ENTITY DELTA APPLICATION	88
THEOREM OF CONSISTENCY CHECKING	89
THEOREM OF CONFLICT CHECKING	90
THEOREM OF APPLICABILITY CHECKING	90
THEOREM OF DELTA APPLICATION	91
EXAMPLE FOR PROOF TRACING	92
EXAMPLE: ACME CORP AND ITS COMPONENTS	93
EXAMPLE: ATTACK	93
EXAMPLE: DEFENSE	94
EXAMPLE: DELTA APPLICATION	94
PROOF TRACING FOR HESTIA'S THEOREM OF CONSISTENCY CHECKING	104
PROOF TRACING FOR HESTIA'S THEOREM OF CONFLICT CHECKING	114
PROOF TRACING FOR HESTIA'S THEOREM OF APPLICABILITY CHECKING	132
CHAPTER CONCLUSION	152

CHAPTER 7: FUTURE WORK AND CONCLUSION	154
SPECIFIC FUTURE WORK ITEMS	154
BROADER SCOPE FOR FUTURE WORK: HESTIA SYSTEM	154
BROADER SCOPE FOR FUTURE WORK: ACME CORP. CASE STUDY	154
CONCLUSION	156
REFERENCES	157
APPENDIX A: COPYRIGHT AND CREDIT NOTICES	162

LIST OF ACRONYMS

ACE - Applicability Check Engine
AMI - Advanced Metering Infrastructure
CCE - Consistency Check Engine
COP - Communications Processor
COTS - Commercially Off-The Shelf
CPS - Cyber Physical Systems
CSO - Chief Security Officer
DSL - Domain Specific Language
HERMES - High-level, Easy to use, and Reconfigurable Machine Environment Specification language
HESTIA - High-level and Extensible System for Training and Infrastructure risk Assessment
HVDC - High Voltage Direct Current
ICS - Industrial Control Systems
IED - Intelligent Electronic Devices
MITRE - An American not-for-profit organization
OPS - Operator Console
PDB - Phasor DataBase
PDC - Phasor Data Consolidator
PMU - Phasor Measurement Unit
RTU - Remote Terminal Unit
SCADA - Supervisory Control and Data Acquisition
SME - Subject Matter Expert
SO - Security Operator
SRE - Specification Rewrite Engine
UML - Unified Modeling Language
VPN - Virtual Private Network

LIST OF DEFINITIONS

`AlphanumericString` - A string containing any combination of English alphabets and/or numbers

ASE - `AlphanumericString` Equality operation

Dotted String - A chain of `AlphanumericStrings` that are separated by dots.

DE - Dictionary Equality operation

DKII - Dictionary Key Intersection Identification operation

DSE - Dotted String Equality operation

DSII - Dotted String Intersection Identification operation

EIC - Entity ID Comparison operation

ETC - Entity Type Comparison operation

FRA - Field Register Application operation

FNC - Field Name Comparison operation

FRC - Field Register Comparison operation

FRRC - Field Register Relaxed Comparison operation

FRTD - Field Register Type Determination operation

FRTC - Field Register Type Comparison operation

EID - Entity Identifier

ET - Entity Type

FN - Field Name

Named Set - An unordered list containing `AlphanumericString` values

NSE - Named Set Equality operation

NSII - Named Set Intersection Identification operation

SC - Size Comparison operation

SD - Size Determination operation

\mathcal{S} - Set of all valid specifications

\mathcal{T} - Set of all valid templates

\mathcal{E} - Set of all valid entities

\mathcal{F} - Set of all valid fields

\mathcal{V} - Set of all valid `AlphanumericString` values

\mathcal{Q} - Set of all valid dotted strings

\mathcal{D} - Set of all valid dictionaries

\mathcal{L} - Set of all valid named sets

α - Consistency Checking operator

\otimes - Conflict Checking operator

?

\vdash - Applicability Checking operator

\vdash - Delta Application operator

CHAPTER 1: INTRODUCTION

1.1 CHAPTER INTRODUCTION

In this chapter, we introduce readers to the terms *specifications*, *high-level*, *extensibility*, and *hardening*. We present the specific contributions laid out in this dissertation. We make reference to a list of coauthored publications that have been produced from this dissertation. Finally, we present an outline of this dissertation document.

1.2 PRELIMINARY DEFINITIONS

Every system or tool has some form a setting, which is variable and which can be changed by users and/or administrators of the system or tool. Such variable settings are called ‘configurations’. ‘Policies’ are when users and/or administrators of a system write rules to define configurations. ‘Specification’ is a set of multiple rules that dictate the state of a system. A specification can represent an existing system. And also, a specification can specify a change to the existing system.

The extent of details laid out in a specification is called ‘granularity’. A ‘high-level’ specification is a set of policies which describe the state or a change to a system using details at the highest level of abstraction. The degree of granularity changes from system to system and as such, it is hard to impose a numerical measure on what constitutes ‘high-level’ or ‘low-level’ granularity.

‘Extensibility’ of a system is the ability of the system to be modular. The system should be able to accept addition of new features or removal of existing features. Extensibility improves the dynamic nature of any system. Thus, dynamic systems should aim to be extensible and modular.

‘Hardening’ of a system is changing the devices, states, and configurations of the system such that the system is less vulnerable to attacks or threats. Hardening can be of several types. For the purposes of this dissertation, we should consider hardening to be any approach to increase the security of a cyber physical system.

1.3 THE CONTRIBUTIONS OF THIS DISSERTATION

The mission of this dissertation is to provide the critical infrastructure cybersecurity community with a system that can enable dynamic risk assessment of their infrastructure. The system is a model that’s based in software engineering and formal theory. In the process of accomplishing our mission, we produced the following contributions that makeup the core of this dissertation:

1. We make an argument that high-granularity, adversary-aware, and semi-automatic specifications-based risk assessment models have the potential to protect against increasing threats and vulnerabilities of critical infrastructure. Chapter 2 primarily deals with the discussion of this contribution. Other chapters reinforce Chapter 2 as well.
2. We introduce HESTIA: a high-level and extensible system for training and infrastructure risk assessment. A brief introduction to HESTIA is presented in Chapter 2. A detailed discussion of HESTIA's architecture is presented in Chapter 3. Other chapters reinforce reader's understanding of the HESTIA workflow.
3. We developed a complete and formal mathematical model for HESTIA's fundamental properties. We present its formal (mathematical) schema in Chapter 4.
4. A holistic and realistic case study was needed to validate HESTIA system. We created one such fictional, yet holistic and realistic case study, named METICS, which is presented in Chapter 5.
5. Validation of HESTIA's schema is presented in Chapter 6. We validate HESTIA's schema formally and as well as using an example that's based on the case study presented in Chapter 5.

1.4 AUTHOR'S RELATED PUBLICATIONS

The need for dissemination of research via utilization of peer-reviewed publications is widely accepted. As such, in the process of my graduate studies and obtaining a 'Doctor of Philosophy' degree; several research manuscripts were produced in co-operation with colleagues. These manuscripts enhance software engineering theory in the areas of cybersecurity and risk assessment of critical infrastructure.

Three of the contributions presented in this dissertation have already been disseminated via peer-reviewed publications. The remaining two contributions are being prepared for publication in peer-reviewed journals. In this section, we list bibliographic entries. Of these four publications, publication number 1 corresponds to Contribution 1. Publications 2 and 3 correspond to Contributions 2 and 4, respectively. Contributions 3 and 5 have not been published to date in academic journals.

1. Ananth A. Jillepalli; Daniel Conte De Leon; Yacine Chakhchoukh; Mohammad Ashrafuzzaman; Brian K. Johnson; Frederick T. Sheldon; Jim Alves-Foss; Predrag T. Tosic; Michael A. Haney, "An architecture for HESTIA: high-level and extensible system for training and infrastructure risk assessment", In *International Journal of Internet of Things and Cyber-Assurance (IJITCA)*, vol. 1, no. 2, pp.173 - 193, 2018. DOI: 10.1504/IJITCA.2018.092478 [1]. Chapter 2.

2. Ananth A. Jillepalli, Daniel Conte de Leon, Mohammad Ashrafuzzaman, Yacine Chakhchoukh, Brian K. Johnson, Frederick T. Sheldon, Jim Alves-Foss, Predrag T. Tomic, Michael A. Haney, “HESTIA: Adversarial Modeling and Risk Assessment for CPCS”, In *the 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, Limassol, pp. 226-231, 2018. DOI: 10.1109/IWCMC.2018.8450297 [2]. Chapter 3.
3. Ananth A. Jillepalli, Daniel Conte de Leon, Brian K. Johnson, Yacine Chakhchoukh, Ibukun A. Oyewumi, Mohammad Ashrafuzzaman, Frederick T. Sheldon, Jim Alves-Foss, and Michael A. Haney, “METICS: A Holistic Cyber Physical System Model for IEEE 14-bus Power System Security”, In *13th International Conference on Malicious and Unwanted Software (MALWARE)*, Nantucket, pp. 95-102, 2018. DOI: 10.1109/MALWARE.2018.8659367 [3]. Chapter 5.

1.5 OVERVIEW OF THIS DISSERTATION

The remainder of this dissertation is organized as follows: Chapter 2 presents a detailed background and related work on the topic of specifications-based risk assessment of critical infrastructure. Chapter 2 also presents the main problem being addressed by this dissertation and introduces HESTIA, a potential solution to the problem. Chapters 3 and 4, respectively, present the architecture and formal model of HESTIA. A case study to validate HESTIA’s formal model is presented in Chapter 5. Chapter 6 presents validation of HESTIA’s model using formal verification an example based on the case study.

Chapter 7 presents a list of future work items to be accomplished in the direction of this dissertation’s research and provides a conclusion to this dissertation. We then present a bibliography, which is comprised of a complete list of references organized in order of their citation’s occurrence. Copyright and credit notices are presented as an appendix at the end of this dissertation.

CHAPTER 2: BACKGROUND, RELATED WORK, AND INTRODUCTION TO HESTIA

2.1 CHAPTER INTRODUCTION

Industrial Control Systems (ICS) are being transitioned into Cyber Physical Systems (CPS) where digital and analog equipment is being replaced by cyber-enabled equipment [4]. Such a transition has increased the vector of vulnerabilities of a control system due to: 1) increased connectivity of CPS to the internet, 2) deployment and use of applications without adequate and complete consideration of security risks, and 3) lack of continuous application of security controls and monitoring [5]. Until recently, most CPS infrastructure related attacks have resulted from insider threats. In the recent years, however, attacks originating from outside the company are becoming more frequent [5]. Nonetheless, a good percentage of both types of attacks, insider and outsider, are attributed to the propagative adoption of cyber-enabled equipment in CPS infrastructure [6]. These changes are increasing the ease of management of CPS. However, they are also worsening the already volatile security scenario of a CPS [6]. Attacks against CPS are occurring at an ever-increasing rate, resulting in financial loss to both governments and industries [7]. Estimates of losses due to cyber-attacks on CPS infrastructure may be as high as \$1.87 billion by 2018 [7].

2.2 CHAPTER PROBLEM

To the Chief Security Officer (CSO) or Security Operator(s) (SO) of a CPS facility, identifying vulnerabilities specific to a particular CPS infrastructure can be a challenge, if there is no high-level security policy specification [8]. In addition, determining a high-level security policy specification of an existing CPS is not sufficient for securing the CPS infrastructure. A CSO or SO should be able to design the best hardening strategy for a particular CPS system. The analysis process for designing a hardening strategy should include investigating questions such as: “which is the best defense for a given vulnerability?”, “where to best apply defense resources?”, “which devices/applications to harden?”, and “in which particular order?”. Several factors come into play in such an investigation: completeness and consistency of the CPS infrastructure specification, applicability of attacks and respective defenses against a particular CPS, and cost of possible attacks versus cost of possible defenses. In this chapter, we use the term *cost* to denote money, personnel resources, and time.

2.3 CHAPTER CONTRIBUTION

We argue that a complete, rigorous, consistent, and iterative process that enables the analysis described above, and a supporting tool-set, are much needed. Such an iterative process should be able to subject a CPS infrastructure’s specification to attack or defense specifications. Such a process and tool-set will: 1) facilitate risk assessment of the CPS infrastructure; 2) concretely identify the basis of risk abatement so that the attack surface diminishes over time, and 3) enable cybersecurity training of personnel.

In this chapter we introduce the architecture of HESTIA: High-level and Extensible System for Training and Infrastructure Risk Assessment. When fully developed, HESTIA will be able to: 1) take an existing CPS infrastructure specification as input, check it for consistency, and produce a consistent specification; and 2) take a consistent specification for a CPS and identify the type of attacks or defenses, which can be applied on the CPS infrastructure. The HESTIA process and tool-set enables an SO to perform this assessment iteratively. The results can then be used to prepare the best hardening strategy for that particular CPS infrastructure.

In this chapter, a CPS specification is a high-level and policy-based representation of a system describing: 1) an organization’s hierarchy and connectivity graph, 2) an organization’s devices within the hierarchy, and 3) the corresponding device configurations. In this chapter, specifications are written in a high-level language called HERMES [9]. We classify specifications into two types: *Snapshot Specification* and *Delta Specification*.

Project HESTIA is one branch of an umbrella research project that aims to improve the state-of-the-art in cybersecurity and risk management for CPS. Other areas being investigated are: designing and creating a CPS testbed for cybersecurity analysis, analyzing and designing secure energy storage systems, designing and testing security controls for High Voltage Direct Current (HVDC) systems, detection of stealth attacks on the power grid using machine learning and data analytics, and analyzing, implementing, and enforcing least-privilege CPS designs and configurations.

2.4 CHAPTER OUTLINE

This chapter is organized as follows: Section 2.5 sets the work in context relative to other CPS infrastructure risk assessment research works and also describes the differences between these related works and HESTIA. Sections 2.6, 2.7, and 2.8 describe the working principles involved in HESTIA’s development, which are: system and attack-defense specifications and consistency and applicability analysis. Conclusion to the chapter and a list of abbreviations follow.

2.5 RELATED WORK ON CYBERSECURITY IN CPS

The goal of project HESTIA is to create a risk analysis and assessment process and tool-set. This process and accompanying tool-set will enable the CSO or SO of an organization in identifying the best hardening strategy for their CPS. This involves identifying where and how to best use hardening resources. After extensive literature searches of the current state-of-the-art, we were unable to find reports of research work with the same objectives. One factor motivating us in pursuing HESTIA’s research is the lack of previous work in the area of specification-based CPS infrastructure risk analysis and assessment. Nonetheless, we review some of the existing works in the field of cybersecurity in CPS.

A ModelicaML-based and scalable modeling mechanism for CPS risk assessment was proposed by [10]. The same team also proposed an aspect-oriented specifications for modeling real time requirements in CPS [11]. The MITRE Corporation developed a comprehensive cybersecurity risk assessment process called *Making Security Measurable* [12]. *Making Security Measurable* helps a CPS SO with assessing threats and vulnerabilities. These, and most other approaches known to the authors, enable a comprehensive but manually-driven modeling and risk assessment process. Whereas, HESTIA enables a semi-automated and iterative risk assessment process.

A real-time, spatial-temporal correlations-based model to detect data injection attacks in smart grids was proposed by [13]. In this model, a smart grid is converted into a collection of states, called ‘cyber states’, and spatial-temporal correlations are discerned between these states using system condition inferring [13]. Further in this model, the real-time schematics are achieved by implementing trust-based voting algorithms [13]. This model falls short at identifying slow-changing data injection attacks.

A data-centric paradigm to detect data injection attacks in large scale smart grids was proposed by [14]. In this mode, the *Margin Setting Algorithm* (MSA) is used to process massive amount of domain-specific data of a large scale smart grid. An *Artificial Neural Network* (ANN) is used to perform comparative analysis of experimental results [14]. This model falls short at identifying data injection attacks in real-time and in data emerging from a sophisticated network of *Phasor Measurement Units* (PMUs) [14].

Two machine-learning based models for data injection attack detection in smart grid systems were proposed by [15]. The first model utilizes the multivariate Gaussian semi-supervised learning algorithm and the second model utilizes a measurement-based deviation analysis algorithm, which requires no learning [15]. Both models use *Principal Component Analysis* (PCA) to control the dimensionality of complex simulations [15]. This model falls short in identifying anomalies in transmission networks.

A deep-learning based model for data injection attack detection in smart meters, to prevent electricity

theft was proposed by [16]. The model utilizes a State vector Estimator (SVE) and a Deep Learning Based Identification (DLBI) algorithm [16]. This model compares the historical measurement data and recognizes a pattern to identify false data injection attacks in real-time [16]. For this model to work, data from a large number of sensing units is required.

A state estimation based model for data injection attack detection in power systems was proposed by [17]. This model takes advantage of state estimates, which analyze a system's Jacobian matrix, independent of the observation vector, given that some confidence intervals of the Jacobian matrix are made available [17]. The model also encourages masking the grid topology with falsified information [17]. The model utilizes PMUs and alternative current (AC) formulation, and as such, it is not useful for power systems without PMUs and AC formulation capability.

[18] also showed that cyber-attacks could target the measurements (power and voltage magnitudes) and grid topology that are communicated to the control center. Robust estimators were used to detect those attacks [18]. A PMU-based state estimator that exploits the time and space correlation in the PMU measurements to provide rejection of bad measurements and to improve the state estimation cybersecurity is proposed in [19].

A virtual testbed, modeled after a real-world power systems laboratory setup was developed by [20]. The purpose of this testbed was to run network-based attacks against the testbed to help in identifying potential weak spots in a real world CPS infrastructure [20]. [21] have also developed a stakeholder-aware and economics-based risk assessment method for security management of CPS infrastructure. Pending further investigation and negotiations, HESTIA may or may not utilize [21]'s risk assessment method for financial cost analysis aspects of HESTIA.

There exist several other models for detection of data injection attacks in CPS. For those interested in further reading, [22] have published a comprehensive survey of data injection attacks and countermeasures. In addition, [23] have also published a short survey of seven data injection attack detection models.

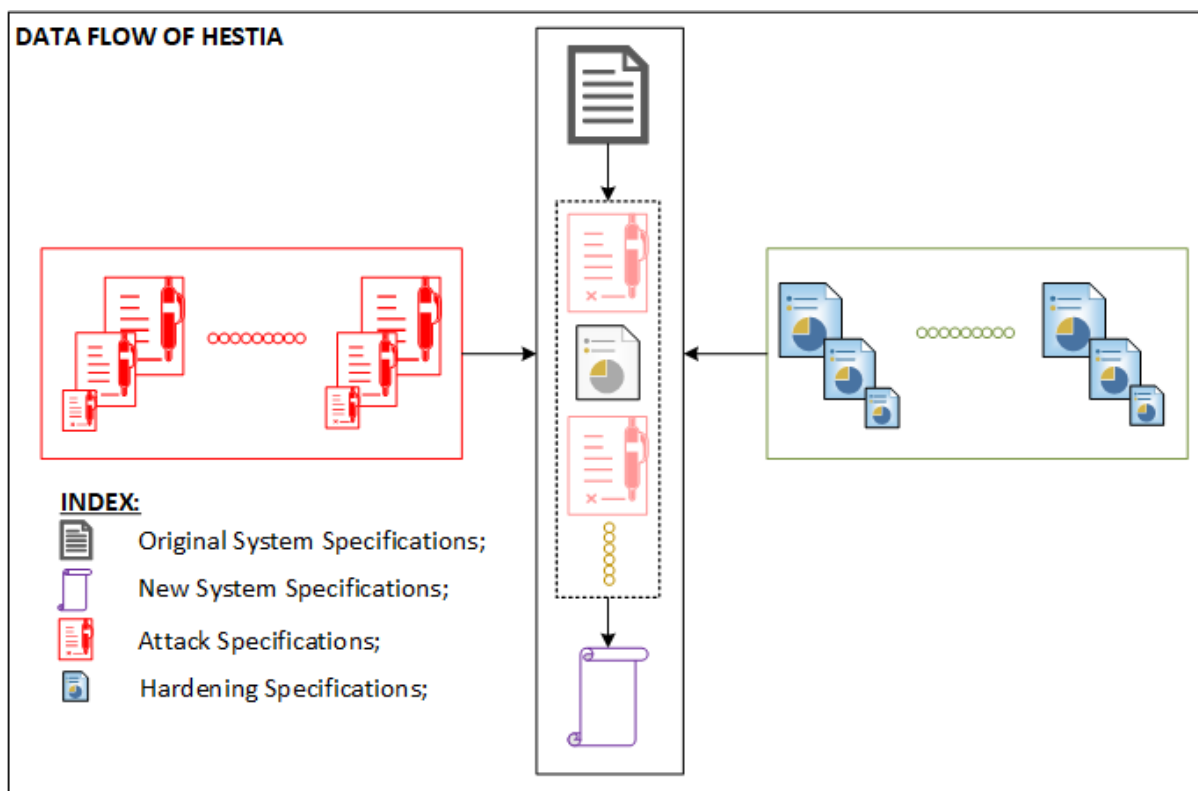


Figure 2.1: Data flow in HESTIA.

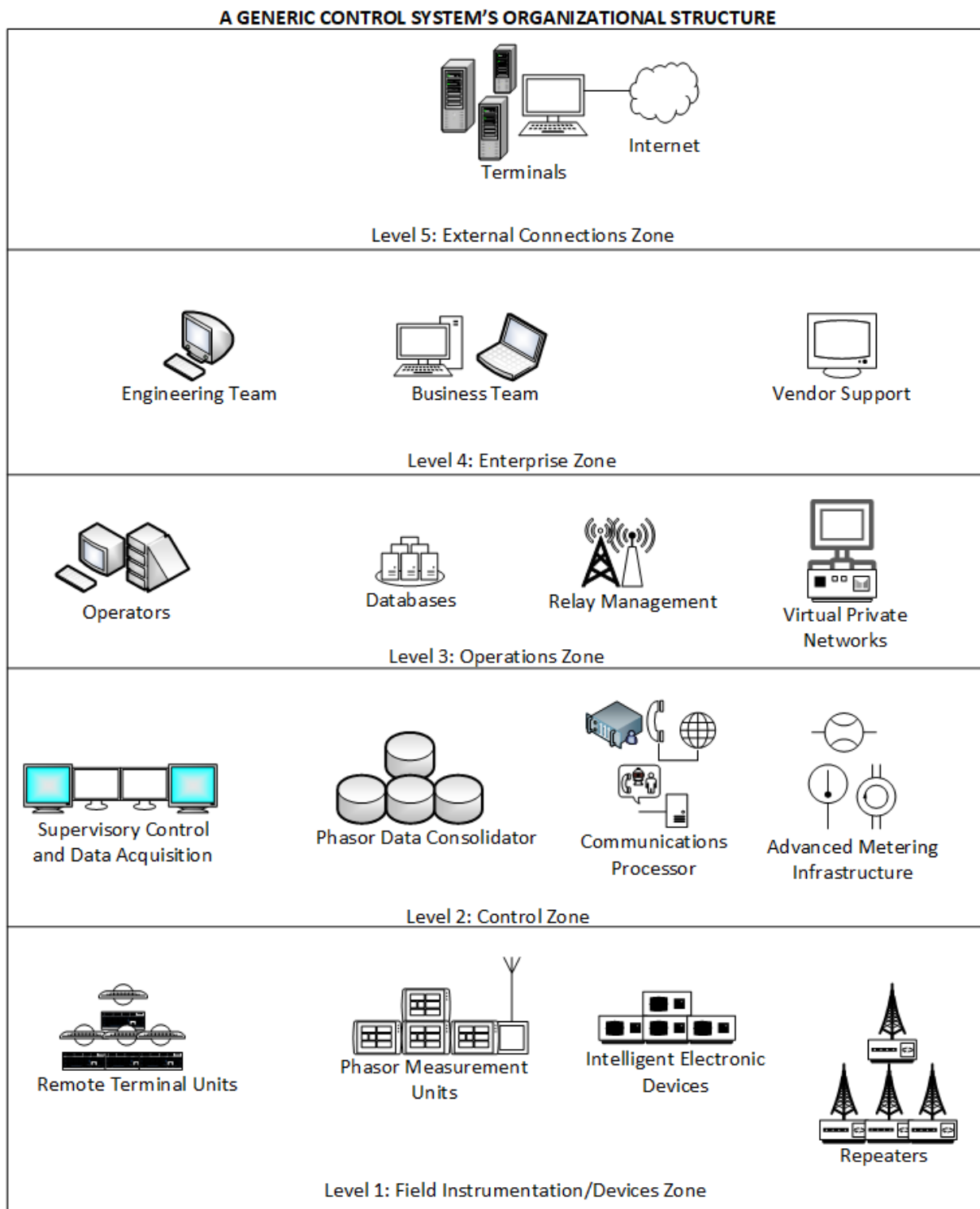


Figure 2.2: Example of a generic control system's organizational structure. The inspiration of this architecture has been derived from [24]'s work.

2.6 DISCUSSION OF HESTIA'S WORKING PRINCIPLES

Figure 2.1 represents a high-level, pictorial representation of HESTIA's workflow. We have developed a language for high-level CPS infrastructure specification. This language is called HERMES, which stands for *High-level, Easily Reconfigurable Machine Environment Specification* [9]. HERMES has also been used as a platform-, and application-independent policy specification language in a high fidelity policy-based web-browser configuration tool called HiFiPol:Browser [25, 26]. Some characteristics of HERMES are:

1. High-level and easily reconfigurable policy specification language.
2. Can specify domain information like groups, sub-groups, users, roles, and devices.
3. Can specify policy information like parent, target, status, field and rationale.
4. Independent of any particular application and platform, i.e., an operating system.

An example CPS' organizational structure is represented by Figure 2.2. The respective organizational hierarchy of this system is represented by Figure 3.2. The corresponding HERMES specification is shown in Listings 2.1 and 2.2. There is no restriction in HERMES with respect to configuration options available. HERMES enables the specification of any configuration option as long as HERMES' nomenclature convention is adhered to [9]. The inspiration behind this example is [24]'s work.

Listing 2.1: HERMES' snapshot specification snippet corresponding to the Figure 2.2 (Part 1 of 2). This listing's corresponding organizational hierarchy is provided in Figure 3.2.

```

1  Domain: CNS
2  {
3  Description: "Control System"
4  List: [CNC, VNS]
5  }
6  SubDomain: CNC
7  {
8  Description: "Control Center"
9  List: [OPS, PDB, RMG]
10 Parent: ACME
11 }
12 ...
13 SubSubDomain: OPS

```

```
14 {
15   Description: "Operator Console"
16   List: [SCADA]
17   Parent: CNC
18 }
19 ...
20 CenterDevices: SCADA
21 {
22   Description: "SCADA"
23   List: [RTU]
24   Parent: OPS
25 }
26 ...
27 FieldDevices: PMU
28 {
29   Description: "Phasor Measurement Unit"
30   List: [PMU.ALL,001,004,006]
31   Parent: PDC
32 }
33 PMUDevices: PMU.ALL
34 {
35   Description: "All Phasor Measremnt Units"
36   Configuration: message-encryption=YES
37   Parent:PMU
38 }
39 PMUDevices: PMU.001
40 {
41   Description: "PMU with an ID of 001"
42   Configuration: message-encryption=YES
43   Parent:PMU
44 }
```

Listing 2.2: HERMES' snapshot specification snippet corresponding to the Figure 2.2 (Part 2 of 2). This listing's corresponding organizational hierarchy is provided in Figure 3.2.

```

1  PMUDevices: PMU.004
2  {
3  Description: "PMU with an ID of 004"
4  Configuration: message-encryption=NO
5  Parent:PMU
6  }
7  PMUDevices: PMU.006
8  {
9  Description: "PMU with an ID of 006"
10 Configuration: message-encryption=NO
11 Configuration: valid-exception=YES
12 Parent:PMU
13 }
14 ...

```

2.7 CONSISTENCY AND APPLICABILITY

Consistency of CPS Specifications: In this work, we define two types of inconsistencies. The first type is direct inconsistency, where multiple occurrences of the same canonical ‘key:value’ pairs are checked for inconsistent values. The second type is hierarchical inconsistency, which occurs when the value of a canonical property as stated by a child object differs from the value specified by a parent for the same property. Valid exceptions are considered only when explicitly declared. Examples of both direct and hierarchical inconsistencies are provided in Figure 2.3. In this figure, direct inconsistency is identified with the numbers 4 and 5, and hierarchical inconsistency is identified with the number 7. More information on Figure 2.3 is provided in Sub-Section 2.8.

Applicability of Attack and Hardening Actions: The ACE checks for applicability of a given delta specification (action) with respect to a given snapshot specification by evaluating the following three characteristics: 1) overall cost of the suggested delta specification measures, 2) feasibility of applying the delta specification measures, and 3) availability of infrastructure. In the context of this chapter, we limit applicability to delta specifications which do not add new devices or equipment and only limit specification changes to values of existing properties within existing devices.

2.8 APPLYING THE HESTIA PROCESS

Figure 2.3 represents an example application of the HESTIA process. In this figure, there are five specifications: two snapshot specifications and three delta specifications. The relationships between these specifications are denoted with numerics from 1 to 7. Number 1 denotes a *Not Applicable* relationship between Delta Specification 1 and Snapshot Specification 1. This non-applicability is because Delta Specification 1 is suggesting a change in the same property but for a non-existent device (RTU). The device being specified in Snapshot Specification 1 is a PMU.

Number 2 denotes an *Applicable* relationship between Delta Specification 2 and Snapshot Specification 1. The applicability is because Delta Specification 2 is suggesting a change in the value for a property of a PMU device. This is the same device being specified in Snapshot Specification 1. Number 3 denotes a *To Be Implemented in Future* relationship between Delta Specification 3 and Snapshot Specification 1. The relationship has been designated as such because Delta Specification 4 suggests adding a new device into the system represented by Snapshot Specification 1.

Number 4 denotes a direct *Inconsistent* relationship between Snapshot Specification 2 and Snapshot Specification 1. Snapshot Specification 1 and 2 are representative of the same system and are specifications of the same iteration in the HESTIA tool-set. This direct inconsistency is because Snapshot Specification 2 specifies a configuration value which is inconsistent with the configuration value specified by Snapshot Specification 1 for the same property within the same device, which is a PMU. Number 5 denotes a direct *Inconsistent* relationship between two specifications within Snapshot Specification 1. This direct inconsistency is similar to the one represented by Number 4.

Number 6 denotes a *Consistent* relationship between two specifications within Snapshot Specification 1. The first specification of Snapshot Specification 1 is a policy suggesting that all PMU communications in the system should be encrypted. The second specification is a policy suggesting non-encrypted communications when originating from PMU 006. These two specifications are hierarchically consistent with each other because the child specification has an explicitly declared exception, allowing it to deviate from the parent specification.

Number 7 denotes an *Inconsistent* relationship between two specifications within Snapshot Specification 1. The first specification of Snapshot Specification 1 is a policy suggesting that all PMU communications in the system shall be encrypted. The second specification is a policy suggesting that the communication originating from PMU 004 should be clear-text. These two specifications are hierarchically inconsistent with each other because the child specification has no explicitly declared exception allowing it to deviate from the parent specification.

2.9 CHAPTER CONCLUSION

HESTIA is a high-level and extensible risk assessment process and tool-set for critical infrastructures (CPS). In this chapter, we have described the architecture and working principles of HESTIA, including the three main subsystems and the components comprising these subsystems. In addition, we have discussed the working principles involved in HESTIA's development. We have also described remaining challenges and strategies for addressing these challenges. We hope that this endeavor will contribute to solving the problem of enabling a Chief Security Officer or the Security Operators to design the best hardening strategy for their particular CPS system. Thereby, help prevent or reduce the the likeliness of successful attacks in CPS.

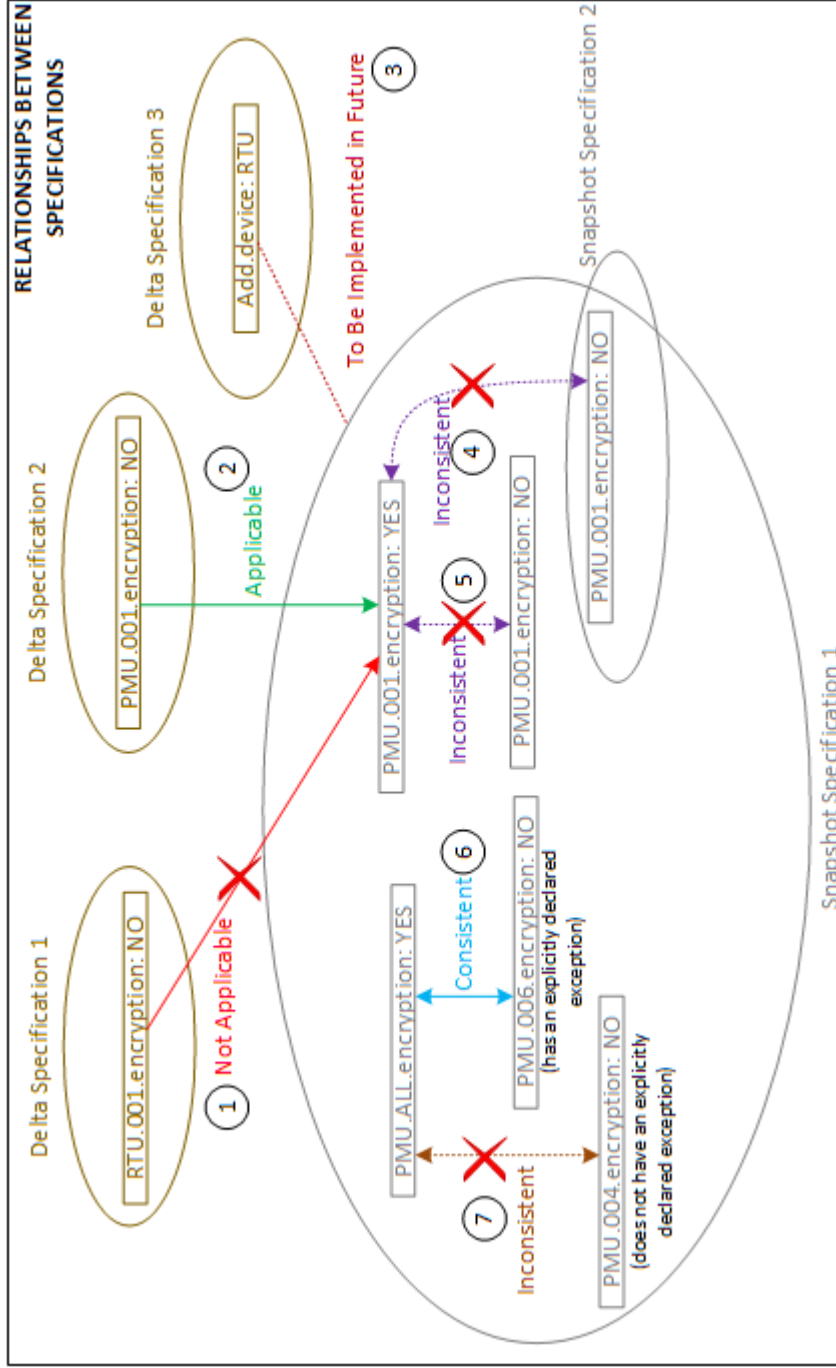


Figure 2.3: Relationship aspects of specifications. PMU: Phasor Measurement Unit and RTU: Remote Terminal Unit.

CHAPTER 3: ARCHITECTURE OF HESTIA

3.1 CHAPTER INTRODUCTION

In modern Cyber Physical Systems (CPS), cyber-enabled and remotely managed devices are replacing previously locally managed devices [4]. This transition has resulted in the addition of new vulnerabilities due to increased connectivity and lack of adequate evaluation and management of security risks [5]. In recent years, attacks on CPS have become more common [5, 27]. The fast-paced introduction of cyber-enabled equipment is thus accelerating the increased risk of cyber-attacks [6]. In December 2015, three power distribution systems were compromised in a large-scale coordinated cyber-attack. This resulted in the loss of power for 225,000 people for 7 hours [27]. There have been reports of several other cyber-attacks against cyber-physical control systems resulting in widespread power outages [28]. Cyber-attacks against CPS are resulting in losses to governments, industry, and the economy [7]. Economic losses due to cyber-attacks on CPS are estimated to reach almost \$2 billion by 2018 [7]. Adequately protecting a CPS requires operators to timely and adequately execute a well-informed and continuous process of system evaluation, modeling, risk assessment, and optimal system hardening and improvement. This requires a deep understanding of their systems, devices, and networks, and the system’s vulnerabilities, potential threats and attacks and their associated risks, and the applicable strategies that would enhance security.

3.2 CHAPTER PROBLEM

Currently, identifying the specific vulnerabilities and optimal mitigation strategies for a particular CPS infrastructure is a grand challenge [8]. In order to do this, security engineers need to describe the system and its devices and connectivity, analyze the system for vulnerabilities, evaluate the possible threats and their risks, and design the best hardening strategy that takes into account factors such as applicability and availability of hardening measures and financial, time, and personnel resources. Such a discovery and design process includes investigating questions along the lines of: “how is this system composed and connected?”, “which types of attacks are possible and likely?”, “where to best apply defense resources?”, “which devices or components to harden and how?”, and “in which particular order?”. No model or tool-set known to the authors enables such an adversarial and computer assisted discovery and risk assessment process.

3.3 CHAPTER CONTRIBUTION

We argue that an adversarial- and specification-based, iterative, and computer-assisted modeling

and risk assessment process is needed to help address this problem. In this chapter we propose such a process and its associated tool-set and describe its architecture and working principles. This process and tool-set, called HESTIA, will facilitate: 1) Concrete and detailed specification of a system, its devices, their connectivity, and the characteristics of the devices; 2) Identification of system vulnerabilities and applicable threats; and 3) Identification of risk abatement strategies. In addition, HESTIA will enable the training of security engineers in the process of identifying CPS vulnerabilities and designing a hardening strategy.

HESTIA stands for *High-level and Extensible System for Training and Infrastructure risk Assessment*. When fully developed, HESTIA will be able to: 1) model an existing CPS infrastructure in the form of specifications (snapshot); 2) model the corresponding attacks and hardening measures in the form of specifications (delta); 3) check both the snapshot and the delta specifications for consistency; 4) check if a delta specification is applicable to a snapshot specification; and 5) if applicable, merge the delta and snapshot specifications to produce an updated snapshot specification. HESTIA conducts this process iteratively and semi-automatically. As such, HESTIA enables a security engineer to design the optimal or best hardening strategy for a particular CPS infrastructure.

3.4 CONTEXT IN PREVIOUS WORK

HESTIA is one branch of an umbrella research project that aims to improve the state-of-the-art in cybersecurity and risk management for CPS. Other areas in this umbrella project are: designing and implementing a CPS testbed for cybersecurity research and development, analyzing and designing secure energy storage systems, designing and testing security controls for High Voltage Direct Current (HVDC) systems, detecting stealth or masked attacks on the power grid using machine learning and data analytics, and analyzing, implementing, and enforcing least-privilege CPS designs and configurations.

