
# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**ANALYSIS OF DOMAIN-SPECIFIC NUCLEAR ONTOLOGY USING MONTEREY PHOENIX BEHAVIOR MODELING**

by

Landa R. McClure

March 2022

Thesis Advisor: Kristin M. Giammarco
Second Reader: Douglas L. Van Bossuyt

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 |
|---|---|---|---|



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE March 2022 | 3. REPORT TYPE AND DATES COVERED Master's thesis |
|---|---|---|

**4. TITLE AND SUBTITLE**
ANALYSIS OF DOMAIN-SPECIFIC NUCLEAR ONTOLOGY USING MONTEREY PHOENIX BEHAVIOR MODELING

**5. FUNDING NUMBERS**

**6. AUTHOR(S)** Landa R. McClure

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Naval Postgraduate School
Monterey, CA 93943-5000

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
N/A

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**
Approved for public release. Distribution is unlimited.

**12b. DISTRIBUTION CODE**
A

**13. ABSTRACT (maximum 200 words)**

Current nuclear energy ontologies are known to lack a common vocabulary to formally verify nuclear energy data relationships for modeling system behaviors. Idaho National Laboratory (INL) developed the Data Integration Aggregated Model and Ontology for Nuclear Deployment (DIAMOND) ontology to provide a standard vocabulary and taxonomy for identifying data relationships in nuclear energy system models. This thesis conducted an analysis of DIAMOND using a Spent Fuel Pool (SFP) Monterey Phoenix (MP) behavior model. The SFP MP behavior modeling application demonstrated components of and interactions among a spent fuel cooling pool and its environment. The MP behavior model demonstrated a viable approach for analyzing nuclear reactor system behavior consistent with DIAMOND and the ability to generate the exhaustive set of nuclear reactor cooling pool behavior scenarios. The results supported the ability of DIAMOND definitions to be used to organize and structure knowledge about SFP's normal and off-normal behaviors. The SPF example showed the application of assets, actions, and triggers from DIAMOND to events and relationships in MP. Assets and actions were represented as MP events, and triggers were represented as precedence relations between MP events. This thesis research verified the DIAMOND ontology was implemented correctly in the model from data representative of operationally realistic behavior and the modeling results validated the MP behavior model was well constrained.

**14. SUBJECT TERMS**
Idaho National Lab, INL, behavior modeling, ontology, Monterey Phoenix, MP, Spent Fuel Pool, SFP, Light Water Reactor, LWR, Data Integration Aggregated Model and Ontology for Nuclear Deployment, DIAMOND

**15. NUMBER OF PAGES**
99

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

# ANALYSIS OF DOMAIN-SPECIFIC NUCLEAR ONTOLOGY USING MONTEREY PHOENIX BEHAVIOR MODELING

Landa R. McClure
Civilian, Department of the Air Force
MBA, Weber State University, 2016

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN SYSTEMS ENGINEERING MANAGEMENT**

from the

**NAVAL POSTGRADUATE SCHOOL**
**March 2022**

Approved by: Kristin M. Giammarco
Advisor

Douglas L. Van Bossuyt
Second Reader

Oleg A. Yakimenko
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Current nuclear energy ontologies are known to lack a common vocabulary to formally verify nuclear energy data relationships for modeling system behaviors. Idaho National Laboratory (INL) developed the Data Integration Aggregated Model and Ontology for Nuclear Deployment (DIAMOND) ontology to provide a standard vocabulary and taxonomy for identifying data relationships in nuclear energy system models. This thesis conducted an analysis of DIAMOND using a Spent Fuel Pool (SFP) Monterey Phoenix (MP) behavior model. The SFP MP behavior modeling application demonstrated components of and interactions among a spent fuel cooling pool and its environment. The MP behavior model demonstrated a viable approach for analyzing nuclear reactor system behavior consistent with DIAMOND and the ability to generate the exhaustive set of nuclear reactor cooling pool behavior scenarios. The results supported the ability of DIAMOND definitions to be used to organize and structure knowledge about SFP's normal and off-normal behaviors. The SPF example showed the application of assets, actions, and triggers from DIAMOND to events and relationships in MP. Assets and actions were represented as MP events, and triggers were represented as precedence relations between MP events. This thesis research verified the DIAMOND ontology was implemented correctly in the model from data representative of operationally realistic behavior and the modeling results validated the MP behavior model was well constrained.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS


I. INTRODUCTION....1
A. RESEARCH QUESTIONS....2
B. RESEARCH METHODOLOGY....2
C. BENEFITS OF RESEARCH....3
D. THESIS ORGANIZATION....4

II. LITERATURE REVIEW....5
A. NUCLEAR REACTOR DISASTERS....5
B. LIGHT WATER REACTORS (LWR)....7
C. ONTOLOGIES AND THE ADVENT OF DIAMOND....10
D. MONTEREY PHOENIX....14
E. CHAPTER SUMMARY....18

III. APPLICATION OF DIAMOND ONTOLOGY USING MONTEREY PHOENIX....19
A. STRUCTURE OF DIAMOND ONTOLOGY....19
1. Structural Layout Methodology....20
2. Types of Elements....21
B. CROSSWALK OF DIAMOND AND MONTEREY PHOENIX....26
1. Ontology Scope Crosswalk....26
2. Object and Relation Crosswalk....27
3. Multiplicity Crosswalk....37
4. Control Flow Crosswalk....40
C. SPENT FUEL POOL (SFP) BACKGROUND....42
D. DEMONSTRATION OF SPENT FUEL POOL MP BEHAVIOR MODEL....45

IV. CONCLUSIONS AND RECOMMENDATIONS....65

APPENDIX. SPENT_FUEL_COOLING_AND_CLEANUP_SYSTEM MP CODE....69

LIST OF REFERENCES....73

INITIAL DISTRIBUTION LIST....77

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF FIGURES


Figure 1. Light water reactor system. Source: Withgott and Brennan (2008). .....9

Figure 2. Fissioning process. Source: Ochiai (2013). .....10

Figure 3. Ontology spectrum from informal to formal. Source: Gruber (1992). .....11

Figure 4. MP-gryphon architecture decomposition. Source: Desai (2021).....16

Figure 5. A view of DIAMOND within the Protégé editor. Source: Al Rashdan, Browning, and Ritter (2019). .....22

Figure 6. Reciprocal relationships designed in DIAMOND. Source: Al Rashdan, Browning, and Ritter (2019). .....24

Figure 7. Screenshot of domain references in the DIAMOND ontology. Source: DIAMOND (2019). .....27

Figure 8. DIAMOND ontology screen shot of class mapped to MP root event and MP behavior model code mapped to DIAMOND ontology. Adapted from DIAMOND (2019) and MP-Firebird (2022).....28

Figure 9. An event has two basic relations: precedence and inclusion. Source: Giammarco (In Press, Chap 3).....29

Figure 10. Example MP schema for root, composite, and some atomic events mapped to event trace flow. Source: MP-Firebird (2022). .....30

Figure 11. Example grammar rules with corresponding event traces. Source: Auguston (2020). .....31

Figure 12. Event trace example showing root, composite, and atomic events in a general supply chain model. Adapted from MP-Firebird (2022).....32

Figure 13. Illustration of object property entities in DIAMOND. Source: DIAMOND (2019).....34

Figure 14. Left: example schema and right: event trace (scope 2 trace 3) illustrates the adding of a precedence relation between events from different roots using a COORDINATE statement. Source: Example 2 on MP-Firebird (2022).....35

Figure 15. Schema and event trace to illustrate event sharing via COORDINATE statement with SHARE. Source: MP-Firebird (2022). ..... 36

Figure 16. Schema and event trace to illustrate event sharing via SHARE ALL statement, which abbreviates the COORDINATE statement with SHARE. Source: Example 3 on MP-Firebird (2022). ..... 36

Figure 17. Nested COORDINATE statements establish many-to-one and one-to-many relationships. Source: MP-Firebird (2022). ..... 38

Figure 18. The SHARE ALL statement establishes an inclusion relation between two root events. Adapted from MP-Firebird (2022)..... 39

Figure 19. Schema of event grammar rules for one or more {+…+} UAV events. Adapted from MP-Firebird (2022). ..... 40

Figure 20. Screenshot for accounting for “trigger” in DIAMOND. Source: DIAMOND (2019)..... 41

Figure 21. Event trace flow of user-defined relations used to indicate input/ outputs in MP ..... 42

Figure 22. Spent fuel basin at Idaho National Laboratory. Source: INL.gov (2022). ..... 43

Figure 23. Irradiated fuel storage facility at Idaho National Lab. Source: INL.gov (2022). ..... 43

Figure 24. Depiction of light water reactor spent fuel pool purpose Source: UCS (2016). ..... 44

Figure 25. Basic flow path of spent fuel pool cooling and cleaning system. Source: USFRC HRTD (Rev 0109). ..... 45

Figure 26. Identification of MP root events from elements of the spent fuel pool cooling and cleaning system are shown with MP schema labeled mapped to system components diagram. Source: USNRC HRTS 14.4 (Rev 0109). ..... 46

Figure 27. Illustration and labeling of key features of MP behavior model in MP-Firebird. Adapted from SFP.mp (2022). ..... 48

Figure 28. SFP model schema name declaration (Line 17). Source: SPF.mp (2022). ..... 49

Figure 29. Spent_Fuel_Pool_Cooling_System roots, composites, and atomic events. Source: SPF.mp (2022).....50

Figure 30. Constraints on the spent fuel pool and other components. Source: SPF.mp (2022). .....51

Figure 31. Additional constraints on the spent fuel pool and other components in the case of low water level. Source: SPF.mp (2022). .....52

Figure 32. Example nominal behavior illustrating the modeling of a complex nuclear spent fuel pool cooling and purification system MP event trace. Green boxes are root events (actors), orange boxes are composite events, and blue boxes are atomic events. Dashed arrows are inclusion (decomposition) relations and solid arrows are precedence relations. Source: SPF.mp (2020). .....53

Figure 33. If event trace scenario found water level low from spent fuel pool, operator is notified by alarm activation and water is added from purification subsystem. SFP Cooling and cleaning model event trace scenario for water levels at normal and temperature high triggering high temperature alarm. Source: SPF.mp (2022).....55

Figure 34. SFP cooling and cleanup model event trace for water levels low and temperature normal triggering low water alert. Source: SPF.mp (2022). .....56

Figure 35. SFP cooling and cleanup model event trace for water at critically low levels triggering low water alarm and high temperatures triggering high temperature alarm. Source: SPF.mp (2022).....57

Figure 36. SFP cooling and cleanup model event trace for water levels at normal and temperature at normal, no action needed and normal monitoring continues. Source: SPF.mp (2022). .....58

Figure 37. SFP cooling and cleanup model event trace for water levels low and temperature high triggering high temperature alarm and low water alert. Source: SPF.mp (2022).....59

Figure 38. SFP cooling and cleanup model event trace for water levels critically low and temperature normal triggering low water alarm. Source SPF.mp (2022). .....60

Figure 39. Sample DIAMOND ontology data. Source: DIAMOND (2019).....61

Figure 40. The mapping illustration shows the visual correlations between assets, actions, and triggers from DIAMOND to events and relationships in MP. Assets and actions are mapped to MP events,

and triggers are mapped as precedence relations between MP events. The concepts of the correlating domain-specific references in the DIAMOND ontology to the concepts of the Monterey Phoenix spent fuel pool cooling and cleaning behavior model domain-specific scenario are shown. The purpose of the visual representation is for showing the analogous concepts between MP and DIAMOND visually by color mapping.....62

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| AI | Artificial Intelligence |
| BFO | Basic Formal Ontology |
| BPMN | Business Process Modeling Notation |
| CIS | Critical Infrastructure Sectors |
| CRP | Coordinated Research Project |
| DIAMOND | Data Integration Aggregated Model and Ontology for Nuclear Deployment |
| DHS | Department of Homeland Security |
| DOD | Department of Defense |
| FFBD | Functional Flow Block Diagram |
| HTML | Hypertext Markup Language |
| IAEA | International Atomic Energy Agency |
| IN | Inclusion |
| INL | Idaho National Lab |
| INSAG | International Nuclear Safety Group |
| LML | Life cycle Modeling Language |
| LWR | Light Water Reactor |
| LWRS | Light Water Reactor Sustainment |
| MITRE | Massachusetts Institute of Technology Research Engineering |
| MP | Monterey Phoenix |
| NEA | Nuclear Energy Agency |
| NPP | Nuclear Power Plant |
| NPS | Naval Postgraduate School |
| OWL | Web Ontology Language |
| PPD-21 | Presidential Policy Directive 21 |
| PRECEDES | Precedence |
| PSA | Probabilistic Safety Assessment |
| RDX | Resource Description Framework |
| SFP | Spent Fuel Pool |
| SGML | Standardized Generalized Markup Language |

| | |
|---|---|
| SoS | System of Systems |
| SysML | Systems Modeling Language |
| TECHDOC | Technical Document |
| TMI | Three Mile Island |
| UML | Unified Modeling Language |
| U.S. NRC | United States Nuclear Regulatory Commission |
| ValVer | Verification and Validation |
| W3C | World Wide Web Consortium |
| XML | Extensible Markup Language |
| XSD | XML Schema Definition Language |

# EXECUTIVE SUMMARY

Analysists and researchers access and share nuclear reactor power plant data in various scattered forms without having any structured continuity. Idaho National Lab (INL) has created Data Integration Aggregated Model and Ontology for Nuclear Deployment (DIAMOND) for use in modeling system behaviors and for uncovering and predicting patterns for further analysis (Al Rashdan, Browning, and Ritter 2019). The DIAMOND ontology acts as the fundamental domain-specific data model enabling integrating data sources. Idaho National Lab's creation of the DIAMOND ontology represents an approach by the nuclear community to organize and structure knowledge about previous nuclear reactor system failures as a tool to avoid the repetition of the failures in other systems and to leverage applications into other modeling activities (Al Rashdan, Browning, and Ritter 2019).

The purpose of this thesis is to conduct an analysis and evaluation of the recently developed INL Nuclear Reactor domain-specific DIAMOND ontology in the context of Monterey Phoenix (MP) behavior modeling to verify and validate a scoped system application to identify any potential gaps or overlaps that may occlude formally modeling nuclear reactor behavior. The scoped scenario selected for this thesis is the Nuclear Reactor Spent Fuel Pool (SFP) cooling and cleaning system of the nuclear power plant energy system process.

The contribution and benefit of this thesis is threefold. First, it provides a validation and verification of behavior-related elements in an INL developed ontology for nuclear reactors. Second, it demonstrates a MP application as a different approach for analysis framework which involves scope-complete analysis of nuclear reactor process for predicting and monitoring system behavior. Finally, it demonstrates an application of the INL ontology directly in the context of the Monterey Phoenix behavior modeling tool.

In this research, the nuclear energy reactor DIAMOND ontology provided by INL, and the MP Cooling Pool Behavior Model developed as part of a sponsored research project at Naval Postgraduate School (NPS) are used as source data. This analysis maps

applicable parts of the DIAMOND nuclear energy reactor ontology to the MP nuclear reactor spent fuel pool cooling and cleaning behavior model developed by NPS with input from nuclear INL subject matter experts.

This analysis research first surveys the public literature for applications of past nuclear energy behavior models or defined ontology. Next, it analyzes and evaluates INL's domain-specific nuclear reactor from viewpoint of behavior ontology aspects and validation. Then finally, the INL DIAMOND ontology maps and tests with the NPS MP Cooling Pool behavior model.

A crosswalk between DIAMOND and MP reveals four behavior-related areas of comparison between the system concepts of DIAMOND and those defined in MP. First; the ontology scope for DIAMOND is defined by classes, properties, and relationships that frame an expandable nuclear specific domain of system data. MP's ontology scope is limited to classes that pertain to behavior but can be applied to any system domain. Secondly; the DIAMOND ontology is constructed from classes that represent a wide range of objects and the relationships between them. The MP behavior model is based on an abstract concept of event that can represent a range of DIAMOND concepts (e.g., asset or action), and precedence, inclusion, and user-defined relations. Third; the current DIAMOND ontology does not make a distinction for optional multiplicity of zero or more relationships and assumes there is always at least one existing relationship. Monterey Phoenix behavior modeling includes zero-or-more relationships for event iterations in addition to one-or-more relationships. And finally, the control flow method for identifying user-defined relations is defined by a "trigger" class which serves the role of Inputs/Outputs in DIAMOND. Monterey Phoenix does not have a fundamental class for information elements of Input/Output or a special property of "trigger," rather all concepts are modeled as events, even data, in terms of operations on the data, and the concept of trigger is implemented using precedence relationships.

The thesis research verified the DIAMOND ontology was implemented correctly in the model from data representative of operationally realistic behavior and validated by the modeling results of the MP behavior model that the SFP Cooling and Cleaning MP behavior model was well constrained. The results of the MP event traces verified the

modeling schema and constraints. Monterey Phoenix behavior modeling has proven a nuclear reactor energy behavior modeling application with the ability to model behavior concepts from system data concepts from the DIAMOND ontology.

Recommendations for future work from this research for INL and other communities of interest is for creation of other scope-complete scenarios of other system processes and to observe and inspect unconstrained cases for unexpected emergent behavior and unwanted behaviors to better state for ideas about of vulnerabilities in the system of systems.

## References

Ahmad Al Rashdan, Jeren Browning, and Christopher Ritter. 2019. Data Integration Aggregated Model and Ontology for Nuclear Deployment (DIAMOND): Preliminary Model and Ontology. INL/EXT-19-55610.

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

With immeasurable gratitude, I express my appreciation to my advisor, Dr. Kristin Giammarco, for her immense encouragement, guidance, and passion for deep learning interlaced with a willingness to share of her kindness, knowledge, and talents.

I would also like to recognize Dr. Douglas Van Bossuyt for his time and input, Dr. Wally Owen for the life-changing opportunity for participation in the Systems Engineering Management program, and Dr. Gary Langford for his encouraging influence to keep learning at a richer level of thinking.

I also send my heartfelt love and admiration to my husband, Brian Williams, for his infinite support and understanding. I am forever grateful for his unconditional love, laughter, and patience throughout the thesis process and throughout our every day and every circumstance.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

The nuclear community can consistently benefit from the ability to formally model nuclear reactor system behaviors. Existing ontologies specific to nuclear energy systems currently express the knowledge in different data languages. As the need for assimilation of nuclear reactor data arises, researchers and decision-makers are conducting data modeling manually and independently. Without a common definition of data relationships, lessons learned from one data modeling activity cannot easily leverage applications into other modeling activities.

According to the U.S. Department of Homeland Security (DHS), these system behaviors belong to the critical infrastructure class of systems and demand high safety and security. Typically, analysists using aggregate data and knowledge about existing or previous nuclear reactor system designs must individually access the various forms of scattered documentation without having any queried uniform structure. Researchers in the nuclear community are currently using ontologies to provide a framework to represent and perform queries on the collected data. To help analysists and researchers access and share nuclear reactor power plant data, Idaho National Lab (INL) has created Data Integration Aggregated Model and Ontology for Nuclear Deployment (DIAMOND) for use in modeling system behaviors and for uncovering and predicting patterns for further analysis. Idaho National Lab developed the domain-specific ontology to aid in providing a standard vocabulary and taxonomy for identifying data relationships in behavior models of nuclear reactor systems. Ontologies are defined as “explicit formal specification(s) of the terms in a domain and the relationships among them” (Gruber 1992, 199).

The focus of this thesis is the DIAMOND ontology in the context of behavior modeling. The DIAMOND ontology represents an approach by the nuclear community to organize and structure knowledge about previous nuclear reactor system failures as a tool to avoid the repetition of the failures in other systems. The purpose of this thesis is to test a scenario application of behavior concepts of the DIAMOND ontology with a Monterey Phoenix (MP) behavior model. The scenario selected for this thesis is the Nuclear Reactor Spent Fuel Pool (SFP) of the nuclear power plant energy system process to aid in

identifying any potential gaps or overlaps that may occlude formally modeling nuclear reactor behavior. Monterey Phoenix is a language, approach, and open-source tool for modeling and simulating system and process behavior developed by the Navy (Giammarco 2017).

The main deliverables of this thesis are the findings from the analysis and evaluation of the domain-specific nuclear ontology regarding the evaluation of an application of an MP behavior model of the NPS-developed SFP scenario.

## A. RESEARCH QUESTIONS

This thesis investigates potential gaps or overlaps that may exist in the DIAMOND ontology when evaluating with an application of MP modeling for nuclear reactor behavior.

Two related research questions focus on how DIAMOND is used with executable behavior models via the MP example, and how DIAMOND does or does not provide sufficient coverage to implement in MP behavior models.

1. What gaps or overlaps exist in the DIAMOND Ontology when used as a basis with MP modeling of nuclear reactor behavior?
2. How can DIAMOND be appropriately scoped to provide coverage for evaluating executable MP behavior models?

## B. RESEARCH METHODOLOGY

The basic strategy of this research methodology is to first gather information about related concepts of systems: example nuclear reactor behaviors, foundational concepts of ontologies, and example MP behavior modeling applications. The second step is to map applicable data from the DIAMOND ontology to the MP framework to look for potential gaps, overlaps, or other challenges at the data structure level. The next step is to conduct an evaluation of a specific MP behavior model of a light water nuclear power plant cooling pool. The final step of the research to provide a summary of the results and recommendations for further research and applications.

An MP model of nuclear reactor scenarios created by NPS using source data from public literature and from consulting with nuclear experts on the NPP scenarios is used to evaluate the DIAMOND ontology. A nuclear reactor cooling pool behavior model developed for INL on a joint project with NPS is chosen as the focus for the evaluation. The selected cooling pool scenario provides a representative of the nuclear energy process of interest to INL.

The analysis maps applicable parts of the DIAMOND nuclear energy reactor ontology to the MP nuclear reactor spent fuel cooling pool behavior model developed by NPS with input from nuclear INL subject matter experts. Results from the Monterey Phoenix evaluation scenario revealing any issues with the ontology or system model are taken to the expert panel to ensure that the scenarios are representative of operationally realistic behavior, and that the ontology is implemented correctly in the model.

## C. BENEFITS OF RESEARCH

The intended contribution and benefits of this thesis for the nuclear community are threefold: first, identification of any potential gaps or overlaps that may occlude implementation and further analysis of nuclear reactor behavior models; second, demonstration of MP's viability as a scenario generation approach for analyzing nuclear reactor behavior; and finally, a scoped description of how to apply the DIAMOND ontology to an MP model in a light water nuclear spent fuel cooling pool scenario.

This research also highlights some advantages or challenges when using the DIAMOND ontology in the context of behavior modeling tools such as MP. Additionally, this thesis will provide INL with a scenario to validate INL's ontology at work in the MP behavior model for applications in other nuclear reactor processes.

Besides MP being of interest to INL for formal behavior modeling, this research supports the multi-disciplinary field of behavior modeling with MP's ability to capture or expose emergent behavior, both expected and unexpected, with an application of DIAMOND.

This thesis assists INL and others in the nuclear energy community to understand how to use MP behavior models to increase awareness of its potential application to analyzing emergent nuclear reactor behaviors. In parallel, an open-source behavior modeling tool provides reliability and flexibility to the end user.

## D. THESIS ORGANIZATION

This thesis organizes the research into chapters as follows. Chapter I introduces the research question, research methodology, and the benefits of conducting the research. Chapter II surveys the public literature for the recorded contributing failure factors of historical disaster events in nuclear reactor power plant systems, the fundamental concepts of ontology, and samples of MP behavior modeling applications related to communicating lessons learned from one system or site to another. Chapter III analyzes the DIAMOND ontology for the structure of data that enables consistency of storage and evaluates an application of MP behavior modeling of a light water reactor (LWR) nuclear power plant (NPP) spent fuel pool (SFP) scenario using elements of the DIAMOND ontology pertinent to the desired behavior. Chapter IV concludes with the results and discoveries concerning the DIAMOND ontology as tested with the SFP scenario application of MP behavior model. The conclusion chapter also provides descriptions of further research and additional applications of MP behavior modeling in nuclear reactor systems for the community of interest.

# II. LITERATURE REVIEW

This chapter first surveys public literature for reports of historical nuclear disasters with contributing failure factors and reviews literature for an overview of the nuclear light water reactor process. Next, the chapter surveys ontologies in literature for development of structural concept timelines and for other tools the nuclear community may use for identifying and sharing behaviors. And finally, this chapter searches public literature for the structural and functional basis of Monterey Phoenix (MP) and for recorded MP behavior modeling applications for discovering emergent behaviors.

## A. NUCLEAR REACTOR DISASTERS

Most, if not all, of the historical nuclear accidents can be traced to unexpected or unwanted system behaviors related to a combination of factors of technology, people, or the environment. The top three of the largest nuclear reactor disasters with these recorded failure factors include the Three Mile Island Disaster, Chernobyl Disaster, and Fukushima Dai'ichi Power Plant Nuclear Disaster.

