Model-Centric Interdependent Critical Infrastructure System Recovery Analysis and Metrics

Kevin Joseph Russell

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

In

Electrical Engineering

Robert P. Broadwater

Lamine M. Mili

Alan J. Brown

William T. Baumann

Dushan Boroyevich

11 April 2016

Blacksburg Virginia

Keywords: Recovery Analysis, Reconfiguration for Fault Isolation and Restoration, Damage Control Automation, Mission Assurance, Critical Infrastructure System

Model-Centric Interdependent Critical Infrastructure System Recovery Analysis and Metrics

Kevin Joseph Russell

Abstract

This dissertation defines and evaluates new operations management modeling concepts for use with interdependent critical infrastructure system reconfiguration and recovery analysis. The work combines concepts from Graph Trace Analysis (GTA), Object Oriented Analysis and Design (OOA&D), the Unified Modeling Language (UML) and Physical Network Modeling; and applies them to naval ship reduced manned Hull, Mechanical, Electrical and Damage Control (HME&DC) system design and operation management. OOA&D problem decomposition is used to derive a natural solution structure that simplifies integration and uses mission priority and mission time constraint relationships to reduce the number of system states that must be evaluated to produce a practical solution. New concepts presented include the use of dependency components and automated system model traces to structure mission priority based recovery analysis and mission readiness measures that can be used for automating operations management analysis. New concepts for developing power and fluid system GTA loop flow analysis convergence measures and acceleration factors are also presented.

Model-Centric Interdependent Critical Infrastructure System Recovery Analysis and Metrics

Kevin Joseph Russell

General Audience Abstract

This work uses naval ship emergency management as an example problem to define a new approach for automating reconfiguration and recovery analysis. Standard ship and utility system reconfiguration analysis are used to identify switchable devices that can be operated to isolate failed components and restore service. Standard naval ship recovery analysis is used to evaluate reconfiguration in regard to the time it takes to complete switching operations and the effect through time that those operations have on defined mission readiness levels. A unique feature of the approach presented is the use of automated traces to derive mission priority relationships between distribution components and service loads directly from a graphical representation of the system. These relationships can be used to reduce the number of system states that need to be evaluated to generate a practical solution. Their use also helps to align analysis with standard emergency management procedures and data. The new approach uses a number of existing concepts from emergency management, software architecture design, and multidiscipline physical network modeling. As a result, the approach can be used with any system that can be drawn out as a network of components with definable through and across characteristics.

Acknowledgements

I would like to thank Dr. Broadwater for his support towards the development of the work presented in this dissertation. I would also like to thank Dr. Lamine Mili, Dr. Alan Brown, Dr. William Baumann, and Dr. Dushan Boroyevich and for serving on my committee.

I would like to thank the Office of Naval Research, and the Army Corps of Engineers Engineering Research and Development Center, Construction Engineering Research Laboratory for their support of model-based interdependent system analysis research under Contract Numbers N00014-05-C-0162 and W9132T-06-C-0038.

I would like to thank Dr. Lynn Feinauer and Dr. David Kleppinger for their contribution to the development of Graph Trace Analysis fluid flow and reconfiguration analysis.

Finally, I would like to thank my wife for patiently listening to me talk about how to automate reduced manned ship operations for twenty years, and to God, who makes long paths straight.

Table Contents

Figures

Tables

Chapter 1 Introduction

Increasing costs, load growth, reliability, resiliency, security, and environmental concerns are driving infrastructure systems to become more integrated and more complex. Navy and Coast Guard automation for reduced manning has driven naval ship design and operation towards the use of growing automation and system integration to reduce personnel costs, improve both efficiency and survivability, and provide increased electrical system capacity for next generation systems. The need to make utility systems more resilient against threats from terrorism and climate change are pushing government and the utility industry to make transmission and distribution systems more interconnected, more automated and more dependent on diverse generation sources. These problems call for the development of new unified modeling and analysis approaches that are fast enough to support real-time operation management and control, and can be applied seamlessly across a larger number of interrelated system types and problem domains. Many of the critical infrastructure related analysis and operation management approaches used today are based on separately developed, domain area specific methods that can be difficult to integrate, implement and maintain.

This dissertation looks at naval ship reduced manned Hull, Mechanical, Electrical and Damage Control (HME&DC) system design and operation as an integrated critical infrastructure system problem using a multidiscipline system operation management point of view. Many of the concepts discussed in this work apply to other areas. This includes commercial power, water and gas utility systems, offshore oil platforms, and industrial processes like steel and pharmaceutical production that can experience a significant financial loss if support systems fail. The main reasons for using reduced man ship HME&DC system design and operation management for problem definition and solution development is that the principles and practices that drive it are well defined and reduced manning pushes efficiency and survivability "trade space"[1] issues to levels that are difficult to address using existing approaches.

[1] The term "trade space" (41) refers to coordinating design objectives such as efficiency and survivability, which have mutually exclusive or competing criteria and constraints.

One of the leading concepts used in the research presented is Object Oriented Analysis and Design (OOA&D) problem decomposition. The OOA part of OOA&D is used to divide complex problems into non-domain specific fundamental objects that can be combined in different ways to define solutions that leverage natural integration structures inherent within the problem itself. More traditional approaches start by dividing complex problems into simpler domain area specific pieces, with the primary focus placed on optimizing the performance of each piece. For complex integrated systems and activities such as reduced manned HME&DC, overall integration is often addressed at the end as a secondary factor, or as unintended consequences discovered after solution pieces have been implemented and put into use. For interdependent critical infrastructure system problems, significant "commonality" (1) can exist between similar functions and data that occur across different system types and problem domains. Approaches that ignore complex problem cross system, cross domain commonality can create significant analysis and data management problems that are difficult to define and measure.

Dealing with commonality related integration issues is particularly critical for systems and activities like HME&DC design and operation that depend on reconfigurable system resources and processes where changes in priorities, system operation constraints and system component status can significantly affect function and data relationships. In the OOA&D solution presented in this dissertation, a shared component-based infrastructure system model that is kept current with its real-world counterpart is combined with model-based automated system trace algorithms and data management. This architecture directly addresses the changing function and data relationships that make interdependent critical infrastructure system analysis complex.

1.1 Research Objective and Contribution

The research presented in this dissertation uses observations from shipboard HME&DC system design and operation to define new problem definition and solution concepts using a multidiscipline system operation management point of view. This perspective plays a major role in the discrete event and steady-state analysis development presented, which is intended to fit well with and also take advantage of how system operators operate and manage infrastructure systems. Much of the interdependent system failure effect and recovery analysis research conducted to date has focused on probability and transient analysis based solutions. Specific benefits from development of interdependent model-based discrete event and steady-state analysis addressed in the problem discussion presented in this work include:

- Defining recoverability analysis criteria and constraints in terms of mission priority and component status can significantly reduce the number of states that must be considered
- Model-based discrete event and steady-state analysis can be run at speeds that support real-time operations management and control for large systems
- Identifying the parts of the overall interdependent system analysis problem that can be addressed using discrete event and steady-state analysis will reduce and simplify the parts of the problem that are best addressed using probability and transient analysis
- Detailed deterministic discrete event and steady-state model-based analysis can be used as a reference for:
    - Structuring design, operation and administrative data management
    - Evaluating partial and uncertain system operation information
- Well trained people with detailed understanding of a given set of interdependent infrastructure systems are generally good at being able to work effectively with partial and uncertain information if they have a detailed understanding of system topology and discrete event and steady-state system behavior
- Model-based analysis can be used to provide detailed system information for:
    - People that do not have extensive experience with the systems being operated
    - People that have extensive experience that work with systems that are too large for a person to effectively memorize

Contributions presented in this dissertation which provides the main elements needed to extend Graph Trace Analysis (GTA) for use in interdependent system operations analysis using a generic, homogeneous analysis conventions and algorithms include:

- Simplifying time-based interdependent system analysis using operations management emergency response objectives and time constraints
- Modeling forward-backward sweep relationships between mission priorities, missions and support services using dependency, activity and event components
- Defining automated trace-based forward-backward sweep propagation rules for failure effects and mission priorities using dependency components
- Using dependency components and Physical Network Modeling to structure forward-backward sweep effort and flow relationships across interconnection points between different system types
- Development and evaluation of loop convergence measures and acceleration factors for use in forward-backward sweep generic system flow analysis
- Development of interactive system traces which replace the need for the use of complex cost functions for evaluating system reconfiguration options

Chapter 2 Literature Review

A number of papers address commercial power utility and shipboard electrical system reconfiguration, which is a key part of integrated critical infrastructure system design and operation. Many of the concepts presented in these papers could be applied to the overall interdependent system design and control, but would be difficult to combine into a comprehensive solution because they are based on separately developed solution approaches that may or may not fit well with both design and operation management. Recent papers that discuss interdependent reconfigurable system design analysis and measures include:

- Cramer, Sudhoff and Zivi (2) and Chan, Sudhoff, Lee and Zivi (3) which presents the development of warship performance metrics, and the use of those metrics together with integrated system simulation and linear programming to perform integrated ship system operability and dependability design analysis. The simulation used to structure analysis models each system type separately in layers that are coordinated together to produce an integrated solution.

- Doerry and Clayton (4) which defines Quality of Service measures that quantify the effect of power continuity on loads that support ship missions
- Weston, Balchanos, Koepp and Marvis (5) which presents a simulation framework that integrates analysis of individually implemented, multidiscipline subsystem component models
- Xu, Nozick, Turnquist and Jones (6) which proposes the use of graphs (links and nodes) to represent interconnections between major infrastructure components, the use of Markov and semi-Markov chains to model changes in links between components, and dividing analysis into major time periods
- Tam (7) that reviews several interdisciplinary analysis approaches, discusses their potential application to large reconfigurable systems, and addresses several important development issues

Chapter 3 Interdependent System Analysis Problem and Solution Definition

Interdependent critical infrastructure system modeling and analysis development work performed for the research presented in this dissertation is based on the use of Graph Trace Analysis (GTA) (8) and (9), Object Oriented Analysis and Design (OOA&D) (1), the Unified Modeling Language (UML) (10) and Physical Network Modeling (11) and (12). OOA&D and UML are used to review and organize the major parts of ship engineering service and damage control system related operation practice, design and data processing into a unified problem with simplified goals and constraints. Graph Trace Analysis and Physical Network Modeling conventions are used to structure forward-backward sweep steady-state flow analysis together with dependency component concepts from UML that are used to model discrete event priority and component status information propagation. This is worked together to structure interdependent system measures and analysis techniques to simulate and quantify damage effect propagation and recovery.

OOA&D was originally developed for design of software systems. The Unified Modeling Language (UML) is a software architecture modeling language used in OOA&D. UML has been extended to a number of different disciplines through development of UML profiles such as the System Modeling Language (SysML) (13) and the Business Process Modeling Language (BPML) (14). The OOA part of OOA&D uses the problem domain to identify fundamental responsibilities, natural lines of decomposition, and fundamental objects that can be organized in different ways to solve different problems. The OOD part of OOA&D is used to refine problem domain solution architectures outlined during the OOA phase using standard patterns, template libraries and other best practice solutions (15).

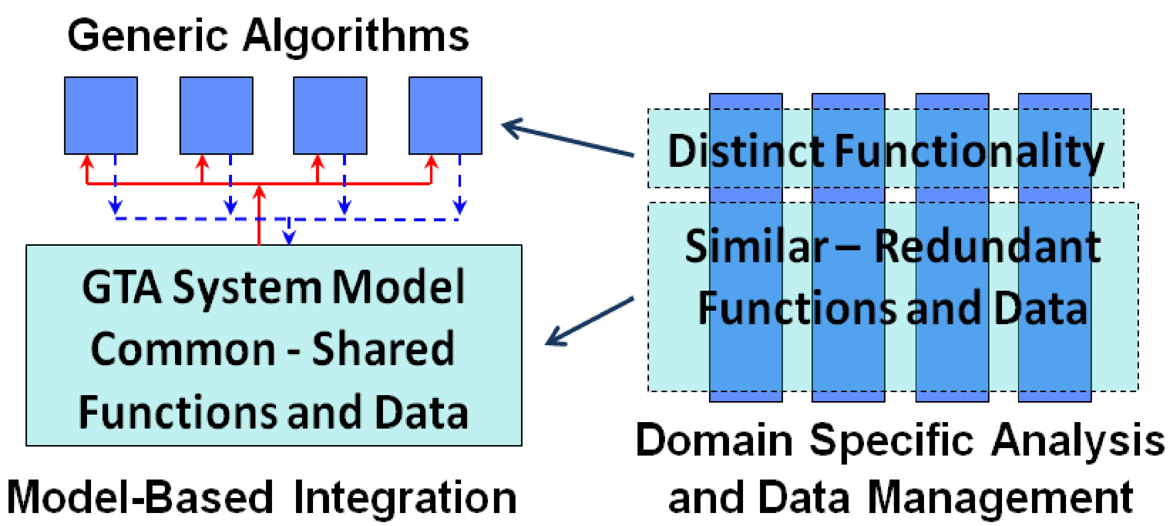

Figure 1 GTA Model-Based vs. Standard Integration

GTA can be used for engineering system analysis in ways that are similar to how OOA&D is used in information management to decompose analysis and data along multidiscipline lines of major responsibility and commonality (1), (8) and (9). Commonality can be used to categorize and leverage low-level common factors among components belonging to different system types and problem domains. Commonality is different from hierarchical structuring. Commonality focuses on the similarity between things that belong to different domains at the same level of abstraction. Hierarchy focuses on how components and systems relate to each other across different levels of abstraction. Both can be used together to define complex system problem definition and solution architectures. In GTA, commonality based fundamental component and system level behavior and data are abstracted from the different system, process and data domains into common and distinct algorithms and data. Common algorithms and data are combined and managed together in a shared model. Remaining distinct applications and data are managed and associated, or integrated together through attachment to the shared model. See Figure 1.

The engineering analysis and information management concepts used in GTA are based on a combination of principles originally developed in Physical Network Modeling, Physical System Modeling and Generic Programming. Physical Network Modeling and Physical System Modeling were formulated in the 1950's and 60's to define new approaches for interdependent system analysis (11) and (12). Physical Network Modeling looks at multidiscipline analysis from an electrical network point of view. Physical System Modeling addresses multidiscipline problems from more of a mechanical engineering perspective. The majority of the work done in this area has been in the development of Bond Graphs (16) and software modeling frameworks such as Modelica which have tended to concentrate on dynamic modeling of integrated systems at the mechanism and subsystem level. GTA is built on many of the same multidiscipline analysis concepts as Bond Graphs, but is focused more on analyzing large systems where discrete and steady-state analysis is used to evaluate system-wide behavior. The primary system analysis approach used in GTA is radial system forward-backward sweep iteration combined with different methods for solving loop flows modeled as cotree connection points between tree radial path end components.

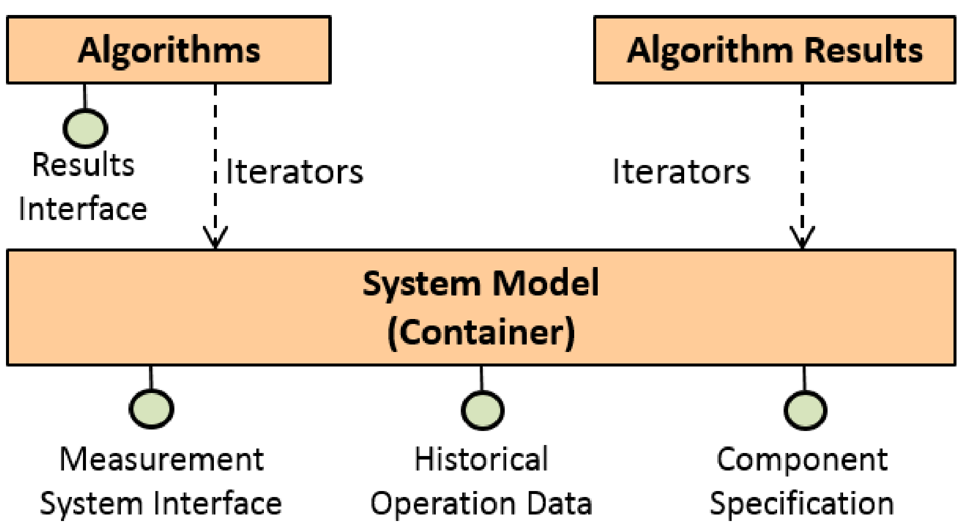

Figure 2 GTA High-Level Architecture

Generic programming is a collection of advanced information management concepts that are implemented in the C++ Standard Template Library (15). For example, a generic sort algorithm that uses iterators provided by a container can be used with any type of object stored in the container, as long as objects can be compared with each other in terms such as “larger than.” Iterators can be implemented with pointers, which “point to” specific locations in memory. GTA uses a special type of iterator called a “topology” iterator that is defined and managed by the system model “container.” See Figure 2.

The term “topology iterator” is used because the iterators process through and define an edge-edge graph that is used to describe the topology of the system. The majority of topology iterators are defined as the model is built. As components are added, modified, deleted or change state in a way that affects topology; their topology iterators are automatically updated. The system component “objects” stored in the model relate to each other through the use of iterators. Iterators are also used to drive analysis and data management. Data is referenced directly to components using iterators which act similar to relational database keys, which are automatically resynchronized as the model is updated. When component state or system topology changes occur, only the iterators that are directly affected by the change need to be updated. This makes it possible to analyze large models quickly even when a large number of configuration changes occur (17).

Algorithms share data and work together to perform analysis through attachment to the model using topology iterators provided by the container. When an algorithm completes a calculation, it attaches results to corresponding components in the model. When an algorithm needs data, it gets it from the model. Algorithms can define their own iterators and also use iterators provided by other algorithms. This constitutes a new paradigm in integrated system analysis that provides the capability to use collaborative algorithms to derive temporal, spatial and physical relationships between components directly from a networked “object” drawing which behaves much in the same way as its real world counterpart.

3.1 GTA System Level Analysis

GTA divides analysis into component and system level equations. System equations are the same for all system types and are based primarily on the use of forward-backward iteration. Component equations are defined in terms of across (e.g., voltage, pressure) and through (e.g., current, mass flow) characteristics, according to system type (power, fluid, gas, thermal and more) and analysis type (discrete, steady-state, and transient). The majority of GTA development completed to date has focused on power system discrete event and steady-state analysis.

GTA discrete and quasi-steady-state analysis, or time-sequence analysis, is performed by running discrete and steady-state analysis sequentially across a set of discrete time points. For each time step, time stamped component measurement and status data are downloaded from available data sources such as Supervisory Control and Data Acquisition (SCADA) and Advanced Metering Infrastructure (AMI) systems, and attached to the model at the data's respective component. When a discrete status measurement (such as open, close or fail) is set on a component, it reacts by changing its state to the one indicated by the measurement. Adjacent components then react according to feeder path associations until loss of service and discrete control related state change actions such as automatic bus transfer (ABT) switch operation are propagated throughout the system. Steady-state analysis is then run to estimate flow related results such as voltage and current at every component in the system. Autonomous devices such as voltage regulators and switched capacitors are operated as part of steady-state analysis. Analysis variables that exceed specified component operation limits are flagged and then reported as constraint violations. These flags can then used by other applications to drive higher level analysis.

3.2 GTA Component Level Modeling and Analysis

Many of the component level behaviors and data used to model different systems and facilitate different types of engineering analysis are similar. For example, the behaviors behind simulating opening and closing a circuit breaker, valve, or vent closure can be defined using the same code if written in a generic way. The formal name for this concept is "Composition Based

Polymorphism" (8) and (18). Most analysis and data management approaches, including Geographic Information Systems (GIS) and Computer-Aided Design (CAD) systems, use "Inheritance Based" structures that are defined vertically along discipline and system specific lines of decomposition. Composition based approaches provide the capability to cut horizontally across different system types which make it possible to take advantage of commonality among components from different system and process domains. For example valves, switches and doors all "open" and "close," and can all "fail." Composition Based Polymorphism (18) and Commonality (1) are relatively new concepts.

Inheritance has long been considered the default object oriented approach to provide for reuse of code, which is one of the reasons it is so pervasive in data management and analysis software. Inheritance uses a parent class to define a base set of attributes and behaviors that its subclasses inherit. The subclasses then modify and add to these inherited characteristics to implement their own distinct attributes and behaviors, much like a child inherits attributes from its parents.

Polymorphism is used in programming to provide a single interface to multiple behaviors. In inheritance-based polymorphism, what an object does is determined at runtime as a function of the behavior of the specific object type. For example, a function that adds two numbers could be programmed to behave differently based on whether it receives a real or complex number. Inheritance based polymorphism can result in problems when the parent class needs to be updated due to the needs of subclasses. This often occurs in development areas that are relatively new, where significant new problem details are still emerging and being defined. This problem is compounded for interdependent system modeling and analysis which involves changing relationships between data and components, and large numbers of components with similar data and behaviors that belong to different system and problem domains (8) and (18). For modeling and analysis areas that are well defined or relatively stable, the use of inheritance-based polymorphism is a reasonable option.

For example, using inheritance, an Automatic Bus Transfer (ABT) switch might be defined as a special type of distribution switch that inherits behaviors defined for distribution control and distribution switch components. Using containment, an ABT defines its internal behavior using a combination of simple behaviors for a switch (open/close device), monitoring point, controller and a bus bar.

Generic GTA components use extensible component DLLs that define behavior via three entry points (8) and (18):

- Across – calculates the nodal parameter (e.g., pressure, voltage)
- Through – calculates the flow parameter (e.g., mass flow, amperage)
- Dialog – brings up the graphical user interface dialogs that allow the user to specify physical parameters and component states

For fluid system modeling, valves use these dlls to implement the same open and close behavior that electrical switches use. To build a pressure reducing valve, a control function is added that compares the valve's downstream across value to a control set point pressure. The function then adjusts the valve's loss characteristic to simulate opening and closing the valve. A check valve is built using the same control function as the pressure reducing valve except that it closes the valve anytime the across value increases instead of decreases, which indicates reversed flow.

Chapter 4 Reduced Manned Ship Critical Infrastructure Problem Definition

Many parts of ship system design and operation have historically been dealt with separately along problem domain area specific lines. Two of the most difficult reduced manning operation things to coordinate in terms of automation and information management are damage control and casualty control. Damage control focuses on isolation and restoration related to compartmentation problems such as flooding and fire. Casualty control focuses on emergency operation and repair of systems directly used in the performance of missions, and systems used to provide auxiliary services to mission systems. For situations where service systems are involved in compartment damage, damage control focuses on shutting these systems down while casualty control focuses on keeping these same systems operational. Electrical distribution system operation is critical to both damage and casualty control because electrical distribution is run through the majority of compartments in a ship, and most systems in a ship require electrical power to operate.

Figure 3 shows a notional integrated ship system schematic that includes compartmentation, electrical distribution and generation, ships service distribution, piping systems and mission equipment loads. This model was developed for the purpose of evaluating the use of location dependencies between systems and compartmentation (19). In the model, the area defined by a compartment boundary is associated spatially with every system component that is located within the defined area. Selecting and failing a compartment boundary "component" results in every system component bounded by the compartment being set to "failed."

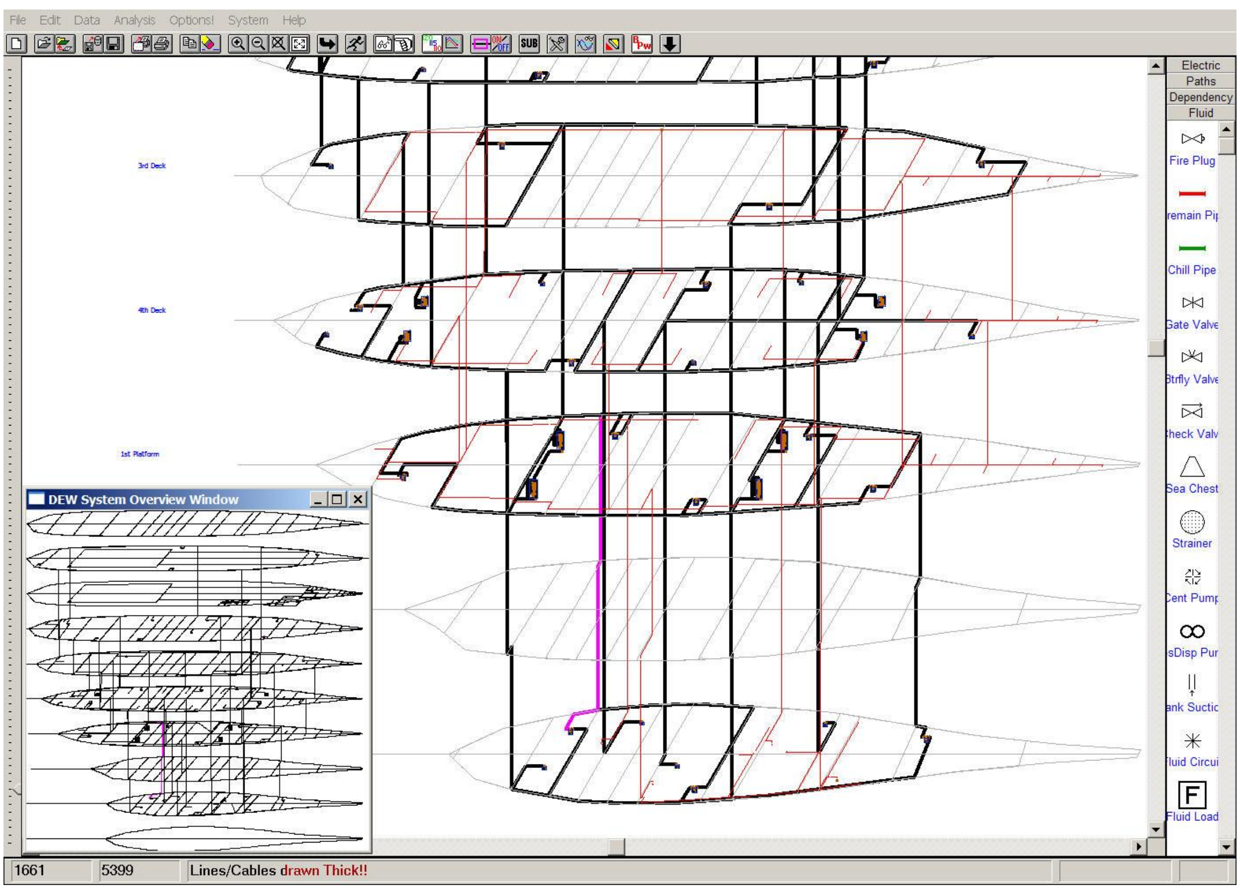

Figure 3 Integrated Ship System Models Laid Out on Line Drawings from USS Midway

Figure 4 shows how current shipboard emergency operations management and engineering analysis domains relate to the major components of a shipboard integrated power system (20). Note the organizational domain area overlaps, which in reduced manning tend to be difficult to coordinate in terms of doctrine driven operation and maintenance related data and situation awareness information management. A possible Object Oriented Analysis (the OOA part of OOA&D) type of model-based solution to HME&DC information management would be to:

- Associate mission and mission vital loads together as part of the "load" domain, and then managed during emergency conditions by Operations
- Combine chill water, electrical distribution, compartmentation, and fire main together as the "distribution" domain that is managed by Damage Control
- Combine propulsion, generation, and main machinery spaces together as the "generation" domain that is managed by Engineering

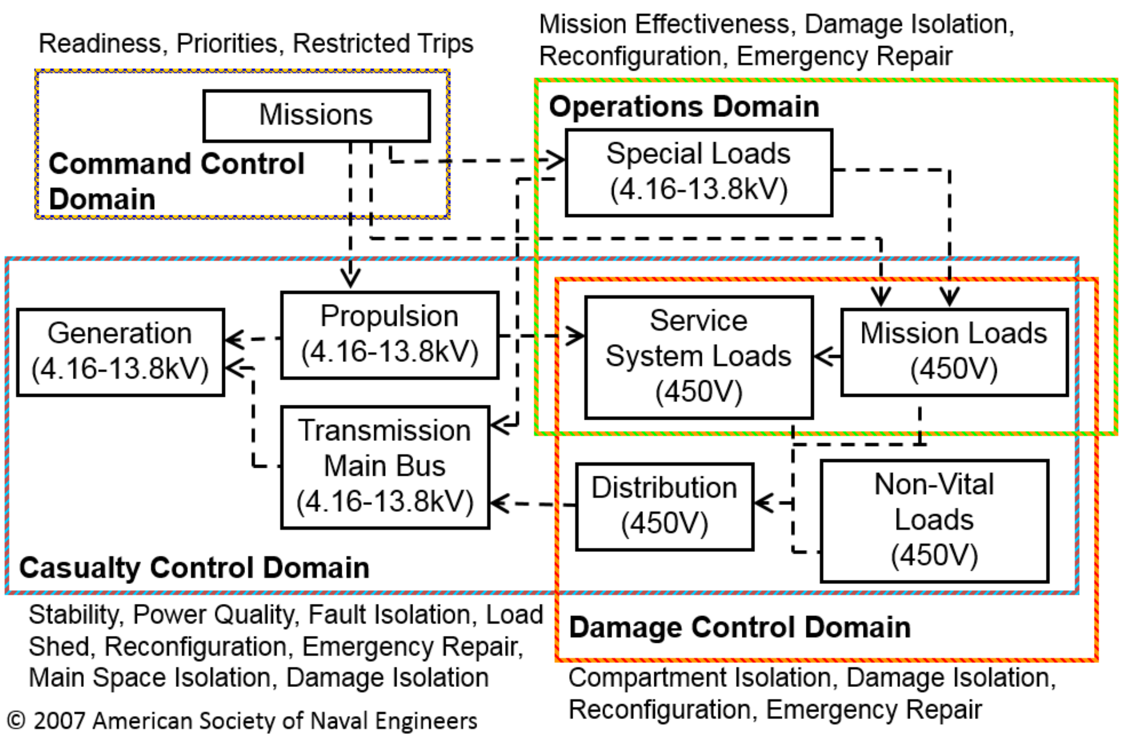

Figure 4 Shipboard Emergency Management and Engineering Analysis Domains Overlaid on Integrated Power System (IPS) Arrangement Diagram

4.1 HME&DC, Program Administration, and Operations Management Related Areas

The following sections provide an overview of the primary engineering activity and administrative program areas that affect shipboard interdependent Hull, Mechanical and Electrical and damage Control (HME&DC) system damage isolation, reconfiguration and recovery related design, automation and operation management. Reconfiguration and survivability, as it occurs in emergency situations on ship, involves a highly interrelated combination of critical factors from HME&DC system arrangement, control and monitoring, damage and casualty control doctrine, interior communications and data management, and logistics and maintenance related system status information. This is especially true for events involving multiple point casualties to more than one system or area where cascading failures from one area or system propagates to other areas and systems.

4.1.1 Engineering and Administrative Program Information Management

Putting together and maintaining accurate emergency response situation awareness information is difficult and time-consuming. This problem combined with the need to quickly stop the spread of the effects of damage after a casualty or damage event occur can lead emergency responders to make critical decisions without having complete and accurate information. This includes determining the true location and extent of damage, and the current state and capabilities of system and personnel resources that can be used to isolate damage and then recover from it.

A significant part of the information needed to maintain situation awareness is tracked and managed as part of engineering management and program administrative activities. Like engineering systems, these activities often overlap and have interdependencies with each other (21) and (22). Figure 5 shows the primary engineering operation and administration program areas that involve information that could be used to keep damage and casualty control event related situation awareness data current.

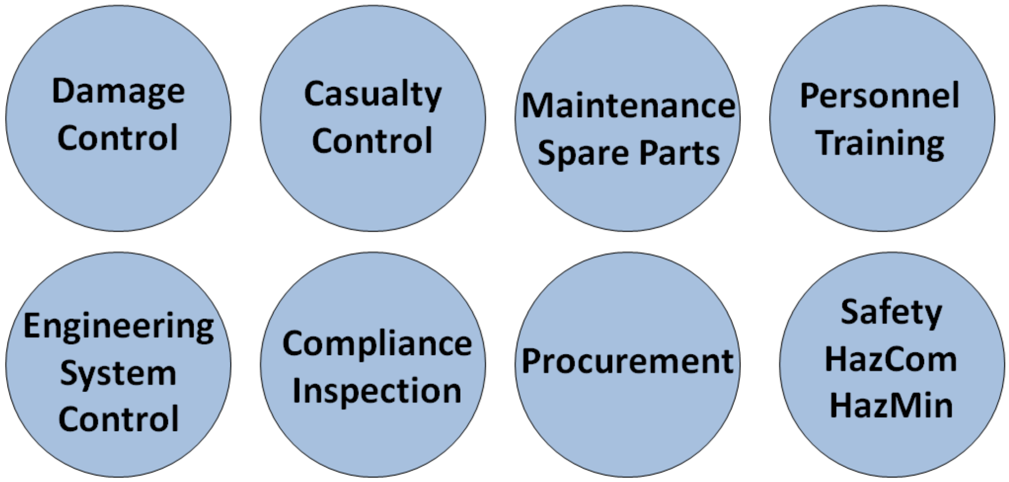

Figure 5 Ship Engineering and Administration Areas Related to Isolation and Recovery

Like integrated engineering and damage control system operation, many of the rules and requirements for managing the areas shown in Figure 5, and the software used to automate management of them are typically designed, implemented and maintained separately. These areas have many interdependencies with each other. From a reduced manned shipboard point of view, these areas look and operate more like the arrangement shown in Figure 6.