This chapter is a shortened and updated version of a previous chapter titled *An Architecture for HESTIA* to appear in the International Journal of Internet of Things and Cyber Assurance [1]. Here, we are also introducing two new contributions. First, we have updated the three subsystems in HESTIA (Figure 3.1) by adding two new components. These new components are: **Device Specification Templates** and **System Policies and Configurations**. Details of the updated architecture (v 2.0) are given in Sections 3.7, 3.8, and 3.9. Descriptions of the newly introduced components are provided in Section 3.10. Second, we have developed solutions to two previously unaddressed applicability problems. These are: **Adding a new device** and **Adding a new characteristic for an existing device**. Details of these two problems and the corresponding solutions are presented in Section 3.11.

3.5 CHAPTER OUTLINE

The rest of this chapter is organized as follows: Section 3.6 presents the updated (version 2.0) architecture of HESTIA. Sections 3.7, 3.8, and 3.9 discuss HESTIA’s three subsystems, which are: *Snapshot Specification Development Subsystem*, *Delta Specification Development Subsystem*, and *Specification Application Subsystem*. Section 3.10 lays out descriptions of the components of HESTIA’s updated architecture. Section 3.11 presents solutions to two new applicability problems related to the addition of new devices. Section 3.12 describes related research and differences with HESTIA. A conclusion provides an ending to this chapter.

3.6 ARCHITECTURE OF HESTIA

When fully developed, HESTIA will be able to take a CPS infrastructure’s original system specifications as input and iteratively subject that specification to changes. These changes occur in the form of applicable attacks and/or hardening specifications, resulting in transformation of a system specification into a new system specification. The new system specification can then be subjected to another iteration of the same process, as determined by the security operator. For every iteration, either an attack or a hardening specification is chosen by the security operator, from a library of attack and hardening specifications. This library of attack and hardening specifications is created by security subject matter experts in collaboration with critical infrastructure engineers and operators.

The HESTIA’s architecture, as depicted in Figure 3.1, is divided into three subsystems: (A) Snapshot Specification Development Subsystem, (B) Delta Specification Development Subsystem, and (C) Specification Application Subsystem.

3.7 SNAPSHOT SPECIFICATION DEVELOPMENT SUBSYSTEM (SUBSYSTEM A)

This subsystem is represented with the letter *A* in Figure 3.1. The input of this subsystem is a *Snapshot Specification* file. The snapshot specification is produced by a human-in-the-loop, denoted with the label *HitL: Snapshot Specification* in Figure 3.1.

To produce the snapshot specification, the human-in-the-loop utilizes information provided by pre-existing device specification templates, and system policies and configurations. More information on device specification templates, and system policies and configurations can be found in Section 3.10.

The snapshot specification file is then checked for any inconsistencies by *Consistency Check Engine*

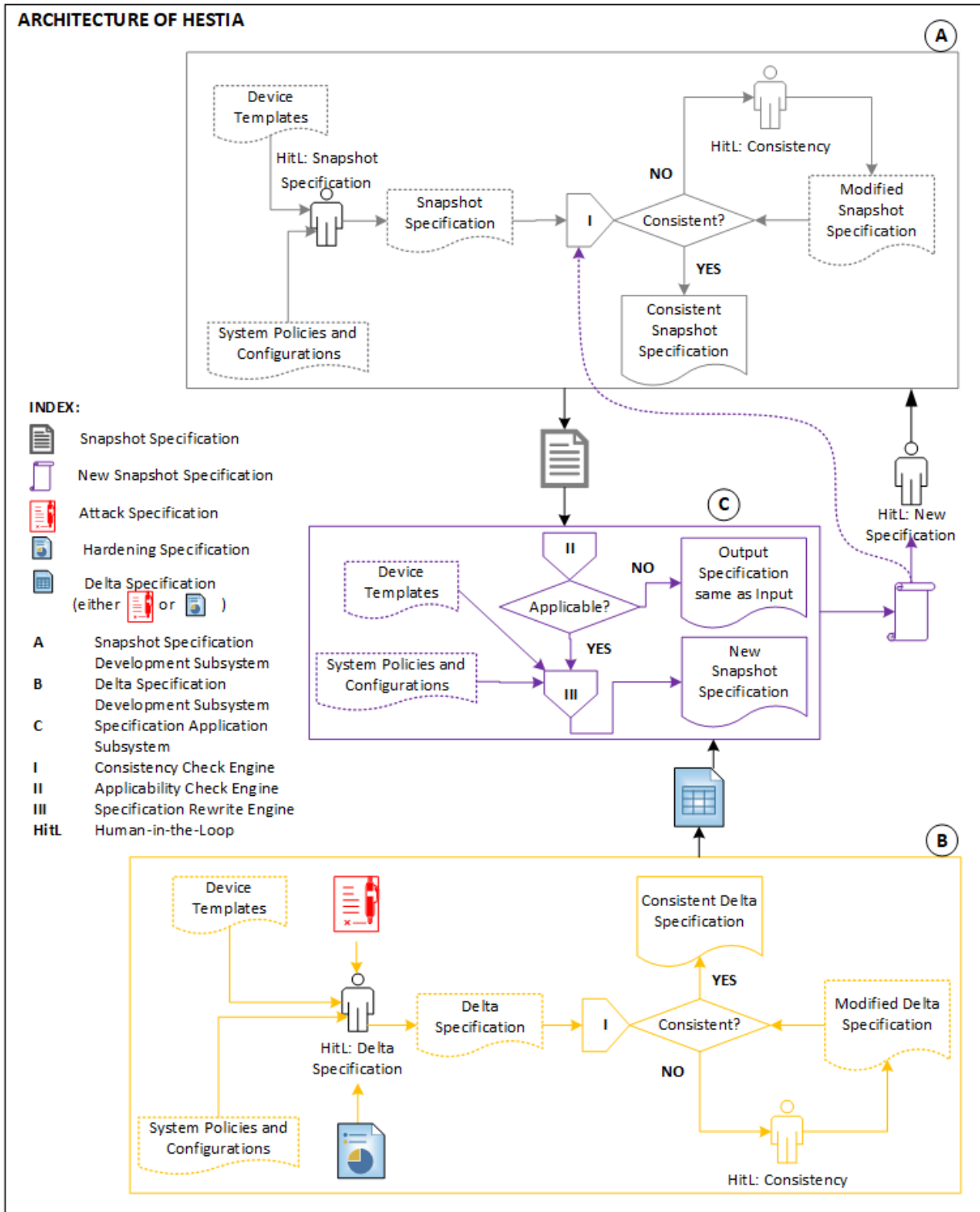


Figure 3.1: Architecture of HESTIA Version 2.0; Updated and enhanced from [1].

(CCE), denoted with the Roman numeral *I* in Figure 3.1. If the snapshot specification file is consistent, the file is forwarded to subsystem *C* (Section 3.9). More information on the process of inconsistency checking and types of inconsistencies is available in a previous chapter [1]. Pending further investigation, the CCE might also check the snapshot specification for completeness of the CPS infrastructure in the snapshot specification file.

If the snapshot specification file is inconsistent, the file is forwarded to a human-in-the-loop (*HitL: Consistency* in Figure 3.1) for corrections. The specification file, which has been corrected by the security operator(s) is called *Modified Snapshot Specification* file. This modified file is then checked by CCE again, until the specification file is consistent.

3.8 DELTA SPECIFICATION DEVELOPMENT SUBSYSTEM (SUBSYSTEM B)

This subsystem is represented with the letter *B* in Figure 3.1. The input of this subsystem is a *Delta Specification* file. The delta specification is produced by a human-in-the-loop, denoted with the label *HitL: Delta Specification* in Figure 3.1.

To produce the delta specification, the human-in-the-loop utilizes information provided by pre-existing device specification templates, and system policies and configurations. More information on device specification templates, and system policies and configurations can be found in Section 3.10. With this information, the human-in-the-loop then chooses a specification from amongst a library of attack or hardening specifications. The library of pre-compiled attack or hardening measures-based specifications is created by security operator(s), via investigation of the latest attack patterns and hardening measures of CPS infrastructure.

The term *Delta Specification* refers to either an attack specification or a hardening specification. The chosen delta specification is then checked for any inconsistencies by CCE, denoted with the Roman numeral *I* in Figure 3.1. If the delta specification file is consistent, the file is forwarded to subsystem *C* (Section 3.9). More information on the process of inconsistency checking and types of inconsistencies is available in [1]. Pending further investigation, the CCE might also check the delta specification for completeness of the attack or hardening measures in the delta specification.

If the delta specification file is inconsistent, the file is forwarded to a human-in-the-loop (*HitL: Consistency* in Figure 3.1) for corrections. The specification file, which has been corrected by the security operator is called *Modified Delta Specification* file. This modified file is then checked by CCE again, until the specification file is consistent.

3.9 SPECIFICATION APPLICATION SUBSYSTEM (SUBSYSTEM C)

This subsystem is represented with the letter *C* in Figure 3.1. There are two inputs for this subsystem: 1) snapshot specification file, which is the output from subsystem *A* (Section 3.7), and 2) delta specification file, which is the output from subsystem *B* (Section 3.8). The *Applicability Check Engine* (ACE), denoted with the Roman numeral *II* in Figure 3.1, checks for applicability of delta specification with respect to several factors. More information on the process of applicability checking and applicability factors is available [1]. If the delta specification is applicable on the given snapshot, the snapshot is rewritten into the *New Snapshot Specification* by the *Specification Rewrite Engine* (SRE), denoted with the Roman numeral *III* in Figure 3.1. SRE incorporates the components of delta specification on the existing snapshot specification, to create the *New Snapshot Specification*.

During the process of rewriting a snapshot specification, the SRE utilizes information provided by pre-existing device specification templates, and system policies and configurations. More information on device specification templates, and system policies and configurations can be found in Section 3.10. With this information, SRE can be programmed to either: a) update the snapshot specification with a compatible characteristic being specified by the delta specification, or b) upgrade the snapshot specification by adding a compatible device being specified by the delta specification, or c) do both items ‘a’ and ‘b’. More information regarding items ‘a’ and ‘b’ in Section 3.11.

The new snapshot specification is then sent to the Consistency Check Engine (CCE), for verification of consistency. If inconsistent, the new snapshot will be modified by a human-in-the-loop (*HitL: Consistency* in Figure 3.1), until consistent. If or when consistent, the new snapshot specification is analyzed by a human-in-the-loop denoted with the label *HitL: New Specification* in Figure 3.1. The human-in-the-loop then decides if the new specification is to be subjected to another iteration of the HESTIA process or not. If another iteration is deemed necessary, the new specification will be forwarded to subsystem *A* (Section 3.7). If not, the new system specification is then used by the security operator(s) for risk assessment of a particular CPS infrastructure.

Listing 3.1: Example CPS specification in HERMES. Adapted from [1].

```

1 Domain: CNS
2 {
3     Subsystems: [CNC, VNS];
4 }
5 ...

```

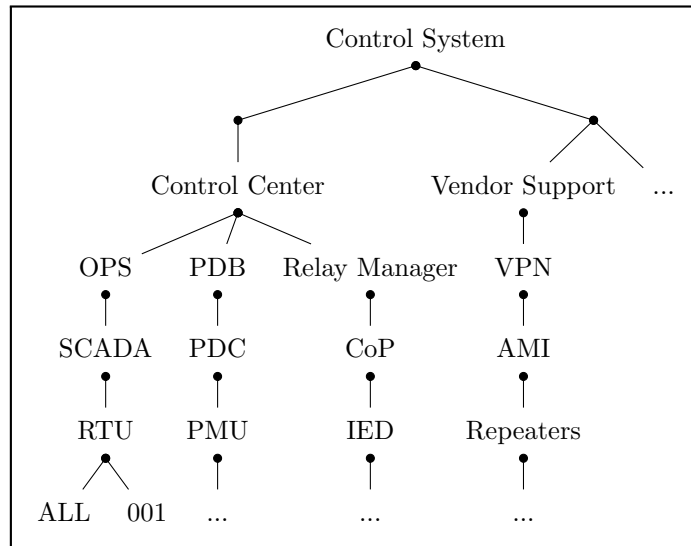


Figure 3.2: Example of a control system’s organizational hierarchy. Index for abbreviated terms: OPS: Operator Console, SCADA: Supervisory Control and Data Acquisition, RTU: Remote Terminal Units, PDB: Phasor Database, PDC: Phasor Data Consolidator, PMU: Phasor Measurement Units, CoP: Communications Processor, IED: Intelligent Electronic Devices, VPN: Virtual Private Networks, and AMI: Advanced Metering Infrastructure. Corresponding HERMES specification snippet is provided in Listing 3.1. From [1]. Based on work from Mahan et al. [24].

```

6 SubDomain: CNC
7 {
8   Components: [OPS, PDB, RMG];
9 }
10 ...
11 SubSubDomain: OPS
12 {
13   Components: [SCADA];
14 }
15 ...
16 CenterDevices: SCADA
17 {
18   Components: [RTU, ...];
19 }
20 ...
21 FieldDevices: RTU

```

```

22 {
23     Components: [RTU.ALL];
24 }
25 ...
26 RTUDevices: RTU.ALL
27 {
28     Config.: message-encrypt=YES;
29     Config.: geo-transmit=YES;
30 }

```

3.10 DEVICE SPECIFICATION TEMPLATES

In this chapter, we are introducing two new concepts called *Device Specification Templates* and *System Policies and Configurations*. A device specification template is a collection of all the characteristics and properties of a device, in a specification format. The device specification template may or may not contain default values for the characteristics of a device. There exists one device specification template for every type of CPS device. These device specification templates maybe: 1) supplied by infrastructure vendors, or 2) written by security engineers of a CPS organization, or 3) a combination of 1) and 2).

A system policy and configuration is a collection of characteristics and properties of all devices being implemented in a CPS. The system policy and configuration will usually contain deployment-time values of all devices involved in deployment of the CPS infrastructure. There exists one system policy and configuration for every type of CPS infrastructure's deployment plan. These system policy and configuration files maybe: 1) written by security engineers of a CPS organization, or 2) supplied by infrastructure vendors, or 3) a combination of both 1 and 2 items.

3.11 APPLICABILITY PROBLEMS AND SOLUTIONS

In a previous chapter [1], we left two applicability problems unaddressed. The two problems are as follows. First, we considered a delta specification to not be applicable when its hardening measure entails adding a new device to a snapshot specification. Second, we considered a delta specification to not be applicable when it suggests a hardening measure that entails adding a new characteristic to an already existing device. In this chapter, we address these two problems by updating the HESTIA process, as indicated below.

First, we consider the delta to be applicable, when the delta specification suggests a hardening measure

which entails adding a new device to the snapshot specification. This applicability is valid as long as the device has a corresponding device specification template. The applicable delta is subsequently merged with the snapshot specification by SRE.

Second, we consider the delta to be applicable, when the delta specification suggests a hardening measure which entails adding a new characteristic to a device already existing in the snapshot specification. This applicability is valid as long as the newly suggested characteristic is listed in the corresponding device specification template. The applicable delta is subsequently merged with the snapshot specification by SRE.

3.12 RELATED WORK

One factor motivating us in pursuing this research is the lack of previous work in the area of adversarial-based, specification-based modeling and risk assessment. After extensive literature searches of the current state-of-the-art we were unable to discover reports of work similar to the objectives of HESTIA. We report here on related but unsimilar work.

Zhang et al. proposed modeling a CPS using ModelicaML [10]. Zhang et al. also proposed using aspect-oriented descriptions for real-time requirements in CPS [11]. MITRE developed a cybersecurity risk assessment process called *Making Security Measurable* [12]. This process helps with assessment of vulnerabilities, threats, and functionality. Jillepalli et al. applied the Cybernomics method developed by Sheldon, Abercrombie, and Mili within the context of a CPS and the NIST SP 800-82r2 ICS security guidelines [21]. Cybernomics is a stakeholder-aware and economics-based risk assessment method. These methods provide processes for risk assessment based on multiple system factors. By contrast, HESTIA aims to provide an adversarial- and specification-based and semi-automatic process and tool-set for modeling and risk assessment of CPS.

3.13 CHAPTER CONCLUSION

In this chapter, we have described the updated architecture of HESTIA, including updates to the three main subsystems. We have presented solutions to two of the previously unaddressed applicability problems. We have also briefed the readers regarding the status of the research and described near term future work, which will help make the proposed system, HESTIA, a reality. We hope that this endeavour will contribute to solving the problem of enabling a CSO to design the best hardening strategy for their particular CPS system and help prevent and reduce the ever-increasing CPS infrastructure-based attacks.

CHAPTER 4: A FORMAL MODEL FOR THE HESTIA PROCESS

4.1 CHAPTER INTRODUCTION

Annual damages related to cyber-attacks are predicted to reach \$6 Trillion by 2021, according to a 2019 Cybersecurity Ventures report [29]. Two factors have the potential to minimize the degree of predicted financial losses. The factors are possessing an ability to: 1) qualify threats against an organization's infrastructure; and 2) organize hardening measures to help protect against threats [5]. Assessing the risk of an organization's critical infrastructure using an adversarial-based process may be effective in identifying security gaps of the organization [1]. Conducting such a risk assessment using manual mechanisms could end up being a costly affair [2]. A risk assessment approach involving utilization of specifications-based mechanisms is of interest due to the ease-of-validation nature of specifications [30]. As such, automating the risk assessment process by using a specifications-based policy auditing approach may help in securing an organization's infrastructure at a lower cost [31].

4.2 CHAPTER PROBLEM

Non-existence of a specifications- and adversary-based risk assessment process is handicapping efforts to automate critical infrastructure security. According to subject matter experts (SMEs), such a non-existence is a result of three factors. They are: 1) research in the area of critical infrastructure security has been prominent recently, since 2008 [5]; 2) creating and implementing a novel process for risk assessment is not a trivial task; and 3) although the critical infrastructure organizations are starting to invest resources in cybersecurity, they remain unconvinced to support efforts to develop a novel risk assessment process.

Risk assessment efforts within critical infrastructure organizations remain manual in nature due to the non-existence of a specifications- and adversary-based risk assessment process. A manual process takes time, where the time required to assess an organization increases exponentially with the number of devices deployed in the organization. The time-taking nature of manual risk assessment processes may deter organizations from using them.

4.3 CHAPTER CONTRIBUTION

To address the problem, we presented the architecture and a partial formalization of HESTIA in previous chapters [1, 2, 31]. HESTIA is an iterative, specification- and adversary-based risk assessment

process. The working principle of HESTIA is driven by interactions between specifications. We present a complete semantic formal model of the HESTIA process.

HESTIA’s formal model can be represented using any Domain Specific Language (DSL) [32]. Any DSL can be used to represent HESTIA formal model as long as the DSL has the following features: 1) represent organizational domain knowledge, across both technical and managerial domains; 2) encode policies as specifications at a granular level, down to the lowest possible node; 3) modularly express specifications in a high-level, English-like, language; 4) function independently of the platform being used; and 5) separate sets of specifications into fully independent entities. We decided to use the HERMES language in our research because its capabilities include the features listed above. The evolving grammar and syntactical notation of the HERMES language has been presented in previous theses [9], [33].

4.4 CHAPTER OUTLINE

The rest of this chapter is organized as follows: Sections 4.7 and 4.17 discuss the formal model of specifications and its constituent components. Sections 4.16, 4.22 and 4.24 present details about the special operations being introduced by the HESTIA process. Section 4.34 introduces HESTIA’s design and presents HESTIA’s workflow. Sections 4.35, 4.36, 4.37, and 4.38 present a formal representation of HESTIA’s specification-relations processing. These relations are: consistency- and conflict-checking. Previous chapters discussed related work items for HESTIA and their differences with HESTIA. Chapter conclusion and a list of abbreviations follows.

4.5 A SEMANTIC FORMAL MODEL

We discuss preliminary concepts that provide a context for our formalization of the HESTIA process.

We formalize the HESTIA process semantically and not syntactically. That is, we model the components involved in HESTIA’s working principle and the operations involved in the HESTIA process. We do not model syntax of the working principle components or operations. For example we model Specifications, a component involved in HESTIA working principle, but we do not model the syntax in which Specifications are used. For the course of our formal model, we use HERMES syntax, which has been defined in previous chapters [9], [33]. However, one may use any syntax to implement the semantic model we present here. By modeling the semantics and not the syntax, we hope that the formal model will be used in implementations that are syntactically different than the HERMES syntax.

4.6 UNIVERSES, CARDINALS, AND INDEXES

A universe is a collection of all instances of a particular component. For example, \mathcal{S} is a specification

universe, which means it is a collection of all instances of Specifications. We use two types of component references, which are: cardinals and indexes. A cardinal of a component represents one given instance of a component in the component universe. The cardinal is denoted by the superscript of that component. For example, S^k should be understood as being the k th instance of the component S that belongs to the component universe \mathcal{S} . An index of a component represents the component's position in a sequence and is denoted by the subscript of that component. For example, p_i should be understood as being the i th position of the component p in a sequence.

4.7 SPECIFICATIONS AND CONSTITUENT COMPONENTS

HESTIA's process deals with specifications at the highest level of abstraction. We discussed HESTIA's process and working principles, based on specifications in previous chapters. In the following sections, we formalize specifications and their constituent components as:

4.8 SPECIFICATIONS

HESTIA's process deals with specifications at the highest level of abstraction. We discussed HESTIA's process and working principles, based on specifications in previous chapters. In the following sections, we formalize specifications and their constituent components as:

A CPS system's Specification, S^k , is an explicitly defined unordered set of Entities, $\{E^a, E^b, \dots, E^n\}$. The set of all specifications is represented by \mathcal{S} , to which the Specification S^k belongs. Definition 1 presents a formal notation of HESTIA's specifications. Throughout this document, the notation \overline{def} should be understood as 'is defined as'.

Definition 1 *Specifications*

$$\{S^a, S^b, \dots, S^p\} \in \mathcal{S}$$

$$S^k \in \mathcal{S}$$

$$S^k \overline{def} \{E^a, E^b, \dots, E^n\}$$

Example: Listing 4.1 presents a specification.

4.9 ENTITIES

An Entity, E^k , is composed of an Entity Identifier, $(EID)^k$, an Entity Type, $(ET)^k$, and a set of Fields, $\{F^a, F^b, \dots, F^n\}$. The set of all entities is represented by \mathcal{E} , to which the Entity E^k belongs. An

$(EID)^k$ and $(ET)^k$ are dotted strings, Q^k , that identify the ID and type of any given entity, E^k . An EID value can also be the keyword “**Template**”, which defines that the entity belongs to a template and not a specification. A formal definition of templates can be found in Sub-Section 4.11. Definition 2 presents a formal notation of HESTIA’s entities. A formal definition of dotted strings can be found in Sub-Section 4.13.

Definition 2 *Entities Specifications*

$$\begin{aligned} \{E^a, E^b, \dots, E^n\} &\in \mathcal{E} \\ E^k &\in \mathcal{E} \\ E^k &\stackrel{=}{def} ((EID)^k \wedge (ET)^k \wedge \{F^a, F^b, \dots, F^n\}), \text{ where :} \\ (ET)^k &= Q^k \\ (EID)^k &= Q^k \vee \\ (EID)^k &= \textit{Template} \end{aligned}$$

Example: Listing 4.1 presents two entities. The (EID) of the two entities are: *RTUDevices.001* and *RTUDevices.002*.

4.10 FIELDS

A Field, F^k , consists of a Field Name, $(FN)^k$, and a Field Register R^k . Where, $(FN)^k$ is an `AlphanumericString`, defining a name for the field. R^k can either be a value, a dotted string, a dictionary, or a named set. The set of all fields is represented by \mathcal{F} , to which the Field F^k belongs. Definition 3 presents a formal notation of HESTIA’s fields. A formal definition for `AlphanumericStrings`, values, dotted strings, and dictionaries can be found in Section 4.17.

Definition 3 *Fields*

$$\begin{aligned} \{F^a, F^b, \dots, F^n\} &\in \mathcal{F} \\ F^k &\in \mathcal{F} \\ F^k &\stackrel{=}{def} ((FN)^k \wedge R^k), \text{ where :} \\ (FN)^k &= \textit{AlphanumericString} \wedge \\ R^k &= (v^n \vee Q^n \vee D^n \vee L^n \vee \emptyset) \end{aligned}$$

Example: Listing 4.1 presents several fields. The (FN) of one such field is: *messageEncrypt*.

4.11 TEMPLATES

A Template, T^k , is a specification whose entities, E^k , have fields, F^k , that have empty field registers. The set of all templates is represented by \mathcal{T} , to which the Template T^k belongs. Definition 4 presents a formal notation of HESTIA's Templates. A formal definition for `AlphanumericStrings` can be found in Sub-Section 4.12

Definition 4 *Templates*

$$\{T^a, T^b, \dots, T^n\} \in \mathcal{T}$$

$$T^k \in \mathcal{T}$$

$$\{E^1, E^2, \dots, E^k\} \in T^k$$

Example: Listing 4.3 presents a template. The (FN) of one such field is: *messageEncrypt*.

4.12 ALPHANUMERIC STRING

An `AlphanumericString`, p , is a simple string structure that is a concatenation of alphabets and digits. An `AlphanumericString`, p , cannot contain symbols. Definition 5 presents a formal notation of HESTIA's `AlphanumericString` structure.

Definition 5 *AlphanumericString*

$$(a - zA - Z0 - 9)^*$$

Example: Listing 4.1 presents several `AlphanumericStrings`. An example of one is: *RTUConfig*.

4.13 DOTTED STRINGS

A dotted string, Q , is a structure that consists of multiple `AlphanumericStrings`, called sub-strings. A dotted string, Q^k , is of the format ' $p_1.p_2.p_3.\dots.p_i$ '. Where, each of ' $p^1\dots p^i$ ' is a sub-string that is an `AlphanumericString`. We use dots, `.`, as a separator between different sub-strings of a dotted string. One may use another separator if they so wish. The set of all dotted strings is represented by \mathcal{Q} , to which the dotted string Q^k belongs. Definition 6 presents a formal notation of HESTIA's dotted strings.

Definition 6 *Dotted Strings*

$$\{Q^a, Q^b, \dots, Q^n\} \in \mathcal{Q}$$

$$Q^k \in \mathcal{Q}$$

$$Q^k \stackrel{=}{def} (p_1^k.p_2^k.p_3^k\dots.p_i^k), \text{ where :}$$

$$p_i^k = \text{AlphanumericString}$$

Example: Listing 4.1 presents several dotted strings. An example of one is: *RTUDevices.001*.

4.14 FIELD REGISTER TYPES: DICTIONARIES

A Dictionary, D , is an unordered set of key-value pairs $(k : v)$, where keys and values, k and v are both `AlphanumericStrings` and cannot be a named set or a dotted string. The set of all dictionaries is represented by \mathcal{D} , to which the dictionary D^k belongs. We use colon, $:$, as a separator between a key k and a value v in a key-value pair $(k : v)$. One may use another separator if they so wish. Definition 7 presents a formal notation of HESTIA’s dictionaries.

Definition 7 *Dictionaries*

$$\{D^a, D^b, \dots, D^n\} \in \mathcal{D}$$

$$D^k \in \mathcal{D}$$

$$D^k \stackrel{=}{def} \{(k : v)^a, (k : v)^b, \dots, (k : v)^i\}, \text{ where :}$$

$$k^i = \text{AlphanumericString}$$

$$v^i = (\text{AlphanumericString} \vee (\text{EID}))$$

Example: Listing 4.1 presents several dictionaries. An example of one is: $\{City : \text{“Seattle”}, State : \text{“Washington”}\}$. Note that the double quotes around v values in the given example are due to HERMES syntactic implementation and are not required by the semantic model.

4.15 FIELD REGISTER TYPES: NAMED SETS

A Named Set, L , comprises of an unordered set of values, v . The set of all named sets is represented by \mathcal{L} , to which the named set L^n belongs. Definition 1 presents a formal notation of HESTIA’s named sets.

Definition 8 *Named Sets*

Listing 4.1: HERMES excerpt of a System Specification.

```

1  %System Specification S
2  RTUDevices.001: RTUConfig
3  {
4      messageEncrypt: YES;
5      geoTransmit: YES;
6      location: {City: 'Pullman', State: 'Washington'}
7      commAccess: [RTUDevices.001]
8  }.
9  .
10 .
11 .
12 RTUDevices.002: RTUConfig
13 {
14     messageEncrypt: YES;
15     geoTransmit: YES;
16     location: {City: 'Pullman', State: 'Washington'}
17     commAccess: ['RTUDevices.001', 'RTUDevices.002']
18 }.
19

```

Listing 4.2: HERMES excerpt of a Delta Specification.

```

1  %Delta Specification M
2  RTUDevices.001: RTUConfig
3  {
4      messageEncrypt: YES;
5      geoTransmit: NO;
6      location: {City: 'Seattle', State: 'Washington'}
7      commAccess: ['RTUDevices.001']
8  }.
9  .
10 .
11 .
12 RTUDevices.002 RTUConfig
13 {
14     messageEncrypt: YES;
15     geoTransmit: NO;
16     location: {City: 'Pullman', State: 'Washington'}
17     commAccess: ['RTUDevices.001', 'RTUDevices.002']
18 }.
19

```

$$\{L^a, L^b, \dots, L^n\} \in \mathcal{L}$$

$$L^k \in \mathcal{L}$$

$$L^k \stackrel{=}{def} [v^a, v^b, \dots, v^i], \text{ where :}$$

$$v^i = \{\text{AlphanumericString} \vee (EID)\}$$

Example: Listing 4.1 presents several named sets. An example of one is: [RTUDevices.001].

Listing 4.3: HERMES excerpt of a Template.

```

1      %Template T
2      Template: RTUConfig
3      {
4          messageEncrypt;
5          geoTransmit;
6          location;
7          comAccess;
8      }.
9      .
10     .
11     .
12

```

4.16 ALPHANUMERIC STRING EQUALITY OPERATION

The `AlphanumericString` Equality operation, ‘ASE’, is a binary boolean operation that matches any two given `AlphanumericStrings`. The string matching occurs via standard alphanumeric comparison of each index value of both the `AlphanumericStrings` [34]. If two `AlphanumericStrings` contain the same value, the equality returns the flag ‘T’. If two `AlphanumericStrings` do not contain the same value, the equality returns the flag ‘F’. Definition 9 presents the formal notation of HESTIA’s `AlphanumericString` Equality operation.

Definition 9 *AlphanumericString Equality operation*

Given two AlphanumericStrings, v^a and v^b :

$$(ASE(v^a, v^b) = T) \Leftrightarrow (v^a = v^b) \vee$$

$$(ASE(v^a, v^b) = F) \Leftrightarrow (v^a \neq v^b)$$

Examples: The following are examples for the `AlphanumericString` equality operation, ‘ASE’:

Let $v^a = Alpha01$ and

$v^b = Alpha00$

$$\Rightarrow (ASE(v^a, v^b) = F)$$

Let $v^a = Alpha02$ and

$v^b = Alpha02$

$$\Rightarrow (ASE(v^a, v^b) = T)$$

4.17 FIELD REGISTER TYPES: VALUES

As defined in Sub-Section 4.10, a field register can contain either a value, a dotted string, a dictionary, or a named set. We formally define the different types of field registers.

A Value, v , consists of an `AlphanumericString` or an `EID`. The set of all values is represented by \mathcal{V} , to which the value v^k belongs. Definition 10 presents a formal notation of HESTIA's values.

Definition 10 *Values*

$$\{v^a, v^b, \dots, v^n\} \in \mathcal{V}$$

$$v^k \in \mathcal{V}$$

$$v^k = (\text{AlphanumericString} \vee (\text{EID}))$$

Example: Listing 4.1 presents several values. An example of one is: `NO`.

4.18 FIELD REGISTER SIZE DETERMINATION OPERATION

The size determination operation, 'SD', is an operation that returns the size of a given field register. The size determination operation identifies the size of a value, dotted string, named set, or a dictionary. Definition 11 presents the formal notation of HESTIA's size determination operation.

Definition 11 *Size Determination operation*

Given a field register R^k where

R^k is one of $\{\mathcal{Q}, \mathcal{D}, \mathcal{L}\}$:

$$SD(R^k) = 'i' \text{ if}$$

$$R^k = p_1^k.p_2^k.\dots.p_i^k \vee$$

$$R^k = [v_1^k, v_2^k, \dots, v_i^k] \vee$$

$$R^k = \{(k : v)_1^k, (k : v)_2^k, \dots, (k : v)_i^k\}$$

$SD(R^k) = 1$ at all other times

Examples: The following are examples for the size determination operation, 'SD':

Let $R^a = 'RTU.onsite.001'$ and

$R^b = 99$ and

$R^c = [RTU, PMU, SCADA, OPS]$

$\Rightarrow (SD(R^a) = 3)$

$\Rightarrow (SD(R^b) = 1)$

$\Rightarrow (SD(R^c) = 4)$

4.19 FIELD REGISTER SIZE COMPARISON OPERATION

The size comparison operation, ‘SC’, is an operation that compares the sizes of any two given field registers. If two field registers contain the same size, the comparison operation returns the flag ‘E’. If the size of the left operand is greater than the size of the right operand, the operation returns the flag ‘L’. If the size of the right operand is greater than the size of the left operand, the operation returns the flag ‘R’. Definition 12 presents the formal notation of HESTIA’s size comparison operation.

Definition 12 *Size Comparison operation*

Given two field registers, R^a and R^b :

$(SC(R^a, R^b) = E) \Leftrightarrow (SD(R^a) = SD(R^b))$

$(SC(R^a, R^b) = L) \Leftrightarrow (SD(R^a) > SD(R^b))$

$(SC(R^a, R^b) = R) \Leftrightarrow (SD(R^a) < SD(R^b))$

Examples: The following are examples for the size comparison operation, ‘SC’:

Let $R^a = 'RTU.onsite.001'$ and

$R^b = 99$ and

$R^c = [RTU, PMU, SCADA, OPS]$ and

$R^c = 100$

$\Rightarrow (SD(R^a, R^b) = L)$

$\Rightarrow (SD(R^a, R^c) = R)$

$\Rightarrow (SD(R^b, R^c) = E)$

4.20 DOTTED STRING EQUALITY OPERATION

The dotted string equality operation, ‘DSE’, is an iterative operation that runs the basic `AlphanumericString` equality operation on individual sub-strings of any two given dotted strings. Since a dotted string is a concatenation of `AlphanumericStrings` with dots as separators, we can use the ‘ASE’ function iteratively on each sub-string to check if two dotted strings are equal. If two sub-strings of a dotted string contain the same `AlphanumericString` content, the operation checks iteratively for the next sub-string’s content until the two given dotted strings are exhausted. The operation returns the true flag, ‘T’, if and only if all the values of the given two dotted strings are equal. If not equal, the equality returns a false flag ‘F’. Definition 13 presents the formal notation of HESTIA’s dotted string equality operation.

Definition 13 *Dotted String Equality operation*

Given two dotted strings, Q^a and Q^b :

$$\forall ((p^a \in Q^a) \wedge (p^b \in Q^b)),$$

$$((DSE(Q^a, Q^b) = T) \Leftrightarrow (SC(Q^a, Q^b) = E \wedge \forall ((p^a \in Q^a) \wedge (p^b \in Q^b))), (\exists! (ASE(p^a, p^b) = F)) \vee$$

$$(DSE(Q^a, Q^b) = F) \Leftrightarrow (SC(Q^a, Q^b) = E \wedge \forall ((p^a \in Q^a) \wedge (p^b \in Q^b))), (\exists (ASE(p^a, p^b) = F)))$$

Examples: The following are examples for the dotted string equality operation, ‘DSE’:

Let $Q^a = \text{‘RTU.onsite.001’}$ and

$Q^b = \text{‘RTU.parent.001’}$

$$\Rightarrow (DSE(Q^a, Q^b) = F)$$

Let $Q^a = \text{‘RTU.onsite.001’}$ and

$Q^b = \text{‘RTU.onsite.001’}$

$$\Rightarrow (DSE(Q^a, Q^b) = T)$$

4.21 DOTTED STRING INTERSECTION IDENTIFICATION OPERATION

The dotted string intersection identification operation, ‘DSII’, is an iterative operation that runs the basic `AlphanumericString` equality operation on sub-strings of any two given dotted strings. Since

individually, the sub-strings are `AlphanumericStrings`, we can use the ‘ASE’ function iteratively on each sub-string of the dotted string to check if there exists the same sub-strings(s) across two non-equal dotted strings. If any two sub-strings of a dotted string contain the same content, the operation returns a true flag, ‘T’. If not, the operation returns a false flag ‘F’. Definition 14 presents the formal notation of HESTIA’s dotted string intersection identification operation.

Definition 14 *Dotted String Intersection Identification operation*

Given two dotted strings, Q^a and Q^b ,

$$\forall ((p^a \in Q^a) \wedge (p^b \in Q^b)),$$

$$(DSII(Q^a, Q^b) = T) \Leftrightarrow (\forall ((p^a \in Q^a) \wedge (p^b \in Q^b)), (\exists (ASE(p^a, p^b) = T)) \vee$$

$$(DSII(Q^a, Q^b) = F) \Leftrightarrow (\forall ((p^a \in Q^a) \wedge (p^b \in Q^b)), (\exists! (ASE(p^a, p^b) = T)))$$

Examples: The following are examples for the dotted string intersection identification operation, ‘DSE’:

Let $Q^a = \text{‘RTU.onsite.001’}$ and

$Q^b = \text{‘RTU.parent.001’}$

$\Rightarrow (DSII(Q^a, Q^b) = T)$

Let $Q^a = \text{‘RTU.onsite.001’}$ and

$Q^b = \text{‘PMU.parent.003’}$

$\Rightarrow (DSII(Q^a, Q^b) = F)$

4.22 ENTITY ID COMPARISON OPERATION

The Entity ID Comparison operation, ‘EIC’, compares the name of any given two entities, E^a and E^b . Such a comparison occurs by parsing through the entity ID, EID , value of the given two entities. If the EID values of the two entities are equal, the operation returns the true flag ‘T’. If the EID values of the two entities are not equal, the operation returns the false flag ‘F’. Definition 15 presents a formal representation of HESTIA’s Entity ID Comparison operation.

Definition 15 *Entity ID Comparison operation*

$$\begin{aligned} (EIC(E^a, E^b) = T) &\Leftrightarrow (DSE((EID)^a, (EID)^b) = T) \vee \\ (EIC(E^a, E^b) = F) &\Leftrightarrow (DSE((EID)^a, (EID)^b) = F) \end{aligned}$$

Examples: The following are examples for the entity ID comparison operation, ‘EIC’:

$$\begin{aligned} \text{Let } (EID)^a &= \text{‘Entity01’ and} \\ (EID)^b &= \text{‘Entity01’} \\ &\Rightarrow (EIC(E^a, E^b) = T) \\ \text{Let } (EID)^a &= \text{‘RTUDevice.Onsite.001’ and} \\ (EID)^b &= \text{‘RTUDevice.Offsite.001’} \\ &\Rightarrow (EIC(E^a, E^b) = F) \end{aligned}$$

4.23 ENTITY TYPE COMPARISON OPERATION

The Entity Type Comparison operation, ‘ETC()’, compares the type of any two entities, E^a and E^b . Such a comparison occurs by parsing through the Entity Type, ET , values of the given two entities. If the ET values of the two entities are equal, the operation returns the true flag ‘T’. If the ET values of the two entities are not equal, the operation returns the false flag ‘F’. Definition 16 presents a formalization of HESTIA’s Entity Type Comparison operation.

