

# An Overview of Digital Engineering Methods for Platform Integration of Power and Energy Systems

Robert M. Ames<sup>1</sup>, Dr. Norbert H. Doerry<sup>1</sup>, Madeleine M. Koerner<sup>1</sup> and Dr. Mark A. Parsons<sup>2,\*</sup>

## ABSTRACT

*US Navy ships, and combatant ships in particular, have requirements for integrated systems that are designed and configured for operational efficiency, redundancy, and survivability. Mission systems today and in the future will not always come with their own energy and many may, at times, require extreme pulse power loads. In addition, the migration away from fossil fuels to hybrid systems with energy storage, or the requirements for autonomous platforms, is challenging our ability to design platforms for these systems. Understanding the interdependencies between components, the systems they support, and the energy domains they are member of is a critical ontological design requirement. This paper will address how we applied digital engineering principles and computer science to the design of ontologies that allow for the modeling of complex operational systems riding on the same shipboard energy network. This system network must support varying levels of detail needed during design while at the same time understanding the impacts of that design on ship system operations and their energy loads over time.*

## KEY WORDS

Design, Power, Energy, Ontology, System

## DEFINITIONS

**Capability** - The ability to achieve a desired effect under specified performance standards and conditions through combinations of ways and means [activities and resources] to perform a set of activities. Capabilities are only described in the abstract and does not specify how a capability is to be implemented. A Capability is structured as a hierarchy of Capabilities, with the most general at the root and most specific at the leaves. At the leaf-level, Capabilities may have a measure specified, along with an environmental condition for the measure.

**Component** – A Component class represents objects that can be thought of as parts of a Concept. They may be connected in ways that provide function and capability. Components can be thought of as any part of a ship that represents something physical and as discrete as a radar or engine, or as abstract as proxy for something temporal or needing further refinement. A Component is in many ways just a small Concept, but it is modeled as something a Concept owns. A Component does not have their own Systems, but Components do have most of everything a Concept does.

**Concept** - A class that defines the attributes, the Systems, the Components and the views of geometry that represents a product in design or under study. It is generic to all things engineered, but is used here to define platforms such as a ship, submarine, aircraft, or other engineered product.

**Connection** - The Connection class provides information on how the contained Component, System, Node, and Connection objects are connected (i). Connections are logical relationship between connection members (Component, System, Node, and other Connections).

**Diagram** - The Diagram class provides the schematic information that defines a connected graph or network. Diagrams can define a connected network between Components, within a Component, and a Connection between Components and Concept structure.

**Domain Network**– A Domain Network is a domain specific subset of the total Energy Network. Domain Networks facilitate the distribution of energy specific to its domain type. Domain types include, but are not limited to, electric, fluid, thermal, chemical, magnetic, structural, and mechanical. Each Domain Network is capable of energy transfer (work) within the Energy Network. Components may participate in zero or more Domain Networks. A Component can be both a Domain Source and a Domain Sink (e.g. battery). The Domain Network does not formalize energy flow, it only defines its possible boundaries.

---

<sup>1</sup> Naval Architecture and Engineering, Naval Surface Warfare Center Carderock, Bethesda, MD, USA

<sup>2</sup> Surface Ship Design and Systems Engineering, Naval Sea Systems Command, Washington DC, USA

**Domain Sink** – A Domain Sink is defined as a domain terminus. The Domain Sink is the last Component specific to a domain that is exclusive of any energy process outside of the sink Component. If the domain is electric, then a radar, light, or motor could be examples of Domain Sinks.

**Domain Source** – A Domain Source is defined as the source of energy specific to a particular domain. The Domain Source is the first Component specific to that domain that is exclusive of any energy transfer process outside of the Component. For example, if the domain is electric then a generator, solar panel, or fuel cell may be the Domain Source.

**Energy Network** – A Diagram defining an interconnected network of Components and Energy Resources that facilitate the distribution of energy throughout the Concept. Energy Networks are bounded by Primary Energy Resources at one end and the Natural Environment on the other. Within the network, energy is transferred from sources (Primary Energy Network) to sinks (Operational Energy Sink). Energy Networks are defined as Connections between Concept and Component via their respective Connections.

**Energy Resource** – An Energy Resource is something that can produce electricity, move objects, generate heat, and power and sustain life. Most shipboard energy today is matter stored energy from the Natural Environment. Forms of Energy Resources include, but are not limited to, fossil fuel, nuclear (fusion or solar, fission), energy stored in chemical bonds used in Components like fuel cells, and biomass.

**Manmade Energy Resource** – Manmade Energy Resources are Components that exist by design as sources of energy. They are not natural. Examples include, but are not limited to, batteries, capacitors, and flywheels. Manmade Energy Resources can be either primary or secondary resources depending on their configuration in the Energy Network and their intended use.

**Natural Energy Resource** – Natural Energy Resources are resources used in the conversion of energy and are extracted from the environment. They may be stored on the platform (e.g. fuel) or extracted from the environment during operation (e.g. solar). They include, but are not limited to, fossil fuels, wind, solar, natural gas, oxygen, uranium, and hydrogen.

**Natural Energy Resource Storage** - Some Natural Energy Resources are stored and consumed by the Concept (e.g. fuel, natural gas, uranium, hydrogen). This does not include Natural Energy Resources that are available through the environment (e.g. oxygen, wind, solar). In some cases, this storage exists within a Component and other cases it is stored in a compartment of space of a vessel.

**Natural Environment** – The Natural Environment is everything that is not owned by the Concept and is not an engineered physical thing. In contrast to the Natural Environment is the build environment. For a vessel operating in the real world, the Natural Environment includes everything that is in nature. This includes people. All elements of the Natural Environment are temporal. They can pass through and around Systems and Components within a Concept (e.g. crew, water, air, light, etc.) and can be stored, consumed, and replenished (e.g. potable water, fossil fuel, etc.) Note: Chemical and biological engineering cross the boundary between natural and man-made. Fossil fuel is a good example. Within this ontology, chemically engineered products like fuel or the generation of potable water from seawater are considered part of the Natural Environment.

**Node** - The Node class represents a location in space either relative to the Concept or relative to a Component. Nodes can be defined in Cartesian space, or they can be defined as a location in relative space (meaning, a Node can be defined as a location on a surface or curve of another object). The surface or curve can be owned by a Concept or a Component.

**Non-Operational Energy Sink** – A Non-Operational Energy Sink is like an Operational Energy Sink but it does not provide an Operational Capability. Non-Operational Energy Sinks include, but are not limited to, Components that reject significant amounts of energy like waste heat to the environment. An exhaust uptake is a good example. Since most Components transfer heat into the environment, Non-Operational Energy Sinks are differentiated by the need to network certain Components to the Energy Network for design reasons and simulate the transfer of energy to the environment through these Components. They are significant to the ship design and are therefore significant to the definition of the Energy Network. In theory, all Components store or transmit energy in some capacity and can participate as passive energy emitters into the Natural Environment (e.g. a Component's thermal energy emitted to a compartment). For purposes in modeling efficiency, not all non-operational energy emitters need to be designated as a Non-Operational Energy Sink. Only the ones that have significant impact at that stage of design.