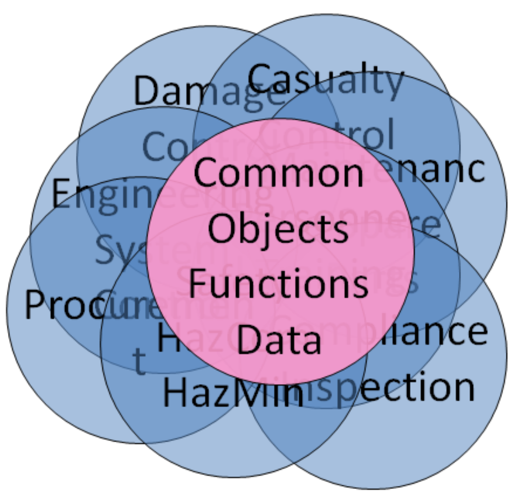

Figure 6 Ship Engineering and Administration on a Reduced Manned Ship

An OOA natural problem decomposition point of view could be used to take advantage of the common objects, functions and data commonality that becomes evident when these activities are carried out and maintained together as part of actual ship operations. The process (See Figure 7) for doing this would be to:

- Separate out the commonality related objects, functions, and data
- Redefine the commonality related parts into generic non-discipline/non-problem domain-specific objects, functions, and data that are managed together as part of a common model
- Simplify the remaining distinct parts so that they are structured to operate through attachment to the model

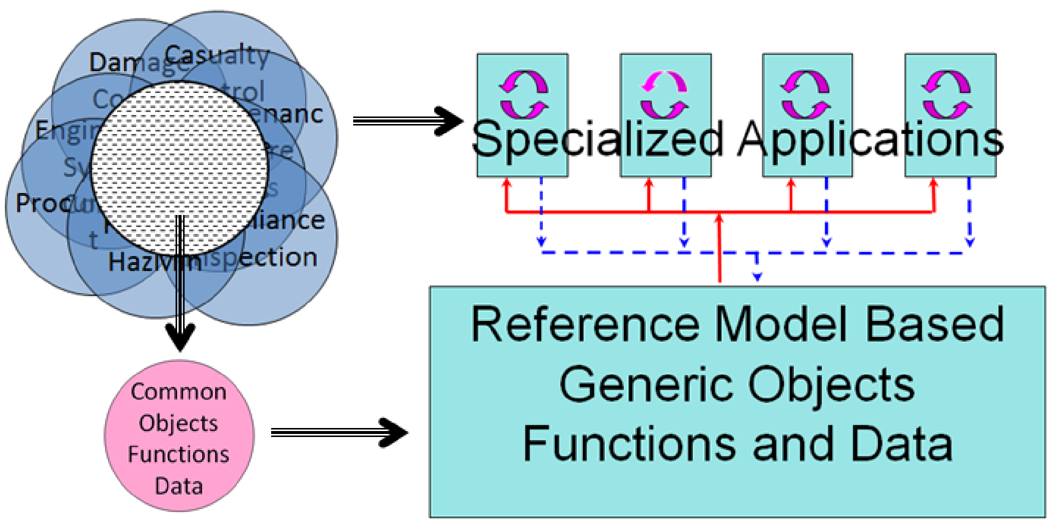

Figure 7 Commonality-Based Integration

4.1.2 Damage and Engineering Casualty Control Doctrine

The terms damage and casualty sometimes get used interchangeably and have many common factors, but when looked at in detail have significant design and operation objective differences. The primary doctrine guidance used in the Navy and Coast Guard for Damage Control is contained in Naval Ships' Technical Manual (NSTM) Chapter 079 Volume 2 Practical Damage Control and NSTM Chapter 555 Volume 1 Surface Ship Firefighting. Casualty Control doctrine is defined in NSTM Chapter 079 Volume 3 Engineering Casualty Control. Damage and casualty control have traditionally been treated as separate areas for training, shipboard organization and program oversight. For main engineering space fires and damage involving multiple areas, damage and casualty control responses need to be closely coordinated.

Damage control deals mainly with containment and mitigation of the initial effects and spread of structural, fire and flooding damage in and through tanks, voids, ventilation ducts, and compartments. Damage control systems are primarily designed to be configurable to isolate damage to compartments and damage control systems, and to maintain damage control services needed to respond to damage. Damage control operations related to engineering and weapons auxiliary system equipment, wiring, piping, and ventilation focus on securing personnel hazards, disrupting paths for propagating the spread of damage, and eliminating potential ignition and fuel sources. This translates into damage control actions that tend towards shutting down wiring, piping, ventilation, and equipment located in or passing through damaged compartments, which can lead to engineering failures in other systems if not performed correctly. Casualty control focuses on isolating equipment failures, piping ruptures and wiring faults. Once a damaged component is isolated, actions are taken to reroute and restore vital services using surviving system resources.

When looked at together, casualty control's primary goal is to isolate damaged components and keep auxiliary systems operational to maintain vital system services. Damage control's primary goal for undamaged components is to shut them down in areas affected by compartment damage. This difference can cause significant problems when damage involves multiple systems and damage to multiple compartments. For example, damage control actions to secure power to electrical cables running through a compartment could secure power to a chill water pump somewhere else resulting in a loss of cooling to a critical system.

Both manual and automated analysis and control for standard single point casualties and damage that affects only one system or compartment is relatively simple. Most training scenarios and control strategies are designed around dealing with this type of damage. The most difficult part of dealing with single point compartment damage or an equipment casualty is collecting quick and accurate equipment, compartment and personnel status information. Manual coordination for multiple point damage that affects more than one system or compartment, using a combination of predefined single point failure action lists can result in redundant and unnecessary actions. If these steps are performed incorrectly, they can lead to cascading casualties where damage and equipment failure effects propagate from the damaged areas to other systems and compartments.

4.1.3 Monitoring and Situational Awareness

During actual casualty situations, making resource trade-off decisions like: "Where to concentrate damage control efforts first," and "Which systems should be shut down and isolated, and which ones should be kept going despite the potential for propagating damage to other areas," are relatively easy decisions for experienced personnel to make if accurate damage, resource status, mission priority and mission performance time constraint information is readily available. Getting and verifying this information is usually one of the hardest parts of responding to an actual casualty. Not having accurate and timely status information is a likely cause for slow and incorrect damage response actions. A second problematic area is quickly and accurately determining how potential propagation effects from actual damage, together with figuring out how effects from crew actions and automated system responses to damage will affect other systems, compartments, and overall ship mission capability.

Getting accurate status information and fully analyzing potential propagation effects is complicated by several factors. First, both of these activities occur during the initial response phase when many things are happening and time is critical. Second, the primary sources for status information are split up across a large number of different systems, doctrine related personnel activities, and hard to coordinate paper records and grease pencil status boards. Correcting this problem by expanding monitoring to be able to collect and manage all of the information needed for automating reconfiguration may not be practical. Implementing standard merchant ship or industrial type monitoring to this level would significantly increase the number of monitoring points and measuring devices. This, in turn, would increase costs, required maintenance, and overall system complexity; which together will tend to cancel out some of the survivability improvements that total ship supervisory control is supposed to provide.

4.1.4 Restricted Maneuvering Doctrine

Merchant ship type automation is well suited and designed for a relatively narrow set of operating conditions. For naval vessels, which are typically designed to operate across a wide range of conditions and perform a wide range of missions, modifying equipment emergency shutdown and alarm functions to fit specific operation conditions can be difficult. For open ocean operations, emergency trip procedures for both merchant and naval ships follow standard casualty procedures. For hazardous operation conditions such as combat or restricted maneuvering, naval ship equipment emergency trip procedures are restricted according to ship specific risk tradeoff decisions and individual engineering plant operating characteristics. For example, during open ocean operations standard trips for casualties such as loss of lube oil pressure and crankcase explosion are actuated automatically by automation systems or as a part of preauthorized emergency procedures carried out by the engineering watch.

During hazardous operating conditions, emergency equipment shutdown is often not performed unless specifically authorized by the Conning Officer or Officer of the Deck. During open ocean steaming engineering personnel and automation systems typically shut down equipment in response to a casualty and then inform the Conning Officer. In restricted maneuvering, the engineering watch and automation systems will run equipment to failure while waiting for authorization from the Conning Officer before securing failing equipment. Carrying out restricted trip operations safely and effectively requires a thorough knowledge of the ship's systems and well-practiced emergency communication skills by both the Bridge and Engineering personnel (21).

On current automated vessels, restricting trip operations sometimes requires bypassing the automation system. This can be difficult to do because many naval ship engineering automation systems are based on merchant ship standards, which are not designed for significant changes in operating condition and equipment automated system response protocols. Operating personnel sometimes compensate for this by running more equipment than would otherwise be necessary at no or low load, or they may decide to bypass all emergency shutdowns and leave equipment unprotected. This practice reduces efficiency, increases operation cost, reduces safety and increases required maintenance.

Incorporating a general equipment arrangement reference model with online reconfiguration analysis with propulsion and engineering auxiliary PLC (Programmable Logic Controller) based automation would provide the capability to automatically adjust vital equipment shutdown and alarm settings to fit current equipment status and ship operating conditions. This type of system could be set to allow standard automated trips until a specified minimum level of capability is reached, as defined in relation to current mission priorities and system status. After that critical point is reached, shutdowns would need to be acknowledged by command and control personnel before the automation system carried them out.

This same type of reconfiguration based control logic could be used to change the way PLC control and equipment monitoring is implemented for groups of equipment. Individual equipment operation and maintenance monitoring and control could be installed by the manufacturer as an integral part of each piece of equipment. The model driven supervisory system would evaluate overall high-level system operation criteria and constraints, and then use that information to modify operation and fault response set points used by equipment level local control. If equipment arrangement is modified, equipment is damaged, or major operating requirements change, the model-based supervisory level system would automatically adjust local reactive control set points to improve coordinated response to damage according to current operating conditions. During normal operations, the supervisory system would continually adjust local equipment level control for best efficiency or some other selected mode of operation, plus keep standby emergency function protocols and system arrangements tuned for response to current most probable threats. If damage occurs and a piece of equipment's local agent loses communications with the supervisor, it would then go into a damage operation mode based on the last damage status and mission priority information it received, coordinated with what it can sense locally on its own.

4.1.5 Survivable Electrical Power

One of the most critical aspects of automating damage and casualty control on a reduced manned ship is resilient electrical power. Throughout modern naval history, major damage has often been followed by significant loss of electrical power. Part of this is due to the direct effects of damage and shock, for which the Navy has done extensive testing and has well-defined design guidance. A less well-defined part of electrical system survivability is Damage Control operability. Damage control systems such as fire main are specifically designed to be easy to operate during emergency conditions, including reconfiguration operations used to isolate damaged pipe sections and route services around damaged compartments. Electrical distribution is primarily designed to meet engineering efficiency, capacity and fault isolation criteria. New types of design and control are needed to help reduce the tradeoff differences between survivability and efficiency that exist for systems like fire main and electrical distribution. Fast and effective electrical distribution isolation to a damaged compartment is also critical to initial damage control response, and power systems need to be efficient to address rising fuel costs and more stringent environmental requirements. Developing zoned, switched loop DC or AC distribution power systems with automatic bus transfer (ABT) switches and automated fault isolation and recovery would provide virtually uninterrupted, highly survivable power that reconfigures for both damage and casualty control operations (21). Development of new model-based analysis approaches could contribute to the design and operation of these types of new system arrangement concepts that are both more efficient and more survivable.

4.1.6 Training

Reduced manned ship operation success generally depends on a few highly trained crew members with significant experience in a large number of areas. One of the key factors to producing experienced, cross-trained crew members is helping them to develop a model-based type of understanding, usually through memorization, of system layout and dependencies. Developing this type of understanding typically requires years of experience. Training that focuses on this type of understanding can be difficult and costly to do. Development of model-based system simulation capability could be used to support implementation of this type of training (21).

4.2 Casualty and Damage Control Problem Decomposition and Collaborative Integration

Figure 8 is a UML Use Case diagram that models shipboard continuity of service management for damage control and engineering casualty control operations (21). The diagram was developed to help define the major goals and interactions involved in automating continuity of service damage and casualty control reconfiguration operation management. The diagram was also used to help develop and refine the fault isolation and recovery analysis, measures, criteria and constraint concepts discussed in the following sections.

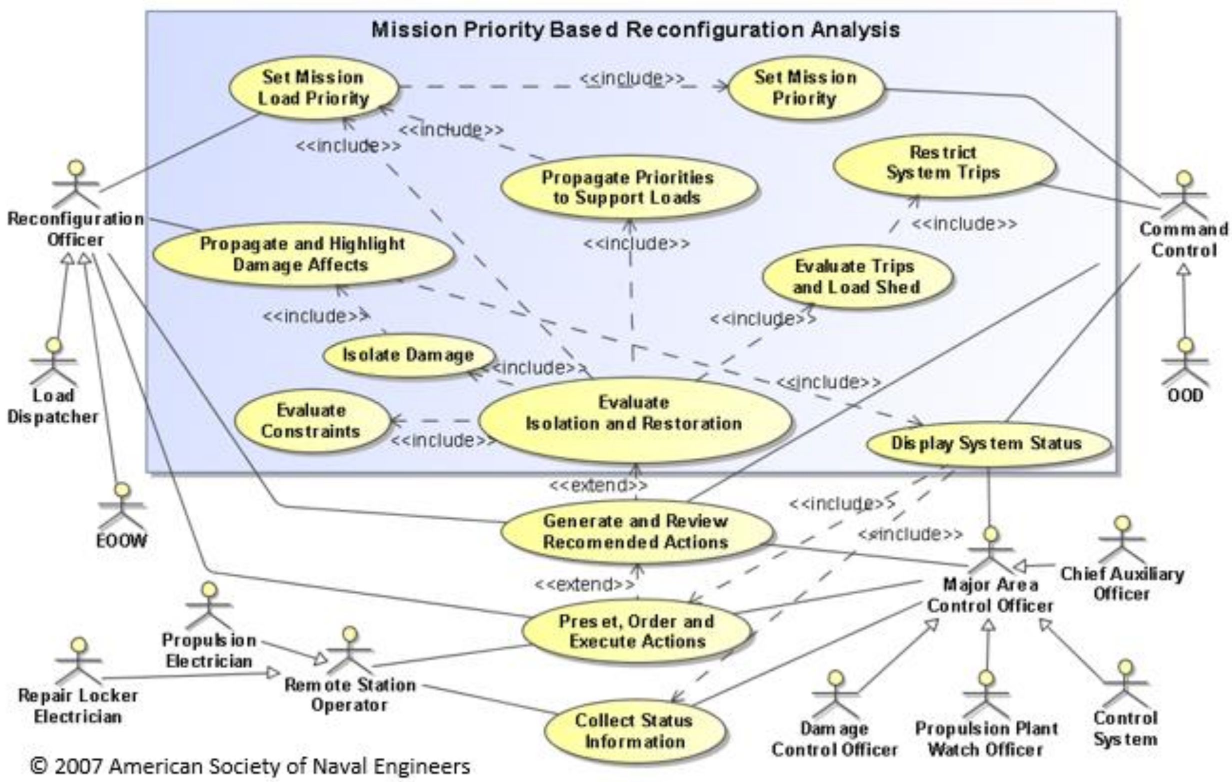

Figure 8 Continuity of Services Management Use Case Diagram

In the use case diagram shown in Figure 8 people and other systems that interact with the continuity of service management use case or "goals and objectives" which are shown as ovals, are represented as stick figure "Actors." Interactions between use case goals and actors are called “associations” and are drawn out as lines that connect actors and use case goals. “Generalizations” are modeled as lines with triangle heads. They show inheritance relationships between actors and use cases. “Includes” indicates reusable use cases that are unconditionally included as parts of other use cases. “Extends” use cases are used conditionally to extend or augment other use cases (10). The large blue box shown in Figure 8 marks what parts of continuity of service management should be combined into the system, and what parts should exist outside the system.

4.3 Priority Management, Mission Goals, Criteria, and Constraints

Shipboard system operation, resource use, and risk management are driven by a complex set of separately developed design practices, standard procedures and doctrine, and regulations. When applied together on ship during emergency operations performed by a well-trained crew, these seemingly disparate elements combine to form well-structured reconfigurable lines of responsibility and communication for both systems and personnel, which are used together to adjust quickly for the loss of vital resources and changing priorities. How personnel responsibilities and system response actions are carried out at any particular time are driven by current operating conditions, current mission capability requirements, time available for response, consequence of failure of both systems and personnel actions, and the correctness and completeness of situation awareness information.

One of the primary obstacles to automating reconfiguration analysis is the number of components, possible component state changes and complex interactions involved. Shipboard personnel simplify management of these factors by evaluating and prioritizing everything, including time and resources needed to verify status information versus the potential benefit that more reliable damage information would provide, according to its relationship with current mission capability and response time requirements. System and personnel actions that do not have a major effect on current mission requirements are given lower priorities. Actions that could be used to improve resource use efficiency or effectiveness, but cannot be completed together with other higher priority actions within mission response time constraints are also given lower priorities. If sufficient resources are available, both high and low priority actions are carried out concurrently.

Recovery management, which includes reconfiguration for damage isolation and restoration of services, can be simplified by breaking it up into phases. Shipboard recovery operations can be divided into four distinct phases, which are summarized in Table 1. The leading constraints used during each phase are allowable response time and available resources, which are used to bound potential recovery actions. Mission priorities and potential consequence for failure are used to order recovery actions, and decide how surviving resources should be allocated.

The reconfiguration analysis capability needed to automate information management and supervisory control is the same for all four phases while the primary goals that drive personnel and control system actions for each phase are different. Together, these factors can be used to simplify analysis and reduce the number of potential recovery actions that need to be evaluated (9). A detailed description of how these factors can be included as a direct part of model-based analysis is provided in the following chapter.

Table 1 Emergency Management Phase Goals, Criteria, and Constraints

Recovery Phase	Primary Goals	Survivability vs. Efficiency (Leading Criteria)	Reaction vs. Deliberation (Leading Analysis)
Readiness: Before damage	**Goal:** Maintain system capability within assigned readiness response time constraints and minimum mission capability requirements.	**Efficiency:** Optimize use of systems, personnel and maintenance resources without violating readiness response time constraints.	**Deliberative:** Use global information to pre-set component reaction protocols before damage occurs. These include protection device default response actions (run to failure, trip point settings, protection coordination settings), source status (online, standby) and load priorities.

Recovery Phase	Primary Goals	Survivability vs. Efficiency (Leading Criteria)	Reaction vs. Deliberation (Leading Analysis)
Initial Action: Immediately after damage.	**Goal:** Isolate damage. Minimize spread of damage affects.	**Survivability:** Isolate damage as quickly as possible.	**Reactive:** Protection devices, sources, and loads react to local information using reaction protocols pre-set during the readiness phase.
Remedial Action: After initial damage is isolated.	**Goal:** Optimize use of surviving resources.	**Survivability:** Make best use of surviving resources to quickly restore mission capability.	**Deliberative:** Reconfigure systems, modify source status as needed, shed load as needed and make temporary repairs. Use mission priority to manage resource use tradeoff decisions.
Restoration: After initial damage is isolated and remedial action is complete.	**Goal:** Return system to original capability.	**Efficiency:** Optimize use of system and repair resources without violating readiness response time constraints.	**Deliberative:** Make permanent repairs.

4.4 Modeling Maintenance and Emergency Repair as Components in a Network Model

Scheduling for maintenance and repair of major equipment is often defined in terms of the personnel qualifications, man-hours, and equipment needed to deal with the major sub-assemblies that the system or equipment being worked on can be divided up into. Man-hour and cost estimates for sub-assembly repair, removal and staging are generally well defined. Developing the capability to model sub-assemblies together with elements representing the problem or "event," and the maintenance or repair "activity" associated with the problem, as well as service system components involved in the repair activity would make it possible to extend reconfiguration analysis to include repair time, resource requirements, and effects on mission readiness, recoverability and reliability. Basic model component definitions for "activities" and "events" are defined below. Concepts for modeling activities and events as part of an integrated system model are shown in Figure 9.

1. **Activity**: Related to a system component using a dependency. Modifies affected component operation time. Includes activities such as Normal and Emergency Repair, Investigation, and Fire Fighting. Can be assigned a priority, or can take on the priority of the system component or components it affects.
2. **Point Event**: Related to a system component using a dependency. Modifies affected component status. Includes events such as Fault, Rupture, and Fire. Can be assigned a priority, or can take on the priority of the system component or components that it affects.
3. **Area Event**: Defined as an object line or shape. Related to a location by virtue of where the line or shape is drawn with respect to affected components. Modifies affected component status. Includes events such as Flooding and Fire. Can be assigned a priority, or can take on the priority of the system component or components that it affects.

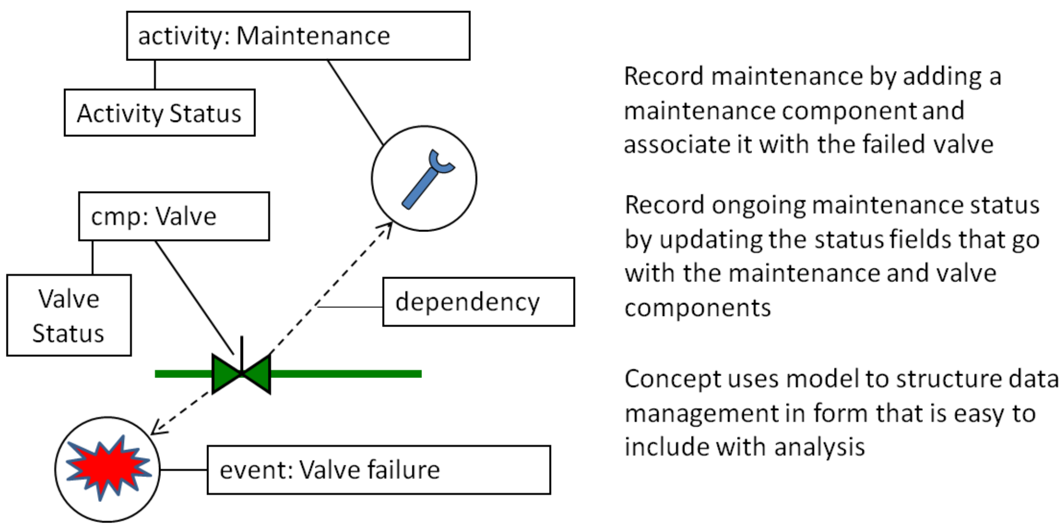

Figure 9 Model-Based Emergency Response, Maintenance, and Repair Modeling Concept

Simple status definitions that are the same or very close to the same for different types of systems and activities could be used to further simplify component status related data management and recovery analysis.

Provided below are a possible set of status definitions that were developed by reviewing the component related information used in the activity and administration program areas illustrated in Figure 5, from a multidiscipline shipboard operations point of view. The goal for this list is to develop generalized attributes, most of which are either 1 for true of 0 for false, that when combined together and associated with a component in a model can be used to record detailed status and operation history information in a standardized way, that can then be managed much like SCADA measurements and event messages. Possible generic component status attributes include:

- cmpID: String
- on: Boolean
- locked: Boolean
- failed: Boolean
- lostService: Boolean
- priority: Integer
- timeToOn: Time
- timeToOff: Time
- allowableInteruptTime:Time
- tagged: Boolean
- restrictedOperation: Boolean
- localControl: Boolean
- autoControl: Boolean
- maxThroughRating: Real
- minThroughRating: Real
- maxAccrossRating: Real
- minAccrossRating: Real

Chapter 5 Analysis Development and Results

In this chapter, the shipboard damage control and engineering casualty control design and operation management problem definition work presented in previous sections are used to define and test model-based GTA analysis concepts for performing integrated critical infrastructure system failure affect propagation, fault isolation and recovery analysis in a form that is well suited for use in both design and real-time operation management. The key elements to this development, which constitute new contributions to this area of analysis are:

- Defining convergence acceleration factors and measures for forward-backward sweep method alternating current (AC) power and fluid system flow analysis that has been extended to include loops
- Defining networked component mission priority and component operation status propagation rules that conform with forward-backward sweep analysis step procedures

- Using dependency components to interconnect systems, mission objects and service system power and fluid loads so that damage affects and mission priority information can be propagated across reconfigurable, interdependent service system networks
- Using abstract mission “loads” and dependencies to map missions to design and operations management readiness measures
- Development of interactive system traces which replace the need for the use of complex cost functions for evaluating system reconfiguration options
- Using mission priorities, system component status and operation constraints to generate discrete time interval based recovery plots

5.1 Forward-Backward Sweep GTA Loop Analysis Improvement

GTA models systems as directed graphs. Iterators are used to define automated traces which traverse the graph. Analysis and data management functions and algorithms are written using these iterators. This means that changes in components and the effects that these changes have on all other components in the system can be derived “on-the-fly.”

GTA was originally developed for use with radial alternating current (AC) power utility distribution systems. GTA power flow analysis uses a forward-backward sweep method with power defined in terms of voltage and current. Research performed by Elis (23) extended the forward-backward sweep method used in GTA to include loops. Loops are modeled as two radial paths that join at the ends using paired GTA “cotree” and “adjacent” current source components controlled together through a switch. In terms of a graph the radial paths form a tree, and the GTA cotree and GTA adjacent current source components form a cotree. For each iteration the radial tree is solved first and then the cotree is used to update current through the cotree and adjacent pair components to reduce the voltage difference between these component pairs to zero. This method has been used successfully to model power utility distribution, power utility transmission, and integrated transmission and distribution (T&D) (24). GTA has also been used to model networked water utility system flow and contaminant propagation (25) and (26).

Extending the forward-backward sweep method to include loops introduces convergence problems that were addressed by Dilek and Broadwater using a continuation method that is based on a combination of load stepping, impedance stepping and damping of cotrees with large changes in flow (27). This approach works reasonably well, but further refinement is needed to improve convergence for contingency and reconfiguration analysis problems involving cotree locations and loop paths that are difficult to solve. Review of water system flow analysis development history shows parallels with development of the loop solution approach used in GTA. To explore potential for improving GTA loop flow convergence, several concepts from past water system loop flow analysis research were modified for use with iterator based traces and then applied to the GTA loop flow method. This includes selecting loop paths which reduce impedance along mutual, overlapping loop path sections and use of convergence acceleration factors (28) and (29) to modify loop flow iteration step update values.

Potential application of these concepts to GTA was evaluated experimentally using simple systems that varied the number and location of cotree components, both with and without the use of convergence accelerator factors for both AC power and fluid flow. To help evaluate results each GTA system solution, for both fluid and AC power flow, was compared against results generated using a hybrid gradient approach developed by Todini and Pilati (29) and (30). The Todini and Pilati approach is commonly used in water utility system analysis.

In GTA, a substation component is used as a reference voltage source. During the forward sweep, the power flow algorithm starts at the reference source and traces out one component at a time. At each component traversed during the forward trace, component voltage is updated based on component impedance and current flowing through the component. At load components, current is updated using the latest voltage values and specified load voltage characteristics. A backward trace is then used to propagate component current values back to the reference source. This process is repeated until convergence tolerances are met or the solution hits a specified maximum number of allowed iterations.

Loops are modeled as radial circuits connected at a cotree component. Cotree components are modeled as two reciprocal current source components connected through a switch. Cotree and adjacent component current source injections are adjusted over each iteration step to reduce the voltage difference across the cotree ends to zero. Each time cotree flow is adjusted, another set of radial forward-backward voltage and current sweep iterations is performed. As discussed in previous sections, this process can be used with any system that can be defined in terms of components with across and through behaviors. The example problems provided in the following sections were done using AC power and fluid flow component equations.

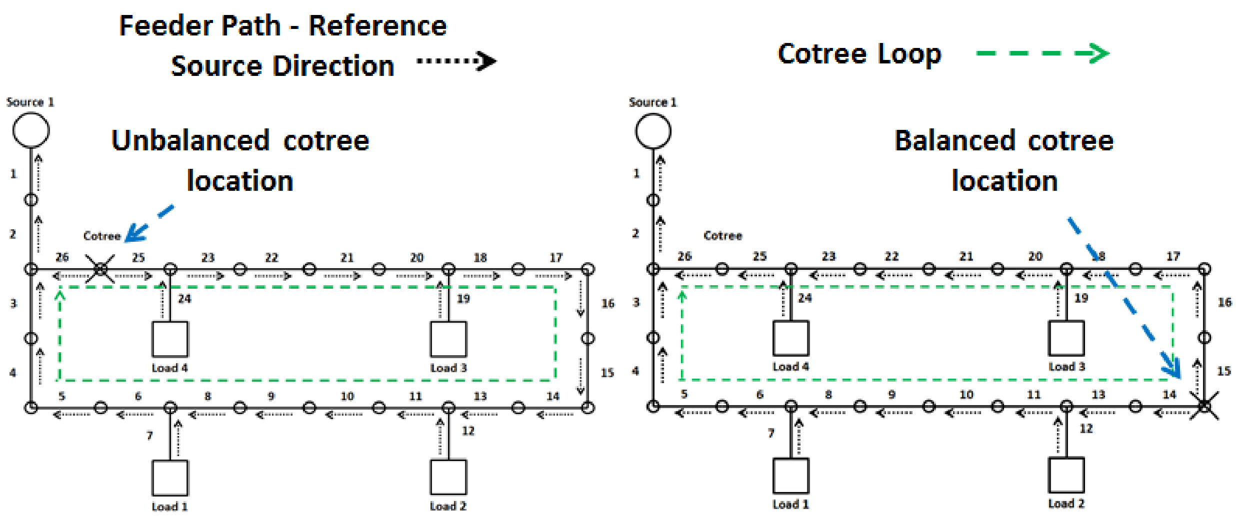

Figure 10 Balanced and Unbalanced Cotree Placement Example Circuits

An "unbalanced cotree" circuit, meaning the cotrees is placed so that there is a large initial voltage difference between the two sides of the cotree switch, is shown alongside a "balanced cotree" circuit in Figure 10. In the unbalanced cotree circuit, the initial voltage trace starts at Source 1 and then propagates to Component 1, Component 2, and Component 3 and so on to component 25. Voltage is then propagated from Component 2 directly to Component 26. During the first trace current through each component is zero, which means that every component gets set to the reference source voltage including both sides of the cotree component. Current from each load, which were modeled as P-Q loads for all AC power flow examples, is set using the voltage propagated from the first voltage trace. The backward current trace starts at Component 25, and then propagates current through each component back up to Component 3. The backward trace then shifts to stepping from Component 26 up to

Component 2. At Component 2 where branch currents from Component 26 and Component 3 meet, the currents from each branch are added. Current from Component 2 is set to Component 1 and then to Source 1. The cotree and adjacent component voltages in the "unbalanced circuit" are then set to equal the difference between the end point voltages of Components 26 and 25.

The forward-backward sweep traces for the "balanced circuit" are the same except that the voltage trace goes from the Source through components 1 through 14, and then from Component 2 to Components 26 through 15. Because the cotree in the "balanced" circuit is located at a point in the system where the loads and impedances between both sides of the cotree back to the source are relatively equal, the initial cotree voltage will be smaller than the one found in the "unbalanced" circuit if the same initial cotree flow guess is used. The circuit shown in Figure 10 is used later in this section to look at how varying cotree location, reference voltage direction, which side of the cotree is set to be plus (+) and which side is set to be negative (-), and initial cotree flow guess affects convergence.

5.1.1 GTA and Gradient Method AC Power and Fluid Flow Equation Definition

The simple four loop circuits shown in Figure 11, Figure 13, and Figure 14 were modeled in Microsoft Excel using AC power and fluid flow component and system level equations and GTA forward-backward sweep traces. The circuit models were designed to provide insight into how management of cotree flow calculations in GTA might be improved. For the power system models, source voltages were set at 39.7 kV with a voltage angle of zero. The fluid system versions of the models used fluid property, tank height and piping characteristics that are representative of those used in water distribution systems. The Five different systems were analyzed using three different cotree solution approaches:

- "Hardy-Cross" which uses partial derivatives of effort with respect to flow (which equals impedance for power systems) for each cotree and ignores mutual partial derivatives between cotree loop paths that overlap with each other

- A matrix based loop path trace solution which uses system traces for each cotree loop path to sum self and mutual, overlapping path partial derivatives of effort with respect to flow for each loop
- A matrix based sensitivity method which uses partial derivative effort with respect to flow values estimated using cotree flow injections and resulting cotree effort changes instead of system traces

The basic equation used to estimate AC current flow correction[2] at each cotree is:

$$\Delta I\ cotree = \frac{\Delta \text{V cotree loop}}{\text{Z cotree loop}} \tag{1}$$

Where each variable in equation (1) is defined as:

$\Delta V\ cotree\ loop$: Sum of the voltages of components in the cotree loop trace path

$\Delta I\ cotree$: Estimated change in current that will reduce ΔV to zero

$Z\ cotree\ loop$: Sum of the partial derivatives of voltage with respect to current for each component in the cotree loop trace path.