Disaster investigators record historical nuclear reactor events as stemming from combinations of failure factors such as reactor design, human error, or the lack of a sustained safety culture. For example, in October 1979, Executive Order 12130, "The Accident at Three Mile Island," established Three Mile Island (TMI) to be the result of malfunctions from a sequence of events of one of the cooling systems reactors. The President's Commission report recorded the investigation and study of TMI from March 28, 1979, at the plant near Harrisburg, Pennsylvania to have experienced a mechanical or electrical failure of components of the piping portion of the plant's primary system, which then led to the full reactor collapse. When the cooling water pump stopped operating, the pressure and temperature continued increasing in the reactor, triggering the pressure relief valve to open in accordance with design function (NRC 1979). Water and steam began to flow out of the reactor, however, as the pressure returned to normal, the pilot-operated pressure relief valve did not properly close, consequently allowing cooling water that was

critically needed to cover and cool the fuel core, to continue escaping from the reactor system. Executive Order 12130 stated,

"The major factor that turned this incident into a serious accident was inappropriate operator action, many factors contributed to the action of the operators, such as deficiencies in their training, lack of clarity in their operating procedures, failure of organizations to learn the proper lessons from previous incidents, and deficiencies in the design of the control room."

This combination of mechanical failures further contributed to the human error of the plant staff being unaware that steam from the cooling was pouring out of the open valve, despite the alarms and warning lights flashing, indicating overheating and low levels of coolant. As a result, the plant staff made assumptions that the core was properly covered with water at the desired high-pressurized high-water level. Unfortunately, without the reactor coolant pumps circulating water, the primary system lost thousands of gallons of emergency cooling water, leading to the full reactor collapse.

The second example is the widely recognized Chernobyl disaster. According to the INSAG-1 report of the Chernobyl accident investigation of April 26, 1986, in Ukraine, the former Soviet Union, the commission team reported,

> the accident was caused by a remarkable range of human errors and violations of operating rules in combination with specific reactor features which compounded and amplified the effects of the errors and led to the reactivity excursion… The operators deliberately and in violation of rules withdrew most control and safety rods from the core and switched off some important safety systems. (World Nuclear Association. [WNA] 2020)

Other records also report the cause to be a combination of design flaw, operator error, and lack of safe environment contributing to the defective reactor design. In the Chernobyl Post-Accident Review Meeting, the International Nuclear Safety Advisory Group agreed with Soviet experts that several operator's role in the Chernobyl accident "compounded and amplified the effects of the errors and led to the reactivity excursion" (INSAG-1 1986). The INSAG-1 investigation recorded that prior to the accident, a sudden surge of power during reactor systems test procedure initially led to the destruction of one of the reactor units at the nuclear power station (INSAG-1 1986). The investigation also

states that "one operator may have inadvertently manually pulled the 84-pound central control rod out approximately 26 inches rather than the 4 inches required as part of maintenance procedures" (INSAG-1 1986). An updated report from the IAEA's 1992 INSAG-7, "The Chernobyl Accident: Updating of INSAG-1" in 1991 by the State Committee, also reiterates the major factors stemming from the unstable behavior of the reactor and non-defined safety procedures for the operators.

For the third example, investigation reports conclude the Fukushima Dai'ichi Power Plant Nuclear Disaster to be a result of a series of equipment failures resulting from the Tohoku Tsunami on 11 March 2011 following the Honshu Island 9.0-magnitude earthquake of Japan. (World Nuclear Association [WNA] 2021). The tsunami triggered by the earthquake damaged many of the generators and battery backup systems causing Fukushima Dai-ichi reactors to lose all power (U.S. NRC 2018). Three of the units continued operating for several hours on backup safety systems before the systems eventually reached failure and the reactors overheated and melted the cores (Kelly 2015).

These and other past historical events in the nuclear domain prompt great interest in transferring design information, as well as related best practices and lessons learned, from one system or site to another. As noted by the U.S. NRC guidance and regulations, nuclear reactor power plants system behaviors remain a critical component for regulation and sustainment of safety operational standards (United States National Regulatory Commission [NRC], n.d.).

## B. LIGHT WATER REACTORS (LWR)

The nuclear reactor energy scenario selected for the purpose of analysis and evaluation for this thesis is a Light Water Reactor (LWR) Nuclear Power Plant (NPP) Spent Fuel Pool (SFP) of the Idaho National Lab nuclear power plant energy system process. Light water nuclear spent fuel pool systems are part of the sixteen identified national infrastructure of safety critical systems. Presidential Policy Directive 21 (PPD-21) identifies these as critical systems whose incapacitation or destruction of assets and systems would have a debilitating effect on security and safety to the United States Critical Infrastructure Sectors (CIS). With nuclear energy as one of the national safety critical

infrastructures, it is essential to monitor nuclear power plant system behaviors and patterns for further analysis.

Nuclear power plants consist of a nuclear reactor core that provides a heat energy source harnessed under pressurized water and then piped to another water supply to generate steam for spinning a turbine which produces the nuclear power. Coolant water is then used for the heat exchange and thermalization of the neutrons. The Light Water Reactor System process for producing nuclear energy is summarized in the depictions and descriptions of Figure 1. Unlike a fossil-fueled plant, a nuclear reactor plant releases heat energy through the process of fissioning, also known as splitting of uranium-235 nucleus atoms, to generate the steam.

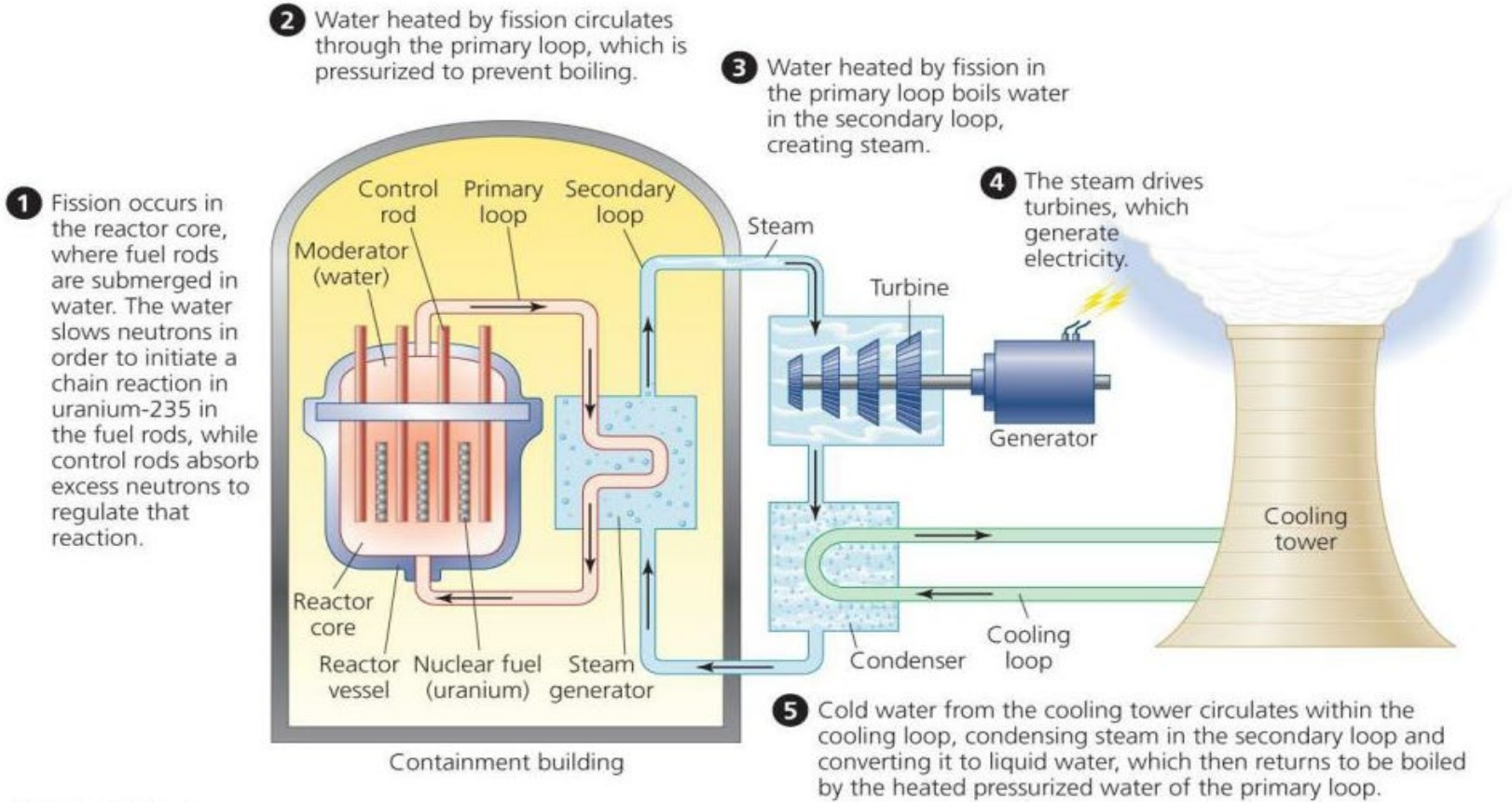


Figure 1. Light water reactor system. Source: Withgott and Brennan (2008).

During the fissioning process, the uranium atom splits and releases 200 MeV energy to generate clean energy. An illustration of the fissioning process is depicted in Figure 2. Uranium-235 is the preferred reactor fuel for nuclear power plants due to a known easier ability to initiate the fission chain reaction to produce the elevated temperature pressure (Harman 2018).

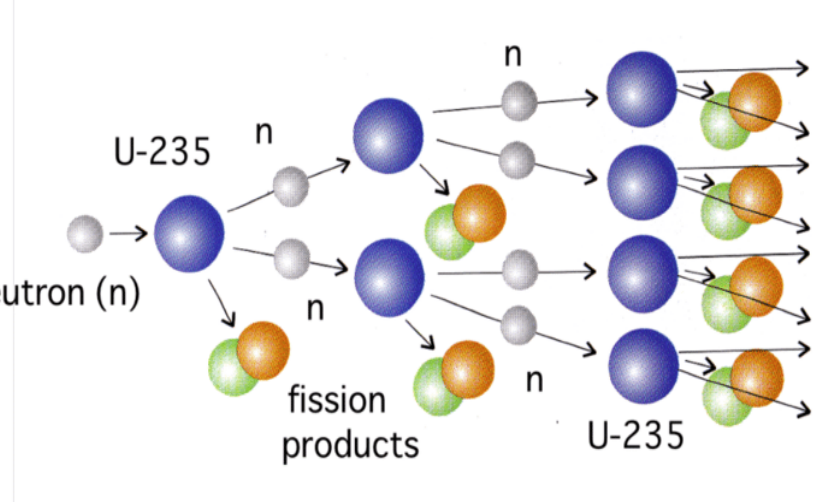


Figure 2. Fissioning process. Source: Ochiai (2013).

## C. ONTOLOGIES AND THE ADVENT OF DIAMOND

Many interpretations of ontology exist in public literature; a frequently used definition of an ontology is "explicit formal specification(s) of the terms in a domain and the relationships among them" (Gruber 1992, 199). Ontologies exist in many various forms depending on the need of the intended specific application of data. The development for structure and progression of ontologies continues over time, moving from singular informal states towards more advanced formal states. An ontology spectrum timeline is shown in Figure 3 to illustrate instances of growth from informal basic organization of data towards more formal structures.

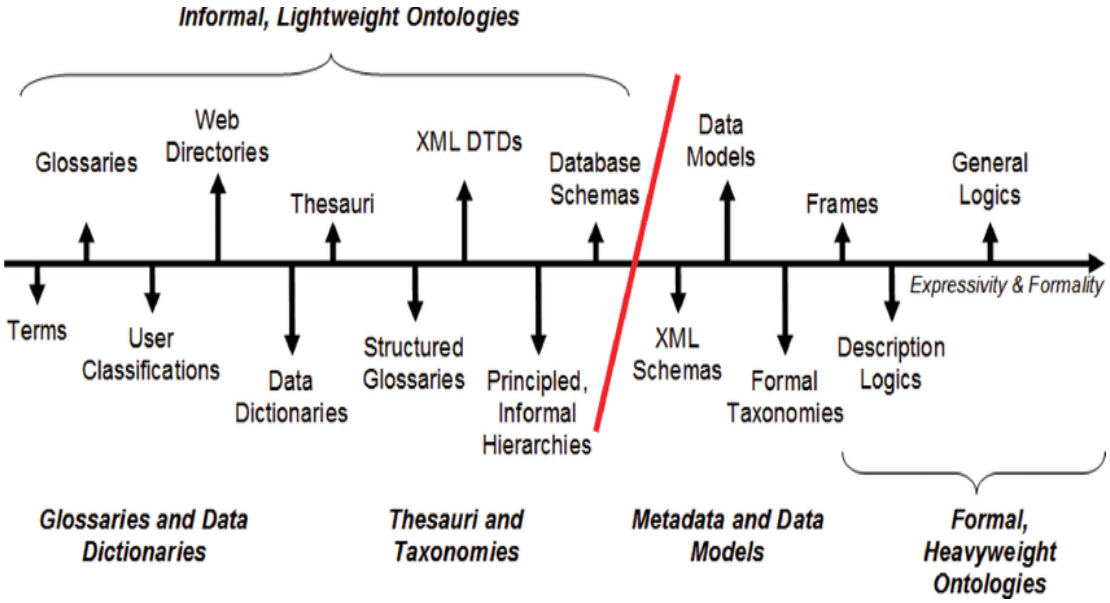


Figure 3. Ontology spectrum from informal to formal. Source: Gruber (1992).

The simplistic form of an ontology began as predefined keywords, glossaries, and taxonomies used to control terminology. The modern ontology history is credited with beginning with the research of Artificial Intelligence (AI);

> In philosophy, one can talk about an ontology as a theory of the nature of existence. In computer and information science, ontology is a technical term denoting an artifact that is designed for a purpose, which is to enable the modeling of knowledge about some domain, real or imagined. (Gruber 1992)

Later, these basic ontologies began being used to classify entities or objects. This basic established vocabulary supported characterizing objects and processes comprised within the domain. During the 1970s, further progression of ontologies led to adding vocabulary tags to the text to become known as Standardized Generalized Markup Language (SGML) (Anderson 2004). After this text markup language was discovered to be too complex for routine use, taxonomy subsets known as Hypertext Markup Language (HTML) and Extensible Markup language (XML) were developed by the late 80s (Raggett et al. 1998). As the ontology progression continued, the XML subset was created to be more flexible in supporting schemas for independent data after HTML applications were

found to be inefficient for data storage and interchange. Further still, the Unified Modeling Language (UML) was initially created in 1994 by Rational Software to standardize unrelated systems and offer a novel approach to visualize software designs. In 2002, Barry Smith and Pierre Grenon developed the Basic Formal Ontology (BFO) for use in supporting information retrieval, analysis, and integration. A more recent development for data sharing among communities of interest is the Life cycle Modeling Language (LML), which was developed with experience gained from predecessor languages UML and Systems Modeling Language (SysML). Formal types of ontologies such as LML focuses on providing a structure and language that can be used to facilitate stakeholder communication throughout the product (Life Cycle Modeling Specification 2017). The SysML created in 2003 is a modeling language that provides semantics and notation. Formal ontologies such as SysML were a critical enabler for model driven systems engineering. SysML 2.0 is under development in 2022 to integrate timing and other concepts that are not present in earlier versions of SysML.

The need to formalize content and to express scientific domain knowledge in a traceable computer language contributed to the impetus to further develop ontologies over time (Arp, Smith, and Spear 2015).

A formal ontology defines common vocabularies among communities of interest and ensures a common understandings and reuse of domain knowledge. Ontologies also provide machine-interpretable definitions of basic concepts and relationships (Noy and McGuiness 2001). As a result, formal ontologies present a useful tool as sharable and reusable knowledge representation about a domain.

One of the most common goals for ontology development is for sharing a common structure of knowledge (Musen 1992; Gruber 1992). Domain-specific ontology environments, such as DIAMOND for supporting nuclear reactor energy communities, are developed through three key components. These three components can be visualized by the corresponding ontology progression timeline. First, this progress of key component elements starts with a controlled vocabulary or glossary; second, then progresses to further defined definitions, and finally, connections are then formed by the categorized relationships between them. This iterative ontology development process consists of

defining instances of classes, interconnections, and refined objects and attributes extending the knowledge base. Dependent upon the need of the ontology application, the selected structural layout for building an ontology determines the mixture of classes, properties, restrictions, and annotations needed for communicating the connections.

In most applications, developers define the use of the term ontology as the formal descriptions of domain knowledge as a set of concepts and the interconnected relationships (Lexico 2022). Likewise, for this purpose of SE research, the DIAMOND ontology concept can be expressed as a knowledge base of structured content comprised of representational vocabulary descriptions that exist in the specific nuclear domain. The DIAMOND ontology expresses known data types of structural specification and provides definitions of classes, relationships, and attributes to share nuclear knowledge among a domain-specific community of interest.

An ontology often uses the instances of its classes as the foundation of the knowledge base (Noy and McGuinness 2001). This type of ontology architecture is known to also serve as support for retrieval of data and reasoning of different classes. To enable ontology descriptions, components are used to formally specify common interpretable definitions for basic concepts in the domain (Noy and McGuinness 2001). Ontologies are then easily expanded and allow for adding new concepts with the connected requirements. The data relationships can then connect to relatable data among them, with the properties of each concept describing various attributes and classes describing concepts in the domain of the ontologies (Noy and McGuinness 2001).

Other formal means of communicating nuclear reactor knowledge and lessons learned come from shared databases accessible by the community of interest. Various structures of shared nuclear data exist in various isolated forms and systems. The need for a domain-specific nuclear research reliability database for use in Probabilistic Safety Assessment (PSA) has been internationally needed for many years (IAEA-TECDOC-478, 1988). Efforts from the early 1990s by the International Atomic Energy Agency (IAEA) Coordinated Research Project (CRP) resulted in the publication of Generic Component Reliability Data for Research Reactor PSA (IAEA TECDOC-930 1997). This technical documentation provides analysists with “updates (to) information in the TECDOC-930 and

has a broader scope that provides information on a wider range of issues pertaining to reliability data for research reactor PSA. Accordingly, in addition to component reliability data, the TECDOC provides information related to preparation and application of data on initiating events, human reliability, common cause failures…" (TECDOC 930). The database also provides guidance on the application of the reliability data for research of NPP probabilistic safety assessments. The use of selected methods and analytical tools vary significantly among applications while working to address many of the same issues and models.

Ontologies also provide the nuclear community of interest a tool for an approach to implement lessons learned from nuclear reactor data. One of the main challenges identified by INL is the integration of data sources to aid the nuclear industry to model system behaviors (Al Rashdan, Browning, and Ritter 2019). An INL development goal of the DIAMOND ontology was to enable knowledge integration with other nuclear community applications (Al Rashdan, Browning, and Ritter 2019).

Idaho National Laboratory's main purpose in developing the DIAMOND ontology began with a first-phase goal to create a model to store and relate information formally with deliberate reusability. Use of star topology to integrate four NPP data sources further exposed the need for an ontology and data modeling (Al Rashdan, Browning, and Ritter 2019). The initial INL DIAMOND ontology development is a next phase approach to addressing three identified challenges with NPP data. These challenges include having to access data independently, having data residing in multiple forms, and data being stored on multiple systems (Al Rashdan, Browning, and Ritter 2019).

## D. MONTEREY PHOENIX

A known challenge for system's architecture is predicting emerging behaviors of a system of systems. Monterey Phoenix (MP) provides an innovative approach in the detection, prediction, classification, and control of emergent system of system (SoS) level behaviors (Giammarco and Giles 2018).

The development timeline of MP began with Dr. Mikhail Auguston of the Department of Computer Sciences of U.S. Navy Naval Postgraduate School (NPS), who,

following decades of research and development and pursuit of user-friendly lightweight formal methods, optimized a software system (See Figure 4 Block 1.3) for generating exhaustive sets of event traces from formal specifications of behavior up to a user-defined scope.

In the early years of MP, there was no graphical user interface, and trace generation took place through a terminal window with trace outputs being read as a text file. Clifford Whitcomb of the NPS Systems Engineering Department identified the application and value of this capability to the systems engineering field and involved his doctoral student at the time, Kristin Giammarco, in examining these applications in 2010. Under her direction, Philip McCullick and Michael Nigh, also of NPS, in 2015 created the graphic user interface and web application known as MP-Firebird, which takes MP model input from the user through a text editor, uses Auguston's trace generator to compile the model and compute the trace outputs, and returns the results to the user as graphs. MP-Firebird was inspired by the first MP implementation prototype that was also a web application. Eagle6, created by Dr. Auguston's doctoral student Joey Rivera, was a key advancement made by the early versions of MP-Firebird in user-friendliness and additionally the implementation of a sequence diagram-like layout for the event traces, which many systems engineers easily recognize. The current version 4 of the MP-Firebird tool, which includes support for user-defined relations and event attributes, was used for this research.

Since MP-Firebird was implemented, it began to attract users, many of whom went on to develop examples of Monterey Phoenix's ability to generate scope-complete sets of behavior scenarios that contained far more behavior variants than the typical tools used in the systems engineering industry. MP provided a means to further define architecture models without departing from existing methods, representations, or tools. This method of behavior modeling supported a deeper learning of the complexity of a system by exposing non-obvious behaviors that may be hidden beneath the surface of a design component. This capability resulted in the employment of "MP to detect, predict, classify and control emergent behaviors" (Giammarco 2019). The MP modeling tool provides multidisciplinary or domain-specific communities a comprehensive ability to transform domain language inputs of a system or systems provided by the user into graphs for ease of communicating the identifying behaviors and interactions among them (Whitcomb, Auguston, and Giammarco 2015).

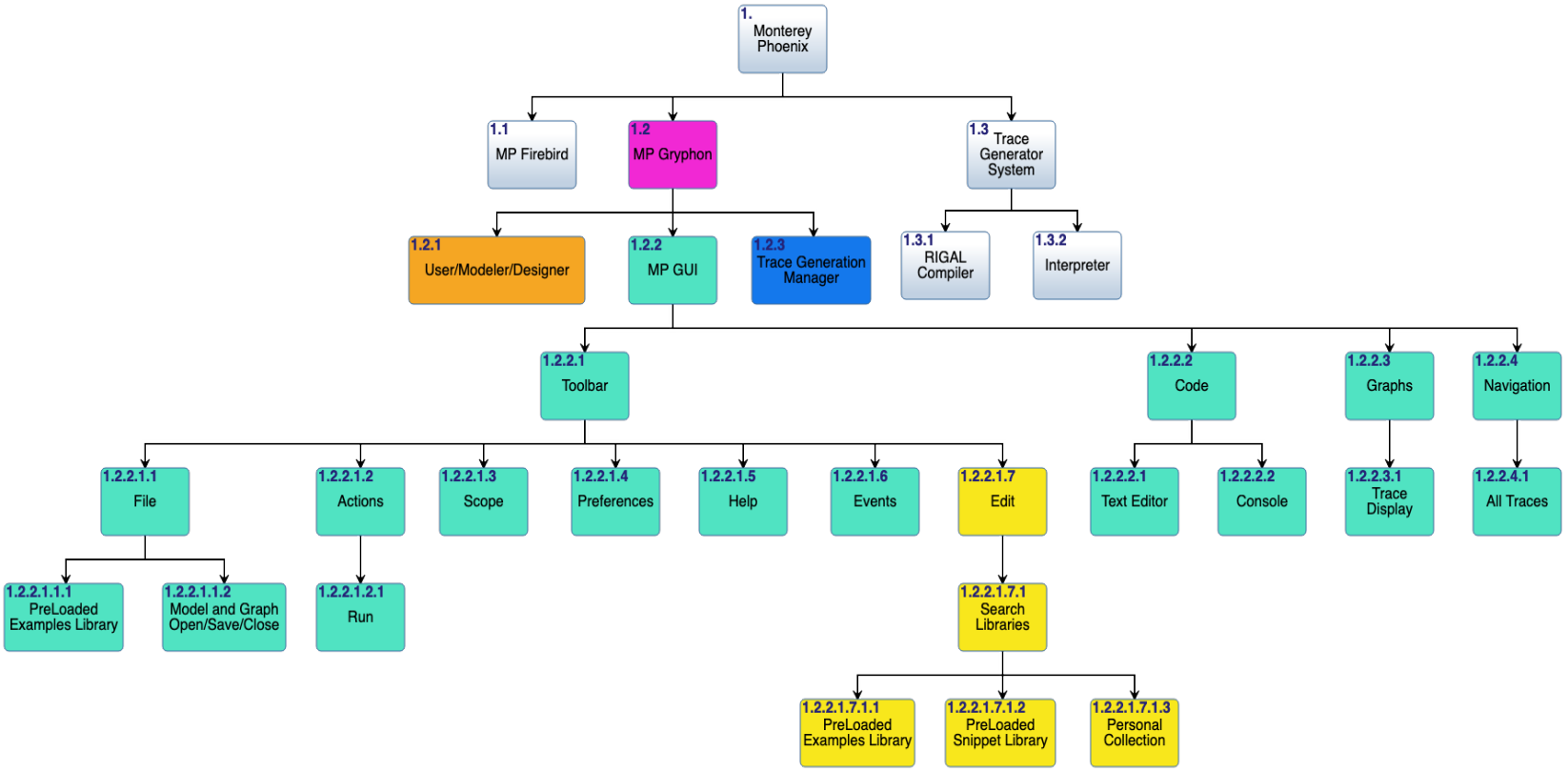


Figure 4. MP-gryphon architecture decomposition. Source: Desai (2021).