**Operational Capability** - An Operational Capability is a Capability of a System to perform a mission, operation, or function in the intended environment. An Operational Capability is a specific type of Capability.

Operational Energy Sink – An Operational Energy Sink is a Component that transforms' energy for the purpose of providing one or more Operational Capabilities. One or more Operational Sink Components can have direct or indirect relationships with Operational Capabilities through System Function Actions. An Operational Sink can be as complex as a weapon or sensor, or as simple as a light fixture. For example, a radar is an Operational Energy Sink that provides detection through the conversion of electrical and thermal energy into heat and radio waves. This energy is transmitted off the platform and is therefore considered an energy sink. A multi-function radar can also provide communications and Electronic Warfare (EW) functions. Because this Component is multi-functional, it has multiple System Function Actions which provide multiple Operational Capabilities. It is the System Function Action that provides the functions and properties that describe and define how the Operational Energy Sink(s) behave as needed to provide an Operational Capability.

Primary Energy Resources – Primary Energy Resources are the sole originating source of energy for the platform. Primary Energy Resources may be a Natural Energy Resource or a Manmade Energy Resource.

Secondary Energy Resources - Secondary Energy Resources are Manmade Energy Resources that provide energy to the Energy Network but are replenished from some Primary Energy Resource. Examples include, but are not limited to, an electric battery, flywheel, or capacitor Components.

System – A collection of connected Components and Energy Resources that support one or more Operational Capabilities. Systems are defined as the aggregation of all Components that are involved in providing energy to Operational Energy Sinks. Systems are bounded at one end by Operational and Non-Operational Energy Sink Components and all Primary Energy Resources necessary for the System to function at the other end. Systems interact and operate along Energy Networks and may involve one or more Energy Domains.

System Domain: A System Domain is defined as all the Components of a System in a Domain Network for a particular System. It is a subset of the Energy Domain specific to a System. The electric Energy Domain of a propulsion System is an example. The mechanical Energy Domain for the same System is another.

System Information Networks: A System Information Network communicates data for the purpose of System monitoring and control. This network may be a subset of a larger network that serves additional capabilities outside of System control. An Information Network includes all the Components necessary to generate information and move that information from the information sources to the information sinks. In many cases, information sources and information sinks could be associated with the same component – (an HMI, engine order telegraph, telephone for example); or they could only be an information source (a sensor) or an information sink (an actuator or display). The System Information Network influences how the Components of the Energy Network operate, and thus should be modeled to some degree. In some cases, the System Information Network includes properties of the energy – voltage level at the bus for reactive power sharing .... Frequency for real power sharing, I2t for circuit protection coordination, etc.) Connections within System Information Networks are logical and from a modeling perspective do not include the transfer of energy; the law of energy conservation does not apply.

System Function Action - A System Function Action defines the action of an Operational Energy Sink needed to support an Operational Capability.

## **INTRODUCTION**

Electric cars, renewable energy, autonomous vehicles, and consumer electronics are evidence that a new power and energy future is emerging in the civilian and military worlds. In the naval world, recent developments in weapon systems have delivered the next-generation in defense capability, and these systems are challenging existing ship design practices, theory, and engineering tools. Like hybrid or all-electric cars, Navy ships are moving to a new paradigm where electric power and energy supply are directly related to ship performance (Ames, R., 2016). In non-military applications, Yara Birkeland, developed by Yara International, is an autonomous, fully electric-powered container ship. She has a 7MWh battery, charged by Norwegian hydropower. In today's environment, everyone wants high-power density, energy-efficient systems, speed, control, survivability, and upgradeability—all at an affordable price in a functional package. For naval applications, the promise of these new systems is so compelling that it will set the stage for warships for the next 50 years and push naval design toward a future centered around high levels of power and energy that can be directed wherever it is needed, whenever it is required.

This new future will require power architectures that include energy storage, thermal management, and specialized power converters. For any future ship design activity, engineers, and the tools they use to design these Energy Networks, must understand the boundaries of what defines a system, how systems use energy, and how that energy is transferred and transformed (Ames, R. et al., 2018).

Ontologies are crucial for software applications as they provide a common vocabulary and relational structure to ensure a shared understanding of information structures. In the case of naval ships, it becomes even more critical to understand the interdependencies between ship Components, the Systems they support, and the Energy Domains they belong to. This understanding is necessary due to the complexity and diversity of the missions naval ships must support. Precise and sophisticated design and analysis are required at an early stage of design to ensure that ship designs meet their naval requirements.

To support these analyses, a comprehensive data model is needed to understand each ship System, its functionality, capabilities, and dependency within the full ship Energy Network. This necessitates an understanding of energy within a Component, between Components, and within a Energy Network of connected Components. Controlling this energy is vital to providing Operational Capabilities and meeting survivability requirements.

This paper will address how we applied digital engineering principles and computer science to designing ontologies that allow for the modeling of complex operational systems riding on the same shipboard Energy Network. This Energy Network must support varying levels of detail needed during design while at the same time giving insight into the impacts of that design on ship system operations and their energy loads over time.

An ontology precisely describes how Concepts can be decomposed into Systems and Components and how those Systems and Components are connected as part of networks. Well-defined ontologies are important to enable multiple design tools to unambiguously have a common understanding of a share Concept data model. See Noy and McGuinness 2001 for a complete description of ontologies and the ontology development process. The ontology described in this paper is being implemented through evolving the class structures of the Leading Edge Architecture for Prototyping Systems (LEAPS) in support of FOCUS. FOCUS is the US Navy-managed ship-specific ontology shared among multiple ship design tools. It is also an acronym for Formal Object Classification for Understanding Ships.

The bulk of the work in ontology building is to make sure that the definitions are sound. In some ways, the true output of this paper is the definitions. It is suggested that the reader refer to the definitions section often as they progress through the paper. Capitalized words in this document indicate that they are being used according their respective definitions.

## **SYSTEMS AND ENERGY NETWORKS**

The relationship between Capabilities, Energy Networks, and operational Systems forms the foundation of the developed ontology. Wikipedia defines a System as; "... a group of interacting or interrelated elements that act according to a set of rules to form a unified whole. A System, surrounded and influenced by its environment, is described by its boundaries, structure, and purpose and is expressed in its functioning." In our ontology, a ship System represents a collection of interconnected Components and subsystems that work together to perform a specific function or set of functions that provide Operational Capabilities. We define a Capability as the ability to achieve a desired effect under specified performance standards and conditions through combinations of ways and means [activities and resources] to perform a set of activities. The boundaries of a ship System are expressed by the System's Capabilities to the operator at one end, and the energy source needed to function on the other. A System uses an Energy Network.

Illustrated in Figures 1 through 4 are simple schematic representations of an IPS propulsion System that spans three Energy Domains: electrical, mechanical, and fluid. We define a System, such as this propulsion System, as a collection of Components and Energy Resources that support one or more Operational Capabilities. An Operational Capability is the Capability of a System to perform a mission, operation, or function in the intended environment. In the case of our propulsion system below, the Capability supported is mobility – power is provided to the propellers and forward pod such that the ship can move from point A to point B.