For fluid flow, where resistance is a function of the flow squared, the fluid system equation for cotree flow change is:

$$\Delta Q\ cotree = \frac{\Delta \text{P cotree loop}}{\text{2KQ cotree loop}} \tag{2}$$

Where each variable in equation (2) is defined as:

$\Delta P\ cotree\ loop$: Sum of the pressures of components in the cotree loop trace path

$\Delta Q\ cotree$: Estimated change in fluid that will reduce ΔP to zero

$2KQ\ cotree\ loop$: Sum of the partial derivatives of pressure with respect to flow for each component in the cotree loop trace path. The K term includes the values that are

[2] Note phasor values are used for AC power calculations

relatively independent of flow and is primarily a function of pipe length, diameter and roughness (29)

For Hardy-Cross method cotree flow change calculations, mutual loop factors are ignored.

The standard fluid flow Hardy-Cross equation (29) for the change in loop flow is:

$$\Delta Q_{lloop} = -\frac{\sum_{l \in lloop} K_l \, Q_l^n}{\sum_{l \in lloop} n K_l |Q_l|^{n-1}} \qquad (3)$$

Where: l is a pipe or line element the connects two nodes

n is the coefficient for the friction loss calculation used

$n = 1.852$ for the Hazen Williams equation

$n = 2$ for the Manning’s equation

After flow change for each loop is calculated, new loop flow values are used to update pressure values at all nodes. This process is repeated until changes in loop flow meet a specified tolerance, or a maximum iteration number is met.

For the GTA matrix based loop flow approach, forward-backward sweep iterations are used to adjust across values (voltage and pressure), and a Jacobian matrix made up of the partial derivatives of the loop self and mutual energy or head loss terms are used to adjust cotree flows. This is equivalent to the fluid flow simultaneous loop equation method developed by Epp and Fowler (29) and (31). The denominator term in the Hardy-Cross change in flow equation is the same as the one used for calculating the diagonal elements of the Jacobian Matrix.

The GTA “Sensitivity” approach calculates the elements of the Jacobian by injecting a flow change at each cotree one at a time, and then uses the resulting change in cotree voltage to calculate each element of the Jacobian Matrix. For the electrical system analysis results shown in the following sections, the matrix elements found for the Matrix and Sensitivity approaches are very close to being the same, with the exception of some differences most likely due to round off error. For the fluid system models, since the $2KQ$ elements are a function of flow the flow injections used for the Matrix approach cotree path component traces and Sensitivity approach cotree flow injections would need to be coordinated in order to be comparable. Based on this and the similarity of results between the Matrix and Sensitivity approaches seen for the electrical system examples, the Sensitivity approach was not used in the fluid system analysis examples. It should be noted that the electrical systems were modeled as single phase alternating current (AC) systems.

The Todini approach was used to generate solutions for both fluid and AC power flow. The basic Todini approach formulation for correction of pipe flow and node pressure is (32):

$$\begin{bmatrix} \boldsymbol{A}_{11} & \vdots & \boldsymbol{A}_{12} \\ \cdots & \cdots & \cdots \\ \boldsymbol{A}_{21} & \vdots & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} \\ \cdots \\ \boldsymbol{H} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{A}_{10}\boldsymbol{H}_0 \\ \cdots \\ -\boldsymbol{q} \end{bmatrix} \tag{4}$$

Where: $\boldsymbol{A}_{11}(k,k) = r_k|Q_k|^{\propto -1}$ where k is the index for a pipe connecting nodes i, j

r_k is a coefficient that depends pipe diameter, length and roughness

$\propto$ is an exponent that = 1.852 if the Hasen-Williams equation is used. It = 2 if the Darcy-Weisbach equations is used

$$\boldsymbol{A}_{12}(i,j) = \begin{cases} -1 \text{ if pipe } i \text{ leaves node } j \\ 0 \text{ if pipe } i \text{ is not connected to node } j \\ +1 \text{ if pipe } i \text{ enters node } j \end{cases}$$

$\boldsymbol{Q}^T = [Q_1, Q_2, \dots Q_p]$ for the $[1, p]$ unknown pipe discharges

$\boldsymbol{H}^T = [H_1, H_2, \dots H_n]$ for the $[1, n]$ unknown nodal heads

$\boldsymbol{H}_0^T = [H_{n+1}, H_{n+22}, \dots H_N]$ for the $[1, N-n]$ known nodal heads

$\boldsymbol{q}^T = [Q_1, Q_2, \dots Q_p]$ for the $[1, p]$ unknown pipe discharges

5.1.2 Effect of Cotree Placement on Convergence

In the example circuits shown below, the effect of cotree position on convergence was evaluated by moving Cotree 2 in System 2 and System 3 to two different positions that were away from center point of the systems. In System 4, Cotree 3 was moved off center. In System 5, Cotree 2 was moved to an off-center location that corresponded with the location of Cotree 3. The electrical systems were analyzed with both light and heavy loading. Light loading was set at 1 kW at each load. Heavy loading was set at 90 MW for each load. Examples that set two loads to zero and two loads to 180 MW each in various combinations were also used. Initial cotree flow guesses were set at zero. Loading conditions and cotree guesses were selected to provide a wide range for comparing results. Source voltages were set at 39.7 kV at an angle of zero. Loads were modeled as P-Q loads with current defined as a function of voltage. Similar loading combinations using fixed flow rate loads were used for fluid flow evaluation.

In each of the systems used for analysis, loop paths were defined by feeder paths shown as black dotted arrows in the system figures. Loop traces start at both sides of a cotree (defined as the cotree and adjacent in GTA), follow the cotree and adjacent side feeder paths back towards the source and then end at the point where the feeder paths meet. For cotree components where cotree and adjacent sides are referenced to two different sources, the loop paths go all of the way back to each reference source.

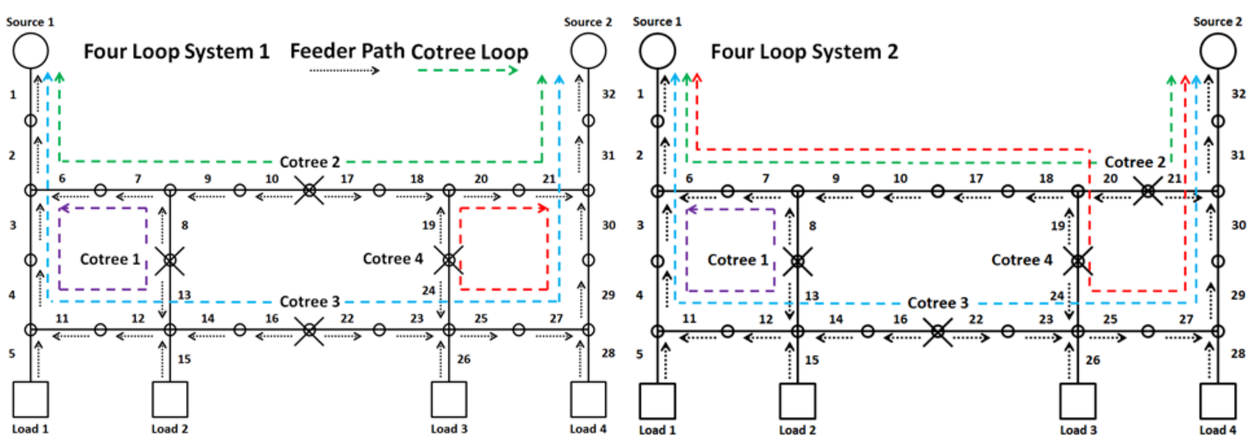

Figure 11 System 1 & 2, Four Loop – Two Source Cotree Placement Test Circuits

System 1	Cot 1	Cot 2	Cot 3	Cot 4	Ratio
Cot 1	1.2536	0.3134	0.6268	0	0.75
Cot 2	0.3134	1.8804	0.6268	0.3134	0.67
Cot 3	0.6268	0.6268	2.5072	0.6268	0.75
Cot 4	0	0.3134	0.6268	1.2536	0.75
System 2	**Cot 1**	**Cot 2**	**Cot 3**	**Cot 4**	**Ratio**
Cot 1	1.2536	0.3134	0.6268	0.3134	1.00
Cot 2	0.3134	1.8804	0.6268	1.567	1.33
Cot 3	0.6268	0.6268	2.5072	1.2536	1.00
Cot 4	0.3134	1.567	1.2536	2.5072	1.25

Figure 12 Cotree Loop Mutual Impedances Change Resulting From Cotree Move

As shown in Figure 11 and Figure 12, moving Cotree 2 increase mutual loop path impedances, shown in purple, between cotrees 1 and 4, 2 and 4, and 3 and 4. It also increases the ratio of self-impedance to mutual impedance. The self-impedance for Cotree loop 4 also increased. It can be seen from the system figures that moving a cotree that is referenced to two sources, see Figure 11, Cotree 2 in Systems 1 and 2, can significantly change feeder path traces which in turn affect cotree loop path self and mutual impedances, which are indicated by overlaps between cotree loop paths across the same components.

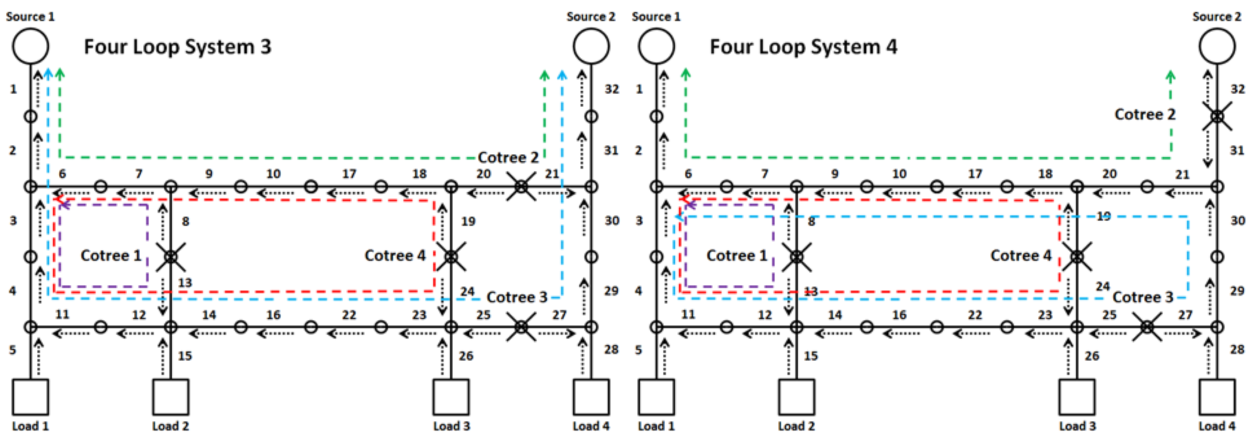

Figure 13 System 3 & 4, Four Loop – Two Source Cotree Placement Test Circuits

Cotree self and mutual impedance ratios, see Figure 13 through Figure 17, and convergence results shown in Table 4, show that in Systems 1 through 5 the systems with the higher cotree loop path impedance ratios tend to not converge as well as the ones with smaller ratios.

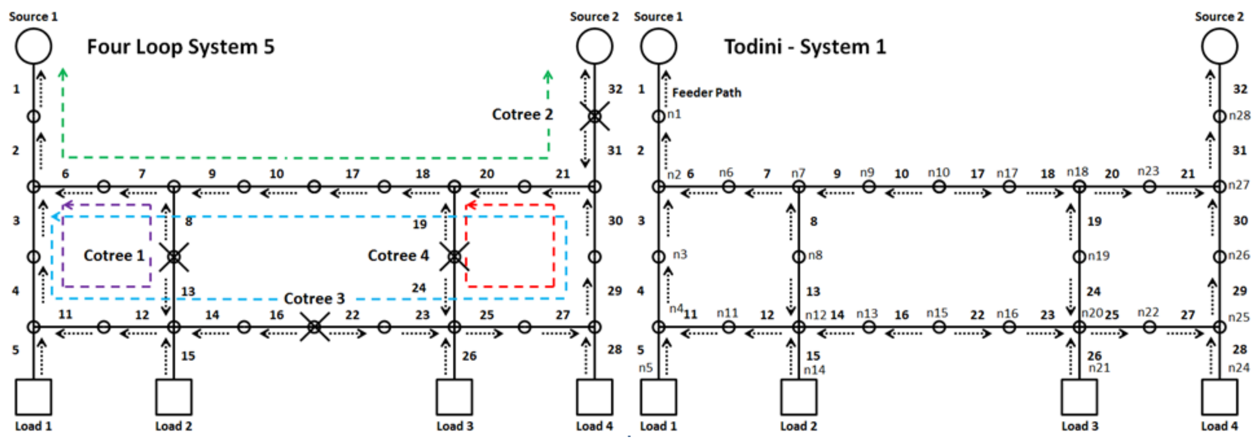

Figure 14 System 5 & 6, Four Loop – Two Source Cotree Placement Test Circuits

The effects of increasing and decreasing mutual cotree path resistance were also evaluated by adding a large impedance at line 8 were no cotree paths overlap, and line 12 where two or three cotree paths overlap, depending on the system. Adding the large impedance at line 8 had little effect on convergence. Adding the large impedance at line 12 had a significant effect on convergence.

System 1	Cot 1	Cot 2	Cot 3	Cot 4	Ratio
Cot 1	1.2536	0.3134	0.6268	0	0.75
Cot 2	0.3134	1.8804	0.6268	0.3134	0.67
Cot 3	0.6268	0.6268	2.5072	0.6268	0.75
Cot 4	0	0.3134	0.6268	1.2536	0.75

System 2	Cot 1	Cot 2	Cot 3	Cot 4	Ratio
Cot 1	1.2536	0.3134	0.6268	0.3134	1.00
Cot 2	0.3134	1.8804	0.6268	1.567	1.33
Cot 3	0.6268	0.6268	2.5072	1.2536	1.00
Cot 4	0.3134	1.567	1.2536	2.5072	1.25

Figure 15 System 1 and 2 Cotree Loop Path Ratios

This agrees with past fluid system analysis research which recommends managing loop paths so that they overlap along paths of least resistance. Experimentation with this measure appeared to show a general relationship between loop path ratio and convergence, but not a definitive one that can be used analytically.

System 3	Cot 1	Cot 2	Cot 3	Cot 4	Ratio
Cot 1	1.2536	0.3134	0.6268	0.9402	1.50
Cot 2	0.3134	1.8804	0.6268	0.9402	1.00
Cot 3	0.6268	0.6268	2.5072	1.2536	1.00
Cot 4	0.9402	0.9402	1.2536	2.5072	1.25

System 4	Cot 1	Cot 2	Cot 3	Cot 4	Ratio
Cot 1	1.2536	0.3134	0.9402	0.9402	1.75
Cot 2	0.3134	1.8804	1.2536	0.9402	1.33
Cot 3	0.9402	1.2536	3.134	2.1938	1.40
Cot 4	0.9402	0.9402	2.1938	2.5072	1.63

Figure 16 Systems 3 and 4 Cotree Loop Path Ratios

System 5	Cot 1	Cot 2	Cot 3	Cot 4	Ratio
Cot 1	1.2536	0.3134	0.9402	0	1.00
Cot 2	0.3134	1.8804	1.2536	0.3134	1.00
Cot 3	0.9402	1.2536	3.134	0.9402	1.00
Cot 4	0	0.3134	0.9402	1.2536	1.00

Figure 17 System 5 Cotree Loop Path Ratios

5.1.3 Loop Flow Acceleration Factors and Measures

The second cotree convergence technique taken from past fluid system analysis is the use of acceleration factors. Research by Williams in 1973 (28) experimented with applying an acceleration factor to loop flow ΔQ terms, which are equivalent to loop flow adjustments made in GTA at cotree components. Williams found that through experimentation on eight example systems that ranged from having 6 to 256 loops, that convergence of the Hardy-Cross loop method could be improved by using a multiplying factor which was applied when the sign of ΔQ in successive iterations did not change. If successive ΔQ's did change signs, the correction factor was removed. Williams found general improvement but did not find any analytical relationship between the correction factor and any other deterministic factors.

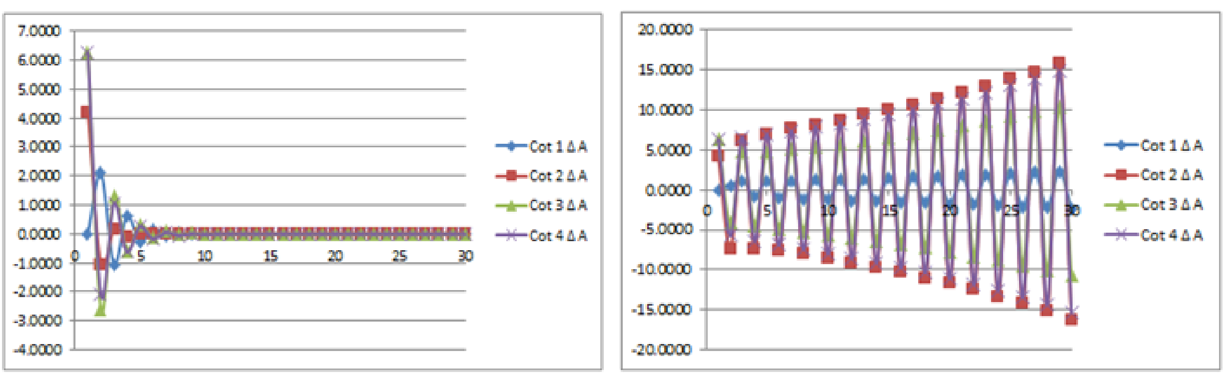

Figure 18 System 1 and 2 Cotree Amp Correction vs. Iteration Step

For the work performed for the research presented in this dissertation two experimental measures that compared estimated versus resulting change in pressure versus change in flow ,which is equivalent to loop impedance for a power system, over successive iteration steps was used. As specified earlier in equations (1) and (2), the Hardy-Cross and Jacobian Matrix elements which are used to estimate change in cotree flow injection are defined as:

$$\Delta I\ cotree = \frac{\Delta \text{V cotree loop}}{\text{Z cotree loop}} \qquad and \qquad \Delta Q\ cotree = \frac{\Delta \text{P cotree loop}}{\text{2KQ cotree loop}}$$

Figure 18 shows example cotree flow correction plots for System 1 and System 2. For the same initial conditions and without any correction factor applied, cotree flows for the system on the left, System 1, converge in a short number of iterations. Cotree flows for the system on the right, System 2, diverge.

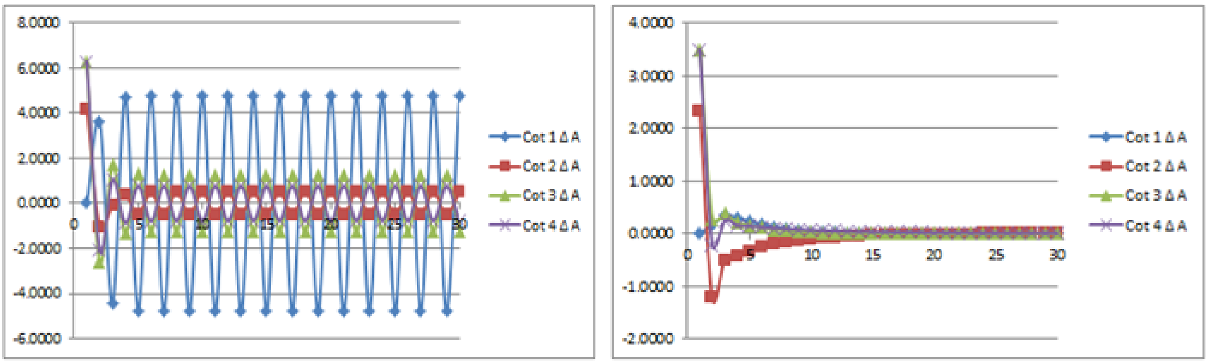

Figure 19 System 1 and 2 Cotree Amp Correction vs. Iteration Step with Damping Applied

Multiplying the diagonal element of the Jacobian matrix or Hardy-Cross element for Cotree 1 in System 1 by a factor of 0.579 causes all of the cotree amp corrections for the System 1 to oscillate. Multiplying all the cotree loop path impedances for System2 by a factor of 1.8 causes the solution for System 2 to converge. See Figure 19.

Experimentation resulted in the definition of two convergence measures. The first measure, which was used in the previous example, compares estimated cotree path impedance, or 2KQ terms for fluid systems, calculated by tracing out the loop. This value was then compared to the “resulting” impedance at the cotree which was found by taking the change in cotree voltage from the previous cotree flow injection and dividing it by the current injection change that caused it. If the ratio of estimated cotree impedance to resulting cotree impedance is greater than some specified value, 1 was used for the analysis shown, the iteration was deemed to be diverging, and the estimated impedance correction factor was increased. If the ratio of estimated impedance to resulting impedance was less than some specified value, 0.1 was used for all analysis cases shown, then the correction factor was decreased.

The second measure compared the average cotree change magnitude for all cotrees at each cotree flow iteration step. If the average cotree flow correction for all cotrees increased, then the cotree impedance factors for all cotree components were increased. Through experimentation, it was found that the two measures could be used together. The "estimated vs. resulting" impedance measure was used to make small adjustments, and the "average total cotree injection flow magnitude increase" measure was used to make large adjustments.

5.1.4 Four Loop AC Power System Convergence Results

Table 2 Defines evaluation cases used to generate convergence comparison for the four loop test systems for changes in loading, solution approach type, and results of analysis performed with and without scaling factors or "damping" applied. Results for four-loop system convergence comparison is shown in Table 3 and Table 4. To further investigate the use of cotree loop path damping and possible effects of loading on cotree convergence, the four-loop systems were modified into two single loop, single source systems and two double loop, single source systems. See Figure 24 and Figure 25. Results for double and single loop system evaluation is shown in Table 5 and Table 6.

Table 2 Four Loop System Cotree Convergence Analysis Case Definitions

Analysis		Load kW				Analysis		Load kW			
Case No.	Solution Method	Load 1	Load 2	Load 3	Load 4	Case No.	Solution Method	Load 1	Load 2	Load 3	Load 4
1	Hardy Cross	1	1	1	1	16	Sensitivity Matrix	0	180000	180000	0
2	Trace Matrix	1	1	1	1	17	Sensitivity Matrix	180000	0	180000	0
3	Sensitivity Matrix	1	1	1	1	18	Sensitivity Matrix	0	180000	0	180000
4	Hardy Cross	90000	90000	90000	90000	19	Hardy Cross	1	0	0	0
5	Trace Matrix	90000	90000	90000	90000	20	Hardy Cross	0	1	0	0
6	Sensitivity Matrix	90000	90000	90000	90000	21	Hardy Cross	0	0	1	0
7	Hardy Cross	180000	0	0	180000	22	Hardy Cross	0	0	0	1
8	Hardy Cross	0	180000	180000	0	23	Trace Matrix	1	0	0	0
9	Hardy Cross	180000	0	180000	0	24	Trace Matrix	0	1	0	0
10	Hardy Cross	0	180000	0	180000	25	Trace Matrix	0	0	1	0
11	Trace Matrix	180000	0	0	180000	26	Trace Matrix	0	0	0	1
12	Trace Matrix	0	180000	180000	0	27	Sensitivity Matrix	1	0	0	0
13	Trace Matrix	180000	0	180000	0	28	Sensitivity Matrix	0	1	0	0
14	Trace Matrix	0	180000	0	180000	29	Sensitivity Matrix	0	0	1	0
15	Sensitivity Matrix	180000	0	0	180000	30	Sensitivity Matrix	0	0	0	1

Cotree loop path convergence acceleration or damping factors applied to analysis Case 10 of Table 2 are shown in the figures provided bellow. The term "damping" factor is used because using an acceleration factor that is greater than 1 increases loop path impedance, which in turn reduces the magnitude of cotree flow change used to adjust cotree component gap voltage. Convergence plots and Damping factors for analysis Case 10 for Systems 1 and 2 are shown in Figure 20 and Figure 21.

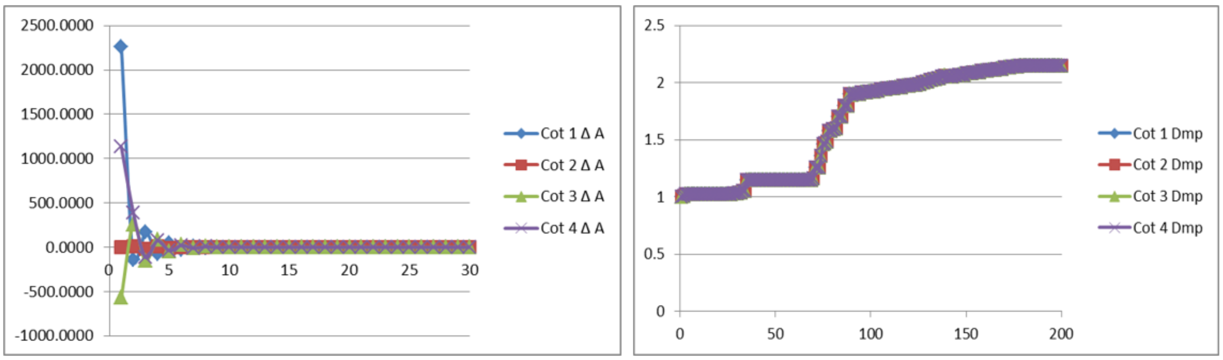

Figure 20 System 1 Cotree Loop Convergence, Case 10 with Damping On

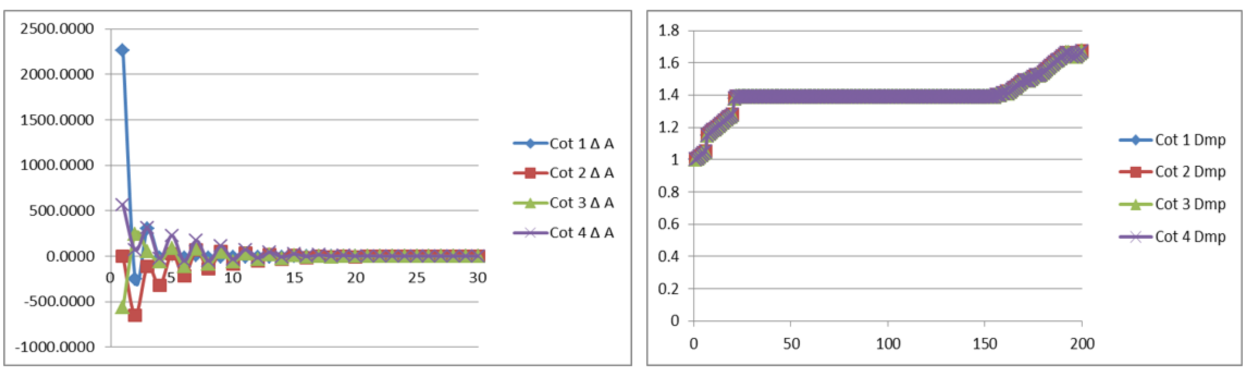

Figure 21 System 2 Cotree Loop Convergence, Case 10 with Damping On

Convergence plots and damping factors for the same analysis case for Systems 3 and 4 are shown in Figure 22 and Figure 23.

In Table 4, the average number of iterations for each system that converged, which was done without damping being used, was 59.2. The average number of times that the four systems converged within 200 iterations, which was the number of iterations enumerated in the test Excel sheet used, was 10.4. With damping, the average number of iterations needed for convergence was 44.9. The average number of times that each system converged within 200 iterations with damping being used was 30. This appears to be a significant improvement for both number of iterations required to converge and number of times that solutions converged within 200 iterations.

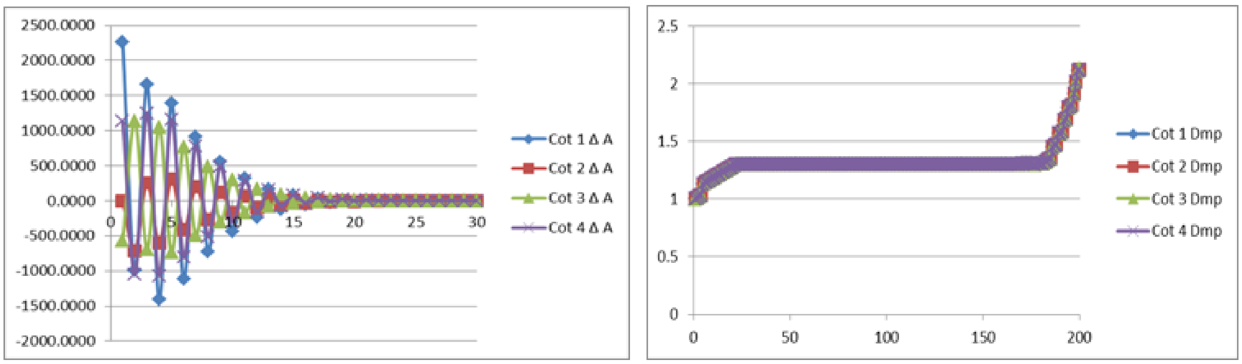

Figure 22 System 3 Cotree Loop Convergence, Case 10 with Damping On

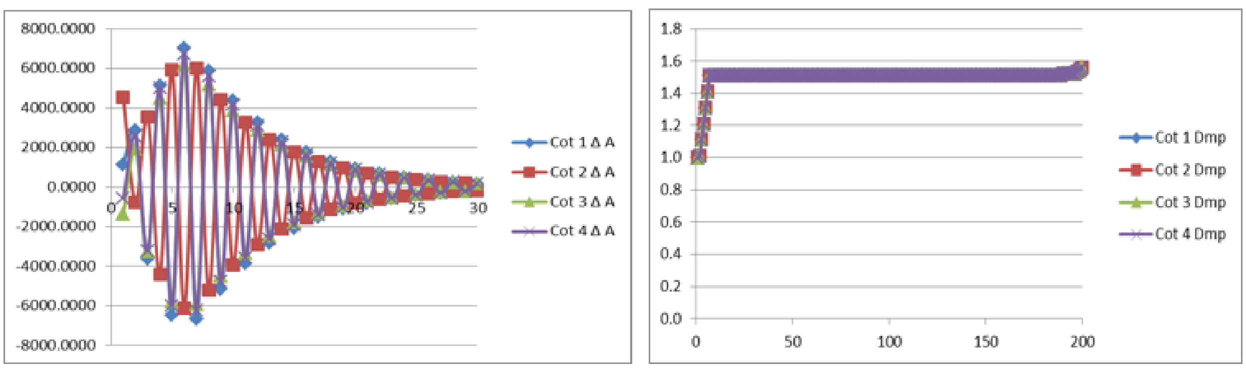

Figure 23 System 4 Cotree Loop Convergence, Case 10 with Damping On

Note that in the convergence numbers for the Todini Gradient method, which was used as a convergence check; results are slightly different for the with and without damping runs for some cases, even though damping was not applied to the Todini solution. This is likely due to the fact that initial node voltage and line current flow guesses used for the Todini solution were taken from the first iteration result generated using GTA.

Table 3 Comparison of Four Cotree System Convergence by Solution Approach

	All Three Approaches		Hardy-Cross Approach		Matrix Approach		Sensitivity Approach	
	Damping Off	Damping On	Damping Off	Damping On	Damping Off	Damping On	Damping Off	Damping On
Average No. of Iterations	59.2	44.9	35.7	43.5	131.1	44.7	124.0	46.6
Average No. Converged	10.4	30.0	5.8	10.0	2.6	10.0	2.0	10.0

A review of results broken down by solution approach for all five system cotree configurations and loading condition cases (30 in total, 10 cases for each cotree flow correction approach used) shows that the use of damping improved the number of cases that converged for all three methods. Note that the colors shown in Table 4 denote solution method (White – Hardy-Cross, Red – Trace Matrix, Green – Sensitivity Matrix), using the color scheme defined in Table 2. Comparison of Table 4 results by solution type is shown in Table 3.

Results show that the Hardy-Cross approach generally performed better than the other two approaches both with and without the use of damping. The poorer performance shown by the Matrix and Sensitivity approaches was not expected because it was assumed that including mutual loop impedances would add additional information to the solution. This was true for both the damped and undamped fluid and electrical system analysis cases. It should be noted that if a combined GTA based Hardy-Cross solution with damping proves to be useful, it will eliminate the need to use matrices to solve loop flows.   This calls for additional research because one of the common reasons given for fluid system analysis moving from the Hardy-Cross method to the Todini gradient method as the preferred standard approach is that the gradient method performed better with large systems. It would be interesting to see if this behavior holds true for implementation of Hardy-Cross and cotree damping using GTA.