Monterey Phoenix behavior modeling describes behavior of operational processes and architecture designs ranging from complex to detailed design levels (Giammarco 2017). The MP approach enables a human-in-the-loop an improved ability for comprehension and analysis of the exhaustive generation, visualization, and querying of scoped event scenarios by separating system component behavior models from the interactions between the system and its environment. Monterey Phoenix uses a common software language and a graphical interface through a user-friendly tool that provides an ability to discern potential logic paths through behavior models. Refining the scenarios ensures behaviors have been captured correctly and early in the life cycle. Capturing scope-complete sets of alternative scenarios early exposes incorrect, unsafe, unsecure, or otherwise undesirable behaviors before they manifest in an actual system thereby saving valuable resources of time and money (Giammarco et al. 2014).

Event grammar rules produce a defined set of event traces for testing, debugging and analysis, as the core concept of software behavior modeling (Auguston, Michael, and Shing 2006). Event grammar in MP represents activity performed within the system or environment using two basic relationships: precedence to establish ordering, referred to as PRECEDES and inclusion to establish hierarchy, referenced as IN (Giammarco et al. 2014). Event traces are extracted from event grammar rules and constraints on those rules within schemas. "The schema framework is amenable to stepwise architecture refinement, reuse, composition, visualization, and application of automated tools for consistency checks" (Auguston 2009, 1). Monterey Phoenix utilizes a set of automatically generated scenarios scoped to a user-defined number of loop iterations, most often between one and three iterations. This principle teaches the concept of exposing errors using small examples to achieve the desired data scope, which is based on Jackson's Small Scope Hypothesis (Giammarco and Giles 2018).

A proven advantage of MP is the coordination between behavior hierarchies with events. Students, researchers, and practitioners are using MP to better understand complexities of systems and the ability to evaluate requirements in the distinct context environment for component behaviors and the interleaved interactions among the component constraints on events in the system The work continues to play a discovery role of emergent behaviors in MP models and enables the early discovery and refinement of design decisions before

incurring significant costs associated with design flaws. Prior research describes system behaviors,

> using operations of concurrency, alternative selection, and iteration, comparable to those used in Functional Flow Block Diagram (FFBD), Unified Modeling Language (UML), Systems Modeling Language (SysML), and Business Process Modeling Notation (BPMN) activity models. (Giammarco et al. 2014)

The MP modeling tool answers dependency behavior questions involving actions or constraints of a system. Capturing the system behaviors and environment interactions "enables the early capture and refinement of design decisions" (Giammarco et al. 2014, 209). This ability to grasp the complexity of a system helps to identify and recognize undesirable or desirable emerging behaviors or deficiencies earlier (Giammarco 2019). Assessments can be achieved by modeling inclusive interactions of subsystems to achieve detecting unexpected and emergent behaviors in SoS environment. For example, different stakeholder views can be automatically extracted and more easily visualized or communicated for better decision- making analysis and for monitoring of needs. This is a result MP's ability for providing stand-alone scenarios that create easier comprehension for all levels of stakeholders to understand.

## E. CHAPTER SUMMARY

The literature review encapsulates the relationship of the framework for selecting the Nuclear Light Water Reactor Spent Fuel Pool cooling system component scenario for the purpose of analysis and evaluation for this thesis. First, the review presents the contributions to the research in the similarities found in the recorded investigations of the Three Mile Island, Chernobyl, and Fukushima Dai'ichi nuclear power plants that found unexpected system behaviors, untrained staff, or the environment as contributing factors to the nuclear disaster. Secondly, the review presents the history and concepts leading up to the advent of DIAMOND ontology as one of the data sources of this research. Third, the review provides an overview for selecting MP behavior modeling as a proven application in the detection and prediction of emergent system of system (SoS) level behaviors.

# III. APPLICATION OF DIAMOND ONTOLOGY USING MONTEREY PHOENIX

This chapter first analyzes the INL Nuclear DIAMOND Ontology for the structural architecture of data pertaining to behavior modeling, and secondly evaluates a scenario application of a Nuclear Spent Fuel Pool (SFP) system in the context of a MP behavior model developed as part of a sponsored research project at Naval Postgraduate School (NPS).

Researchers and scientists with Idaho National Lab identify one of the main challenges impacting the nuclear community to be the lack of a standard domain ontology for shared access from the nuclear industry (Al Rashdan, Browning, and Ritter 2019). In addressing this challenge, Idaho National Lab developed the DIAMOND ontology as the first iteration of a standard to collect, store and use nuclear data and to enable consistency of storage and analysis for researchers. This analysis and evaluation of the DIAMOND ontology in the scenario application of MP behavior modeling identifies any oversights that may obstruct modeling nuclear reactor behavior using the data structure housed in DIAMOND. This analysis provides research forecasters with the benefit of continuous refinement of the DIAMOND ontology source data and provides preparation for applications of interface behavior modeling as also demonstrated in this chapter with the MP behavior model evaluation.

## A. STRUCTURE OF DIAMOND ONTOLOGY

DIAMOND is an assembly of concepts and relationships within a nuclear domain designed into an ontology structure. The DIAMOND ontology acts as the fundamental domain-specific data model enabling integrating data sources.

Idaho National Lab highlights the need for a central data warehouse for facilitating integration applications as a host and standard for the systems such as Light Water Reactor Sustainment (LWRS) Program, as well s System of Systems, resulting from the lack of a standard domain ontology for the community of interest (Al Rashdan, Browning, and Ritter 2019). Idaho National Lab further expresses that deploying a data warehouse can offer the

perspicacity to understand the integration of data sources and to better communicate the data with stakeholders (Al Rashdan, Browning, and Ritter 2019). Over the past years, the standard methodology for enabling data exchange is by point-to-point integration between individual applications. This approach is noted by INL to be both time consuming and difficult for stakeholders to comprehend. Further, this methodology is known to be restricting for integration growth, complicated by the dependency upon multiple systems, and confusing for communicating domain terminology and knowledge (Al Rashdan, Browning, and Ritter 2019). The DIAMOND ontology was created to fill this need.

### 1. Structural Layout Methodology

Idaho National Lab selected Web Ontology Language (OWL) as the preferable semantic web language to "define the relationships…data properties, and…existential, universal, or cardinal relationships to properties" (Al Rashdan, Browning, and Ritter 2019, 6). Representing DIAMOND ontology in OWL provides a structural framework of commonality for classes, relationships, and the properties connecting them for use by end users. This aggregated data model and ontology developed by INL supports the development of a data warehouse for the nuclear industry. The benefits of a defined set of domain-specific nomenclature and a domain dictionary from the DIAMOND ontology further provides use of application integration of data for scientists and researchers. DIAMOND was released as open-source software to the public, which enables developers to view and modify the OWL file. This allows developers the continuous value and straightforward expansion process for the end users. DIAMOND software was prepared in 2019 by Battelle Energy Alliance, LLC under contract with the United States (U. S.) Department of Energy (DOE).

To structure the ontology of DIAMOND, INL developers selected foundational ontologies consisting of elements pulled from Basic Formal Ontology (BFO) and Life cycle Modeling Language (LML) to with proven domain-specific classes for better end user buy-in. This inclusion of BFO and LML provides the advantage of validation and integration potential (Al Rashdan, Browning, and Ritter 2019). Use of Basic Formal Ontology (BFO) provides an upper-level ontology for retrieving information for analysis,

and integration for communities of interest. To further assist in promoting clarity and understanding within the nuclear domain, some BFO classes are relabeled in common nuclear terminology while other properties or relationships associated with BFO classes remain unchanged to maintain the BFO structure within DIAMOND and enable integration with other BFO structured ontologies (Al Rashdan, Browning, and Ritter 2019). Capturing data using LML architecture provides a modeling language for combining logical constructs with ontologies. For interoperability, the OWL syntax can consist of a variety of forms, such as functional syntax, OWL2 XML, and Manchester. According to World Wide Web Consortium (W3C), the most common form is RDF/XML syntax and is the syntax used in development of DIAMOND.

> The DIAMOND ontology developers and future end users avoid potential circularity concerns, traceability failure of node connections, and challenges in personnel working simultaneously by following several essential principles already proven successful in development of other ontologies. (Al Rashdan, Browning, and Ritter 2019, 6)

These key series of principles identify the benefit of utilizing terms already common to other domain experts, avoiding acronyms, and using singular common terms. Other crucial principles adding value to the development of DIAMOND recognize the need for the ontology to possess an ability to adapt to changes and uncertainty (Al Rashdan, Browning, and Ritter 2019).

### 2. Types of Elements

Three key elements identified by developers for the structural ontology layout of DIAMOND include classes, object properties, and data properties. Relationships are represented by object properties and attributes use data properties. In addition, content of the class tables use annotations for creating the data properties. Protégé is the open-source tool selected by INL to incorporate these elements into an OWL file for use in DIAMOND. The tool organizes the ontology in hierarchical views for an ease of editing of OWL files as depicted in Figure 5. The Protégé tool was developed by Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine to acquire, represent,

and process biomedical data for decision making (Guarino, Nicola, and Musen 2015). The Protégé tool offers a way to search for synonyms or related terms that can be reused.

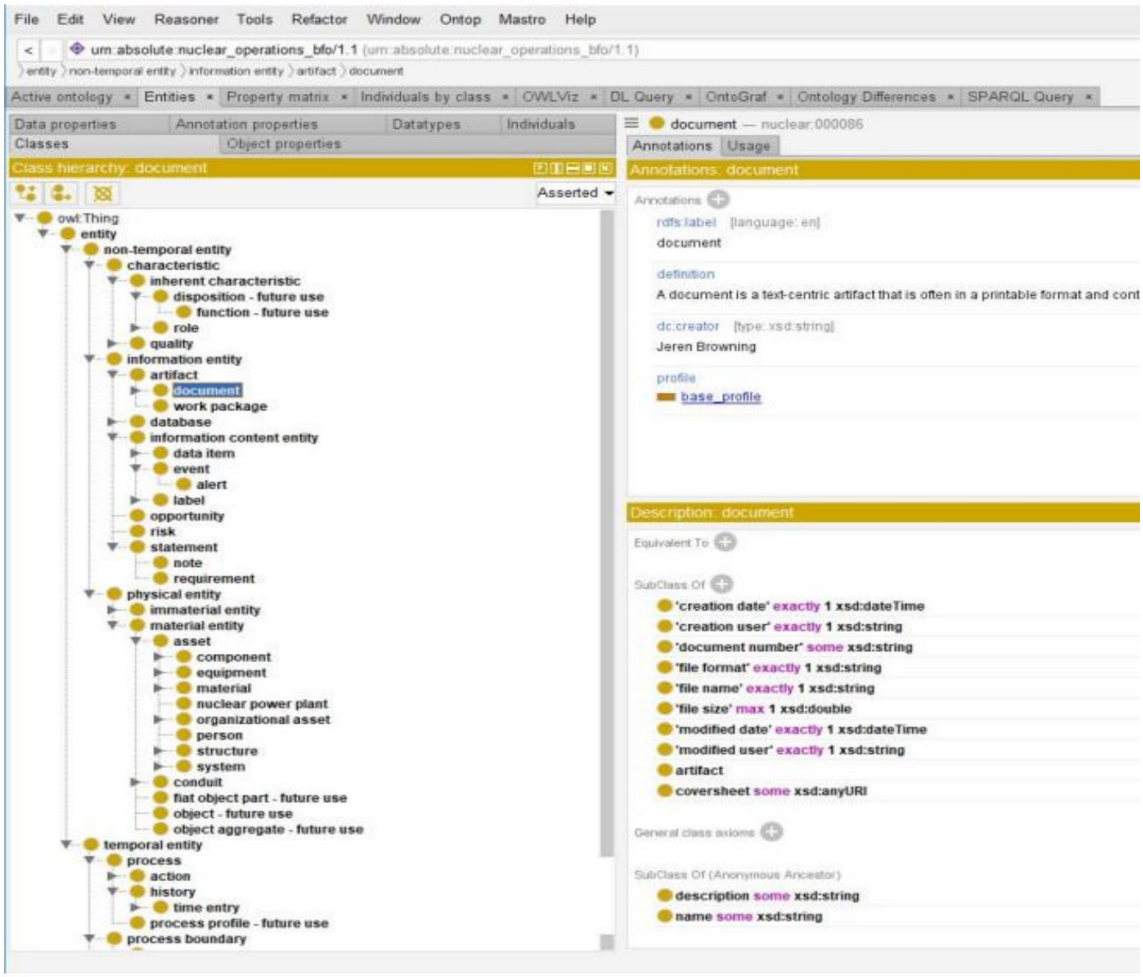


Figure 5. A view of DIAMOND within the Protégé editor. Source: Al Rashdan, Browning, and Ritter (2019).

"BFO creates classes that can categorize any entity. This includes processes, material things, immaterial things, and … concepts like qualities and roles." (Al Rashdan, Browning, and Ritter 2019, 11). The Classes element is the key object primarily constructed for the ontology. Classes define multiple nuclear domain entities as a hierarchy view and indicate inheritance relationships of those classes. These class entities define six groups: the domain processes, material things, immaterial things, quantity, roles, and

unknowns. The group entity of a class is then further defined by defining additional specific continent or occurrent properties and labeled accordingly. his structure style mirrors the LML class structure in other industry-domain ontologies.

In DIAMOND, classes are labeled in PascalCase case with the most unique common name possible and includes alternative names, annotations, and acronyms added as "alternative term" annotations. The unique name aids to avoid confusion and ensures compatibility for related applications. Sample class grammar used in DIAMOND ontology is listed below (Al Rashdan, Browning, and Ritter 2019). DIAMOND ontology structure does not currently contain a designated class for Input/Output, rather it uses a trigger class serving the role of inputs or outputs.

- Action – Generates effects and may have pre-conditions before it can be executed. This Action can include transforming inputs into outputs.
- Artifact – Specifies a document or other source of information that is referenced by or generated in the knowledge base.
- Asset – Specifies an object, person, or organization that performs an action, such as a system, subsystem, component, or element.
- Conduit – Specifies the means for physically transporting Input/ Output entities between Asset entities. It has limitations (attributes) of capability and latency.
- Opportunity – A circumstance with the potential for some benefit.
- Requirement – Identifies a capability, characteristic, or quality factor of a system that must exist for the system to have value and utility to the user.
- Risk – Specifies the combined probability and consequence in achieving objectives.
- Statement – Specifies text referenced by the knowledgebase and usually contained in an Artifact. (Al Rashdan, Browning, and Ritter 2019, 13)

Labels from a parent class, such as role, are often seen in the child's classes, such as operations role, to fully define the entity without user assuming any portion of the desired class name. "The properties of a class (also) inherit parent classes that are grouped into 'SubClass Of (Anonymous Ancestor)' group within the description of a class. Properties that are specific to class entity (along with an automatically generated entry indicating the parent class) are placed in the 'SubClass Of' group" (Al Rashdan, Browning,

and Ritter 2019, 7). Sibling classes are marked as disjoint relationships in higher levels in the ontology to "ensure DIAMOND follows a strict specialization hierarchy where each class has only one parent" (Al Rashdan, Browning, and Ritter 2019, 7. Multiple inheritance is discouraged to promote clear definitions of classes.

Secondly, object properties (relationships) are labeled in camelCase and describes other user defined relationships applied to the classes. DIAMOND ontology primarily bases these relationships as LML relationships with additionally created relationships including "has role" and "includes" relationships as well as the inverse "included by" relationship. For example, a student would be listed as "SubClass Of" of person, rather as the role. Most of these relationships have a known inverse. The relationships without a known inverse are specified by labeling to quality, generic association or inverse relationship, and the class role (Al Rashdan, Browning, and Ritter 2019). The reciprocal relationships defined in DIAMOND are shown in Figure 6.

```
fact {performedby = ~ performs}
fact {generatedby = ~ generates}
fact {receivedby = ~ receives}
fact {connectedby = ~ connects}
fact {transferredby = ~ transfers}
fact {specifiedby = ~ specifies}
fact {tracedfrom = ~ tracedto}
fact {sourcedby = ~ sourceof}
```

Figure 6. Reciprocal relationships designed in DIAMOND. Source: Al Rashdan, Browning, and Ritter (2019).

A restriction type must be specified when developing the object properties to link to a class. DIAMOND developers describe restriction types as Min, Max, and Exactly, and require a number to also be provided. Additional restriction types as, Some and Only are existential restriction and indicate that a defined relationship is unknown, but that one or more may exist. "Important concepts are considered, such as whether a child can have more than one parent, whether an attempt is being made to describe reality or concept, etc." (Al Rashdan, Browning, and Ritter 2019, 4). "This restriction type is the default restriction type for the DIAMOND ontology because it allows for optional properties on a class" (Al

Rashdan, Browning, and Ritter 2019, 9). When the restriction type is known as required for the class to exist, then the Min, Max, or Exactly restriction types apply. Development guidelines for DIAMOND used the 'Only' restriction type for indicating restriction types to a predefined set of classes or data properties. "Object property characteristics also denote whether an object property relationship is functional, transitive, or symmetric" (Al Rashdan, Browning, and Ritter 2019, 9). The 'Some' type is a restriction used for indicating when one or more of that relationship exists. These one or more relationships are the default restriction types for the DIAMOND ontology (Al Rashdan, Browning, and Ritter 2019). The current DIAMOND ontology does not make a distinction for optional multiplicity of zero or more relationships and assumes there is always at least one existing relationship.

The third element, labeled in lower case is data properties (attributes) which can be envisioned as fields that organize the content of defined classes like those stored in database tables. Data properties are grouped by type and are often a string or date timestamp. "These types of attributes from the XML Schema Definition Language (XSD) are named accordingly within the Protégé tool" (Al Rashdan, Browning, and Ritter 2019, 9). The data property can be applied multiple classes. Before adding new data, properties developers are advised to ensure the attribution does not already exist in other ontologies.

Making DIAMOND available to the public to enable developers to view and modify the OWL file is a key milestone for DIAMOND. DIAMOND's open access requirement and extendable nature allows a continuous and straightforward expansion process (Al Rashdan, Browning, and Ritter 2019). For DIAMOND's ontology to be fully compatible and user-friendly with other ontologies, high-level BFO classes with noninstinctive names were assigned more intuitive alternative labels in the DIAMOND ontology. "BFO classes included in DIAMOND not currently in use by the ontology are labeled "future use" to indicate that the class is not in use but may be used in the future" (Al Rashdan, Browning, and Ritter 2019, 12). A list of the updates to BFO class labels with the BFO original label first, followed by DIAMOND's new label:

- Continuant: Nontemporal Entity

- Occurrent: Temporal Entity
- Generically Dependent Continuant: Information Entity
- Independent Continuant: Physical Entity
- Specifically Dependent Continuant: Characteristic
- Realizable Entity: Inherent Characteristic (Al Rashdan, Browning, and Ritter 2019, 12)

The BFO class "Specifically Dependent Continuant," is referred to as "Characteristic" in DIAMOND ontology (Arp et al. 2015). Process is intuitive and may occur during a period of time or as an input or output from the actor performing the process. It should exist at any time the associated physical entity "Independent Continuant" exists. "This means that attributes of some physical entity that can be measured over time or scientifically evaluated should be included in the ontology as qualities. This includes entities such as mass, voltage, pressure, etc." (Al Rashdan, Browning, and Ritter 2019, 11). Nontemporal entity is a general parent entity that includes physical items or descriptions of the items representing qualities and roles. "Temporal class entities that only exist along sometimeline. Examples might include actions or the history of some entity" (Al Rashdan, Browning, and Ritter 2019, 13).

Cited benefits of the ontology include "ease of application integration, a defined set of domain-specific nomenclature, and a domain dictionary … extended to include further levels … or other domains" (Al Rashdan, Browning, and Ritter 2019, 4).

## B. CROSSWALK OF DIAMOND AND MONTEREY PHOENIX

A crosswalk between DIAMOND and MP reveals four behavior-related axes of comparison: ontology scope, objects and relations, use of multiplicity, and control flow. The following subsections analyze the elements of comparison along each of these axes. This crosswalk supports mapping of the corresponding relevant system data housed in the DIAMOND ontology and the selected scoped scenario modeled as an MP schema in later sections of this chapter.

### 1. Ontology Scope Crosswalk

The first point of comparison pertains to ontology scope. The DIAMOND ontology defines classes, properties, and relationships that frame an expandable nuclear specific

domain of system data. The DIAMOND ontology houses "a set of concepts and categories in a subject area or domain that shows their properties and the relations between them" (Al Rashdan, Browning, and Ritter 2019. 4). As noted in Figure 7, the DIAMOND database has grown to a warehouse of 562 defined Classes, 21 Object Properties, 410 Data Properties, 46 Annotation Properties, and 13 Datatypes of NPP database data.

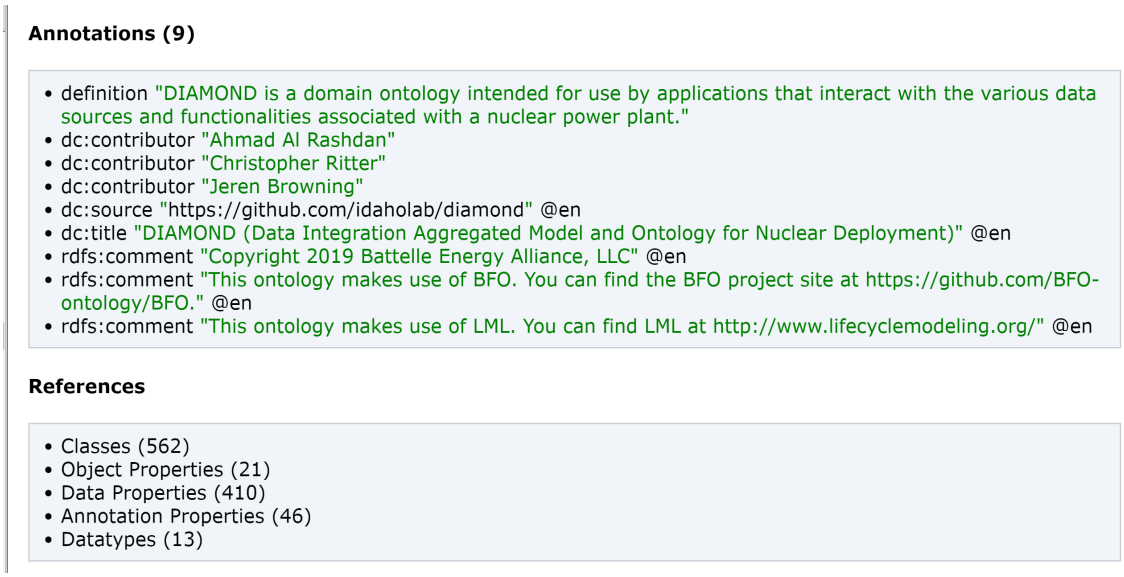


Figure 7. Screenshot of domain references in the DIAMOND ontology. Source: DIAMOND (2019).

In contrast, MP's ontology scope is limited to classes that pertain to behavior. An MP schema is a formal behavior specification that is defined as a "collection of rules describing behavior of components within the system, external actors, and their interactions" (Auguston 2020). "The MP behavior model is based on the concept of *event* as an abstraction of activity" (Auguston 2020, 6). MP uses a simple event grammar language for framing and describing behaviors and interactions to generate and query across sets of extracted instances of behavior, known as event traces.

### 2. Object and Relation Crosswalk

The second point of comparison noted was of precedence, inclusion, and user-defined relations in DIAMOND and in MP. The DIAMOND ontology is constructed from

*classes* that represent a much wider range of objects and relationships between them. Whereas the MP event grammar is domain-neutral (able to be used to represent behavior of systems in any domain), the DIAMOND ontology is specific to the nuclear domain. An example of DIAMOND ontology classes is shown in Figure 8 (left). DIAMOND object classes are typically singular nouns and uniquely named.