Systems are bounded at one end by Operational Energy Sinks and on the other end by the Energy Resources necessary for the System to function. Energy sinks are Components that transfer energy into the Natural Environment. Operational Energy Sinks transform energy to provide one or more Operational Capabilities. Non-Operational Energy Sinks transfer significant amounts of energy to the Natural Environment but do not provide an Operational Capability. The seawater cooling outlets in Figure 4 are examples of Non-Operational Energy Sinks; they reject energy as waste heat into the environment. The propellers and the pod in Figure 2 are Operational Energy Sinks. The generator sets in Figure 1 are Electrical Domain Energy Sources. A Domain Source is the energy source specific to a particular Energy Domain. The Domain Source is the first Component specific to that domain that is exclusive of any energy transfer process outside the Component. A Domain Sink is defined as a domain terminus. The Domain Sink is the last Component specific to a domain exclusive to any energy process outside of the sink Component. Each propeller in Figure 2 is an example of a mechanical Domain Sink.

The Primary Energy Resource for the System is the fuel in the tank (Figure 3). An Energy Resource can produce electricity, move objects, generate heat, and power and sustain life. Most shipboard energy today is in the form of Natural Energy Resources. Natural Energy Resources are resources used to convert energy and are extracted from the

Natural Environment. They may be stored on the platform (e.g., fuel) or extracted from the environment during operation (e.g., solar). They include, but are not limited to, fossil fuels, wind, solar, natural gas, oxygen, uranium, and hydrogen. Manmade Energy Resources are Components that exist by design as sources of energy. Energy resources can also be categorized as either primary or secondary. Primary Energy Resources are the sole originating source of energy for the platform and may be natural or manmade. Secondary Energy Resources are Manmade Energy Resources that provide energy to the Energy Network but are replenished from some Primary Energy Resource. Examples include, but are not limited to, an electric battery, flywheel, or capacitor Components.

Each System is part of a larger Energy Network that defines the network of Components and Energy Resources that facilitate the distribution of energy throughout the Concept. Energy Networks are bounded by Primary Energy Resources at one end and the Natural Environment on the other. Within the Energy Network, energy is transferred from source to sink. Energy Networks consist of all of the Connections between concept and Components, between Components, and within Components.

A Domain Network is a domain specific subset of the total Energy Network. Domain Networks facilitate the distribution of energy specific to its domain type. Domain types include, but are not limited to, electric, fluid, thermal, chemical, magnetic, structural, and mechanical. A System Domain is defined as all the Components and Connections of a System in a Domain Network for a particular System. Figure 1 depicts the electrical System Domain for the propulsion System. Figure 2 shows the mechanical System Domain. Figure 3 and Figure 4 show the system in the fluid Domain.