Table 4 Four Loop System Cotree Convergence Analysis Results - Damping On and Off

Analysis Case	Number of Iterations to Converge Cotree Loop Damping Off						Number of Iterations to Converge Cotree Loop Damping On (Snych Mode)					
	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Todini	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Todini
1	9	No	21	No	18	2	9	21	21	28	18	2
2	No	No	71	No	108	2	18	18	24	28	30	2
3	No	No	71	No	No	2	18	18	24	28	30	2
4	11	No	79	No	59	13	10	71	83	115	67	13
5	No	No	179	No	No	13	43	39	95	59	113	13
6	No	No	179	No	No	13	43	39	95	59	113	13
7	10	No	76	No	60	13	9	68	77	92	66	13
8	12	No	104	No	62	15	10	73	84	129	70	15
9	26	No	77	No	59	14	24	72	83	90	67	14
10	26	No	79	No	61	14	24	69	80	108	69	14
11	No	No	173	No	200	13	40	37	87	64	112	13
12	No	No	180	No	No	15	45	43	99	59	115	15
13	No	No	173	No	No	14	41	38	96	58	11	14
14	No	No	176	No	No	14	44	41	91	64	112	14
15	No	No	173	No	No	13	40	37	87	64	112	13
16	No	No	180	No	No	15	45	42	99	59	115	15
17	No	No	177	No	No	14	41	38	96	58	111	14
18	No	No	176	No	No	14	44	41	91	64	112	13
19	9	No	19	No	18	2	9	21	19	28	19	2
20	9	No	20	No	18	2	9	21	20	28	16	2
21	9	No	20	No	18	2	9	21	20	27	16	2
22	9	No	No	No	18	2	9	21	18	23	16	2
23	No	No	71	No	108	2	18	18	24	28	29	2
24	No	No	71	No	No	2	18	18	23	28	29	2
25	No	No	71	No	No	2	18	18	24	28	30	2
26	No	No	71	No	No	2	18	18	24	28	30	2
27	No	No	71	No	No	2	18	18	24	28	29	2
28	No	No	71	No	No	2	18	18	23	28	29	2
29	No	No	71	No	No	2	18	18	24	28	30	2
30	No	No	71	No	No	2	18	18	24	28	30	2
Avgerage	13.0	N/A	102.4	N/A	62.1	7.9	24.3	34.4	56.0	51.9	58.2	7.9
Total Converged	10	0	29	0	13	30	30	30	30	30	30	30

5.1.5 One and Two Loop AC Power System Convergence Results

For the single loop systems, increasing loading and unbalancing loading between the four loads affected both systems in terms of the number of iterations required for convergence relatively the same. Results for the "balanced cotree" system, System 2, showed better results than the "unbalanced cotree" system. For fluid flow analysis shown in the following section, initial cotree flow guess had a more significant effect. The two systems were then modified into two loop systems, and results were again compared for analysis performed both with and without the use of damping.

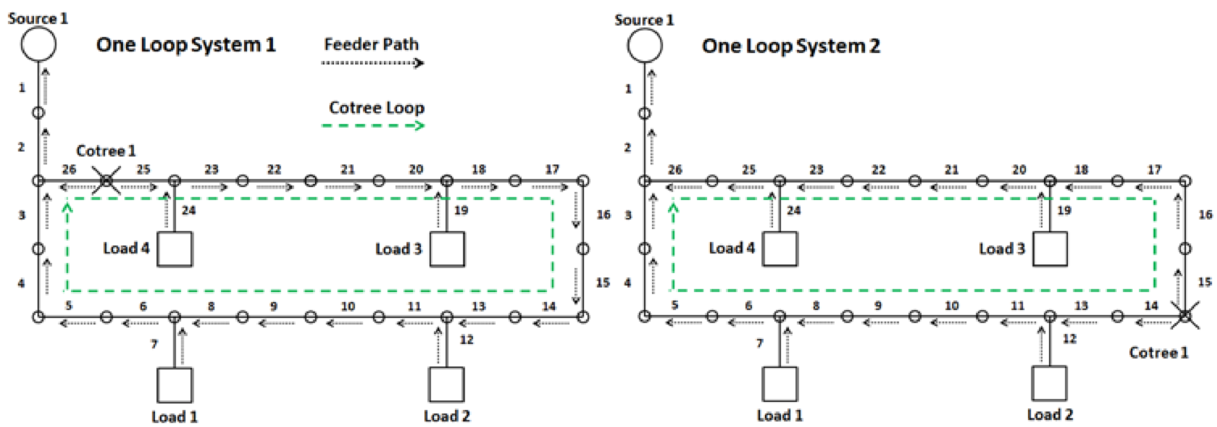

Figure 24 One Loop System 1 and System 2

Table 5 Single Source, Single Loop System 1 and System 2 Convergence with no Damping

Analysis Case No.	Solution Method	Load kW				Iteration No.	
		Load 1	Load 2	Load 3	Load 4	Sys 1	Sys 2
1	Hardy Cross	1	1	1	1	2	1
2	Hardy Cross	1000	1000	1000	1000	3	2
3	Hardy Cross	10000	10000	10000	10000	4	3
4	Hardy Cross	100000	100000	100000	100000	6	4
5	Hardy Cross	400000	0	0	0	6	4
6	Hardy Cross	0	400000	0	0	6	5
7	Hardy Cross	0	0	400000	0	7	5
8	Hardy Cross	0	0	0	400000	6	4
Average number of iterations						5	3.5

For the two loop system analysis cases, improvement for analysis performed with and without damping, and "balanced" versus "unbalanced" cotree placement was mixed. The best result was for the balanced system with cotree damping. See Table 6. Cotree loop damping appeared to extend the range of convergence as load was added to both systems. System 3 is a modified version of System 1 with the sign convention for Cotree 1 reversed. Changing direction of the initial cotree flow guess did not affect results.

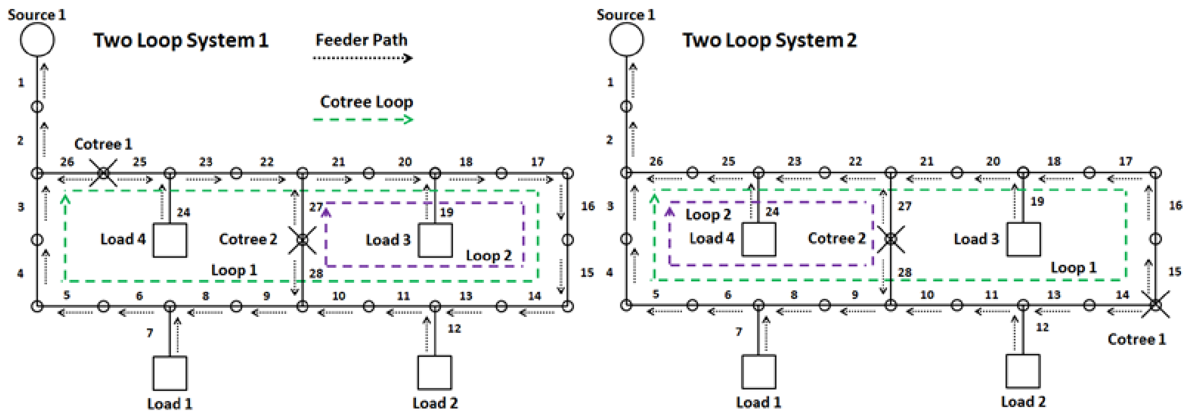

Figure 25 Two Loop System 1 and System 2

Table 6 Two Loop System Comparison of Convergence with Damping and Varied Loading

Analysis Case No.	Two Cotree System	Number Iterations With Cotree Damping Off			Number Iterations With Cotree Damping On		
	Load kW	Sys 1	Sys 2	Sys 3	Sys 1	Sys 2	Sys 3
1	1	11	6	11	9	6	9
2	100	21	16	21	19	14	19
3	1000	27	22	27	25	18	25
4	10000	36	28	36	32	25	32
5	15000	40	30	40	33	26	33
6	20000	45	31	45	34	27	34
7	25000	50	33	50	34	27	34
8	60000	No	43	No	36	30	36
9	100000	No	71	No	41	33	41
10	110000	No	90	No	47	34	47
11	133000	No	No	No	93	40	70
	Average Iterations	32.9	37.0	32.9	36.6	25.5	34.5
	Number. Converged	7	10	7	11	11	11

5.1.6 Fluid System Convergence Comparison

Fluid system results in respect to the effects of cotree placement and damping were similar to the ones found for AC electrical system analysis where cotree damping significantly improved convergence, with the exception that for fluid flow not all cases converged when cotree damping was applied. A possible reason for this difference is that fluid resistance is a function of fluid flow, which makes analysis more sensitive to initial cotree flow guesses. A comparison of results with varying initial cotree flow guesses for the two and four loop systems is shown in Table 8 and Table 9. Table 7 shows the analysis case loading used for the results shown in Table 9. As mentioned earlier, the Sensitivity Matrix approach was not used for the fluid systems because of the difficulty of coordinating cotree flow guess values between the matrix and sensitivity approaches, and the relatively small differences between these two approaches found in the electrical system examples.

Table 7 Fluid System Four Loop Method and Fluid Load Cases

Analysis Case No.	Solution Method	Load (Gallons Per Minute - GPM)				Analysis Case No.	Solution Method	Load (Gallons Per Minute - GPM)			
		Load 1	Load 2	Load 3	Load 4			Load 1	Load 2	Load 3	Load 4
1	Hardy Cross	1	1	1	1	16	Hardy Cross	0	0	0	1
2	Trace Matrix	1	1	1	1	17	Trace Matrix	1	0	0	0
3	Hardy Cross	1000	1000	1000	1000	18	Trace Matrix	0	1	0	0
4	Trace Matrix	1000	1000	1000	1000	19	Trace Matrix	0	0	1	0
5	Hardy Cross	2000	0	0	2000	20	Trace Matrix	0	0	0	1
6	Hardy Cross	0	2000	2000	0	21					
7	Hardy Cross	2000	0	2000	0	22					
8	Hardy Cross	0	2000	0	2000	23					
9	Trace Matrix	2000	0	0	2000	24					
10	Trace Matrix	0	2000	2000	0	25					
11	Trace Matrix	2000	0	2000	0	26					
12	Trace Matrix	0	2000	0	2000	27					
13	Hardy Cross	1	0	0	0	28					
14	Hardy Cross	0	1	0	0	29					
15	Hardy Cross	0	0	1	0	30					

In both fluid flow two and four-loop system analysis cases, varying cotree flow initial guesses had a noticeable effect on convergence. The three initial flow guess choices used were 10 Gallons Per Minute (GPM), 500 GPM and a flow guesses based on the average GPM specified for all four loads. Use of the average of the fluid load attached at loads one through four appeared to perform the best for all cases in terms of the average number of iterations and the total number of cases that converged. A comprehensive study of initial guesses on the convergence behavior for different fluid flow analysis algorithms can be found in the work by Todini and Rossman (32).

Table 8 Convergence Comparison Varying Cotree Flow Guess using Hardy-Cross

Analysis Case	Number of Iterations to Converge With Cotree Damping - Cotree Flow Initial Guess Set to Average Load						Number of Iterations to Converge With Cotree Damping Cotree Flow Initial Guess Set to 500 GPM						Number of Iterations to Converge With Cotree Damping Cotree Initial Guess Set to 10 GPM					
	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Todini	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Todini	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Todini
1	17	50	149	121	25	4	32	132	No	No	39	13	17	101	No	No	27	8
3	93	45	150	147	73	5	83	53	145	134	61	4	43	144	145	138	79	4
5	104	117	No	No	89	5	105	60	No	157	77	4	52	104	154	142	94	5
6	89	1	139	129	74	6	84	127	159	150	61	5	44	170	No	No	71	4
7	76	46	147	135	72	6	64	118	145	141	50	5	51	170	No	No	65	5
8	91	53	148	148	51	5	88	52	153	137	61	5	58	113	126	121	72	5
13	14	No	130	124	37	4	24	No	No	No	35	15	13	No	No	No	20	10
14	8	No	118	113	10	4	22	No	No	No	32	15	11	No	No	No	17	9
15	8	No	61	60	No	2	22	149	No	No	36	15	14	No	No	No	21	9
16	15	31	136	133	50	3	24	No	No	No	36	14	15	No	No	No	21	8
Avgerage	**51.5**	**49.0**	**130.9**	**123.3**	**53.4**	**4.4**	**54.8**	**98.7**	**150.5**	**143.8**	**48.8**	**9.5**	**31.8**	**133.7**	**141.7**	**133.7**	**48.7**	**6.7**
Total Converged	**11**	**8**	**10**	**10**	**10**	**11**	**11**	**8**	**5**	**6**	**11**	**11**	**11**	**7**	**4**	**4**	**11**	**11**

Average No. Iterations 81.6
Average No. Converged 9.8

Average No. Iterations 90.3
Average No. Converged 8.2

Average No. Iterations 92.9
Average No. Converged 7.4

Table 9 Four Loop Systems Cotree Convergence Comparison Results

Analysis Case	Number of Iterations to Converge With Cotree Loop Damping Off						Number of Iterations to Converge With Cotree Loop Damping On					
	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Todini	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Todini
1	17	No	No	No	31	7	17	101	No	No	27	8
2	16	No	No	No	No	7	16	93	118	115	41	8
3	41	No	No	No	67	4	43	144	145	138	79	4
4	76	No	No	No	No	4	70	123	136	137	124	4
5	51	No	No	No	84	5	52	104	154	142	94	5
6	42	No	No	No	69	4	44	170	No	No	71	4
7	52	No	No	No	57	5	51	170	No	No	65	5
8	59	No	No	No	61	5	58	113	126	121	72	5
9	111	No	No	No	No	5	101	145	157	153	102	5
10	89	No	No	No	No	4	109	157	No	197	125	4
11	91	No	No	No	No	5	80	153	197	197	122	5
12	70	No	No	No	No	5	73	139	135	133	No	5
13	13	No	No	No	21	9	13	No	No	No	20	10
14	11	No	No	No	19	9	11	No	No	No	17	9
15	14	No	No	No	49	9	14	No	No	No	21	9
16	17	No	No	No	50	9	15	No	No	No	21	8
17	44	No	No	No	No	10	63	91	122	119	30	9
18	82	No	No	No	No	9	No	91	120	117	29	9
19	18	No	No	No	No	9	17	No	No	No	40	9
20	14	No	No	No	No	9	14	No	No	No	41	8
Avgerage	**42.2**	**N/A**	**N/A**	**N/A**	**42.3**	**6.0**	**41.0**	**112.1**	**117.5**	**120.7**	**54.3**	**6.0**
Total Converged	**22**	**1**	**1**	**1**	**12**	**22**	**21**	**16**	**12**	**13**	**21**	**22**

Table 10 Four Loop Systems Convergence Comparison Varying Initial Cotree Flow Guess using Cotree Loop Trace Matrix Approach

Analysis Case	Number of Iterations to Converge With Cotree Damping - Cotree Flow Initial Guess Set to Average Load						Number of Iterations to Converge With Cotree Damping - Cotree Flow Initial Guess Set to 500 GPM						Number of Iterations to Converge With Cotree Damping - Cotree Initial Guess Set to 10 GPM					
	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Todini	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Todini	Sys 1	Sys 2	Sys 3	Sys 4	Sys 5	Todini
2	19	19	112	109	33	4	28	131	150	147	63	4	16	93	118	115	41	8
4	58	36	155	152	128	5	59	106	146	145	128	4	70	123	136	137	124	4
9	92	56	161	156	147	5	83	44	163	159	150	4	101	145	157	153	102	5
10	78	1	141	141	127	6	73	128	141	142	126	5	109	157	No	197	125	4
11	75	77	145	144	143	6	97	113	143	144	123	5	80	153	197	197	122	5
12	60	52	153	148	135	5	56	71	154	149	113	5	73	139	135	133	No	5
17	27	169	115	110	19	4	76	132	152	150	55	15	63	91	122	119	30	9
18	16	157	110	106	17	4	24	132	150	148	54	15	No	91	120	117	29	9
19	5	85	91	87	29	2	23	129	148	146	62	15	17	No	No	No	40	9
20	6	24	104	101	No	3	21	129	150	147	63	14	14	No	No	No	41	8
Avgerage	**43.6**	**67.6**	**128.7**	**125.4**	**86.4**	**4.4**	**54.0**	**111.5**	**149.7**	**147.7**	**93.7**	**8.6**	**60.3**	**124.0**	**140.7**	**146.0**	**72.7**	**6.6**
Total Converged	**10**	**10**	**10**	**10**	**9**	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**9**	**8**	**7**	**8**	**9**	**10**
	Average No. Iterations	90.3					Average No. Iterations	102.2					Average No. Iterations	98.0				
	Average No. Converged	9.8					Average No. Converged	10.0					Average No. Converged	8.4				

5.2 Power, Discrete Event and Mission Priority Information Propagation across Interdependent Systems

This section defines generalized system trace logic and interconnection point information propagation conventions which can be used to together with a heuristic algorithm to automate fault isolation and restoration analysis for multi-domain, interdependent, reconfigurable systems. The concept descriptions and examples provided in this subchapter define how logical system models made up of abstract dependency components, mission objects, and critical loads can be integrated together with detailed physical system models to perform analysis.

The majority of existing optimization-based approaches used to evaluate system reconfiguration do not address load priority. Those that do, use load priority data that is generated before analysis is run. For analysis involving large systems and large numbers of possible damage scenarios, the amount of load priority and status data required is significant for both design and operations analysis, and there is no standardized approach available that is well suited for generating and maintaining this type of data.

In reference (33), Srivastava and Butler-Purry present a heuristic shipboard electrical distribution reconfiguration approach that uses predefined load priorities, location categories, and total power rating to define switching step order for restoration and load shedding actions. In references (2) and (3), Cramer, Chan, Sudhoff, Lee and Zivi present a multi-domain ship system design analysis approach that uses linear programming to simulate electrical system operation, which is coordinated with domain specific simulation of other support systems such as cooling and sea water. The approach uses weighting factors that are defined prior to analysis to address relative importance between individual loads with respect to each other and the operational mission objectives they support. These weights are used to structure electrical system load sharing and reconfiguration for a particular event using a standard simplex linear programming approach defined by Chong and Zak in reference (34). They are assigned to power cables and energy storage to maximize power delivered to prioritized loads and encourage energy storage charging when possible. One of the main findings of this work was that to properly model load sharing, all possible generator and zonal distribution load sharing combinations for each damage case had to be enumerated and then evaluated individually using optimization-based power flow. Based on power flow results, the best option is selected for use in further analysis. Coordinated results from electrical system analysis and the other systems being modeled are used together with load priority weights to calculate "operability," which is a metric defined by the authors in reference (3) to represent total ship integrated system capability for satisfying specified operation requirements for an individual damage scenario. Operation objective "Commanded" states are assigned to each load, which are also determined a priori, and then used together with load priority related weights using the following equation:

$$O(\theta) = \frac{\int_{t_0}^{t_f} \sum_{i=1}^{I} \omega_i(t,\theta) o_i^*(t) o_i(t)}{\int_{t_0}^{t_f} \sum_{i=1}^{I} \omega_i(t,\theta) o_i^*(t)} \tag{5}$$

Where: $\omega_i(t,\theta)$ is the relative weight for an individual i_{th} load in respect to other loads

$o_i^*(t)$ is commanded operational status for an individual i_{th} load

$o_i(t)$ is the operational status for an individual i_{th} load

t_0 and t_f are the start and end times for the damage event scenario

Operability values for each defined event are combined using a "dependability" metric with minimum and average values that are used to quantify overall system damage response related performance. Reference (3) states that somewhere between $10^3 - 10^6$ time-based system simulations are needed to calculate dependability metrics for a particular design.

As stated above, load priority weights and relationships to mission objectives must be defined for each individual load before analysis is run. The level of detail to which load priority related data is defined has a significant effect on results. The level of manual and computational effort needed to generate this type of data using existing approaches for use in real-time operations analysis would be prohibitive.

Another option for relating loads to the mission objectives is to arrange loads together into sets, which can then be assigned a mission priority and status as a group. This concept lines up with how standard ship design load studies are performed, and is essentially what the automated priority, loss of service and feeder path GTA traces presented in this subchapter subsection do as a consequence of being run together on a model. For the mission priorities and loads shown in Figure 27, the mission critical loads sets (CLS) are:

{CLS: Mission 1} = [Load 2]

{CLS: Mission 2} = [Load 2, Load 3, Motor]

{CLS: Mission 3} = [Load 2, Load 3, Motor]

Using notation similar to what is defined for equation (5) for commanded state and actual state, a cost function could be defined as:

$$C = W_{Mission\ 1} * (CLS_1\ Commanded\ State - CLS_1\ Actual\ State) +_{\dots} \qquad (6)$$

Where $CLS_1\ Commanded\ State$ is set equal to one to denote that performing Mission 1 is a current objective, and $CLS_1\ Actual\ State$ is set equal to 1 to denote that all loads in the critical mission set have service. If one or more loads in a critical mission set loses service, then $CLS_1\ Actual\ State$ would be set to zero. Organizing critical loads into sets makes it possible to represent that loads defined within a set do not have value in respect to supporting a specified mission unless all loads within the set have service. Organizing vital loads into sets that are defined according to their relationship to defined missions is also important for evaluating restoration times and scheduling best use of available resources.

For the simple example system shown in Figure 27, defining load data at this level of detail for a basic set of damage scenarios results in defining 256 load priority sets. The defined damage scenarios for this set include single point failures to cables supplying vital loads and partial damage to Generator 2. The load priority sets include 64 sets for normal operation without failures and 192 sets for single point cable failures. An excerpt of the 265 damage state specific load priority sets is shown in Table 11.

Table 11 Excerpt of Enumerated Load Priority Data for Example Problem

Priroty	Mission Assignment			Normal - No Failures			Cable 2 Failed			Cable 4 Failed			Cable 5 Failed		
Case No.	Mission 1	Mission 2	Mission 3	Load 2	Load 4	Motor	Load 2	Load 4	Motor	Load 2	Load 4	Motor	Load 2	Load 4	Motor
31	3	1	0	1	0	1	1	0	1	1	0	1	3	0	0
32	3	0	1	3	1	1	0	1	1	3	0	0	0	0	0
33	0	1	3	1	3	1	0	3	3	1	0	1	0	0	0
34	0	3	1	3	1	1	0	1	1	3	0	3	0	0	0
35	3	2	0	2	0	2	2	0	2	2	0	2	3	0	0
36	3	0	2	3	2	2	0	2	2	3	0	0	0	0	0
37	2	3	0	2	0	3	2	0	3	2	0	3	2	0	0
38	2	0	3	2	3	3	2	0	0	2	0	0	2	0	0
39	0	3	2	3	2	2	0	2	2	3	0	3	0	0	0
40	0	2	3	2	3	2	0	3	3	2	0	2	0	0	0
41	1	2	3	1	3	2	1	0	2	1	0	2	1	0	0
42	1	3	2	1	2	2	1	0	3	1	0	3	1	0	0
43	2	1	3	1	3	1	1	0	1	1	0	1	2	0	0
44	2	3	1	2	1	1	0	1	1	2	0	3	0	0	0

If multiple, simultaneous failures are to be considered, the number of possible priority sets required for enumerating damage events for real systems becomes too numerous to manage manually.

Other cost functions for things such as losses and number of switch operations used in standard optimization analysis can be used without significant modification for use in interdependent system analysis. Standard constraints to define system operation limits for things like maximum current and minimum voltage, topology constraints to ensure that usable loads are not isolated and systems are configured correctly, and flow analysis constraints to ensure that conservation of energy laws such as Kirchhoff's Voltage and Current laws (KVL and KCL) are satisfied, can also be applied. A comparison of the major steps involved in performing optimization and GTA-based reconfiguration analysis is shown in Table 12.

Table 12 Comparison of Optimization and GTA-Based Reconfiguration Analysis Steps

Analysis Type and Major Steps	Optimization-Based Design Analysis	GTA-Based Operations Analysis
Objective	Find **optimal** system configuration for maximizing power to prioritized loads.	Find **viable** system configuration to isolate damage and restore services to prioritized missions.
Define load priority and damage scenario data	Define damage event scenario by assigning load priorities for all loads and status information for components directly affected by initial damage.	Define damage event scenario by assigning load priority for mission objects and status information for components directly affected by initial damage.
Damage effect analysis	Propagate damage effects: Use coordinated system simulations to propagate damage effects.	Isolate damage: Configure system to isolate damaged components using a rule-based algorithm.
Damage effect analysis output	Record resulting load states at each damage event scenario time step for use in calculating system performance in terms of defined load priorities at each time step. Use data to calculate damage response metrics.	Record switchable device operations and resulting mission object states. Output switchable device operation list for use in automated and manual damage isolation operations. Use data to calculate damage response metrics.
System restoration analysis	Restore services: Use optimization-based analysis to simulate restoration actions to maximize load delivered to prioritized loads at each damage event scenario time step.	Restore service: Run rule-based restoration algorithm that uses feeder path, loss of service traces and priority traces on integrated system model to define viable restoration paths for maintaining service to prioritized mission objects.
Restoration analysis output	Record resulting load states at each damage event scenario time step for use in calculating system performance in terms of defined load priorities at each time step.	Record switchable device operations and resulting mission object states. Generate switchable device operation list for use in automated and manual restoration

Analysis Type and Major Steps	Optimization-Based Design Analysis	GTA-Based Operations Analysis
	Use data to calculate damage response metrics.	operations. Use data to calculate damage response metrics.

The GTA modeling and system trace analysis presented in this section provides the capability to automatically derive and assign load priorities and status as a direct part of simulation and analysis. Using these concepts, defining a damage scenario is reduced to specifying a list of damaged components and defining high-level mission objectives by setting simple priorities on a relatively small number of abstract mission object "components." The abstract mission objects are used to represent high-level mission capability definitions which are commonly used in standard ship design and ship operation readiness reporting.

Subchapter section 5.2 presents these new concepts in successive levels of detail. Subsection 5.2.1 uses a simplified interdependent system model that includes prioritized missions, mission objects, an automated bus tie switch, and a cooling pump that is power by an electric motor. Subsection 5.2.2 and 5.2.3 discuss the use of dependency components to structure steady-state power flow, loss of service and mission priority information for forward and backward iterative analysis across interconnection points defined between different types of systems. Subsection 5.2.4 discusses the use of dependency components and abstract mission objects to model relationships between system operation objectives or goals and physical system components together, as part of the same "Total System" model. Subsection 5.2.5 presents the use of loss of service and priority information propagation together with abstract mission object components to structure automated generation of readiness measures and recovery plots.

5.2.1 Simplified Interdependent System Model Example

The example presented in this subsection steps through a simplified interdependent system reconfiguration for fault isolation and restoration problem. Load level priority and status information is derived at each major step of the analysis, directly from the model using automated system traces. Derived load level information is then used by a heuristic reconfiguration algorithm that uses traces to walk through the system and perform fault isolation, load restoration, and load shedding operations configured to maintain critical services to prioritized mission critical loads. The resulting switch operation steps constitute a viable damage isolation and service restoration scheme suitable for use in operations.

The basic concepts for using dependency components to relate missions, mission critical loads, and reconfigurable physical system model components is shown in Figure 26. Mission objects are used to represent performance objectives or capability goals. Dependency components, modeled using dotted line arrows, point to components or mission objects that represent services required to support the performance of missions. If surviving resources after damage isolation are not sufficient for supporting all missions, mission priority information propagated down to lower level component objects are used by the reconfiguration algorithm to perform failure isolation and restoration analysis that restores services to critical loads in mission priority order.

Contributions illustrated in this example are:

1. Use of abstract mission objects, dependency components and GTA priority and status propagation conventions to automatically assign load priorities and load status throughout the physical system model.
2. Use of a dependency component to define service connectivity and priority propagation across a fluid pump – electric motor system interconnection point.
3. Modeling missions and mission critical systems as a direct part of an integrated electrical power and fluid system model, all of which can be analyzed together as a generic, homogeneous system.
4. Performing mission priority based reconfiguration for fault isolation and restoration analysis across multi-domain interdependent physical and logic based systems using common algorithms.
5. Development of interactive system traces which replace the need for the use of complex cost functions for evaluating system reconfiguration options.

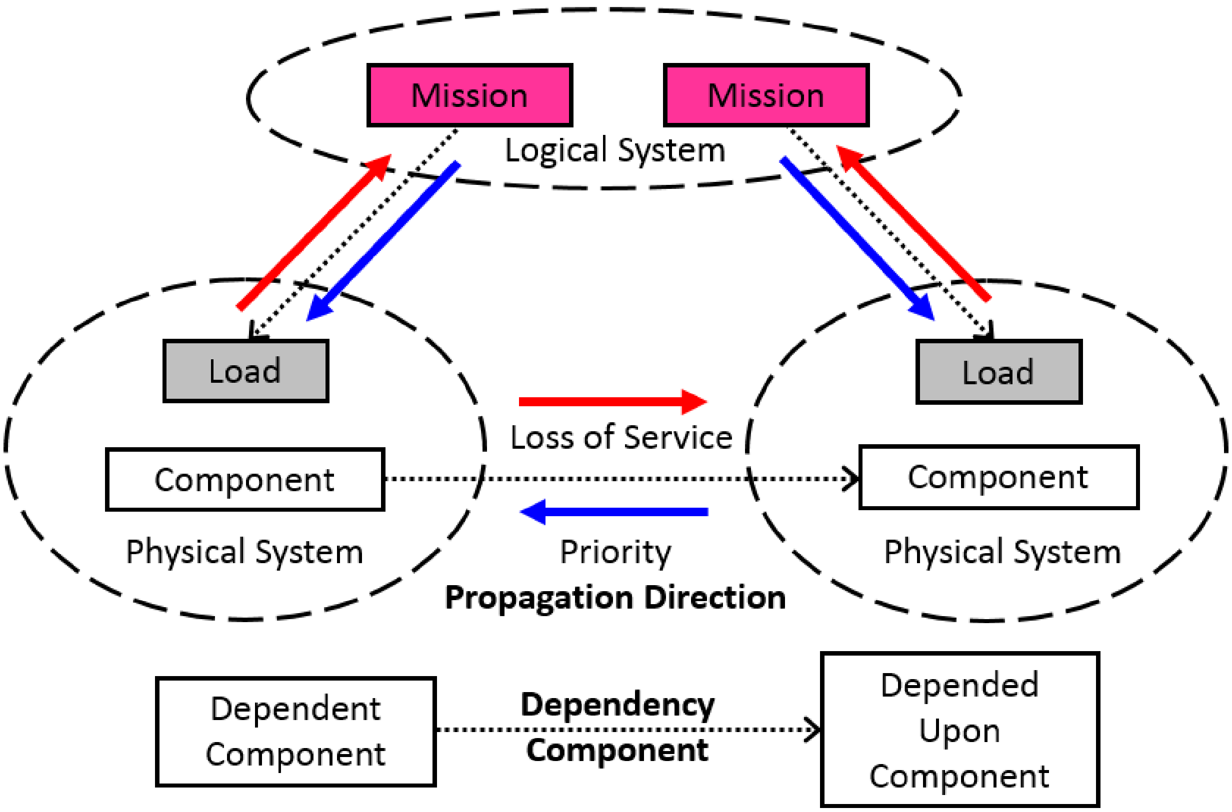

Figure 26 Dependency Component Information Propagation between Systems

5.2.1.1 Example Problem Initial Conditions

The example starts with the system in a normal operating state with mission priorities set as shown at mission objects (Mission 1, Mission 2 and Mission 3). Prior to the start of the example, system feeder paths, forward and backward iteration paths for performing flow analysis, and dependency and priority assignment propagation traces are performed.

5.2.1.1.1 Feeder Path Trace Definition

Feeder paths are defined automatically using a system trace that starts at a defined source and then propagates along each branch to its end component. This trace is the basic structure used by GTA to translate system component elements into a directed graph.

The feeder path trace (FPT) for Load 2 running back to its source, Generator 1, using standard GTA notation is:

$$FPT_{Load2} = \{ABT, Cable\ 2, CB\ 2, Bus\ 1, GB\ 1, Generator\ 1\} \quad (7)$$

The feeder path trace for Load 3 is:

$$FPT_{Load3} = \{Pipe\ 2, Valve\ 1, Cooling\ Pump, Pipe\ 1, Cooling\ Water\ Source\} \quad (8)$$

It should be noted that feeder paths do not cross over dependencies. Feeder paths are defined only within individual systems types. Dependencies are used to exchange information between components in different systems, using defined propagation rules which are uniform for all physical and logical systems that can be modeled using GTA. This means that components in one system do not need to directly "know about" or monitor components in another system. An exception would be a control device that operates using remote information taken from a component in another system. This concept can be used with any system or process that can be modeled as a network of objects with definable across and through behaviors.

Additional traces include Forward and Backwards Traces (which define the order with which components are stored in the linked list used to define a circuit), Brother Trace (which is used to manage branching relationships for summing flow from different branches and identifying jumps in topology), and Adjacent Trace (which is used to manage loops) are used to structure analysis (8) and (17). Primary traces used in the simple example are defined as follows:

FT_p = set of components in forward trace from component p (9)

BT_p = set of components in backward trace from component p (10)

FPT_p = set of components in feeder path trace from component p. (11)

BRT_p = set of components in brother trace from component p (12)

For the example shown in Figure 27, the main traces defined at the beginning of analysis (before Cable 2 is failed) are as follows[3]:

[3] For the example, traces are defined in terms of component names used in the system diagram instead of unique component identifier numbers.