Definition 16 *Entity Type Comparison operation*

$$\begin{aligned} (ETC(E^a, E^b) = T) &\Leftrightarrow (DSE((ET)^a, (ET)^b) = T) \vee \\ (ETC(E^a, E^b) = F) &\Leftrightarrow (DSE((ET)^a, (ET)^b) = F) \end{aligned}$$

Examples: The following are examples for the entity type comparison operation, ‘ETC’:

$$\begin{aligned} \text{Let } (ET)^a &= \text{PMUConfig and} \\ (ET)^b &= \text{RTUConfig} \\ &\Rightarrow (ETC(E^a, E^b) = F) \\ \text{Let } (ET)^a &= \text{RTUConfig and} \\ (ET)^b &= \text{RTUConfig} \\ &\Rightarrow (ETC(E^a, E^b) = T) \end{aligned}$$

4.24 FIELD NAME COMPARISON OPERATION

The Field Name Comparison operation, ‘FNC()’, compares the name of any given two fields, F^a and F^b . Such a comparison occurs by parsing through the field name value, FN , value of the given two fields. If the FN values of the two fields are equal, the operation returns the true flag ‘T’. If the FN values of the two fields are not equal, the operation returns the false flag ‘F’. Definition 17 presents a formal representation of HESTIA’s Field Name Comparison operation.

Definition 17 *Field Name Comparison operation*

$$(FNC(F^a, F^b) = T) \Leftrightarrow (ASE((FN)^a, (FN)^b) = T) \vee$$

$$(FNC(F^a, F^b) = F) \Leftrightarrow (ASE((FN)^a, (FN)^b) = F)$$

Examples: The following are examples for the field name comparison operation, ‘FNC’:

Let $(FN)^a = messageEncrypt$ and

$$(FN)^b = geoTransmit$$

$$\Rightarrow (FNC(F^a, F^b) = F)$$

Let $(FN)^a = location$ and

$$(FN)^b = location$$

$$\Rightarrow (FNC(F^a, F^b) = T)$$

4.25 FIELD REGISTER TYPE DETERMINATION OPERATION

The field register type determination operation, ‘FRTD’, is a basic type determination operation. The operation returns the flag ‘U’ if the given field register contains a value v . The operation returns the flag ‘G’ if the given register contains a dotted string Q . The operation returns the flag ‘C’ if the given register contains a dictionary D . The operation returns the flag ‘I’ if the given register contains a named set L . Definition 18 presents the formal notation of HESTIA’s field register type determination operation.

Definition 18 *Field Register Type Determination operation*

Given a field register, R^a :

$$(FRTD(R^a) = U) \Leftrightarrow (R^a \in \mathcal{V}) \vee$$

$$(FRTD(R^a) = G) \Leftrightarrow (R^a \in \mathcal{Q}) \vee$$

$$(FRTD(R^a) = C) \Leftrightarrow (R^a \in \mathcal{D}) \vee$$

$$(FRTD(R^a) = I) \Leftrightarrow (R^a \in \mathcal{L})$$

Examples: The following are examples for the field register type determination operation, ‘FRTD’:

$$\text{Let } R^a = 9$$

$$\Rightarrow (FRTD(R^a) = U)$$

$$\text{Let } R^a = RTU.onsite.001$$

$$\Rightarrow (FRTD(R^a) = G)$$

$$\text{Let } R^a = \{\text{Cookies} : \text{“Disabled”}\}$$

$$\Rightarrow (FRTD(R^a) = C)$$

$$\text{Let } R^a = [RTUDevices, PMUDevices]$$

$$\Rightarrow (FRTD(R^a) = I)$$

4.26 FIELD REGISTER TYPE COMPARISON OPERATION

The field register type comparison operation, ‘FRTC’, is a basic type checking operation. If two field registers contain the same type of value, the operation returns a true flag ‘T’. If two registers do not contain the same type of value, the check returns a false flag ‘F’. Definition 19 presents the formal notation of HESTIA’s field register type check operation.

Definition 19 *Field Register Type Comparison operation*

Given two field registers, R^a and R^b :

$$(FRTC(R^a, R^b) = T) \Leftrightarrow (FRTD(R^a) = FRTD(R^b)) \vee$$

$$(FRTC(R^a, R^b) = F) \Leftrightarrow (FRTD(R^a) \neq FRTD(R^b))$$

Examples: The following are examples for the field register type comparison operation, ‘FRTC’:

Let $R^a = 9$ and

$R^b = RTU.onsite.001$

$\Rightarrow (\text{FRTC}(R^a, R^b) = \text{F})$

Let $R^a = [\text{RTUDevices}]$ and

$R^b = [\text{RTUDevices}, \text{PMUDevices}]$

$\Rightarrow (\text{FRTC}(R^a, R^b) = \text{T})$

4.27 FIELD REGISTER DICTIONARY EQUALITY OPERATION

The dictionary equality operation, ‘DE’, is an iterative operation that runs the basic `AlphanumericString` equality operation on key-value pairs of any two given dictionaries. Since individually, both the keys and values are `AlphanumericStrings`, we can use the ‘ASE’ function iteratively on each key and value to check if two dictionaries are equal. If two key `AlphanumericStrings` contain the same content, the operation checks for the `AlphanumericString` content of the values, corresponding to the keys. If the two value `AlphanumericStrings` also contain the same content, the operation checks iteratively for the next key-value pair contents until the two given dictionaries are exhausted. The operation returns the true flag, ‘T’, if and only if all the key-value pairs of the given two dictionaries are equal. If not equal, the operation returns a false flag ‘F’. Definition 20 presents the formal notation of HESTIA’s dictionary equality operation.

Definition 20 *Dictionary Equality operation*

Given two Dictionaries, D^a and D^b ,

$(\text{DE}(D^a, D^b) = \text{T}) \Leftrightarrow$

$\text{SC}(D^a, D^b) = \text{E} \wedge$

$\forall (((k : v)^a \in D^a) \wedge ((k : v)^b \in D^b)),$

$((\text{ASE}(k^a, k^b) = \text{T}) \wedge (\text{ASE}(v^a, v^b) = \text{T})) \vee$

Otherwise, $(\text{DE}(D^a, D^b) = \text{F})$

Examples: The following are examples for the dictionary equality operation, ‘DE’:

Let $D^a = \{Cookie : \text{“Enabled”}, JavaScript : \text{“Enabled”}\}$ and

$D^b = \{Cookie : \text{“Enabled”}, JavaScript : \text{“Disabled”}\}$

$$\Rightarrow (DE(D^a, D^b) = F)$$

Let $D^a = \{Cookie : \text{“Enabled”}\}$ and

$D^b = \{Cookie : \text{“Enabled”}\}$

$$\Rightarrow (DE(D^a, D^b) = T)$$

4.28 FIELD REGISTER DICTIONARY KEY INTERSECTION IDENTIFICATION OPERATION

The dictionary key intersection identification operation, ‘DKII’, is an iterative operation that runs the basic `AlphanumericString` equality operation on keys of any two given dictionaries. Since individually, keys are `AlphanumericStrings`, we can use the ‘ASE’ function iteratively on each key to check if there exists the same key(s) across two non-equal dictionaries. If two any key `AlphanumericStrings` contain the same content, the operation returns a true flag, ‘T’. If not, the operation returns a false flag ‘F’. Definition 21 presents the formal notation of HESTIA’s dictionary key intersection identification operation.

Definition 21 *Dictionary Key Intersection Identification operation*

Given two Dictionaries, D^a and D^b :

$$(DKII(D^a, D^b) = T) \Leftrightarrow (\forall (((k : v)^a \in D^a) \wedge ((k : v)^b \in D^b)), (\exists ((ASE(k^a, k^b) = T))))$$

$$\textit{Otherwise, } (DKII(D^a, D^b) = F)$$

Examples: The following are examples for the dictionary key intersection identification operation, ‘DKII’:

Let $D^a = \{Cookie : \text{“Enabled”}, JavaScript : \text{“Enabled”}\}$ and

$D^b = \{Cookie : \text{“Enabled”}, JavaScript : \text{“Disabled”}\}$

$$\Rightarrow (DKII(D^a, D^b) = T)$$

Let $D^a = \{Cookie : \text{“Enabled”}\}$ and

$D^b = \{messageEncrypt : \text{“Enabled”}\}$

$$\Rightarrow (DKII(D^a, D^b) = F)$$

4.29 FIELD REGISTER NAMED SET EQUALITY OPERATION

The named set equality operation, ‘NSE’, is an iterative operation that runs the basic `AlphanumericString` equality operation on values of any two given named sets. Since a named set is a collection of `AlphanumericStrings`, we can use the ‘ASE’ function iteratively on each named set value to check if two named sets are equal. If two values of a named set contain the same `AlphanumericString` content, the operation checks iteratively for the next named set value’s content until the two given named sets are exhausted. The operation returns the true flag, ‘T’, if and only if all the values of the given two named sets are equal. If not equal, the operation returns a false flag ‘F’. Definition 22 presents the formal notation of HESTIA’s named set equality operation.

Definition 22 *Named Set Equality operation*

Given two Named Sets, L^a and L^b :

$$(NSE(L^a, L^b) = T) \Leftrightarrow (SC(L^a, L^b) = E) \wedge (\forall ((v^a \in L^a) \wedge (v^b \in L^b)), (\exists! (ASE(v^a, v^b) = F)))$$

$$\textit{Otherwise, } (NSE(L^a, L^b) = F)$$

Examples: The following are examples for the named set equality operation, ‘NSE’:

Let $L^a = [Cookies, JavaScript, messageEncrypt]$ and

$L^b = [Cookies, JavaScript, messageDecrypt]$

$$\Rightarrow (NSE(L^a, L^b) = F)$$

Let $L^a = [Cookies, JavaScript]$ and

$L^b = [Cookies, JavaScript]$

$$\Rightarrow (NSE(L^a, L^b) = T)$$

4.30 FIELD REGISTER NAMED SET INTERSECTION IDENTIFICATION OPERATION

The named set intersection identification operation, ‘NSII’, is an iterative operation that runs the basic `AlphanumericString` equality operation on values of any two given named sets. Since individually,

named set values are `AlphanumericStrings`, we can use the ‘ASE’ function iteratively on each value of the named set to check if there exists any common values(s) across two non-equal named sets. If any two value `AlphanumericStrings` of a named set contain the same content, the operation returns a true flag, ‘T’. If not, the operation returns a false flag ‘F’. Definition 23 presents the formal notation of HESTIA’s named set intersection identification operation.

Definition 23 *Named Set Intersection Identification operation*

Given two Named Sets, L^a and L^b :

$$(NSII(L^a, L^b) = T) \Leftrightarrow (\forall ((v^a \in L^a) \wedge (v^b \in L^b)), (\exists (ASE(v^a, v^b) = T)))$$

∨

$$(NSII(L^a, L^b) = F) \Leftrightarrow (\forall ((v^a \in L^a) \wedge (v^b \in L^b)), (\exists! (ASE(v^a, v^b) = T)))$$

Examples: The following are examples for the named set intersection identification operation, ‘NSII’:

Let $L^a = [Cookies, JavaScript, messageEncrypt]$ and

$L^b = [Cookies, JavaScript, messageDecrypt]$

$$\Rightarrow (NSII(L^a, L^b) = T)$$

Let $L^a = [messageEncrypt, JavaScript]$ and

$L^b = [Cookies, messageDecrypt]$

$$\Rightarrow (NSII(L^a, L^b) = F)$$

4.31 FIELD REGISTER COMPARISON OPERATION

The field register comparison operation, ‘FRC’, is a register content comparison operation. For two given registers of the same type, the operation checks if the register contents are equal. If the two registers contain the same content, the operation returns a true flag ‘T’. If the two registers do not contain the same content, the check returns a false value ‘F’. Definition 24 presents the formal notation of HESTIA’s field register comparison operation.

Definition 24 *Field Register Comparison operation*

Given two field registers, R^a and R^b :

$$\wedge (SC(R^a, R^b) = E)$$

$$\wedge (FRTC(R^a, R^b) = T) :$$

$$(FRC(R^a.R^b) = T) \Leftrightarrow$$

$$((ASE(R^a, R^b) = T) \text{ when } (FRTD(R^a) = U)) \vee$$

$$(FRC(R^a.R^b) = T) \Leftrightarrow$$

$$((DSE(R^a, R^b) = T) \text{ when } (FRTD(R^a) = G)) \vee$$

$$(FRC(R^a.R^b) = T) \Leftrightarrow$$

$$((DE(R^a, R^b) = T) \text{ when } (FRTD(R^a) = C)) \vee$$

$$(FRC(R^a.R^b) = T) \Leftrightarrow$$

$$((NSE(R^a, R^b) = T) \text{ when } (FRTD(R^a) = I))$$

$$\text{Otherwise } FRC(R^a, R^b) = F$$

Example (Values): The following are examples for the field register comparison operation, ‘FRC’, in case of the field registers being values:

$$\text{Let } R^a = 9 \text{ and}$$

$$R^b = 9$$

$$\Rightarrow (FRC(R^a, R^b) = T)$$

$$\text{Let } R^a = 9 \text{ and}$$

$$R^b = 99$$

$$\Rightarrow (FRC(R^a, R^b) = F)$$

Example (Dotted Strings): The following are examples for the field register comparison operation, ‘FRC’, in case of the field registers being dotted strings:

$$\text{Let } R^a = \text{‘RTU.onsite.001’ and}$$

$$R^b = \text{'RTU.onsite.001'}$$

$$\Rightarrow (\text{FRC}(R^a, R^b) = \text{T})$$

Let $R^a = \text{'RTU.onsite.001'}$ and

$$R^b = \text{'RTU.onsite.002'}$$

$$\Rightarrow (\text{FRC}(R^a, R^b) = \text{F})$$

Example (Dictionaries): The following are examples for the field register comparison operation, 'FRC', in case of the field registers being dictionaries:

Let $R^a = \{\text{Cookie} : \text{"Enabled"}, \text{JavaScript} : \text{"Enabled"}\}$ and

$$R^b = \{\text{Cookie} : \text{"Enabled"}, \text{JavaScript} : \text{"Enabled"}\}$$

$$\Rightarrow (\text{FRC}(R^a, R^b) = \text{T})$$

Let $R^a = \{\text{Cookie} : \text{"Enabled"}, \text{JavaScript} : \text{"Enabled"}\}$ and

$$R^b = \{\text{Cookie} : \text{"Enabled"}, \text{JavaScript} : \text{"Disabled"}\}$$

$$\Rightarrow (\text{FRC}(R^a, R^b) = \text{F})$$

Example (Named Sets): The following are examples for the field register comparison operation, 'FRC', in case of the field registers being named sets:

Let $R^a = [\text{Cookies}, \text{JavaScript}, \text{messageEncrypt}]$ and

$$R^b = [\text{Cookies}, \text{JavaScript}, \text{messageEncrypt}]$$

$$\Rightarrow (\text{FRC}(R^a, R^b) = \text{T})$$

Let $R^a = [\text{Cookies}, \text{JavaScript}]$ and

$$R^b = [\text{JavaScript}, \text{messageEncrypt}]$$

$$\Rightarrow (\text{FRC}(R^a, R^b) = \text{F})$$

4.32 FIELD REGISTER RELAXED COMPARISON OPERATION

The field register relaxed comparison operation, ‘FRRC’, is a register content comparison operation. For two given registers of the same type, the operation checks if the register contents are equal. If the two registers contain common content, the operation returns a true flag ‘T’, even if they are not completely equal. If the two registers do not contain any common content, the operation returns a false value ‘F’. Definition 25 presents the formal notation of HESTIA’s field register relaxed comparison operation.

Definition 25 *Field Register Relaxed Comparison operation*

(Given two field registers, R^a and R^b :

$$\wedge (FRTC(R^a, R^b) = T) :$$

$$(FRRC(R^a.R^b) = T) \Leftrightarrow$$

$$((ASE(R^a, R^b) = T) \text{ where } (FRTD(R^a) = U)) \vee$$

$$(FRRC(R^a.R^b) = T) \Leftrightarrow$$

$$((DSE(R^a, R^b) = T) \text{ where } (FRTD(R^a) = G)) \vee$$

$$(FRRC(R^a.R^b) = T) \Leftrightarrow$$

$$((DSII(R^a, R^b) = T) \text{ where } (FRTD(R^a) = G)) \vee$$

$$(FRRC(R^a.R^b) = T) \Leftrightarrow$$

$$((DE(R^a, R^b) = T) \text{ where } (FRTD(R^a) = C)) \vee$$

$$(FRRC(R^a.R^b) = T) \Leftrightarrow$$

$$((DKII(R^a, R^b) = T) \text{ where } (FRTD(R^a) = C)) \vee$$

$$(FRRC(R^a.R^b) = T) \Leftrightarrow$$

$$((NSE(R^a, R^b) = T) \text{ where } (FRTD(R^a) = I)) \vee$$

$$(FRRC(R^a.R^b) = T) \Leftrightarrow$$

$$((NSII(R^a, R^b) = T) \text{ where } (FRTD(R^a) = I)) \vee$$

$$\text{Otherwise } \text{FRRC}(R^a, R^b) = F$$

Example (Values): The following are examples for the field register relaxed comparison operation, ‘FRRC’, in case of the field registers being values:

$$\text{Let } R^a = 9 \text{ and}$$

$$R^b = 9$$

$$\Rightarrow (\text{FRRC}(R^a, R^b) = T)$$

$$\text{Let } R^a = 9 \text{ and}$$

$$R^b = 99$$

$$\Rightarrow (\text{FRRC}(R^a, R^b) = F)$$

Example (Dotted Strings): The following are examples for the field register relaxed comparison operation, ‘FRRC’, in case of the field registers being dotted strings:

$$\text{Let } R^a = \text{'RTU.onsite.001'} \text{ and}$$

$$R^b = \text{'RTU.parent'}$$

$$\Rightarrow (\text{FRRC}(R^a, R^b) = T)$$

$$\text{Let } R^a = \text{'RTU.onsite.001'} \text{ and}$$

$$R^b = \text{'PMU.parent'}$$

$$\Rightarrow (\text{FRRC}(R^a, R^b) = F)$$

Example (Dictionaries): The following are examples for the field register relaxed comparison operation, ‘FRRC’, in case of the field registers being dictionaries:

$$\text{Let } R^a = \{\text{Cookie} : \text{"Enabled"}\} \text{ and}$$

$$R^b = \{\text{Cookie} : \text{"Disabled"}, \text{JavaScript} : \text{"Enabled"}\}$$

$$\Rightarrow (\text{FRRC}(R^a, R^b) = T)$$

$$\text{Let } R^a = \{\text{Cookie} : \text{"Enabled"}, \text{JavaScript} : \text{"Enabled"}\} \text{ and}$$

$$R^b = \{messageEncrypt : \text{“Enabled”}, geoTransmit : \text{“Disabled”}\}$$

$$\Rightarrow (FRRC(R^a, R^b) = F)$$

Example (Named Sets): The following are examples for the field register relaxed comparison operation, ‘FRRC’, in case of the field registers being named sets:

$$\text{Let } R^a = [Cookies, JavaScript, messageEncrypt] \text{ and}$$

$$R^b = [Cookies, JavaScript, messageDecrypt]$$

$$\Rightarrow (FRRC(R^a, R^b) = T)$$

$$\text{Let } R^a = [Cookies, JavaScript] \text{ and}$$

$$R^b = [messageDecrypt, messageEncrypt]$$

$$\Rightarrow (FRRC(R^a, R^b) = F)$$

4.33 FIELD REGISTER APPLICATION OPERATION

The field register application operation, ‘FRA’, is a register content application operation. For two given registers of the same type, the operation applies the content of the first register onto the second register. The operation overwrites the contents of the second register. Definition 26 presents the formal notation of HESTIA’s field register application operation.

Definition 26 *Field Register Application operation*

Given two field registers, R^a and R^b

$$\wedge (FRTC(R^a, R^b) = T) :$$

$$(FRA(R^a, R^b)) \stackrel{=}{def} (R^{b_p}), \text{ where :}$$

$$(R^{b_p} = R^a)$$

Example (Values): The following is an example for the field register application operation, ‘FRA’, in case of the field registers being values:

$$\text{Let } R^a = 9 \text{ and}$$

$$\begin{aligned}
R^b &= 99 \\
(\text{FRA}(R^a, R^b)) \\
\Rightarrow R^a &= 9 \text{ and} \\
\Rightarrow R^b &= 9 \text{ and} \\
\Rightarrow R^{b^p} &= 99
\end{aligned}$$

Example (Dotted Strings): The following is an example for the field register application operation, ‘FRA’, in case of the field registers being dotted strings:

$$\begin{aligned}
\text{Let } R^a &= \text{‘RTU.onsite.001’ and} \\
R^b &= \text{‘RTU.onsite.002’} \\
(\text{FRA}(R^a, R^b)) \\
\Rightarrow R^a &= \text{‘RTU.onsite.001’ and} \\
\Rightarrow R^b &= \text{‘RTU.onsite.001’ and} \\
\Rightarrow R^{b^p} &= \text{‘RTU.onsite.002’}
\end{aligned}$$

Example (Dictionaries): The following is an example for the field register application operation, ‘FRA’, in case of the field registers being dictionaries:

$$\begin{aligned}
\text{Let } R^a &= \{\text{Cookie : “Enabled”}\} \text{ and} \\
R^b &= \{\text{Cookie : “Disabled”, JavaScript : “Enabled”}\} \\
(\text{FRA}(R^a, R^b)) \\
\Rightarrow R^a &= \{\text{Cookie : “Enabled”}\} \text{ and} \\
\Rightarrow R^b &= \{\text{Cookie : “Enabled”, JavaScript : “Enabled”}\} \text{ and} \\
\Rightarrow R^{b^p} &= \{\text{Cookie : “Disabled”, JavaScript : “Enabled”}\}
\end{aligned}$$

Example (Named Sets): The following is an example for the field register application operation, ‘FRA’, in case of the field registers being named sets:

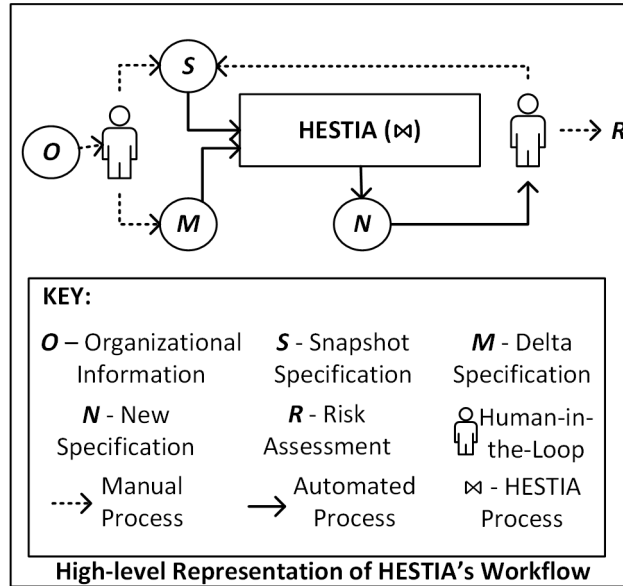


Figure 4.1: High-level representation of HESTIA's workflow.

Let $R^a = [Cookies, JavaScript, messageEncrypt]$ and

$R^b = [Cookies, JavaScript, messageDecrypt]$

$(FRA(R^a, R^b))$

$\Rightarrow R^a = [Cookies, JavaScript, messageEncrypt]$ and

$\Rightarrow R^b = [Cookies, JavaScript, messageDecrypt]$ and

$\Rightarrow R^{b_p} = [Cookies, JavaScript, messageDecrypt]$

4.34 THE SPECIFICATION- AND ADVERSARY-BASED RISK ASSESSMENT PROCESS: HESTIA

HESTIA is a specifications- and adversary-based risk assessment process. The core tenets of HESTIA are: 1) all aspects of an organizational environment, e.g. platforms, potential attack vectors, defenses, etc., are converted into specifications; 2) a human then classifies specifications into two categories: snapshot or delta; 3) relationships between the specifications, both across and within categories, drive the risk assessment formulation; and 4) use of specifications enables HESTIA to be utilized iteratively.

As shown in Figure 4.1, HESTIA's workflow can be described as follows: By processing an organization's CPS system information, \mathcal{O} , a human-in-the-loop produces a Snapshot Specification, S , and a Delta

Specification, M . Every Specification has a corresponding Template, T , which is written by \mathcal{H} . HESTIA, denoted by \bowtie symbol, takes S and M and processes them, to produce a New Specification (N). The definitions of S , D , and N were provided in a previous chapter [2]. HESTIA process, \bowtie , involves three checking and one application processes. These processes are: consistency check (\propto), conflicts check (\otimes), applicability check ($\overset{?}{\vdash}$), and delta application (\vdash). The human-in-the-loop performs one of either actions: sends N to HESTIA, for one more iteration with a new Delta Specification OR uses N to produce a risk assessment document, R . A formal representation of HESTIA's workflow can be found in Definition 27.

Definition 27 *HESTIA's workflow at a high-level*

$$\begin{aligned}
& \mathcal{H}(\mathcal{O}) \xrightarrow{pro} (S \wedge M) \\
& (S \wedge M) \in \mathcal{S} \\
& \forall (S \wedge M), \mathcal{H} \xrightarrow{pro} T \\
& \bowtie (S, M) \xrightarrow{pro} N \quad \dots(1) \\
& \bowtie (S, M) \stackrel{=}{def} ((\propto (S) \wedge \propto (M)) \\
& \wedge (\otimes(S, M) \vee (\overset{?}{\vdash} (S, M) \wedge (M \vdash S)))) \\
& \mathcal{H}(N) \rightarrow (1) \quad \vee \\
& \mathcal{H}(N) \xrightarrow{pro} R
\end{aligned}$$

Where: \mathcal{O} is an organization's CPS system information, S is Snapshot Specification. M is Delta Specification. N is New Specification. \mathcal{H} is Human-in-the-loop. R is the risk assessment document. The symbol \bowtie represents HESTIA process, the symbol \xrightarrow{pro} represents the term 'produces', and the symbol \rightarrow represents the term 'sends to'. Although there exist several relationships between specifications, we designed HESTIA to be capable of processing three relationships: consistency, conflicts, and applicability.

4.35 CHECKING CONSISTENCY

HESTIA ensures consistency, \propto , between specifications via template equality. By processing an organization's CPS system information, \mathcal{O} , a Human-in-the-loop, \mathcal{H} , produces a System Specification S . For every System Specification, S , \mathcal{H} writes Entities: E^a . Similarly, for every Template, T , \mathcal{H} writes Entities: E^b . For all Entities of the Template, E^b , HESTIA checks to see if same type of Entities can

be found in Entities of System Specification, E^a . For all Entities of the Template, E^b , whose type equates with Entities of System Specification, E^a , HESTIA checks to see if the names of Fields within the Entities are equal or not. The System Specification S is said to be consistent if and only if the Field names are equal, for all Fields of all Entities of the Template, T , whose type equates with Entities of System Specification, S . HESTIA's consistency processing has been formally represented in Definitions 28 and 29.

Definition 28 *HESTIA's consistency checking process for entities*

$$\begin{aligned}
 & \text{Given two entities, } E^a \text{ and } E^b, \\
 & ((\{E_s^1, E_s^2, \dots, E_s^a\} \in S) \wedge (\{E_t^1, E_t^2, \dots, E_t^b\} \in T) \wedge \\
 & (F_s^x \in \{F_a^1, F_a^2, \dots, F_a^a\} \in E_s^a) \wedge (F_t^y \in \{F_t^1, F_t^2, \dots, F_t^b\} \in E_t^b)) : \\
 & ((\alpha(E_s^a, E_t^b) = T) \Leftrightarrow (ETC(E_s^a, E_t^b) = T) \wedge \forall F_t^y, \exists F_s^x : (FNC(F_s^x, F_t^y) = T) \\
 & \text{Otherwise } \alpha(E^a, E^b) = F)
 \end{aligned}$$

Definition 29 *HESTIA's consistency checking process for specifications*

$$\begin{aligned}
 & \text{Given two specifications, } S \text{ and } T, \\
 & ((\{E_s^1, E_s^2, \dots, E_s^a\} \in S) \wedge (\{E_t^1, E_t^2, \dots, E_t^b\} \in T)) : \\
 & ((\alpha(S, T) = F) \Leftrightarrow \forall ((E_s^x, E_t^y) \in \{S \times T\}) : \exists ((ETC(E_s^x, E_t^y) = T) \wedge (\alpha(E_s^x, E_t^y) = F)) \\
 & \text{Otherwise } \alpha(S, T) = T)
 \end{aligned}$$

Listings 4.1, 4.2, and 4.3 show a System Specification, S , a Delta Specification, M , and a Template, T ; all written using HERMES. After performing the consistency check, HESTIA declares that the System and Delta Specifications, S and M are consistent with the Template T . HESTIA declares so because all Field names (`geoTransmit` & `messageEncrypt`) of T are available as Field names in S and M , for the Entities of the same Type, *RTUConfig*.

4.36 CHECKING CONFLICTS

HESTIA checks for conflicts, \otimes , between consistent specifications by comparing field registers. By processing an organization's CPS system information, \mathcal{O} , a Human-in-the-loop, \mathcal{H} , produces a consistent System Specification S , and a consistent Delta Specification M . For every consistent System Specification, S , \mathcal{H} writes Entities: E^a . Also for every consistent Delta Specification, M , \mathcal{H} writes Entities: E^b . For all Entities of the consistent System Specification, E^a , HESTIA checks to see if same type of Entities can be found in Entities of the consistent Delta Specification, E^b . For all Entities of the System Specification, E^a , whose type equates with the Entities of Delta Specification, E^b , HESTIA checks to see if IDs of the Entities are equal or not.

For all Entities of the consistent System Specification, E^a , whose type and ID are equal with Entities of the consistent Delta Specification, E^b , HESTIA checks to see if names of Fields within the Entities equate or not. For all Entities of the consistent System Specification, E^a , whose type, ID equate with Entities of the consistent Delta Specification, E^b , HESTIA checks to see if registers of the Fields within the Entities equate or not, for all Fields whose names equate across E^a , and E^b . HESTIA declares that there exists a conflict between consistent System Specification S and consistent Delta Specification M if and only if the Field registers do not equate, for all Fields of equal names within all Entities, whose type and ID equate across S , & M . HESTIA's conflict processing has been formally represented in Definitions 30 and 31.

Definition 30 *HESTIA's conflict checking process for entities*

$$\begin{aligned}
 & \text{Given two entities, } E^a \text{ and } E^b, \\
 & ((\{E_s^1, E_s^2, \dots, E_s^a\} \in S) \wedge (\{E_m^1, E_m^2, \dots, E_m^b\} \in M) \wedge \\
 & (\{F_s^1, F_s^2, \dots, F_s^a\} \in E_s^a) \wedge (\{F_m^1, F_m^2, \dots, F_m^b\} \in E_m^b)) : \\
 & (\otimes(E_s^a, E_m^b) = T) \Leftrightarrow ((ETC(E_s^a, E_m^b) = T) \wedge \\
 & (EIC(E_s^a, E_m^b) = T) \wedge \\
 & \forall ((F_s^x, F_m^y) \in \{\{F_s^1, F_s^2, \dots, F_s^a\} \times \{F_m^1, F_m^2, \dots, F_m^b\}\}) : \exists ((FNC(F_s^x, F_m^y) = T) \wedge \\
 & (FRC(F_s^x, F_m^y) = F))) \\
 & \text{Otherwise } \otimes(E_s^a, E_m^b) = F
 \end{aligned}$$

Definition 31 *HESTIA's conflict checking process for specifications*

Given two specifications, S and M :

$$((\{E_s^1, E_s^2, \dots, E_s^a\} \in S) \wedge (\{E_m^1, E_m^2, \dots, E_m^b\} \in M))$$

$$(\otimes(S, M) = T) \Leftrightarrow \forall ((E_s^x, E_m^y) \in \{S \times M\}) : \exists(\otimes(E_s^x, E_m^y) = T)$$

Otherwise $\otimes(S, T) = F$

Listings 4.1, 4.2, and 4.3 show two consistent Specifications, System Specification S & Delta Specification M , all written using HERMES. After performing the conflict check, HESTIA declares that the consistent System Specification S is in a conflict with the consistent Delta Specification M . HESTIA does so because the field register value is different for consistent Specifications S and M ; given equal field name parameter, ‘`geoTransmit`’, for the Entities of the same name and type across S , and M .

4.37 CHECKING APPLICABILITY

HESTIA checks for applicability, $\vdash^?$, between consistent specifications by comparing field registers. By processing an organization’s CPS system information, \mathcal{O} , a Human-in-the-loop, \mathcal{H} , produces a consistent System Specification S and a consistent Delta Specification M . For every consistent System Specification, S , \mathcal{H} writes Entities: E^a . Also for every consistent consistent Delta Specification, M , \mathcal{H} writes Entities: E^b . For all Entities of the consistent Delta Specification, E^b , HESTIA checks to see if the same type of Entities can be found in entities of the consistent System Specification, E^a . For all Entities of the consistent Delta Specification, E^b , whose type equates with the Entities of consistent System Specification, E^a , HESTIA checks to see if IDs of the Entities equate or not.

For all Entities of the consistent Delta Specification, E^b , whose type and ID equate with Entities of the consistent System Specification, E^a , HESTIA checks to see if names of Fields within the Entities equate or not. For all Entities of the consistent Delta Specification, E^b , whose type, ID equate with Entities of the consistent System Specification, E^a , HESTIA checks to see if registers of the Fields within the Entities equate or not, for all Fields whose names equate across E^a and E^b . HESTIA declares that the consistent Delta Specification, M , is applicable on the consistent System Specification, S , if and only if the Field registers do not equate, for all Fields of equal names within all Entities, whose type and ID equate across M and S . HESTIA’s applicability processing has been formally represented in Definitions 32 and 33.

Definition 32 *HESTIA's applicability checking process for entities*

$$\begin{aligned}
& \text{Given two entities, } E^a \text{ and } E^b : \\
& ((\{E_s^1, E_s^2, \dots, E_s^a\} \in S) \wedge (\{E_m^1, E_m^2, \dots, E_m^b\} \in M) \wedge \\
& (\{F_s^1, F_s^2, \dots, F_s^a\} \in E_s^a) \wedge (\{F_m^1, F_m^2, \dots, F_m^b\} \in E_m^b)) \\
& (\vdash^? (E_s^a, E_m^b) = T) \Leftrightarrow ((ETC(E_s^a, E_m^b) = T) \wedge \\
& (EIC(E_s^a, E_m^b) = T) \wedge \\
& \forall ((F_s^x, F_m^y) \in \{\{F_s^1, F_s^2, \dots, F_s^a\} \times \{F_m^1, F_m^2, \dots, F_m^b\}\}) : \exists ((FNC(F_s^x, F_m^y) = T) \wedge \\
& (FRC(F_s^x, F_m^y) = F))) \\
& \text{Otherwise } \vdash^? (E_s^a, E_m^b) = F
\end{aligned}$$

Definition 33 *HESTIA's applicability checking process for specifications*

$$\begin{aligned}
& \text{Given two specifications, } S \text{ and } M : \\
& ((\{E_s^1, E_s^2, \dots, E_s^a\} \in S) \wedge (\{E_m^1, E_m^2, \dots, E_m^b\} \in M) \wedge \\
& (\vdash^? (S, M) = T) \Leftrightarrow \forall ((E_s^x, E_m^y) \in \{S \times M\}) : \exists (\vdash^? (E_s^x, E_m^y) = T) \\
& \text{Otherwise } \vdash^? (S, T) = F
\end{aligned}$$

Listings 4.1, 4.2, and 4.3 shows two consistent Specifications, System Specification S , and Delta Specification M , both written using HERMES. After performing the applicability check, HESTIA declares that the consistent Delta Specification M is applicable to the consistent System Specification S . HESTIA does so because the field register value is different for Specifications M and S ; given equal field name parameter, 'geoTransmit', for the Entities of the same name and type across T , M and S .

4.38 DELTA APPLICATION

When a Delta Specification, M , is applicable towards a System Specification, S , a human-in-the-loop, \mathcal{H} may want to apply M towards S . This process is called Delta Application, \vdash . Through the Delta Application action, we can apply one field register content towards another register content using the field register application (FRA) operation. A formal definition of the FRA operation has been given in Definition 26. HESTIA's Delta Application action, \vdash , has been formally represented in Definitions 34 and 35.

Definition 34 *HESTIA's delta application process for entities*

$$\begin{aligned}
& \text{Given two entities, } E^a \text{ and } E^b : \\
& ((\{E_s^1, E_s^2, \dots, E_s^a\} \in S) \wedge (\{E_m^1, E_m^2, \dots, E_m^b\} \in M) \wedge \\
& (\{F_s^1, F_s^2, \dots, F_s^a\} \in E_s^a) \wedge (\{F_m^1, F_m^2, \dots, F_m^b\} \in E_m^b)) \\
& (\vdash (E_s^a, E_s^b) = E_n^c), \text{ where} \\
& (E_n^c \stackrel{=}{\text{def}} E_m^b) \iff (\forall ((F_s^x, F_m^y) \in \{\{F_s^1, F_s^2, \dots, F_s^a\} \times \{F_m^1, F_m^2, \dots, F_m^b\}\}) : (\vdash^? (E_s^a, E_m^b) = F)) \\
& \quad \vee \\
& (E_n^c \stackrel{=}{\text{def}} \{F_n^1, F_n^2, \dots, F_n^c\}) \iff (\forall ((F_s^x, F_m^y) : (\vdash^? (E_s^a, E_m^b) = T)), \text{ where} \\
& \quad (\forall F_n^z \in \{F_n^1, F_n^2, \dots, F_n^c\}) : \\
& \quad (F_n^z \stackrel{=}{\text{def}} F_m^y) \iff ((FNC(F_s^x, F_m^y) = F) \vee \\
& \quad (F_n^z \stackrel{=}{\text{def}} FRA(F_s^x, F_m^y)) \iff ((FNC(F_s^x, F_m^y) = T) \wedge (FRC(F_s^x, F_m^y) = F))
\end{aligned}$$

Definition 35 *HESTIA's delta application process for specifications*

$$\begin{aligned}
& \text{Given two specifications, } S \text{ and } M : \\
& ((\{E_s^1, E_s^2, \dots, E_s^a\} \in S) \wedge (\{E_m^1, E_m^2, \dots, E_m^b\} \in M)) \\
& (\vdash (S, M) = N) \\
& N \stackrel{=}{\text{def}} \{E_n^1, E_n^2, \dots, E_n^c\}, \text{ where} \\
& (\forall E_n^z \in \{E_n^1, E_n^2, \dots, E_n^c\}) \wedge (\forall ((E_s^x, E_m^y) \in \{S \times M\})) : \\
& \quad (E_n^z \stackrel{=}{\text{def}} (\vdash (E_s^x, E_m^y)))
\end{aligned}$$

Listings 4.1 and 4.2 shows two Specifications, System Specification S and Delta Specification M , written using HERMES. As discussed in Section 4.37, HESTIA declares that Delta Specification, M , is applicable to the System Specification, S . The human-in-the-loop, \mathcal{H} , would like to apply M to S and invokes the Delta Application action, \vdash . After the \vdash action is executed, the System Specification, S , is changed into a New Specification, N . Listing 4.4 presents the New Specification using the HERMES language. One can observe the difference between the old System Specification S and the New System Specification N by comparing Listings 4.1 and 4.4.