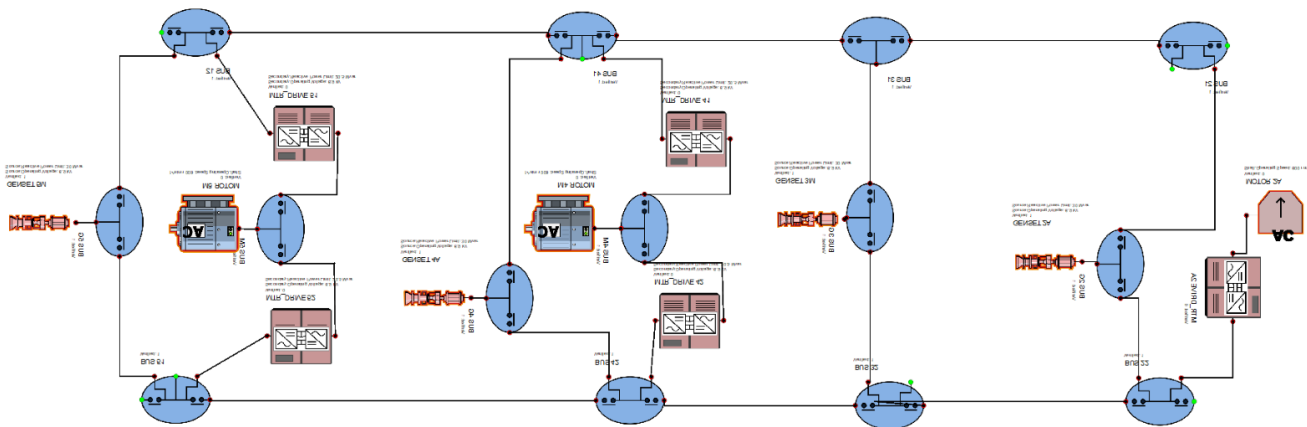


Figure 1- Example IPS propulsion System, electrical domain view

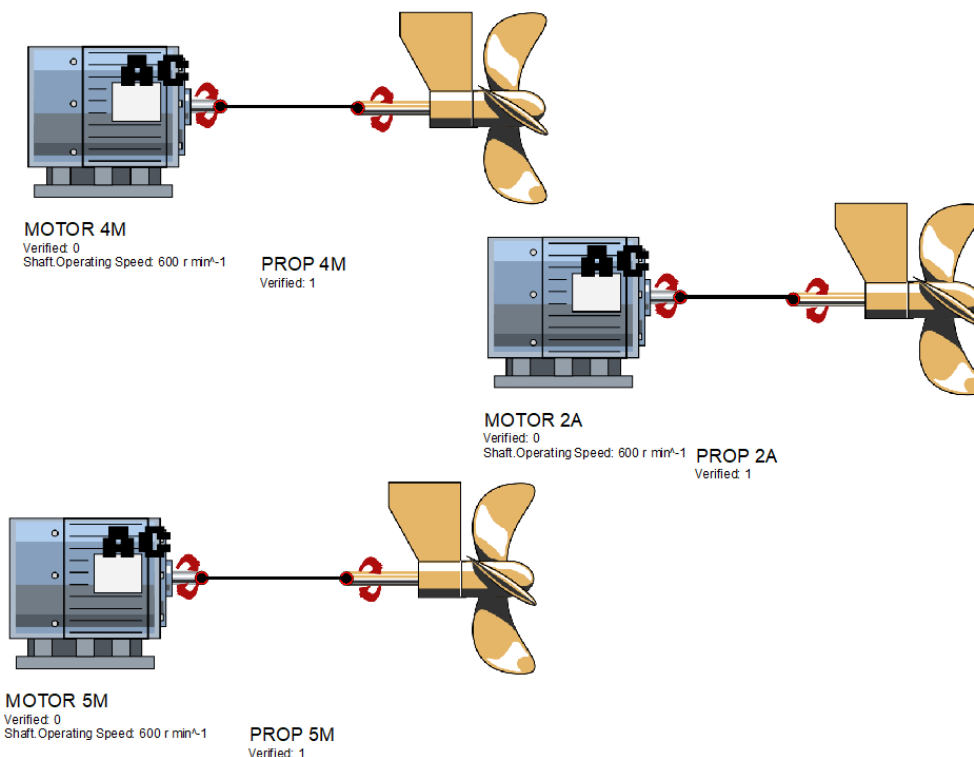


Figure 2- Example IPS propulsion system, mechanical domain view

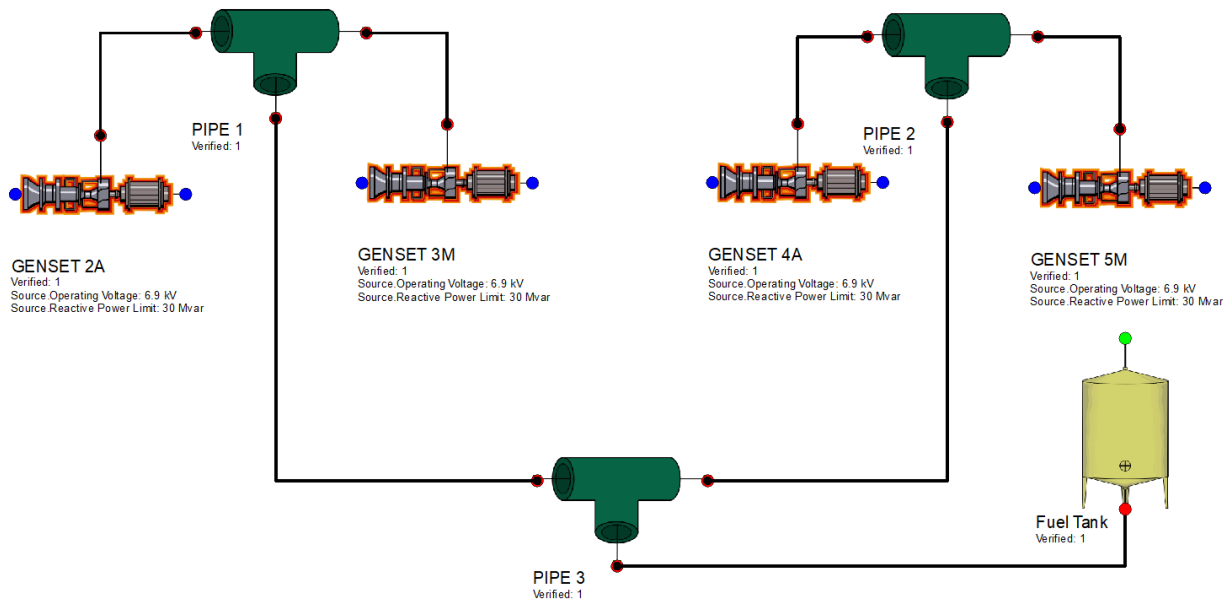


Figure 3- Example IPS propulsion System, fuel proxy, fluid domain

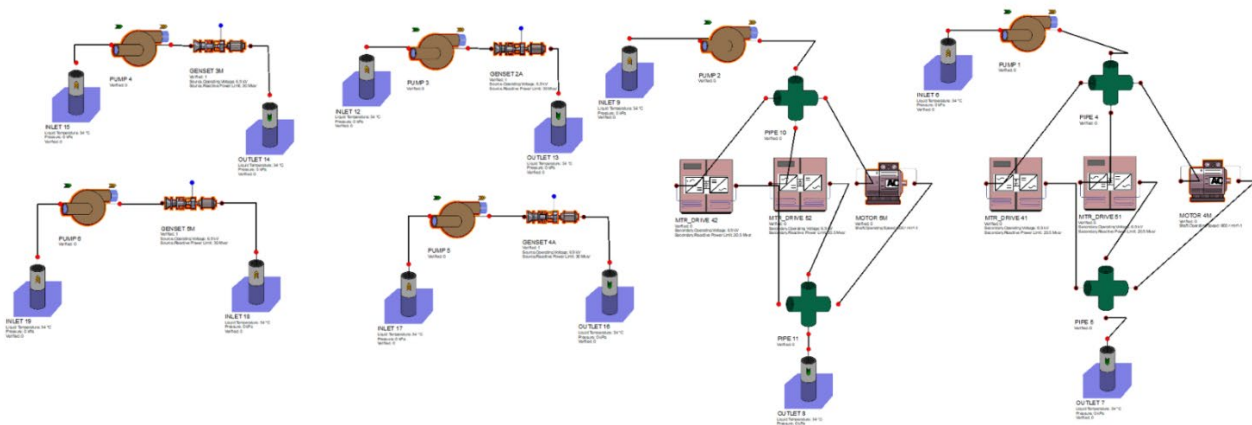


Figure 4- Example IPS propulsion System, seawater cooling, fluid domain

We should not design and engineer Systems independently, but rather we should design and engineer Energy Networks that provide and transport energy through these Systems. The differentiation between Systems and Energy Networks is foundational to any modeling requirement used during the design process. For example, the sizing of generators, chillers, and load centers, and the design of zonal architectures are done to support multiple ship Systems. In this way, a ship design objective is to engineer an Energy Network that supports the operational requirements of all ship Systems; and to discover the interdependence between these Systems and the Components that they share. The challenge becomes designing an Energy Network while at the same time meeting System requirements.

Designing Energy Networks and analyzing System performance cannot be properly executed without appropriate tools. These tools must be able to model and analyze the design; the design analyzed should be based on a commonly understood definition of the Concept's design. This requires a data model or ontology that can capture the complexity of the network/system relationship while dealing with the various levels of knowledge about the design as it matures along an acquisition process.

An Energy Network can be thought of as a graph defining how Components and Energy Resources are connected to distribute energy throughout the ship. Within the Energy Network, energy moves from one or more primary Energy Resources, like oxygen and fuel, through connected Components and terminates at Operational Energy Sinks where the energy provides an action necessary for an Operational Capability to exist. The actions of these terminus Components transfer that energy to the natural environment. Once the understanding of energy flow and conversion is gained, building an ontology model that represents this Energy Network and the Systems riding on it becomes the challenge.

The modeling philosophy that was applied makes the following assumptions.

1. Through the law of energy conservation, energy is a conserved quantity that cannot be created or destroyed but is instead converted in form.
2. Energy resources are part of the Energy Network. For example, if a ship runs out of stored fuel, or the batteries of a fully electric ship are depleted, and the ship is dead in the water, then energy was converted, is now in the environment, and is no longer onboard and available for use.
3. Energy is distributed, controlled, and converted along one or more distinct Energy Networks to support required Operational Capabilities.
4. Energy Networks have multiple Energy Domains. These include but are not limited to electrical, mechanical, and fluid (gas and liquid).
5. Humans are not part of the Energy Network. They are operators and are part of the Natural Environment.

Most engineers will logically think of Energy Networks in the context of a network of connected Components. Engineers model these Components and networks within modeling and simulation tools/applications appropriate to answer specific questions. The ontology used by these modeling and simulation applications must provide the mechanism to define and simulate these Components and networks unambiguously; the definitions must be understood the same way among multiple applications.