$FT_{Generator1}$ = {GB 1, Bus 1, CB 1, Cable 1, Load 1, CB 2, Cable 2, Load 2, CB 3 Cable 5, Motor} (13)

$FT_{Generator2}$ = {GB 2, Bus 2, CB 4, Cable 3, CB 5, Cable 4, Load 4} (14)

$FT_{CoolingWaterSource}$ = {Pipe 1, Cooling Pump, Valve 1, Pipe 2, Load 3} (15)

BRT_{Bus1} = {CB 2, CB 3}, BRT_{CB2} = {Bus 1, CB 3} and $BRT_{CB\ 3}$ = {Bus 1, CB 2} (16)

BRT_{Bus2} = {CB 5} and BRT_{CB5} = {Bus 2} (17)

BT_{Motor} = {Cable 5, CB 3, Load 2, Cable 2, CB 2, Load 1, Cable 1, CB 1, Bus 1, GB 1, Generator 1} (18)

BT_{Load2} = {Cable 2, CB 2, Load 1, Cable 1, CB 1, Bus 1, GB 1, Generator 1} (19)

BT_{Load1} = {Cable 1, CB 1, Bus 1, GB 1, Generator 1} (20)

BT_{Load3} = {Pipe 2, Valve 1, Cooling Pump, Cooling Water Source} (21)

BT_{Load4} = {Cable 4, CB 5, Cable 3, CB 4, Bus 2, GB 2, Generator 2} (22)

FPT_{Load2} = {Load 2, Cable 3, CB 4, Bus 2, GB 2, Generator 2} (23)

FPT_{Load4} = {Load 4, Cable 4, CB 5, Bus 2, GB 2, Generator 2} (24)

FPT_{Motor} = {Cable 5, CB 3, Bus 1, GB 1, Generator 1} (25)

These traces are defined when the system file is initially "Opened" and then loaded into memory, prior to any analysis being run, which reduces the time required to perform analysis. During analysis, traces are automatically updated for affected components whenever a component is added, removed, reconnected or some other type of topology change such as a switch operation or failure occurs. Components which are not directly affected by a topology change are not updated.

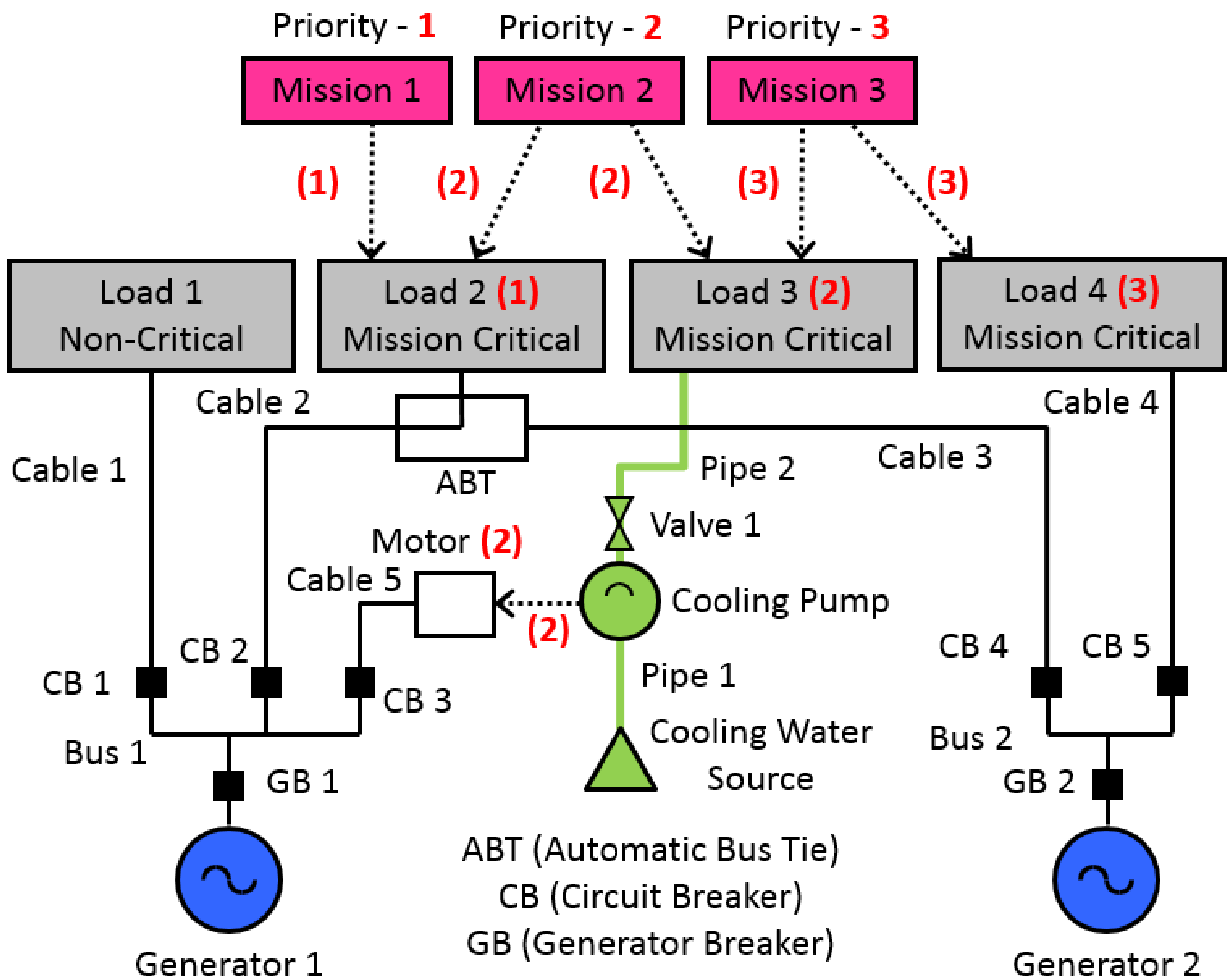

Figure 27 Simplified Interdependent Mission, Load, Electrical, and Fluid System Model

5.2.1.1.2 Priority and Loss of Service Propagation

Feeder path traces are also used to structure loss of service propagation, fault isolation, service restoration and mission priority propagation. For the priority levels used in the example, Level 1 is the highest priority. Level 0, which is not shown in the diagram, is used at components that do not have an assigned priority. In the example, mission priority propagation results for components that are assigned priorities are shown in red. Priorities are propagated from loads towards sources. The sources modeled in the example are Generators 1 and 2, and the Cooling Water Source, which feeds the Cooling Pump.

As GTA reconfiguration is run, load priority and loss of service information is propagated throughout the model at each analysis step. This information is stored as attribute data with its respective component. Automated traces are used to derive and assign these and other attributes using input data that has been set at mission objects and failed components. Automated GTA traces work by structuring components into ordered sets, which are manipulated using GTA operators. For reconfiguration analysis each component has 16 defined elements[4]:

$$C = \{p, type, c, f, f_{req}, ft, fpt, bt, brt, adjt, AD, OD, pri, status, status_{dep}, operable\} \quad (26)$$

where:
p = unique component identifier,
type = LOAD, SOURCE, SWITCH, OTHER,
c = capacity, or rating of a component,
f = flow, where $f \leq c$,
f_{req} = required flow if type == LOAD,
ft, fpt, bt, brt, adjt = components related to C via forward, feeder path, backward, brother, and adjacent trace, respectively, where a value of 0 implies the component does not exist,
AD= set of 'AND' dependency components,
OD= set of 'OR' dependency components,
pri = component priority,
status = status of component-ON, OFF, FAILED,
$status_{dep}$ = status of component's dependencies,
operable = whether the component can be turned on or off – YES, NO.

5.2.1.1.2.1 Automated Load Priority Assignment

Load priorities are assigned by first assigning priorities to loads that directly support missions, using the associations defined by dependency components. The feeder paths for each direct supporting load are then used to find loads which provide support service through interconnection points between systems. For the example problem, the electrical motor, which provides torque to the cooling pump, provides support to the pump and indirectly provides support to Load 3. Direct support loads are stored in each component's AD and OD element sets as defined in equation (26). In addition to being used to define dependencies with other

[4] Reconfiguration standard component element, operator and trace definitions are as provided in references (12) and (19) .

components, the AD and OD sets are also used to manage dependee component operation status. An example of a dependee component is the cooling pump used in the simple example problem. Dependee components always occur at the tail of the "dashed dependency arrow." For a dependee component to have a status of "ON," all of the components in the AD set must be "ON", and at least one of the components in the OD set need to be "ON."

For the example problem, the AD and OD sets for each dependee component, which is defined through attachment to dependency component, are:

$$AD_{Mission\ 1} = \{Load\ 2\},\ OD_{Mission\ 1} = \{\ \} \quad (27)$$

$$AD_{Mission\ 2} = \{Load\ 2,\ Load\ 3\},\ OD_{Mission\ 2} = \{\ \} \quad (28)$$

$$AD_{Mission\ 3} = \{Load\ 3,\ Load\ 4\},\ OD_{Mission\ 3} = \{\ \} \quad (29)$$

$$AD_{CoolongPump} = \{Motor\},\ OD_{Mission\ 3} = \{\ \} \quad (30)$$

Critical loads sets, which are used to assign priorities, are made up of the union of the direct AD and OD load sets for each mission, and indirect support loads which are identified by the reconfiguration algorithm using feeder path traces. Note that AD and OD sets can contain other types of components besides loads.

The first step to defining critical load sets is searching through the AD and OD sets for each component in the system and finding loads which directly provide support, as follows:

$$\text{Direct\_Support\_Loads}_p = S \rightarrow sum((AD \rightarrow select(type == LOAD),$$
$$OD \rightarrow select(type == LOAD)) \quad (31)$$

For the example, only loads are used as support components, and there are no OD components. As a result, the Direct_Support_Load sets defined using operation (31) equal the AD sets defined in sets (27) through (30).

The next step is to use the feeder path traces for each load contained in each Direct_Support_Load set to find loads that indirectly provide support, and then combine those loads with the Direct_Support_Loads defined using GTA operation (31), as follows:

$$Critical\_Loads_{p} = union(FTP_{Direct\_Support\_Loadsp} \rightarrow select(Direct\_Support\_Loads \rightarrow type == LOAD), Direct\_Support\_Loads_{p}) \quad (32)$$

The resulting critical load sets for the example problem, before Cable 2 is failed are

$$Critical\_Loads_{Mission1} = \{Load\ 2\} \quad (33)$$

$$Critical\_Loads_{Mission1} = \{Load\ 2,\ Load\ 3,\ Motor\} \quad (34)$$

$$Critical\_Loads_{Mission1} = \{Load\ 3,\ Load\ 4,\ Motor\} \quad (35)$$

These loads are then used to directly assign load priorities that were set by the user at mission objects, to the mission objects respective critical loads. Whenever topology changes or load priorities at mission objects are changed, priorities are automatically reset on affected components.

5.2.1.1.2.2 Failing and Isolating Components

When reconfiguration runs, the algorithm searches for components whose operation status has been set to "failed" and then uses a trace to find the closest operable sectionalizing device (e.g., a switch or a valve) that can be used to isolate it. For the example, the identified isolation switch is CB 2. When Cable 2 is isolated by the reconfiguration algorithm opening CB 2, the Automatic Bus Tie (ABT) reacts by switching its source to Cable 3, which adds load to Generator 2.

Reconfiguration runs the following trace on the system (S) to find the failed component:

$$S \rightarrow select(p \rightarrow status == FAILED) = \{Cable\ 2\} \quad (36)$$

After a failed component is found, reconfiguration uses the following trace to find the closest operable switch which can be used to isolate Cable 2:

$FPT_{Cable\ 2}$ → select(p → type == SWITCH and c → operable == YES)

= {CB 2} (37)

After CB 2 is opened, the components in the circuit which no longer have a connected path to a source are set to "OFF" to represent that they have lost service:

FT_{CB2} → collect(state|if p – 1 → state == "OFF", p → state == "OFF") (38)

Final priority propagation values and component states for analysis steps illustrated in Figure 27 are shown in Table 13. Equipment rating, and load values were selected for example purposes. For flow analysis, the electrical system was modeled as a three phase, delta system with balanced loads and normal operation voltage levels. Fluid system calculations included the use of fluid characteristics, elevation and piping characteristics for determining friction losses.

Table 13 Simplified Interdependent System Example Initial Component States

Component	Status	Through Result	Across Result	Priority	Load/Capacity
Mission 1	Has service			1	
Mission 2	Has service			2	
Mission 3	Has service			3	
Load 1	Has service	128.3 Amps	440.1 Volts	0	100 kVA, 0.9 pf
Load 2	Has service	128.3 Amps	442.6 Volts	1	100 kVA, 0.9 pf
Load 3	Has service	81.3 psi	100 GPM	2	100 GPM
Load 4	Has service	128.3 Amps	440.1 Volts	3	100 kVA, 0.9 pf
Motor	On	64.2 Amps	449.8 Volts	2	50 kVA, 0.9 pf
Cable 1	Has service	128.3 Amps	440.1 Volts	0	231.0 Amps max
Cable 2	Has service	128.3 Amps	442.6 Volts	1	231.0 Amps max
Cable 3	Has service	0 Amps	450.0 Volts	0	231.0 Amps max
Cable 4	Has service	128.3 Amps	440.1 Volts	3	231.0 Amps max
Cable 5	Has service	64.2 Amps	449.8 Volts	2	231.0 Amps max
Pump	On	100.0 psi	100 GPM	2	150 GPM/150 psi
Pipe 1	Has service	2.3 psi	100 GPM	2	250 psi max
Pipe 2	Has service	81.3 psi	100 GPM	2	250 psi max
Valve	Open (on)	99.8 psi	100 GPM	2	150 psi max
CB 1	Closed (on)	128.3 Amps	450.0 Volts	0	150 Amps max
CB 2	Closed (on)	128.3 Amps	450.0 Volts	1	150 Amps max
CB 3	Closed (on)	64.2Amps	450.0 Volts	2	100 Amps max
CB 4	Closed (on)	128.3 Amps	450.0 Volts	0	150 Amps max
CB 5	Closed (on)	128.3 Amps	450.0 Volts	3	150 Amps max

Component	Status	Through Result	Across Result	Priority	Load/Capacity
ABT	Primary (on)	128.3 Amps	442.6 Volts	1	150 Amps max
GB 1	Closed (on)	320.7 Amps	450.0 V	1	350 Amps max
GB 2	Closed (on)	128.3 Amps	450.0 V	3	250 Amps max

5.2.1.2 Defining the Damage Scenario by Setting Cable 2 to "Failed"

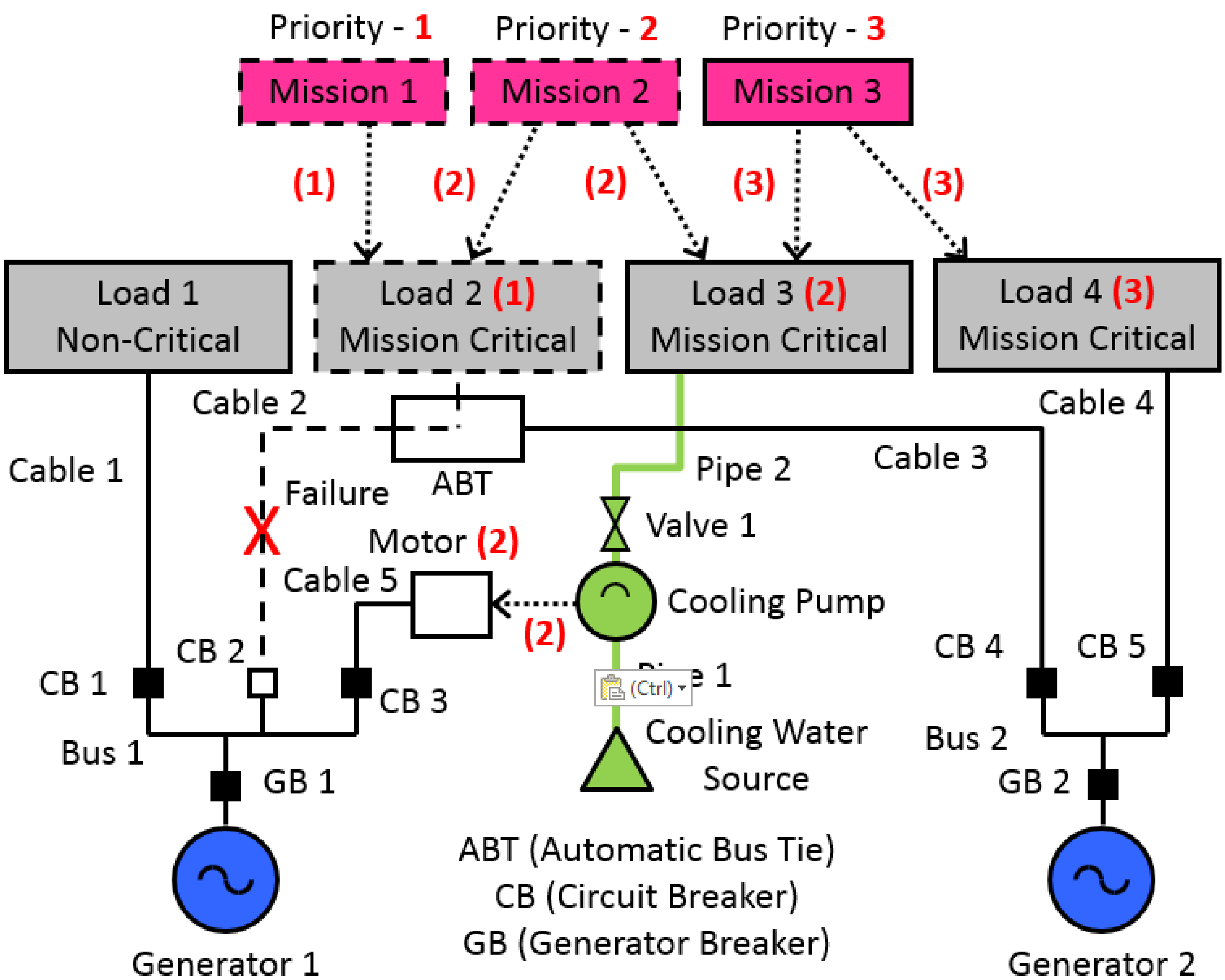

Figure 28 Simplified Interdependent System effect of Cable 2 Failure and Isolation on Mission Status, Before ABT Operates

The primary input for initiating a reconfiguration problem is to "fail" a component. In Figure 28, Cable 2's status is set to "failed," which represents Cable 2 sustaining damage that makes it non-serviceable. When reconfiguration is run, the algorithm searches through the system to find all failed components and then traces back along the each failed component's feeder path towards its source to the first switchable device it finds. If the device is set to operable, reconfiguration operates the device which changes its status to “off,” which in turn interrupts service flow through it. For electrical devices, the “off” state is “open.” For fluid systems, the “off” state is “closed.”

When Circuit Breaker 2 (CB 2) is opened, the circuit breaker is set to “Lost Service.” Successive components receiving services through CB 2, in feeder path order, lose service when the component before them loses service. This reaction chain continues to the end point of the feeder path trace up to Load 2. When Load 2 loses service, loss of service is propagated to the Mission 1 and Mission 2 objects using the AD and OD set relationships defined using dependency components. Loss of service is indicated in the figure using dashed lines. Loss of service interruption to Mission 1 and Mission 2 are noted by Reconfiguration and reported as part of analysis results.

As discussed in subsection 5.2.4, discrete event reaction related traces and data propagation such as loss of service can be performed significantly faster than flow analysis. As a result, constraints and objectives represented by discrete reaction-based relationships between components can be used by trace-based algorithms to evaluate large systems relatively quickly. Reconfiguration options which do not satisfy trace-driven component violation constraint and priority based mission objective requirements can be removed from the set of possible solutions, which after being reduced in number can then be analyzed using flow analysis based constraints and objectives. Component state changes for Figure 28, shown in Table 14 and in the following example analysis step result tables are marked in Bold.

Table 14 Simplified Interdependent System Example Component Status following Failure and Isolation of Cable 2, Before ABT Operates

Component	Status	Through Result	Across Result	Priority	Load/Capacity
Mission 1	**Lost service**			1	
Mission 2	**Lost service**			2	
Mission 3	Has service			3	
Load 1	Has service	128.3 Amps	440.1 Volts	0	100 kVA, 0.9 pf
Load 2	**Lost service**	**0 Amps**	**0 Volts**	1	100 kVA, 0.9 pf

Component	Status	Through Result	Across Result	Priority	Load/Capacity
Load 3	Has service	81.3 psi	100 GPM	2	100 GPM
Load 4	Has service	128.3 Amps	440.1 Volts	3	100 kVA, 0.9 pf
Motor	On	64.2 Amps	449.8 Volts	2	50 kVA, 0.9 pf
Cable 1	Has service	128.3 Amps	440.1 Volts	0	231.0 Amps max
Cable 2	**Lost Service**	**0 Amps**	**0 Volts**	1	231.0 Amps max
Cable 3	Has service	0 Amps	450.0 Volts	0	231.0 Amps max
Cable 4	Has service	128.3 Amps	440.1 Volts	3	231.0 Amps max
Cable 5	Has service	64.2 Amps	449.8 Volts	2	231.0 Amps max
Pump	On	100.0 psi	100 GPM	2	150 GPM/150 psi
Pipe 1	Has service	2.3 psi	100 GPM	2	250 psi max
Pipe 2	Has service	81.3 psi	100 GPM	2	250 psi max
Valve	Open (on)	99.8 psi	100 GPM	2	150 psi max
CB 1	Closed (on)	128.3 Amps	450.0 Volts	0	150 Amps max
CB 2	**Open (off**)	0 Amps	450.0 Volts	1	150 Amps max
CB 3	Closed (on)	64.2 Amps	450.0 Volts	2	100 Amps max
CB 4	Closed (on)	128.3 Amps	450.0 Volts	0	150 Amps max
CB 5	Closed (on)	128.3 Amps	450.0 Volts	3	150 Amps max
ABT	Primary (on) **Lost Service**	0 Amps	0 Volts	1	150 Amps max
GB 1	Closed (on)	**192.4 Amps**	450.0 V	1	350 Amps max
GB 2	Closed (on)	128.3 Amps	450.0 V	3	250 Amps max

5.2.1.3 Device Level ABT Reaction to Loss of Cable 2 and GB 2 Overload

In Figure 29, the ABT sees that its primary side connection loses service when Cable 2 loses service and reacts by switching to receive power from its secondary source side (Cable 3), which still has service. When the feeder path for the ABT is changed, the ABT initiates a new feeder path trace.

Feeder path trace for the ABT before isolation of Cable 2:

$$FPT_{Load2} = \{ABT, Cable\ 2, CB\ 2, Bus\ 1, GB\ 1, Generator\ 1\} \quad (39)$$

Feeder path trace for the ABT after isolation of Cable 2 and automated switching of the ABT:

$$FPT_{Load2} = \{ABT, Cable\ 3, CB\ 4, Bus\ 2, GB\ 2, Generator\ 2\} \quad (40)$$

Following ABT switching, the service path to Load 2 is restored. Load 2 reacts to its feeder path predecessor component having source connectivity and resets its status to "has service," and then initiates an update of its (Load 2) feeder path trace. The dependencies for the Mission 1 and Mission 2 objects then transfer the "has service" state information to Mission 1 and Mission 2, which then reset their states to "has service." This state change is noted by the reconfiguration algorithm and recorded as part of results.

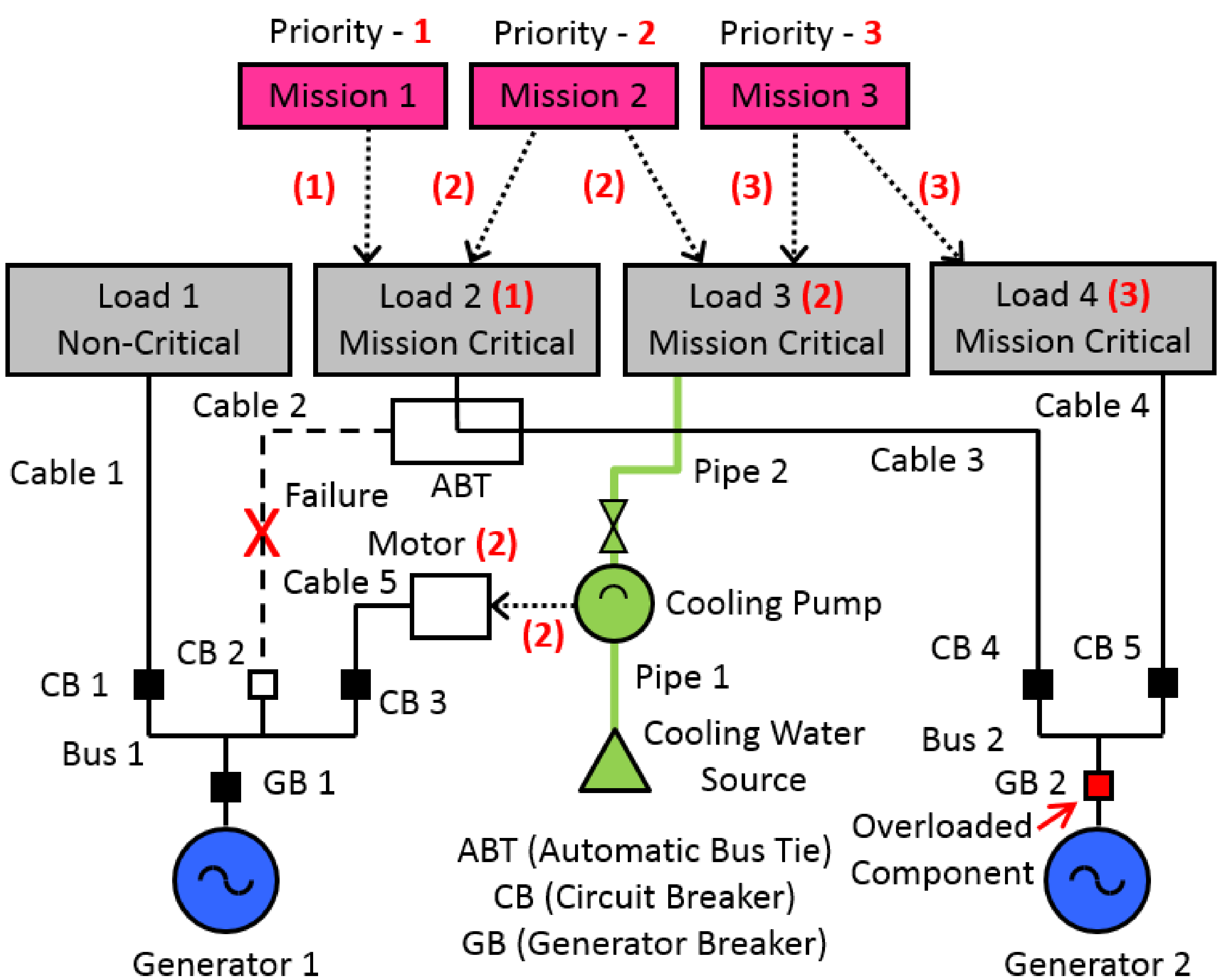

Figure 29 Simplified Interdependent System with Failed Cable, Switched ABT and Propagated Loss of Service State Changes

The state changes for Mission 1 and Mission 2 result in each mission object initiating a new priority propagation operation, which begins by clearing all previously set priorities. The dependency component for Mission 1 sets the priority for Load 2, which is part of the physical system, to Level 1. The new feeder path trace for Load 2 is then used to re-propagate priorities for Load 2, which end up being the same as before the ABT switched.

After flow analysis is run, each component object in the physical system parts of the integrated model check they're respective across (voltage and pressure) and through (amps and flow) variables against their respective across and through operating constraint values. If a violation exists, the affected component object changes its status to “violation.” In Figure 29, the amp rating for Generator Breaker 2 (GB 2) is exceeded when flow through it is increased as a result of the ABT switching to its secondary source, which is Cable 3. The maximum load setting for GB 2 was set to represent a reduced operating limit which in turn can be used to represent a partial damage state or temporary maintenance state. Component status after Cable 2 is isolated, and the ABT switches to receiving power from Cable 3 is shown in Table 15.

Table 15 Simplified Interdependent System Component States after Fault Isolation and ABT Operation

Component	Status	Through Result	Across Result	Priority	Load/Capacity
Mission 1	**Has service**			1	
Mission 2	**Has service**			2	
Mission 3	Has service			3	
Load 1	Has service	128.3 Amps	440.1 Volts	0	100 kVA, 0.9 pf
Load 2	**Has service**	**128.3 Amps**	**440.1 Volts**	1	100 kVA, 0.9 pf
Load 3	Has service	81.3 psi	100 GPM	2	100 GPM
Load 4	Has service	128.3 Amps	440.1 Volts	3	100 kVA, 0.9 pf
Motor	On	64.2 Amps	449.8 Volts	2	50 kVA, 0.9 pf
Cable 1	Has service	128.3 Amps	440.1 Volts	0	231.0 Amps max
Cable 2	Lost Service	0 Amps	0 Volts	**0**	231.0 Amps max
Cable 3	Has service	**128.3 Amps**	**440.1 Volts**	**1**	231.0 Amps max
Cable 4	Has service	128.3 Amps	440.1 Volts	3	231.0 Amps max
Cable 5	Has service	64.2 Amps	449.8 Volts	2	231.0 Amps max
Pump	On	100.0 psi	100 GPM	2	150 GPM/150 psi
Pipe 1	Has service	2.3 psi	100 GPM	2	250 psi max
Pipe 2	Has service	81.3 psi	100 GPM	2	250 psi max
Valve	Open (on)	99.8 psi	100 GPM	2	150 psi max
CB 1	Closed (on)	128.3 Amps	450.0 Volts	0	150 Amps max
CB 2	Open (off)	0 Amps	450.0 Volts	**0**	150 Amps max
CB 3	Closed (on)	64.2 Amps	450.0 Volts	2	100 Amps max

Component	Status	Through Result	Across Result	Priority	Load/Capacity
CB 4	Closed (on)	128.3 Amps	450.0 Volts	**1**	150 Amps max
CB 5	Closed (on)	128.3 Amps	450.0 Volts	3	150 Amps max
ABT	**Secondary (on)**	**128.3 Amps**	**440.1 Volts**	1	150 Amps max
GB 1	Closed (on)	**192.4 Amps**	450.0 V	**2**	350 Amps max
GB 2	Closed (on) **Overloaded**	**256.5 Amps**	450.0 V	**1**	250 Amps max

5.2.1.4 Reconfiguration Removes Overload from the System

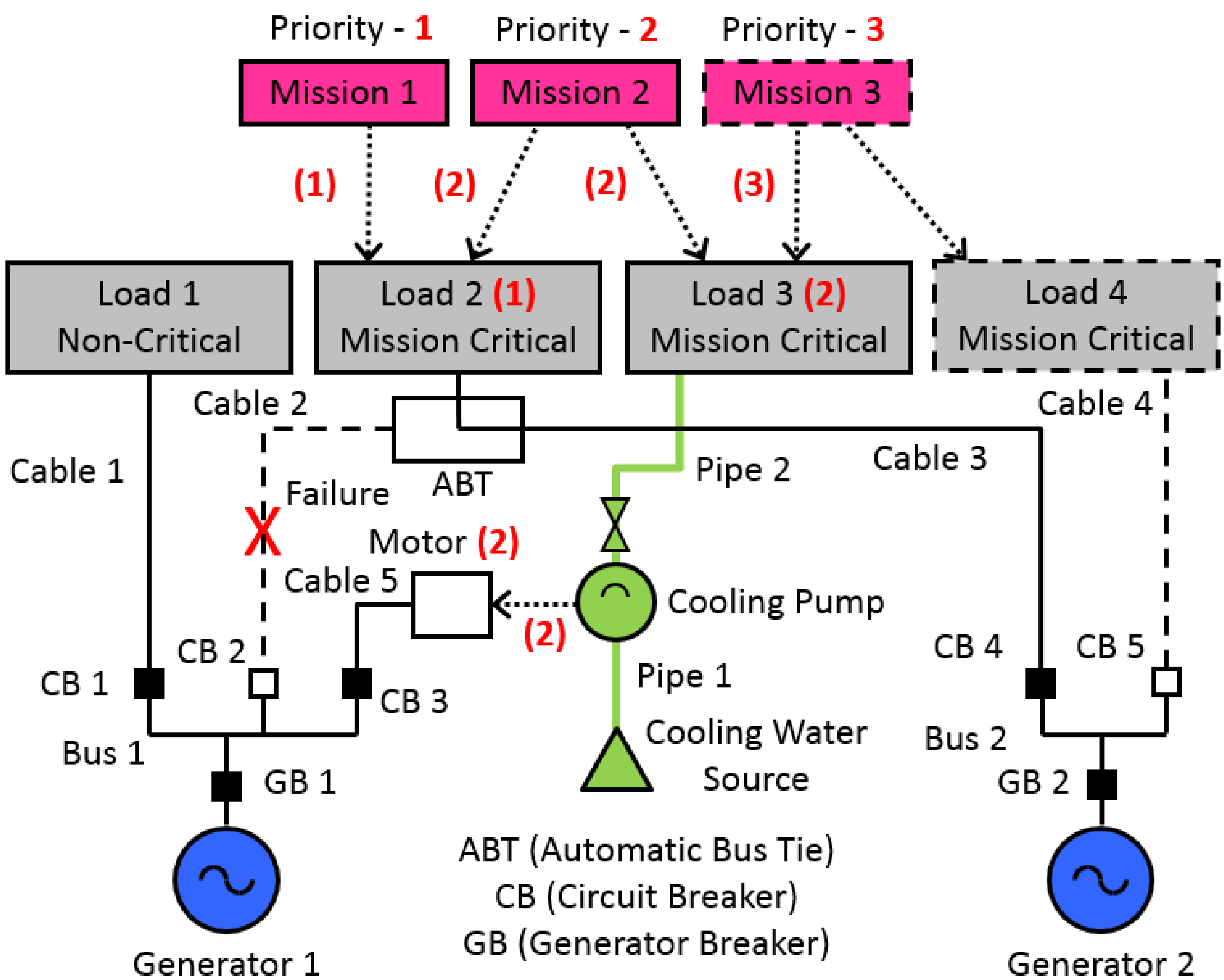

Figure 30 Simplified Interdependent System with Switch Opened to Eliminate Overloaded Generator Circuit Breaker

After identifying the overload at GB 2, reconfiguration notes the violation at GB 2, uses feeder path traces associated with GB 2 and finds that service to Load 2 and Load 4 is supplied through

GB 2. Reconfiguration compares the priorities propagated for these loads and determines that Load 4 has the lowest priority. Reconfiguration uses the feeder path trace to identify the closest operable device that can be used to interrupt service to Load 4, which is CB 5:

The feeder path trace (FPT) for Load 4 is:

$$FPT_{Load4} = \{\text{Cable 4, CB 5, Bus 2, GB 2, Generator 2}\} \quad (41)$$

Reconfiguration opens Circuit Breaker 5 (CB 5) to isolate power to Load 4. Cable 4, Load 4 and then Mission 3 react to the loss of service at CB 5 by changing states to "lost service" and initiating appropriate trace updates. Reconfiguration then runs flow analysis again on the physical parts of the integrated model. No violations are found, and reconfiguration is complete.