Listing 4.4: HERMES excerpt of the New Specification.

```

1  %New Specification N
2  RTUDevices.001: RTUConfig
3  {
4      messageEncrypt: YES;
5      geoTransmit: NO;
6      geoTransmit.Old: YES;
7      location: {City: ‘Seattle’, State: ‘Washington’}
8      commAccess: [‘RTUDevices.001’]
9  }.
10 .
11 .
12 .
13 RTUDevices.002: RTUConfig
14 {
15     messageEncrypt: YES;
16     geoTransmit: NO;
17     geoTransmit.Old: YES;
18     location: {City: ‘Pullman’, State: ‘Washington’}
19     commAccess: [‘RTUDevices.001’, ‘RTUDevices.002’]
20 }.
21

```

4.39 CHAPTER CONCLUSION

Cyber-attacks against critical infrastructure have the potency to cause tremendous damage, suffering, and loss. One approach to help protect organizations from cyber-attacks is an automated, adversary-based risk assessment of critical infrastructure. Utilization of specifications can be one mechanism to automate adversarial risk assessment. Factors lending a hand in the non-existence of a specification-based, adversarial risk assessment process have been presented. A previously presented Specification- and Adversary-based Risk Assessment process, HESTIA, was briefly discussed. We presented the formal model of HESTIA process, describing the three relationship-processing operations, which are consistency-, conflict-, and applicability-checking. We discussed limitations of the current HESTIA process and avenues to expand the process. We hope that HESTIA’s formal model will help organizations build their own implementations of the risk assessment system to defend their critical infrastructure.

FORMAL NOTATION NOMENCLATURE

AlphanumericString - A string containing English alphabets and/or numbers

ASE - **AlphanumericString** Equality operation

Dotted String - A chain of **AlphanumericStrings** that are separated by dots.

DE - Dictionary Equality operation

DKII - Dictionary Key Intersection Identification operation

DSE - Dotted String Equality operation
 DSII - Dotted String Intersection Identification operation
 EIC - Entity ID Comparison operation
 ETC - Entity Type Comparison operation
 FRA - Field Register Application operation
 FNC - Field Name Comparison operation
 FRC - Field Register Comparison operation
 FRRC - Field Register Relaxed Comparison operation
 FRTD - Field Register Type Determination operation
 FRTC - Field Register Type Comparison operation
 EID - Entity Identifier
 ET - Entity Type
 FN - Field Name
Named Set - An unordered list containing `AlphanumericString` values
 NSE - Named Set Equality operation
 NSII - Named Set Intersection Identification operation
 SC - Size Comparison operation
 SD - Size Determination operation
 \mathcal{S} - Set of all valid specifications
 \mathcal{T} - Set of all valid templates
 \mathcal{E} - Set of all valid entities
 \mathcal{F} - Set of all valid fields
 \mathcal{V} - Set of all valid `AlphanumericString` values
 \mathcal{Q} - Set of all valid dotted strings
 \mathcal{D} - Set of all valid dictionaries
 \mathcal{L} - Set of all valid named sets
 \propto - Consistency Checking operator
 \otimes - Conflict Checking operator
 $\overset{?}{\vdash}$ - Applicability Checking operator
 \vdash - Delta Application operator

CHAPTER 5: A HOLISTIC CYBER PHYSICAL SYSTEM

CASE STUDY

5.1 CHAPTER INTRODUCTION

The need and importance of securing Cyber Physical Systems (CPS) has been previously documented [21], [20]. Conducting research into CPS security and validating the results requires accurate and complete models from CPS organizations and their systems. For some types of security research, obtaining information about the cyber part of a CPS organization might be sufficient and for others, physical aspects of a CPS organization would be adequate. However, there exist security research projects which require test models containing accessible, detailed, realistic, coherent, and holistic CPS organizational data [1], [2], [35]. In this Chapter, we will use the term “holistic” to refer to all the above mentioned qualities. Such holistic models need to span across the physical, control, and cyber domains of a CPS organization.

5.2 CHAPTER PROBLEM

The unavailability of holistic CPS organizational models is the problem. By consulting with subject matter experts (SMEs), we were able to identify three factors that are contributing to the persistence of the problem at hand. The factors are: 1) preparing detailed and holistic organizational models takes time and resources; 2) most CPS organizations usually do not invest time and resources in preparing such data; and 3) CPS organizations which do have detailed and holistic data are unwilling to share that information with third party researchers.

The current unavailability of holistic CPS models hinders adequate validation of CPS security research. Designing and creating holistic CPS models will help researchers to: 1) validate their research results; 2) compare similar research projects via a common medium; and 3) test different solutions for practical applicability. In addition, holistic CPS models may also be used as an instrument as an instructional tool.

5.3 CHAPTER CONTRIBUTION

We describe a holistic CPS model for a fictional power utility, Acme Corporation. This model is based on the IEEE 14-Bus System bus structure [36]. To design Acme Corp.’s organizational and system model we reviewed current relevant literature and consulted subject matter experts. More information

on the resources and the construction process can be found in Section 5.5. As seen in Figure 5.1, Acme Corp. is designed to have one enterprise center, one operations center, fourteen substations, one utility owned power generator, four third party power generators, ten power loads, and one critical load. The Acme Corp. model is a part of a model collection named METICS and is freely available at: <https://github.com/METICS-CPS>. METICS stands for Models for ExTensIble Cyber-physical system Security, where extensible means scalable. Interested persons are encouraged to use, share, and improve upon these models.

This contribution is part of a comprehensive CPS security research program whose goals are: 1) to develop cutting edge cybersecurity and risk management tools for CPS; 2) to perform big-data, machine learning, and visualization research for CPS cybersecurity; 3) to design and test security controls for high voltage direct current (HVDC) systems; 4) to design and implement a distributed CPS testbed; 5) to analyze, implement, and enforce least-privilege CPS designs and configurations; and 6) to analyze and design secure energy storage, transmission, and distribution systems.

5.4 CHAPTER OUTLINE

The rest of this Chapter is organized as follows: Section 5.5 discusses the Acme Corporation design process and presents the high-level organizational layout of Acme Corp. Sections 5.6, 5.7, 5.8, and 5.9 present the details about each of the Acme Corp. organizational structures, which are: substations, substation controls, operations center, and enterprise center, respectively. Section 5.10 presents examples of CPS security research outcomes that may be validated using the model introduced in this Chapter. Section 5.11 discusses related works and the differences between these and our contribution. A conclusion to the chapter is provided at the end.

5.5 THE FICTIONAL-YET-REALISTIC POWER UTILITY: ACME CORPORATION

To fully enable the description of a holistic CPS model we created Acme Corp. which is a fictional power utility. We then developed a holistic model of the organization and systems for the Acme Corp. utility. The inspiration for Acme Corp.'s model was derived from the following sources: 1) power flow and bus structure is derived from IEEE 14-bus system [36]; 2) the Supervisory control and Data Acquisition (SCADA) aspects are derived from Mahan et al.'s guide for SCADA systems [24]; 3) information about network connections, both internal and external, has been derived from Stouffer et al.'s guide to Industrial Control Systems (ICS) security [8]; and 4) all other aspects are derived from consulting with Subject

Matter Experts (SMEs). SMEs have also been used to verify and validate the accuracy and realistic nature of the Acme Corp. model.

Acme Corp.’s high-level design is illustrated in Figure 5.1. Acme Corp. is designed to have the following components: 1) one enterprise center, represented with the label ‘EC-1’ in Figure 5.1; 2) one operations center, represented with the label ‘OC-1’ in Figure 5.1; 3) fourteen substations, represented with the labels ‘SS-#’ in Figure 5.1; 4) one utility-owned power generator, represented with the label ‘G’ in Figure 5.1; 5) four third-party power generators, represented with the label ‘TG’ in Figure 5.1; 6) ten power loads, represented with the label ‘L’ in Figure 5.1; and 7) one critical power load, represented with the label ‘CL’ in Figure 5.1.

All of the substations have a substation control center. These substation controls are either connected to other substations or connected to the operations center. Power transmission and communication between substations occurs via buses and network cables, respectively. The Operations center has wired network connectivity to all substation controls either directly or indirectly. Some substation controls have wireless connectivity to the operations center. Some loads also have wireless capability to send their meter readings to substation controls. The Enterprise center has direct access to the Internet. Remote engineering access to the operations center and the enterprise center is also available. Authentication of keys for remote access occurs in the enterprise center. Remote access for vendor support is made available throughout the organization at all levels. Measurement points for SCADA monitoring, relay, and phasor aspects are indicated using solid circular symbols in Figure 5.1. All communication occurs via either power line carrier or fiber optic cable networking, unless otherwise specified. The protocol used for inter-substation communication is the IEEE C37.118.2.¹ DNP3 (Distributed Network Protocol) is used for upstream communication between substations and the operations center.

5.6 ACME CORP.’S SUBSTATIONS

Acme Corp. substations’ high-level design is illustrated in Figure 5.2. Acme Corp. substations’ components are divided into two blocks by design, which are: substation instrumentation and substation control. All of the devices in a substation, across both substation and substation control, are categorized into four zones, which are: SCADA Zone, Relay Zone, Phasor Measurement Zone, and AMI (Advanced Metering Infrastructure) Zone.²

¹Inter-substation communication is not common, apart from the protective-relay communication, which often uses proprietary protocols from the relay vendors. In our Acme Corp. model we decided to use the IEEE C37.118.2 protocol to represent the communication between Phasor Measurement Units (PMUs) and Phasor Data Concentrators (PDCs).

²Not every power utility in the U.S. uses the devices in the Phasor and AMI zones, but their usage is increasing [24]. We decided to include the Phasor and AMI zones in Acme Corp.’s design to be inclusive of the growing number of utilities that use these devices.

Three outside communication channels are available in Acme Corp. These are: market channel, vendor support channel, and Regional Transmission Operators (RTOs) channel. The market channel is used to buy and sell power from the substation. The vendor support channel is used by the device vendors to provide technical support for the devices across all zones. The RTO channel is used to communicate power transmission logistics between the RTO and the substation. Each substation is connected to other substations via electrically via power line carriers and also to the substation control via fiber optic cables. All communication occurs via either power line carrier or fiber optic cable networking, unless otherwise specified. Communication between substation instrumentation and substation control's SCADA zone uses the RS-485 serial protocol. Phasor zone's devices communicate using DNP3 protocol. Communication between Relay zone devices of a substation is carried out using the RS-232 serial protocol. AMI (Advanced Metering Infrastructure) zone devices communicate using TCP/IP and IEEE 802.11 protocols.

The function of a substation block is as follows: electric power is transmitted from a power generator to the substation bus. From the generator bus, power is transmitted across the system via transmission lines to the load buses. On the buses, measurement and relay protection instrumentation devices are installed. Instrumentation transformer's performance/condition readings are sent to a RTU (Remote Terminal Unit) that is present in the substation control's SCADA Zone. Load buses distribute power to the loads and obtain consumption meter readings. The meter readings are increasingly obtained in a wireless manner, that is via a drive-by wireless meter reading. These meter readings are forwarded to the AMI devices in a wireless fashion. AMI devices forward the readings to a repeater via a wireless access point, that is present in the substation control's AMI Zone. Intelligent Electronic Devices (IEDs) and Phasor Measurement Units (PMUs) of the Relay and Phasor zones are connected to the buses at various locations to take measurements for ensuring power transmission safety. These measurements are forwarded to a communications processor and a Phasor Data Concentrator (PDC), respectively, that are present in the substation control's Relay and Phasor Zones.

5.7 ACME CORP.'S SUBSTATION CONTROLS

Acme Corp. substation high-level control design is illustrated in Figure 5.2. Substation control is a block within the substation. Substation control is also commonly called 'Control'. Each Control unit is connected to the substation master via fiber optic cables. Each Control is also connected to the operations center via fiber optic cables, either directly, or indirectly via other substations. All communication occurs via fiber optic cable networking, unless otherwise specified. The protocol used for inter-substation communication is IEEE C37.118.2. DNP3 (Distributed Network Protocol) is used for upstream communication between substations and operations center.

The function of a Control block is as follows: Control receives data from multiple substation instruments. Control processes the data and forwards the processed data to the operations center. Control's SCADA zone consists of a Remote Terminal Unit (RTU), which receives data from the substations SCADA Zone instruments. RTU processes the data and forwards it to the Front End Processor (FEP) in the operations center. Control's communications processor falls under Relay zone, which receives data from IEDs in the substations. The communication processor then sorts the IED readings on basis of the substation's region and forwards the sorted readings to the relay manager in the operations center. Control's PDCs are categorized into the Phasor Zone, which receive PMU readings from substations. PDCs produce a real-time, time-aligned, output data stream, which is sent to Phasor Applications in the operations center. Control's AMI zone consists of repeater devices, which receive meter readings from substation devices and forward those readings to an AMI data aggregator in the operations center. Repeaters transmit the readings via a wireless network mechanism.

5.8 ACME CORP.'S OPERATIONS CENTER

Acme Corp. operations center's high-level design is illustrated in Figure 5.3. The operations center's network is protected by a De-Militarized Zone (DMZ). The operations center is also commonly called 'Operations'. Vendor remote support services and Acme Corp.'s enterprise center are allowed to connect into Operations through a VPN server. The VPN server has an active firewall. Remote engineering access to all of the operation center's devices is provided on an 'as-needed' basis. Authentication of remote engineering access occurs in the enterprise center. Operations is connected to each of Acme Corp.'s Controls via fiber optic cables, either directly, or indirectly via other Controls. All communication occurs via fiber optic cable networking, unless otherwise specified. DNP3 (Distributed Network Protocol) is used for downstream communication between Operations and substations. TCP/IP is the protocol used for upstream communication between operations and enterprise center. Communication within operations uses the TCP/IP protocol stack.

Function of the Operations is as follows: Operations receives data from multiple Control devices. Operations processes the data and forwards the processed data to the enterprise center. Operations' Phasor Zone consists of Phasor applications, which are a series of measures to prevent power transmission's operational security failure. Phasor applications use synchronized phasor measurement data received from the PDCs, which are present in Control. Operations' relay manager is categorized into Relay zone, which receives regionally-sorted IED data from communications processor, which is present in Control. The relay manager provides an interface for operators to see how relay protection is being managed across all substations controls. Operations' AMI zone consists of AMI data aggregators, which receive repeated

meter readings from Controls. AMI data aggregators aggregate the meter readings and forward the aggregated data to SCADA zone. The aggregated AMI data is also forwarded to the enterprise center's engineering consoles and billing department.

Operations' SCADA zone devices consists of: a Front End Processor (FEP), an Historian, SCADA monitoring and management (M&M) suite, and operator consoles. The FEP receives data from RTUs, which are present in Control. FEP processes the data and forwards the data to SCADA M&M suite. The Historian gathers logs & process data from all devices in the Operations. The SCADA M&M suite processes data received from FEP, relay manager, phasor applications, and AMI data aggregators to provide an avenue for operators to supervise and control the control substation devices and data. Operator consoles are the interface through which operators can access SCADA M&M suite. Thereby, Operations has substation-device-level supervisory monitoring and control ability and also has provisions to facilitate an operator to access Control data and change substation device settings.

5.9 ACME CORP.'S ENTERPRISE CENTER

Acme Corp. enterprise center's high-level design is illustrated in Figure 5.4. Enterprise center's network is protected by a De-Militarized Zone (DMZ). Enterprise center is also commonly referred to as 'Enterprise'. Vendor support services, Internet access, engineering remote access, email server services, web server services, and Acme Corp.'s Operations are allowed to connect into enterprise through a VPN server. The VPN server has an active firewall. Remote engineering access to Enterprise's workstations and engineering consoles is provided on an 'as-needed' basis. Authentication of the remote engineering access occurs in the Enterprise via an authentication server. Enterprise is connected to Acme Corp.'s operation center via fiber optic cables and a wireless connectivity mechanism also connects both enterprise and Operations. All communication occurs via fiber optic cable networking, unless otherwise specified. TCP/IP is the protocol used for upstream communication between operations and Enterprise. Communication within operations uses TCP/IP and IEEE 802.11 protocols.

Function of the Enterprise is as follows: enterprise provides Acme Corp.'s engineers with a graphic user interface (GUI) to the CPS infrastructure of Acme Corp. via the workstations and engineering consoles. Acme Corp.'s business servers are also hosted in the Enterprise. The business servers contain financial, executive, and operational data of Acme Corp. Access to the enterprise workstations, business servers, billing department services, and engineering consoles is controlled by the authentication server. Remote access is allowed for a select group of users and it is also controlled by the authentication server. The billing department receives aggregated AMI data from Operations. The billing department processes the data to generate power utility bills. The aggregated AMI data is also provided to engineering

consoles. Acme Corp. Enterprise’s engineering consoles offer a visualized supervisory monitoring and control interface to the SCADA monitoring & management suite, which is present in the Operations. The visualization capabilities include, but are not limited to: 1) visual models of device conditions in Operation, Control, and to a limited extent of the substation instrumentation as well; and 2) visual representation of transmission, distribution, relay protection, and power consumption statistics.

5.10 UTILIZATION OF ACME CORP. MODEL FOR CYBER SECURITY TESTING

A holistic model of a CPS organization, such as the one we present, can be utilized in validating multiple CPS security research projects. The utilization can be achieved by encoding Acme Corp. model in a desired data format. A few example use cases for the Acme Corp. model are presented below.

By encoding Acme Corp. model data in a policy specification format, it can be used to validate policy specifications-based CPS security research projects like the HESTIA project [1]. HESTIA is an adversarial- and specification-based, iterative, and computer-assisted modeling and risk assessment process for CPS organizations. An example snippet specification of Acme Corp.’s enterprise and Operations can be seen in Listing 5.1. The specification language used in Listing 5.1 is called HERMES, which is a high-level, easy-to-use, machine and environment specification language [9].

Listing 5.1: HERMES policy specification snippet for Acme Corp.’s (Figure 5.1) Enterprise.

```

1 Node: EC-1
2 {
3   SubSystems: [EMPLOYEES, WKS, DMZ, ENT-VPN, OC-1];
4 }.
5 Node: EMPLOYEES
6 {
7   EmployeeList: [EID-000];
8 }.
9 Node: WKS
10 {
11   WKSList: [WKS-000];
12 }.
13 Policy: PID_00-000
14 {
15   InternetAccess: [WKS, DMZ];
16 }.
17 Policy: PID_00-001

```

```

18 {
19   LogOnAccess: [EID-000, WKS-000, AUTHSRV];
20 }.
21 Node: ENT-VPN
22 {
23   RemoteAccessList: [EID-000];
24   SubSystems: [AUTHSRV];
25 }.
26 ...

```

5.11 RELATED WORK

One of the motivating factors behind our decision to develop the Acme Corp. model is the lack of CPS organizational models that are holistic in nature. After extensive literature searches of the current state-of-the-art, we were unable to find models that fit our necessity. We list here related, but non-similar works.

Cheng et al. presented a Virtual Power Plant (VPP) that modeled a CPS organization's physical aspects and a brief snippet of the cyber aspects [37]. Vellaithurai et al. presented structures of smart power grids and modeled their smart grid research testbed in a detailed fashion [38]. Mahan et al. discuss cyber aspects of a CPS organization and present a detailed SCADA architecture model for smart power grids [24]. Stouffer et al. present an overall and generic operational structure and communication topologies of an Industrial Control System (ICS) [8]. Our work, Acme Corp., draws inspiration from some of these discussed models. However, in contrast, Acme Corp. models both the physical, control, and cyber domains of a CPS organization in a holistic manner.

5.12 CHAPTER CONCLUSION

The need for securing CPS is well published. Validation of some CPS security research projects require a holistic CPS organizational model, with details spanning across both physical and cyber aspects of a CPS organization. Factors contributing to the lack of accessible, detailed, realistic, coherent, and holistic CPS organizational models have been discussed. We presented Acme Corporation, a power utility-based model, that is fictional-yet-realistic, detailed, coherent, and holistic in nature. Acme Corp. is a part of a cyber physical systems' modeling project, Project METICS. We presented examples of potential CPS security research projects that can use Acme Corp. model. We discussed about related work and presented the difference between these related works and the Acme Corp. model. We hope that our endeavor, Acme Corp., will help CPS security researchers validate and share their research and results

using the holistic CPS model introduced in this Chapter. Interested persons can freely use and improve upon the model. Please visit the GitHub page at <https://github.com/METICS-CPS>.

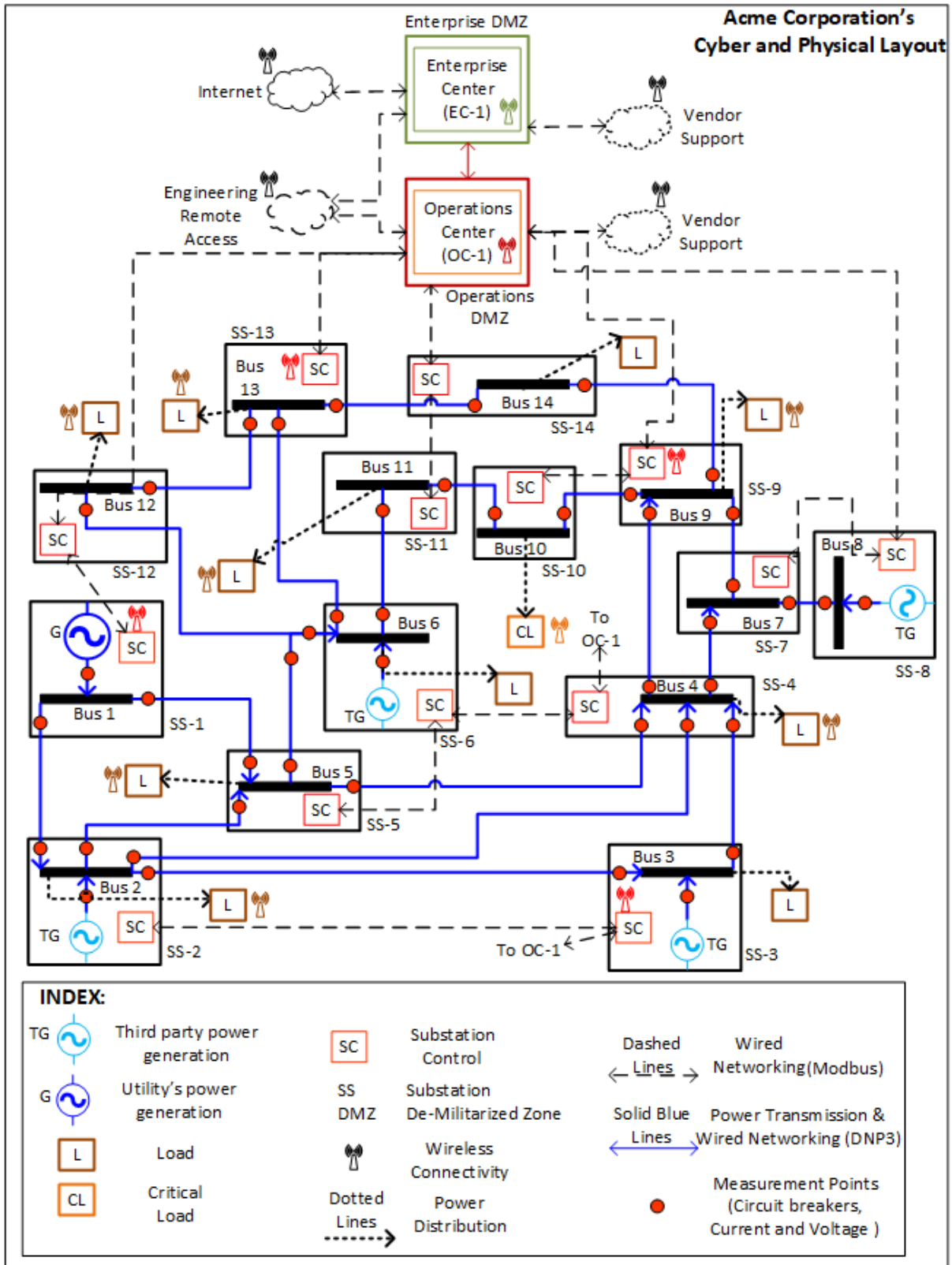


Figure 5.1: A high-level diagram of Acme Corp., showing both cyber and physical networks. Inspiration for Acme Corp.'s power flow and bus structure has been derived from IEEE 14-bus system [36]. Figure License CC-BY.

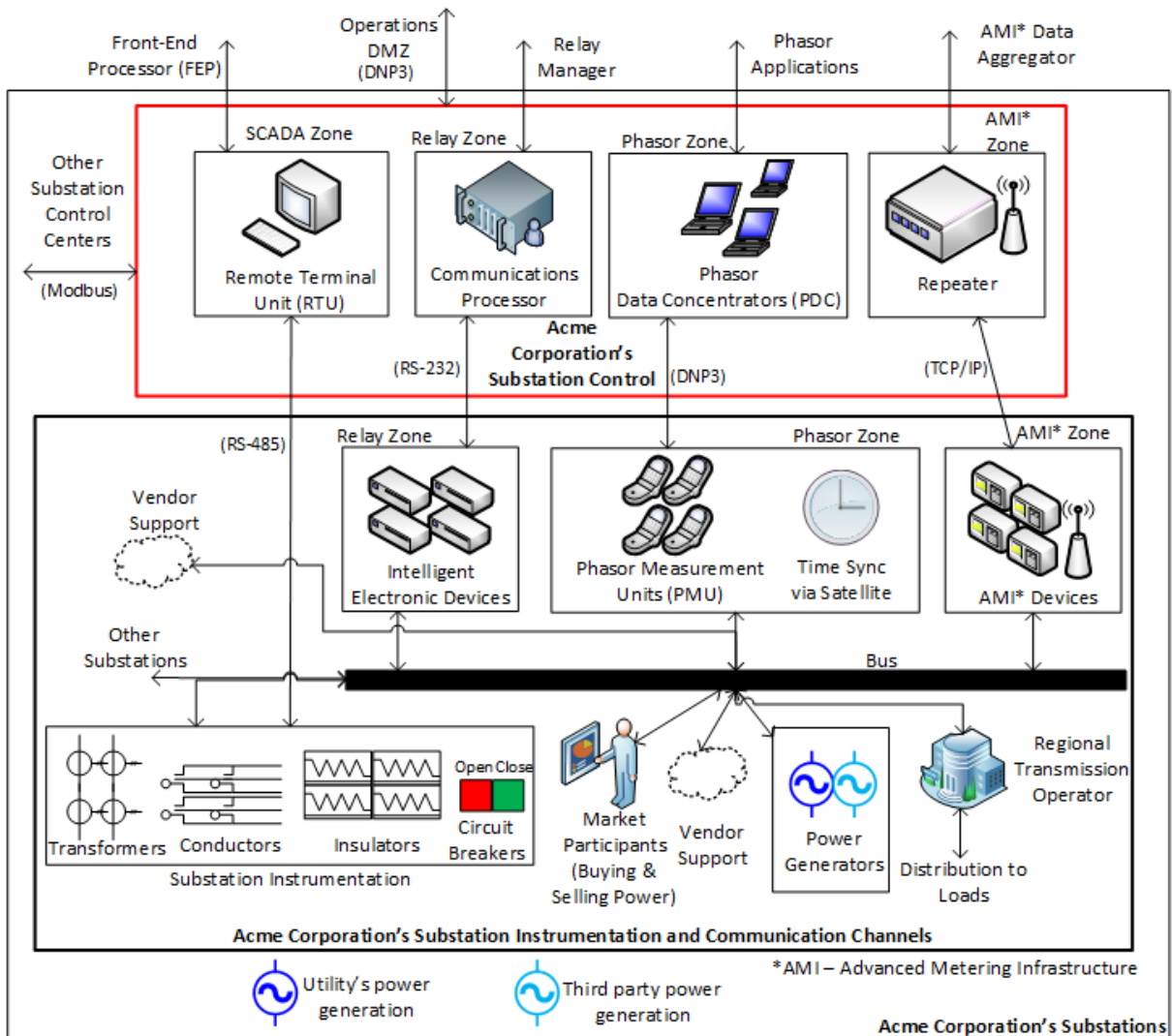


Figure 5.2: A high-level diagram of an Acme Corp.'s substation, showing both cyber and physical networks. Figure License CC-BY.

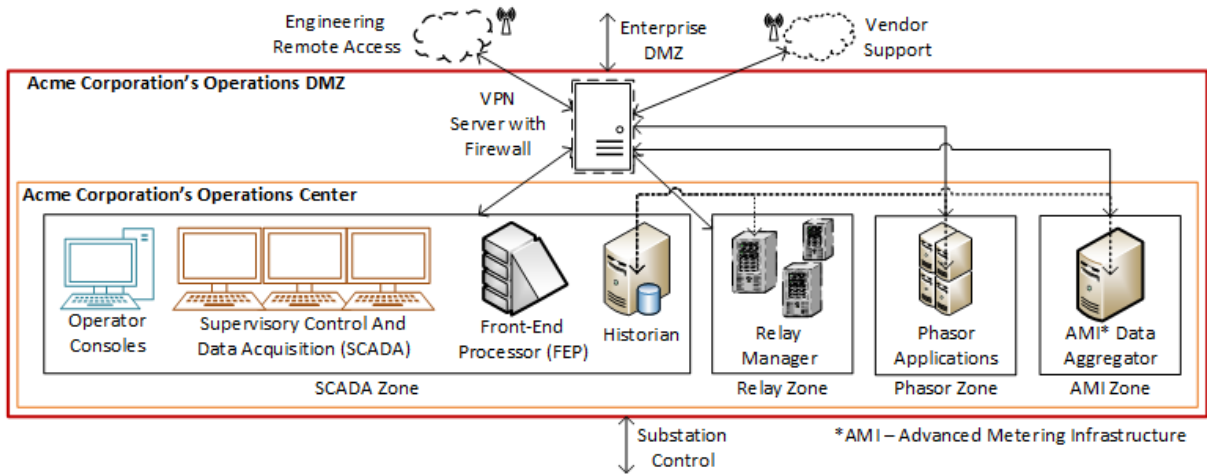


Figure 5.3: A high-level diagram of Acme Corp.'s Operations Center. Figure License CC-BY.

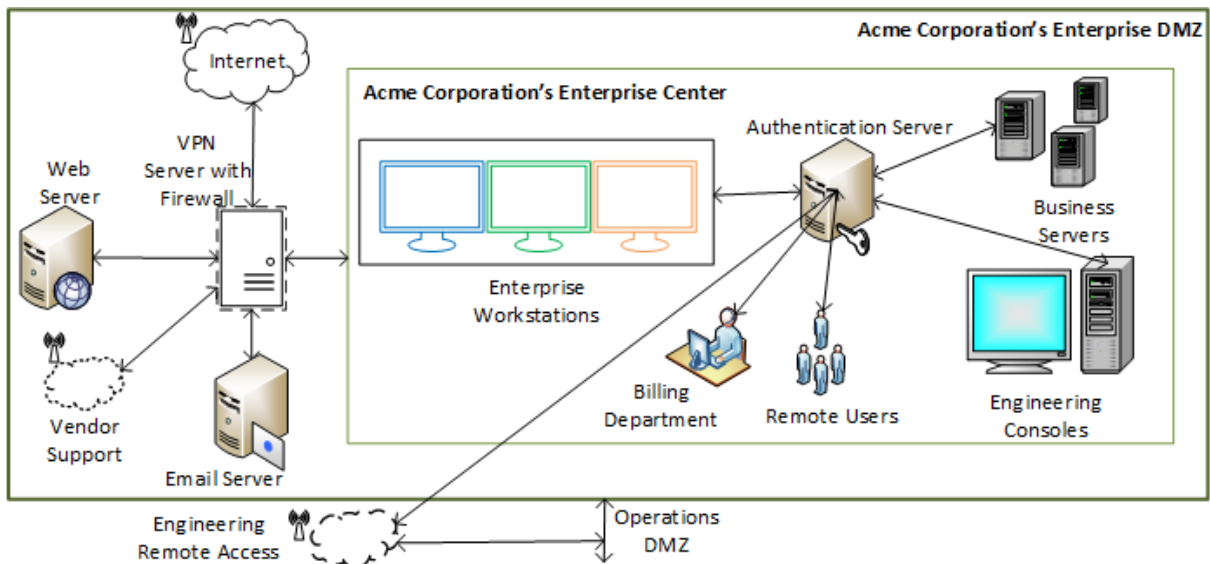


Figure 5.4: A high-level diagram of Acme Corp.'s Enterprise Center. Figure License CC-BY.

CHAPTER 6: FORMAL VERIFICATION AND PROOF

TRACING FOR THE HESTIA PROCESS

6.1 CHAPTER CONTRIBUTION

We present a formal verification of HESTIA’s semantic model by using the ‘proof by structural induction’ strategy. We formally verify all four of HESTIA’s definitions: consistency checking, conflict checking, applicability checking, and delta application. Further, we present a pseudo code implementation of HESTIA on an example. The example we use in our pseudo code implementation is based upon METICS, which is a previously published holistic model for IEEE 14-bus power system security [3].

Specifications used in the example are written using the HERMES language. The evolving grammar and syntactic notation of the HERMES language has been presented in previous Chapters [9], [33]. Interested parties can represent specifications using any Domain Specific Language (DSL) [32], as long as the DSL has the following features: 1) represent organizational domain knowledge, across both technical and managerial domains; 2) encode policies as specifications at a granular level, down to the lowest possible node; 3) modularly express specifications in a high-level, English-like, language; 4) function independently of the platform being used; and 5) separate sets of specifications into fully independent entities. We decided to use the HERMES language in our example because its capabilities include the features listed above.

6.2 CHAPTER OUTLINE

The rest of this Chapter is organized as follows: Sections 6.3 through 6.28 provide a formulation of lemmas and theorems for HESTIA’s operational definitions and formal proofs for the formulated lemmas and theorems. Section 6.29 presents an overview of our proof tracing strategy for the HESTIA formal system, using an example based on the previously described Acme Corp case study [3]. Sections 6.34, 6.35, and 6.36 respectively present proof traces for theorems of consistency checking, conflict checking, and applicability checking. Chapter conclusion and a list of abbreviations follows.

6.3 FORMALLY VERIFYING LEMMAS AND THEOREMS FOR HESTIA

We have presented definitions of the HESTIA process’ operational definitions in previous Chapter [39]. Here we formulate lemmas and theorems based on the definitions of operational definitions.

There exist several strategies for formal verification, some of which are: 1) proof by structural in-

duction, 2) proof by example or counterexample, 3) proof by exhaustion, 4) proof by reduction to the absurd, 5) proof by contraposition, and 6) proof by mathematical induction. Extensive literature exists to define each type of proofing strategy, differences between them, and their uses [40], [41].

Our goal is to formally verify all lemmas and theorems of HESTIA. We have chosen the formal proofing strategy of *proof by structural induction* due to the semantic nature of the HESTIA process.

6.4 LEMMA OF ALPHANUMERIC STRING EQUALITY

Lemma 1 *Given two `AlphanumericString` variables, HESTIA always returns the true flag, `T`, when the variables are equal or always returns the false flag, `F`, when they are not equal.*

Proving Lemma of AlphanumericString Equality

PROOF SKETCH: We assume two `AlphanumericString` variables and we obtain a confirmation of the lemma for the two variables by using the definition of ASE operation [39].

ASSUME:

1. $v^a \in \mathcal{V}$
2. $v^b \in \mathcal{V}$

PROVE:

$$(\text{ASE}(v^a, v^b) = \text{T}) \text{ OR } (\text{ASE}(v^a, v^b) = \text{F})$$

PROOF: By ASE operational definition [39],

1. $\{\text{ASE}(v^a, v^b) = \text{T}\} \iff \{v^a = v^b\}$
2. $\{\text{ASE}(v^a, v^b) = \text{F}\} \iff \{v^a \neq v^b\}$

Quod Erat Demonstrandum(Q.E.D.)

6.5 LEMMA OF SIZE DETERMINATION

Lemma 2 *Given a dotted string, a named string, or a dictionary register variable, HESTIA always returns the last index position number of the register. Given a value register variable, HESTIA always returns the number 1.*

Proving Lemma of Size Determination

PROOF SKETCH: We assume a field register variable and we obtain a confirmation of the lemma for the variable by using the SD definition [39].

ASSUME:

$$1. R^k \in \{\mathcal{Q}\} \vee \{\mathcal{D}\} \vee \{\mathcal{L}\} \vee \{\mathcal{V}\}$$

PROVE:

$$(\text{SD}(R^k)) \neq \emptyset$$

PROOF: By SD operational definition [39],

$$1. \{\text{SD}(R^k) = i\} \text{ when}$$

$$(a) R^k = p_1^k \cdot p_2^k \dots p_i^k$$

$$(b) R^k = [v_1^k \cdot v_2^k \dots v_i^k]$$

$$(c) R^k = \{(k : v)_1^k \cdot (k : v)_2^k \dots (k : v)_i^k\}$$

$$2. \text{ Otherwise } \{\text{SD}(R^k) = 1\}$$

Q.E.D.

6.6 LEMMA OF SIZE COMPARISON

Lemma 3 *Given two field register variables, HESTIA always returns the equal flag E when the two registers are equal in size. HESTIA always returns the greater flag, L, when the left register is larger in size than the right one. Finally, HESTIA always returns the lesser flag, R, when the left register is smaller in size than the right one.*

Proving Lemma of Size Comparison

PROOF SKETCH: We assume two field register variables and we obtain a confirmation of the lemma for the variables by using the SC definition [39].

ASSUME:

$$1. R^a \in \mathcal{F}$$

$$2. R^b \in \mathcal{F}$$

PROVE:

$$(\text{SC}(R^a, R^b) = \text{E}) \text{ OR } (\text{SC}(R^a, R^b) = \text{L}) \\ \text{OR } (\text{SC}(R^a, R^b) = \text{R})$$

PROOF: By SC operational definition [39],

1. $\{\text{SC}(R^a, R^b) = \text{E}\} \iff \{\text{SD}(R^a) = \text{SD}(R^b)\}$
2. $\{\text{SC}(R^a, R^b) = \text{L}\} \iff \{\text{SD}(R^a) > \text{SD}(R^b)\}$
3. $\{\text{SC}(R^a, R^b) = \text{R}\} \iff \{\text{SD}(R^a) < \text{SD}(R^b)\}$
4. Using Lemma 6.5,

Q.E.D.