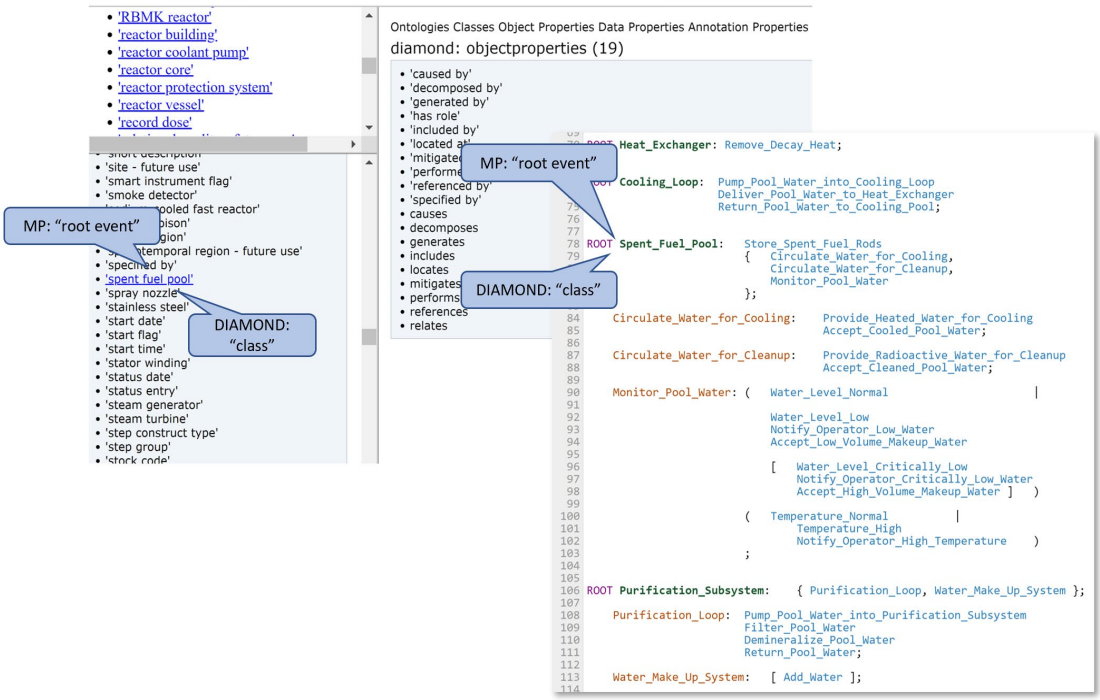


Figure 8. DIAMOND ontology screen shot of class mapped to MP root event and MP behavior model code mapped to DIAMOND ontology. Adapted from DIAMOND (2019) and MP-Firebird (2022).

In Monterey Phoenix, events are the fundamental building blocks of the behavior modeling. Top-level events in MP schemas are designed as root events (Figure 8, right). Root events are used to represent behaviors of separate systems, parts of systems, or processes. Root events include sub- events belonging to that system, part, or process. A composite event is both included in another event and includes other event(s) (Auguston 2020). Composite events are used to bundle related events into one event and are included in a root event or another composite event. The lowest-level events in schemas are

designated as atomic events. Atomic events are included in root or composite events and may be later refined into composite events, as needed (Figure 9, right).

Event grammar rules establish formal specifications using MP language constructs for dependencies among events and are used for describing event traces. The structure of event types is specified bygrammar rules in terms of IN and PRECEDES relations (Figure 9, left). “The event grammar models the behavior as a set of events (event trace) with two basic relations, where the PRECEDES relation captures the ordering dependency relationship, and the IN relation represents the hierarchical relationship” (Auguston 2020, 5).

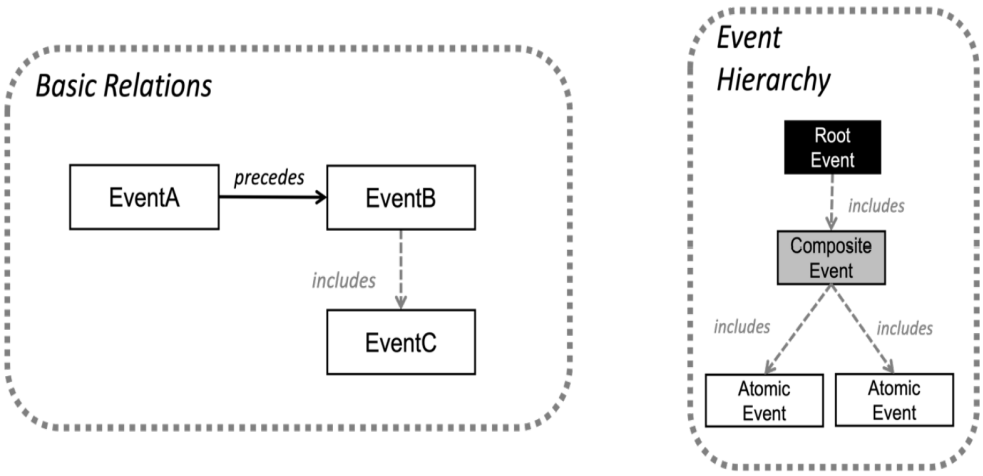


Figure 9. An event has two basic relations: precedence and inclusion.
Source: Giammarco (In Press, Chap 3).

An event can have a time duration for the action to be accomplished and other event attributes that are number or interval types. MP events may be singular or plural nouns, verbs, or verb phrases, and can have the same names as other events (leaving the formal specification of whether they are the same or different instance of event as a separate constraint). Figure 10 illustrates a schema and example event trace flow layered with event types.

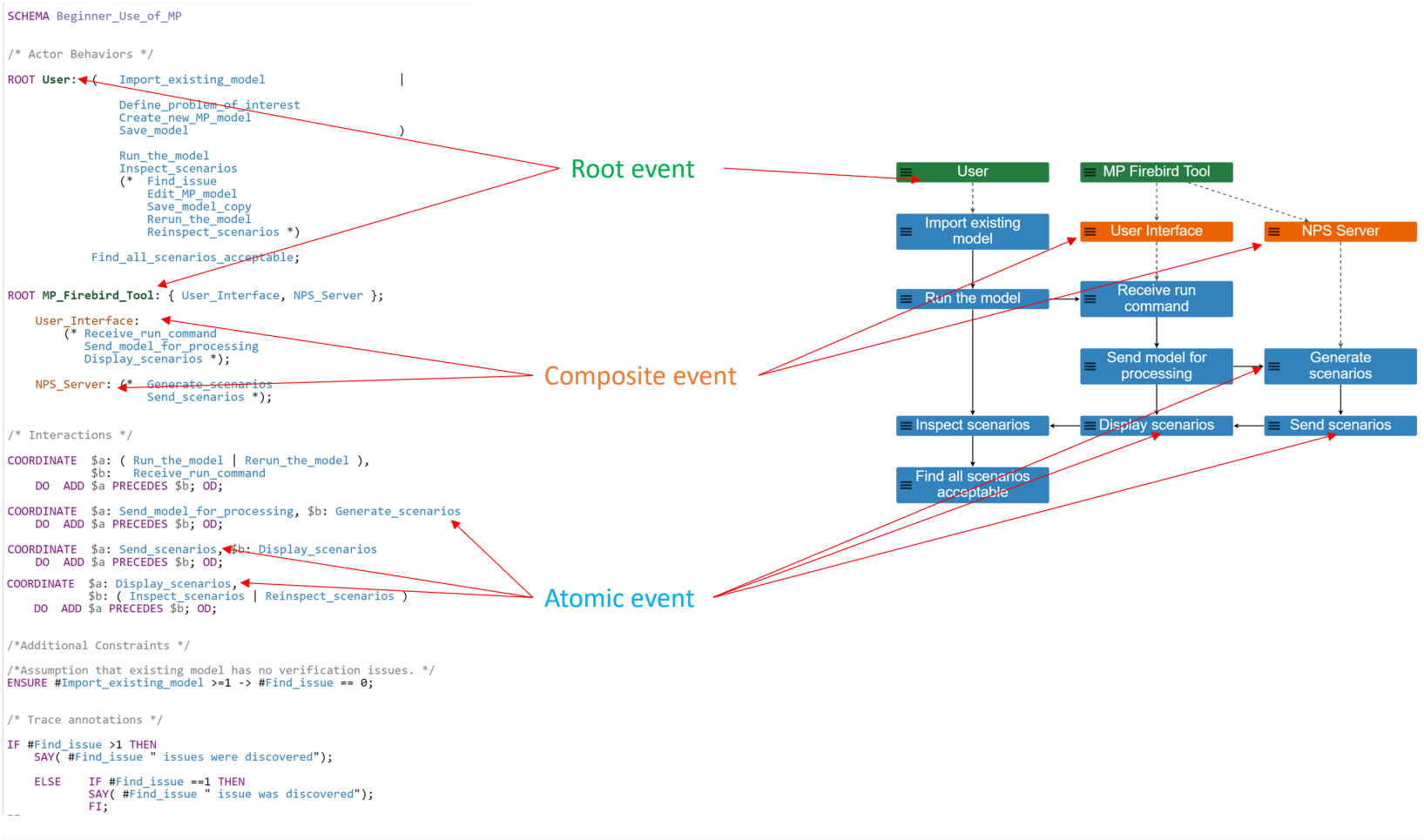


Figure 10. Example MP schema for root, composite, and some atomic events mapped to event trace flow. Source: MP-Firebird (2022).

The MP language uses formal event grammar to enable users to construct sentences for modeling the behavior. As shown in Figure 11, event grammar rules provide formal specifications for dependencies among events using MP language constructs.

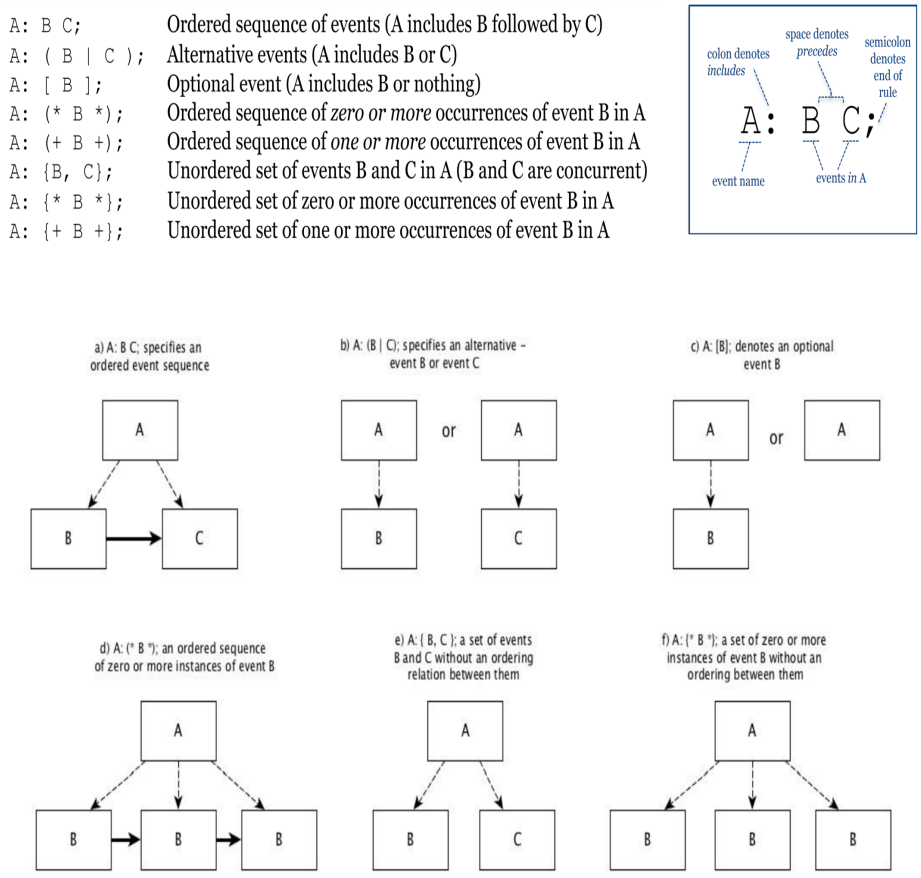


Figure 11. Example grammar rules with corresponding event traces. Source: Auguston (2020).

Event is an abstract class in MP that can be used to represent an object, activity, condition, state, occurrence, outcome (Giammarco In Press, Chapter 3), and certain concepts that have been specialized as different classes in the DIAMOND ontology. For example, the class "spent fuel pool" from the DIAMOND ontology may be represented in an MP schema as a root event named "Spent_Fuel_Pool" (Figure 8). Because there is no

formal separation of information types into classes in MP, MP events (as abstract representations) may employ naming conventions to signal the class it represents. Events have three types: root, composite and atomic. The top-level event having no parent event is named the root event. The mid-level event having at least one parent and one child is named the composite event and is normally used to bundle several related events into one event within another composite or root event. Finally, the lowest-level event is named atomic event and represents behavior at the lowest level. The three types of events are illustrated in the event trace in Figure 12 representing consumers and suppliers interacting in a general supply chain. The MP event trace illustrated in Figure 12 represents root events (actors) with green boxes, composite events with orange boxes, and atomic events with blue boxes. Relationships shows dashed arrows representing inclusion (decomposition) relations, solid black lines representing precedence relations, and solid blue lines representing user-defined relations.

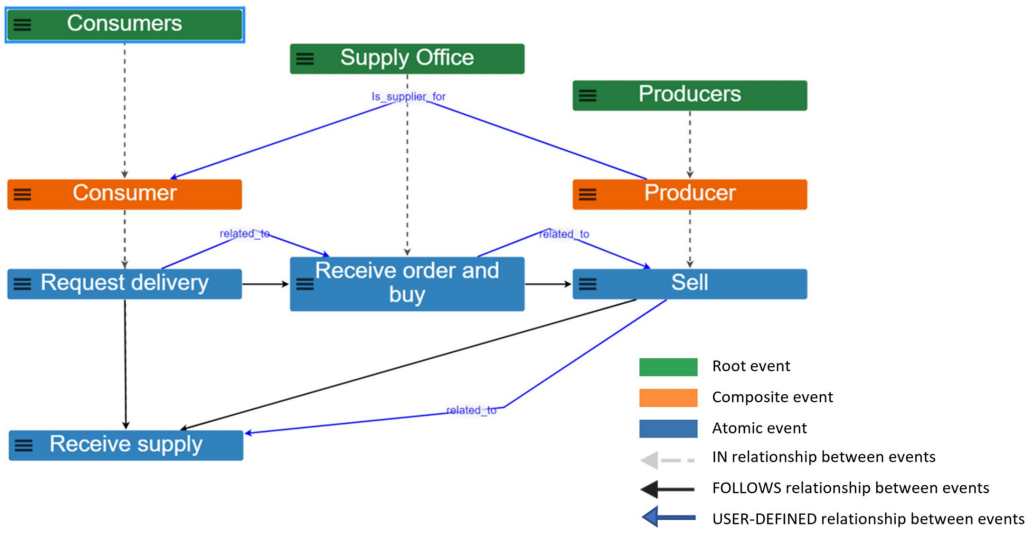


Figure 12. Event trace example showing root, composite, and atomic events in a general supply chain model. Adapted from MP-Firebird (2022).

MP events expressed in the form of noun-oriented names are typically root or composite events representing physical objects, and MP events expressed using verb-oriented names denote activities performed by the physical objects (usually composite or atomic events but may also be root events) (Auguston 2020). Relations like "performed by" may be encoded as user-defined relations in MP while basic relations of precedence and inclusion are available by default on MP event traces.

In DIAMOND, "*Properties* of a class inherited from parent classes are grouped into the 'SubClass Of (Anonymous Ancestor)' group within the description of a class along with properties that are specific to that class" (Al Rashdan, Browning, and Ritter 2019, 7). Figure 13 illustrates object properties relationship in the DIAMOND ontology. DIAMOND classes use "planned-action class" for actions unique in a process. The "actual-action class" is used to represent a unique instance of "that action within a process. This includes attributes such as actual start date, actual finish date, actual duration, etc." (Al Rashdan, Browning, and Ritter 2019, 14). The DIAMOND action class is analogous to the MP event, which can represent an action with a beginning and end time and/or a duration (Auguston 2020).

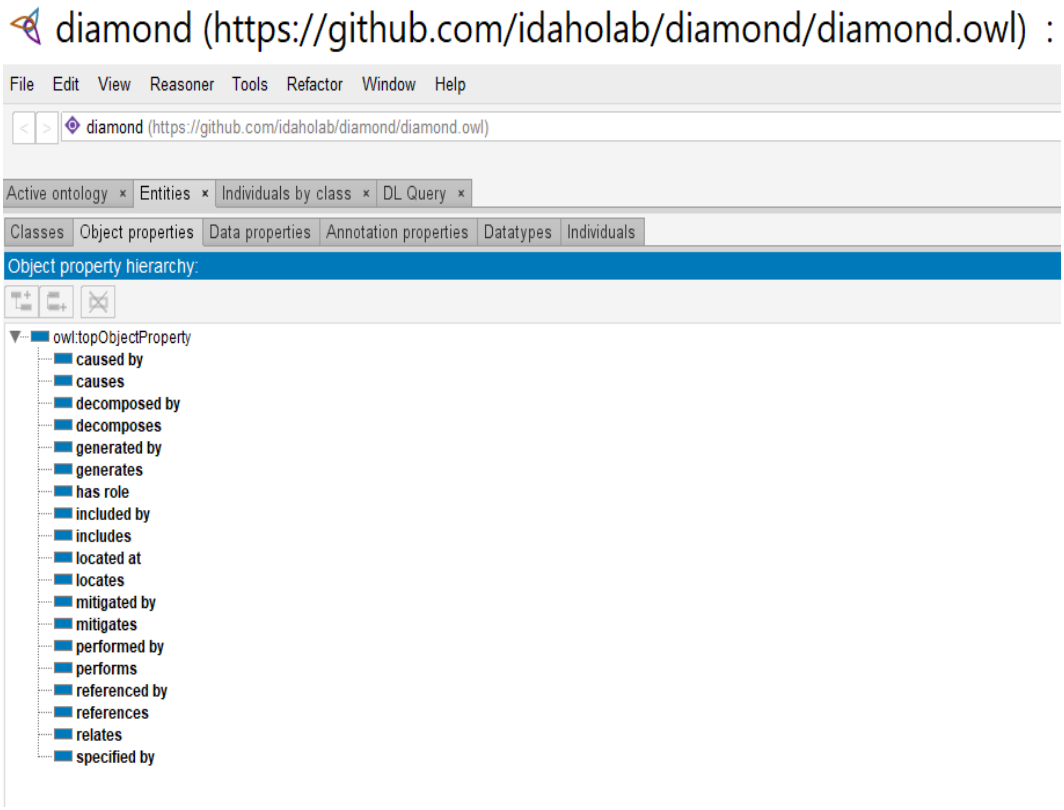


Figure 13. Illustration of object property entities in DIAMOND. Source: DIAMOND (2019).

The defined *relationships* within DIAMOND include LML relationships with eight other additions (shown in Figure 13) such as "decomposes" with inverse "decomposed by." Relationship data is used to connect information together in models. Monterey Phoenix uses basic relations of precedence as a causal or temporal dependency for establishing cases in which an event precedes another and inclusion as a hierarchical dependency for events that are nested within other events.

In MP, behaviors in different roots are coordinated using COORDINATE or SHARE ALL statements. COORDINATE using PRECEDES specifies an ordering relation between events in different roots whereas SHARE ALL causes events in different roots with the same name to be merged (Auguston 2020). In either case, separate grammar rules are set up for each system or component to describe each independent algorithm of

behavior, and dependencies are established by separate constraint statements among events in the different grammar rules.

COORDINATE constraints allow one to add dependencies such as precedence, inclusion, user-defined, and other types of relations for reasoning about behavior (Auguston 2020). Figure 14 illustrates example use of the COORDINATE statement and its impact on an event trace.

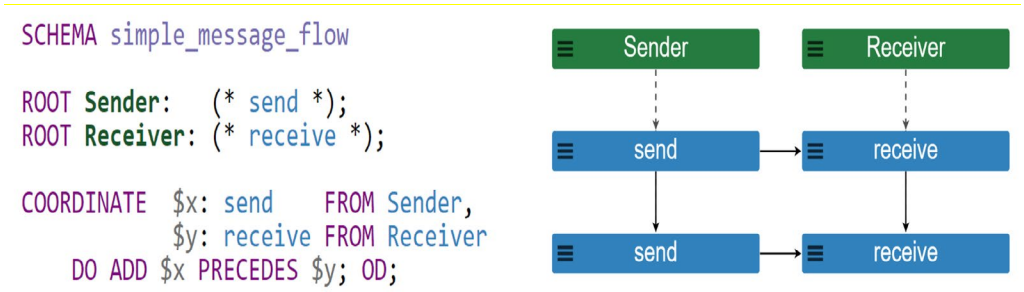


Figure 14. Left: example schema and right: event trace (scope 2 trace 3) illustrates the adding of a precedence relation between events from different roots using a COORDINATE statement. Source: Example 2 on MP-Firebird (2022).

The $x and $y variables temporarily store send and receive events respectively as the trace generator runs. The DO loop pairs off existing x's and y's in the order they occur, adding a precedence relation between them. The COORDINATE statements act as constraints on the emergent behaviors of the model. Removing COORDINATE constraints allows discovery of every potential path of execution to be considered, whether considered as physically or logically invalid or exposing unexpected or undesired behaviors in the actual system (Giammarco and Auguston 2019).

Monterey Phoenix permits multiple inheritance from root events to composite or atomic events as shared inclusion relations. Thus, two or more parents are allowable in MP hierarchy. Shared events of the "same type have the same event type name and hence have the same event attribute" (Giammarco and Auguston 2019) Figures 15 and 16 illustrate examples from MP-Firebird (2022) of event sharing. The event traces in these figures show

the dashed lines indicating multiple "parent" events to demonstrate the function of the SHARE ALL statement. Writer and Reader are components, and the File is the data item shared between the components. Both figures illustrate use of schemas produce the same traces, with one using a COORDINATE with SHARE statement, and the other using the SHARE ALL composition that is an abbreviation of the former.

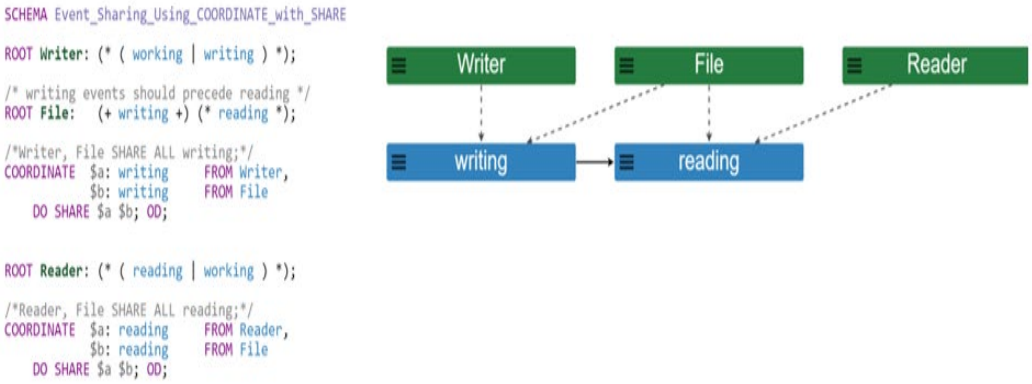


Figure 15. Schema and event trace to illustrate event sharing via COORDINATE statement with SHARE. Source: MP-Firebird (2022).