System configuration after reconfiguration is complete is shown in Figure 30. Status and flow analysis result for each component after reconfiguration is complete is shown in Table 16. In Figure 31, priorities are reset, representing a change in service objectives which is something that can happen quickly in real operations, and the system is reconfigured accordingly.

Table 16 Simplified Interdependent System Final Component States

Component	**Status**	**Through Result**	**Across Result**	**Priority**	**Load/Capacity**
Mission 1	Has service			1	
Mission 2	Has service			2	
Mission 3	**Lost service**			3	
Load 1	Has service	128.3 Amps	440.1 Volts	0	100 kVA, 0.9 pf
Load 2	Has service	128.3 Amps	440.1 Volts	1	100 kVA, 0.9 pf
Load 3	Has service	81.3 psi	100 GPM	2	100 GPM
Load 4	**Lost service**	**0 Amps**	**0 Volts**	3	100 kVA, 0.9 pf
Motor	On	64.2 Amps	449.8 Volts	2	50 kVA, 0.9 pf
Cable 1	Has service	128.3 Amps	440.1 Volts	0	231.0 Amps max
Cable 2	Lost Service	0 Amps	0 Volts	**0**	231.0 Amps max
Cable 3	Has service	128.3 Amps	440.1 Volts	**1**	231.0 Amps max
Cable 4	**Lost service**	**0 Amps**	**0 Volts**	3	231.0 Amps max
Cable 5	Has service	64.2 Amps	449.8 Volts	2	231.0 Amps max
Pump	On	100.0 psi	100 GPM	2	150 GPM/150 psi

Component	Status	Through Result	Across Result	Priority	Load/Capacity
Pipe 1	Has service	2.3 psi	100 GPM	2	250 psi max
Pipe 2	Has service	81.3 psi	100 GPM	2	250 psi max
Valve	Open (on)	99.8 psi	100 GPM	2	150 psi max
CB 1	Closed (on)	128.3 Amps	450.0 Volts	0	150 Amps max
CB 2	Open (off)	0 Amps	450.0 Volts	**0**	150 Amps max
CB 3	Closed (on)	64.2 Amps	450.0 Volts	2	100 Amps max
CB 4	**Open (off)**	**0 Amps**	450.0 Volts	**1**	150 Amps max
CB 5	Closed (on)	128.3 Amps	450.0 Volts	3	150 Amps max
ABT	Secondary (on)	128.3 Amps	440.1 Volts	1	150 Amps max
GB 1	Closed (on)	192.4 Amps	450.0 V	**2**	350 Amps max
GB 2	Closed (on) **(No-Overload)**	**128.2 Amps**	450.0 V	**1**	250 Amps max

5.2.1.5 Priorities are reset and Reconfiguration adjusts System Configuration

Before the failed section in Cable 2 is repaired and returned to service, mission priorities are reset at mission objects to represent a change in commanded mission objectives. As a result of the priority change at the mission objects, priorities are re-propagated and reconfiguration is run again. Reconfiguration finds that changing priorities makes Load 4 more important than Load 2, restores service to Load 4, finds an overload at GB 2, and then opens CB 4 to remove the overload condition affecting GB 2 and the operation of Load 4. This results in Load 2 losing service, which causes Mission 1 and Mission 2 to lose service. Final system component, critical load and mission states and priorites following completion of reconfiguration, are shown in Figure 31 and Table 17.

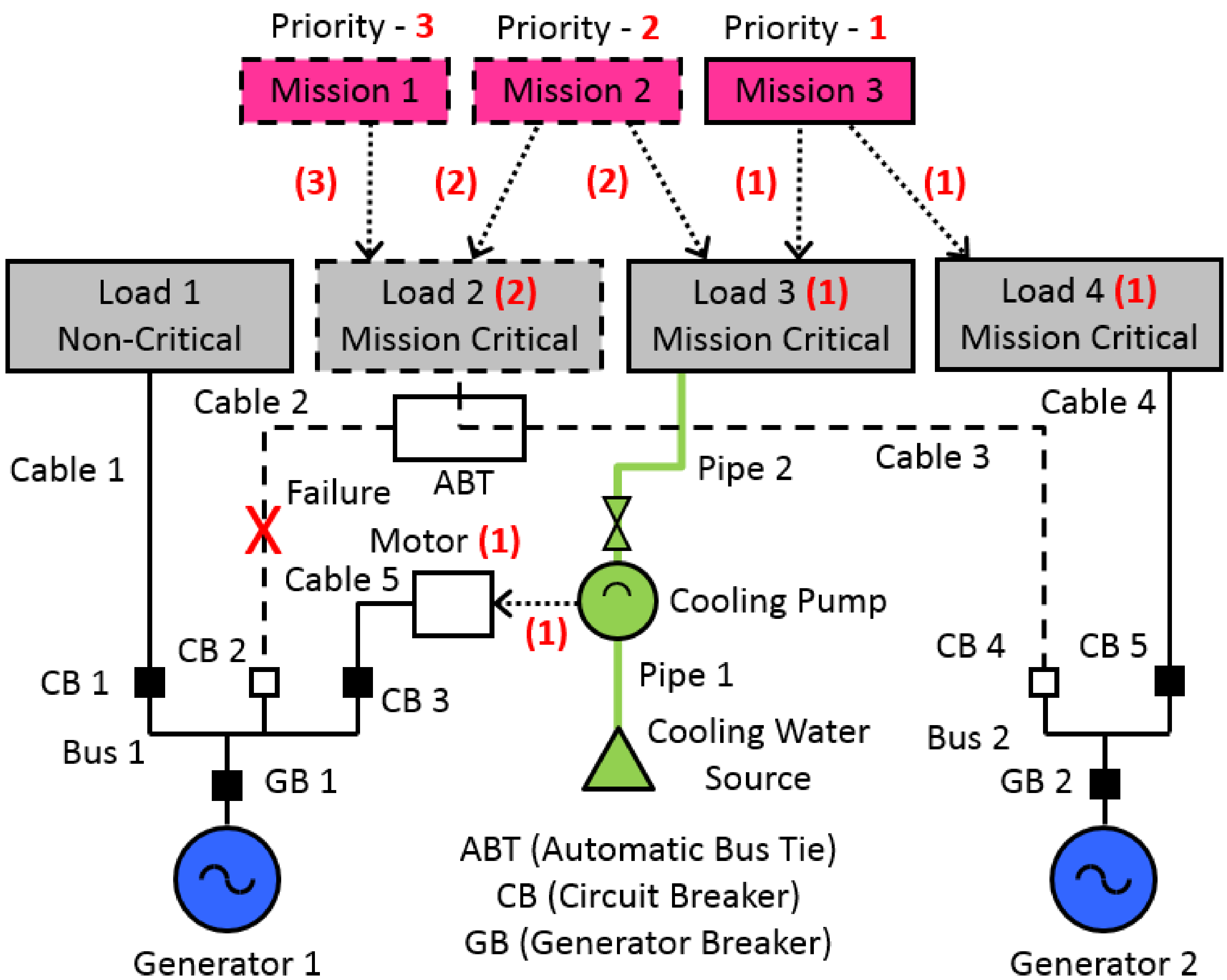

Figure 31 Simplified Interdependent System Reconfigured to Support Different Mission Priorities with Cable 2 Still Failed

Table 17 Simplified Interdependent System Final Component States after Mission Priorities are reset and Reconfiguration is Run Again

Component	**Status**	**Through Result**	**Across Result**	**Priority**	**Load/Capacity**
Mission 1	**Lost Service**			**3**	
Mission 2	**Lost Service**			2	
Mission 3	**Has service**			**1**	
Load 1	Has service	128.3 Amps	440.1 Volts	0	100 kVA, 0.9 pf
Load 2	**Lost service**	**0 Amps**	**0 Volts**	**2**	100 kVA, 0.9 pf
Load 3	Has service	81.3 psi	100 GPM	2	100 GPM
Load 4	**Has service**	**128.3 Amps**	**440.1 Volts**	**1**	100 kVA, 0.9 pf
Motor	On	64.2 Amps	449.8 Volts	2	50 kVA, 0.9 pf

Component	**Status**	**Through Result**	**Across Result**	**Priority**	**Load/Capacity**
Cable 1	Has service	128.3 Amps	440.1 Volts	0	231.0 Amps max
Cable 2	Lost Service	0 Amps	0 Volts	0	231.0 Amps max
Cable 3	**Lost service**	**0 Amps**	**0 Volts**	1	231.0 Amps max
Cable 4	**Has service**	**128.3 Amps**	**440.1 Volts**	**1**	231.0 Amps max
Cable 5	Has service	64.2 Amps	449.8 Volts	2	231.0 Amps max
Pump	On	100.0 psi	100 GPM	2	150 GPM/150 psi
Pipe 1	Has service	2.3 psi	100 GPM	2	250 psi max
Pipe 2	Has service	81.3 psi	100 GPM	2	250 psi max
Valve	Open (on)	99.8 psi	100 GPM	2	150 psi max
CB 1	Closed (on)	128.3 Amps	450.0 Volts	0	150 Amps max
CB 2	Open (off)	0 Amps	450.0 Volts	0	150 Amps max
CB 3	Closed (on)	64.2 Amps	450.0 Volts	2	100 Amps max
CB 4	**Open (off)**	**0 Amps**	450.0 Volts	**2**	150 Amps max
CB 5	**Closed (on)**	**128.3 Amps**	450.0 Volts	3	150 Amps max
ABT	Secondary (on)	128.3 Amps	440.1 Volts	1	150 Amps max
GB 1	Closed (on)	192.4 Amps	450.0 V	2	350 Amps max
GB 2	Closed (on)	128.2 Amps	450.0 V	1	250 Amps max

5.2.1.6 System Size and Complexity Analysis and Data Issues

The example problem was designed to show how a networked set of logic and physical model-based objects can be used to represent interdependent infrastructure systems and missions in a generic, homogenous way that can be analyzed using common algorithms that run across the entire interdependent system model. The concepts presented is this example were developed by combining problem definition observations taken from shipboard system design and operations management, and extension of existing GTA analysis techniques. They were designed to be sufficient for automating operations management which:

- Needs to be fast enough to support real-time operations management
- Involves analysis across multiple types of systems and operation doctrine driven processes involving restoration paths, discrete event reaction, and defined service system and critical mission system resources

- Focuses on identifying viable restoration options that meet mission capability requirements, system physical operation constraints, and mission readiness time constraints[5]

In standard operations management analysis, both naval ships and commercial utilities often use large sets of pre-evaluated scenarios and data, each specifically defined for a given set of damage and system operating conditions. When an actual damage event occurs, the predefined scenarios are searched to find the one that most closely fits the given current conditions. This typically works well for single contingency events and normal operating conditions, but can become difficult to use and maintain for events involving abnormal operating conditions or simultaneous damage at two or more locations where the effects of damage and response efforts overlap (35).

Model size and complexity for analysis of real systems is also an issue. Real system models can contain tens of thousands to millions of components. For example, the actual distribution system pilot circuit model used for the smart-grid supervisory control development discussed in section 5.3 contains over 7,600 components. The model for the entire distribution system which the pilot circuit was taken from contains over 360,000 components. Another GTA operations management analysis development project used an urban distribution network pilot circuit which contained over 10,000 components. Electrical distribution system models cited in literature for use in optimized reconfiguration analysis are typically small test systems or relatively small, simplified versions of real systems. Model size and complexity limits suitable for use with standard optimization analysis are not well documented. It is generally accepted that standard optimization approaches are best suited for systems that do not exceed being made up of thousands of components. This is an area that remains for further research.

5.2.2 Dependency Component Development

Development of a “dependency” component is one of the key elements used for structuring interdependent system analysis using a GTA based forward-backward sweep iterative solution

---

[5] The use of switch operation times to model mission readiness time constraints is addressed in subchapter section 5.2.5.

process. The dependency component is used to define mission performance objective, discrete event reaction, and physics based power flow propagation at interconnection points between systems. GTA dependency components use the dependency symbol and direction properties defined in the Unified Modeling Language (UML) (10). In UML, a dependency is drawn as an arrow that starts at the dependent component and points to the component that is being depended on upon. In Figure 32 the pump is "dependent" on the motor because the pump receives its driving power from the motor. If the motor loses service, the pump stops. In GTA, the dependency component is used to structure exchange of power, loss of service and mission load priorities information between components at interconnection points. This includes defining direction of propagation for each type of information, which is structured to conform to standard conventions used in forward-backward iteration.

The following section outlines forward-backward sweep iteration steps for the interdependent motor – pump system illustrated in Figure 32. The description uses the propagation rules defined in Figure 32 and Physical System Modeling equations defined by Wellstead (12). A detailed simulation was run in Dew to test this concept using a model of a fluid turbine turning a mechanical shaft that in turn drove a centrifugal pump that feeds a nozzle.   Flow to the turbine was controlled by a controllable valve that senses output pressure at discharge side of the pump. See subsection 5.2.3.

Determining steady-state operating points for nonlinear coupled systems like an electrical system and a motor driving a centrifugal pump that in turn drives pressure and flow in a fluid system requires some type of nonlinear, iterative analysis. Characteristics for each of the components in the system can be defined using equations, a family of curves or a combination of both. For a DC motor driving a centrifugal pump, shaft rpm is one of the steady-state operating points that must be solved for. For a simple AC motor, rpm is fixed by virtue of motor design and system operating frequency. The across and through constituent equations used in Physical System and Physical Network modeling work well with forward-backward sweep analysis.

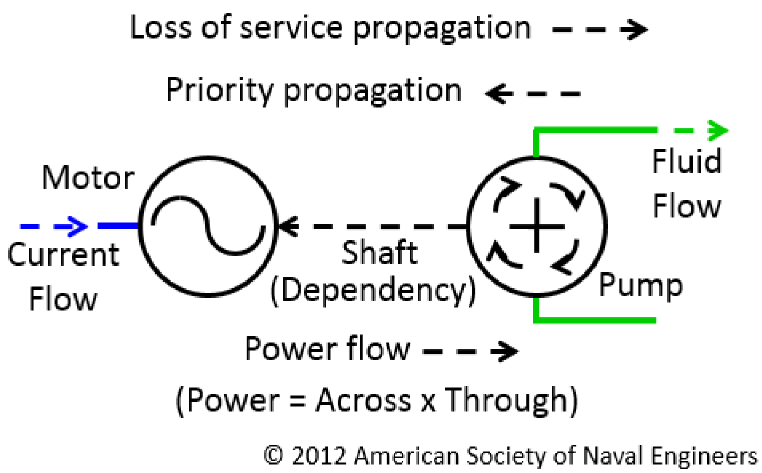

Figure 32 GTA Dependency Component Data Propagation

The purpose for the iteration steps outlined below is to show how a dependency component can be used with forward-backward iteration to model an electrical motor and fluid pump as two GTA "circuits" joined together through a dependency component. Distributed Engineering Workstation (Dew) software, which is produced by Electrical Engineering Design, Inc., was used for the modeling and automated analysis performed for the research and development presented in this dissertation. At the time that the work presented in this dissertation was performed, Dew GTA interdependent system analysis could be used to run loss of service and mission priority information across multi-discipline system dependency points, but was set up to only be able to run flow analysis on one system type at a time. This was a function of programming, and not a limitation of the approach being used. Running flow analysis across multiple system types at the same time using Dew is subject to further development.

First forward iteration:

1. Forward electrical circuit iteration:
    a. Propagate voltage from electrical system source to the terminals of the motor.

       Where: V is terminal voltage, v is armature voltage, ω is shaft rotation speed and τ is shaft torque.

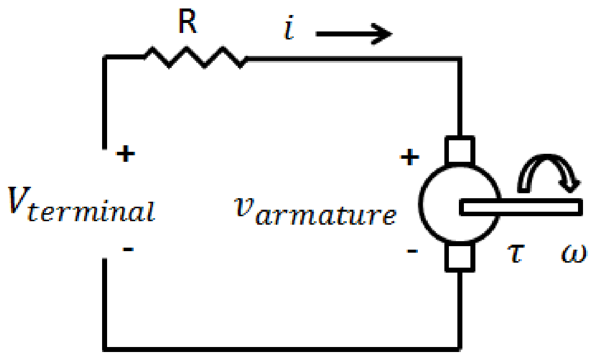

Figure 33 DC Motor Circuit

b. Shaft speed is zero for the first voltage sweep which results in armature current being equal to locked rotor current.

$$w = 0, \quad v = 0, \quad i = \frac{V-v}{R} \quad and \;\; \tau = k\emptyset_1 i \qquad (42)$$

Where the constitutive relations for a DC motor (12) are:

$$v = n\omega \quad and \quad \tau = ni \qquad (43)$$

$n = k\emptyset_1$ is the transformer constitutive modulus

$\emptyset_1$ is flux

2. Forward shaft component iteration:
   a. Motor torque is used to set shaft torque at the pump.
3. Forward fluid circuit iteration:
   a. Propagate pressure from the fluid source to the suction side of the pump.

   Where: P_1 is pump suction side pressure

   P_2 is pump discharge side pressure

   ω is shaft rotation speed

   τ is shaft torque

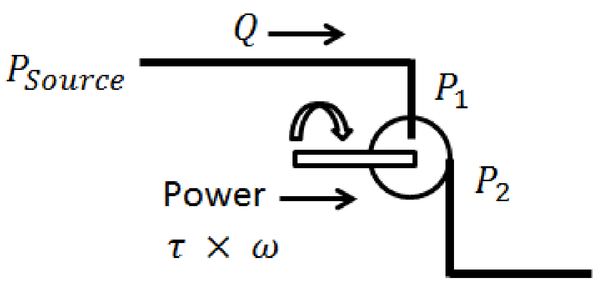

Figure 34 Positive Displacement Pump Circuit

b. Set pump suction and discharge pressure using propagated pressure from the source and shaft torque.

Where the constitutive relations for the pump (12) are:

$$\omega = rQ \quad and \quad P_2 - P_1 = r\tau \tag{44}$$

$r = k_1 = k_2$ is the gyrational constitutive modulus (45)

c. Propagate pressure from the pump discharge side to the end of the fluid circuit.

First backward iteration:

1. Backward fluid circuit iteration:
   a. The fluid system load is used to set flow through the fluid circuit.
   b. Flow through the pump is used to set pump rotational speed.

$$Q = \frac{\omega}{r} \tag{46}$$

2. Backward shaft component iteration:
   a. Pump rotation speed is used to set shaft rotational speed.
3. Backward motor circuit iteration.
   a. Motor current is defined as a function of shaft torque and flux (flux is a function of motor field circuit current).

$$i = \frac{\tau}{n} \qquad n = k\emptyset_1 \tag{47}$$

   b. Motor current is propagated back to the electrical system source.

This process is repeated until all circuits reach their steady-state operating point. The forward-backward sweep conventions shown in Figure 32 line up well with the standard constitutive relations used in Physical System Modeling and the auxiliary system design and control, and process management conventions used in shipboard casualty control and damage control.

5.2.3 Propagating Power across Dependency Components between Systems

Standard system equation definitions and analysis conventions for across, through and power can be found in a number of physical system modeling books. Standard system variable and equation definitions for electrical, fluid and rotational mechanical systems from Blackwell (11) and Wellstead (12) are shown in Table 18.

Table 18 Typical System Type Across and Through Variable Definitions

System Type	Across Variable	Through Variable	Power
AC Electrical	Voltage (**V**)	Amps (**A**)	**S** = **V** x **A**
DC Electrical	Voltage (V)	Amps (A)	P = V x A
Fluid	Pressure (P)	Flow (Q)	P = P x Q
Mechanical	Torque (τ)	Rotational Speed (ω)	P = τ x ω

Note: for AC Electrical **S**, **V** and **A** are complex number - phasor quantities

The following example uses a fluid turbine, shaft, fluid pump combination modeled using a fluid system flow algorithm implemented using a modified version of Dew. This example was programmed using a set of arbitrary pump and turbine characteristic equations that had not yet been standardized to follow the conventions shown in Figure 32, but were sufficient to show that power flow (effort x flow) and discrete event information analysis between systems could be performed by extending forward-backward sweep iterations across a dependency component. Additional work needs to be performed to refine and validate pump and turbine speed and pressure relation equations for use in analysis. As shown in Figure 36, the dependency component “shaft” propagates both discrete and steady-state related data between the water turbine side of the system and the pump-nozzle side. Loss of service at the turbine results in the pump switching to off.

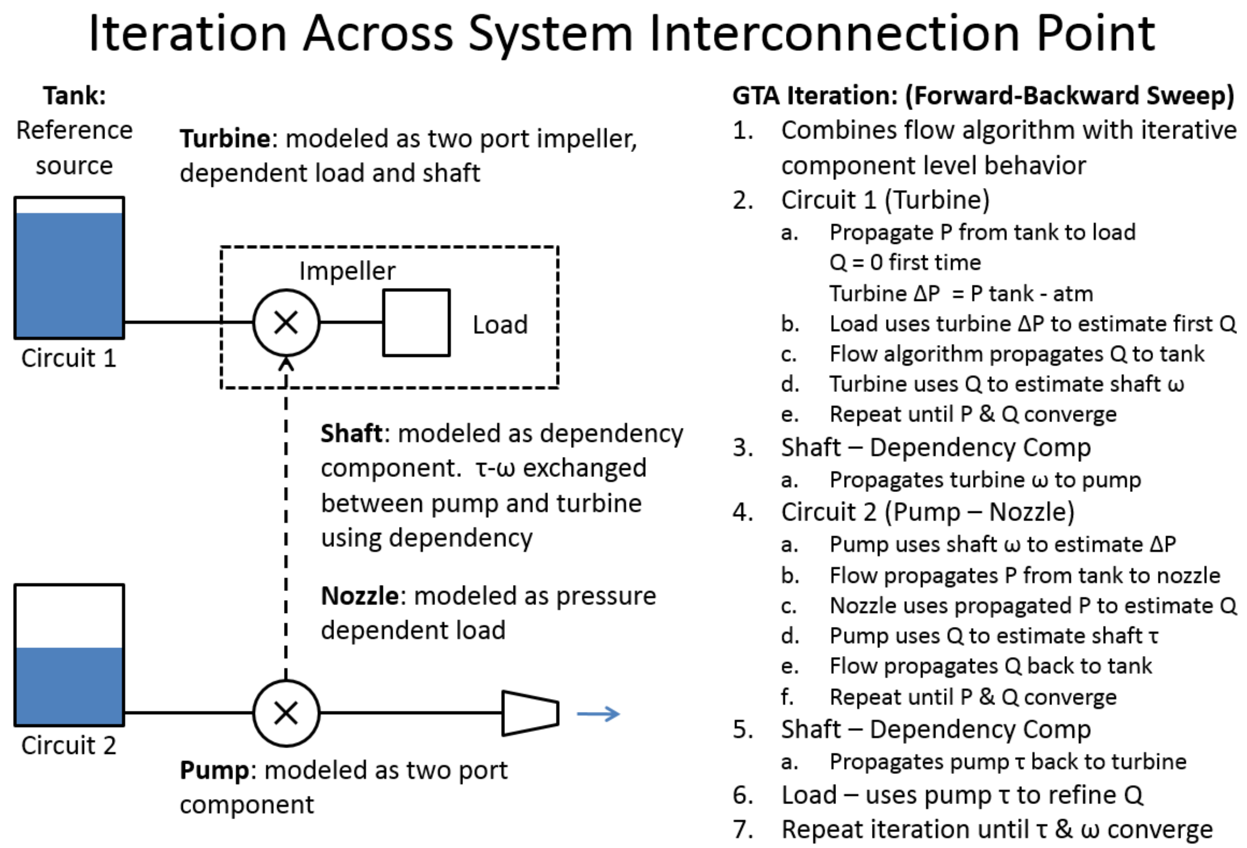

Figure 35 Dependency Component Power Flow Test System

Figure 37 through Figure 39 show results for the first few iterations for the two interdependent fluid system circuits shown in Figure 35, which are each supplied by separate water tank source components labeled as Tank components (1) in Figure 36. The two circuits are joined by a dependency component labeled component (2) that represents the shaft driven by turbine component (4). The arrow indicates that the pump (3) is dependent upon the turbine (4) in the upper system. The nozzle (5) acts as a pressure dependent load in the lower system. A controlled globe valve in the upper system, component (6), is set to maintain pressure supplied by the pump to the nozzle in the lower system. Operation of the valve affects flow in both systems.

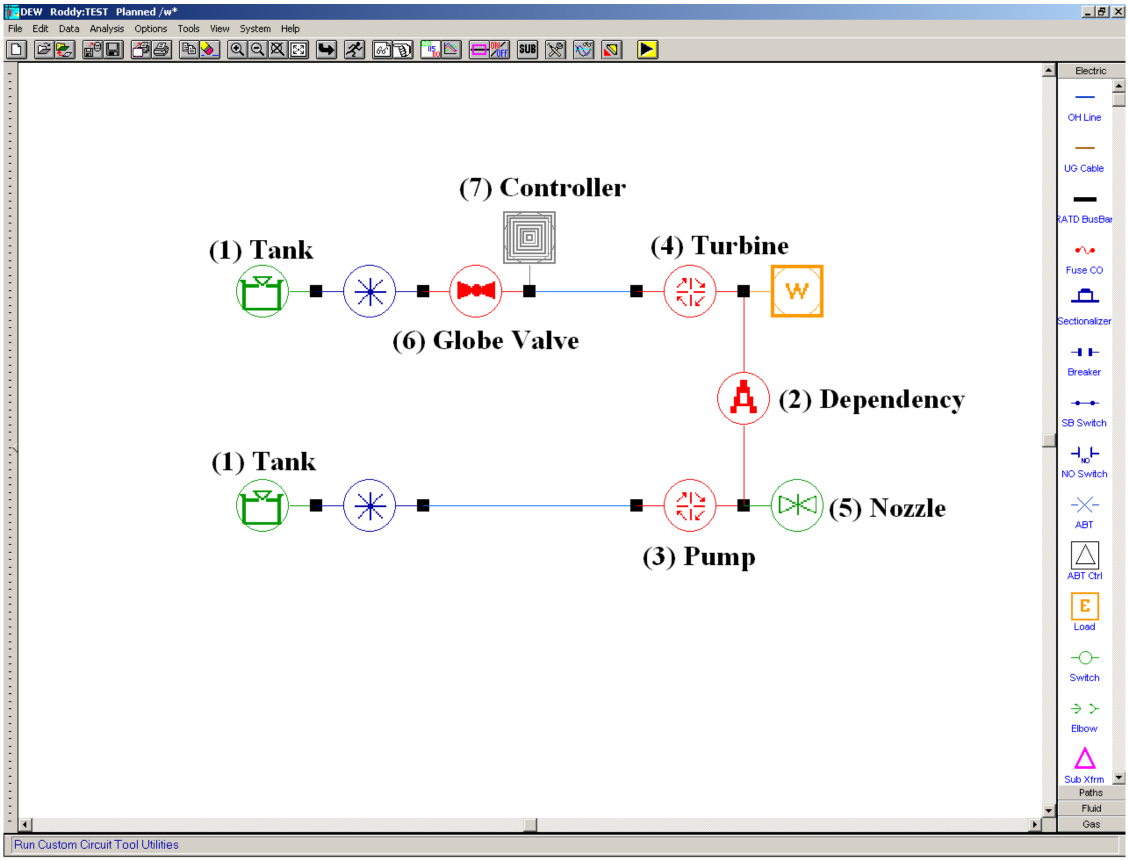

Figure 36 Turbine-Pump/Nozzle Interdependent Test System Dew model

The globe valve controller in the top circuit was set to control the pressure at the outlet side of the pump in the second circuit to 150 psi. The globe turbine control valve in the top circuit was initially set at a position equivalent to 50% open. The turbine tank pressure in the turbine circuit was initially set at 77.9 psi. After the first iteration, the pressure and flow values converged to the ones shown in Figure 37. Pressure decreases along the upper system due to resistance driven pressure drops across the supply pipe and the globe valve. Resistance based pressure drop is also modeled in the lower circuit.

The pressure to the inlet side of the pump is determined by the pressure set at the tank, which is a function of water level in the tank. The pressure at the outlet side of the pump is determined by the sum of the inlet pressure and the increase in pressure across the pump caused by mechanical rotation driven by the turbine.

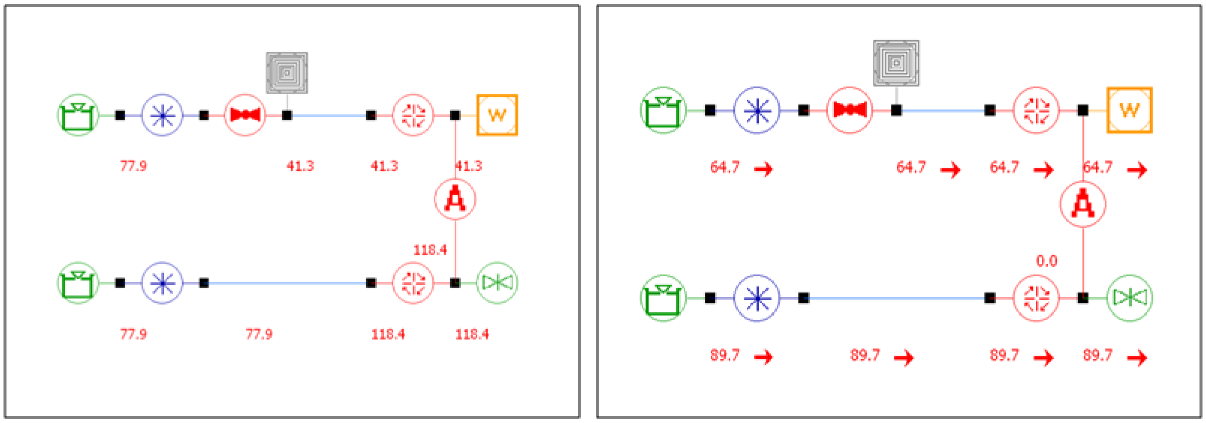

Figure 37 First Iteration Pressure and Flow

Since the pressure at the pump is lower than the set point amount, the controller, which is set to read pressure at the outlet side of the pump and is also set to control the globe valve, opens the globe valve to increase flow through the turbine. This, in turn, results in an increase in the pressure at the pump in the lower system.

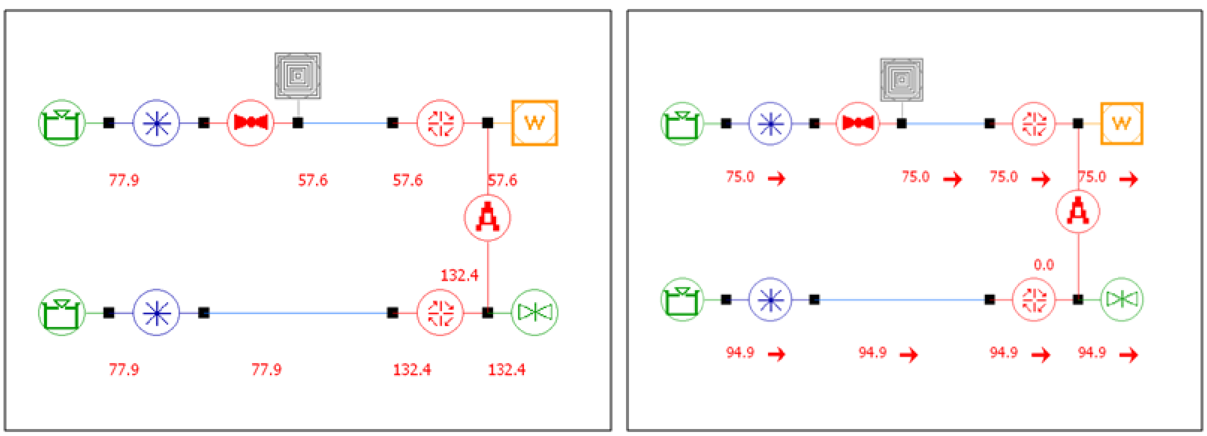

Figure 38 Second Iteration Pressure and Flow

In the next iteration, see Figure 39, the pump outlet pressure overshoots and the controller reduces the position setting on the globe valve. The system then approaches equilibrium after several more iterations.