6.7 LEMMA OF DOTTED STRING EQUALITY

Lemma 4 *Given two dotted string variables, HESTIA always returns the true flag, T, when the variables are equal or returns the false flag, F, when they are not equal.*

Proving Lemma of Dotted String Equality

PROOF SKETCH: We assume two dotted string variables and we obtain a confirmation of the lemma for the two variables by using the definition of DSE operation [39].

ASSUME:

1. $Q^a \in \mathcal{Q}$
2. $Q^a = \{p^1.p^2.....p^a\}$
3. $Q^b \in \mathcal{Q}$
4. $Q^b = \{p^1.p^2.....p^b\}$

PROVE:

$$(\text{DSE}(Q^a, Q^b) = \text{T}) \text{ OR } (\text{DSE}(Q^a, Q^b) = \text{F})$$

PROOF: By DSE operational definition [39],

1. $(\text{DSE}(Q^a, Q^b) = \text{T}) \iff (\text{SC}(Q^a, Q^b) = \text{E} \wedge \forall ((p^a \in Q^a) \wedge (p^b \in Q^b))), (\exists! (\text{ASE}(p^a, p^b) = \text{F})) \vee$
2. $(\text{DSE}(Q^a, Q^b) = \text{F}) \iff (\text{SC}(Q^a, Q^b) = \text{E} \wedge \forall ((p^a \in Q^a) \wedge (p^b \in Q^b))), (\exists (\text{ASE}(p^a, p^b) = \text{F}))$
3. Using Lemma 6.4,

Q.E.D.

6.8 LEMMA OF DOTTED STRING INTERSECTION IDENTIFICATION

Lemma 5 *Given two dotted string variables, HESTIA always returns the true flag, T, when the variables have at least one equal substring at the same index. Otherwise, HESTIA returns the false flag, F.*

Proving Lemma of Dotted String Intersection Identification

PROOF SKETCH: We assume two dotted string variables and we obtain a confirmation of the lemma for the two variables by using the definition of DSII operation [39].

ASSUME:

1. $Q^a \in \mathcal{Q}$
2. $Q^a = \{p^1.p^2.....p^a\}$
3. $Q^b \in \mathcal{Q}$
4. $Q^b = \{p^1.p^2.....p^b\}$

PROVE:

$$(\text{DSII}(Q^a, Q^b) = \text{T}) \text{ OR } (\text{DSII}(Q^a, Q^b) = \text{F})$$

PROOF: By DSE operational definition [39],

1. $(\text{DSII}(Q^a, Q^b) = \text{T}) \Leftrightarrow (\forall ((p^a \in Q^a) \wedge (p^b \in Q^b)), (\exists (\text{ASE}(p^a, p^b) = \text{T}))) \vee$
2. $(\text{DSII}(Q^a, Q^b) = \text{F}) \Leftrightarrow (\forall ((p^a \in Q^a) \wedge (p^b \in Q^b)), (\exists! (\text{ASE}(p^a, p^b) = \text{T})))$
3. Using Lemma 6.4,

Q.E.D.

6.9 LEMMA OF ENTITY ID COMPARISON

Lemma 6 *Given two entity ID variables, HESTIA always returns the true flag, T, when the two variables are equal. Otherwise, HESTIA returns the false flag, F.*

Proving Lemma of Entity ID Comparison

PROOF SKETCH: We assume two entity ID variables and we obtain a confirmation of the lemma for the two variables by using the definition of EIC operation [39].

ASSUME:

1. $E^a \in \mathcal{E}$
2. $E^a \stackrel{\overline{=}}{\text{def}} \{ (EID)^a \wedge (ETC)^a \wedge \{F^1, F^2, \dots, F^a\} \}$
3. $E^b \in \mathcal{E}$
4. $E^b \stackrel{\overline{=}}{\text{def}} \{ (EID)^b \wedge (ETC)^b \wedge \{F^1, F^2, \dots, F^b\} \}$

PROVE:

$$(\text{EIC}(E^a, E^b) = \text{T}) \text{ OR } (\text{EIC}(E^a, E^b) = \text{F})$$

PROOF: By EIC operational definition [39],

1. $\{\text{EIC}(E^a, E^b) = \text{T}\} \iff \{(\text{DSE}((EID)^a, (EID)^b) = \text{T})\}$
2. $\{\text{EIC}(E^a, E^b) = \text{F}\} \iff \{(\text{DSE}((EID)^a, (EID)^b) = \text{F})\}$
3. Using Lemma 6.7,

Q.E.D.

6.10 LEMMA OF ENTITY TYPE COMPARISON

Lemma 7 *Given two entity type variables, HESTIA always returns the true flag, T, when the two variables are equal. Otherwise, HESTIA returns the false flag, F.*

Proving Lemma of Entity Type Comparison

PROOF SKETCH: We assume two entity type variables and we obtain a confirmation of the lemma for the two variables by using the definition of ETC operation [39].

ASSUME:

1. $E^a \in \mathcal{E}$
2. $E^a \stackrel{\overline{=}}{\text{def}} \{ (EID)^a \wedge (ETC)^a \wedge \{F^1, F^2, \dots, F^a\} \}$
3. $E^b \in \mathcal{E}$
4. $E^b \stackrel{\overline{=}}{\text{def}} \{ (EID)^b \wedge (ETC)^b \wedge \{F^1, F^2, \dots, F^b\} \}$

PROVE:

$$(\text{ETC}(E^a, E^b) = \text{T}) \text{ OR } (\text{ETC}(E^a, E^b) = \text{F})$$

PROOF: By ETC operational definition [39],

1. $\{\text{ETC}(E^a, E^b) = \mathbf{T}\} \iff \{(\text{DSE}((ET)^a, (ET)^b) = \mathbf{T})\}$
2. $\{\text{ETC}(E^a, E^b) = \mathbf{F}\} \iff \{(\text{DSE}((ET)^a, (ET)^b) = \mathbf{F})\}$
3. Using Lemma 6.7,

Q.E.D.

6.11 LEMMA OF FIELD NAME COMPARISON

Lemma 8 *Given two field name variables, HESTIA always returns the true flag, T, when the two variables are equal. Otherwise, HESTIA returns the false flag, F.*

Proving Lemma of Field Name Comparison

PROOF SKETCH: We assume two field name variables and we obtain a confirmation of the lemma for the two variables by using the definition of FNC operation [39].

ASSUME:

1. $F^a \in \mathcal{F}$
2. $F^a \stackrel{=}{def} \{(FN)^a \wedge R^a\}$
3. $F^b \in \mathcal{F}$
4. $F^b \stackrel{=}{def} \{(FN)^b \wedge R^b\}$

PROVE:

$$(\text{FNC}(F^a, F^b) = \mathbf{T}) \text{ OR } (\text{FNC}(F^a, F^b) = \mathbf{F})$$

PROOF: By FNC operational definition [39],

1. $\{\text{FNC}(F^a, F^b) = \mathbf{T}\} \iff \{(\text{ASE}((FN)^a, (FN)^b) = \mathbf{T})\}$
2. $\{\text{FNC}(F^a, F^b) = \mathbf{F}\} \iff \{(\text{ASE}((FN)^a, (FN)^b) = \mathbf{F})\}$
3. Using Lemma 6.4,

Q.E.D.

6.12 LEMMA OF FIELD REGISTER TYPE DETERMINATION

Lemma 9 *Given a field register variable, HESTIA always returns the flag, U, when the register is a value. The flag, G, is always returned when the register is a dotted string. The flag, C, is always returned when the register is a dictionary. The flag, I, is always returned when the register is a named set.*

Proving Lemma of Field Register Type Determination

PROOF SKETCH: We assume two field register variables and we obtain a confirmation of the lemma for the two variables by using the definition of FRTD operation [39].

ASSUME:

$$1. R^a \in \mathcal{F}$$

PROVE:

$$(\text{FRTD}(R^a) = \text{U}) \text{ OR } (\text{FRTD}(R^a) = \text{G})$$

$$\text{OR } (\text{FRTD}(R^a) = \text{C}) \text{ OR } (\text{FRTD}(R^a) = \text{I})$$

PROOF: By FRTD operational definition [39],

$$1. (\text{FRTD}(R^a) = \text{U}) \iff (R^a \in \mathcal{V})$$

$$2. (\text{FRTD}(R^a) = \text{G}) \iff (R^a \in \mathcal{Q})$$

$$3. (\text{FRTD}(R^a) = \text{C}) \iff (R^a \in \mathcal{D})$$

$$4. (\text{FRTD}(R^a) = \text{I}) \iff (R^a \in \mathcal{L})$$

Q.E.D.

6.13 LEMMA OF FIELD REGISTER TYPE COMPARISON

Lemma 10 *Given two field register variables, HESTIA always returns the true flag, T, when types of the registers match. Otherwise, a false flag, F, is returned.*

Proving Lemma of Field Register Type Comparison

PROOF SKETCH: We assume two field register variables and we obtain a confirmation of the lemma for the two variables by using the definition of FRTC operation [39].

ASSUME:

1. $R^a \in \mathcal{F}$
2. $R^b \in \mathcal{F}$

PROVE:

$$(\text{FRTC}(R^a, R^b) = \text{T}) \text{ OR } (\text{FRTC}(R^a, R^b) = \text{F})$$

PROOF: By FRTC operational definition [39],

1. $\{\text{FRTC}(R^a, R^b) = \text{T}\} \iff \{\text{FRTD}(R^a) = \text{FRTD}(R^b)\}$
2. $\{\text{FRTC}(R^a, R^b) = \text{F}\} \iff \{\text{FRTD}(R^a) \neq \text{FRTD}(R^b)\}$
3. Using Lemma 6.12,

Q.E.D.

6.14 LEMMA OF DICTIONARY EQUALITY

Lemma 11 *Given two dictionary variables, HESTIA always returns the true flag, T, when the key-value pairs of the two given dictionaries are the same. Otherwise, a false flag, F, is returned.*

Proving Lemma of Dictionary Equality

PROOF SKETCH: We assume two dictionary variables and we obtain a confirmation of the lemma for the two variables by using the definition of DE operation [39].

ASSUME:

1. $D^a \in \mathcal{D}$
2. $D^a \stackrel{\overline{\text{def}}}{=} \{(k : v)^1, (k : v)^2, \dots, (k : v)^a\}$
3. $D^b \in \mathcal{D}$
4. $D^b \stackrel{\overline{\text{def}}}{=} \{(k : v)^1, (k : v)^2, \dots, (k : v)^b\}$
5. $\text{SC}(D^a, D^b) = \text{E}$

PROVE:

$$(\text{DE}(D^a, D^b) = \text{T}) \text{ OR } (\text{DE}(D^a, D^b) = \text{F})$$

PROOF: By DE operational definition [39],

$$\forall\{(k : v)^1, (k : v)^2, \dots, (k : v)^a\} \wedge$$

$$\forall\{(k : v)^1, (k : v)^2, \dots, (k : v)^b\} :$$

$$1. \{\text{DE}(D^a, D^b) = \text{T}\} \iff \{(\text{ASE}(k^a, k^b) = \text{T}) \wedge (\text{ASE}(v^a, v^b) = \text{T})\}$$

$$2. \{\text{DE}(D^a, D^b) = \text{F}\} \iff \{(\text{ASE}(k^a, k^b) = \text{F})\}$$

3. Using Lemma 6.4,

Q.E.D.

6.15 LEMMA OF DICTIONARY KEY INTERSECTION IDENTIFICATION

Lemma 12 *Given two dictionary variables, HESTIA always returns the true flag, T, if there exists at least one same key across both the dictionaries. Otherwise, the false flag, F is returned.*

Proving Lemma of Dictionary Key Intersection Identification

PROOF SKETCH: We assume two dictionary variables and we obtain a confirmation of the lemma for the two variables by using the definition of DKII operation [39].

ASSUME:

$$1. D^a \in \mathcal{D}$$

$$2. D^a \stackrel{\text{def}}{=} \{(k : v)^1, (k : v)^2, \dots, (k : v)^a\}$$

$$3. D^b \in \mathcal{D}$$

$$4. D^b \stackrel{\text{def}}{=} \{(k : v)^1, (k : v)^2, \dots, (k : v)^b\}$$

PROVE:

$$(\text{DKII}(D^a, D^b) = \text{T}) \text{ OR } (\text{DKII}(D^a, D^b) = \text{F})$$

PROOF: By DSE operational definition [39],

$$1. (\text{DKII}(D^a, D^b) = \text{T}) \Leftrightarrow (\forall(((k : v)^a \in D^a) \wedge ((k : v)^b \in D^b)), (\exists((\text{ASE}(k^a, k^b) = \text{T})))$$

2. Using Lemma 6.4,

Q.E.D.

6.16 LEMMA OF NAMED SET EQUALITY

Lemma 13 *Given two named set variables, HESTIA always returns the true flag, T, when all the values of given two named sets are the same. Otherwise, a false flag, F, is returned.*

Proving Lemma of Named Set Equality

PROOF SKETCH: We assume two named set variables and we obtain a confirmation of the lemma for the two variables by using the definition of NSE operation [39].

ASSUME:

1. $L^a \in \mathcal{L}$
2. $L^a \stackrel{=}{def} [v^1, v^2, \dots, v^a]$
3. $L^b \in \mathcal{L}$
4. $L^b \stackrel{=}{def} [v^1, v^2, \dots, v^b]$

PROVE:

$$(\text{NSE}(L^a, L^b) = \text{T}) \text{ OR } (\text{NSE}(L^a, L^b) = \text{F})$$

PROOF: By NSEE operational definition [39],

1. $(\text{NSE}(L^a, L^b) = \text{T}) \Leftrightarrow (\text{SC}(L^a, L^b) = \text{E}) \wedge (\forall ((v^a \in L^a) \wedge (v^b \in L^b))), (\exists! (\text{ASE}(v^a, v^b) = \text{F}))$
2. Using Lemma 6.4,

Q.E.D.

6.17 LEMMA OF NAMED SET INTERSECTION IDENTIFICATION

Lemma 14 *Given two named set variables, HESTIA always returns the true flag, T if there exists at least one equal value between the two named sets. Otherwise, a false flag F is returned.*

Proving Lemma of Named Set Intersection Identification

PROOF SKETCH: We assume two named set variables and we obtain a confirmation of the lemma for the two variables by using the definition of NSII operation [39].

ASSUME:

1. $L^a \in \mathcal{L}$
2. $L^a \stackrel{\text{def}}{=} [v^1, v^2, \dots, v^a]$
3. $L^b \in \mathcal{L}$
4. $L^b \stackrel{\text{def}}{=} [v^1, v^2, \dots, v^b]$

PROVE:

$$(\text{NSII}(L^a, L^b) = \text{T}) \text{ OR } (\text{NSII}(L^a, L^b) = \text{F})$$

PROOF: By DSE operational definition [39],

1. $(\text{NSII}(L^a, L^b) = \text{T}) \Leftrightarrow (\forall ((v^a \in L^a) \wedge (v^b \in L^b))), (\exists (\text{ASE}(v^a, v^b) = \text{T}))$
2. $(\text{NSII}(L^a, L^b) = \text{F}) \Leftrightarrow (\forall ((v^a \in L^a) \wedge (v^b \in L^b))), (\exists! (\text{ASE}(v^a, v^b) = \text{T}))$
3. Using Lemma 6.4,

Q.E.D.

6.18 LEMMA OF FIELD REGISTER COMPARISON

Lemma 15 *Given two field register variables, HESTIA always returns the true flag, T, if the two registers are equal. Otherwise, a false flag, F, is returned.*

Proving Lemma of Field Register Comparison

PROOF SKETCH: We assume two field register variables and we obtain a confirmation of the lemma for the two variables by using the definition of FRC operation [39].

ASSUME:

1. $R^a \in \mathcal{F}$
2. $R^b \in \mathcal{F}$
3. $\text{SC}(R^a, R^b) = \text{E}$
4. $\text{FRTC}(R^a, R^b) = \text{T}$

PROVE:

$$(\text{FRC}(R^a, R^b) = \text{T}) \text{ OR } (\text{FRC}(R^a, R^b) = \text{F})$$

PROOF: By FRC operational definition [39],

1. $\{\text{FRC}(R^a, R^b) = \text{T}\} \iff \{(\text{ASE}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{U})\}$
2. $\{\text{FRC}(R^a, R^b) = \text{T}\} \iff \{(\text{DSE}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{G})\}$
3. $\{\text{FRC}(R^a, R^b) = \text{T}\} \iff \{(\text{DE}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{C})\}$
4. $\{\text{FRC}(R^a, R^b) = \text{T}\} \iff \{(\text{NSE}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{I})\}$
5. *Otherwise* $\{\text{FRC}(R^a, R^b) = \text{F}\}$
6. Using Lemmas $\{6.4, 6.7, 6.14, 6.16, 6.12\}$

Q.E.D.

6.19 LEMMA OF FIELD REGISTER RELAXED COMPARISON

Lemma 16 *Given two field register variables, HESTIA always returns the true flag, T, if the two registers have at least one similar register content. Otherwise, a false flag, F, is returned.*

Proving Lemma of Field Register Relaxed Comparison

PROOF SKETCH: We assume two field register variables and we obtain a confirmation of the lemma for the two variables by using the definition of FRRC operation [39].

ASSUME:

1. $R^a \in \mathcal{F}$
2. $R^b \in \mathcal{F}$
3. $\text{FRTC}(R^a, R^b) = \text{T}$

PROVE:

$$(\text{FRRC}(R^a, R^b) = \text{T}) \text{ OR } (\text{FRRC}(R^a, R^b) = \text{F})$$

PROOF: By FRRC operational definition [39],

1. $\{\text{FRRC}(R^a, R^b) = \text{T}\} \iff \{(\text{ASE}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{U})\}$
2. $\{\text{FRRC}(R^a, R^b) = \text{T}\} \iff \{(\text{DSE}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{G})\} \text{ OR}$
3. $\{\text{FRRC}(R^a, R^b) = \text{T}\} \iff \{(\text{DSII}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{G})\}$

4. $\{\text{FRRC}(R^a, R^b) = \text{T}\} \iff \{(\text{DE}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{C})\}$ OR
5. $\{\text{FRRC}(R^a, R^b) = \text{T}\} \iff \{(\text{DKII}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{C})\}$
6. $\{\text{FRRC}(R^a, R^b) = \text{T}\} \iff \{(\text{NSE}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{I})\}$ OR
7. $\{\text{FRRC}(R^a, R^b) = \text{T}\} \iff \{(\text{NSE}(R^a, R^b) = \text{T}) \vee (\text{FRTD}(R^a) = \text{I})\}$
8. *Otherwise* $\{\text{FRRC}(R^a, R^b) = \text{F}\}$
9. Using Lemmas $\{6.4, 6.7, 6.8, 6.14, 6.15, 6.16, 6.17, 6.12\}$

Q.E.D.

6.20 LEMMA OF FIELD REGISTER APPLICATION

Lemma 17 *Given two field register variables, HESTIA always overwrites the contents of right register with the contents of left register.*

Proving Lemma of Field Register Application

PROOF SKETCH: We assume two field register variables and we obtain a confirmation of the lemma for the two variables by using the definition of FRA operation [39].

ASSUME:

1. $R^a \in \mathcal{F}$
2. $R^b \in \mathcal{F}$
3. $\text{FRTC}(R^a, R^b) = \text{T}$

PROVE:

$$(\exists \text{FRA}(R^a, R^b))$$

PROOF: By FRA operational definition [39],

1. $(\text{FRA}(R^a, R^b)) \stackrel{=}{\text{def}} (R^{b^p}), \text{ where } : (R^{b^p} = R^a)$

Q.E.D.

6.21 LEMMA OF ENTITY CONSISTENCY CHECKING

Lemma 18 *Given an Entity, E_s^a , and its corresponding template, E_t^b , HESTIA always returns the true flag, T , when the given entity is consistent with its corresponding template. Otherwise, a false flag, F , is returned.*

Proving Lemma of Entity Consistency Checking

PROOF SKETCH: We assume two entities and we obtain a confirmation of the lemma for the two entities by using the definition of Consistency Checking (\propto) operation [39].

ASSUME:

1. $\{(E_s^1, E_s^2, \dots, E_s^a) \in \mathcal{E}\}$
2. $\{(E_t^1, E_t^2, \dots, E_t^b) \in \mathcal{E}\}$
3. $\{(F_s^1, F_s^2, \dots, F_s^a) \in \mathcal{F}\} \wedge \{(F_s^1, F_s^2, \dots, F_s^a) \in E_s^a\}$
4. $\{(F_t^1, F_t^2, \dots, F_t^b) \in \mathcal{F}\} \wedge \{(F_t^1, F_t^2, \dots, F_t^b) \in E_t^b\}$
5. *a and b are finite*

RECALL: Entity Consistency Checking (\propto) operational definition [39],

$$((\propto (E_s^a, E_t^b) = T) \Leftrightarrow (\text{ETC}(E_s^a, E_t^b) = T) \wedge \forall F_t^y, \exists F_s^x : (\text{FNC}(F_s^x, F_t^y) = T) \\ (\text{FRC}(F_s^x, F_t^y) = F))$$

PROVE:

$$(\propto (E_s^a, E_t^b) = T) \text{ OR } (\propto (E_s^a, E_t^b) = F)$$

PROOF:

1. Using Lemmas {6.10 and 6.11}: $(\propto (E_s^a, E_t^b) = T) \vee (\propto (E_s^a, E_t^b) = F)$

Q.E.D.

6.22 LEMMA OF ENTITY CONFLICT CHECKING

Lemma 19 *Given two consistent Entities, E_s^a and E_m^b , HESTIA always returns the true flag, T , when the two given entities have a conflict among them. Otherwise, a false flag, F , is returned.*

Proving Lemma of Entity Conflict Checking

PROOF SKETCH: We assume two consistent entities and we obtain a confirmation of the lemma for the two entities by using the definition of Conflicts Checking (\otimes) operation [39].

ASSUME:

1. $\{(E_s^1, E_s^2, \dots, E_s^a) \in \mathcal{E}\}$
2. $\{(E_m^1, E_m^2, \dots, E_m^b) \in \mathcal{E}\}$
3. $\{\{E_t^1, E_t^1\} \in \mathcal{E}\} \wedge \{\{E_t^1, E_t^1\} \in \mathcal{T}\}$
4. $(\propto(E_s^a, E_t^1)) \wedge (\propto(E_m^b, E_t^2))$
5. $\{(F_s^1, F_s^2, \dots, F_s^a) \in \mathcal{F}\} \wedge \{(F_s^1, F_s^2, \dots, F_s^a) \in E_s^a\}$
6. $\{(F_m^1, F_m^2, \dots, F_m^b) \in \mathcal{F}\} \wedge \{(F_m^1, F_m^2, \dots, F_m^b) \in E_m^b\}$
7. *a and b are finite*

RECALL: Conflict Checking (\otimes) operational definition [39],

$$\begin{aligned}
 (\otimes(E_s^a, E_m^b) = \mathbf{T}) &\Leftrightarrow ((\text{ETC}(E_s^a, E_m^b) = \mathbf{T}) \wedge \\
 &(\text{EIC}(E_s^a, E_m^b) = \mathbf{T}) \wedge \\
 \forall ((F_s^x, F_m^y) \in \{\{F_s^1, F_s^2, \dots, F_s^a\} \times \{F_m^1, F_m^2, \dots, F_m^b\}\}) : &\exists((\text{FNC}(F_s^x, F_m^y) = \mathbf{T}) \wedge \\
 &(\text{FRC}(F_s^x, F_m^y) = \mathbf{F})))
 \end{aligned}$$

PROVE:

$$(\otimes(E_s^a, E_m^b) = \mathbf{T}) \text{ OR } (\otimes(E_s^a, E_m^b) = \mathbf{F})$$

PROOF:

1. Using Lemmas {6.10, 6.9, 6.11, and 6.18}: $(\otimes(E_s^a, E_m^b) = \mathbf{T}) \vee (\propto(E_s^a, E_m^b) = \mathbf{F})$

Q.E.D.

6.23 LEMMA OF ENTITY APPLICABILITY CHECKING

Lemma 20 *Given two consistent Entities, E_m^a and E_s^b , HESTIA always returns the true flag, T, when the left entity, E_m^a , is applicable to the right entity, E_s^b . Otherwise, a false flag, F, is returned.*

Proving Lemma of Entity Applicability Checking

PROOF SKETCH: We assume two consistent entities and we obtain a confirmation of the lemma for the two entities by using the definition of Applicability Checking ($\overset{?}{\vdash}$) operation [39].

ASSUME:

1. $\{(E_s^1, E_s^2, \dots, E_s^a) \in \mathcal{E}\}$
2. $\{(E_m^1, E_m^2, \dots, E_m^b) \in \mathcal{E}\}$
3. $\{\{E_t^1, E_t^1\} \in \mathcal{E}\} \wedge \{\{E_t^1, E_t^1\} \in \mathcal{T}\}$
4. $(\propto(E_s^a, E_t^1)) \wedge (\propto(E_m^b, E_t^2))$
5. $\{(F_s^1, F_s^2, \dots, F_s^a) \in \mathcal{F}\} \wedge \{(F_s^1, F_s^2, \dots, F_s^a) \in E_s^a\}$
6. $\{(F_m^1, F_m^2, \dots, F_m^b) \in \mathcal{F}\} \wedge \{(F_m^1, F_m^2, \dots, F_m^b) \in E_m^b\}$
7. *a and b are finite*

RECALL: Applicability Checking ($\overset{?}{\vdash}$) operational definition [39],

$$\overset{?}{\vdash}(E_s^a, E_m^b) = \mathbf{T} \Leftrightarrow ((\text{ETC}(E_s^a, E_m^b) = \mathbf{T}) \wedge$$

$$(\text{EIC}(E_s^a, E_m^b) = \mathbf{T}) \wedge$$

$$\forall ((F_s^x, F_m^y) \in \{\{F_s^1, F_s^2, \dots, F_s^a\} \times \{F_m^1, F_m^2, \dots, F_m^b\}\}) : \exists((\text{FNC}(F_s^x, F_m^y) = \mathbf{T}) \wedge$$

$$(\text{FRC}(F_s^x, F_m^y) = \mathbf{F})))$$

PROVE:

$$\overset{?}{\vdash}(E_m^b, E_s^a) = \mathbf{T} \text{ OR } \overset{?}{\vdash}(E_m^b, E_s^a) = \mathbf{F}$$

PROOF:

1. Using Lemmas {6.10, 6.9, 6.11, 6.18, and 6.19}: $(\otimes(E_m^b, E_s^a) = \mathbf{T}) \vee (\propto(E_m^b, E_s^a) = \mathbf{F})$

Q.E.D.

6.24 LEMMA OF ENTITY DELTA APPLICATION

Lemma 21 *Given two consistent entities, E_s^a and E_m^b , and given that the left entity, E_s^a , is applicable to the right entity, E_m^b , HESTIA can apply left entity, E_s^a , to the right entity, E_m^b , and it always returns a new entity, E_n^c .*

Proving Lemma of Entity Delta Application

PROOF SKETCH: We assume two consistent entities and we obtain a confirmation of the lemma for the two entities by using the definition of Delta Application (\vdash) operation [39].

ASSUME:

1. $\{(E_s^1, E_s^2, \dots, E_s^a) \in \mathcal{E}\}$
2. $\{(E_m^1, E_m^2, \dots, E_m^b) \in \mathcal{E}\}$
3. $\{E_s^t, E_m^t\} \in \mathcal{E}$
4. $(\propto (E_s^a, E_t^1)) \wedge (\propto (E_m^b, E_t^2)) \wedge (\vdash^? (E_s^a, E_m^b))$
5. $\{(F_s^1, F_s^2, \dots, F_s^a) \in \mathcal{F}\} \wedge \{(F_s^1, F_s^2, \dots, F_s^a) \in E_s^a\}$
6. $\{(F_m^1, F_m^2, \dots, F_m^b) \in \mathcal{F}\} \wedge \{(F_m^1, F_m^2, \dots, F_m^b) \in E_m^b\}$
7. *a and b are finite*

RECALL: Delta Application (\vdash) operational definition [39],

$$(E_n^c \stackrel{=}{def} E_m^b) \iff (\forall ((F_s^x, F_m^y) \in \{\{F_s^1, F_s^2, \dots, F_s^a\} \times \{F_m^1, F_m^2, \dots, F_m^b\}\}) : (\vdash^? (E_s^a, E_m^b) = \mathbf{F}))$$

\(\vee\)

$$(E_n^c \stackrel{=}{def} \{F_n^1, F_n^2, \dots, F_n^c\}) \iff (\forall ((F_s^x, F_m^y) : (\vdash^? (E_s^a, E_m^b) = \mathbf{T})), \text{where}$$

$$(\forall F_n^z \in \{F_n^1, F_n^2, \dots, F_n^c\}) :$$

$$(F_n^z \stackrel{=}{def} F_m^y) \iff ((\mathbf{FNC}(F_s^x, F_m^y) = \mathbf{F}) \vee$$

$$(F_n^z \stackrel{=}{def} \mathbf{FRA}(F_s^x, F_m^y)) \iff ((\mathbf{FNC}(F_s^x, F_m^y) = \mathbf{T}) \wedge (\mathbf{FRC}(F_s^x, F_m^y) = \mathbf{F}))$$

PROVE:

$$(\vdash (E_s^a, E_m^b) = E_n^c)$$

PROOF:

1. Using Lemmas 6.11, 6.18, and 6.23: $(\otimes(E_s^a, E_m^b) = E_n^c)$

Q.E.D.

6.25 THEOREM OF CONSISTENCY CHECKING

Theorem 1 *Given a specification, S and its corresponding template, T , HESTIA always returns the true flag, T , when the given specification is type consistent with its corresponding template. Otherwise, a false flag, F , is returned.*

Proving Theorem of Consistency Checking

PROOF SKETCH: We assume two specifications and we obtain a confirmation of the theorem for the two specifications by using the definition of Consistency Checking (α) operation [39].

ASSUME:

1. $\{S \in \mathcal{S}\}$
2. $\{T \in \mathcal{T}\}$
3. $\{(E_s^1, E_s^2, \dots, E_s^a) \in \mathcal{E}\} \wedge \{(E_s^1, E_s^2, \dots, E_s^a) \in S\}$
4. $\{(E_t^1, E_t^2, \dots, E_t^b) \in \mathcal{E}\} \wedge \{(E_t^1, E_t^2, \dots, E_t^b) \in T\}$
5. *a and b are finite*

RECALL: Consistency Checking (α) operational definition [39],

$$((\alpha(S, T) = \mathbf{F}) \Leftrightarrow \forall((E_s^x, E_t^y) \in \{S \times T\}) : \exists((\text{ETC}(E_s^x, E_t^y) = \mathbf{T}) \wedge (\alpha(E_s^x, E_t^y) = \mathbf{F}))$$

PROVE:

$$(\alpha(S, T) = \mathbf{T}) \vee (\alpha(S, T) = \mathbf{F})$$

PROOF:

1. Using Lemma 6.21: $(\alpha(S, T) = \mathbf{T}) \vee (\alpha(S, T) = \mathbf{F})$

Q.E.D.

6.26 THEOREM OF CONFLICT CHECKING

Theorem 2 *Given two consistent specifications, S and M , HESTIA always returns the true flag, T , when the two given specifications have a conflict among them. Otherwise, a false flag, F , is returned.*

Proving Theorem of Conflict Checking

PROOF SKETCH: We assume two consistent specifications and we obtain a confirmation of the theorem for the two specifications by using the definition of Conflicts Checking (\otimes) operation [39].

ASSUME:

1. $\{S, M\} \in \mathcal{S}$
2. $\{T_s, T_m\} \in \mathcal{T}$
3. $(\propto(S, T_s)) \wedge (\propto(M, T_m))$
4. $\{(E_s^1, E_s^2, \dots, E_s^a) \in \mathcal{E}\} \wedge \{(E_s^1, E_s^2, \dots, E_s^a) \in S\}$
5. $\{(E_m^1, E_m^2, \dots, E_m^b) \in \mathcal{E}\} \wedge \{(E_m^1, E_m^2, \dots, E_m^b) \in M\}$
6. *a and b are finite*

RECALL: Conflict Checking (\otimes) operational definition [39],

$$(\otimes(S, M) = \mathbf{T}) \Leftrightarrow \forall ((E_s^x, E_m^y) \in \{S \times M\}) : \exists (\otimes(E_s^x, E_m^y) = \mathbf{T})$$

PROVE:

$$(\otimes(S, M) = \mathbf{T}) \text{ OR } (\otimes(S, M) = \mathbf{F})$$

PROOF:

1. Using Lemma 6.22: $(\otimes(S, M) = \mathbf{T}) \vee (\otimes(S, M) = \mathbf{F})$
2. *Q.E.D.*

6.27 THEOREM OF APPLICABILITY CHECKING

Theorem 3 *Given two consistent specifications, S and M , HESTIA always returns the true flag, T , when the left specification, S , is applicable to the right specification, S . Otherwise, a false flag, F , is returned.*

Proving Theorem of Applicability Checking

PROOF SKETCH: We assume two consistent specifications and we obtain a confirmation of the theorem for the two specifications by using the definition of Applicability Checking ($\overset{?}{\vdash}$) operation [39].

ASSUME:

1. $\{S, M\} \in \mathcal{S}$
2. $\{T_s, T_m\} \in \mathcal{T}$
3. $(\propto(S, T_s)) \wedge (\propto(M, T_m))$
4. $\{(E_s^1, E_s^2, \dots, E_s^a) \in \mathcal{E}\} \wedge \{(E_s^1, E_s^2, \dots, E_s^a) \in S\}$
5. $\{(E_m^1, E_m^2, \dots, E_m^b) \in \mathcal{E}\} \wedge \{(E_m^1, E_m^2, \dots, E_m^b) \in M\}$
6. *a and b are finite*

RECALL: Applicability Checking ($\overset{?}{\vdash}$) operational definition [39],

$$(\overset{?}{\vdash}(S, M) = \mathbf{T}) \Leftrightarrow \forall ((E_s^x, E_m^y) \in \{S \times M\}) : \exists (\overset{?}{\vdash}(E_s^x, E_m^y) = \mathbf{T})$$

PROVE:

$$(\overset{?}{\vdash}(S, M) = \mathbf{T}) \text{ OR } (\overset{?}{\vdash}(S, M) = \mathbf{F})$$

PROOF:

1. Using Lemma 6.23: $(\overset{?}{\vdash}(S, M) = \mathbf{T}) \vee (\overset{?}{\vdash}(S, M) = \mathbf{F})$

Q.E.D.

6.28 THEOREM OF DELTA APPLICATION

Theorem 4 *Given two consistent specifications, S and M, and given that the left specification (spec), S, is applicable to the right spec, M, HESTIA can apply left spec, S, to the right spec, M, and it always returns a new specification, N.*

Proving Theorem of Delta Application

PROOF SKETCH: We assume two consistent, applicable specifications and we obtain a confirmation of the theorem for the two specifications by using the definition of Delta Application (\vdash) operation [39].

ASSUME:

1. $\{S, M\} \in \mathcal{S}$
2. $\{T_s, T_m\} \in \mathcal{T}$
3. $(\propto (S, T_s)) \wedge (\propto (M, T_m)) \wedge (\vdash (M, S) = \mathbf{T})$
4. $\{(E_s^1, E_s^2, \dots, E_s^a) \in \mathcal{E}\} \wedge \{(E_s^1, E_s^2, \dots, E_s^a) \in S\}$
5. $\{(E_m^1, E_m^2, \dots, E_m^b) \in \mathcal{E}\} \wedge \{(E_m^1, E_m^2, \dots, E_m^b) \in M\}$
6. *a and b are finite*

RECALL: Delta Application Operation (\vdash) operational definition [39],

$$(\vdash (S, M) = N)$$

$$N \stackrel{=}{def} \{E_n^1, E_n^2, \dots, E_n^c\}, \text{ where}$$

$$(\forall E_n^z \in \{E_n^1, E_n^2, \dots, E_n^c\}) \wedge (\forall ((E_s^x, E_m^y) \in \{S \times M\})) :$$

$$(E_n^z \stackrel{=}{def} (\vdash (E_s^x, E_m^y)))$$

PROVE:

$$(\vdash (S, M) = N)$$

PROOF:

1. Using Lemma 6.24: $(\vdash (S, M) = N)$

Q.E.D.

We have formally verified HESTIA's Lemmas and Theorems.

6.29 EXAMPLE FOR PROOF TRACING

We present a proof tracing of the HESTIA process. We do that by applying the HESTIA process to an example. We discuss details about the example as follows.

6.30 EXAMPLE: ACME CORP AND ITS COMPONENTS

The Acme Corp enterprise setting of our example is based upon METICS, which is a previously published case study for IEEE 14-bus power system security [3]. Our example has an enterprise center, EC1, which is in turn connected to an operations center via a network De-Militarized Zone (DMZ). EC1 has employees working in it using computers called workstations. There are four employees working in EC1. The names and employee IDs of the employees are : Joe Vandal, EID01; Gold Roger, EID02; Silver Rayleigh, EID03; and Copper Naomi, EID04. The employees work in EC1 using workstations, whose IDs are: WKS01, WKS02, WKS03, WKS04, WKS05, and WKS06. Each employee is assigned a particular workstation and there exist some workstations that can be used by any employee.

EC1's DMZ allows the employees to access the company's email, Virtual Private Network (VPN), and authentication server. Three of the four enterprise employees have enterprise email accounts and there exist some global policies for the enterprise email system. Two of the four employees have access to the enterprise VPN. Active Directory (AD) is the working mechanism of the enterprise authentication server. EC1 is connected to the corporation's operations center, OC1, via the DMZ. OC1 is connected to the corporation's control system, SCADA1. SCADA1 possesses control of the real time unit, RTU1, and the relay controller, IED1. Listing 6.1 presents the HERMES representation of our example system in the form of a system specification. Listing 6.5 presents the template for the system specification.

6.31 EXAMPLE: ATTACK

Data exfiltration via spearphishing is the attack setting of our example, where all employees of Acme Corp.'s enterprise center receive a deceptive email. The deceptive email contains real-like content that has been tailored to catch the eye of each individual enterprise employee of Acme Corp. However, behind the real-like looking contents, there exist as payloads, malicious Trojan delivery scripts. If successful in infecting an employee's computer, the Trojan will record all the keystrokes of an employee and will exfiltrate the data to an outside data hub. The exfiltrated data can then be used to compromise an employee's authorization credentials. Once compromised, an employee's authorization credentials will allow an attacker access to Acme Corp.'s control devices and the attacker can now turn off the relay. Turning off the relay will shutdown the system and cause a disruption in electricity supply. The attack pattern of our example is a simplified version of the December 2015 Ukraine power grid cyberattack [42]. Listings 6.2 and 6.3 present two HERMES representations of our attack example in the form of an attack delta specification. Listing 6.6 presents the template for the attack delta specification.