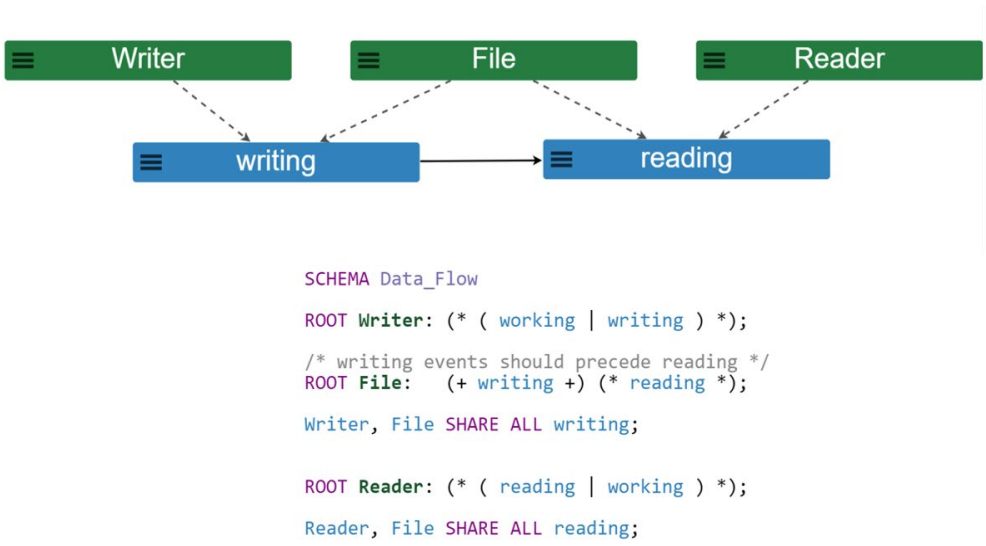


Figure 16. Schema and event trace to illustrate event sharing via SHARE ALL statement, which abbreviates the COORDINATE statement with SHARE. Source: Example 3 on MP-Firebird (2022).

The DIAMOND ontology allows disjointed subclasses to help ensure following a hierarchy of each class having only one parent. It is noted that multiple inheritance is not prohibited in DIAMOND but is discouraged "to promote clear definitions of classes and a structure that is easy to interpret" (Al Rashdan, Browning, and Ritter 2019, 7).

Among the fifteen data categories that DIAMOND developers deemed pertinent to daily monitoring of nuclear operations are the process controls (inputs) and instruments (outputs) class. The parent class of instrument has multiple children classes of controls, such as instrument panels.

> The relationship from panel or any given panel may target one or more of these classes. Creating the relationship to the parent-class, instrument, ensures that the scope of the relationship is kept to the instrument and its subclasses, but prevents the need for a relationship to each potential type of instrument (Al Rashdan, Browning, and Ritter 2019, 19).

### 3. Multiplicity Crosswalk

The current DIAMOND ontology has the means to apply multiplicity to certain relationships. For example, the restriction types, Min, Max, and Exactly apply when cardinality is known. The default Some type is an existential restriction and indicates one or more may exist and allows for optional class properties. The restriction type of Only is used for restricting property indicating enumerators to a predefined set of classes using data properties (Al Rashdan, Browning, and Ritter 2019). DIAMOND does not make a distinction for optional multiplicity of zero or more relationships and assumes there is always at least one existing relationship.

There are at least two ways to represent multiplicity in MP behavior models: one-to-one and one-to-many COORDINATE and SHARE ALL statements, and zero-or-more or one-or-more iteration. The center COORDINATE statement in Figure 17 is an example of a one-to-one precedence relationship being made between a pair of events in different roots. One-to-many and many-to-one precedence relations may also be made using COORDINATE statements. The top and bottom COORDINATE statements in Figure 17 illustrate how to use nested coordination to establish one-to-many and many-to-one precedence relationships between pairs of events in different roots. The bottom statement

accomplishes a single message to many UAVs, and the top statement accomplishes messages from different UAVs to a single receiver.

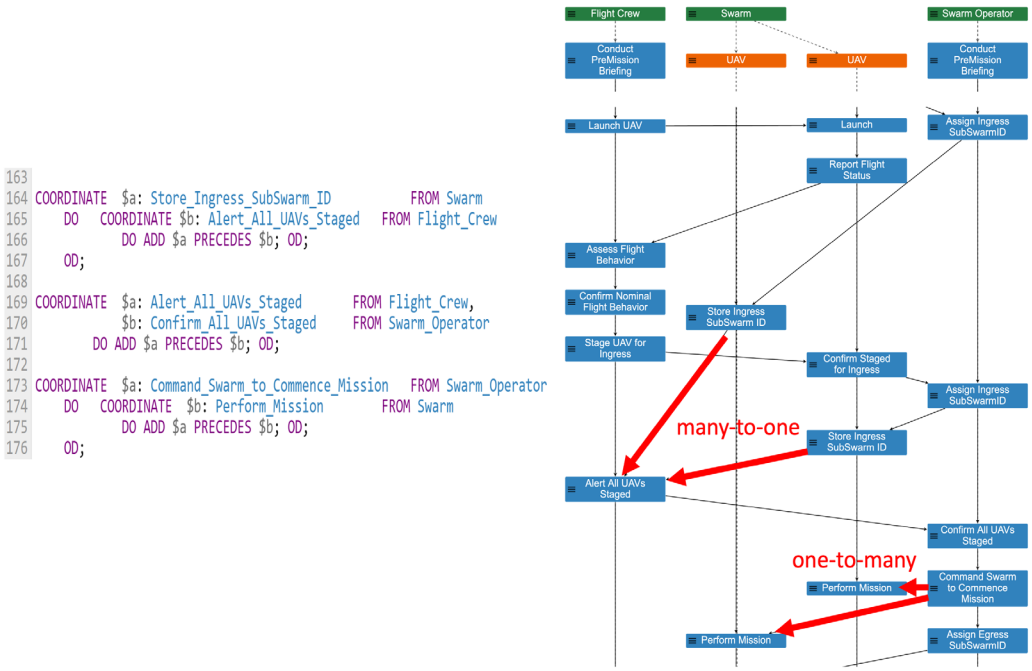


Figure 17. Nested COORDINATE statements establish many-to-one and one-to-many relationships. Source: MP-Firebird (2022).

The SHARE ALL statement in Figure 18 creates an inclusion relation between two root events and a single atomic event (2 to 1).

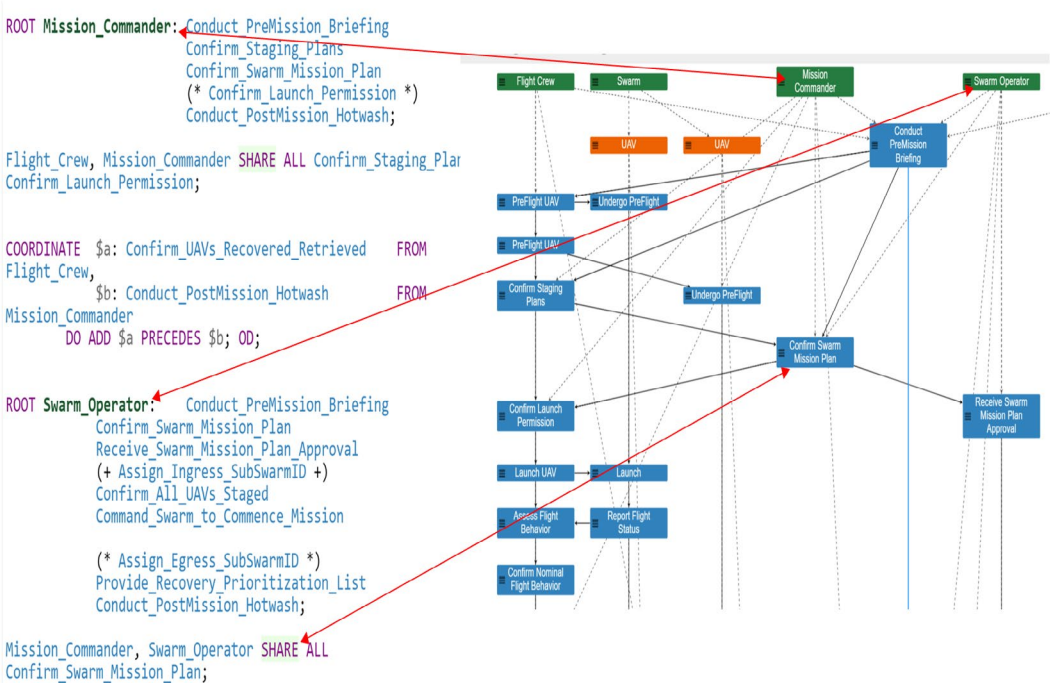


Figure 18. The SHARE ALL statement establishes an inclusion relation between two root events. Adapted from MP-Firebird (2022).

Multiplicity may also be represented as event iteration in MP. As discussed earlier, MP event grammar rules may contain zero-or-more ordered iteration (*…*), one-or-more ordered iteration (+…+), zero-or-more unordered iteration {*…*} and one-or-more unordered iteration {+…+} of enclosed events. The global scope controls the "or-more" upper limit on this multiplicity and local scope settings can be made to set custom lower and upper boundaries for iteration. An example of an MP schema of event grammar rules for zero or more unordered iteration {+…+} is shown and labeled in Figure 19. The enclosed event is UAV, so a global scope setting of 2 will instantiate two UAVs, a setting of 3 will instantiate three UAVs, etc.

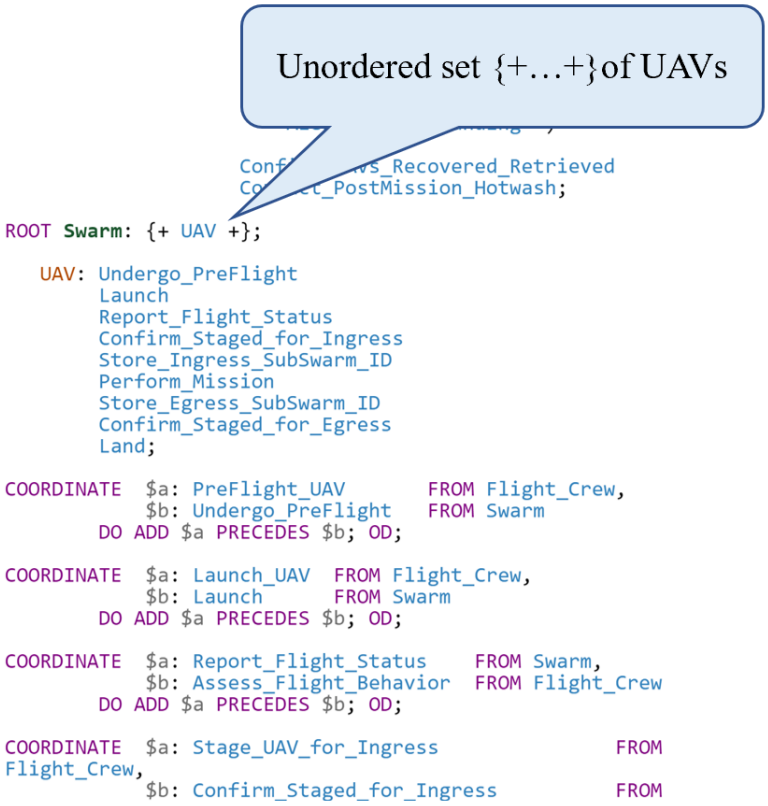


Figure 19. Schema of event grammar rules for one or more {+…+} UAV events. Adapted from MP-Firebird (2022).

## 4. Control Flow Crosswalk

The DIAMOND class designed as Process Instruments and Control incorporates the input (controls) and output (instruments) of processes within a nuclear plant. “Trigger” is used as a threshold for input (controls) and for output (instruments). For example, as a threshold input is met, an ouput instrument alarm is triggered (Al Rashdan, Browning, and Ritter 2019). Figure 20 captures a screen shot example for showing the concept of “trigger” in INPUT/OUTPUT in the context of DIAMOND.

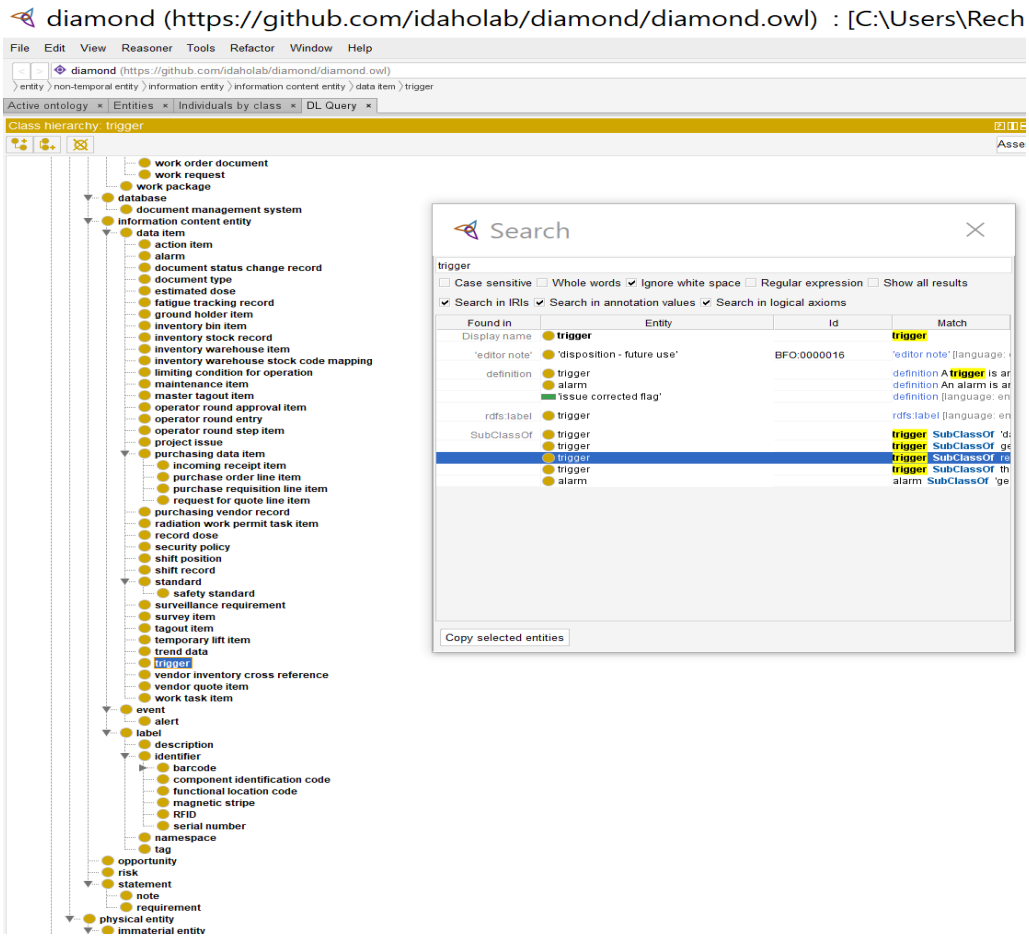


Figure 20. Screenshot for accounting for "trigger" in DIAMOND. Source: DIAMOND (2019).

Monterey Phoenix does not have a fundamental class for information elements like Input/Output or a separate property of "trigger" on input/outputs, rather all concepts are modeled as events, and data is modeled in terms of operations on the data. For MP, event is the foundational building block for behavior with event representing an action, a condition, a state, an occurrence, or an outcome. Use of user-defined relations create associations that clearly connect chains of related events for capturing energy, matter, and information transfers with the precedence relation acting as the "trigger" which is the precedence relation established using the COORDINATE statement. Figure 21 illustrates a simple event trace with user-defined relations capturing the energy, matter and information flows.

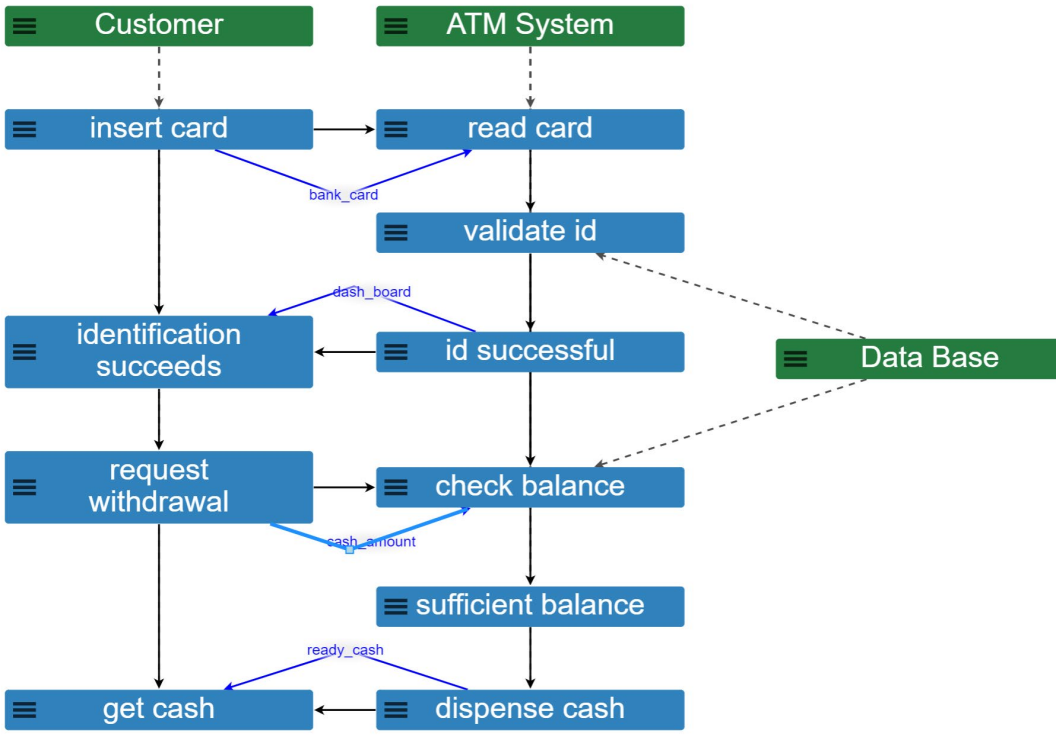


Figure 21. Event trace flow of user-defined relations used to indicate input/ outputs in MP

## C. SPENT FUEL POOL (SFP) BACKGROUND

The Light Water Nuclear Spent Fuel Pool cooling system was selected as the scoped scenario for this research. As recognized by the literature review for this research, the historical nuclear accidents were traced to unexpected or unwanted system behaviors related to a combination of factors of technology, people, or the environment. The PPD-21 (2013) states that the light water nuclear spent fuel pool systems are part of the sixteen identified national infrastructure of safety critical systems. Images in Figures 22 and 23 capture views of INL's storage pool and fuel canisters in storage cooling.

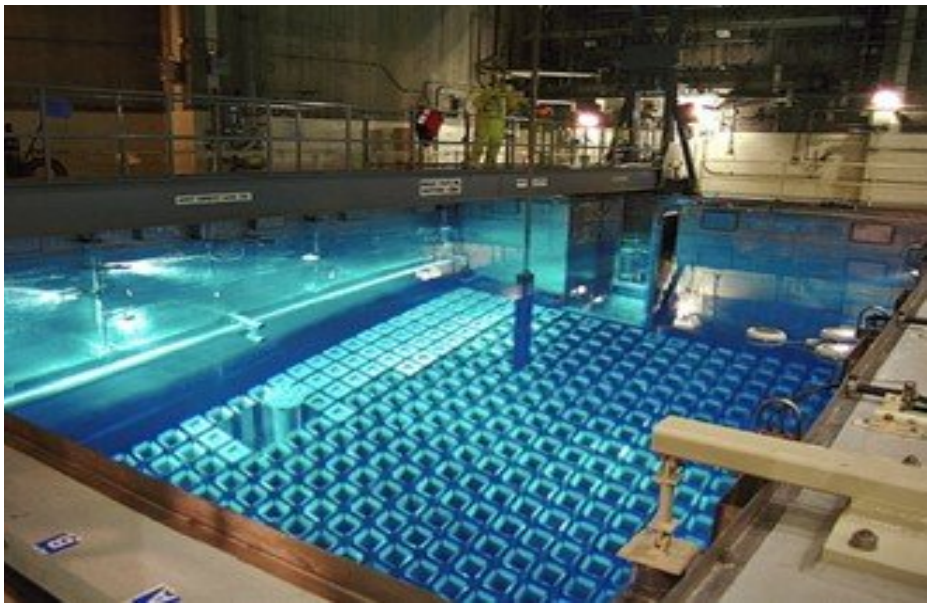

Figure 22. Spent fuel basin at Idaho National Laboratory. Source: INL.gov (2022).

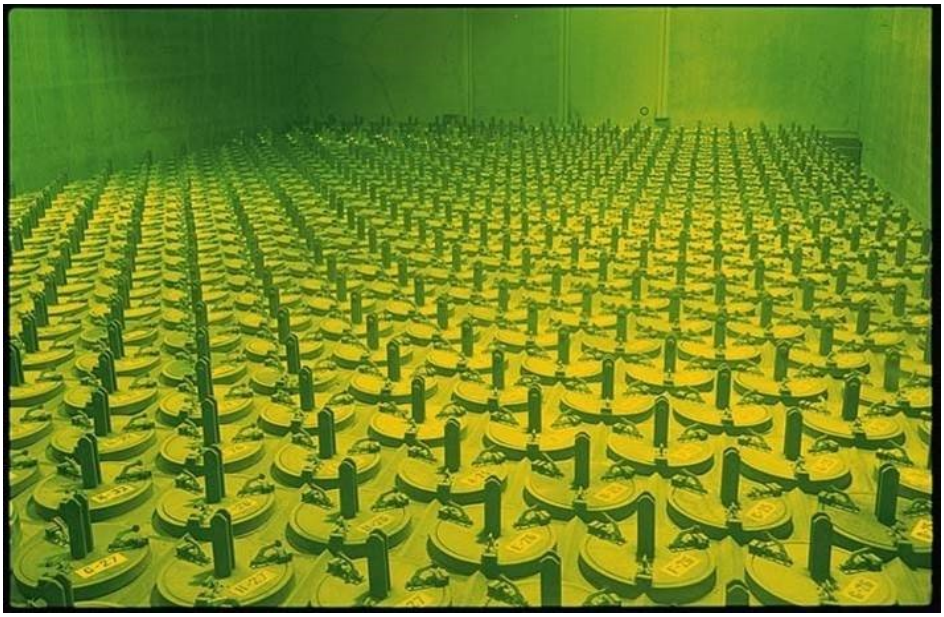

Figure 23. Irradiated fuel storage facility at Idaho National Lab. Source: INL.gov (2022).

Every 18 to 24 months, workers discharge spent fuel from nuclear reactor cores into storage pools known as spent fuel pool (SFP) assemblies for cooling. Spent fuel pools

are vulnerable to extensive damage triggered by inadequate cooling flow leading to overheating and release of reactivity of the fuel assemblies in the SFP.

> Spent fuel assemblies contain large amounts of unstable radionuclides, byproducts from the atomic fissions that powered the reactor. The radioactive decay of these radionuclides generates heat. Spent fuel assemblies also contain large amounts of uranium and plutonium atoms, fissionable material that could reignite a nuclear chain reaction. (Union of Concerned Scientists 2016)

The spent fuel pool cooling system and cleaning system consists of three subsystems: the cooling, purification, and skimmer subsystems (USNRC HRTS 14.4 Rev 0109) As illustrated in Figure 24, the SFP cooling and cleaning system removes decay heat from the stored fuel and maintains the coolant water level. The system flow path of the components is depicted in Figure 25. The system filters and maintains the coolant purity and clarity.

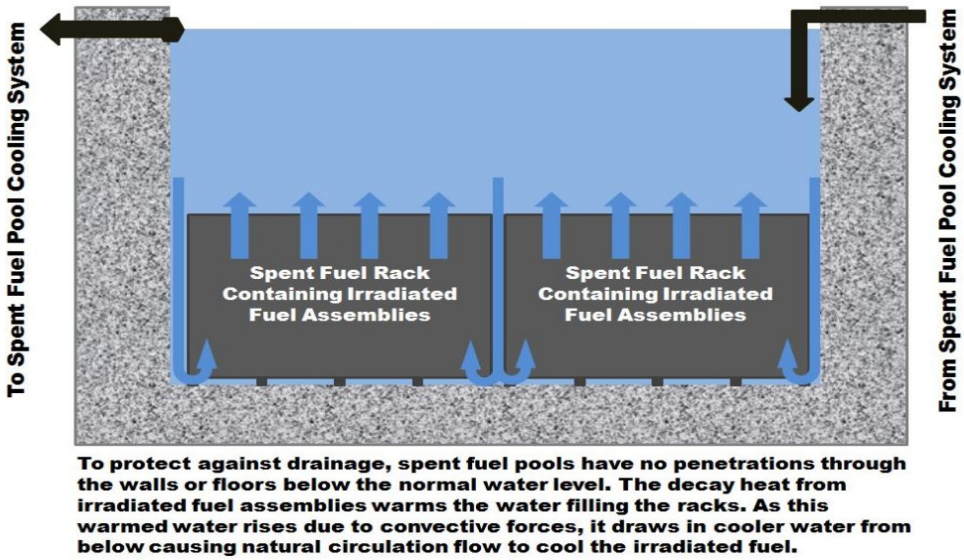


Figure 24. Depiction of light water reactor spent fuel pool purpose Source: UCS (2016).

Used reactor fuel assemblies contain hot and radioactive spent fuel. Fuel assemblies containing spent fuel are stored under water to provide a shield from radiation and to also provide cooling (Ali 2013, Chap 48).