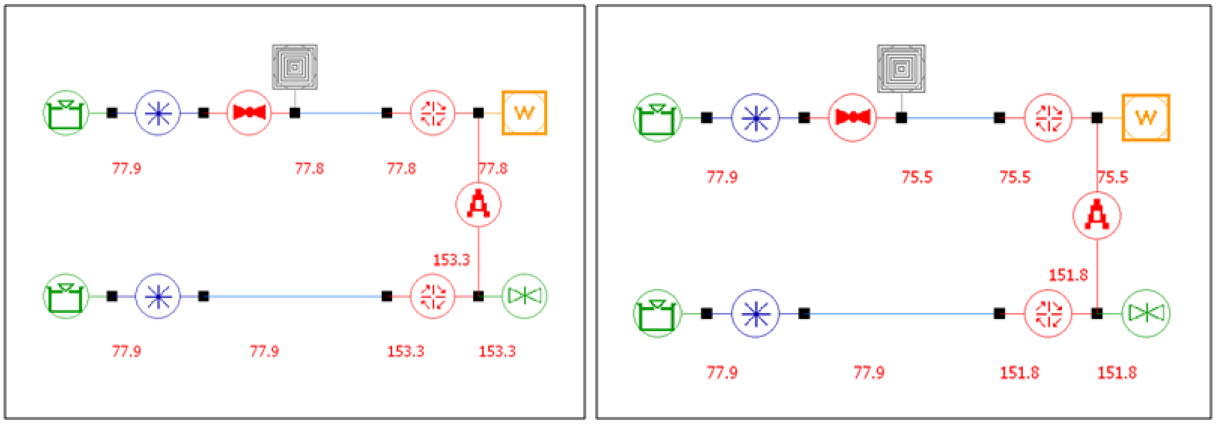

Figure 39 Additional Pressure Iteration Results

5.2.4 Propagating Loss of Service and Mission Priority Information

The same dependency component conventions used to propagate across and through variable, loss of service and priority information across interdependent points between physical systems can also be used with abstract "logical" components. Logical components can be used to model "missions," which represent activities or goals which require some combination of physical system services and process related resources to be performed. Further "logical system" detail can be added to logic components that represent operation measures such as mission readiness performance levels, and logical "mission critical equipment and systems," which can be represented as individual logic blocks or a combination of logic block components and physical system components.

Figure 40 is a simplified ship system modeled using Dew software. The total system is made up of a logical system containing abstract missions (AAW and ASW) and abstract critical mission systems (Radar 1, Gun and Radar 1) which the AAW and ASW missions are dependent upon. The dependencies between the logical component missions and mission critical systems are defined by inserting dependency components into the model. This part of the model can be defined as a "logical system" or subsystem that is associated with an electrical power and fluid system model.

The Radar 1, Gun and Radar 2 mission critical logic block components are associated with physical system fluid and electrical load components. If the physical system load components directly supporting the critical mission system blocks lose service, the effect of that loss is then propagated through the logical system part of the model. Abstract system blocks representing propulsion system equipment (Propulsion 1, Propulsion 2 and Propulsion 3) are also included in the model, which are used to show electrical and fluid system operation loss of service impact on propulsion.

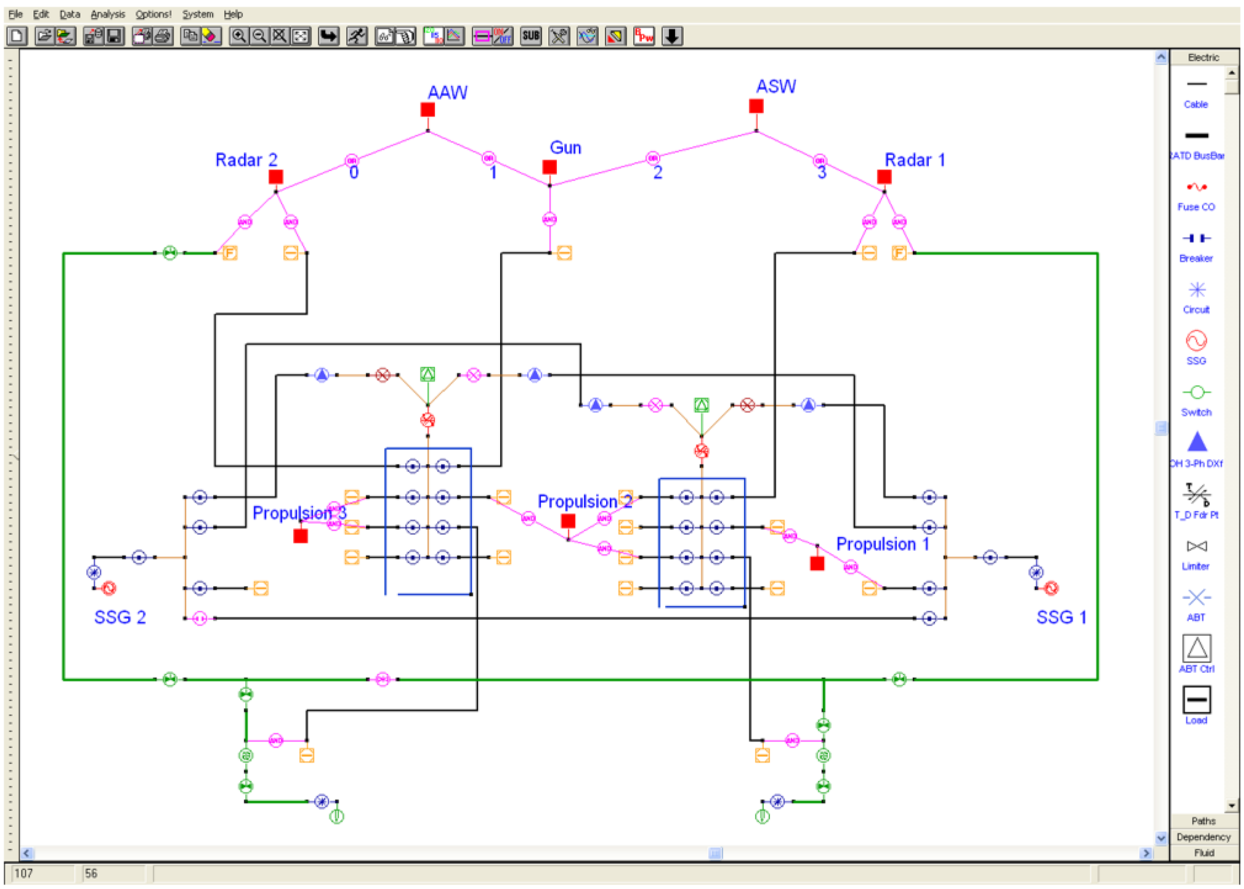

Figure 40 Simplified Ship Mission, Mission System, and Physical System Model

The "And" and "Or" dependency component designations shown in Figure 41 are used to structure service loss logic where the dependent component receives service from more than one depended on component. If a component has multiple "And" dependencies with other components, service loss to any one of those supporting components will cause the dependent component to lose service. For multiple "Or" dependencies, all supporting components must lose service before the dependent component loses service (36). These components and component relationships were designed to make it possible to generate mission readiness and recovery analysis results and measures directly from the model, that are designed to fit well with mission capability measures currently used in naval ship design and operation readiness reporting.

Figure 41 is a zoomed in section of the example system model shown in Figure 49. This model was used to generate an integrated system recovery plot that shows the effect on system readiness vs. time of reconfiguration for damage isolation and restoration. This purpose for the model was to test dependency component priority and loss of service propagation conventions using a complex mix of interdependent fluid and electrical system interconnections, interconnections between mission system components and physical system loads, interconnections between mission system equipment components and mission components, and direct interconnections run from mission to mission.

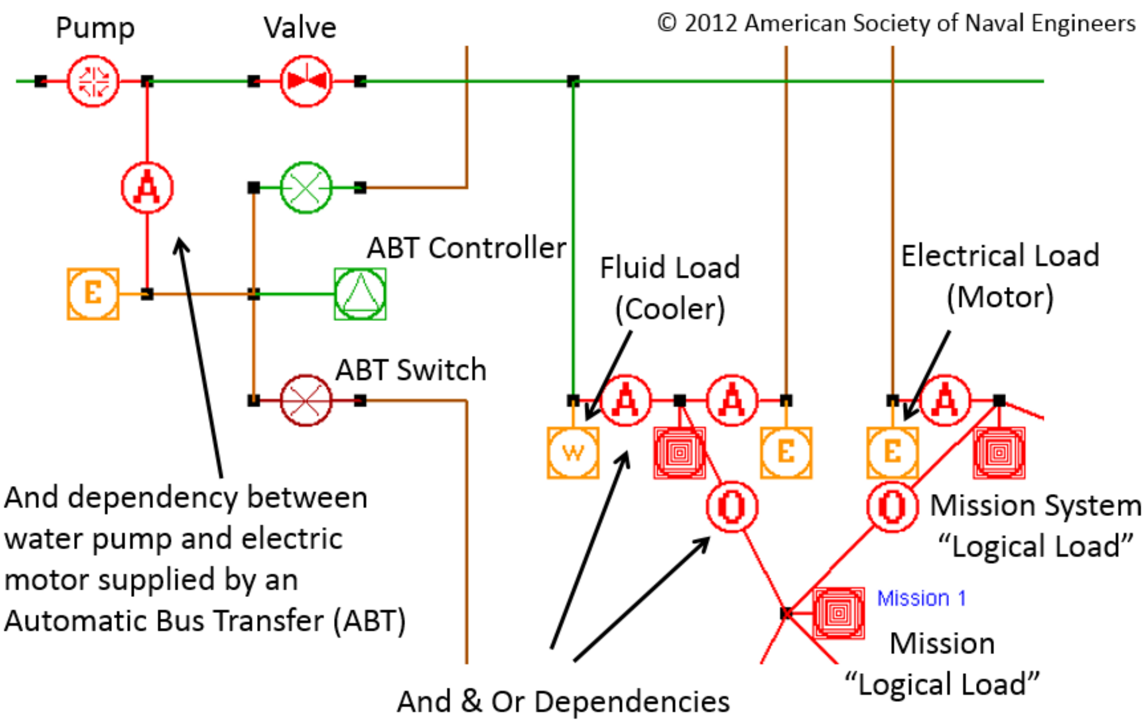

Figure 41 Abstract Mission Objects, Mission System Loads and Dependency Components

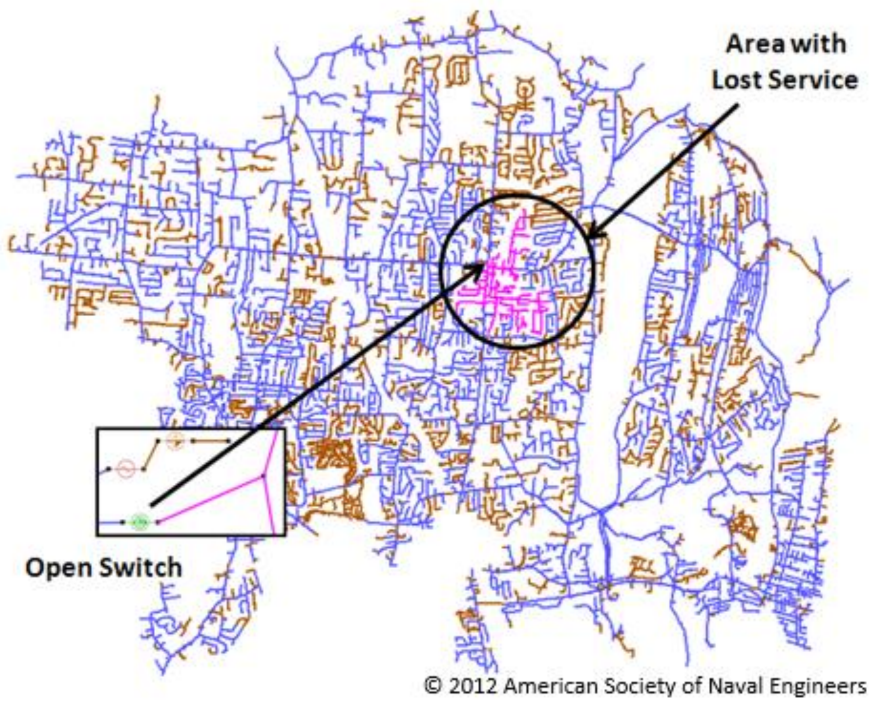

Figure 42 Loss of Service Propagation in an Electrical Distribution Model

Figure 42 shows service loss propagation results (highlighted in pink) from opening a switch in a 50,000 component, commercial power utility system modeled in Dew. Service loss is propagated through the system using traces that simulate individual components reacting to loss of service by the component immediately preceding them in accordance with their defined feeder path trace. This is an existing function in Dew. This feature is used to quickly identify components which are not operable, or have “lost service.” This can be used to speed up analysis since algorithms like power flow do not need to be run on circuits that do not have service. One of the contributions provided by the work presented in this dissertation is to extend loss of service analysis across different types of interdependent systems, see Figure 32.

As discussed earlier, major equipment can often be divided up into sub-assemblies. These sub-assembly groupings are often used to define work and staffing requirements, which in turn could be used to provide information for analysis.

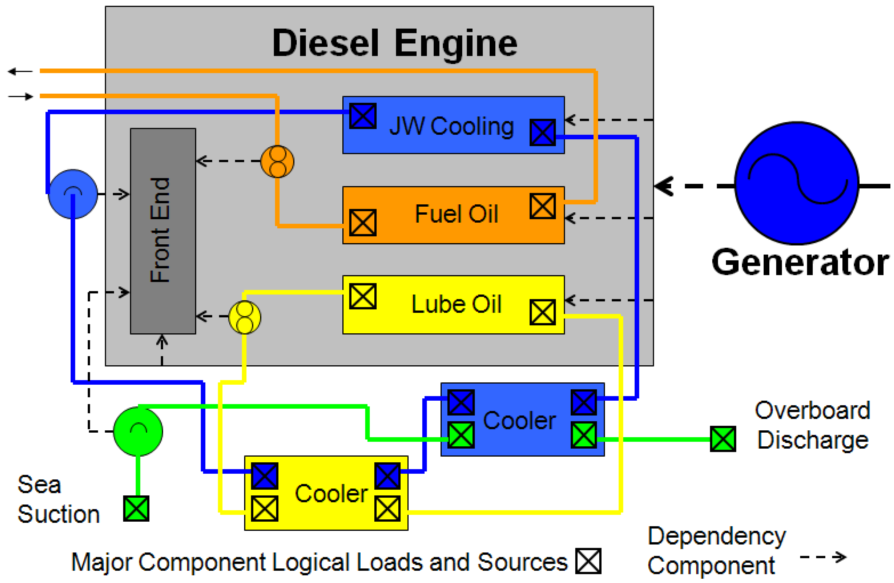

Figure 43 Dependency Component Model of Diesel Generator Set

Figure 43 shows how dependency components can be used to model interdependencies between sub-assemblies. Setting a sub-assembly's status to "fail" will automatically propagate service loss effects to the rest of the system. This concept could be used in future research to extend GTA based reconfiguration for fault isolation and restoration to include repair activities using the activity and event component definitions provided in the problem definition work presented in subchapter section 4.4. This in turn, would make it possible to structure emergency response and regular maintenance management using the same analysis and data. This could also be used to structure reliability analysis that uses current emergency response and maintenance status to generate a system's readiness state in terms of mission capability and reliability based calculations.

5.2.5 Recovery Analysis Measure Development

Performance, as it relates to military system survivability and commercial utility system resilience, involves a number of factors and includes evaluation of a large number of system operation and recovery states and behaviors. Naval ship design and operation readiness reports are structured on the use of defined mission performance levels, and the status of systems, services, and people needed to meet the requirements specified for each performance level (9) and (21). The following sections define how GTA system models and dependency components can be used to define interdependent system recoverability analysis using the standard mission readiness structures used for naval ship design and operation readiness reporting.

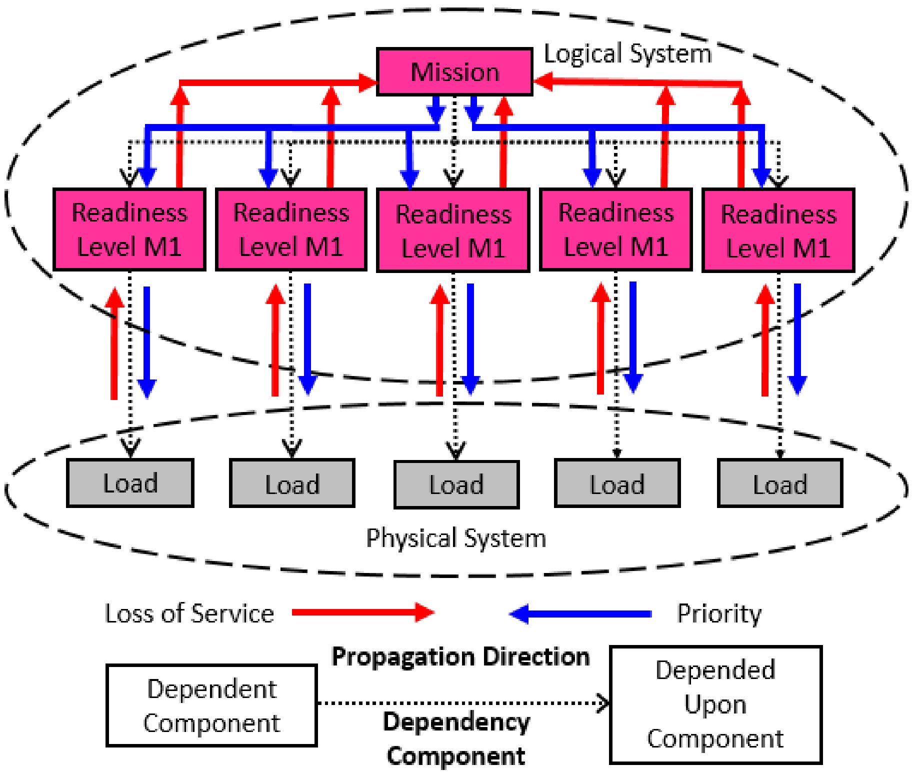

Figure 44 Readiness Level Logical Blocks added to Logical Model

5.2.5.1 Using Logical "Loads" to Model Mission Readiness Level

Additional readiness management analysis detail can be modeled by adding a set of readiness level blocks to the logical part of the system model. When added, these blocks become a traceable part of the overall system. Figure 44 provides a simplified illustration of this concept.

In Figure 45 a mission, a set of readiness level blocks (M1 – M5), system level logical load blocks, electrical loads, and dependency components are modeled together with a physical model of an electrical distribution systems using a modified version of the Dew software. The operation of any combination of circuit breakers shown in the physical part of the system results in loss of service propagation that causes service to be lost or restored to the mission readiness level blocks. For the convention used in the figure, the highest readiness level block that has service represents the readiness level for the system. In Figure 45, the mission readiness level resulting from opening circuit breaker CB 2, shown in red, is "M3."

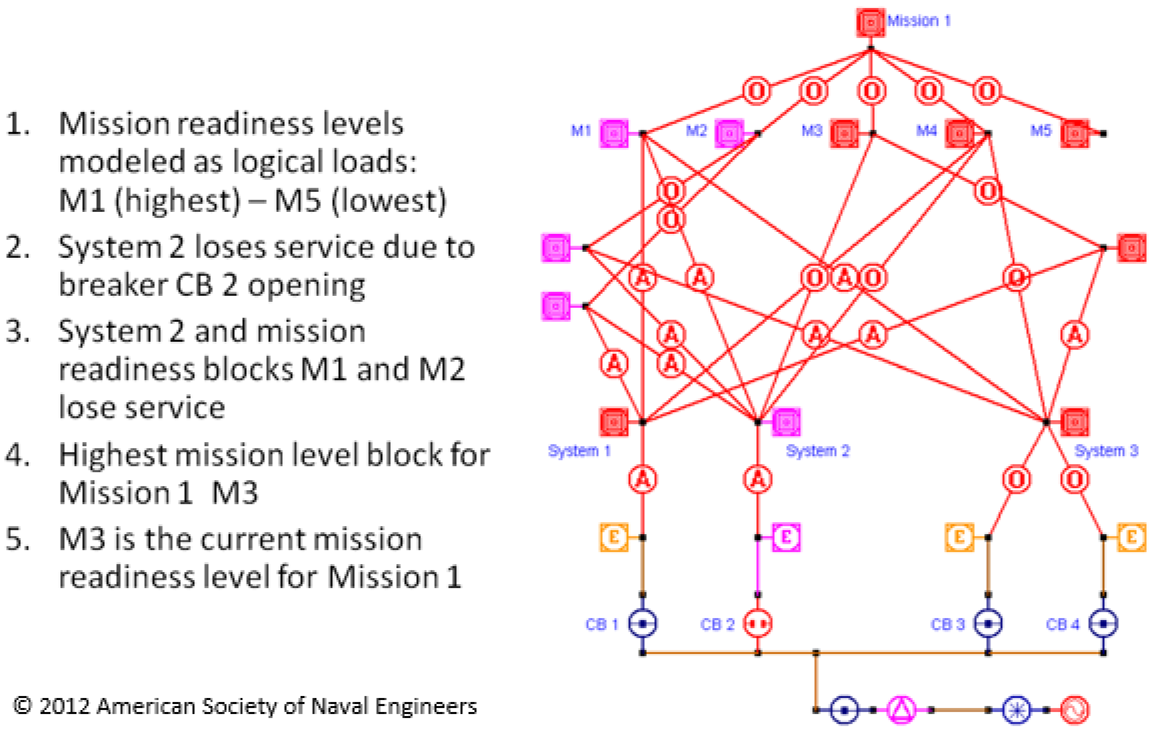

Figure 45 Mapping Mission to Mission Readiness using Dependency Components and Logical Mission Readiness Loads

When circuit breaker CB 2 is opened, electrical service supporting mission System 2 is lost. As a result, mission readiness level blocks M1 and level M2, which are dependent on System 2, also lose service. Mission level M3 and M4, which are related to Systems 1, 2, and 3 through a combination of "And" and "Or" dependencies still have service. M3 is a higher readiness level than M4. Using the convention defined for the model, M3 is then the current readiness level for the system. The mission capacity (M) readiness level combinations modeled in Figure 45 are as follows:

- M1 – Systems 1 & 2 & 3 in service
- M2 – Systems 1 & 2 or Systems 2 & 3 in service

- M3 – System 2 or Systems 1 & 3 in service
- M4 – System 1 or System 2 or System 3 in service
- M5 – No systems in service

5.2.5.2 Using Readiness Measures to Simplify Evaluation of Configuration Options

Potential exists for using component priority and loss of service system trace analysis to preprocess large numbers of potential restoration solutions, using service connectivity constraints and mission readiness measures to identify non-viable or lower interest solutions. Remaining solutions could then be evaluated in further detail using analysis which requires longer run times to perform. Extending GTA so that it can be used to automatically evaluate potential contingency states in terms of failure affects and mission priority is a contribution provided by the work performed for this dissertation.

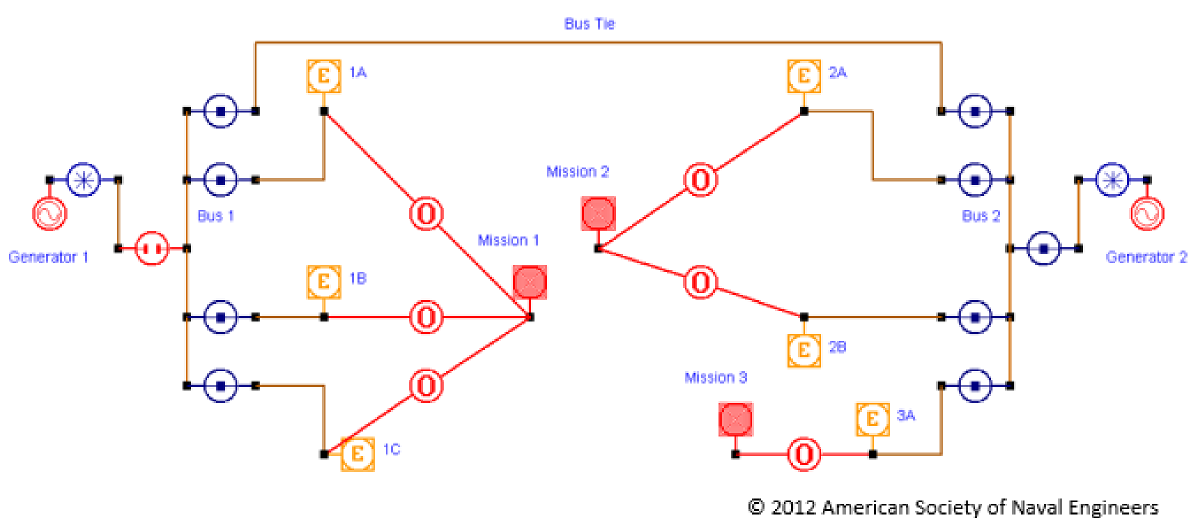

Figure 46 Three Mission Contingency Enumeration List Example System

For the example system shown in Figure 46, the number of bus and cable components that can fail is 17 if failure states for switches and loads are not included. The total number of damage states (in terms of components being failed or not failed) for a networked system is equal to 2^n, where n is the number of components in the system. For this example, this results in a total of 2^{17} = 131,072 different possible damage states.

If the system is broken up into segments, which are groups of components that share common isolation devices, the number of possible damage states reduces to 2^{11} = 2048. Note that GTA reconfiguration analysis uses automated traces to automatically identify and work with segments. The number of states that must be analyzed can be reduced further by filtering possible states according to their relation to major changes in readiness and current component operation status.

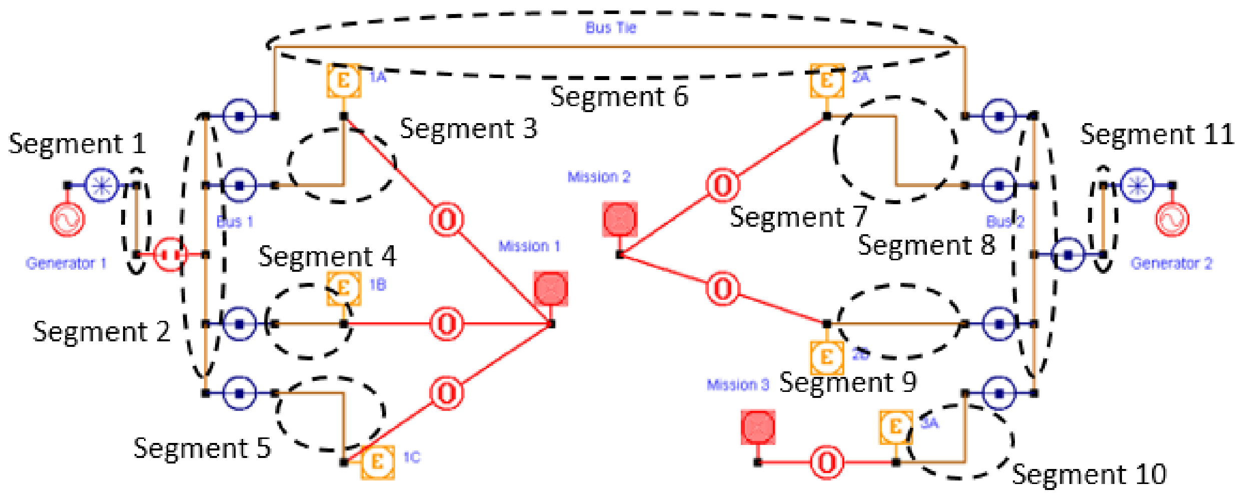

Figure 47 Three Mission Enumeration Example Divided into Segments

To further classify the 2048 segment damage states, all possible states for the example problem were enumerated using Microsoft Excel. These damage state combinations were mapped to show their effect on electrical load service status. Electrical loads shown as yellow blocks in Figure 47 are modeled as physical components in the electrical system part of the model. Electrical load states were mapped to the three mission logical load blocks, in terms of how their states affected mission readiness status as defined using the mapping shown in Table 19. Note that the mission readiness level blocks shown in the previous section in Figure 45 were not included in the screen shot of the Dew model shown in Figure 46 and Figure 47, but the dependency logic for readiness levels was included in the Excel example mapping shown in Table 19.

In the example, Missions are numbered 1 to 3. Dependency components are used to map missions to the different combinations of service loads required to support various mission readiness levels (numbered M1 through M5) for each of the three missions. In Table 19, M1 is defined as the highest readiness state. M5 is the lowest.

Table 19 Three Mission System Load Status to Mission Mapping

Mission	1A	1B	1C	2A	2B	3A	Mission Readiness Level
1	1	1	1				M1
1	0	1	1				M2
1	1	0	1				M2
1	1	1	0				M2
1	0	0	1				M3
1	1	0	0				M4
1	0	1	0				M4
1	0	0	0				M5
2				1	1		M1
2				1	0		M2
2				0	1		M3
2				0	0		M5
3						1	M1
3						0	M5

Load Status: 0 - Failed, 1 - Not Failed

The Excel spreadsheet for the system shown in Figure 46 was filtered to show all possible 2^n segment contingency failure states with load 1B already failed. This was done to show how the number of states needed to be analyzed for an operations problem could be significantly reduced if current damage state was included. See Figure 48.

Using the current damage state as the starting point for analysis reduces the number of contingency state combinations of interest from 2048 down to ten states that can then be used to evaluate the consequences of additional potential damage. If the contingency states are filtered according to their impact on mission readiness, which is one of the primary measures used by shipboard personnel for operations management, there are only two possible contingency states that will cause Mission 1 (M1 in the table) or Mission 2 (M2 in the table) to drop to level 3, one contingency state that will drop Mission M1 to readiness level 4, and three that will drop Missions M1 and M2, M2 and M3, or M3 down to readiness level M5.

Cont	Feeder Failure Status (1 - not failed 0 Failed)											Resulting Mission Load Status						Resulting Mission Readiness		
Level	Gen 1	Bus 1	Bus Tie	1A	1B	1C	2A	2B	3A	Bus 2	Gen 2	1A	1B	1C	2A	2B	3A	M1	M2	M3
2	1	1	1	1	0	1	1	1	1	1	0	1	0	1	1	1	1	2	1	1
2	1	1	1	1	0	1	1	1	1	0	1	1	0	1	0	0	0	2	5	5
2	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	2	1	5
2	1	1	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	2	2	1
2	1	1	1	1	0	1	0	1	1	1	1	1	0	1	0	1	1	2	3	1
2	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	4	1	1
2	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	3	1	1
2	1	1	0	1	0	1	1	1	1	1	1	1	0	1	1	1	1	2	1	1
2	1	0	1	1	0	1	1	1	1	1	1	0	0	0	1	1	1	5	1	1
2	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	2	1	1

Figure 48 Readiness Management Contingency State Enumeration

5.2.5.3 System Recovery Plots and Measures

The recovery plot shown in Figure 50 was generated using the example system shown in Figure 49. The system was analyzed using a GTA reconfiguration algorithm implemented in Dew. The plot is designed to conform to the readiness measure conventions used for naval ship design and operation reporting. This plot shows that GTA can be used to automatically generate integrated system readiness plots for interdependent systems that are a function of switchable device discrete operation times and set mission priorities. This capability is a contribution provided by the work performed for this dissertation.

The example system includes mission, mission system and mission readiness level logical load blocks for five missions. Dependencies are defined using dependency components placed between electrical and fluid system components, fluid and electrical system components and mission systems, and between missions. Loss of service and priority information propagation was performed across AC power, fluid, mission load, mission readiness and mission components as described in the previous sections.

In the example system, electrical components are fed from four separate generators which were modeled as constant voltage buses. Electrical system simulation was done using three phase AC power operating at 60 Hz and 480 volts. At the time that the example was run, the Dew software used to perform the simulation was not capable of running fluid and AC electrical system flow at the same time during the same analysis run. As a result, the example shown in Figure 50 included consideration of AC electrical voltage and overload constraints but not fluid pressure constraints. Fluid flow and mechanical power interconnection between fluid flow and power flow components were also not modeled. Fluid system modeling, and interdependent system modeling between fluid, electrical and logical component objects was limited to simulating loss of service effects and mission priority relationship affects. Full implementation of interdependent system physical flow using Dew (AC and DC, fluid, mechanical, thermal, gas, etc.) remains for future work.

The fluid system in the example has two pumps which receive power from the electrical system through Automatic Bus Transfer (ABT) switches. The ABT's are set to automatically switch from primary to secondary power feed when primary power is lost. Each mission is supported by a combination of mission system logical loads that in turn are supported by electrical and cooling system physical component loads. Mission systems could be modeled physically using additional components if more detail is required. The model also contains two loop fed loads that are supplied by cables that overload if both are not supplying power to their connected load. The loop fed loads were included for algorithm testing and are not typical for ship systems.