6.32 EXAMPLE: DEFENSE

To counter attempts of data exfiltration via spearphishing, our example presents a hardening measure at the global email settings policy level. Whereas the original system has the spam filter on and it did not allow macros, it still allowed images, scripts, and hyperlinks. Our hardening measure is to prohibit all email content except for plain text. By doing so, a spearphished employee will not be able to download the Trojan payload onto their system even if they ignorantly went about clicking on all their email attachments. Listing 6.4 presents the HERMES representations of our defense example in the form of a defense delta specification. Listing 6.5 includes a template for the defense delta specification.

6.33 EXAMPLE: DELTA APPLICATION

The hardening specification that is presented in Sub-Section 6.32, can be applied to the system specification, which is presented in Sub-Section 6.30. Such an application of a hardening specification on the system specification results in a new system specification. Listing 6.7 presents the HERMES representations of the new system specification. Listing 6.5 remains the template for the new system specification.

Listing 6.1: HERMES system specification for Acme Corp.'s enterprise, operations center, control centers, and substations [3].

```

1      % Specification S
2      Acme: acme1
3      {
4          Fqn: acme1;
5          SubSystems: [ec1];
6      }.
7      EC: ec1
8      {
9          Fqn: acme1.ec1;
10         SubSystems: [emp1, wks1, dmz1, oc1];
11         Parent: acme1;
12     }.
13     EMP: emp1
14     {
15         Fqn: acme1.ec1.employees;
16         EmployeeDictionary: {eid01: "Joe Vandal", eid02: "Gold Roger", eid03: "
Silver Rayleigh", eid04: "Copper Naomi", eidall: "All employees"};
17         Parent: acme1.ec1;

```

```
18     }.
19 WKS: wks1
20     {
21     Fqn: acme1.ec1.wks1;
22     WKSList: [wks01, wks02, wks03, wks04, wks05, wks06];
23     Parent: acme1.ec1;
24     }.
25 DMZ: dmz1
26     {
27     Fqn: acme1.ec1.dmz1;
28     Subsystems: [email1, vpn1, authsrv1];
29     Parent: acme1.ec1;
30     }.
31 EMAIL: email1
32     {
33     Fqn: acme1.ec1.dmz1.email1;
34     HasEMailAccount: [eid01, eid02, eid04];
35     SpamFilter: yes;
36     AllowMacros: no;
37     AllowImages: yes;
38     AllowScripts: yes;
39     AllowHyperlinks: yes;
40     Parent: acme1.ec1.dmz1;
41     }.
42 VPN: vpn1
43     {
44     Fqn: acme1.ec1.dmz1.vpn1;
45     RemoteAccessList: [eid01, eid04];
46     Parent: acme1.ec1.dmz1;
47     }.
48 AUTHSRV: authsrv1
49     {
50     Fqn: acme1.ec1.dmz1.authsrv1;
51     ActiveDirectory: yes;
52     Parent: acme1.ec1.dmz1;
53     }.
54 OC: oc1
55     {
56     Fqn: acme1.ec1;
57     SubSystems:[scada1];
58     Parent: acme1.ec1;
```

```

59     }.
60     SCADA: scada1
61     {
62         Fqn: acme1.ec1.oc1.scada1;
63         SubSystems: [rtu1, ied1];
64         Parent: acme1.ec1.oc1;
65     }.
66     RTU: rtu1
67     {
68         Fqn: acme1.ec1.oc1.scada.rtu1;
69         RealTimeUnit: yes;
70         Parent: acme1.ec1.oc1.scada1;
71     }.
72     IED: ied1
73     {
74         Fqn: acme1.ec1.oc1.scada.ied1;
75         RelayControl: yes;
76         RelayStatus: on;
77         Parent: acme1.ec1.oc1.scada1;
78     }.
79     PolicyInternet: p1
80     {
81         Fqn: acme1.ec1.p1;
82         InternetAccess: {wks: yes, dmz: yes};
83         Parent: acme1.ec1;
84     }.
85     PolicyAuth: p2
86     {
87         Fqn: acme1.ec1.p2;
88         AuthMode: authsrv1;
89         LogOnAccess: {eid01: wks01, eid02: wks02, eid03: wks03, eid04: wks04,
90         eidall: wks05, eidall: wks06};
91         Parent: acme1.ec1;
92     }.

```

Listing 6.2: HERMES attack specification version 1 (inconsistent), specifying a spear-phishing attack on Acme Corp.'s enterprise employees.

```

1     % Specification M
2     SPEAR: spear1

```

```

3      {
4          Fqn: spear1;
5          SubSystems: [bademail1];
6          DeliveryMedium: email1;
7          Target: [eid01, eid02, eid03, eid04];
8      }.
9      BADEMAIL: bademail1
10     {
11         Fqn: spear1.bademail1;
12         SubSystems: [mask1, script1];
13         Parent: spear1;
14     }.
15     MASK: mask1
16     {
17         Fqn: spear1.bademail1.mask1;
18         AttractiveContent: yes;
19         RealisticContent: yes;
20         DeceptiveContent: yes;
21         Parent: spear1.bademail1;
22     }.
23     SCRIPT: script1
24     {
25         Fqn: spear1.bademail1.script1;
26         TrojanDelivery: yes;
27         SilentExecution: no;
28         DataExfiltration: yes;
29         RelayStatus: off;
30         Parent: spear1.bademail1;
31     }.
32

```

Listing 6.3: HERMES attack specification version 2 (consistent), specifying a spear-phishing attack on Acme Corp.’s enterprise employees.

```

1      % Specification N
2      SPEAR: spear1
3      {
4          Fqn: spear1;
5          SubSystems: [bademail1];
6          DeliveryMedium: email1;
7          Target: [eid01, eid02, eid03, eid04];

```

```

8     }.
9     BADEMAIL: bademail
10    {
11      Fqn: spear1.bademail;
12      SubSystems: [mask1, script1];
13      Parent: spear1;
14    }.
15    MASK: mask1
16    {
17      Fqn: spear1.bademail.mask1;
18      AttractiveContent: yes;
19      RealisticContent: yes;
20      DeceptiveContent: yes;
21      Parent: spear1.bademail;
22    }.
23    SCRIPT: script1
24    {
25      Fqn: spear1.bademail.script1;
26      TrojanDelivery: yes;
27      SilentExecution: yes;
28      DataExfiltration: yes;
29      RelayStatus: off;
30      Parent: spear1.bademail;
31    }.
32

```

Listing 6.4: HERMES defense specification specifying a hardening measure to counter a potential spear-phishing attack.

```

1     % Specification O
2     EMAIL: email1
3     {
4       Fqn: acme1.ec1.dmz1.email1;
5       HasEMailAccount: [eid01, eid02, eid04];
6       SpamFilter: yes;
7       AllowMacros: no;
8       AllowImages: no;
9       AllowScripts: no;
10      AllowHyperlinks: no;
11      Parent: acme1.ec1.dmz1;
12    }.

```

13

Listing 6.5: HERMES template for Acme Corp.'s enterprise, operations center, control centers, and substations [3].

```

1      % Specification T1
2      Template: Acme
3      {
4          Fqn;
5          SubSystems;
6      }.
7      Template: EC
8      {
9          Fqn;
10         SubSystems;
11         Parent;
12     }.
13     Template: EMP
14     {
15         Fqn;
16         EmployeeDictionary;
17         Parent;
18     }.
19     Template: WKS
20     {
21         Fqn;
22         WKSList;
23         Parent;
24     }.
25     Template: DMZ
26     {
27         Fqn;
28         Subsystems;
29         Parent;
30     }.
31     Tempalte: EMAIL
32     {
33         Fqn;
34         HasEMailAccount;
35         SpamFilter;
36         AllowMacros;

```

```
37     AllowImages ;
38     AllowScripts ;
39     AllowHyperlinks ;
40     Parent ;
41 }.
42 Template: VPN
43 {
44     Fqn ;
45     RemoteAccessList ;
46     Parent ;
47 }.
48 Template: AUTHSRV
49 {
50     Fqn ;
51     ActiveDirectory ;
52     Parent ;
53 }.
54 Template: OC
55 {
56     Fqn ;
57     SubSystems ;
58     Parent ;
59 }.
60 Template: SCADA
61 {
62     Fqn ;
63     SubSystems ;
64     Parent ;
65 }.
66 Template: RTU
67 {
68     Fqn ;
69     RealTimeUnit ;
70     Parent ;
71 }.
72 Template: IED
73 {
74     Fqn ;
75     RelayControl ;
76     RelayStatus ;
77     Parent ;
```



```

78     }.
79     Template: PolicyInternet
80     {
81         Fqn;
82         Description;
83         InternetAccess;
84         Parent;
85     }.
86     Template: PolicyAuth
87     {
88         Fqn;
89         AuthMode;
90         LogOnAccess;
91         Parent;
92     }.
93

```

Listing 6.6: HERMES template for a spear-phishing attack on Acme Corp.'s enterprise employees.

```

1     % Specification T2
2     Template: SPEAR
3     {
4         Fqn;
5         SubSystems;
6         DeliveryMedium;
7         Target;
8     }.
9     Template: BADEMAIL
10    {
11        Fqn;
12        SubSystems;
13        Parent;
14    }.
15    Template: MASK
16    {
17        Fqn;
18        AttractiveContent;
19        RealisticContent;
20        DeceptiveContent;
21        parent;
22    }.
23    Template: SCRIPT

```

```

24     {
25         Fqn;
26         TrojanDelivery;
27         SilentExecution;
28         DataExfiltration;
29         RelayStatus;
30         parent;
31     }.
32

```

Listing 6.7: HERMES new specification for Acme Corp.'s enterprise, operations center, control center, and substations [3].

```

1     % Specification W
2     Acme: acme1
3     {
4         Fqn: acme1;
5         SubSystems: [ec1];
6     }.
7     EC: ec1
8     {
9         Fqn: acme1.ec1;
10        SubSystems: [emp1, wks1, dmz1, oc1];
11        Parent: acme1;
12    }.
13    EMP: emp1
14    {
15        Fqn: acme1.ec1.employees;
16        EmployeeDictionary: {eid01: "Joe Vandal", eid02: "Gold Roger", eid03: "
Silver Rayleigh", eid04: "Copper Naomi", eidall: "All employees"};
17        Parent: acme1.ec1;
18    }.
19    WKS: wks1
20    {
21        Fqn: acme1.ec1.wks1;
22        WKSList: [wks01, wks02, wks03, wks04, wks05, wks06];
23        Parent: acme1.ec1;
24    }.
25    DMZ: dmz1
26    {
27        Fqn: acme1.ec1.dmz1;

```

```
28     Subsystems: [email1, vpn1, authsrv1];
29     Parent: acme1.ec1;
30 }.
31 EMAIL: email1
32 {
33     Fqn: acme1.ec1.dmz1.email1;
34     HasEMailAccount: [eid01, eid02, eid04];
35     SpamFilter: yes;
36     AllowMacros: no;
37     AllowImages: no;
38     AllowScripts: no;
39     AllowHyperlinks: no;
40     Parent: acme1.ec1.dmz1;
41 }.
42 VPN: vpn1
43 {
44     Fqn: acme1.ec1.dmz1.vpn1;
45     RemoteAccessList: [eid01, eid04];
46     Parent: acme1.ec1.dmz1;
47 }.
48 AUTHSRV: authsrv1
49 {
50     Fqn: acme1.ec1.dmz1.authsrv1;
51     ActiveDirectory: yes;
52     Parent: acme1.ec1.dmz1;
53 }.
54 OC: oc1
55 {
56     Fqn: acme1.ec1;
57     SubSystems:[scada1];
58     Parent: acme1.ec1;
59 }.
60 SCADA: scada1
61 {
62     Fqn: acme1.ec1.oc1.scada1;
63     SubSystems: [rtu1, ied1];
64     Parent: acme1.ec1.oc1;
65 }.
66 RTU: rtu1
67 {
68     Fqn: acme1.ec1.oc1.scada.rtu1;
```

```

69     RealTimeUnit: yes;
70     Parent: acme1.ec1.oc1.scada1;
71 }.
72 IED: ied1
73 {
74     Fqn: acme1.ec1.oc1.scada.ied1;
75     RelayControl: yes;
76     RelayStatus: on;
77     Parent: acme1.ec1.oc1.scada1;
78 }.
79 PolicyInternet: p1
80 {
81     Fqn: acme1.ec1.p1;
82     InternetAccess: {wks: yes, dmz: yes};
83     Parent: acme1.ec1;
84 }.
85 PolicyAuth: p2
86 {
87     Fqn: acme1.ec1.p2;
88     AuthMode: authsrv1;
89     LogOnAccess: {eid01: wks01, eid02: wks02, eid03: wks03, eid04: wks04,
eidall: wks05, eidall: wks06};
90     Parent: acme1.ec1;
91 }.
92

```

6.34 PROOF TRACING FOR HESTIA'S THEOREM OF CONSISTENCY CHECKING

To recall, HESTIA's Theorem of Consistency Checking is defined in Section 6.25.

TRACE SKETCH: From our example, we assume 'EMAIL' entity from the system specification and the 'EMAIL' entity from the corresponding template specification. We then obtain a confirmation of the theorem for the entity by making substitutions of the theorem's variables with the values from the assumed entities.

ASSUME:

1. $\{Listing\ 6.1\} \in S$
2. $\{Listing\ 6.5\} \in T$

$$3. \{email(S)\} \in E^a$$

$$4. \{email(T)\} \in E^b$$

PROVE: Prove that the system specification entity, E^a , is consistent.

PROOF BY TRACE:

$$1. (ETC(E^b, E^a)) \text{ invokes } (DSE(F^a, F^b))$$

$$2. F^a = 'email' \ \& \ F^b = 'email'$$

$$3. (DSE(F^a, F^b)) \text{ invokes } (ASE(p^a, p^b))$$

$$4. p^a = 'e' \ \& \ p^b = 'e'$$

$$5. (ASE(e, e)) = T$$

$$6. p^a = 'm' \ \& \ p^b = 'm'$$

$$7. (ASE(m, m)) = T$$

$$8. p^a = 'a' \ \& \ p^b = 'a'$$

$$9. (ASE(a, a)) = T$$

$$10. p^a = 'i' \ \& \ p^b = 'i'$$

$$11. (ASE(i, i)) = T$$

$$12. p^a = 'l' \ \& \ p^b = 'l'$$

$$13. (ASE(l, l)) = T$$

$$14. (DSE(email, email)) = T$$

$$15. (ETC(email, email)) = T$$

$$16. (FNC(F^a, F^b)) \text{ invokes } (DSE(F^a, F^b))$$

$$17. F^a = 'fqn' \ \& \ F^b = 'fqn'$$

$$18. (DSE(F^a, F^b)) \text{ invokes } (ASE(p^a, p^b))$$

$$19. p^a = 'f' \ \& \ p^b = 'f'$$

$$20. (ASE(f, f)) = T$$

$$21. p^a = 'q' \ \& \ p^b = 'q'$$

$$22. (\text{ASE}(q, q)) = \text{T}$$

$$23. p^a = 'n' \ \& \ p^b = 'n'$$

$$24. (\text{ASE}(n, n)) = \text{T}$$

$$25. (\text{DSE}(fqn, fqn)) = \text{T}$$

$$26. (\text{FNC}(fqn, fqn)) = \text{T}$$

$$27. (\text{FNC}(F^a, F^b)) \text{ invokes } (\text{DSE}(F^a, F^b))$$

$$28. F^a = 'hasemailaccount' \ \& \ F^b = 'hasemailaccount'$$

$$29. (\text{DSE}(F^a, F^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$30. p^a = 'h' \ \& \ p^b = 'h'$$

$$31. (\text{ASE}(h, h)) = \text{T}$$

$$32. p^a = 'a' \ \& \ p^b = 'a'$$

$$33. (\text{ASE}(a, a)) = \text{T}$$

$$34. p^a = 's' \ \& \ p^b = 's'$$

$$35. (\text{ASE}(s, s)) = \text{T}$$

$$36. p^a = 'e' \ \& \ p^b = 'e'$$

$$37. (\text{ASE}(e, e)) = \text{T}$$

$$38. p^a = 'm' \ \& \ p^b = 'm'$$

$$39. (\text{ASE}(m, m)) = \text{T}$$

$$40. p^a = 'a' \ \& \ p^b = 'a'$$

$$41. (\text{ASE}(a, a)) = \text{T}$$

$$42. p^a = 'i' \ \& \ p^b = 'i'$$

$$43. (\text{ASE}(i, i)) = \text{T}$$

$$44. p^a = 'l' \ \& \ p^b = 'l'$$

$$45. (\text{ASE}(l, l)) = \text{T}$$

$$46. p^a = 'a' \ \& \ p^b = 'a'$$

$$47. (\text{ASE}(a, a)) = \text{T}$$

$$48. p^a = 'c' \ \& \ p^b = 'c'$$

$$49. (\text{ASE}(c, c)) = \text{T}$$

$$50. p^a = 'c' \ \& \ p^b = 'c'$$

$$51. (\text{ASE}(c, c)) = \text{T}$$

$$52. p^a = 'o' \ \& \ p^b = 'o'$$

$$53. (\text{ASE}(o, o)) = \text{T}$$

$$54. p^a = 'u' \ \& \ p^b = 'u'$$

$$55. (\text{ASE}(u, u)) = \text{T}$$

$$56. p^a = 'n' \ \& \ p^b = 'n'$$

$$57. (\text{ASE}(n, n)) = \text{T}$$

$$58. p^a = 't' \ \& \ p^b = 't'$$

$$59. (\text{ASE}(t, t)) = \text{T}$$

$$60. (\text{DSE}(\text{hasemailaccount}, \text{hasemailaccount})) = \text{T}$$

$$61. (\text{FNC}(\text{hasemailaccount}, \text{hasemailaccount})) = \text{T}$$

$$62. (\text{FNC}(F^a, F^b)) \text{ invokes } (\text{DSE}(F^a, F^b))$$

$$63. F^a = 'spamfilter' \ \& \ F^b = 'spamfilter'$$

$$64. (\text{DSE}(F^a, F^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$65. p^a = 's' \ \& \ p^b = 's'$$

$$66. (\text{ASE}(s, s)) = \text{T}$$

$$67. p^a = 'p' \ \& \ p^b = 'p'$$

$$68. (\text{ASE}(p, p)) = \text{T}$$

$$69. p^a = 'a' \ \& \ p^b = 'a'$$

$$70. (\text{ASE}(a, a)) = \text{T}$$

$$71. p^a = 'm' \ \& \ p^b = 'm'$$

$$72. (\text{ASE}(m, m)) = \text{T}$$

$$73. p^a = 'f' \ \& \ p^b = 'f'$$

$$74. (\text{ASE}(f, f)) = \text{T}$$

$$75. p^a = 'i' \ \& \ p^b = 'i'$$

$$76. (\text{ASE}(i, i)) = \text{T}$$

$$77. p^a = 'l' \ \& \ p^b = 'l'$$

$$78. (\text{ASE}(l, l)) = \text{T}$$

$$79. p^a = 't' \ \& \ p^b = 't'$$

$$80. (\text{ASE}(t, t)) = \text{T}$$

$$81. p^a = 'e' \ \& \ p^b = 'e'$$

$$82. (\text{ASE}(e, e)) = \text{T}$$

$$83. p^a = 'r' \ \& \ p^b = 'r'$$

$$84. (\text{ASE}(r, r)) = \text{T}$$

$$85. (\text{DSE}(\text{spamfilter}, \text{spamfilter})) = \text{T}$$

$$86. (\text{FNC}(\text{spamfilter}, \text{spamfilter})) = \text{T}$$

$$87. (\text{FNC}(F^a, F^b)) \text{ invokes } (\text{DSE}(F^a, F^b))$$

$$88. F^a = 'allowmacros' \ \& \ F^b = 'allowmacros'$$

$$89. (\text{DSE}(F^a, F^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$90. p^a = 'a' \ \& \ p^b = 'a'$$

$$91. (\text{ASE}(a, a)) = \text{T}$$

$$92. p^a = 'l' \ \& \ p^b = 'l'$$

93. $(ASE(l, l)) = T$
94. $p^a = 'l' \ \& \ p^b = 'l'$
95. $(ASE(l, l)) = T$
96. $p^a = 'o' \ \& \ p^b = 'o'$
97. $(ASE(o, o)) = T$
98. $p^a = 'w' \ \& \ p^b = 'w'$
99. $(ASE(w, w)) = T$
100. $p^a = 'm' \ \& \ p^b = 'm'$
101. $(ASE(m, m)) = T$
102. $p^a = 'a' \ \& \ p^b = 'a'$
103. $(ASE(a, a)) = T$
104. $p^a = 'c' \ \& \ p^b = 'c'$
105. $(ASE(c, c)) = T$
106. $p^a = 'r' \ \& \ p^b = 'r'$
107. $(ASE(r, r)) = T$
108. $p^a = 'o' \ \& \ p^b = 'o'$
109. $(ASE(o, o)) = T$
110. $p^a = 's' \ \& \ p^b = 's'$
111. $(ASE(s, s)) = T$
112. $(DSE(allowmacros, allowmacros)) = T$
113. $(FNC(allowmacros, allowmacros)) = T$
114. $(FNC(F^a, F^b)) \text{ invokes } (DSE(F^a, F^b))$
115. $F^a = 'allowimages' \ \& \ F^b = 'allowimages'$
116. $(DSE(F^a, F^b)) \text{ invokes } (ASE(p^a, p^b))$

$$117. p^a = 'a' \ \& \ p^b = 'a'$$

$$118. (\text{ASE}(a, a)) = \text{T}$$

$$119. p^a = 'l' \ \& \ p^b = 'l'$$

$$120. (\text{ASE}(l, l)) = \text{T}$$

$$121. p^a = 'l' \ \& \ p^b = 'l'$$

$$122. (\text{ASE}(l, l)) = \text{T}$$

$$123. p^a = 'o' \ \& \ p^b = 'o'$$

$$124. (\text{ASE}(o, o)) = \text{T}$$

$$125. p^a = 'w' \ \& \ p^b = 'w'$$

$$126. (\text{ASE}(w, w)) = \text{T}$$

$$127. p^a = 'i' \ \& \ p^b = 'i'$$

$$128. (\text{ASE}(i, i)) = \text{T}$$

$$129. p^a = 'm' \ \& \ p^b = 'm'$$

$$130. (\text{ASE}(m, m)) = \text{T}$$

$$131. p^a = 'a' \ \& \ p^b = 'a'$$

$$132. (\text{ASE}(a, a)) = \text{T}$$

$$133. p^a = 'g' \ \& \ p^b = 'g'$$

$$134. (\text{ASE}(g, g)) = \text{T}$$

$$135. p^a = 'e' \ \& \ p^b = 'e'$$

$$136. (\text{ASE}(e, e)) = \text{T}$$

$$137. p^a = 's' \ \& \ p^b = 's'$$

$$138. (\text{ASE}(s, s)) = \text{T}$$

$$139. (\text{DSE}(\text{allowimages}, \text{allowimages})) = \text{T}$$

$$140. (\text{FNC}(\text{allowimages}, \text{allowimages})) = \text{T}$$

141. $(\text{FNC}(F^a, F^b)) \text{ invokes } (\text{DSE}(F^a, F^b))$

142. $F^a = \text{'allowscripts' } \& F^b = \text{'allowscripts'}$

143. $(\text{DSE}(F^a, F^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$

144. $p^a = \text{'a' } \& p^b = \text{'a'}$

145. $(\text{ASE}(a, a)) = \text{T}$

146. $p^a = \text{'l' } \& p^b = \text{'l'}$

147. $(\text{ASE}(l, l)) = \text{T}$

148. $p^a = \text{'l' } \& p^b = \text{'l'}$

149. $(\text{ASE}(l, l)) = \text{T}$

150. $p^a = \text{'o' } \& p^b = \text{'o'}$

151. $(\text{ASE}(o, o)) = \text{T}$

152. $p^a = \text{'w' } \& p^b = \text{'w'}$

153. $(\text{ASE}(w, w)) = \text{T}$

154. $p^a = \text{'s' } \& p^b = \text{'s'}$

155. $(\text{ASE}(s, s)) = \text{T}$

156. $p^a = \text{'c' } \& p^b = \text{'c'}$

157. $(\text{ASE}(c, c)) = \text{T}$

158. $p^a = \text{'r' } \& p^b = \text{'r'}$

159. $(\text{ASE}(r, r)) = \text{T}$

160. $p^a = \text{'i' } \& p^b = \text{'i'}$

161. $(\text{ASE}(i, i)) = \text{T}$

162. $p^a = \text{'p' } \& p^b = \text{'p'}$

163. $(\text{ASE}(p, p)) = \text{T}$

164. $p^a = \text{'t' } \& p^b = \text{'t'}$

$$165. (\text{ASE}(t, t)) = \text{T}$$

$$166. p^a = 's' \ \& \ p^b = 's'$$

$$167. (\text{ASE}(s, s)) = \text{T}$$

$$168. (\text{DSE}(\text{allowscripts}, \text{allowscripts})) = \text{T}$$

$$169. (\text{FNC}(\text{allowscripts}, \text{allowscripts})) = \text{T}$$

$$170. (\text{FNC}(F^a, F^b)) \text{ invokes } (\text{DSE}(F^a, F^b))$$

$$171. F^a = 'allowhyperlinks' \ \& \ F^b = 'allowhyperlinks'$$

$$172. (\text{DSE}(F^a, F^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$173. p^a = 'a' \ \& \ p^b = 'a'$$

$$174. (\text{ASE}(a, a)) = \text{T}$$

$$175. p^a = 'l' \ \& \ p^b = 'l'$$

$$176. (\text{ASE}(l, l)) = \text{T}$$

$$177. p^a = 'l' \ \& \ p^b = 'l'$$

$$178. (\text{ASE}(l, l)) = \text{T}$$

$$179. p^a = 'o' \ \& \ p^b = 'o'$$

$$180. (\text{ASE}(o, o)) = \text{T}$$

$$181. p^a = 'w' \ \& \ p^b = 'w'$$

$$182. (\text{ASE}(w, w)) = \text{T}$$

$$183. p^a = 'h' \ \& \ p^b = 'h'$$

$$184. (\text{ASE}(h, h)) = \text{T}$$

$$185. p^a = 'y' \ \& \ p^b = 'y'$$

$$186. (\text{ASE}(y, y)) = \text{T}$$

$$187. p^a = 'p' \ \& \ p^b = 'p'$$

$$188. (\text{ASE}(p, p)) = \text{T}$$

$$189. p^a = 'e' \ \&\ \& p^b = 'e'$$

$$190. (\text{ASE}(e, e)) = \text{T}$$

$$191. p^a = 'r' \ \&\ \& p^b = 'r'$$

$$192. (\text{ASE}(r, r)) = \text{T}$$

$$193. p^a = 'l' \ \&\ \& p^b = 'l'$$

$$194. (\text{ASE}(l, l)) = \text{T}$$

$$195. p^a = 'i' \ \&\ \& p^b = 'i'$$

$$196. (\text{ASE}(i, i)) = \text{T}$$

$$197. p^a = 'n' \ \&\ \& p^b = 'n'$$

$$198. (\text{ASE}(n, n)) = \text{T}$$

$$199. p^a = 'k' \ \&\ \& p^b = 'k'$$

$$200. (\text{ASE}(k, k)) = \text{T}$$

$$201. p^a = 's' \ \&\ \& p^b = 's'$$

$$202. (\text{ASE}(s, s)) = \text{T}$$

$$203. (\text{DSE}(\text{allowhyperlinks}, \text{allowhyperlinks})) = \text{T}$$

$$204. (\text{FNC}(\text{allowhyperlinks}, \text{allowhyperlinks})) = \text{T}$$

$$205. (\text{FNC}(F^a, F^b)) \text{ invokes } (\text{DSE}(F^a, F^b))$$

$$206. F^a = 'parent' \ \&\ \& F^b = 'parent'$$

$$207. (\text{DSE}(F^a, F^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$208. p^a = 'p' \ \&\ \& p^b = 'p'$$

$$209. (\text{ASE}(p, p)) = \text{T}$$

$$210. p^a = 'a' \ \&\ \& p^b = 'a'$$

$$211. (\text{ASE}(a, a)) = \text{T}$$

$$212. p^a = 'r' \ \&\ \& p^b = 'r'$$

$$213. (\text{ASE}(r, r)) = \text{T}$$

$$214. p^a = 'e' \ \& \ p^b = 'e'$$

$$215. (\text{ASE}(e, e)) = \text{T}$$

$$216. p^a = 'n' \ \& \ p^b = 'n'$$

$$217. (\text{ASE}(n, n)) = \text{T}$$

$$218. p^a = 't' \ \& \ p^b = 't'$$

$$219. (\text{ASE}(t, t)) = \text{T}$$

$$220. (\text{DSE}(\text{parent}, \text{parent})) = \text{T}$$

$$221. (\text{FNC}(\text{parent}, \text{parent})) = \text{T}$$

$$222. \propto (E^a, E^b)$$

QuodEratDemonstrandum(Q.E.D.)

By iterative extrapolation of the proof on all entities of S , we can declare that $\propto (S)$. Since Listing 6.1 was assumed to be S , we can consider that Listing 6.1 is consistent. Similarly, we can also prove that Listings 6.2 and 6.3 are consistent as well.

6.35 PROOF TRACING FOR HESTIA'S THEOREM OF CONFLICT CHECKING

To recall, HESTIA's Theorem of Conflict Checking is defined in section 6.26.

TRACE SKETCH: From our example, we see that there exist two versions of the attack policy specifications. We assume the 'script1' entity from each version of the attack policy specifications. We then obtain a confirmation of the theorem for the entity by making substitutions of the theorem's variables with the values from the assumed entities.

ASSUME:

$$1. \{\text{Listing 6.2}\} \in M$$

$$2. \{\text{Listing 6.3}\} \in N$$

$$3. \{\text{script1}(M)\} \in E^a$$

$$4. \{script1(N)\} \in E^b$$

PROVE: The attack specification version 1 entity, E^a , is in conflict with the attack specification version 2 entity, E^b .

PROOF BY TRACE:

$$1. (ETC(E^a, E^b)) \text{ invokes } (DSE(F^a, F^b))$$

$$2. F^a = 'script' \ \& \ F^b = 'script'$$

$$3. (DSE(F^a, F^b)) \text{ invokes } (ASE(p^a, p^b))$$

$$4. p^a = 's' \ \& \ p^b = 's'$$

$$5. (ASE(s, s)) = T$$

$$6. p^a = 'c' \ \& \ p^b = 'c'$$

$$7. (ASE(c, c)) = T$$

$$8. p^a = 'r' \ \& \ p^b = 'r'$$

$$9. (ASE(r, r)) = T$$

$$10. p^a = 'i' \ \& \ p^b = 'i'$$

$$11. (ASE(i, i)) = T$$

$$12. p^a = 'p' \ \& \ p^b = 'p'$$

$$13. (ASE(p, p)) = T$$

$$14. p^a = 't' \ \& \ p^b = 't'$$

$$15. (ASE(t, t)) = T$$

$$16. (DSE(script, script)) = T$$

$$17. (ETC(script, script)) = T$$

$$18. (EIC(E^a, E^b)) \text{ invokes } (DSE(F^a, F^b))$$

$$19. F^a = 'script1' \ \& \ F^b = 'script1'$$

$$20. (DSE(F^a, F^b)) \text{ invokes } (ASE(p^a, p^b))$$

$$21. p^a = 's' \ \& \ p^b = 's'$$

$$22. (\text{ASE}(s, s)) = \text{T}$$

$$23. p^a = 'c' \ \& \ p^b = 'c'$$

$$24. (\text{ASE}(c, c)) = \text{T}$$

$$25. p^a = 'r' \ \& \ p^b = 'r'$$

$$26. (\text{ASE}(r, r)) = \text{T}$$

$$27. p^a = 'i' \ \& \ p^b = 'i'$$

$$28. (\text{ASE}(i, i)) = \text{T}$$

$$29. p^a = 'p' \ \& \ p^b = 'p'$$

$$30. (\text{ASE}(p, p)) = \text{T}$$

$$31. p^a = 't' \ \& \ p^b = 't'$$

$$32. (\text{ASE}(t, t)) = \text{T}$$

$$33. p^a = '1' \ \& \ p^b = '1'$$

$$34. (\text{ASE}(1, 1)) = \text{T}$$

$$35. (\text{DSE}(\text{script1}, \text{script1})) = \text{T}$$

$$36. (\text{EIC}(\text{script1}, \text{script1})) = \text{T}$$

$$37. (\text{FNC}(F^a, F^b)) \text{ invokes } (\text{DSE}(F^a, F^b))$$

$$38. F^a = 'fqm' \ \& \ F^b = 'fqm'$$

$$39. (\text{DSE}(F^a, F^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$40. p^a = 'f' \ \& \ p^b = 'f'$$

$$41. (\text{ASE}(f, f)) = \text{T}$$

$$42. p^a = 'q' \ \& \ p^b = 'q'$$

$$43. (\text{ASE}(q, q)) = \text{T}$$

$$44. p^a = 'n' \ \& \ p^b = 'n'$$

$$45. (\text{ASE}(n, n)) = \text{T}$$

$$46. (\text{DSE}(fqn, fqn)) = \text{T}$$

$$47. (\text{FNC}(fqn, fqn)) = \text{T}$$

$$48. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{SC}(R^a, R^b))$$

$$49. (\text{SC}(R^a, R^b)) \text{ invokes } (\text{SD}(R^a) \wedge \text{SD}(R^b))$$

$$50. R^a = \text{'spear1.bademai1.script1'} \ \& \ R^b = \text{'spear1.bademai1.script1'}$$

$$51. \text{SD}(R^a) = 3$$

$$52. \text{SD}(R^b) = 3$$

$$53. \text{SD}(R^a) = \text{SD}(R^b)$$

$$54. (\text{SC}(R^a, R^b)) = \text{E}$$

$$55. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{FRTC}(R^a, R^b))$$

$$56. (\text{FRTC}(R^a, R^b)) \text{ invokes } (\text{FRTD}(R^a) \wedge \text{FRTD}(R^b))$$

$$57. R^a \in \mathcal{Q}$$

$$58. \text{FRTD}(R^a) = \text{G}$$

$$59. R^b \in \mathcal{Q}$$

$$60. \text{FRTD}(R^b) = \text{G}$$

$$61. \text{FRTC}(R^a, R^b) = \text{T}$$

$$62. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{DSE}(R^a, R^b))$$

$$63. R^a = \text{'spear1.bademai1.script1'} \ \& \ R^b = \text{'spear1.bademai1.script1'}$$

$$64. (\text{DSE}(R^a, R^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$65. p^a = \text{'s'} \ \& \ p^b = \text{'s'}$$

$$66. (\text{ASE}(s, s)) = \text{T}$$

$$67. p^a = \text{'p'} \ \& \ p^b = \text{'p'}$$

$$68. (\text{ASE}(p, p)) = \text{T}$$

$$69. p^a = 'e' \ \& \ p^b = 'e'$$

$$70. (\text{ASE}(e, e)) = \text{T}$$

$$71. p^a = 'a' \ \& \ p^b = 'a'$$

$$72. (\text{ASE}(a, a)) = \text{T}$$

$$73. p^a = 'r' \ \& \ p^b = 'r'$$

$$74. (\text{ASE}(r, r)) = \text{T}$$

$$75. p^a = '1' \ \& \ p^b = '1'$$

$$76. (\text{ASE}(1, 1)) = \text{T}$$

$$77. p^a = 'b' \ \& \ p^b = 'b'$$

$$78. (\text{ASE}(b, b)) = \text{T}$$

$$79. p^a = 'a' \ \& \ p^b = 'a'$$

$$80. (\text{ASE}(a, a)) = \text{T}$$

$$81. p^a = 'd' \ \& \ p^b = 'd'$$

$$82. (\text{ASE}(d, d)) = \text{T}$$

$$83. p^a = 'e' \ \& \ p^b = 'e'$$

$$84. (\text{ASE}(e, e)) = \text{T}$$

$$85. p^a = 'm' \ \& \ p^b = 'm'$$

$$86. (\text{ASE}(m, m)) = \text{T}$$

$$87. p^a = 'a' \ \& \ p^b = 'a'$$

$$88. (\text{ASE}(a, a)) = \text{T}$$

$$89. p^a = 'i' \ \& \ p^b = 'i'$$

$$90. (\text{ASE}(i, i)) = \text{T}$$

$$91. p^a = 'l' \ \& \ p^b = 'l'$$

$$92. (\text{ASE}(l, l)) = \text{T}$$

$$93. p^a = '1' \ \& \ p^b = '1'$$

$$94. (\text{ASE}(1, 1)) = \text{T}$$

$$95. p^a = 's' \ \& \ p^b = 's'$$

$$96. (\text{ASE}(s, s)) = \text{T}$$

$$97. p^a = 'c' \ \& \ p^b = 'c'$$

$$98. (\text{ASE}(c, c)) = \text{T}$$

$$99. p^a = 'r' \ \& \ p^b = 'r'$$

$$100. (\text{ASE}(r, r)) = \text{T}$$

$$101. p^a = 'i' \ \& \ p^b = 'i'$$

$$102. (\text{ASE}(i, i)) = \text{T}$$

$$103. p^a = 'p' \ \& \ p^b = 'p'$$

$$104. (\text{ASE}(p, p)) = \text{T}$$

$$105. p^a = 't' \ \& \ p^b = 't'$$

$$106. (\text{ASE}(t, t)) = \text{T}$$

$$107. p^a = '1' \ \& \ p^b = '1'$$

$$108. (\text{ASE}(1, 1)) = \text{T}$$

$$109. (\text{DSE}(\text{spearl1.bademai1.script1}, \text{spearl1.bademai1.script1})) = \text{T}$$

$$110. (\text{FRC}(\text{spearl1.bademai1.script1}, \text{spearl1.bademai1.script1})) = \text{T}$$

$$111. (\text{FNC}(F^a, F^b)) \text{ invokes } (\text{DSE}(F^a, F^b))$$

$$112. F^a = 'trojandelivery' \ \& \ F^b = 'trojandelivery'$$

$$113. (\text{DSE}(F^a, F^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$114. p^a = 't' \ \& \ p^b = 't'$$

$$115. (\text{ASE}(t, t)) = \text{T}$$

$$116. p^a = 'r' \ \& \ p^b = 'r'$$

117. $(ASE(r, r)) = T$

118. $p^a = 'o' \ \& \ p^b = 'o'$

119. $(ASE(o, o)) = T$

120. $p^a = 'j' \ \& \ p^b = 'j'$

121. $(ASE(j, j)) = T$

122. $p^a = 'a' \ \& \ p^b = 'a'$

123. $(ASE(a, a)) = T$

124. $p^a = 'n' \ \& \ p^b = 'n'$

125. $(ASE(n, n)) = T$

126. $p^a = 'd' \ \& \ p^b = 'd'$

127. $(ASE(d, d)) = T$

128. $p^a = 'e' \ \& \ p^b = 'e'$

129. $(ASE(e, e)) = T$

130. $p^a = 'l' \ \& \ p^b = 'l'$

131. $(ASE(l, l)) = T$

132. $p^a = 'i' \ \& \ p^b = 'i'$

133. $(ASE(i, i)) = T$

134. $p^a = 'v' \ \& \ p^b = 'v'$

135. $(ASE(v, v)) = T$

136. $p^a = 'e' \ \& \ p^b = 'e'$

137. $(ASE(e, e)) = T$

138. $p^a = 'r' \ \& \ p^b = 'r'$

139. $(ASE(r, r)) = T$

140. $p^a = 'y' \ \& \ p^b = 'y'$

141. $(ASE(y, y)) = T$
142. $(DSE(fqn, fqn)) = T$
143. $(FNC(fqn, fqn)) = T$
144. $(FRC((FR)^a, (FR)^b))$ invokes $(SC(R^a, R^b))$
145. $(SC(R^a, R^b))$ invokes $(SD(R^a) \wedge SD(R^b))$
146. $R^a = 'yes' \ \& \ R^b = 'yes'$
147. $SD(R^a) = 1$
148. $SD(R^b) = 1$
149. $SD(R^a) = SD(R^b)$
150. $(SC(R^a, R^b)) = E$
151. $(FRC((FR)^a, (FR)^b))$ invokes $(FRTC(R^a, R^b))$
152. $(FRTC(R^a, R^b))$ invokes $(FRTD(R^a) \wedge FRTD(R^b))$
153. $R^a \in \mathcal{V}$
154. $FRTD(R^a) = U$
155. $R^b \in \mathcal{V}$
156. $FRTD(R^b) = U$
157. $FRTC(R^a, R^b) = T$
158. $(FRC((FR)^a, (FR)^b))$ invokes $(DSE(R^a, R^b))$
159. $R^a = 'yes' \ \& \ R^b = 'yes'$
160. $(DSE(R^a, R^b))$ invokes $(ASE(p^a, p^b))$
161. $p^a = 'y' \ \& \ p^b = 'y'$
162. $(ASE(y, y)) = T$
163. $p^a = 'e' \ \& \ p^b = 'e'$
164. $(ASE(e, e)) = T$