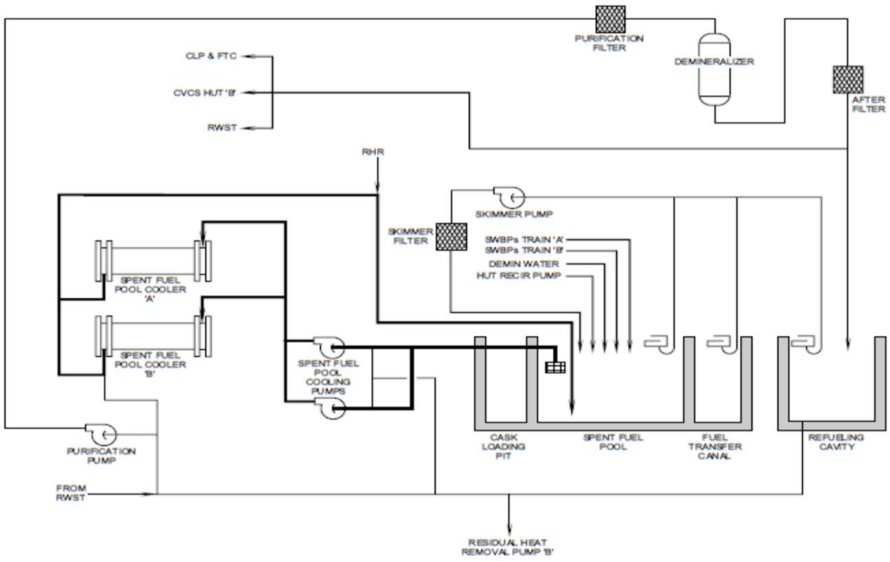


Figure 25. Basic flow path of spent fuel pool cooling and cleaning system. Source: USFRC HRTD (Rev 0109).

The spent fuel pool cooling and cleanup systems is designed to function to maintain the spent fuel pool water temperature at less than or equal to 140 degrees Fahrenheit by removing the decay heat (USNRC HRTS 14.4 Rev 0109). This is accomplished with the assistance of the residual heat removal system by the heat exchangers component using cooling water. Emergency diesel generators power other pumps to supply extra water for the pool as needed. Other functions of the system include maintaining the purity of the borated water by supplying sufficient and emergency cooling in the event a fuel assembly or foreign object is dropped into the spent fuel pool (USNRC HRTS Rev 0109). If cooling pool water flow is not sufficiently maintained, workers only have hours to reestablish mandatory cooling to the spent fuel pool assemblies to prevent an accident. Therefore, continuously flowing water from the water pumps to supply water to cool the spent fuel is essential.

## D. DEMONSTRATION OF SPENT FUEL POOL MP BEHAVIOR MODEL

System behavior models assist analysists to answer questions and make decisions. “Monterey Phoenix helps its users remove ambiguity, generate consistent and scope complete scenario sets, and expose unexpected behaviors latent in a design” (Giammarco

In Press, Chapter 3). Monterey Phoenix uniquely "models behaviors and interactions separately, then combines them" (Giammarco In Press, Chapter 3) to generate scenarios supporting inspection for emergent behavior.

The evaluation of DIAMOND, within the context of the behavior model, demonstrates an application to refine and interrelate underlying relationships in the behavior model through a scoped scenario (use case) to verify DIAMOND standard vocabulary. Nuclear Reactor SFP event traces of the MP behavior scenarios and the analysis with the ontology were documented and reviewed by an INL expert panel to ensure that the scenarios are representative of operationally realistic behavior, and the DIAMOND data was implemented correctly in the model.

Figure 26 demonstrates identifying the components of and interactions among a spent nuclear fuel cooling pool and its environment from a source schematic (USNRC HRTS 14.4 Rev 0109). Root events for spent fuel pool cleaning and cooling system are labeled and linked to system components shown in the schematic.

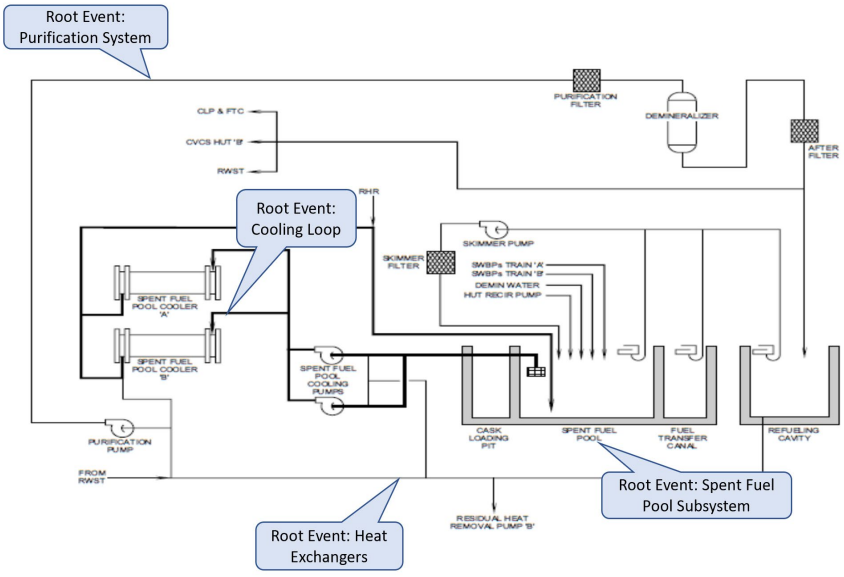


Figure 26. Identification of MP root events from elements of the spent fuel pool cooling and cleaning system are shown with MP schema labeled mapped to system components diagram. Source: USNRC HRTS 14.4 (Rev 0109).

The MP behavior model of the Spent Fuel Pool Cooling and Cleanup System was initially created by Kristin Giammarco and edited by Douglas Van Bossuyt in May 2020 both of NPS with input from nuclear subject matter experts.

The Figure 27 illustration captures a visual from the user viewpoint when accessing the behavior model to use MP-Firebird. The MP-Firebird displays the main menu across the top, the typed or loaded code on the left, and the graph results on the right. Simple controls in the main menu include The Code, Split and Graph buttons in the upper left of the main menu to enable controlling the display with the additional focus options of the code, both the code and graphs, or just the graphs. The Import menu allows loading the existing Spent Fuel Pool model from the library in the directory. The About menu contains links to the Monterey Phoenix website and the MP user manual. The *code window* on the left displays the imported MP behavior model SFP code. The *scope of execution* is identified by the slider bar for selecting the number of iterations desired. When the *run button* is pressed, the MP code is sent to the event trace generator on the Firebird server for processing. The *console window* provides statistics about the trace derivation process, such as the average, min, and max number of events per trace, and elapsed time. When trace generation is complete, the console window displays the *number of traces* generated and *trace window* display graphical view. The navigation pane on the far right displays thumbnail views of the generated *event traces* under the total number of traces generated for each run.

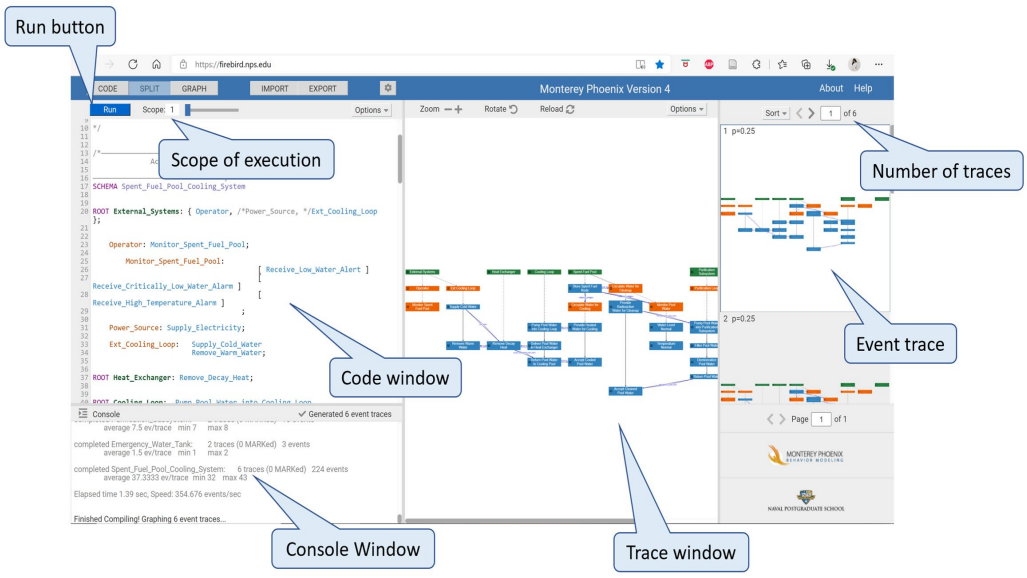


Figure 27. Illustration and labeling of key features of MP behavior model in MP-Firebird. Adapted from SFP.mp (2022).

To build the SFP behavior model, the behavior of the SFP was coded using knowledge of the subject matter, and a scope-complete set of event traces was generated to define the selected SFP use case variants. Scope-complete indicates inclusion of the user-specified maximum number of iterations of events enclosed in (*…*), (+…+), {*…*}, and {+…+} notation. First, the schema was defined for the selected behavior of the system with the external systems in the environment as indicated in Figure 28. In MP, "schema" for coding the behavior model is a collection of rules describing behavior of components or systems and their interactions.

```
13 /*_____________________________
14                Actors
15
16 _____________________________*/
17 SCHEMA Spent_Fuel_Pool_Cooling_System
18
19
20 ROOT External_Systems: { Operator, /*Power_Source, */Ext_Cooling_Loop };
21
22
23     Operator: Monitor_Spent_Fuel_Pool;
24
25         Monitor_Spent_Fuel_Pool:
26                                         [ Receive_Low_Water_Alert ]
27                                         [ Receive_Critically_Low_Water_Alarm ]
28                                         [ Receive_High_Temperature_Alarm ]
29                                     ;
30
31     Power_Source: Supply_Electricity;
32
33     Ext_Cooling_Loop:   Supply_Cold_Water
34                         Remove_Warm_Water;
```

Figure 28. SFP model schema name declaration (Line 17). Source: SPF.mp (2022).

Secondly, components and their activities were defined following Auguston's concept of abstract events. Recall that events are the fundamental building blocks of behavior for representing the activity, action, and process of scoped scenarios (Auguston 2009). Assigning root events for the MP schema develops from selecting system components determined to have system behavior. The "Spent Fuel" pool cooling subsystem root event represents a subsystem behavior where one pump suctions from the spent fuel pool through the strainer and a second pump discharges back to the spent fuel pool cooling through heat exchangers. The root event "Purification Subsystem" consists of a branch line connected to the pumps, filter, demineralizer, and an after-filter. A root event "cooling loop" return line located above the fuel assembly storage racks discharges directly from connections to the residual heat removal system located between spent fuel pool and the cooling pump and return piping between the pool and spent fuel pool exchangers. Demineralized makeup water flowing from the water makeup system is designed to compensate for loss of coolant., When needed, there is emergency water source from the service water system. During accident conditions, the root event "Refueling Water Storage Tank" tank provides a source of water the emergency core cooling systems pumps. The borated water provides cooling of the core and replacement inventory. The root events selected to build event traces are External Systems, Heat Exchanger, Cooling Loop, Spent

Fuel Pool, Purification System, and Emergency Water Tank as applicable behavior activity in the context of its environment.

The MP model of architecture-level behavior for the spent fuel pool cooling and cleanup system is here described and illustrated, along with example event traces extracted from the model. MP SFP schema of root events, composite events, and atomic events shown in Figure 29. The purpose of this model is to identify the components of and interactions among a spent nuclear fuel cooling pool and its environment. This model is used in this research to demonstrate the mapping of concepts in MP to concepts in DIAMOND.

```
ROOT Heat_Exchanger: Remove_Decay_Heat;

ROOT Cooling_Loop:  Pump_Pool_Water_into_Cooling_Loop
                    Deliver_Pool_Water_to_Heat_Exchanger
                    Return_Pool_Water_to_Cooling_Pool;


ROOT Spent_Fuel_Pool:   Store_Spent_Fuel_Rods
                        {   Circulate_Water_for_Cooling,
                            Circulate_Water_for_Cleanup,
                            Monitor_Pool_Water
                        };

    Circulate_Water_for_Cooling:    Provide_Heated_Water_for_Cooling
                                    Accept_Cooled_Pool_Water;

    Circulate_Water_for_Cleanup:    Provide_Radioactive_Water_for_Cleanup
                                    Accept_Cleaned_Pool_Water;

    Monitor_Pool_Water: (   Water_Level_Normal                  |

                            Water_Level_Low
                            Notify_Operator_Low_Water
                            Accept_Low_Volume_Makeup_Water

                            [   Water_Level_Critically_Low
                                Notify_Operator_Critically_Low_Water
                                Accept_High_Volume_Makeup_Water ]   )

                        (   Temperature_Normal          |
                                Temperature_High
                                Notify_Operator_High_Temperature    )
                        ;


ROOT Purification_Subsystem:    { Purification_Loop, Water_Make_Up_System };

    Purification_Loop:  Pump_Pool_Water_into_Purification_Subsystem
                        Filter_Pool_Water
                        Demineralize_Pool_Water
                        Return_Pool_Water;

    Water_Make_Up_System:   [ Add_Water ];


ROOT Emergency_Water_Tank:  [ Add_Water ];
```

Figure 29. Spent_Fuel_Pool_Cooling_System roots, composites, and atomic events. Source: SPF.mp (2022).

Events across roots are coordinated using COORDINATE and SHARE ALL statements. COORDINATE and SHARE ALL statements provide constraints for

impacting the statements on the model. Separate grammar rules for each system or component describe independent behavior for each. Separate constraint statements add dependencies among events in the different grammar rules. COORDINATE constraints, for example, allow one to add dependencies such as precedence, inclusion, user-defined, and other types of relations for reasoning about behavior (Auguston 2020). Constraints on the Spent Fuel Pool and other components are shown in Figure 30.

```
/*_____________________________
                Interactions
_____________________________*/

/*COORDINATE    $a: Supply_Electricity  FROM External_Systems,
            $b: Provide_Radioactive_Water_for_Cleanup   FROM Spent_Fuel_Pool,
            $d: Provide_Heated_Water_for_Cooling    FROM Spent_Fuel_Pool,
            $e: Intake_Pool_Water                   FROM Purification_Subsystem
    DO  ADD $a electricity $b;
        ADD $a electricity $d;
        ADD $a electricity $e;
    OD;*/
COORDINATE  $a: Store_Spent_Fuel_Rods           FROM Spent_Fuel_Pool,
            $b: Circulate_Water_for_Cooling     FROM Spent_Fuel_Pool,
            $c: Circulate_Water_for_Cleanup     FROM Spent_Fuel_Pool,
            $d: Monitor_Pool_Water              FROM Spent_Fuel_Pool
    DO  ADD $a radioactive_particles $b;
        ADD $a thermal_energy $b;
        ADD $a radioactive_particles $c;
        ADD $a thermal_energy $c;
        ADD $a radioactive_particles $d;
        ADD $a thermal_energy $d;
OD;
COORDINATE  $a: Provide_Heated_Water_for_Cooling        FROM Spent_Fuel_Pool,
            $b: Pump_Pool_Water_into_Cooling_Loop       FROM Cooling_Loop
    DO  ADD $a PRECEDES $b;
        ADD $a hot_radioactive_pool_water $b;
    OD;
COORDINATE  $a: Deliver_Pool_Water_to_Heat_Exchanger    FROM Cooling_Loop,
            $b: Remove_Decay_Heat       FROM Heat_Exchanger
    DO  ADD $a PRECEDES $b;
        ADD $a hot_radioactive_pool_water $b;
    OD;
COORDINATE  $a: Supply_Cold_Water   FROM External_Systems,
            $b: Remove_Decay_Heat   FROM Heat_Exchanger,
            $c: Remove_Warm_Water   FROM External_Systems
    DO  ADD $a PRECEDES $b;
        ADD $b PRECEDES $c;
        ADD $a cold_water $b;
        ADD $b hot_water $c;
OD;
COORDINATE  $a: Provide_Radioactive_Water_for_Cleanup   FROM Spent_Fuel_Pool,
            $b: Pump_Pool_Water_into_Purification_Subsystem     FROM Purification_Subsystem
    DO  ADD $a PRECEDES $b;
        ADD $a hot_radioactive_pool_water $b;
OD;

COORDINATE  $a: Remove_Decay_Heat   FROM Heat_Exchanger,
            $b: Return_Pool_Water_to_Cooling_Pool   FROM Cooling_Loop
    DO  ADD $a PRECEDES $b;
        ADD $a cooled_radioactive_pool_water $b;
OD;
COORDINATE  $a: Return_Pool_Water_to_Cooling_Pool   FROM Cooling_Loop,
            $b: Accept_Cooled_Pool_Water            FROM Spent_Fuel_Pool
    DO  ADD $a PRECEDES $b;
        ADD $a cooled_radioactive_pool_water $b;
    OD;
COORDINATE  $a: Return_Pool_Water          FROM Purification_Subsystem,
            $b: Accept_Cleaned_Pool_Water  FROM Spent_Fuel_Pool
    DO  ADD $a PRECEDES $b;
        ADD $a hot_cleaned_pool_water $b;
    OD;

```

Figure 30. Constraints on the spent fuel pool and other components. Source: SPF.mp (2022).

Precedence relations for modeling the water level behavior between two different root events are established using a COORDINATE statement, as displayed in Figure 31. The $x and $y variables temporarily store send and receive events respectively as the trace generator runs. The DO loop pairs off existing x's and y's in the order that they occur, adding a precedence relation between them. Constraints for cases in which water levels of the nuclear spent fuel pool system are low are shown in Figure 31.

```
/* If Water Level Low */

COORDINATE  $a: Water_Level_Low            FROM Spent_Fuel_Pool,
            $b: Add_Water                  FROM Purification_Subsystem
    DO  ADD $a PRECEDES $b;
        ADD $a set_point_1_trigger $b;
    OD;

COORDINATE  $a: Notify_Operator_Low_Water   FROM Spent_Fuel_Pool,
            $b: Receive_Low_Water_Alert            FROM External_Systems
    DO  ADD $a PRECEDES $b;
        ADD $a set_point_1_alert $b;
    OD;

COORDINATE  $a: Water_Level_Critically_Low      FROM Spent_Fuel_Pool,
            $b: Add_Water                       FROM Emergency_Water_Tank
    DO  ADD $a PRECEDES $b;
        ADD $a set_point_2_trigger $b;
    OD;

COORDINATE  $a: Notify_Operator_Critically_Low_Water    FROM Spent_Fuel_Pool,
            $b: Receive_Critically_Low_Water_Alarm          FROM External_Systems
    DO  ADD $a PRECEDES $b;
        ADD $a set_point_2_alarm $b;
    OD;

COORDINATE  $a: Add_Water                           FROM Purification_Subsystem,
            $b: Accept_Low_Volume_Makeup_Water      FROM Spent_Fuel_Pool
    DO  ADD $a PRECEDES $b;
        ADD $a clean_cool_water $b;
    OD;

COORDINATE  $a: Add_Water                           FROM Emergency_Water_Tank,
            $b: Accept_High_Volume_Makeup_Water     FROM Spent_Fuel_Pool
    DO  ADD $a PRECEDES $b;
        ADD $a clean_cool_water $b;
    OD;

COORDINATE  $a: Notify_Operator_High_Temperature    FROM Spent_Fuel_Pool,
            $b: Receive_High_Temperature_Alarm          FROM External_Systems
    DO  ADD $a PRECEDES $b;
        ADD $a high_temp_alarm $b;
    OD;

```

Figure 31. Additional constraints on the spent fuel pool and other components in the case of low water level. Source: SPF.mp (2022).

The combined MP code from Figures 28, 29, 30, and 31are in Appendix. After the MP model is composed, the model is run by sending it to the server, which compiles the MP code, generates the scenarios, and finally displays the generated scenarios back to the MP modeler. This process is repeated as many times as needed until the model runs as expected and there are no unwanted behaviors in any of the generated scenarios (use case variants). A simple swim lanes diagram generated from MP shows example behavior and interactions among components in the Spent Fuel Pool Cooling and Cleaning System behavior model (Figure 32). This model demonstrates example interactions among components of a spent nuclear fuel cooling pool and its environment.

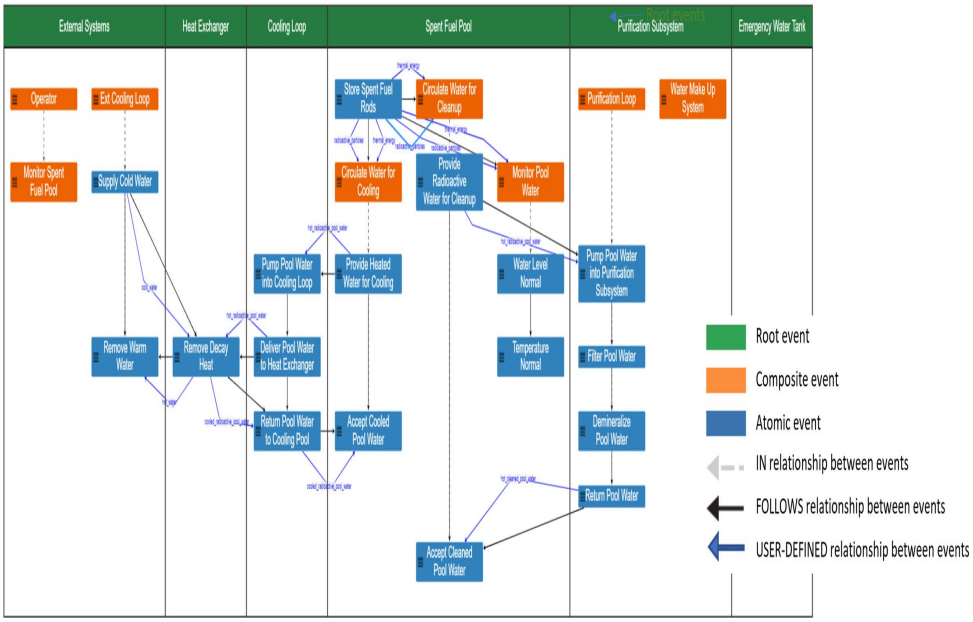


Figure 32. Example nominal behavior illustrating the modeling of a complex nuclear spent fuel pool cooling and purification system MP event trace. Green boxes are root events (actors), orange boxes are composite events, and blue boxes are atomic events. Dashed arrows are inclusion (decomposition) relations and solid arrows are precedence relations. Source: SPF.mp (2020).

The following set of figures illustrate the six event traces generated from the MP model of the nuclear spent fuel pool cooling and purification system. Event traces are

typically read top to bottom and left to right. The trace in Figure 33 shows that the spent fuel pool cooling system has an external operator (orange box) and external cooling loop (orange box) as external systems (green box) where the operator monitors the spent fuel pool. When the water level is normal and temperature high, the system notifies the operator that temperature is high by activation of high temperature alarm (blue boxes) triggering supply of cooling water. The external cooling loop (green box) then supplies cold water or removes warm water (orange boxes). The heat exchanger (green box) removes decay heat (orange box) and the cooling loop pumps pool water into cooling loop, delivers pool water to heat exchanger, and returning pool water to cooling pool (blue boxes).

The spent fuel pool (green box) stores spent fuel rods, circulates water for cooling by providing heated water for cooling and accepting cooled pool water, circulates water for cleanup by providing radioactive water for cleanup and accepts cleaned pool water, monitors pool water by monitoring for water levels.

Figures 34–38 show other possible behavior variants. When water level is detected as being low water level and with temperature as normal, the system notifies the operator that the water levels are low by activation of low temperature alert (blue boxes) triggering supply of cooling water to accept low volume make-up water (Figure 34). When water level is normal there is no need for extra action and monitoring continues. When water level is detected as being critically low, then the external operator is notified by triggered alert of the low water and also notified by high temperature alarm triggered by high temperature. The spent fuel pool countermeasure is to begin accepting high volume make-up water (Figure 35). When water level is normal, and the temperature is normal then there is no need for any extra action and the system will continue in normal monitoring action mode (Figure 36). However, when the water level is normal, but the temperature is detected as high, the system notifies the external operator from the triggered alarm of the high temperature risk (Figure 37). The purification subsystem consists of purification loop and water make-up system. The purification loop pumps pool water into purification subsystem, filters pool water, demineralize pool water, and returns pool water when triggered by system of water levels being critically low (Figure 38). The water make-up system adds water from the emergency water tank until normal water level and normal temperature returns and is maintained.