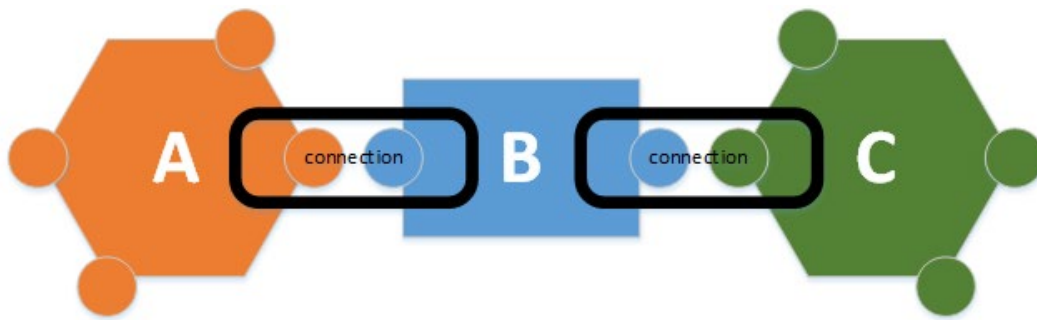


Figure 5- Connected Components

In Figure 5 we have three connected Components, but we know nothing about A, B, and C individually; we know only that they are connected. Concerning their Connections, we may have information about the Ports on each Component, the Energy Domains, properties, and possibly even the physical interface of each Port. We can formalize the Connection object, which can have properties. But at this level of modeling, we do not know what happens inside each Component or how it behaves in relation to its connected neighbors. For all Components of an Energy Network to simulate, applications would need to know specific details about each Component's behavior and its ports. Each Component needs to be defined, formalized, unique, and have associated code modeling its behavior.

A simplistic example of these Energy Networks might include an engine connected to a generator by way of a shaft (Figure 6). A cable is connected to a generator that powers some electrical device, like a radar, through a switchboard and/or load center.

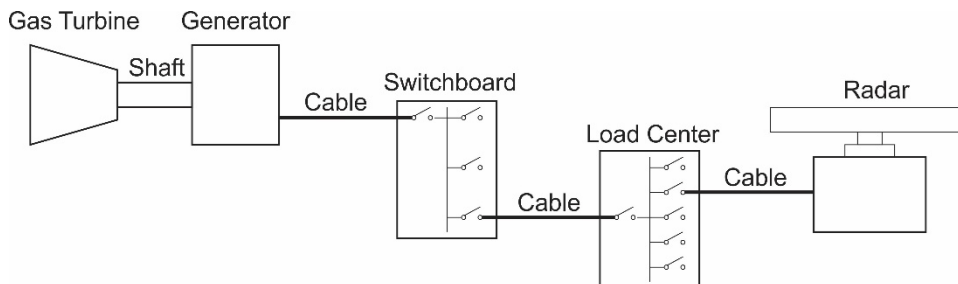


Figure 6-Simple Engine and Electric Load

In the model shown in Figure 6, the network is incomplete as an energy model. It is incapable of being simulated without additional modeling assumptions. It could exist as a pattern or subnetwork for reuse, but it is incapable of an end-to-end energy simulation as no energy can be generated for use in one or more Systems; there is no energy source or resource. The model also does not include dependent Components like chilled water that are required for it to function. This increased level of detail is depicted in Figure 7. Like the generic model depicted in Figure 5, to perform an end-to-

end energy simulation, the constituent parts of this model, including their Connections, must be formalized and connected to an energy source. For early stage design, this level of detail is limited but may be sufficient in most use cases with adequately formalized and modeled Components and Energy Networks.

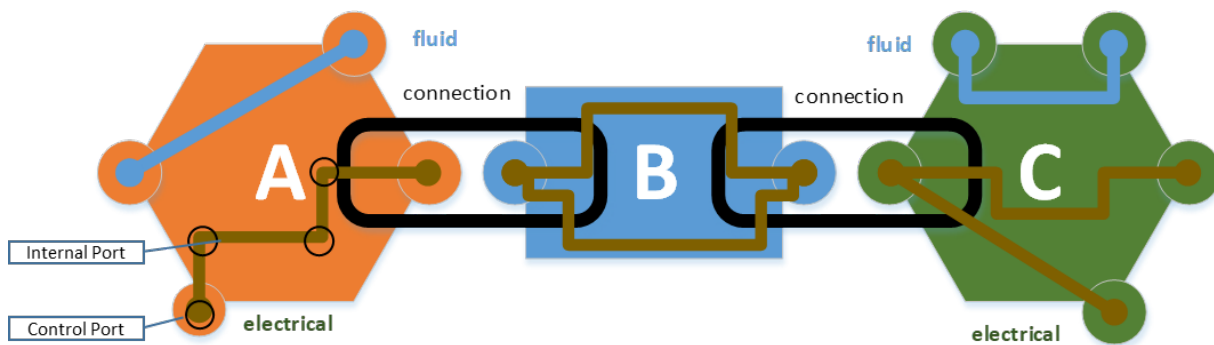


Figure 7 - Components with Internal Connectivity

When the level of detail increases and the definition of the behavior and control of Components operating in an Energy Network becomes critical to answering design questions, the ontology model must allow for the refinement of Components. These refinements include, but are not limited to, Energy Network paths within each Component. This also affords multiple configurations of the same Component type. Switchboards, computers, and cables are examples of standard Components that may have many different numbers of Ports and internal conductors or fluid paths, but functionally belong to the same family of Components.

If we consider again the model in Figure 6 as a use case, we can see how this Energy Network meets the requirements of a single System supporting a single radar and its Operational Capabilities. We also know that many of these Components will serve more than this single System function within the larger context of the ship design. This simple example can quickly get complicated when the fuel necessary to power the engine is stored in a compartment in the ship, and we add additional Components needed to support additional capabilities like ship propulsion (Figure 8).