165. $p^a = 's' \ \& \ p^b = 's'$

166. $(ASE(s, s)) = T$

167. $(DSE(yes, yes)) = T$

168. $(FRC(yes, yes)) = T$

169. $(FNC(F^a, F^b)) \text{ invokes } (DSE(F^a, F^b))$

170. $F^a = 'silentexecution' \ \& \ F^b = 'silentexecution'$

171. $(DSE(F^a, F^b)) \text{ invokes } (ASE(p^a, p^b))$

172. $p^a = 's' \ \& \ p^b = 's'$

173. $(ASE(s, s)) = T$

174. $p^a = 'i' \ \& \ p^b = 'i'$

175. $(ASE(i, i)) = T$

176. $p^a = 'l' \ \& \ p^b = 'l'$

177. $(ASE(l, l)) = T$

178. $p^a = 'e' \ \& \ p^b = 'e'$

179. $(ASE(e, e)) = T$

180. $p^a = 'n' \ \& \ p^b = 'n'$

181. $(ASE(n, n)) = T$

182. $p^a = 't' \ \& \ p^b = 't'$

183. $(ASE(t, t)) = T$

184. $p^a = 'e' \ \& \ p^b = 'e'$

185. $(ASE(e, e)) = T$

186. $p^a = 'x' \ \& \ p^b = 'x'$

187. $(ASE(x, x)) = T$

188. $p^a = 'e' \ \& \ p^b = 'e'$

189. $(ASE(e, e)) = T$
190. $p^a = 'c' \ \& \ p^b = 'c'$
191. $(ASE(c, c)) = T$
192. $p^a = 'u' \ \& \ p^b = 'u'$
193. $(ASE(u, u)) = T$
194. $p^a = 't' \ \& \ p^b = 't'$
195. $(ASE(t, y)) = T$
196. $p^a = 'i' \ \& \ p^b = 'i'$
197. $(ASE(i, i)) = T$
198. $p^a = 'o' \ \& \ p^b = 'o'$
199. $(ASE(o, o)) = T$
200. $p^a = 'n' \ \& \ p^b = 'n'$
201. $(ASE(n, n)) = T$
202. $(DSE(silentexecution, silentexecution)) = T$
203. $(FNC(silentexecution, silentexecution)) = T$
204. $(FRC((FR)^a, (FR)^b)) \text{ invokes } (SC(R^a, R^b))$
205. $(SC(R^a, R^b)) \text{ invokes } (SD(R^a) \wedge SD(R^b))$
206. $R^a = 'yes' \ \& \ R^b = 'no'$
207. $SD(R^a) = 1$
208. $SD(R^b) = 1$
209. $SD(R^a) = SD(R^b)$
210. $(SC(R^a, R^b)) = E$
211. $(FRC((FR)^a, (FR)^b)) \text{ invokes } (FRTC(R^a, R^b))$
212. $(FRTC(R^a, R^b)) \text{ invokes } (FRTD(R^a) \wedge FRTD(R^b))$

213. $R^a \in \mathcal{V}$
214. $\text{FRTD}(R^a) = \text{U}$
215. $R^b \in \mathcal{V}$
216. $\text{FRTD}(R^b) = \text{U}$
217. $\text{FRTC}(R^a, R^b) = \text{T}$
218. $(\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{DSE}(R^a, R^b))$
219. $R^a = \text{'yes' } \& R^b = \text{'no'}$
220. $(\text{DSE}(R^a, R^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$
221. $p^a = \text{'y' } \& p^b = \text{'n'}$
222. $(\text{ASE}(y, y)) = \text{F}$
223. $p^a = \text{'e' } \& p^b = \text{'o'}$
224. $(\text{ASE}(e, e)) = \text{F}$
225. $p^a = \text{'s' } \& p^b = \text{'$
226. $(\text{ASE}(s, s)) = \text{F}$
227. $(\text{DSE}(\text{yes}, \text{no})) = \text{F}$
228. $(\text{FRC}(\text{yes}, \text{no})) = \text{F}$
229. $(\text{FNC}(F^a, F^b)) \text{ invokes } (\text{DSE}(F^a, F^b))$
230. $F^a = \text{'dataexfiltration' } \& F^b = \text{'dataexfiltration'}$
231. $(\text{DSE}(F^a, F^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$
232. $p^a = \text{'d' } \& p^b = \text{'d'}$
233. $(\text{ASE}(d, d)) = \text{T}$
234. $p^a = \text{'a' } \& p^b = \text{'a'}$
235. $(\text{ASE}(a, a)) = \text{T}$
236. $p^a = \text{'t' } \& p^b = \text{'t'}$

$$237. (\text{ASE}(t, t)) = \text{T}$$

$$238. p^a = 'a' \ \& \ p^b = 'a'$$

$$239. (\text{ASE}(a, a)) = \text{T}$$

$$240. p^a = 'e' \ \& \ p^b = 'e'$$

$$241. (\text{ASE}(e, e)) = \text{T}$$

$$242. p^a = 'x' \ \& \ p^b = 'x'$$

$$243. (\text{ASE}(x, x)) = \text{T}$$

$$244. p^a = 'f' \ \& \ p^b = 'f'$$

$$245. (\text{ASE}(f, f)) = \text{T}$$

$$246. p^a = 'i' \ \& \ p^b = 'i'$$

$$247. (\text{ASE}(i, i)) = \text{T}$$

$$248. p^a = 'l' \ \& \ p^b = 'l'$$

$$249. (\text{ASE}(l, l)) = \text{T}$$

$$250. p^a = 't' \ \& \ p^b = 't'$$

$$251. (\text{ASE}(t, t)) = \text{T}$$

$$252. p^a = 'r' \ \& \ p^b = 'r'$$

$$253. (\text{ASE}(r, r)) = \text{T}$$

$$254. p^a = 'a' \ \& \ p^b = 'a'$$

$$255. (\text{ASE}(a, a)) = \text{T}$$

$$256. p^a = 't' \ \& \ p^b = 't'$$

$$257. (\text{ASE}(t, t)) = \text{T}$$

$$258. p^a = 'i' \ \& \ p^b = 'i'$$

$$259. (\text{ASE}(i, i)) = \text{T}$$

$$260. p^a = 'o' \ \& \ p^b = 'o'$$

$$261. (\text{ASE}(o, o)) = \text{T}$$

$$262. p^a = 'n' \ \& \ p^b = 'n'$$

$$263. (\text{ASE}(n, n)) = \text{T}$$

$$264. (\text{DSE}(\text{dataexfiltration}, \text{dataexfiltration})) = \text{T}$$

$$265. (\text{FNC}(\text{dataexfiltration}, \text{dataexfiltration})) = \text{T}$$

$$266. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{SC}(R^a, R^b))$$

$$267. (\text{SC}(R^a, R^b)) \text{ invokes } (\text{SD}(R^a) \wedge \text{SD}(R^b))$$

$$268. R^a = 'yes' \ \& \ R^b = 'yes'$$

$$269. \text{SD}(R^a) = 1$$

$$270. \text{SD}(R^b) = 1$$

$$271. \text{SD}(R^a) = \text{SD}(R^b)$$

$$272. (\text{SC}(R^a, R^b)) = \text{E}$$

$$273. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{FRTC}(R^a, R^b))$$

$$274. (\text{FRTC}(R^a, R^b)) \text{ invokes } (\text{FRTD}(R^a) \wedge \text{FRTD}(R^b))$$

$$275. R^a \in \mathcal{V}$$

$$276. \text{FRTD}(R^a) = \text{U}$$

$$277. R^b \in \mathcal{V}$$

$$278. \text{FRTD}(R^b) = \text{U}$$

$$279. \text{FRTC}(R^a, R^b) = \text{T}$$

$$280. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{DSE}(R^a, R^b))$$

$$281. R^a = 'yes' \ \& \ R^b = 'yes'$$

$$282. (\text{DSE}(R^a, R^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$283. p^a = 'y' \ \& \ p^b = 'y'$$

$$284. (\text{ASE}(y, y)) = \text{T}$$

$$285. p^a = 'e' \ \& \ p^b = 'e'$$

$$286. (\text{ASE}(e, e)) = \text{T}$$

$$287. p^a = 's' \ \& \ p^b = 's'$$

$$288. (\text{ASE}(s, s)) = \text{T}$$

$$289. (\text{DSE}(\text{yes}, \text{yes})) = \text{T}$$

$$290. (\text{FRC}(\text{yes}, \text{yes})) = \text{T}$$

$$291. (\text{FNC}(F^a, F^b)) \text{ invokes } (\text{DSE}(F^a, F^b))$$

$$292. F^a = 'relaystatus' \ \& \ F^b = 'relaystatus'$$

$$293. (\text{DSE}(F^a, F^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$294. p^a = 'r' \ \& \ p^b = 'r'$$

$$295. (\text{ASE}(r, r)) = \text{T}$$

$$296. p^a = 'e' \ \& \ p^b = 'e'$$

$$297. (\text{ASE}(e, e)) = \text{T}$$

$$298. p^a = 'l' \ \& \ p^b = 'l'$$

$$299. (\text{ASE}(l, l)) = \text{T}$$

$$300. p^a = 'a' \ \& \ p^b = 'a'$$

$$301. (\text{ASE}(a, a)) = \text{T}$$

$$302. p^a = 'y' \ \& \ p^b = 'y'$$

$$303. (\text{ASE}(y, y)) = \text{T}$$

$$304. p^a = 's' \ \& \ p^b = 's'$$

$$305. (\text{ASE}(s, s)) = \text{T}$$

$$306. p^a = 't' \ \& \ p^b = 't'$$

$$307. (\text{ASE}(t, t)) = \text{T}$$

$$308. p^a = 'a' \ \& \ p^b = 'a'$$

$$309. (\text{ASE}(a, a)) = \text{T}$$

$$310. p^a = 't' \ \& \ p^b = 't'$$

$$311. (\text{ASE}(t, t)) = \text{T}$$

$$312. p^a = 'u' \ \& \ p^b = 'u'$$

$$313. (\text{ASE}(u, u)) = \text{T}$$

$$314. p^a = 's' \ \& \ p^b = 's'$$

$$315. (\text{ASE}(s, s)) = \text{T}$$

$$316. (\text{DSE}(\text{relaystatus}, \text{relaystatus})) = \text{T}$$

$$317. (\text{FNC}(\text{relaystatus}, \text{relaystatus})) = \text{T}$$

$$318. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{SC}(R^a, R^b))$$

$$319. (\text{SC}(R^a, R^b)) \text{ invokes } (\text{SD}(R^a) \wedge \text{SD}(R^b))$$

$$320. R^a = 'yes' \ \& \ R^b = 'yes'$$

$$321. \text{SD}(R^a) = 1$$

$$322. \text{SD}(R^b) = 1$$

$$323. \text{SD}(R^a) = \text{SD}(R^b)$$

$$324. (\text{SC}(R^a, R^b)) = \text{E}$$

$$325. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{FRTC}(R^a, R^b))$$

$$326. (\text{FRTC}(R^a, R^b)) \text{ invokes } (\text{FRTD}(R^a) \wedge \text{FRTD}(R^b))$$

$$327. R^a \in \mathcal{V}$$

$$328. \text{FRTD}(R^a) = \text{U}$$

$$329. R^b \in \mathcal{V}$$

$$330. \text{FRTD}(R^b) = \text{U}$$

$$331. \text{FRTC}(R^a, R^b) = \text{T}$$

$$332. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{DSE}(R^a, R^b))$$

333. $R^a = \text{'yes'}$ & $R^b = \text{'yes'}$
334. $(DSE(R^a, R^b))$ invokes $(ASE(p^a, p^b))$
335. $p^a = \text{'y'}$ & $p^b = \text{'y'}$
336. $(ASE(y, y)) = T$
337. $p^a = \text{'e'}$ & $p^b = \text{'e'}$
338. $(ASE(e, e)) = T$
339. $p^a = \text{'s'}$ & $p^b = \text{'s'}$
340. $(ASE(s, s)) = T$
341. $(DSE(yes, yes)) = T$
342. $(FRC(yes, yes)) = T$
343. $(FNC(F^a, F^b))$ invokes $(DSE(F^a, F^b))$
344. $F^a = \text{'parent'}$ & $F^b = \text{'parent'}$
345. $(DSE(F^a, F^b))$ invokes $(ASE(p^a, p^b))$
346. $p^a = \text{'p'}$ & $p^b = \text{'p'}$
347. $(ASE(p, p)) = T$
348. $p^a = \text{'a'}$ & $p^b = \text{'a'}$
349. $(ASE(a, a)) = T$
350. $p^a = \text{'r'}$ & $p^b = \text{'r'}$
351. $(ASE(r, r)) = T$
352. $p^a = \text{'e'}$ & $p^b = \text{'e'}$
353. $(ASE(e, e)) = T$
354. $p^a = \text{'n'}$ & $p^b = \text{'n'}$
355. $(ASE(n, n)) = T$
356. $p^a = \text{'t'}$ & $p^b = \text{'t'}$

$$357. (\text{ASE}(t, t)) = \text{T}$$

$$358. (\text{DSE}(\text{parent}, \text{parent})) = \text{T}$$

$$359. (\text{FNC}(\text{parent}, \text{parent})) = \text{T}$$

$$360. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{SC}(R^a, R^b))$$

$$361. (\text{SC}(R^a, R^b)) \text{ invokes } (\text{SD}(R^a) \wedge \text{SD}(R^b))$$

$$362. R^a = \text{'spear1.bademai1'} \ \& \ R^b = \text{'spear1.bademai1'}$$

$$363. \text{SD}(R^a) = 2$$

$$364. \text{SD}(R^b) = 2$$

$$365. \text{SD}(R^a) = \text{SD}(R^b)$$

$$366. (\text{SC}(R^a, R^b)) = \text{E}$$

$$367. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{FRTC}(R^a, R^b))$$

$$368. (\text{FRTC}(R^a, R^b)) \text{ invokes } (\text{FRTD}(R^a) \wedge \text{FRTD}(R^b))$$

$$369. R^a \in \mathcal{Q}$$

$$370. \text{FRTD}(R^a) = \text{G}$$

$$371. R^b \in \mathcal{Q}$$

$$372. \text{FRTD}(R^b) = \text{G}$$

$$373. \text{FRTC}(R^a, R^b) = \text{T}$$

$$374. (\text{FRC}((FR)^a, (FR)^b)) \text{ invokes } (\text{DSE}(R^a, R^b))$$

$$375. R^a = \text{'spear1.bademai1'} \ \& \ R^b = \text{'spear1.bademai1'}$$

$$376. (\text{DSE}(R^a, R^b)) \text{ invokes } (\text{ASE}(p^a, p^b))$$

$$377. p^a = \text{'s'} \ \& \ p^b = \text{'s'}$$

$$378. (\text{ASE}(s, s)) = \text{T}$$

$$379. p^a = \text{'p'} \ \& \ p^b = \text{'p'}$$

$$380. (\text{ASE}(p, p)) = \text{T}$$

$$381. p^a = 'e' \ \& \ p^b = 'e'$$

$$382. (\text{ASE}(e, e)) = \text{T}$$

$$383. p^a = 'a' \ \& \ p^b = 'a'$$

$$384. (\text{ASE}(a, a)) = \text{T}$$

$$385. p^a = 'r' \ \& \ p^b = 'r'$$

$$386. (\text{ASE}(r, r)) = \text{T}$$

$$387. p^a = '1' \ \& \ p^b = '1'$$

$$388. (\text{ASE}(1, 1)) = \text{T}$$

$$389. p^a = 'b' \ \& \ p^b = 'b'$$

$$390. (\text{ASE}(b, b)) = \text{T}$$

$$391. p^a = 'a' \ \& \ p^b = 'a'$$

$$392. (\text{ASE}(a, a)) = \text{T}$$

$$393. p^a = 'd' \ \& \ p^b = 'd'$$

$$394. (\text{ASE}(d, d)) = \text{T}$$

$$395. p^a = 'e' \ \& \ p^b = 'e'$$

$$396. (\text{ASE}(e, e)) = \text{T}$$

$$397. p^a = 'm' \ \& \ p^b = 'm'$$

$$398. (\text{ASE}(m, m)) = \text{T}$$

$$399. p^a = 'a' \ \& \ p^b = 'a'$$

$$400. (\text{ASE}(a, a)) = \text{T}$$

$$401. p^a = 'i' \ \& \ p^b = 'i'$$

$$402. (\text{ASE}(i, i)) = \text{T}$$

$$403. p^a = 'l' \ \& \ p^b = 'l'$$

$$404. (\text{ASE}(l, l)) = \text{T}$$

$$405. p^a = '1' \ \& \ p^b = '1'$$

$$406. (\text{ASE}(1, 1)) = \text{T}$$

$$407. (\text{DSE}(\text{spear1.bademail}, \text{spear1.bademail})) = \text{T}$$

$$408. (\text{FRC}(\text{spear1.bademail}, \text{spear1.bademail})) = \text{T}$$

$$409. \exists (E^a \otimes E^b)$$

QuodEratDemonstrandum(Q.E.D.)

By iterative extrapolation of the proof on all entities of M and N , we find that there exist no other conflicts between M and N . However, we declare that there exists a conflict between M and N , since there exists at least one conflict between the entities of M and N .

6.36 PROOF TRACING FOR HESTIA'S THEOREM OF APPLICABILITY CHECKING

To recall, HESTIA's Theorem of Consistency Checking is defined in Section 6.27.

TRACE SKETCH: From our example, we see that there exists a defense policy to counter a potential spear-phishing attack on the system specification. We assume the 'email1' entity from the defense and system specifications. We then obtain a confirmation of the theorem for the entity by making substitutions of the theorem's variables with the values from the assumed entities.

ASSUME:

1. $\{\text{Listing 6.1}\} \in S$
2. $\{\text{Listing 6.4}\} \in O$
3. $\{\text{script1}(S)\} \in E^a$
4. $\{\text{script1}(O)\} \in E^b$

PROVE: The defense specification entity, E^b , is applicable to the system specification entity, E^a .

PROOF BY TRACE:

1. $(\text{ETC}(E^b, E^a)) \text{ invokes } (\text{DSE}(F^b, F^a))$
2. $F^b = 'email' \ \& \ F^a = 'email'$

3. $(DSE(F^b, F^a))$ invokes $(ASE(p^b, p^a))$
 4. $p^b = 'e'$ & $p^a = 'e'$
 5. $(ASE(e, e)) = T$
 6. $p^b = 'm'$ & $p^a = 'm'$
 7. $(ASE(m, m)) = T$
 8. $p^b = 'a'$ & $p^a = 'a'$
 9. $(ASE(a, a)) = T$
 10. $p^b = 'i'$ & $p^a = 'i'$
 11. $(ASE(i, i)) = T$
 12. $p^b = 'l'$ & $p^a = 'l'$
 13. $(ASE(l, l)) = T$
14. $(DSE(email, email)) = T$
15. $(ETC(email, email)) = T$
16. $(EIC(E^b, E^a))$ invokes $(DSE(F^b, F^a))$
 17. $F^b = 'email1'$ & $F^a = 'email1'$
18. $(DSE(F^b, F^a))$ invokes $(ASE(p^b, p^a))$
 19. $p^b = 'e'$ & $p^a = 'e'$
 20. $(ASE(e, e)) = T$
 21. $p^b = 'm'$ & $p^a = 'm'$
 22. $(ASE(m, m)) = T$
 23. $p^b = 'a'$ & $p^a = 'a'$
 24. $(ASE(a, a)) = T$
 25. $p^b = 'i'$ & $p^a = 'i'$
 26. $(ASE(i, i)) = T$

$$27. p^b = 'l' \ \& \ p^a = 'l'$$

$$28. (\text{ASE}(l, l)) = \text{T}$$

$$29. p^b = '1' \ \& \ p^a = '1'$$

$$30. (\text{ASE}(l, l)) = \text{T}$$

$$31. (\text{DSE}(\text{email1}, \text{email1})) = \text{T}$$

$$32. (\text{EIC}(\text{email1}, \text{email1})) = \text{T}$$

$$33. (\text{FNC}(F^b, F^a)) \text{ invokes } (\text{DSE}(F^b, F^a))$$

$$34. F^b = 'fqn' \ \& \ F^a = 'fqn'$$

$$35. (\text{DSE}(F^b, F^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$$

$$36. p^b = 'f' \ \& \ p^a = 'f'$$

$$37. (\text{ASE}(f, f)) = \text{T}$$

$$38. p^b = 'q' \ \& \ p^a = 'q'$$

$$39. (\text{ASE}(q, q)) = \text{T}$$

$$40. p^b = 'n' \ \& \ p^a = 'n'$$

$$41. (\text{ASE}(n, n)) = \text{T}$$

$$42. (\text{DSE}(fqn, fqn)) = \text{T}$$

$$43. (\text{FNC}(fqn, fqn)) = \text{T}$$

$$44. (\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{FRTC}(R^b, R^a))$$

$$45. (\text{FRTC}(R^b, R^a)) \text{ invokes } (\text{FRTD}(R^b) \wedge \text{FRTD}(R^a))$$

$$46. R^b \in \mathcal{Q}$$

$$47. \text{FRTD}(R^b) = \text{G}$$

$$48. R^a \in \mathcal{Q}$$

$$49. \text{FRTD}(R^a) = \text{G}$$

$$50. \text{FRTC}(R^b, R^a) = \text{T}$$

51. $(\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{DSE}(R^b, R^a))$

52. $R^b = \text{'acme1.ec1.dmz1.email1'}$ & $R^a = \text{'acme1.ec1.dmz1.email1'}$

53. $(\text{DSE}(R^b, R^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$

54. $p^b = \text{'a'}$ & $p^a = \text{'a'}$

55. $(\text{ASE}(a, a)) = \text{T}$

56. $p^b = \text{'c'}$ & $p^a = \text{'c'}$

57. $(\text{ASE}(c, c)) = \text{T}$

58. $p^b = \text{'m'}$ & $p^a = \text{'m'}$

59. $(\text{ASE}(m, m)) = \text{T}$

60. $p^b = \text{'e'}$ & $p^a = \text{'e'}$

61. $(\text{ASE}(e, e)) = \text{T}$

62. $p^b = \text{'1'}$ & $p^a = \text{'1'}$

63. $(\text{ASE}(1, 1)) = \text{T}$

64. $p^b = \text{'e'}$ & $p^a = \text{'e'}$

65. $(\text{ASE}(e, e)) = \text{T}$

66. $p^b = \text{'c'}$ & $p^a = \text{'c'}$

67. $(\text{ASE}(c, c)) = \text{T}$

68. $p^b = \text{'1'}$ & $p^a = \text{'1'}$

69. $(\text{ASE}(1, 1)) = \text{T}$

70. $p^b = \text{'d'}$ & $p^a = \text{'d'}$

71. $(\text{ASE}(d, d)) = \text{T}$

72. $p^b = \text{'m'}$ & $p^a = \text{'m'}$

73. $(\text{ASE}(m, m)) = \text{T}$

74. $p^b = \text{'z'}$ & $p^a = \text{'z'}$

$$75. (\text{ASE}(z, z)) = \text{T}$$

$$76. p^b = '1' \ \& \ p^a = '1'$$

$$77. (\text{ASE}(1, 1)) = \text{T}$$

$$78. p^b = 'e' \ \& \ p^a = 'e'$$

$$79. (\text{ASE}(e, e)) = \text{T}$$

$$80. p^b = 'm' \ \& \ p^a = 'm'$$

$$81. (\text{ASE}(m, m)) = \text{T}$$

$$82. p^b = 'a' \ \& \ p^a = 'a'$$

$$83. (\text{ASE}(a, a)) = \text{T}$$

$$84. p^b = 'i' \ \& \ p^a = 'i'$$

$$85. (\text{ASE}(i, i)) = \text{T}$$

$$86. p^b = 'l' \ \& \ p^a = 'l'$$

$$87. (\text{ASE}(l, l)) = \text{T}$$

$$88. p^b = '1' \ \& \ p^a = '1'$$

$$89. (\text{ASE}(1, 1)) = \text{T}$$

$$90. (\text{DSE}(\text{acme1.ec1.dmz1.email1}, \text{acme1.ec1.dmz1.email1})) = \text{T}$$

$$91. (\text{FRRC}(\text{acme1.ec1.dmz1.email1}, \text{acme1.ec1.dmz1.email1})) = \text{T}$$

$$92. (\text{FNC}(F^b, F^a)) \text{ invokes } (\text{DSE}(F^b, F^a))$$

$$93. F^b = 'hasemailaccount' \ \& \ F^a = 'hasemailaccount'$$

$$94. (\text{DSE}(F^b, F^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$$

$$95. p^b = 'h' \ \& \ p^a = 'h'$$

$$96. (\text{ASE}(h, h)) = \text{T}$$

$$97. p^b = 'a' \ \& \ p^a = 'a'$$

$$98. (\text{ASE}(a, a)) = \text{T}$$

$$99. p^b = 's' \ \&\ \& \ p^a = 's'$$

$$100. (\text{ASE}(s, s)) = \text{T}$$

$$101. p^b = 'e' \ \&\ \& \ p^a = 'e'$$

$$102. (\text{ASE}(e, e)) = \text{T}$$

$$103. p^b = 'm' \ \&\ \& \ p^a = 'm'$$

$$104. (\text{ASE}(m, m)) = \text{T}$$

$$105. p^b = 'a' \ \&\ \& \ p^a = 'a'$$

$$106. (\text{ASE}(a, a)) = \text{T}$$

$$107. p^b = 'i' \ \&\ \& \ p^a = 'i'$$

$$108. (\text{ASE}(i, i)) = \text{T}$$

$$109. p^b = 'l' \ \&\ \& \ p^a = 'l'$$

$$110. (\text{ASE}(l, l)) = \text{T}$$

$$111. p^b = 'a' \ \&\ \& \ p^a = 'a'$$

$$112. (\text{ASE}(a, a)) = \text{T}$$

$$113. p^b = 'c' \ \&\ \& \ p^a = 'c'$$

$$114. (\text{ASE}(c, c)) = \text{T}$$

$$115. p^b = 'c' \ \&\ \& \ p^a = 'c'$$

$$116. (\text{ASE}(c, c)) = \text{T}$$

$$117. p^b = 'o' \ \&\ \& \ p^a = 'o'$$

$$118. (\text{ASE}(o, o)) = \text{T}$$

$$119. p^b = 'u' \ \&\ \& \ p^a = 'u'$$

$$120. (\text{ASE}(u, u)) = \text{T}$$

$$121. p^b = 'n' \ \&\ \& \ p^a = 'n'$$

$$122. (\text{ASE}(n, n)) = \text{T}$$

$$123. p^b = 't' \ \& \ p^a = 't'$$

$$124. (\text{ASE}(t, t)) = \text{T}$$

$$125. (\text{DSE}(\text{hasemailaccount}, \text{hasemailaccount})) = \text{T}$$

$$126. (\text{FNC}(\text{hasemailaccount}, \text{hasemailaccount})) = \text{T}$$

$$127. (\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{FRTC}(R^b, R^a))$$

$$128. (\text{FRTC}(R^b, R^a)) \text{ invokes } (\text{FRTD}(R^b) \wedge \text{FRTD}(R^a))$$

$$129. R^b \in \mathcal{L}$$

$$130. \text{FRTD}(R^b) = \text{I}$$

$$131. R^a \in \mathcal{L}$$

$$132. \text{FRTD}(R^a) = \text{I}$$

$$133. \text{FRTC}(R^b, R^a) = \text{T}$$

$$134. (\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{NSE}(R^b, R^a))$$

$$135. (\text{NSE}(R^b, R^a)) \text{ invokes } (\text{SC}(R^b, R^a))$$

$$136. (\text{SC}(R^b, R^a)) \text{ invokes } (\text{SD}(R^b) \wedge \text{SD}(R^a))$$

$$137. R^b = '[\text{eid01}, \text{eid02}, \text{eid04}]' \ \& \ R^a = '[\text{eid01}, \text{eid02}, \text{eid04}]'$$

$$138. \text{SD}(R^b) = 3$$

$$139. \text{SD}(R^a) = 3$$

$$140. \text{SD}(R^b) = \text{SD}(R^a)$$

$$141. (\text{SC}(R^b, R^a)) = \text{E}$$

$$142. (\text{NSE}(R^b, R^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$$

$$143. p^b = 'e' \ \& \ p^a = 'e'$$

$$144. (\text{ASE}(e, e)) = \text{T}$$

$$145. p^b = 'i' \ \& \ p^a = 'i'$$

$$146. (\text{ASE}(i, i)) = \text{T}$$

147. $p^b = 'd' \ \& \ p^a = 'd'$

148. $(ASE(d, d)) = T$

149. $p^b = '0' \ \& \ p^a = '0'$

150. $(ASE(0, 0)) = T$

151. $p^b = '1' \ \& \ p^a = '1'$

152. $(ASE(1, 1)) = T$

153. $p^b = 'e' \ \& \ p^a = 'e'$

154. $(ASE(e, e)) = T$

155. $p^b = 'i' \ \& \ p^a = 'i'$

156. $(ASE(i, i)) = T$

157. $p^b = 'd' \ \& \ p^a = 'd'$

158. $(ASE(d, d)) = T$

159. $p^b = '0' \ \& \ p^a = '0'$

160. $(ASE(0, 0)) = T$

161. $p^b = '2' \ \& \ p^a = '2'$

162. $(ASE(2, 2)) = T$

163. $p^b = 'e' \ \& \ p^a = 'e'$

164. $(ASE(e, e)) = T$

165. $p^b = 'i' \ \& \ p^a = 'i'$

166. $(ASE(i, i)) = T$

167. $p^b = 'd' \ \& \ p^a = 'd'$

168. $(ASE(d, d)) = T$

169. $p^b = '0' \ \& \ p^a = '0'$

170. $(ASE(0, 0)) = T$

$$171. p^b = '4' \ \& \ p^a = '4'$$

$$172. (\text{ASE}(4, 4)) = \text{T}$$

$$173. (\text{NSE}([\text{eid}01, \text{eid}02, \text{eid}04], [\text{eid}01, \text{eid}02, \text{eid}04])) = \text{T}$$

$$174. (\text{FRRC}([\text{eid}01, \text{eid}02, \text{eid}04], [\text{eid}01, \text{eid}02, \text{eid}04])) = \text{T}$$

$$175. (\text{FNC}(F^b, F^a)) \text{ invokes } (\text{DSE}(F^b, F^a))$$

$$176. F^b = 'spamfilter' \ \& \ F^a = 'spamfilter'$$

$$177. (\text{DSE}(F^b, F^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$$

$$178. p^b = 's' \ \& \ p^a = 's'$$

$$179. (\text{ASE}(s, s)) = \text{T}$$

$$180. p^b = 'p' \ \& \ p^a = 'p'$$

$$181. (\text{ASE}(p, p)) = \text{T}$$

$$182. p^b = 'a' \ \& \ p^a = 'a'$$

$$183. (\text{ASE}(a, a)) = \text{T}$$

$$184. p^b = 'm' \ \& \ p^a = 'm'$$

$$185. (\text{ASE}(m, m)) = \text{T}$$

$$186. p^b = 'f' \ \& \ p^a = 'f'$$

$$187. (\text{ASE}(f, f)) = \text{T}$$

$$188. p^b = 'i' \ \& \ p^a = 'i'$$

$$189. (\text{ASE}(i, i)) = \text{T}$$

$$190. p^b = 'l' \ \& \ p^a = 'l'$$

$$191. (\text{ASE}(l, l)) = \text{T}$$

$$192. p^b = 't' \ \& \ p^a = 't'$$

$$193. (\text{ASE}(t, t)) = \text{T}$$

$$194. p^b = 'e' \ \& \ p^a = 'e'$$

195. $(ASE(e, e)) = T$
196. $p^b = 'r' \ \& \ p^a = 'r'$
197. $(ASE(r, r)) = T$
198. $(DSE(spamfilter, spamfilter)) = T$
199. $(FNC(spamfilter, spamfilter)) = T$
200. $(FRRC((FR)^b, (FR)^a)) \text{ invokes } (FRTC(R^b, R^a))$
201. $(FRTC(R^b, R^a)) \text{ invokes } (FRTD(R^b) \wedge FRTD(R^a))$
202. $R^b = 'yes' \ \& \ R^a = 'yes'$
203. $R^b \in \mathcal{V}$
204. $FRTD(R^b) = U$
205. $R^a \in \mathcal{V}$
206. $FRTD(R^a) = U$
207. $FRTC(R^b, R^a) = T$
208. $(FRRC((FR)^b, (FR)^a)) \text{ invokes } (ASE(p^b, p^a))$
209. $p^b = 'y' \ \& \ p^a = 'y'$
210. $(ASE(y, y)) = T$
211. $p^b = 'e' \ \& \ p^a = 'e'$
212. $(ASE(e, e)) = T$
213. $p^b = 's' \ \& \ p^a = 's'$
214. $(ASE(s, s)) = T$
215. $(ASE(yes, yes)) = T$
216. $(FRRC(yes, yes)) = T$
217. $(FNC(F^b, F^a)) \text{ invokes } (DSE(F^b, F^a))$
218. $F^b = 'allowmacros' \ \& \ F^a = 'allowmacros'$