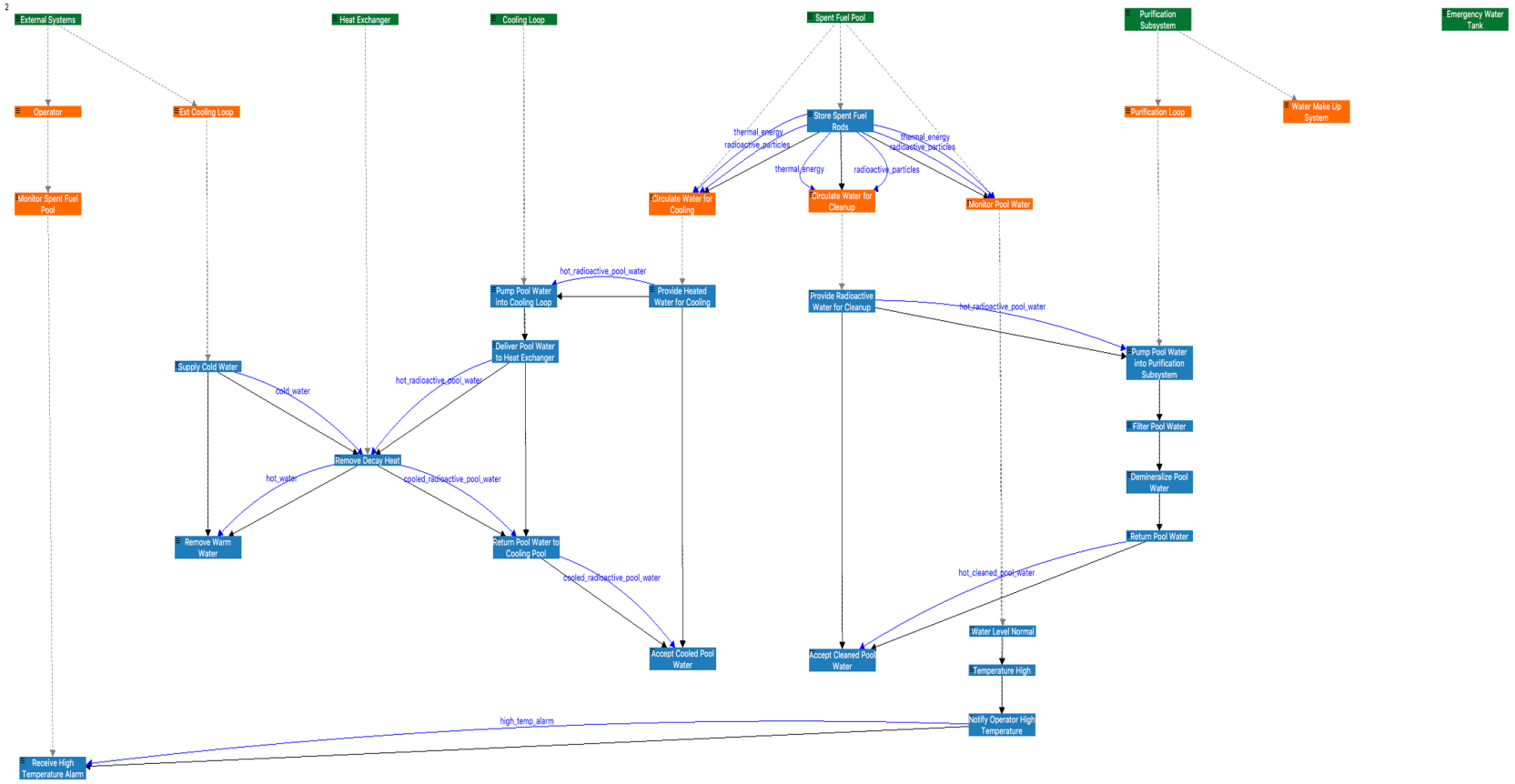


Figure 33. If event trace scenario found water level low from spent fuel pool, operator is notified by alarm activation and water is added from purification subsystem. SFP Cooling and cleaning model event trace scenario for water levels at normal and temperature high triggering high temperature alarm. Source: SPF.mp (2022).

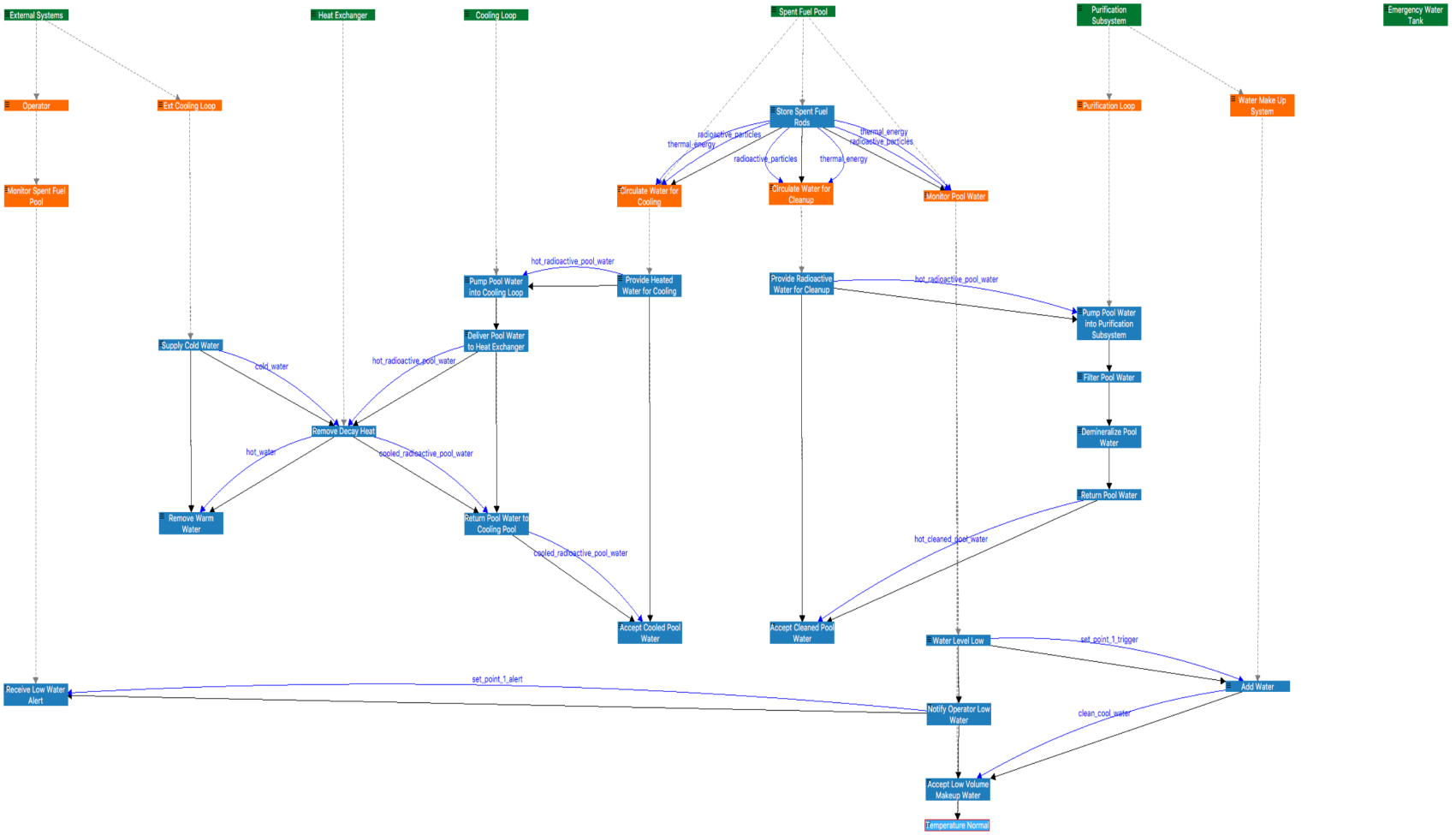


Figure 34. SFP cooling and cleanup model event trace for water levels low and temperature normal triggering low water alert. Source: SPF.mp (2022).

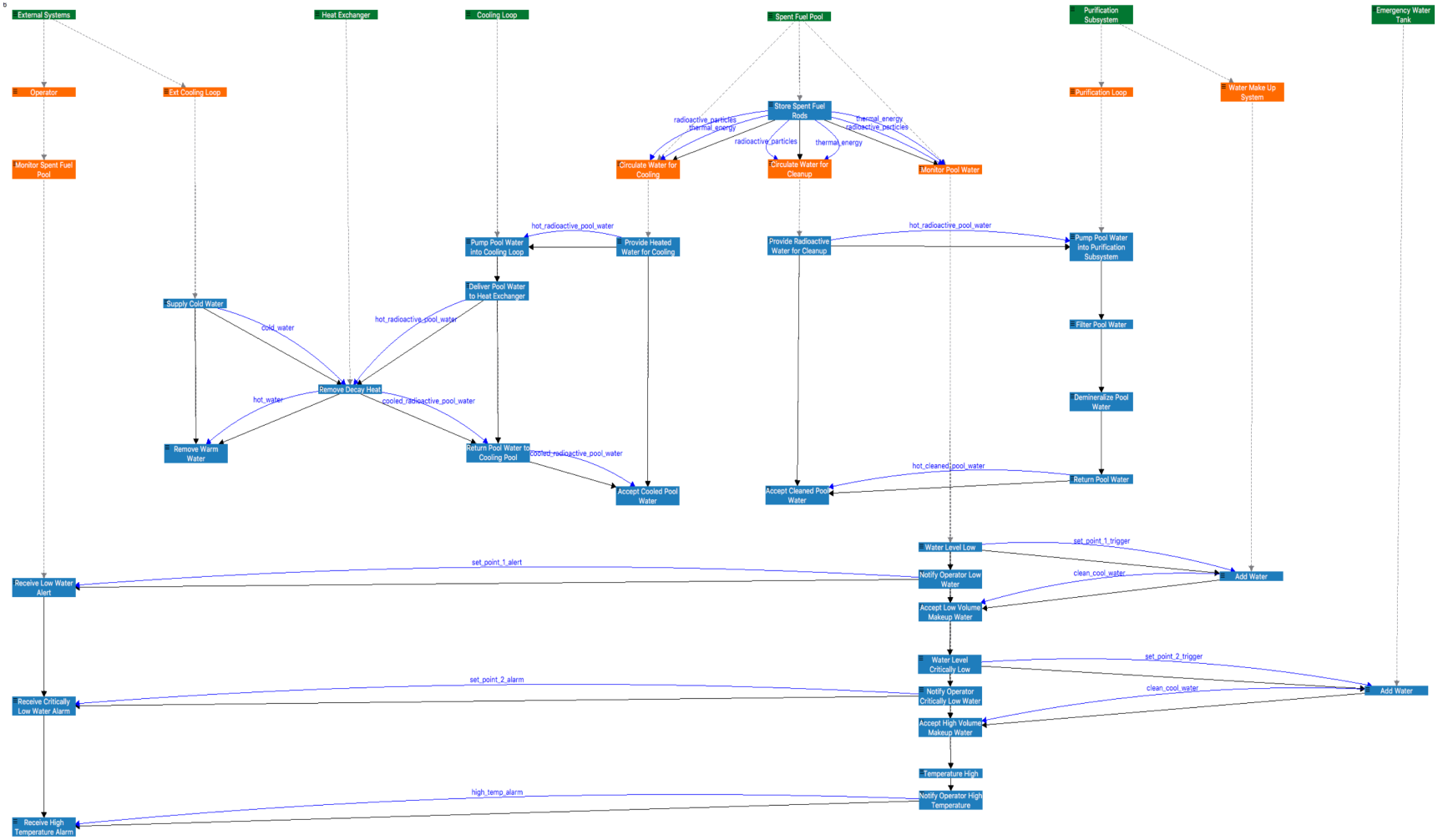


Figure 35. SFP cooling and cleanup model event trace for water at critically low levels triggering low water alarm and high temperatures triggering high temperature alarm. Source: SPF.mp (2022).

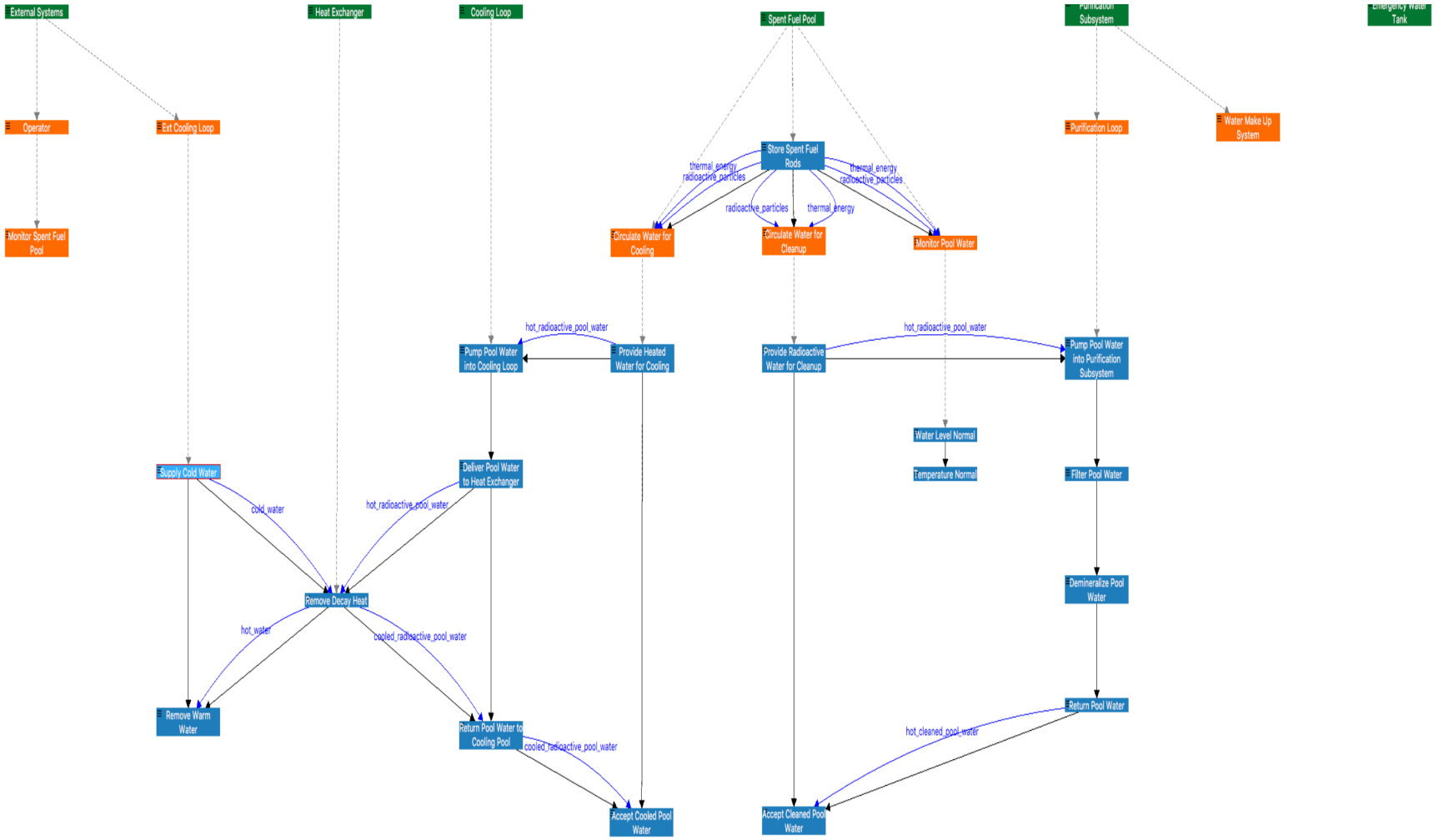


Figure 36. SFP cooling and cleanup model event trace for water levels at normal and temperature at normal, no action needed and normal monitoring continues. Source: SPF.mp (2022).

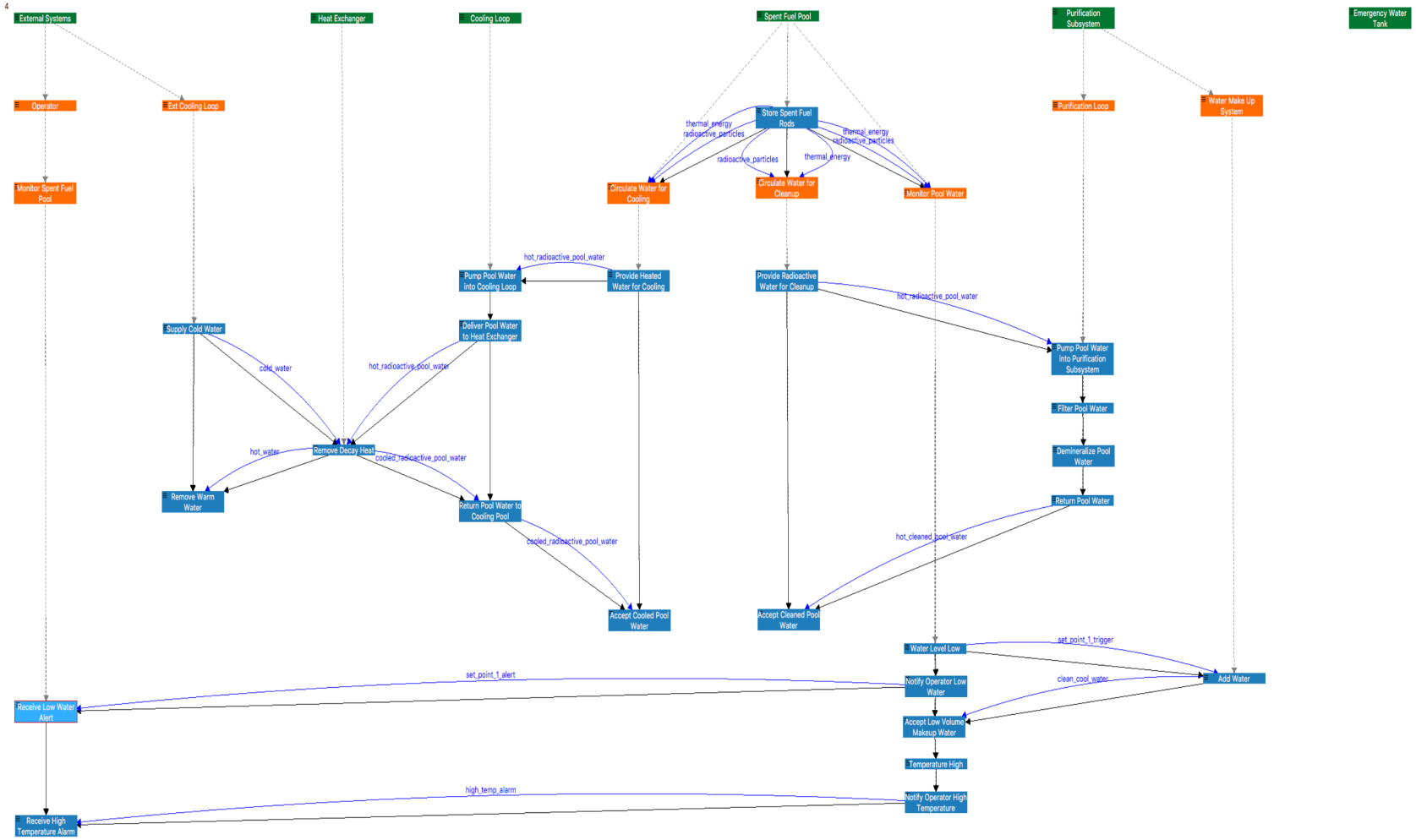


Figure 37. SFP cooling and cleanup model event trace for water levels low and temperature high triggering high temperature alarm and low water alert. Source: SPF.mp (2022).

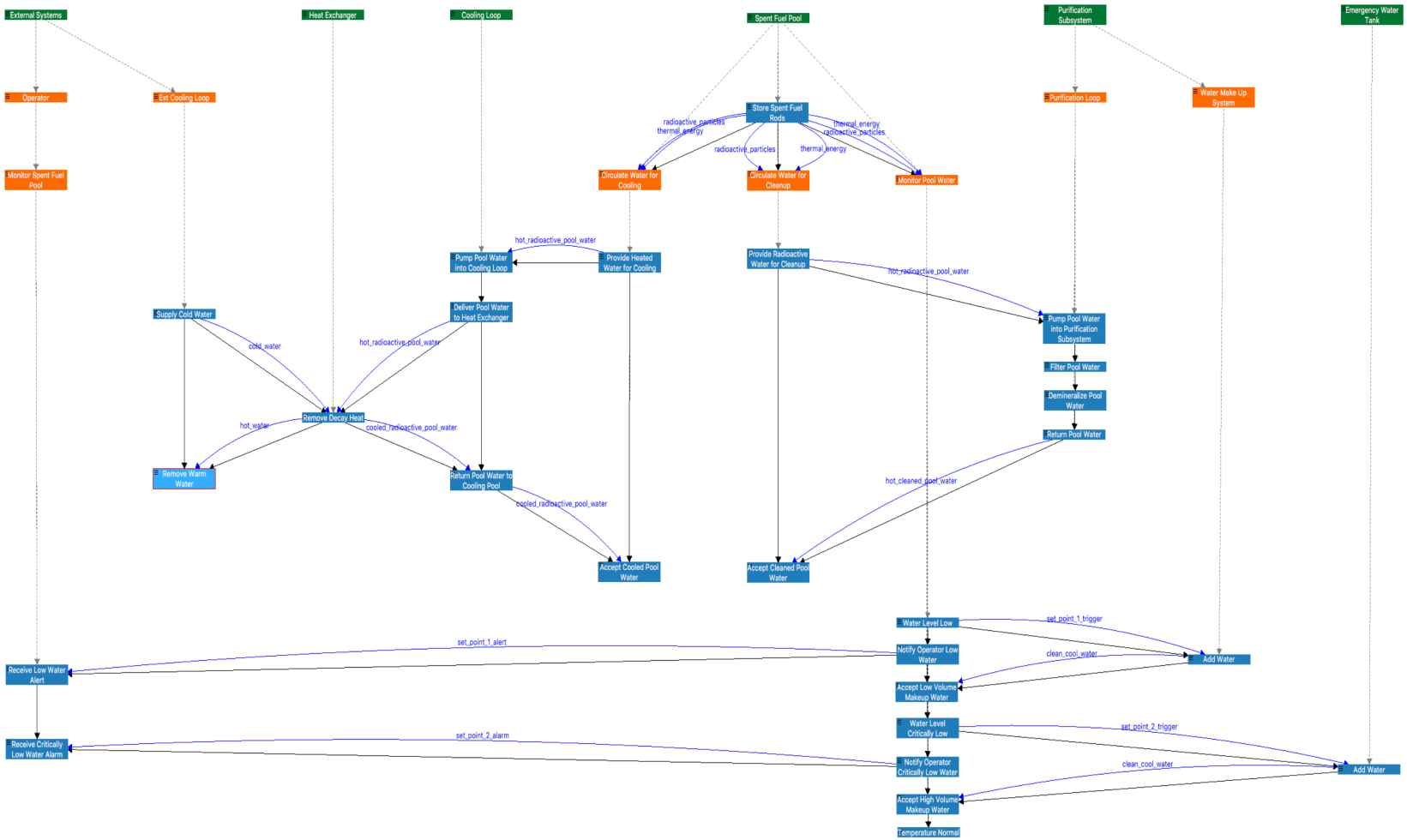


Figure 38. SFP cooling and cleanup model event trace for water levels critically low and temperature normal triggering low water alarm. Source SPF.mp (2022).

Mapping begins by selecting behavior-related DIAMOND ontology elements and associated attributes and relationships along with the spent fuel pool cooling and cleaning system elements for a crosswalk with MP event grammar.

A sample of nuclear reactor data stored in the DIAMOND ontology which includes author of submitted data, class, and definition is shown in Figure 39. Sample data selected include two system components for use in MP schema and event traces.

- Submitted by Jeren Browning
- Class: Reactor coolant pump
- Definition: A reactor coolant pump is a machine element that is made up of some material and is used to control the temperature of some asset.

- Submitted by Jeren Browning
- Class: Heat exchanger
- Definition: A heat exchanger is an equipment that uses the energy in some liquid or gas to transform it into an alternate state (e.g., from liquid to gas)

Figure 39. Sample DIAMOND ontology data. Source: DIAMOND (2019).