At each major time step the GTA reconfiguration application sets switchable devices to "operable" or "non-operable" depending on each device's individual "operation time" setting. If a switches' operation time setting is less than or equal to the current total elapsed time, the reconfiguration algorithm treats it as operable and includes it in the list of devices that can be used to reconfigure the system. For the example used to generate the plot shown in Figure 50, operation times for the circuit breakers for each generator were set to 0.3, 0.5, 0.7 and 0.9 hours respectively, and only the circuit breaker with its time set at 0.3 hours was initially set to closed (on). The other generator circuit breakers were initially set to open (off). This essentially modeled the second, third and fourth generators as being offline in standby at the start of the simulation. At each major time point, an additional generator was added to supply the system by the algorithm according to the standby time set for each generators main circuit breaker. Operation times for all of the other switches and valves used in the example were set to zero, which meant that they were operable throughout the simulation.

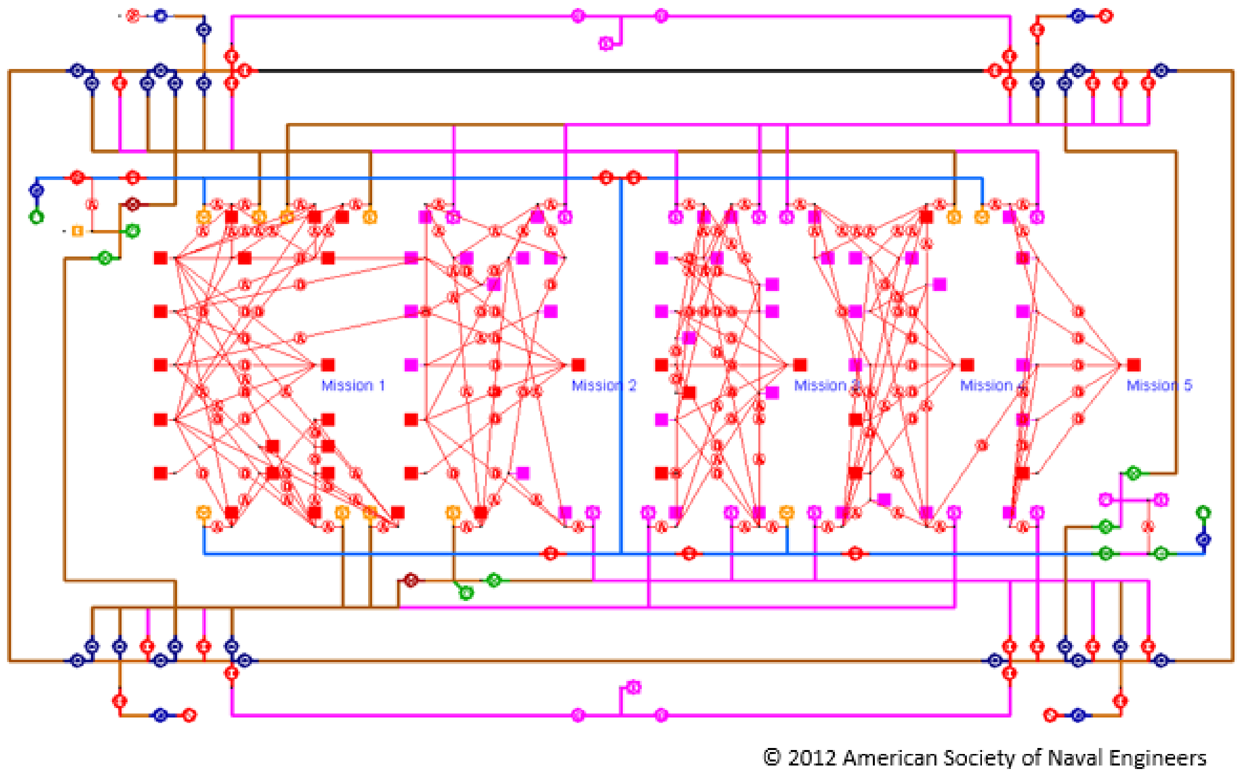

Figure 49 Recovery Analysis Example Result with Elapsed Time Set to 0.3 Hours

In real systems, recovery time is generally dominated by the time required for personnel to operate switches and valves, bring major systems online from some assigned standby status, and to make emergency repairs. These resource and time operation constraints can be modeled by assigning "time to on" and "time to off" operation time attribute data to applicable components.

The bus tie highlighted in black in Figure 49 was set as failed at the beginning of analysis. System loading was set so that loss of the failed bus tie would cause overloading in other cables and switches, depending on which generators provide power. Figure 49 shows reconfiguration loss of service, switch state and valve state results with elapsed time set to 0.3 hours and priorities for missions set to Mission 1 – priority 5, Mission 2 – priority 4, Mission 3 – priority 3, Mission 4 – priority 2, and Mission 5 – priority 1. For this example, priority 5 was treated as the highest priority, and priority 1 was the lowest. Time-dependent readiness levels for the system with given loading and mission priorities are plotted in Figure 50.

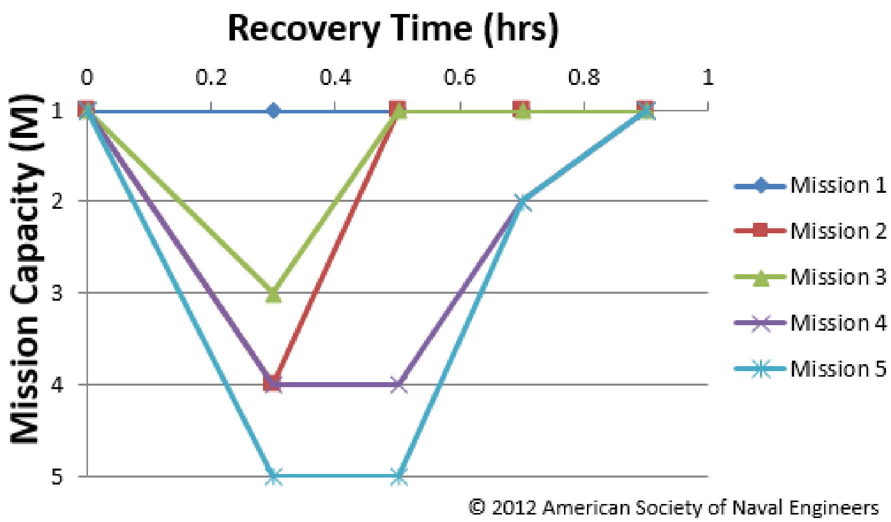

Figure 50 Integrated System Recovery Plot

Readiness levels at each time step were determined using an automated system trace used to find the readiness measure logical load with the highest "M" level connected to each mission through dependencies that have not lost service. Readiness measure logical load status is driven by dependency connection of mission readiness level measure blocks to mission system logical loads. Mission system logical load status is in turn driven by dependency connections to fluid and electrical load components. The plot starts with all missions at readiness level M1, which is full capability. The initial damage occurs at time 0.3, which drops Mission 1 to M5, Missions 2 and 4 to M4, Mission 3 to M3 and Mission 1 is not affected. At time point 0.5 hours, switchable devices with component level specified operation times that are less than or equal to 0.5 hours are operated by the reconfiguration algorithm to further isolate damage and restore services. This results in Mission 2 and Mission 3 being restored to M1 (fully capable) and Mission 4 and Mission 5 remaining at readiness levels M4 and M5 respectively. At time point 0.7 hours, additional devices are operated according to their specified individual operation time constraints and Mission 4 and Mission 5 are restored to readiness M2. At time point 0.9 hours, Mission 4 and Mission 5 are restored to full capability.

5.3 Utility Power System Reconfiguration Analysis and Supervisory Level Control

The shipboard system reconfiguration for fault isolation and recovery concepts developed for the research presented in this dissertation were applied to smart-grid reconfiguration control development at Orange and Rockland Utilities (ORU) (37). The fault analysis and recovery phases (Initial Action, remedial Action and Recovery) defined in Table 1 in Chapter 4 were used as a framework for defining control and design analysis process steps for this work, and the concepts defined in subchapter section 5.2 were used to support reconfiguration supervisory control algorithm development.

For smart-grid automated reconfiguration control developed for ORU, a software-based controller that is interfaced to SCADA uses a GTA distribution system model to run fault isolation and restoration which is then used to generate isolation and restoration switch operation commands. Those commands are forward by the GTA controller to their respective field devices through interface to SCADA. The controller maintains a copy of the system model which it keeps synchronized with the real system using start of circuit measurements and switch operation signals it receives from SCADA, and manual switch operation updates it receives through an interface with a Graphical Information System (GIS).

The protection device scheme used with the reconfiguration controller uses standard electronic reclosers which are responsible for initial fault isolation and restoration of service. The reclosers are augmented by the addition of SCADA operated switches with fault indication sensors that can be controlled remotely by operation center personnel or automatically by the reconfiguration controller. This strategy which combines the use of reclosers with SCADA operated switches was developed by ORU to reduce the number of high cost devices needed to automate reconfiguration, to provide for incremental installation and to structure a tiered approach that uses conventional protection device coordination to provide primary protection and more complex smart-grid automated control to refine isolation and improve restoration response.

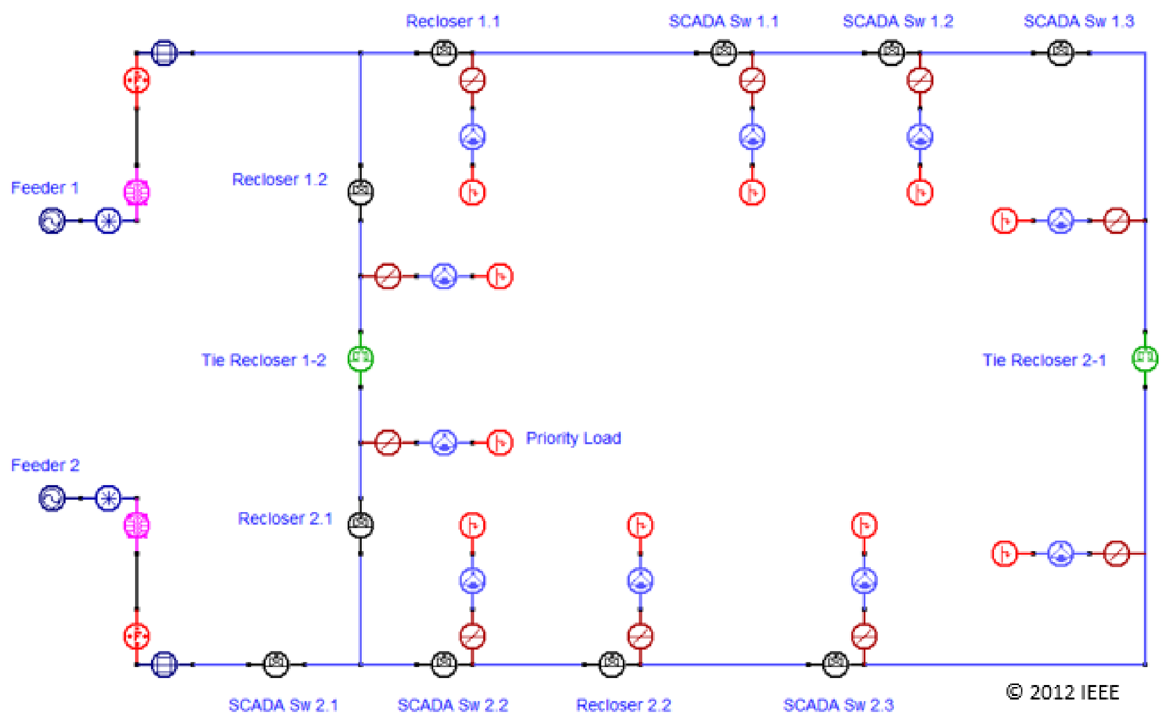

Figure 51 Reconfiguration Control Development Test System Model

The reclosers and SCADA switches used in the system are equipped with fault indicators which report through SCADA to the reconfiguration controller. The reconfiguration algorithm (36) uses fault indications to rapidly identify the segment where the fault is located. The algorithm then generates a list of switch operations which the controller uses to isolate the fault to the smallest section possible using available SCADA operable switches, reclosers, and breakers. The algorithm uses power flow to check restoration actions against component operation constraints using circuit measurements and historical load model data attached to each load component in the model. The controller takes appropriate actions to block tie-recloser operation and to reduce the number of loads to be restored if needed where not doing so would result in a voltage or current constraint violation. Figure 51 and Figure 52 illustrate reconfiguration results for isolation of a fault on a primary feed. Figure 51 shows the example system before the fault.

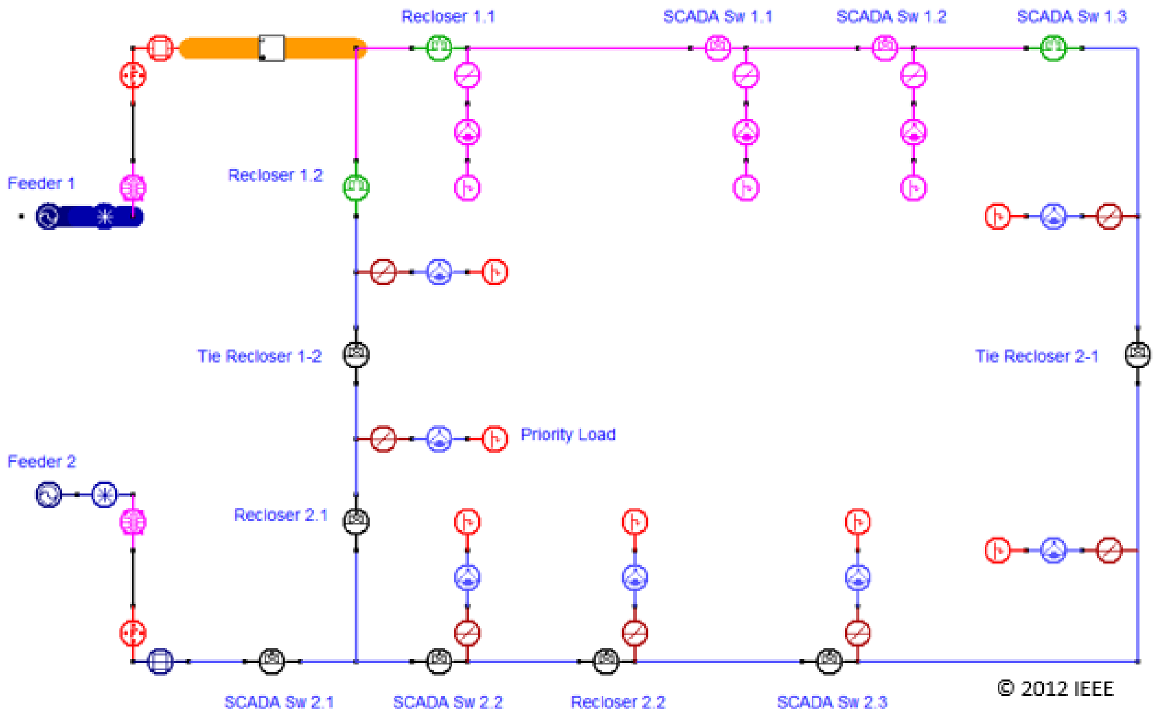

Figure 52 Maximum Number of Customers Restored with No Overloads or Voltage Violations

Figure 52 shows the system after reconfiguration analysis has been performed. The faulted line component is highlighted in orange. Segments that have lost power are shown in pink. Upon receiving a fault trip signal from the circuit breaker providing protection for Feeder 1, and noting a lack of fault indications from Recloser 1.1 and 1.2, the reconfiguration algorithm opens Reclosers 1.1 and 1.2 to isolate the fault down to the smallest section possible. Before Tie-Recloser 2-1 times out and closes according to its device level programming, the algorithm determines that allowing the tie recloser to operate normally would result in overloading SCADA Switch 2.1. The algorithm blocks open SCADA Switch 1.3 to prevent overload of Switch 2.1 and then closes Tie-Recloser 2-1.

The same reconfiguration algorithm developed for the work discussed in references (36) and (38) was also used together with the emergency management phase definitions from Table 1, to develop measures for utility system reconfiguration analysis. The setup dialog and a sample output measures generated using the model shown in Figure 52, are shown in Figure 53 and Figure 54.

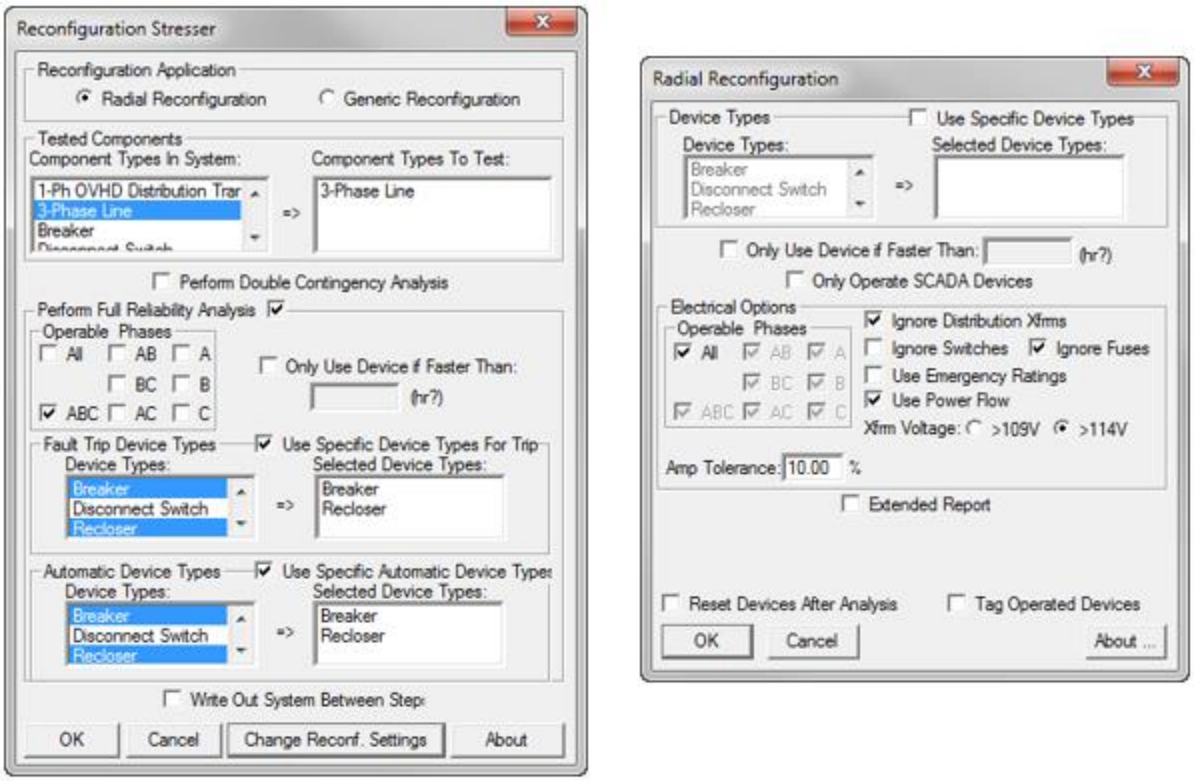

Figure 53 Reconfiguration Contingency Analysis Setup Dialog

Stage	TotalLoads	Dropped By Failure	Restored	With Service	Without Service	Isolating SwtUIDs	Blocked Open Swt UIDs	Closed UIDs	Opened UIDs	Switches Causing Violations
Fault Trip	9	4	0	5	4	FB 1|				
Automatic Device Reconfiguration	9	4	3	8	1	SS 1.1|R 2|	FB 1|	TR 2|TR 1|		
Manual Device Reconfiguration	9	1	0	8	1	Load 1.3|	R 2|SS 1.1|			
Failures Restored	9	0	1	9	0			FB 1|SS 1.1|Loac	TR 2|TR 1|	
Fault Trip	9	0	0	9	0	R 2|				
Automatic Device Reconfiguration	9	0	0	9	0		R 2|			
Manual Device Reconfiguration	9	0	0	9	0		R 2|			
Failures Restored	9	0	0	9	0			R 2|		
Fault Trip	9	4	0	5	4	FB 1|				
Automatic Device Reconfiguration	9	4	3	8	1	SS 1.1|R 2|	FB 1|	TR 2|TR 1|		
Manual Device Reconfiguration	9	1	0	8	1	Load 1.3|	R 2|SS 1.1|			
Failures Restored	9	0	1	9	0			FB 1|SS 1.1|Loac	TR 2|TR 1|	

Figure 54 Reconfiguration Results for Example Smart-Grid System

Chapter 6 Conclusions and Further Research

6.1 Conclusion

The research presented in this dissertation combines concepts from OOA&D, Physical Network and Physical System Modeling, UML and GTA to develop model-based interdependent infrastructure system analysis that structures physical system components, dependencies between systems, missions, and mission readiness levels as a network of interdependent components. The work presents detailed problem definition based on multidiscipline observation and review of naval ship system design, engineering administration, emergency operation doctrine, and standard operation management practice and then uses that to structure the new analysis concepts presented in this work. These concepts are presented using simple models and examples which are also used for evaluation. Problem definition and example results, combined with supporting work referenced in this dissertation demonstrates how these concepts can be used in current and future interdependent critical infrastructure system operations management analysis.

The work also presents initial experimentation with new loop flow convergence measures and acceleration factors that were developed using GTA trace analysis functionality combined with concepts from past fluid flow analysis research. Being able to solve networked system loop flows quickly and accurately is critical to the development of GTA as a viable approach for ship and shore utility system emergency management modeling and analysis. Results for the cotree flow solution methods evaluated shows reasonable potential and warrants further investigation.

6.2 Contributions

The work presented in this dissertation makes several contributions towards the development of a unified approach for interdependent critical infrastructure system control and operations management analysis. Specific contributions include definition and evaluation of:

- Dependency component power, loss of service and mission priority information propagation which make it possible to structure interdependent critical infrastructure systems analysis using a forward-backward sweep method to manage component status information, simulate discrete event component reactive behavior, and perform physics based flow analysis

- Automated system trace-based propagation of operation goals, criteria, and constraints that can be used to define analysis related relationships between components that are modeled as part of integrated GTA logical and physical network models
- Recoverability measures that fit well with military system design, mission capability requirement definitions, and operational readiness reporting
- Mission priority and readiness time requirement constraints to structure time-based discrete event operations management analysis
- Development of interactive system traces which replace the need for the use of complex cost functions for evaluating system reconfiguration options
- Cotree scaling measures and acceleration factors to improve GTA looped power and fluid flow analysis convergence

6.3 Further Research

The main goal of this work was to define new concepts needed to extend the application of Graph Trace Analysis (GTA) for use in automating multi-domain, integrated critical infrastructure system analysis. The contributions made under this work were tested in different combinations and levels in coordination with other GTA commercial and government funded research. Existing GTA power flow and loss of service propagation used in this work is currently being used commercially to model distribution and transmission systems, the largest one containing over 3,000,000 components and 3000 multiphase loops. GTA has also been used to model large urban low voltage distribution networks, integrated transmission and distribution networks, and switched radial smart-grid distribution. This includes a three phase, unbalanced 120,000 component model of Staten Island that contains loops, which solves in approximately 15 seconds using a standard PC.

For radially operated systems, existing GTA power flow and loss of service propagation, and reconfiguration analysis (which was developed under separate work in coordination with the concept development presented in this dissertation), has been shown to operate fast enough to support real-time supervisory level control and operation management of commercial distribution (37). The priority propagation and logical system based mission readiness level analysis presented in this work is implemented using software functionality that is very similar to loss of service propagation that is currently used in large commercial power system models, so it is reasonable to assume that priority and logical system modeling should also scale for use with much larger systems.

GTA has also been used to develop fluid system modeling that includes flow, temperature, and chemical mixing and reaction modeling as part of military utility system hazard management modeling sponsored by the Army Corp of Engineers (25). This work was coordinated with the work presented in this dissertation. The one area that stood out from this work that is also sometimes a factor in existing GTA power utility network analysis is that network system convergence and analysis speed can vary significantly depending on the number and location of loops modeled in the system. In order to ensure consistent speed and convergence performance for use in operations management and supervisory level control, additional work in this area needs to be done. The cotree convergence work presented in subchapter section 5.1 was performed to provide initial work for addressing this problem. The next step for this development should be to test and refine GTA loop analysis measures and scaling, and compare it to a Todini and Pilati (30) gradient method solution, using large commercial system models. Both of these approaches are well suited for use with switching and discrete event related information propagation. It is expected that the main tradeoff will model size and matrix inversion time requirements between GTA method convergence and speed which does not require matrix inversion if used with a Hardy Cross type solution method, and the Todini gradient method which requires matrix inversion.

References

1. **Booch, G., et al.** *Object-Oriented Analysis and Design with Applications.* Third Edition. Addison-Wesley : s.n., 2007.

2. *Performance Metrics for Electric Warship Integrated Engineering Plant Battle Damage Response.* **Cramer, Aaron M., Sudhoff, Scott D. and Zivi, Edwin L.** 1, January 2011, IEEE Transactions on Aerospace and Electronic Systems, Vol. 47, pp. 643-646.

3. *A Linear Programming Approach to Shipboard Electrical System Modeling.* **Chan, Ricky R., et al.** 2009. IEEE Electric Ship Technologies Symposium. pp. 261-269.

4. *Shipboard Electrical Power Quality of Service," IEEE Electric Ship Technologies.* **Doerry, N.H., D.H. Clayton.** s.l. : IEEE Electric Ship Technologies, 2007. pp. 274-279.

5. *Strategies for integrating models of interdependent.* **Weston, N.R., M.G. Balchanos, M.R. Koepp, D.N. Marvis.** s.l. : IEEE Proceedings of the 38th Southeastern, 2006.

6. *Optimizing Investment for Recovery in Interdependent.* **Xu, N., L.K. Nozick, and M.A. Turnquist.** s.l. : IEEE 40th Annual Hawaii International Conference on System Sciences, HICSS 2007, 2007.

7. *Modeling Approaches for Large-Scale Reconfigurable Engineering Systems.* **Tam, K.S.** s.l. : World Academy of, 2006. Vol. 17.

8. *Graph Trace Analysis and Generic Algorithms for Interdependent Reconfigurable System Design and Control.* **Feinnauer, L., Russell, K. and Broadwater, R.** 1, s.l. : American Society of Naval Engineers, 2008, Naval Engineers Journal, Vol. 120, pp. 29-40.

9. *Graph Trace Analysis Based Interdependent System Recoverability Analysis Criteria, Constraints and Measures.* **Russell, Kevin and Broadwater, Robert.** 3, s.l. : Naval Engineers Journal, 2012, Vol. 124, pp. 49-58.

10. **Steven, P. and Pooley, R.** *Using UML: Software Engineering with Objects and Components.* s.l. : Addison-Wesley, 2006.

11. **Blackwell, W. A.** *Mathematical Modeling of Physical Networks.* New York : The MacMillian Company, 1968.

12. **Wellstead, P. E.** *Introduction to Physical System Modeling.* s.l. : Academic Press, 1979.

13. **Friedenthal, S., Moore, A. and Steiner, R.** *A Practical Guide to SysML.* s.l. : Morgan Kaufmann and Object Management Group, 2008.

14. **Harvey, M.** *Essential Business Process Modeling.* s.l. : O'Reily Media, Inc., 2005.

15. **Musser, D. R. and Saini, A.** *STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library.* s.l. : Addison-Wesley, 1996.

16. **Karnopp, D., Margolis, D. and Rosenburg, R.** *System Dynamics, Modeling and Simulation of Mechatronic Systems.* Fourth. Hoboken : John Wiley & Sons, 2006.

17. *Pointers and Linked List In Electric Power Distribution Circuit Analysis.* **Broadwater, R., Thompson, T. and McDermott, T.** Baltimore : Proceedings of 1991 IEEE PICA Conference, 1991.

18. **Feinauer, L.** Generic Algorithms for Analysis of Interdependent Multi-Domain Distributed Network Systems. *Doctoral Dissertation.* Blacksburg, VA : Virginia Tech, October 2009.

19. **Russell, K., et al.** *Recovery Algorithms for Ship Survivability SBIR N04-134 Phase II Final Report, Contract N00014-05-C-162.* s.l. : Navy Personnel Research, Science and Technology Division, Office of Naval Research, 2007.

20. *Generic Analysis Based Integrated System Problem Decomposition and High-Level Architecture Design.* **Russell, K. and Broadwater, R.** s.l. : ASNE Automation and Controls (ACS) Proceedings, 2007.

21. *Rethinking Survivability Automation from a Reduced Manning Reconfiguration Centered Point of View.* **Russell, K. and Broadwater, R.** s.l. : ASNE EMTS 2004 Naval Symposium, 2005.

22. *Physical Model-Based, Iterator Driven, Integrated Design, Control and Information Management.* **Russell, K., Broadwater, R. and Rapp, J.** s.l. : Proceedings of the ASNE EMTS 2004 Naval Symposium, 2004.

23. **Elis, M.V.** The Ladder Load-Flow Method Extended to Distribution Networks. *Doctoral Thesis.* 1994.

24. **Sugg, A,, Seguin, R. and Scirbona, C.** Three Utilities Implement Packaged Distribution Systems. *Transmission & Distribution World.* September 1, 2002.

25. **Russell, K., et al.** *Collaborative Hazard Management, Contract W9132T-06-C-0038.* s.l. : U.S. Army Corps ofEngineers, Engineer Research Development Center, U.S. Army Construction Engineering Research Laboratory, 2009.

26. **Russell, Kevin and Broadwater, Robert.** *Predictive Simulation of Contaminants in Potable Water Systems, .* s.l. : Army Corps of Engineers, Engineer Research and Development Center, U.S. Army Construction Engineering Research Laboratory, 2010.

27. *A Robust Multiphase Power Flow for General Distribution Networks.* **Dilek, M., et al.** 2, s.l. : IEEE Transactions on Power Systems, 2101, Vol. 25.

28. *Enhancement of Convergence of Pipe Network Solutions.* **Williams, G.** 7, s.l. : Journal of the Hydraulic Division, Vol. 99, pp. 1057-1067.

29. **Boulos, P., Lansey, K. and Karney, B.** *Comprehensive Water Distribution Systems Analysis Handbook for Engineers and Planners.* Second. s.l. : MWH Soft, 2006.

30. **Todini, E.** *Towards Realistic Extended Period Simulations (EPS) in Looped Pipe Network.* Cincinnati : 8th Annual Water Distribution Systems Analysis Symposium, 2006.

31. *Efficient Code for Steady-State Flows in Networks.* **Epp, R. and Fowler, A.** 1, s.l. : Journal of the Hydraulics Division, ASCE, 1970, Vol. 96, pp. 43-56.

32. *Unified Framework for Deriving Simultaneous Equation Algorithms for Water Distribution Networks.* **Todoni, E. and Rossman, A.** 5, s.l. : ASCE, 2013, Journal of Hydraulic Engineering, Vol. 139, pp. 511-526.

33. *Probability-based predictive self-healing reconfiguration for shipboard power systems.* **Srivastava, S. K. and Butler-Purry, K. L.** 3, s.l. : Institute of Engineering Technology, 2007, IET Generation, Transmission & Distribution, Vol. 1, pp. 405-413.

34. **Chong, E. K. and Zak, S. H.** *An Introduction to Optimization.* 2nd. s.l. : Wiley, 2001.

35. *Rethinking Reduced Manning Design and Optimization Using a Modified Systems Approach.* **Russell, K.** s.l. : Proceedings of the 2000 Engineering the Total Ship Symposium, ASNE, 2000.

36. **Kleppinger, D.** Prioritized Reconfiguration of Interdependent Critical Infrastructure Systems. *Doctoral Dissertation.* Blacksburg, VA : Virginia Tech, March 2009.

37. *Model-Based Automated Reconfiguration for Fault Isolation and Restoration.* **Russell, K. and Broadwater, R.** s.l. : IEEE PES, 2012, Innovative Smart Grid Technologies (ISGT), pp. 1-4.

38. *Graph Trace Analysis Based Shipboard HM&E System Priority Management and Recovery Analysis.* **Kleppinger, D., Russell, K. and Broadwater, R.** May 2007, IEEE Electric Ship Technologies Symposium, pp. 109-114.

39. **McLaughlin, B., Pollice, G. and West, D.** *Head First, Object-Oriented Analysis and Design.* s.l. : Oreilly Media, 2007.

40. **Modelica Association.** Modelica Tutorial Version 1.4. [Online] December 15, 2000. www.modelica.org/documents/ModelicaTutorial14.pdf.

41. **Brantley, Mark W., McFaddden, Willie J. and Davis, Mark J.** *Expanding the Trade Space: An Analysis of Requirements Tradeoffs Affecting System Design.* New York : U.S. Military Academy, West Point, 2002.

42. **Karnopp, Dean C., Margolis, Donald L. and Rosenburg, Ronald C.** *System Dynamics - Modeling and Simulation of Mechatronic Systems.* Fourth. Hoboken, New Jersey : John Wiley & Sons, 2006.

43. *Topology Model Based Shipboard System Reconfiguration, Hierarchical Control and Integrated Information Management.* **Russell, K. and Broadwater, R.** s.l. : ASNE Day 2001 Proceedings, ASNE, 2001.

44. **Zhu, Jizhong.** *Optimization of Power System Operation, The Institute of Electrical and Electronic Engineers, Inc.* s.l. : John Wiley & Sons Inc., 2015.