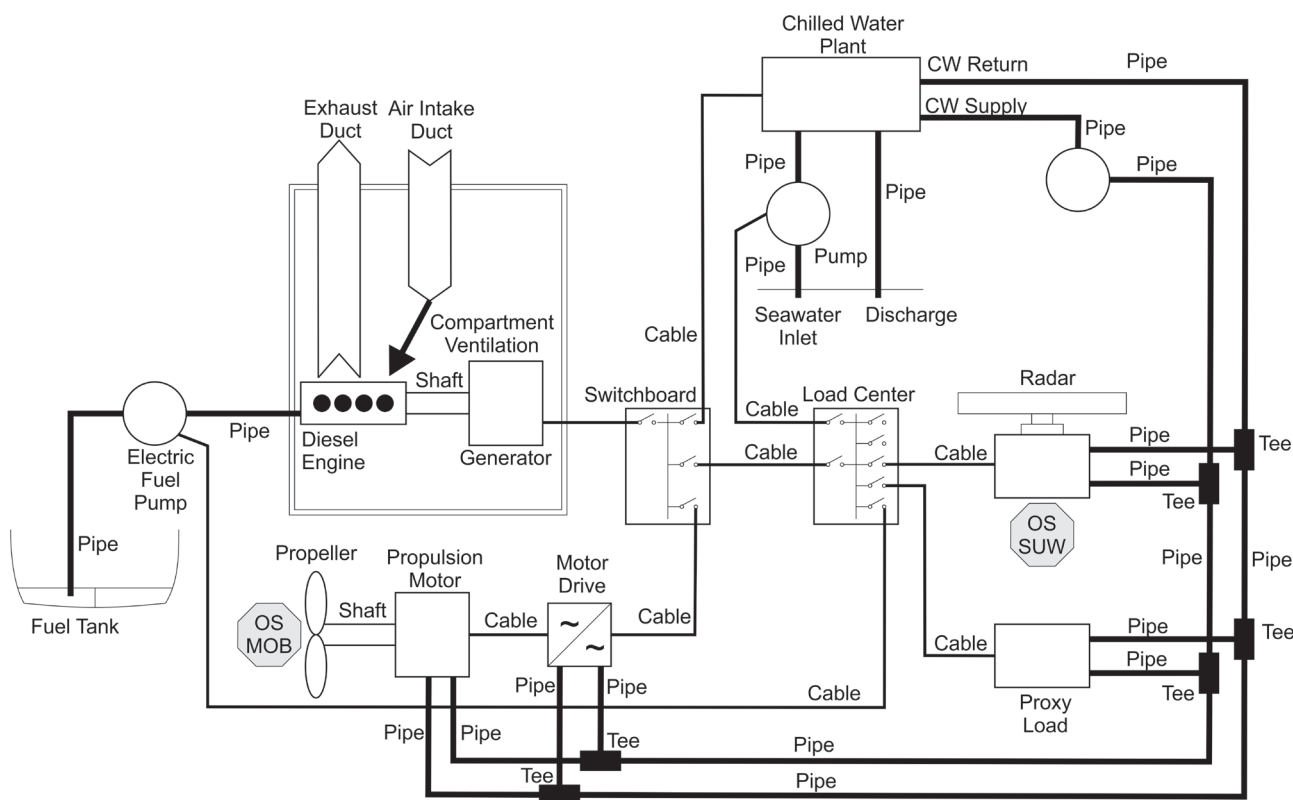


Figure 8- Simplified Notional Energy Network



Now, there are two ship functions that share some, but not all, of the Energy Network Components. Both are dependent on fuel and oxygen to provide energy to the network and, therefore, to each system. The ontological relationships between Energy Network Components are critical. Note that the capacity of the tank is directly related to the design of the ship and space available for fuel storage and, therefore, defines the operational limits of ship Systems. Two Components of Figure 8, the radar and the propeller are the Operational Sinks for two Operational Capabilities; these two Operational Capabilities are implemented through two interconnected Energy Sinks Networks. During their operation, these two Energy Networks consume Energy Resources and act as an energy sink to transfer energy off the Energy Network through some action. This relationship breaks down into well-defined model objects that ultimately define each of the two Systems (the propulsion system and the air radar sensing system.)

In the example shown in Figure 8, the air in this naturally aspirated combustion engine comes from air within a compartment. The compartment air is serviced by an air intake duct via a ventilation System that terminates at a Domain Source. This Source Port has a Connection to a Natural Energy Resource which is oxygen from outside air. Likewise, the engine is cooled by seawater; without cooling, the engine will fail. This figure also illustrates the multiple domains of the Energy Network: electrical, mechanical, and fluid.

Figure 9 shows a 3D view of the Compartment/air relationship. The HVAC duct serves to both cool the Compartment and to ensure that air is provided to the engines. Since the Compartment doors can be closed, a dedicated external air source is required. In the event the HVAC fans fail, or the intake air is blocked, the Compartment doors can offer a redundant air source. An Energy Network should be able to model this relationship. While not shown in the figure, a System Information Network could be used to sense whether the HVAC fan fails and to cause the Compartment door to be opened, either manually or automatically. In this example, the exhaust duct would be an example of a Non-Operational Energy Sink.

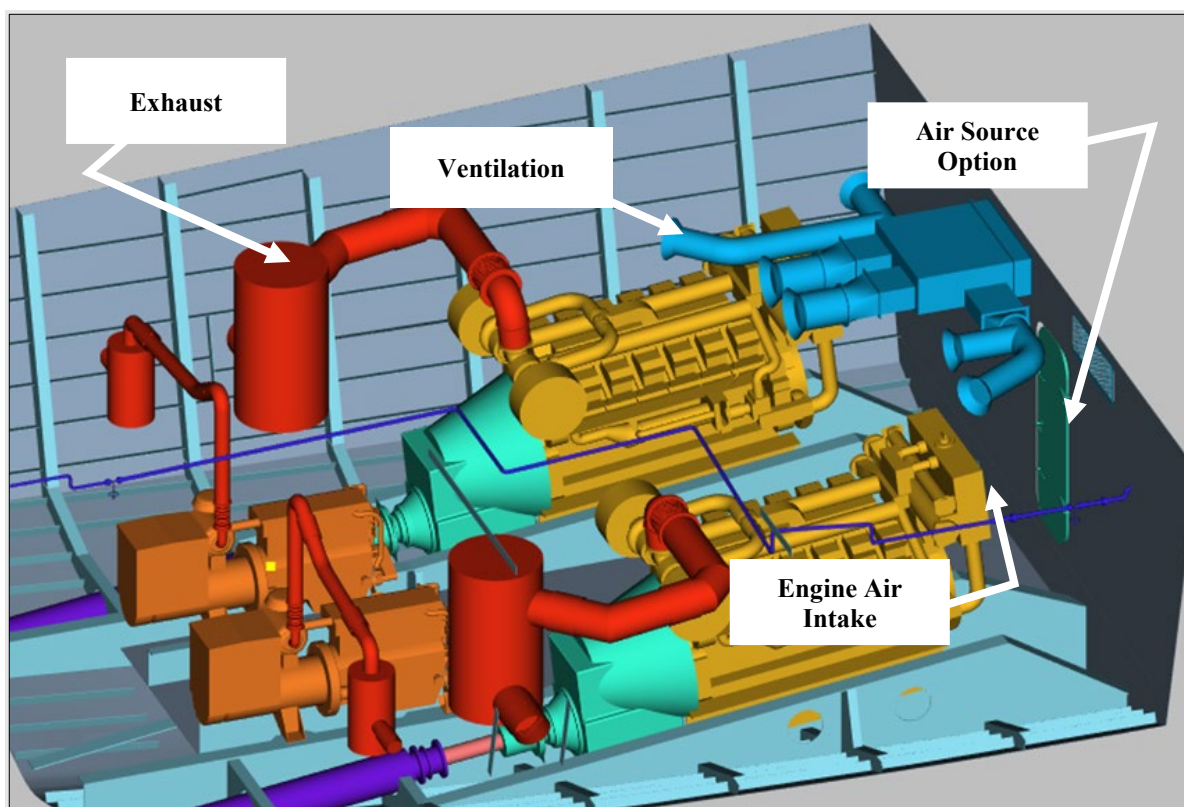


Figure 9 - Notional Machinery Room

## DEVELOPMENT OF NETWORK ONTOLOGY

The requirements and use cases that drive the modeling of an Energy Network should be identified prior to ontology development. The objective is to enable Component and platform sizing, technology fusion, and operational measures of performance during early stages of design. However, the strategy of ontology design and development follows a philosophy of supporting all stakeholders of data at all stages of design. The ontology model must be rich in information related to connectivity, properties, and operational state. The energy flow model must be able to understand this ontology, analyze energy flow, and be able to change the state of Components. The ontology model must be able to characterize behaviors between Components, within Components, and between Components and Concept space (e.g. arrangements).

In ship design it is easy to ignore the complexity of some standard ship Components simply because there are so many of them, and modeling everything at all levels of detail is not possible or desirable. To the engineer worried about the performance of some of those Components, however, these details matter. Ontology development should not care what you choose to model, or how much detail you provide or require, only that you can provide the data necessary to model the problem. This leaves the detail of details to the application generating or using the data.