Figure 40 illustrates the layering of concepts from DIAMOND over the concepts appearing in the MP event traces using Figure 35 as an example. The mapping was created by selecting the DIAMOND ontology structure concepts of the designated Action class (light blue box) which DIAMOND defines as "the generated effects and those which may have pre-conditions before it can be executed" (Al Rashdan, Browning, and Ritter 2019, 13). The Action also includes transforming inputs into outputs which are represented in the ontology as triggers (orange box) serving the role of inputs or outputs. The Asset class (dark blue box) is defined in the DIAMOND ontology as that which "specifies an object, person, or organization that performs an action, such as a system, subsystem, component, or element" (Al Rashdan, Browning, and Ritter 2019). The green boxes were used to map user-defined relations which were applied to indicate input and outputs in MP in the event trace flow. And finally, the grey box was used to represent the external systems in the MP root event schema and mapped to assets in DIAMOND.

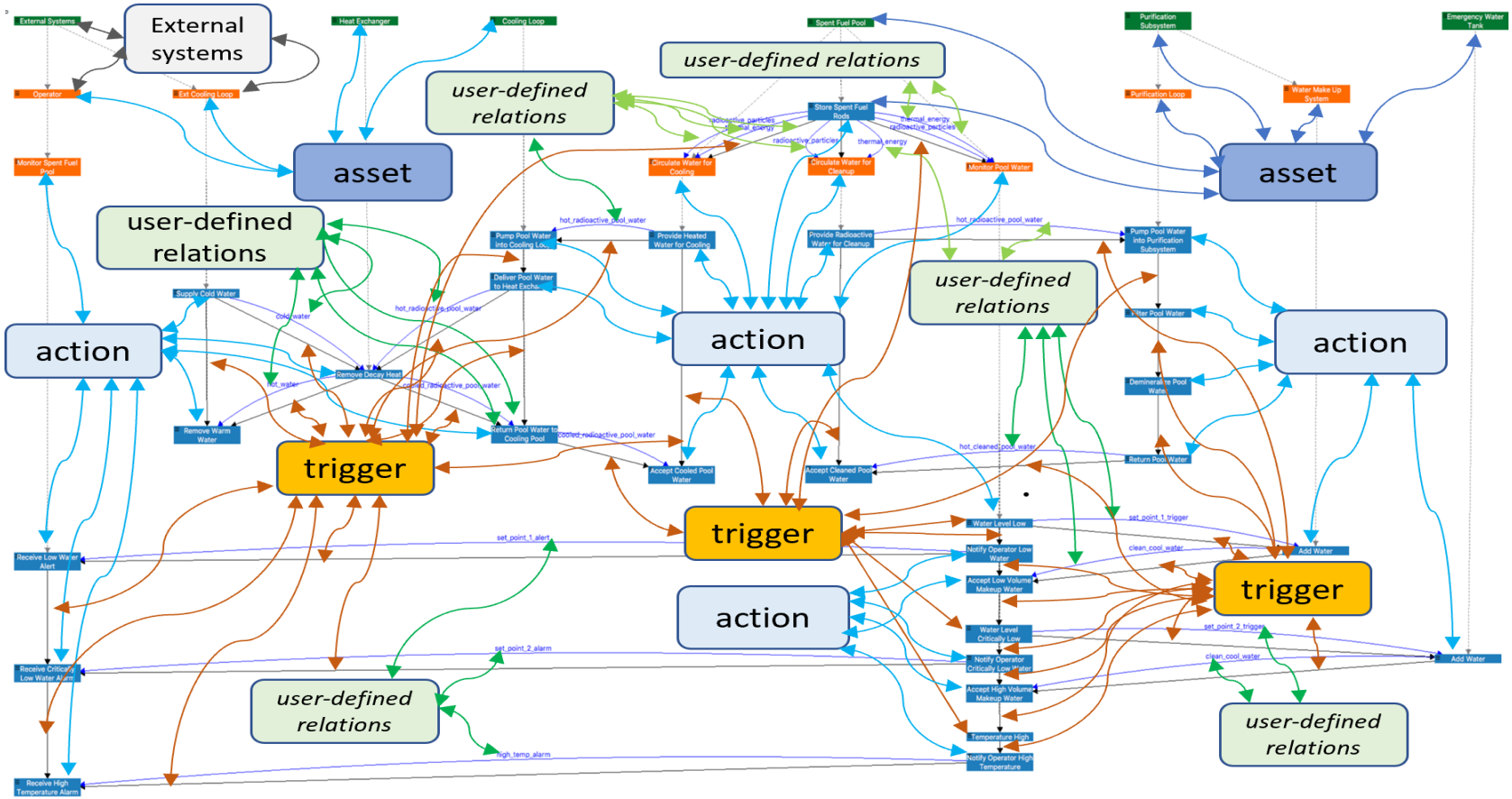


Figure 40. The mapping illustration shows the visual correlations between assets, actions, and triggers from DIAMOND to events and relationships in MP. Assets and actions are mapped to MP events, and triggers are mapped as precedence relations between MP events. The concepts of the correlating domain-specific references in the DIAMOND ontology to the concepts of the Monterey Phoenix spent fuel pool cooling and cleaning behavior model domain-specific scenario are shown. The purpose of the visual representation is for showing the analogous concepts between MP and DIAMOND visually by color mapping.

Triggers (controlling input/outputs) are indicated on the MP event trace between separate activities as horizontal and diagonal black arrows. To keep the mapping more succinct, not all triggers of the MP event trace are labeled on the illustration. Visually, one can locate, and trace numerous triggers depicted in the behavior model. There are also internal triggers represented by precedence relations in the event trace. These are illustrated by the vertical black lines and for purposes of this mapping, are not labeled. The user-defined relations provide a valuable tool for tracing and analyzing the modeled behavior events and for communicating the results and refining the behavior model. The user-defined relations (light green callout box) are labeled in the illustration mapping.

The selected event trace shows that the spent fuel pool cooling system has an external operator (MP orange box and grey callout box) and external cooling loop (orange box) as external systems (green box) where the operator monitors the spent fuel pool. The spent fuel pool (MP green box) stores spent fuel rods, circulates water for cooling by providing heated water for cooling and accepting cooled pool water, circulates water for, monitors pool water by monitoring for water levels. In this selected event trace, the water level is detected as being critically low, this then activates the low water alarm (action) to the external operator (asset). In this event trace, the high temperature alarm is also activated which notifies the operator. These alarms trigger the spent fuel pool to begin releasing accepting high volume make-up water. The water make-up system adds water from the emergency water tank until normal water level and normal temperature returns and normal monitoring continues. The external cooling loop (asset) then supplies cold water or removes warm water (action). The heat exchanger (asset) removes decay heat (action) and the cooling loop pumps pool water into cooling loop (action), delivers pool water to heat exchanger (action), and returns pool water (action) to cooling pool (asset).

The mapping visually demonstrates the overlaps between the DIAMOND ontology and an example MP model of nuclear reactor behavior. The research leading up to and including the mapping of the DIAMOND ontology functional system language to the corresponding MP functional behavior language elements validate support for the scoped nuclear energy spent fuel cooling pool reactor behavior model using the nuclear reactor SFP event traces from the MP behavior model.

THIS PAGE INTENTIONALLY LEFT BLANK

# IV. CONCLUSIONS AND RECOMMENDATIONS

This conclusion chapter first revisits the research objective and next summarizes the main technical contribution comparing DIAMOND ontology elements and MP event grammar. General findings and spent fuel pool application findings are then summarized. Finally, potential future work is described for using these frameworks for the modeling and analysis of nuclear reactor cooling system environments.

This thesis aimed to answer the following research question: What potential gaps or overlaps may exist in the DIAMOND ontology when evaluating with an appropriately scoped application of MP modeling for nuclear reactor behavior?

The research analysis successfully demonstrated mapping the DIAMOND ontology of the nuclear energy functional system language into analogous MP functional behavior language elements to validate a scoped scenario of the SFP cooling pool reactor behavior model. The answering of this research question was supported using the nuclear reactor SFP event traces from the MP behavior scenarios. The analysis was reviewed by an INL expert panel to ensure that the MP model scenarios were representative of operationally realistic behavior, and that the DIAMOND data was implemented correctly in the model.

A crosswalk between DIAMOND and MP revealed four general behavior-related findings between the concepts of DIAMOND and those defined in MP.

1. The ontology scope for DIAMOND is defined by classes, properties, and relationships that frame an expandable nuclear specific domain of system data. MP's ontology scope is limited to classes that pertain to behavior but can be applied to any system domain.
2. The DIAMOND ontology is constructed from classes that represent a wide range of objects and relationships between them. The MP behavior model is based on an abstract concept of event that can represent a range of DIAMOND concepts (e.g., asset or action), and precedence, inclusion, and user-defined relations.
3. The current DIAMOND ontology does not make a distinction for optional multiplicity of zero or more relationships and assumes there is always at least one existing relationship. Monterey Phoenix

behavior modeling includes zero-or-more relationships for event iterations in addition to one-or-more relationships.

4. The control flow method defined by a "trigger" class serves the role of Inputs/Outputs in DIAMOND. Monterey Phoenix does not have a fundamental class for information elements of Input/Output or a special property of "trigger," rather all concepts are modeled as events, even data, in terms of operations on the data, and the concept of trigger is implemented using precedence relationships. (MP)

The Spent Fuel Pool Cooling and Cleaning System MP behavior model application revealed a well-constrained behavior model. This model demonstrated components of and interactions among a spent nuclear fuel cooling pool and its environment.

3. The MP behavior model demonstrated the ability to generate the exhaustive set of nuclear reactor cooling pool behavior scenarios for visualization.

4. The MP model provided results supporting the ability of the DIAMOND ontology definitions to be used to organize and structure knowledge about a spent fuel pool's normal and off-normal behaviors.

5. The SPF example showed the application of assets, actions, and triggers from DIAMOND to events and relationships in MP. Assets and actions are represented as MP events, and triggers are represented as precedence relations between MP events.

6. Monterey Phoenix provided a viable approach for analyzing nuclear reactor system behavior consistent with the DIAMOND ontology.

Future work using these frameworks for the modeling and analysis of nuclear reactor cooling system environments may include analysts using MP to further inspect and challenge scenarios by commenting out constraints to draw connections and make discoveries of potential emerging behaviors within the system or system of systems.

Additional future work for nuclear reactor communities of interest may include each nuclear reactor energy system of systems to be elaborated on in MP models to communicate behaviors more fully with scientists and researchers monitoring the nuclear reactor systems of systems for emergent behaviors.

Future work in the DIAMOND ontology may include additional assembly of nuclear energy domain concepts and relationships by adding to the three key elements in the structural ontology layout for classes, object properties (relationships), and data properties (attributes). First, to differentiate hierarchy order of actions or activities, distinctions of subclass inclusion, or as analysist-defined relations in the classes to represent specific objects and the associated properties or attributes between those classes or objects. Secondly, to further distinguish optional multiplicity properties (relationships) of zero or more, one or more, and many to one to remove any assumption of known, as well as unknown properties (relationships). And third, to expand with a distinction of inputs and outputs for matter, energy, or information flows among actions. These additional concept efforts in the DIAMOND ontology may ease application of use by communities of interest seeking the objective of behavior analysis concepts for future modeling.

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX. SPENT_FUEL_COOLING_AND_CLEANUP_SYSTEM MP CODE

The combined MP code for the Model of Spent Fuel Cooling and Cleanup created by Kristin Giammarco on May 4, 2020. First modified by Kristin Giammarco with Douglas Van Bossuyt on May 8, 2020, and second modification on May 13, 2020. Code edited by Keane Reynolds in July 2021 and later edited by Pamela Dyer in July and August 2021. Spent_Fuel_Cooling_and_Cleanup_System file in Applications folder available for access on MP-firebird at https://firebird.nps.edu.

```
/*_____________________________
              Actors
_____________________________*/
SCHEMA Spent_Fuel_Cooling_and_Cleanup_System


ROOT External_Systems: { Operator, Ext_Cooling_Loop };


    Operator: Monitor_Spent_Fuel_Pool;

        Monitor_Spent_Fuel_Pool:    [ Receive_Low_Water_Alert ]
                                    [ Receive_Critically_Low_Water_Alarm ]
                                    [ Receive_High_Temperature_Alarm ]
                                    ;


    Ext_Cooling_Loop:   Supply_Cold_Water
                        Remove_Warm_Water;


ROOT Heat_Exchanger: Remove_Decay_Heat;


ROOT Cooling_Loop:  Pump_Pool_Water_into_Cooling_Loop
                    Deliver_Pool_Water_to_Heat_Exchanger
                    Return_Pool_Water_to_Cooling_Pool;


ROOT Spent_Fuel_Pool:   Store_Spent_Fuel_Rods
                        {   Circulate_Water_for_Cooling,
                            Circulate_Water_for_Cleanup,
                            Monitor_Pool_Water
                        };

    Circulate_Water_for_Cooling:    Provide_Heated_Water_for_Cooling
                                    Accept_Cooled_Pool_Water;

    Circulate_Water_for_Cleanup:    Provide_Radioactive_Water_for_Cleanup
                                    Accept_Cleaned_Pool_Water;

    Monitor_Pool_Water: (   Water_Level_Normal                      |

                            Water_Level_Low
                            Notify_Operator_Low_Water
                            Accept_Low_Volume_Makeup_Water

                            [   Water_Level_Critically_Low
                                Notify_Operator_Critically_Low_Water
                                Accept_High_Volume_Makeup_Water ]   )

                        (   Temperature_Normal          |
                                Temperature_High
                                Notify_Operator_High_Temperature    )
                        ;


ROOT Purification_Subsystem:    { Purification_Loop, Water_Make_Up_System };

    Purification_Loop:  Pump_Pool_Water_into_Purification_Subsystem
                        Filter_Pool_Water
                        Demineralize_Pool_Water
                        Return_Pool_Water;

    Water_Make_Up_System:   [ Add_Water ];


ROOT Emergency_Water_Tank:  [ Add_Water ];
```

```
/*_____________________________
                Interactions
_____________________________*/

COORDINATE  $a: Store_Spent_Fuel_Rods           FROM Spent_Fuel_Pool,
            $b: Circulate_Water_for_Cooling     FROM Spent_Fuel_Pool,
            $c: Circulate_Water_for_Cleanup     FROM Spent_Fuel_Pool,
            $d: Monitor_Pool_Water              FROM Spent_Fuel_Pool

    DO  ADD $a radioactive_particles $b;
        ADD $a thermal_energy $b;
        ADD $a radioactive_particles $c;
        ADD $a thermal_energy $c;
        ADD $a radioactive_particles $d;
        ADD $a thermal_energy $d;
OD;


COORDINATE  $a: Provide_Heated_Water_for_Cooling        FROM Spent_Fuel_Pool,
            $b: Pump_Pool_Water_into_Cooling_Loop       FROM Cooling_Loop
    DO  ADD $a PRECEDES $b;
        ADD $a hot_radioactive_pool_water $b;
    OD;


COORDINATE  $a: Deliver_Pool_Water_to_Heat_Exchanger    FROM Cooling_Loop,
            $b: Remove_Decay_Heat       FROM Heat_Exchanger
    DO  ADD $a PRECEDES $b;
        ADD $a hot_radioactive_pool_water $b;
    OD;


COORDINATE  $a: Supply_Cold_Water   FROM External_Systems,
            $b: Remove_Decay_Heat   FROM Heat_Exchanger,
            $c: Remove_Warm_Water   FROM External_Systems
    DO  ADD $a PRECEDES $b;
        ADD $b PRECEDES $c;
        ADD $a cold_water $b;
        ADD $b hot_water $c;
OD;


COORDINATE  $a: Provide_Radioactive_Water_for_Cleanup   FROM Spent_Fuel_Pool,
            $b: Pump_Pool_Water_into_Purification_Subsystem     FROM Purification_Subsystem
    DO  ADD $a PRECEDES $b;
        ADD $a hot_radioactive_pool_water $b;
OD;


COORDINATE  $a: Remove_Decay_Heat                       FROM Heat_Exchanger,
            $b: Return_Pool_Water_to_Cooling_Pool       FROM Cooling_Loop
    DO  ADD $a PRECEDES $b;
        ADD $a cooled_radioactive_pool_water $b;
OD;


COORDINATE  $a: Return_Pool_Water_to_Cooling_Pool   FROM Cooling_Loop,
            $b: Accept_Cooled_Pool_Water            FROM Spent_Fuel_Pool
    DO  ADD $a PRECEDES $b;
        ADD $a cooled_radioactive_pool_water $b;
    OD;


COORDINATE  $a: Return_Pool_Water           FROM Purification_Subsystem,
            $b: Accept_Cleaned_Pool_Water   FROM Spent_Fuel_Pool
    DO  ADD $a PRECEDES $b;
        ADD $a hot_cleaned_pool_water $b;
    OD;

```

```
191
192 /* If Water Level Low */
193
194 COORDINATE  $a: Water_Level_Low             FROM Spent_Fuel_Pool,
195             $b: Add_Water                   FROM Purification_Subsystem
196     DO  ADD $a PRECEDES $b;
197         ADD $a set_point_1_trigger $b;
198
199     OD;
200
201 COORDINATE  $a: Notify_Operator_Low_Water   FROM Spent_Fuel_Pool,
202             $b: Receive_Low_Water_Alert             FROM External_Systems
203     DO  ADD $a PRECEDES $b;
204         ADD $a set_point_1_alert $b;
205     OD;
206
207
208 COORDINATE  $a: Water_Level_Critically_Low      FROM Spent_Fuel_Pool,
209             $b: Add_Water                       FROM Emergency_Water_Tank
210     DO  ADD $a PRECEDES $b;
211         ADD $a set_point_2_trigger $b;
212     OD;
213
214 COORDINATE  $a: Notify_Operator_Critically_Low_Water    FROM Spent_Fuel_Pool,
215             $b: Receive_Critically_Low_Water_Alarm          FROM External_Systems
216     DO  ADD $a PRECEDES $b;
217         ADD $a set_point_2_alarm $b;
218     OD;
219
220
221 COORDINATE  $a: Add_Water                           FROM Purification_Subsystem,
222             $b: Accept_Low_Volume_Makeup_Water      FROM Spent_Fuel_Pool
223     DO  ADD $a PRECEDES $b;
224         ADD $a clean_cool_water $b;
225     OD;
226
227
228 COORDINATE  $a: Add_Water                           FROM Emergency_Water_Tank,
229             $b: Accept_High_Volume_Makeup_Water     FROM Spent_Fuel_Pool
230     DO  ADD $a PRECEDES $b;
231         ADD $a clean_cool_water $b;
232     OD;
233
```

# LIST OF REFERENCES


Ahmad, Al Rashdan, Browning, Jeren, and Ritter, Christopher. 2019. Data Integration Aggregated Model and Ontology for Nuclear Deployment (DIAMOND): Preliminary Model and Ontology. INL/EXT-19-55610.

Ali, Mir F. 2013. *Nuclear Energy: Peaceful Ways to Serve Humanity*. WordPress. https://nuclearenergyebook.wordpress.com.

Anderson, T. 2004. Introducing XML. Retrieved on January 31, 2022. from http://www.itwriting.com/xmlintro.php

Arp, Robert, Smith, Barry, and Spear, Andrew. 2015. "Principles of Best Practice II: Terms, Definitions, and Classification," in *Building Ontologies with Basic Formal Ontology*, MIT Press.

Auguston, Mikhail. 2009. "Software Architecture Built from Behavior Models." Monterey, California. Naval Postgraduate School.

———. 2020. "Monterey Phoenix System and Software Behavior Modeling Language (Version 4.0)." Documentation - Monterey Phoenix - NPS Wiki. February 14, 2022. https://wiki.nps.edu/display/MP/Documentation

Desai, Rusita H. 2021. "Reusable Monterey Phoenix Code Libraries for Behavior Models and Model Segments." Master's Thesis, Naval Postgraduate School. https://calhoun.nps.edu/handle/10945/68319

In Press. Giammarco, Kristin. "Exposing and Controlling Emergent Behaviors Using Models with Reasoning," Chapter 3 in Emergent Behavior in System of Systems: Real World Applications, Rainey, L.B., Holland, O.T.., Eds.; CRC Press Taylor & Francis Group: Boca Raton, FL, USA.

Giammarco, Kristin. 2017."Practical Modeling Concepts for Engineering Emergence in Systems of Systems."

———. 2019 'Exposing and Controlling Emergent Behaviors in a Systems of Systems (SoS) Model."

———. 2020. Monterey Phoenix Model of Spent Fuel Cooling and Cleanup. May 4, 2020. Accessed January 5, 2022. Naval Postgraduate School. https://www.firebird.nps.edu

Giammarco, K., Auguston, M. 2019. "Part 3: Modeling Systems and Interactions." Monterey Phoenix. https://wiki.nps.edu/display/MP

Giammarco, Kristin, and Kathleen Giles. 2018. Verification and Validation of Behavior Models Using Lightweight Formal Methods. Springer.

Giammarco, Kristin, Mikhail Monica Auguston, W. Clifton Baldwin, Ji'on Crump, and Monica Farah-Stapleton. 2014. "Controlling Design Complexity with the Monterey Phoenix Approach."

Gruber, Thomas. 1992. A Translation Approach to Portable Ontology Knowledge Acquisition. Report KSL 92–71. Standford, CA: Knowledge Systems Laboratory.

Gruber, Thomas. 1993. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Report KSL93-04. Standford, CA: Knowledge Systems Laboratory.

Guarino, Nicola and Musen, Mark. 2015. "Applied ontology: The next decade begins." *Applied ontology*.10.

International Atomic Energy Agency. 1988. Component Reliability Data for Use in Probabilistic Safety Assessment. IAEA TECDOC-478.

International Atomic Energy Agency. 1997. Generic Component Reliability Data for Research Reactor PSA, IAEA TECDOC- 930

International Nuclear Safety Advisory Group. 1986. "Post-Accident Review Meeting on the Chernobyl Accident." September. INSAG-1, https://www.iaea.org/publication/3786/the-chernobyl-accident-updating-of-insag-1.

———. 1992. The Chernobyl Accident: Updating of INSAG-1. INSAG-7. https://www-pub.iaea.org/MTCD/publications/PDF/Pub913ae_web.pdf

Kelly, John. 2015. Office of Nuclear Energy Response to Fukushima Dia-ichi Accident. U.S. Department of Energy. https://www.energy.gov/sites/prod/files

Musen, M.A. 1992. Dimension of Knowledge Sharing and Reuse. Computers and Biomedical Research. 25

Lexico. Ontology. 2019. Retrieved January 2022 from: https://en.oxforddictionaries.com/definition/ontology

Life cycle Modeling Organization. 2017. Life cycle Modeling Language (LML) Specification. December 1, 2017. Retrieved August 2021. http://www.lifecyclemodeling.org/spec/LML_Specification_1_1.pdf.

Nuclear Regulatory Commission. 1979. *Three Mile Island: A Report to the Commissioners and to the Public*. Volume 1. April 4, 1979. Thermal TMI. https://www.osti.gov/servlets/purl/5395798.

———. 2018. *NRC Response to Lessons Learned from Fukushima.* September 2018. https://www.nrc.gov/ML1126/ML112660383.pdf

Noy Natalya F., and Deborah L. McGuinness, 2001. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory.

Ochiai, Eiichiro. 2014. "Nuclear Power and Radiation." In Hiroshima to Fukushima, 15. 10.1007/978-3-642-38727-2_2.

Office of the Federal Register. 1979. *Jimmy Carter, President's Commission on the Accident at Three Mile Island Online by Gerhard Peters and John T. Woolley*. Executive Order 12130. The American Presidency Project https://www.presidency.ucsb.edu/node/249805

Office of the Press Secretary. 2013. Presidential Policy Directive – Critical Infrastructure Security and Resilience. PPD-21. Washington, D.C., Whitehouse. https://obamawhitehouse.archieves.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrasture-security-and-resilience

Union of Concerned Scientists. 2016. "Nuclear Spent Fuel Damage: Pool Accident." April 19, 2016. https://allthingsnculear.org/dlochbaum/nuclear-spent-fuel-damage-pool-accident.

U.S. Nuclear Regulatory Commission. Westinghouse Technology Systems Manual. Section 14.4 Spent Fuel Pool Cooling and Cleanup System. Rev 0109. https://www.nrc.gov/docs/ML1215/ML12158A334.pdf.

Raggett, Dave, Lam, Jenny, Alexander, Ian, Kmiec, Michael. 1998. Raggett on HTML 4, 2nd Ed. Addison-Wesley Logman Publishing Co., Inc.

Withgott, Jay H. and Brennan, Scott R. 2008. "Environment; the Science Behind the Stories, 3d Ed." Pearson Education, Inc.

Whitcomb, Clifford A., Mikhail Auguston, and Kristin Giammarco. 2015. Composition of Behavior Models for Systems Architecture. John Wiley & Sons.

World Nuclear Association. 2021. "World Nuclear Performance Report." April 2021. https://world-nuclear.org/Information-Library/Fukushima-Daiichi-Accident.aspx

THIS PAGE INTENTIONALLY LEFT BLANK

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California