219. $(DSE(F^b, F^a)) \text{ invokes } (ASE(p^b, p^a))$

220. $p^b = 'a' \ \& \ p^a = 'a'$

221. $(ASE(a, a)) = T$

222. $p^b = 'l' \ \& \ p^a = 'l'$

223. $(ASE(l, l)) = T$

224. $p^b = 'l' \ \& \ p^a = 'l'$

225. $(ASE(l, l)) = T$

226. $p^b = 'o' \ \& \ p^a = 'o'$

227. $(ASE(o, o)) = T$

228. $p^b = 'w' \ \& \ p^a = 'w'$

229. $(ASE(w, w)) = T$

230. $p^b = 'm' \ \& \ p^a = 'm'$

231. $(ASE(m, m)) = T$

232. $p^b = 'a' \ \& \ p^a = 'a'$

233. $(ASE(a, a)) = T$

234. $p^b = 'c' \ \& \ p^a = 'c'$

235. $(ASE(c, c)) = T$

236. $p^b = 'r' \ \& \ p^a = 'r'$

237. $(ASE(r, r)) = T$

238. $p^b = 'o' \ \& \ p^a = 'o'$

239. $(ASE(o, o)) = T$

240. $p^b = 's' \ \& \ p^a = 's'$

241. $(ASE(s, s)) = T$

242. $(DSE(allowmacros, allowmacros)) = T$

243. $(\text{FNC}(\text{allowmacros}, \text{allowmacros})) = \text{T}$
244. $(\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{FRTC}(R^b, R^a))$
245. $(\text{FRTC}(R^b, R^a)) \text{ invokes } (\text{FRTD}(R^b) \wedge \text{FRTD}(R^a))$
246. $R^b = \text{'yes'} \ \& \ R^a = \text{'yes'}$
247. $R^b \in \mathcal{V}$
248. $\text{FRTD}(R^b) = \text{U}$
249. $R^a \in \mathcal{V}$
250. $\text{FRTD}(R^a) = \text{U}$
251. $\text{FRTC}(R^b, R^a) = \text{T}$
252. $(\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$
253. $p^b = \text{'y'} \ \& \ p^a = \text{'y'}$
254. $(\text{ASE}(y, y)) = \text{T}$
255. $p^b = \text{'e'} \ \& \ p^a = \text{'e'}$
256. $(\text{ASE}(e, e)) = \text{T}$
257. $p^b = \text{'s'} \ \& \ p^a = \text{'s'}$
258. $(\text{ASE}(s, s)) = \text{T}$
259. $(\text{ASE}(\text{yes}, \text{yes})) = \text{T}$
260. $(\text{FRRC}(\text{yes}, \text{yes})) = \text{T}$
261. $(\text{FNC}(F^b, F^a)) \text{ invokes } (\text{DSE}(F^b, F^a))$
262. $F^b = \text{'allowimages'} \ \& \ F^a = \text{'allowimages'}$
263. $(\text{DSE}(F^b, F^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$
264. $p^b = \text{'a'} \ \& \ p^a = \text{'a'}$
265. $(\text{ASE}(a, a)) = \text{T}$
266. $p^b = \text{'l'} \ \& \ p^a = \text{'l'}$

$$267. (\text{ASE}(l, l)) = \text{T}$$

$$268. p^b = 'l' \ \& \ p^a = 'l'$$

$$269. (\text{ASE}(l, l)) = \text{T}$$

$$270. p^b = 'o' \ \& \ p^a = 'o'$$

$$271. (\text{ASE}(o, o)) = \text{T}$$

$$272. p^b = 'w' \ \& \ p^a = 'w'$$

$$273. (\text{ASE}(w, w)) = \text{T}$$

$$274. p^b = 'i' \ \& \ p^a = 'i'$$

$$275. (\text{ASE}(i, i)) = \text{T}$$

$$276. p^b = 'm' \ \& \ p^a = 'm'$$

$$277. (\text{ASE}(m, m)) = \text{T}$$

$$278. p^b = 'a' \ \& \ p^a = 'a'$$

$$279. (\text{ASE}(a, a)) = \text{T}$$

$$280. p^b = 'g' \ \& \ p^a = 'g'$$

$$281. (\text{ASE}(g, g)) = \text{T}$$

$$282. p^b = 'e' \ \& \ p^a = 'e'$$

$$283. (\text{ASE}(e, e)) = \text{T}$$

$$284. p^b = 's' \ \& \ p^a = 's'$$

$$285. (\text{ASE}(s, s)) = \text{T}$$

$$286. (\text{DSE}(\text{allowimages}, \text{allowimages})) = \text{T}$$

$$287. (\text{FNC}(\text{allowimages}, \text{allowimages})) = \text{T}$$

$$288. (\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{FRIC}(R^b, R^a))$$

$$289. (\text{FRIC}(R^b, R^a)) \text{ invokes } (\text{FRTD}(R^b) \wedge \text{FRTD}(R^a))$$

$$290. R^b = 'no' \ \& \ R^a = 'yes'$$

291. $R^b \in \mathcal{V}$
292. $\text{FRTD}(R^b) = \text{U}$
293. $R^a \in \mathcal{V}$
294. $\text{FRTD}(R^a) = \text{U}$
295. $\text{FRTC}(R^b, R^a) = \text{T}$
296. $(\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$
297. $p^b = 'n' \ \& \ p^a = 'y'$
298. $(\text{ASE}(n, y)) = \text{F}$
299. $p^b = 'o' \ \& \ p^a = 'e'$
300. $(\text{ASE}(o, e)) = \text{F}$
301. $p^b = " \ \& \ p^a = 's'$
302. $(\text{ASE}(, s)) = \text{F}$
303. $(\text{ASE}(no, yes)) = \text{F}$
304. $(\text{FRRC}(no, yes)) = \text{F}$
305. $(\text{FNC}(F^b, F^a)) \text{ invokes } (\text{DSE}(F^b, F^a))$
306. $F^b = 'allowimages' \ \& \ F^a = 'allowimages'$
307. $(\text{DSE}(F^b, F^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$
308. $p^b = 'a' \ \& \ p^a = 'a'$
309. $(\text{ASE}(a, a)) = \text{T}$
310. $p^b = 'l' \ \& \ p^a = 'l'$
311. $(\text{ASE}(l, l)) = \text{T}$
312. $p^b = 'l' \ \& \ p^a = 'l'$
313. $(\text{ASE}(l, l)) = \text{T}$
314. $p^b = 'o' \ \& \ p^a = 'o'$

$$315. (\text{ASE}(o, o)) = \text{T}$$

$$316. p^b = 'w' \ \& \ p^a = 'w'$$

$$317. (\text{ASE}(w, w)) = \text{T}$$

$$318. p^b = 's' \ \& \ p^a = 's'$$

$$319. (\text{ASE}(s, s)) = \text{T}$$

$$320. p^b = 'c' \ \& \ p^a = 'c'$$

$$321. (\text{ASE}(c, c)) = \text{T}$$

$$322. p^b = 'r' \ \& \ p^a = 'r'$$

$$323. (\text{ASE}(r, r)) = \text{T}$$

$$324. p^b = 'i' \ \& \ p^a = 'i'$$

$$325. (\text{ASE}(i, i)) = \text{T}$$

$$326. p^b = 'p' \ \& \ p^a = 'p'$$

$$327. (\text{ASE}(p, p)) = \text{T}$$

$$328. p^b = 't' \ \& \ p^a = 't'$$

$$329. (\text{ASE}(t, t)) = \text{T}$$

$$330. p^b = 's' \ \& \ p^a = 's'$$

$$331. (\text{ASE}(s, s)) = \text{T}$$

$$332. (\text{DSE}(\text{allowscripts}, \text{allowscripts})) = \text{T}$$

$$333. (\text{FNC}(\text{allowscripts}, \text{allowscripts})) = \text{T}$$

$$334. (\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{FRTC}(R^b, R^a))$$

$$335. (\text{FRTC}(R^b, R^a)) \text{ invokes } (\text{FRTD}(R^b) \wedge \text{FRTD}(R^a))$$

$$336. R^b = 'no' \ \& \ R^a = 'yes'$$

$$337. R^b \in \mathcal{V}$$

$$338. \text{FRTD}(R^b) = \text{U}$$

$$339. R^a \in \mathcal{V}$$

$$340. \text{FRTD}(R^a) = \text{U}$$

$$341. \text{FRTC}(R^b, R^a) = \text{T}$$

$$342. (\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$$

$$343. p^b = 'n' \ \& \ p^a = 'y'$$

$$344. (\text{ASE}(n, y)) = \text{F}$$

$$345. p^b = 'o' \ \& \ p^a = 'e'$$

$$346. (\text{ASE}(o, e)) = \text{F}$$

$$347. p^b = ' ' \ \& \ p^a = 's'$$

$$348. (\text{ASE}(, s)) = \text{F}$$

$$349. (\text{ASE}(no, yes)) = \text{F}$$

$$350. (\text{FRRC}(no, yes)) = \text{F}$$

$$351. (\text{FNC}(F^b, F^a)) \text{ invokes } (\text{DSE}(F^b, F^a))$$

$$352. F^b = 'allowimages' \ \& \ F^a = 'allowimages'$$

$$353. (\text{DSE}(F^b, F^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$$

$$354. p^b = 'a' \ \& \ p^a = 'a'$$

$$355. (\text{ASE}(a, a)) = \text{T}$$

$$356. p^b = 'l' \ \& \ p^a = 'l'$$

$$357. (\text{ASE}(l, l)) = \text{T}$$

$$358. p^b = 'l' \ \& \ p^a = 'l'$$

$$359. (\text{ASE}(l, l)) = \text{T}$$

$$360. p^b = 'o' \ \& \ p^a = 'o'$$

$$361. (\text{ASE}(o, o)) = \text{T}$$

$$362. p^b = 'w' \ \& \ p^a = 'w'$$

$$363. (\text{ASE}(w, w)) = \text{T}$$

$$364. p^b = 'h' \ \& \ p^a = 'h'$$

$$365. (\text{ASE}(i, i)) = \text{T}$$

$$366. p^b = 'y' \ \& \ p^a = 'y'$$

$$367. (\text{ASE}(m, m)) = \text{T}$$

$$368. p^b = 'p' \ \& \ p^a = 'p'$$

$$369. (\text{ASE}(a, a)) = \text{T}$$

$$370. p^b = 'e' \ \& \ p^a = 'e'$$

$$371. (\text{ASE}(g, g)) = \text{T}$$

$$372. p^b = 'r' \ \& \ p^a = 'r'$$

$$373. (\text{ASE}(e, e)) = \text{T}$$

$$374. p^b = 'l' \ \& \ p^a = 'l'$$

$$375. (\text{ASE}(s, s)) = \text{T}$$

$$376. p^b = 'i' \ \& \ p^a = 'i'$$

$$377. (\text{ASE}(s, s)) = \text{T}$$

$$378. p^b = 'n' \ \& \ p^a = 'n'$$

$$379. (\text{ASE}(s, s)) = \text{T}$$

$$380. p^b = 'k' \ \& \ p^a = 'k'$$

$$381. (\text{ASE}(s, s)) = \text{T}$$

$$382. p^b = 's' \ \& \ p^a = 's'$$

$$383. (\text{ASE}(s, s)) = \text{T}$$

$$384. (\text{DSE}(\text{allowhyperlinks}, \text{allowhyperlinks})) = \text{T}$$

$$385. (\text{FNC}(\text{allowhyperlinks}, \text{allowhyperlinks})) = \text{T}$$

$$386. (\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{FRIC}(R^b, R^a))$$

387. $(\text{FRTC}(R^b, R^a)) \text{ invokes } (\text{FRTD}(R^b) \wedge \text{FRTD}(R^a))$

388. $R^b = \text{'no'} \ \& \ R^a = \text{'yes'}$

389. $R^b \in \mathcal{V}$

390. $\text{FRTD}(R^b) = \text{U}$

391. $R^a \in \mathcal{V}$

392. $\text{FRTD}(R^a) = \text{U}$

393. $\text{FRTC}(R^b, R^a) = \text{T}$

394. $(\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$

395. $p^b = \text{'n'} \ \& \ p^a = \text{'y'}$

396. $(\text{ASE}(n, y)) = \text{F}$

397. $p^b = \text{'o'} \ \& \ p^a = \text{'e'}$

398. $(\text{ASE}(o, e)) = \text{F}$

399. $p^b = \text{' ' } \ \& \ p^a = \text{'s'}$

400. $(\text{ASE}(, s)) = \text{F}$

401. $(\text{ASE}(no, yes)) = \text{F}$

402. $(\text{FRRC}(no, yes)) = \text{F}$

403. $(\text{FNC}(F^b, F^a)) \text{ invokes } (\text{DSE}(F^b, F^a))$

404. $F^b = \text{'parent'} \ \& \ F^a = \text{'parent'}$

405. $(\text{DSE}(F^b, F^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$

406. $p^b = \text{'p'} \ \& \ p^a = \text{'p'}$

407. $(\text{ASE}(p, p)) = \text{T}$

408. $p^b = \text{'a'} \ \& \ p^a = \text{'a'}$

409. $(\text{ASE}(a, a)) = \text{T}$

410. $p^b = \text{'r'} \ \& \ p^a = \text{'r'}$

411. $(\text{ASE}(r, r)) = \text{T}$
412. $p^b = 'e' \ \& \ p^a = 'e'$
413. $(\text{ASE}(e, e)) = \text{T}$
414. $p^b = 'n' \ \& \ p^a = 'n'$
415. $(\text{ASE}(n, n)) = \text{T}$
416. $p^b = 't' \ \& \ p^a = 't'$
417. $(\text{ASE}(t, t)) = \text{T}$
418. $(\text{DSE}(\text{parent}, \text{parent})) = \text{T}$
419. $(\text{FNC}(\text{parent}, \text{parent})) = \text{T}$
420. $(\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{FRTC}(R^b, R^a))$
421. $(\text{FRTC}(R^b, R^a)) \text{ invokes } (\text{FRTD}(R^b) \wedge \text{FRTD}(R^a))$
422. $R^b \in \mathcal{Q}$
423. $\text{FRTD}(R^b) = \text{G}$
424. $R^a \in \mathcal{Q}$
425. $\text{FRTD}(R^a) = \text{G}$
426. $\text{FRTC}(R^b, R^a) = \text{T}$
427. $(\text{FRRC}((FR)^b, (FR)^a)) \text{ invokes } (\text{DSE}(R^b, R^a))$
428. $R^b = 'acme1.ec1.dmz1' \ \& \ R^a = 'acme1.ec1.dmz1'$
429. $(\text{DSE}(R^b, R^a)) \text{ invokes } (\text{ASE}(p^b, p^a))$
430. $p^b = 'a' \ \& \ p^a = 'a'$
431. $(\text{ASE}(a, a)) = \text{T}$
432. $p^b = 'c' \ \& \ p^a = 'c'$
433. $(\text{ASE}(c, c)) = \text{T}$
434. $p^b = 'm' \ \& \ p^a = 'm'$

$$435. (\text{ASE}(m, m)) = \text{T}$$

$$436. p^b = 'e' \ \& \ p^a = 'e'$$

$$437. (\text{ASE}(e, e)) = \text{T}$$

$$438. p^b = '1' \ \& \ p^a = '1'$$

$$439. (\text{ASE}(1, 1)) = \text{T}$$

$$440. p^b = 'e' \ \& \ p^a = 'e'$$

$$441. (\text{ASE}(e, e)) = \text{T}$$

$$442. p^b = 'c' \ \& \ p^a = 'c'$$

$$443. (\text{ASE}(c, c)) = \text{T}$$

$$444. p^b = '1' \ \& \ p^a = '1'$$

$$445. (\text{ASE}(1, 1)) = \text{T}$$

$$446. p^b = 'd' \ \& \ p^a = 'd'$$

$$447. (\text{ASE}(d, d)) = \text{T}$$

$$448. p^b = 'm' \ \& \ p^a = 'm'$$

$$449. (\text{ASE}(m, m)) = \text{T}$$

$$450. p^b = 'z' \ \& \ p^a = 'z'$$

$$451. (\text{ASE}(z, z)) = \text{T}$$

$$452. p^b = '1' \ \& \ p^a = '1'$$

$$453. (\text{ASE}(1, 1)) = \text{T}$$

$$454. (\text{DSE}(acme1.ec1.dmz1, acme1.ec1.dmz1)) = \text{T}$$

$$455. (\text{FRRC}(acme1.ec1.dmz1, acme1.ec1.dmz1)) = \text{T}$$

QuodEratDemonstrandum(Q.E.D.)

By iterative extrapolation of the proof on all entities of O and S , we find that there exist no other applicable entities between O and S . However, we declare that O is applicable to S , since there exists at least one applicable entity between the specifications O and S .

6.37 CHAPTER CONCLUSION

Cyber-attacks against critical infrastructure have the potential to cause tremendous damage, suffering, and loss. One approach to help protect organizations from cyber-attacks is an automated, adversary-based risk assessment of critical infrastructure. Utilization of specifications can be one mechanism to automate adversarial risk assessment. Factors lending a hand in the non-existence of a specification-based, adversarial risk assessment process have been presented. A previously presented Specification- and Adversary-based Risk Assessment process, HESTIA, was briefly discussed. We validated the formal model of the HESTIA process by using formal verification and proof by tracing using an example derived from our case study.

FORMAL NOTATION NOMENCLATURE

AlphanumericString - A string containing English alphabets and/or numbers

ASE - **AlphanumericString** Equality operation

Dotted String - A chain of **AlphanumericStrings** that are separated by dots.

DE - Dictionary Equality operation

DKII - Dictionary Key Intersection Identification operation

DSE - Dotted String Equality operation

DSII - Dotted String Intersection Identification operation

EIC - Entity ID Comparison operation

ETC - Entity Type Comparison operation

FRA - Field Register Application operation

FNC - Field Name Comparison operation

FRC - Field Register Comparison operation

FRRC - Field Register Relaxed Comparison operation

FRTD - Field Register Type Determination operation

FRTC - Field Register Type Comparison operation

EID - Entity Identifier

ET - Entity Type

FN - Field Name

Named Set - An unordered list containing **AlphanumericString** values

NSE - Named Set Equality operation

NSII - Named Set Intersection Identification operation

SC - Size Comparison operation

SD - Size Determination operation

\mathcal{S} - Set of all valid specifications

\mathcal{T} - Set of all valid templates

\mathcal{E} - Set of all valid entities

\mathcal{F} - Set of all valid fields

\mathcal{V} - Set of all valid `AlphanumericString` values

\mathcal{Q} - Set of all valid dotted strings

\mathcal{D} - Set of all valid dictionaries

\mathcal{L} - Set of all valid named sets

\propto - Consistency Checking operator

\otimes - Conflict Checking operator

$\overset{?}{\vdash}$ - Applicability Checking operator

\vdash - Delta Application operator

CHAPTER 7: FUTURE WORK AND CONCLUSION

In this chapter, we present the specific future work items that can be pursued as a direct continuation of this dissertation. We also present a discussion about a broader scope of possible future work. Finally, we present a conclusion to this dissertation document.

7.1 SPECIFIC FUTURE WORK ITEMS

The following future work items can be pursued, as a direct continuation of this dissertation:

1. Implementing hierarchical and/or inheritance checking for consistency, conflicts, and applicability check engines;
2. Adding specification of exceptions for the consistency, conflicts, and applicability checks;
3. Designing operations to allow selective and partial application of values, dotted strings, dictionaries, and named sets;
4. Extensions to allow merging of values, dotted strings, dictionaries, and named sets.

Figure 7.1 presents HESTIA's architecture after incorporating the second aforementioned future work item.

7.2 BROADER SCOPE FOR FUTURE WORK: HESTIA SYSTEM

Another avenue for future work in this project would be to evaluate the feasibility and practicality of automating the generation of: 1) original system specifications and 2) library of attack/hardening measure specifications, which are currently written/generated by humans. A potential future work item can also be automating consistency resolution, which is currently handled by humans-in-the-loop. Furthermore, another future work item can be incorporating specific metrics such as: a) costs of attacks & defenses, b) measuring safety level or score of a system, and c) hazard measurement of applying attacks. Developing a practical implementation of the HESTIA system is also a noteworthy future work item that is commercially promising.

7.3 BROADER SCOPE FOR FUTURE WORK: ACME CORP. CASE STUDY

By encoding Acme Corp. model in a simulated environment, it can be used to validate machine-learning algorithms intended to detect attacks on CPS organizations like the BEDMM project [43]. BEDMM is a best-effort damage mitigation model for cyber-attacks on smart grids. Another example of

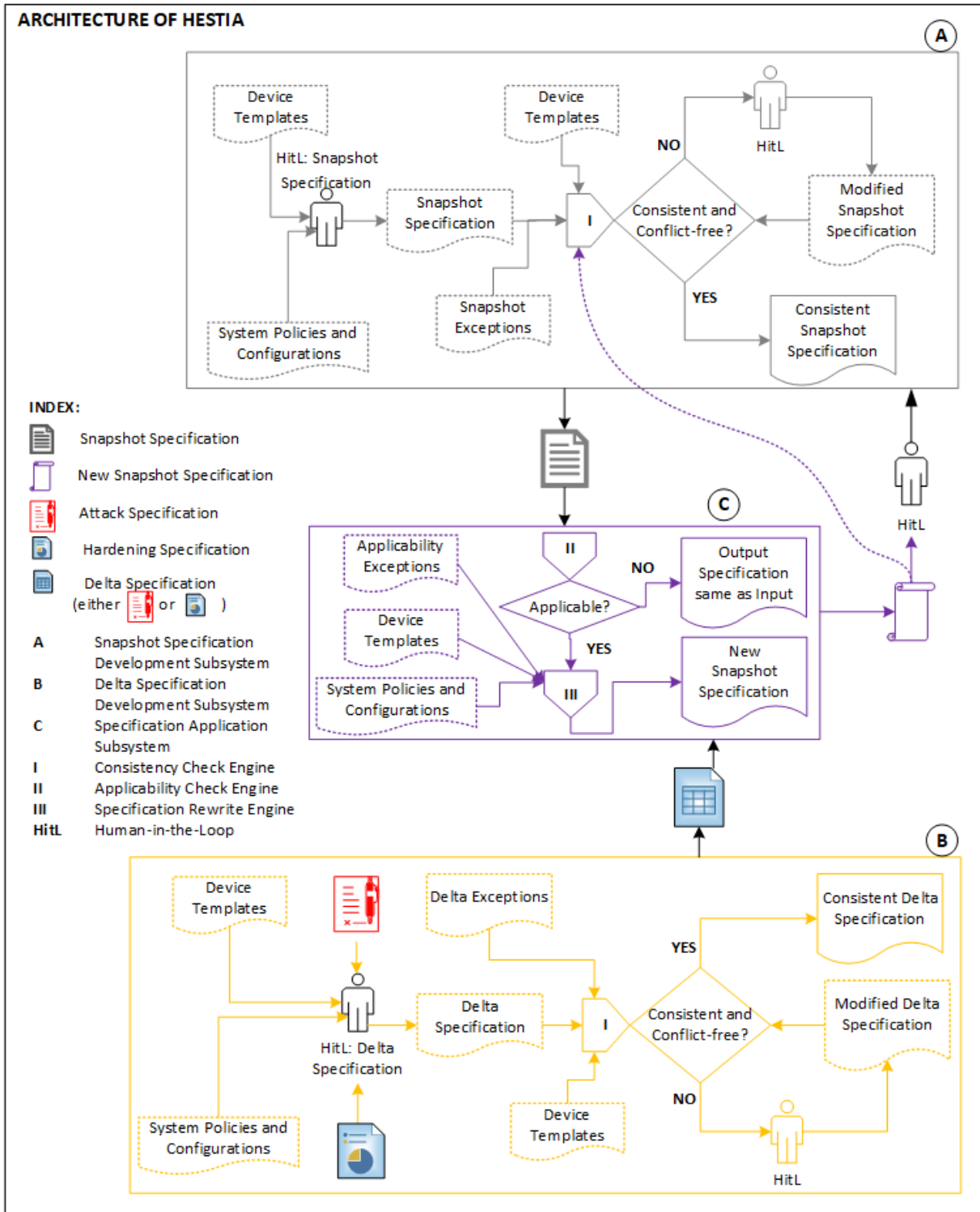


Figure 7.1: HESTIA tool-set’s high-level architecture. Adapted from a previously published diagram [2].

MATLAB data format-based validation of machine-learning algorithms is the SFDI project [35]. SFDI is a deep-learning-based approach to detect stealthy false data injection attacks in CPS organizations. In previous Chapters, researchers of the HESTIA and SFDI projects have stressed on the need for holistic models of a CPS organization, for the purpose of validating their research [2], [35].

The Acme Corp. model can be used to test the validity and functionality of cyber physical testbed(s). Constructing testbed(s) that can simulate the Acme Corp. model will also help facilitate cross-over collaboration of other research projects mentioned in this section. For example, a collaboration between HESTIA, BEDMM, and SFDI projects can be established if the researchers are able to simulate the Acme Corp. model on a testbed [44].

The Acme Corp. model is not only useful for validating research projects, but can also be used as a common platform for comparison and testing of multiple CPS security research approaches and as an instrument to teach CPS security. In addition, in our consultation with subject matter experts (SMEs) we were informed that there is a need in both academia and research industry for holistic CPS models such as the one we present in this dissertation. For example, this model could be used to replay a cyber attack to observe and measure: 1) the system's vulnerabilities; 2) effectiveness of the system's defense mechanisms; and 3) impact of the attack on the system. One of our immediate objectives is to replicate the December 2015 cyber attack on the Ukraine power grid [45].

7.4 CONCLUSION

The world is becoming more technologically dynamic by the day. Manual auditing and risk assessment processes are no longer sufficient to protect our infrastructure. To conclude, we would like to emphasize that it is high time to start investing resources into advancing automatic or semi-automatic risk assessment systems for critical infrastructure. We hope that our dissertation's main contribution, HESTIA, will enable the cybersecurity researchers to start paying more attention to automated or semi-automated risk assessment systems.

Overall, the results of our research demonstrate that it is possible to further automate risk assessment processes of critical infrastructure. We are hopeful that the contributions presented in this dissertation will enable the cybersecurity community to at least move towards a more secured critical infrastructure environment.

REFERENCES

- [1] A. A. Jillepalli, D. Conte de Leon, Y. Chakhchoukh, M. Ashrafuzzaman, B. K. Johnson, F. T. Sheldon, J. Alves-Foss, P. Tomic, and M. A. Haney, “An Architecture for HESTIA: High-level and Extensible System for Training and Infrastructure risk Assessment,” *International Journal of Internet of Things and Cyber-Assurance*, vol. 2, no. 5, pp. 103–121, 2018.
- [2] A. A. Jillepalli, D. Conte de Leon, M. Ashrafuzzaman, Y. Chakhchoukh, B. K. Johnson, F. T. Sheldon, J. Alves-Foss, P. Tomic, and M. A. Haney, “HESTIA: Adversarial modeling and risk assessment for CPCS,” in *2018 14th International Wireless Communications and Mobile Computing Conference (IWCMC)*, August 2018, pp. 226–231.
- [3] A. A. Jillepalli, D. Conte de Leon, B. K. Johnson, Y. Chakhchoukh, I. A. Oyewumi, M. Ashrafuzzaman, F. T. Sheldon, J. Alves-Foss, and M. A. Haney, “METICS: A holistic cyber physical system model for ieee 14-bus power system security,” in *2018 13th International Conference on Malicious and Unwanted Software (MALCON)*, October 2018, pp. 95–102.
- [4] C.-C. Sun, C.-C. Liu, and J. Xie, “Cyber-physical system security of a power grid: State-of-the-art,” *Electronics*, vol. 5, no. 3, 2016. [Online]. Available: <http://www.mdpi.com/2079-9292/5/3/40>
- [5] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A.-R. Sadeghi, M. Maniatakos, and R. Karri, “The cybersecurity landscape in industrial control systems,” *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1039–1057, 2016.
- [6] Y. Mo, T. H.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, “Cyber-physical security of a smart grid infrastructure,” *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, Jan 2012.
- [7] S. Tan, D. De, W.-Z. Song, J. Yang, and S. K. Das, “Survey of security advances in smart grid: A data driven approach,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 397–422, First Quarter 2017.
- [8] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, “Guide to industrial control systems (ICS) security,” National Institute of Standards and Technology, Tech. Rep., May 2015.
- [9] A. Jillepalli, D. C. de Leon, S. Steiner, and F. T. Sheldon, “HERMES: A high-level policy language for high-granularity enterprise-wide secure browser configuration management,” in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2016, pp. 1–9.

- [10] L. Zhang, “Modeling large scale complex cyber physical control systems based on system of systems engineering approach,” in *2014 20th International Conference on Automation and Computing*, Sept 2014, pp. 55–60.
- [11] —, “Formal specification for real time cyber physical systems using aspect-oriented approach,” in *2011 Fifth International Conference on Theoretical Aspects of Software Engineering*, Aug 2011, pp. 213–216.
- [12] The MITRE Corporation, “Making security measurable,” <https://makingsecuritymeasurable.mitre.org/>, July 2013.
- [13] P.-Y. Chen, S. Yang, J. A. McCann, J. Lin, and X. Yang, “Detection of false data injection attacks in smart-grid systems,” *IEEE Communications Magazine*, vol. 53, no. 2, pp. 206–213, February 2015.
- [14] Y. Wang, M. M. Amin, J. Fu, and H. B. Moussa, “A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids,” *IEEE Access*, vol. 5, pp. 26 022–26 033, 2017.
- [15] M. Esmalifalak, N. T. Nguyen, R. Zheng, and Z. Han, “Detecting stealthy false data injection using machine learning in smart grid,” in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec 2013, pp. 808–813.
- [16] Y. He, G. J. Mendis, and J. Wei, “Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism,” *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, Sept 2017.
- [17] Y. Chakhchoukh and H. Ishii, “Coordinated cyber-attacks on the measurement function in hybrid state estimation,” *IEEE Transactions on Power Systems*, vol. 30, no. 5, pp. 2487–2497, September 2015.
- [18] —, “Enhancing robustness to cyber-attacks in power systems through multiple least trimmed squares state estimations,” *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4395–4405, 2016.
- [19] Y. Chakhchoukh, V. Vittal, G. T. Heydt, and H. Ishii, “Lts-based robust hybrid se integrating correlation,” *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3127–3135, 2017.
- [20] V. S. Koganti, M. Ashrafuzzaman, A. A. Jillepalli, F. T. Sheldon, D. Conte de Leon, and B. K. Johnson, “A virtual testbed for security management of industrial control systems,” in *2018 12th International Malicious and Unwanted Software Conference (MALCON)*, Oct 2017, pp. 85–90.

- [21] A. A. Jillepalli, F. T. Sheldon, D. Conte de Leon, M. A. Haney, and R. K. Abercrombie, "Security management of cyber physical control systems using NIST SP 800-82r2," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, June 2017, pp. 1864–1870.
- [22] Z. Guan, N. Sun, Y. Xu, and T. Yang, "A comprehensive survey of false data injection in smart grid," *International Journal of Wireless and Mobile Computing - Inderscience Publishers*, vol. 8, no. 1, pp. 27–33, 2015.
- [23] X. Liu and Z. Li, "False data attack models, impact analyses and defense strategies in the electricity grid (subsection 5.2)," *The Electricity Journal - Elsevier*, vol. 30, pp. 35–42, 2017.
- [24] R. E. Mahan, J. D. Fluckiger, S. L. Clements, C. W. Tews, J. R. Burnette, C. A. Goranson, and H. Kirkham, "Secure data transfer guidance for industrial control and SCADA systems," Pacific Northwest National Laboratory (PNNL), Richland, WA (US), Tech. Rep. PNNL-20776, 2011. [Online]. Available: http://www.pnl.gov/main/publications/external/technical_reports/PNNL-20776.pdf
- [25] A. A. Jillepalli and D. C. d. Leon, "An architecture for a policy-oriented web browser management system: HiFiPol: Browser," in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, June 2016, pp. 382–387.
- [26] A. A. Jillepalli, "HiFiPol: Browser-securing the web browsing ecosystem," Master's thesis, University of Idaho, May 2017.
- [27] D. E. Whitehead, K. Owens, D. Gammel, and J. Smith, "Ukraine cyber-induced power outage: Analysis and practical mitigation strategies," in *2017 70th Annual Conference for Protective Relay Engineers (CPRE)*, April 2017, pp. 1–8, 10.1109/CPRE.2017.8090056.
- [28] A. France-Presse, "Massive power failure plunges 80% of Pakistan into darkness," www.theguardian.com/world/2015/jan/25/massive-power-failure-plunges-80-of-pakistan-into-darkness, January 2015, Online. Visited: December 11, 2017.
- [29] Morgan, Steve, "Global cybercrime damages predicted to reach \$6 trillion annually by 2021," <https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021/>, May 2019.
- [30] "Form and style for ASTM standards," ASTM International, Tech. Rep. ASTM-042018, 2018. [Online]. Available: https://www.astm.org/bluebook_FormStyle.pdf

- [31] A. A. Jillepalli, D. Conte de Leon, I. A. Oyewumi, J. Alves-Foss, B. K. Johnson, C. L. Jeffery, Y. Chakhchoukh, M. A. Haney, and F. T. Sheldon, “Formalizing the HESTIA process: Checking consistency and conflicts,” in *2019 3rd IEEE Texas Power and Energy Conference (TPEC)*, Mar 2019.
- [32] M. Mernik, J. Heering, and A. M. Sloane, “When and how to develop domain-specific languages,” *ACM Comput. Surv.*, vol. 37, no. 4, pp. 316–344, Dec. 2005. [Online]. Available: <http://doi.acm.org/10.1145/1118890.1118892>
- [33] S. Steiner, “A semantic least privilege and semi-automated approach to preventing cyber attacks on web applications,” PhD dissertation, University of Idaho, 2018.
- [34] Wikipedia, “Comparison of programming languages (strings),” [https://en.wikipedia.org/wiki/Comparison_of_programming_languages_\(strings\)](https://en.wikipedia.org/wiki/Comparison_of_programming_languages_(strings)), accessed: 12th June 2020.
- [35] M. Ashrafuzzaman, Y. Chakhchoukh, A. A. Jillepalli, P. Tomic, D. Conte de Leon, F. T. Sheldon, and B. K. Johnson, “Detecting stealthy false data injection attacks in power grids using deep learning,” in *2018 14th International Wireless Communications and Mobile Computing Conference (IWCMC)*, August 2018, pp. 219–225.
- [36] “14 Bus Power Flow Test Case,” <https://www2.ee.washington.edu/research/pstca/pf14/pg-tca14bus.htm>, accessed: 30th April 2018.
- [37] Z. Cheng, J. Duan, and M.-Y. Chow, “To centralize or to distribute: That is the question: A comparison of advanced microgrid management systems,” *IEEE Industrial Electronics Magazine*, vol. 12, no. 1, pp. 6–24, March 2018.
- [38] C. B. Vellaithurai, S. S. Biswas, R. Liu, and A. Srivastava, *Real Time Modeling and Simulation of Cyber-Power System*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 43–74. [Online]. Available: https://doi.org/10.1007/978-3-662-45928-7_3
- [39] A. A. Jillepalli, D. Conte de Leon, J. Alves-Foss, C. L. Jeffery, M. A. Haney, and F. T. Sheldon, “A formal model for hestia: an automated, adversary-aware risk assessment process for critical infrastructure,” in *PhD Dissertation, University of Idaho*, August 2020, p. 178.
- [40] L. Lamport, “How to write a proof,” Published: <https://lamport.azurewebsites.net/pubs/lamport-how-to-write.pdf>, February 1993.

- [41] “Formal proof: Understanding, writing and evaluating proofs,” Open University of Catalonia, Tech. Rep. PID00154272, 2015. [Online]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/57346/1/Formal%20Proof%3B%20Understanding%2C%20writing%20and%20evaluating%20proofs.pdf>
- [42] D. E. Whitehead, K. Owens, D. Gammel, and J. Smith, “Ukraine cyber-induced power outage: Analysis and practical mitigation strategies,” in *2017 70th Annual Conference for Protective Relay Engineers (CPRE)*, April 2017, pp. 1–8.
- [43] M. Ashrafuzzaman, H. Jamil, Y. Chakhchoukh, and F. T. Sheldon, “A best-effort damage mitigation model for cyber-attacks on smart grids,” in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 02, July 2018, pp. 510–515.
- [44] I. A. Oyewumi, A. A. Jillepalli, P. Richardson, M. Ashrafuzzaman, Y. Chakhchoukh, B. K. Johnson, M. A. Haney, F. T. Sheldon, and D. Conte de Leon, “ISAAC: The idaho CPS smart grid cybersecurity testbed,” in *2019 3rd IEEE Texas Power and Energy Conference (TPEC)*, Mar 2019.
- [45] R. M. Lee, M. J. Assante, and T. Conway, “Analysis of the cyber attack on the ukrainian power grid,” Electronic Information Sharing and Analysis Center (E-ISAC), Washington, DC (US), Tech. Rep. EISAC-18032018, 2018. [Online]. Available: https://www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf

APPENDIX A: COPYRIGHT AND CREDIT NOTICES

As listed in the Chapter 1.4, the Chapters 2, 3, and 5 of this dissertation are currently copyrighted by Inderscience Publishers and the Institute of Electrical and Electronics Engineers Inc. (IEEE). Permission for reproduction of complete article, towards use in theses/dissertations has been given by both Inderscience and the IEEE, provided the following statements are placed in the dissertation. Figures A.1, A.2 and A.3 provides proof of this authorization. The content of these reproduced articles has been changed to suit the format of a dissertation; instead of a scholarly article. The numbering of figures and listings has changed, due to being reproduced in this dissertation. However, the figures and listings, and their captions themselves remain unchanged.

Chapter 2 of this dissertation ©2018 Inderscience. Reprinted, with permission. Citation: Ananth A. Jillepalli; Daniel Conte De Leon; Yacine Chakhchoukh; Mohammad Ashrafuzzaman; Brian K. Johnson; Frederick T. Sheldon; Jim Alves-Foss; Predrag T. Tomic; Michael A. Haney, “An architecture for HESTIA: high-level and extensible system for training and infrastructure risk assessment”, In *International Journal of Internet of Things and Cyber-Assurance (IJITCA)*, vol. 1, no. 2, pp.173 - 193, 2018. DOI: 10.1504/IJITCA.2018.092478 [1].

Chapter 3 of this dissertation ©2018 IEEE. Reprinted, with permission. Citation: Ananth A. Jillepalli, Daniel Conte de Leon, Mohammad Ashrafuzzaman, Yacine Chakhchoukh, Brian K. Johnson, Frederick T. Sheldon, Jim Alves-Foss, Predrag T. Tomic, Michael A. Haney, “HESTIA: Adversarial Modeling and Risk Assessment for CPCS”, In *the 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, Limassol, pp. 226-231, 2018. DOI: 10.1109/IWCMC.2018.8450297 [2].

Chapter 5 of this dissertation ©2018 IEEE. Reprinted, with permission. Citation: Ananth A. Jillepalli, Daniel Conte de Leon, Brian K. Johnson, Yacine Chakhchoukh, Ibukun A. Oyewumi, Mohammad Ashrafuzzaman, Frederick T. Sheldon, Jim Alves-Foss, and Michael A. Haney, “METICS: A Holistic Cyber Physical System Model for IEEE 14-bus Power System Security”, In *13th International Conference on Malicious and Unwanted Software (MALWARE)*, Nantucket, pp. 95-102, 2018. DOI: 10.1109/MALWARE.2018.8659367 [3].

For portions not copyrighted by IEEE or any other publisher (as of the date of this publication), and for this dissertation as a whole; copyrights are retained by the author. Permission for not-for-profit and academic use is granted. For any other right to copy, transfer, reprint, republish, or for-profit use; permission must be sought from the author (Ananth A. Jillepalli).

3. What am I entitled to once my article is published and what am I allowed to do with the published article?

Authors of accepted articles will receive a PDF file of their published article. Hardcopies of journal issues may be purchased at a special price for authors from subs@inderscience.com.

Authors can use their article **for non-commercial purposes** after publication in these ways:

1. Posting the *Author's Original** on the Author's personal or departmental web pages and/or institutional repositories and/or subject repositories without embargo and sharing it as much as desired. For open [freely available] repositories, if the manuscript was funded by either RCUK or the Wellcome Trust, use the [CC BY: Creative Commons Attribution-NoDerivs 4.0](#). Otherwise, follow the licensing restrictions applied to all material copyrighted by Inderscience;

2. *Accepted Manuscript**

- Internally sharing the *Accepted Manuscript* within their research collaboration groups only, at any point after publication
- Posting the *Accepted Manuscript* on institutional repositories and/or subject repositories, subject to an embargo of *12 months* after publication (Green Open Access)
- Posting the *Accepted Manuscript* on academic social networks or social media, subject to an embargo of *24 months* after publication (Green Open Access)

Note for authors of articles funded by Research Councils UK (RCUK) and Wellcome Trust and other governmental organisations: If you are required to deposit your accepted manuscript into your institutional repository within 90 days of acceptance and our embargo period is longer than that permitted by your funder, please choose [Gold Open Access](#). If this is not possible for you, please speak to your institution about applying for an exception to HEFCE's Research Excellence Framework policy.

3. Posting the *Version of Record** to a subject-based repository such as PubMed Central *only* in cases where a funding agency providing the grant for the research on which the Article is based requires this of the Author, upon condition that it shall not be accessible until after six months from Inderscience's publication date. The PDF of the VoR should not be posted anywhere else unless it has been published as Open Access. This also applies to any Author who has published with Inderscience in the past;

4. Using the article in further research and in courses that the Author is teaching;

5. Incorporating the article content in other works by the Author.

In all cases, acknowledgement in the form of a full citation must be given to the journal as the original source of publication, together with a link to the journal webpage and/or DOI as soon as they are available.

Figure A.1: Permission from Inderscience to reproduce an article as Chapter 2 of this dissertation.

HESTIA: Adversarial Modeling and Risk Assessment for CPCS



Conference Proceedings:

2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)

Author: Ananth A. Jillepalli

Publisher: IEEE

Date: June 2018

Copyright © 2018, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:


- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis online.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

Figure A.2: Permission from IEEE to reproduce an article as Chapter 3 of this dissertation.



Requesting
permission
to reuse
content from
an IEEE
publication

METICS: A Holistic Cyber Physical System Model for IEEE 14-bus Power System Security

Conference Proceedings:
2018 13th International Conference on Malicious and Unwanted Software (MALWARE)

Author: Ananth A. Jillepalli

Publisher: IEEE

Date: Oct. 2018

Copyright © 2018, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

Figure A.3: Permission from IEEE to reproduce an article as Chapter 4 of this dissertation.