The following are some of the decisions that were made about ontology development given the requirements we established.

From a data modeling standpoint,

1. Capabilities correspond to a hierarchy of required Operational Capabilities. Capabilities are only described in the abstract and does not specify how a capability is to be implemented. A Capability is structured as a hierarchy of Capabilities, with the most general at the root and most specific at the leaves. At the leaf-level, Capabilities may have a measure specified, along with an environmental condition for the measure.
2. An Operational Capability is the last Capability in the hierarchy related to a System Function Action.
3. A System Function Action defines the action of an Operational Energy Sink needed to support an Operational Capability. An Operational Sink is the Component, or Components, providing the System Function Action that transfers the energy off platform.
4. Operational Sinks define the function of a System and are the terminus of the System boundary.
5. Systems are defined as all connected Components of the Energy Network required to provide energy to the System Function Action that enables the Operational Capability of the System. This includes Components required to maintain operational integrity that do not necessarily provide energy to the System. (e.g. cooling).

Table 1 provides examples of Operational Capabilities and how they are implemented through Systems, Operational Sinks, and System Function Action.

**Table 1: Example of linkages among Capability, Systems, Operational Sinks, and System Function Actions.**

Mission	Capability	Operational Capability	System	Operational Sink	System Function Action
MOB	MOB 1.7-Transit at high speed.	Move	Propulsion Prime Mover System	propeller(s)	generate thrust
MOB	MOB 7.3- Get Underway/Moor	Hoist	Mooring System	windlass (s)	rotate
AW	AW 6.7- Detect, classify, track	track	Surface Surveillance Radar System	radar antenna(s)	Transmit-Receive-Rotate
AW	AW 6.7- Detect, classify, track	track	Surface Surveillance Radar System	display monitor (s)	emit

## LEAPS, FOCUS, and UML

The system ontology developed is generic in its approach and was modeled using the Unified Modeling Language (UML). The UML model was built as part of the Leading Edge Architecture for Prototyping Systems (LEAPS) software development framework. Two software products, the Rapid Ship Design Environment (RSDE) and the Smart Ship Systems Design (S3D) tool (Rigterink, et al, 2016) are both leveraging this ontology model as part of design tool integration. RSDE is a 3D ship design tool that is having its Machinery Module modified to support Energy Networks and System analysis, and S3D will be a plugin within RSDE that will provide for multi-domain Energy Network modeling, energy flow, and System discovery. Once completed, these tools and others will be able to support the integration of operational requirements, Energy Network design, System discovery, and operational analysis like Electric Plant Load Analysis (EPLA). As described by Snyder et al. (2019), LEAPS and FOCUS are also being used to integrate other ship design tools.

The LEAPS MetaModel is part of the LEAPS framework. The LEAPS MetaModel is a class structure - a set of classes and the relationships between them. It is designed to be flexible enough to represent any engineered system or structure. The current version, LEAPS 6, was upgraded to facilitate the work described in this paper. A few key classes are introduced below.

**Concept class** - A class that defines the attributes, the Systems, the Components and the views of geometry that represents a product in design or under study. It is generic to all things engineered, but is used here to define platforms such as a ship, submarine, aircraft, or other engineered product.

**Component class** – represents physical objects that can be thought of as parts of a Concept. They may be connected in ways that provide function and capability. Components can be thought of as any part of a ship that represents something as discrete as a radar or engine, but also includes parts like stiffeners, plates, foundations, and mounts. A Component in many ways acts as a small Concept, but it is modeled as something a Concept owns. Components do not have their own Systems, but Components do have most of everything a Concept does.

Node class – can be owned by either a concept or a Component. Used to connect Component interfaces to either a concept interface or another Component interface. Can also be used to represent physical location or as a container for domain specific properties. Ex: Component location Nodes, ports

Diagram class - The Diagram class provides the schematic information that defines a connected graph or network. Diagrams can define a connected network between Components, within a Component, and a Connection between Components and Concept structure.

Connection class – can be owned by either a concept or a Component. Used to represent logical relationships, such as energy flow, both within a Component and across a concept. Used to connect Components to each other and the concept. Ex: interface Connections, conductors

System class – can have both diagrams and Components. The system class allows us the flexibility to model both physical and logical relationships. Ex: assemblies – collection of Components that are in physical proximity that also have a logical relationship. Ex: System, Energy Network, Domain Networks

All of these classes inherit properties. The relationships among these classes is depicted in Figure 10.

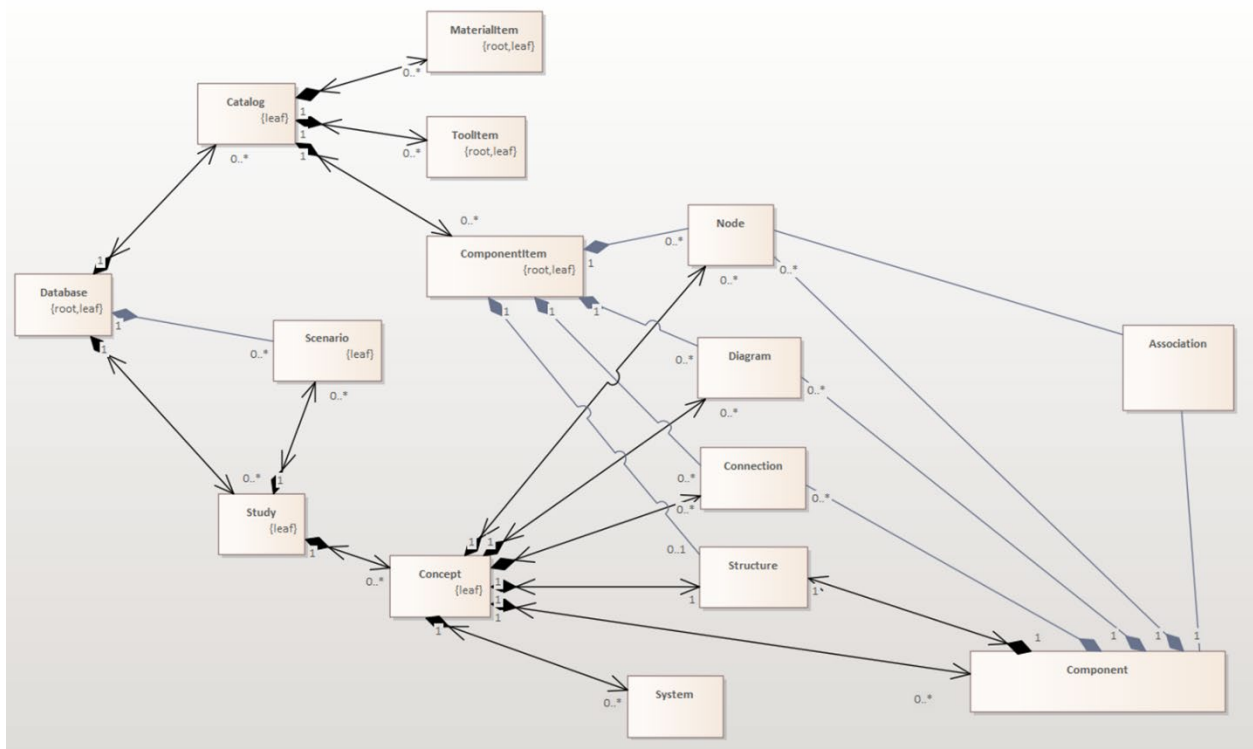


Figure 10 - LEAPS Meta Model (high level view)

Our data model is formalized at the object level. A product meta model (PMM) is an object metamodel that represents the structure and behavior of a product, such as a ship. It takes a data centric approach to integration as opposed to a data driven design (McComb, 2018). The PMM is composed of objects of the LEAPS metamodel classes and the relationships between them. It is at this level where we begin to formalize products as objects with recognizable names, such as ship Concepts, energy storage Components, etc.

Once we understand the class structure, we can look at how we use instance objects in our models. As we have mentioned earlier, one of the more challenging modeling requirements involves the ontology modeling of Components.

### Component Ontology Modeling

There are varying levels of abstraction when it comes to modeling the internal workings of a Component. At the lowest level of abstraction, Figure 5, we need to look at energy flow into and out of a Component, any network relationships (Port Connections) that are necessary for a Component to function (e.g., air and fuel for combustion), and what Energy Domains a Component participates in by way of its Ports.

At a higher level of abstraction, a Component can be modeled with more detail and that detail may vary for each Energy Domain. This modeling can involve detailed internal Component networks when controls and interacting behaviors must

be simulated. The gas turbine in Figure 11 supports an explanation of Energy Network Diagrams as they relate to the behavior of a Component.

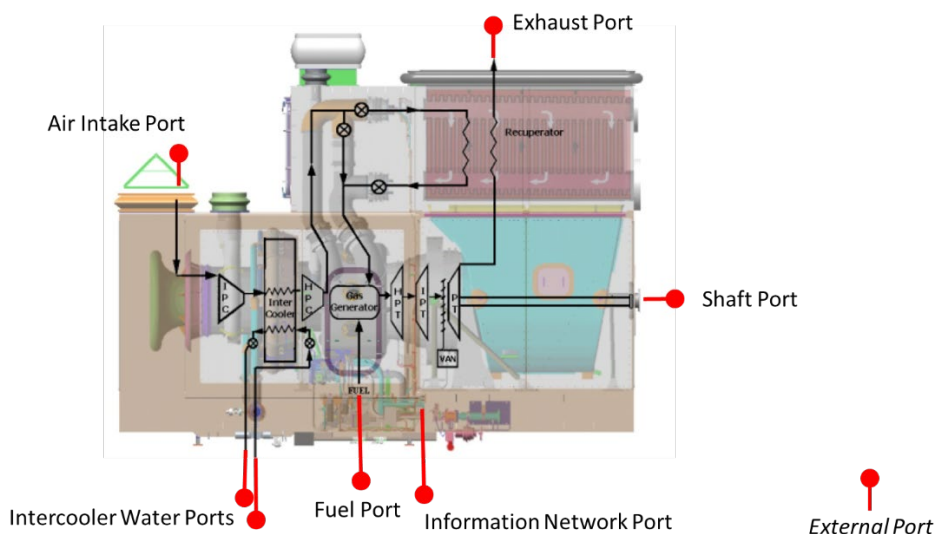


Figure 11-External Ports for Gas Turbine Diagram

In this example, the gas turbine has internal Component energy transformation as well as external Port Connections to Energy Domains like fluid-HVAC via intake and uptake ducts. Energy Domains within Components may have interacting behaviors. Consider a Component fluid domain designed to cool the electrical domain or mechanical domain – these domains will have a direct dependency on each other's behavior. It is the responsibility of the Component to capture the dependency/behavior relationship between Energy Domains interacting with each other within a Component. In the gas turbine example, when connected into the Energy Network, the HVAC ducting Components will impact the gas turbine's performance based on losses from their routing or the ambient intake air temperature. Figure 12 illustrates this relationship. In this case, pressure losses for the design of the intake duct (fluid domain Port) and the exhaust duct (fluid domain Port) are affecting rated output power at the shaft (mechanical domain Port).

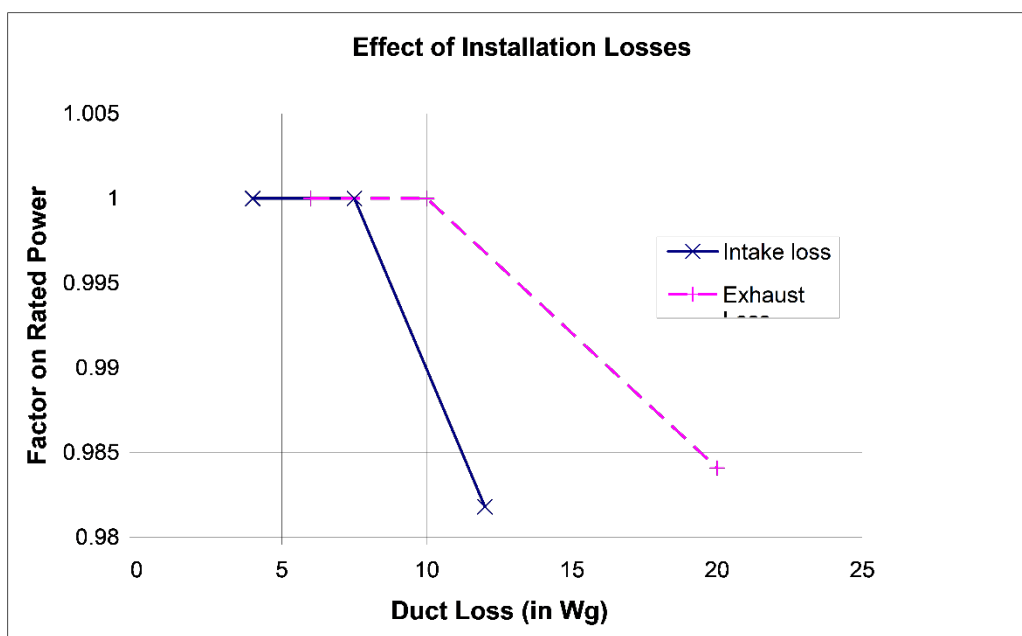


Figure 12- Port and Component Dependency

This use case requirement for Components to have internal network Diagrams demonstrates that Components can model a complexity level usually associated with Concepts, and that they have their own operational states, performance, and controls, just as a Concept does.

Ultimately the Energy Network is a graph composed of both Concept and Component networked objects. Components can have varying levels of configuration in addition to varying levels of modeling detail. Configurable Components add significant complexity when differentiating one from another. A computer Component is a perfect example. Because

Concepts and Components can have any number of Diagrams having any number of Ports (Nodes) and Port Connections, the ability to formalize something like a computer at the instance level is possible. Both the tablet and the blade server can be modeled. The challenge is in the building of specific computer instances that represent an actual Component. What will differentiate one from the other will be the Component's Diagrams. Defining these internal Energy Networks will take time and effort. As such, the need for a library of available Components with their respective Diagrams is essential.

## Logical and Physical Views

The ontology follows a data centric approach. This means that the data model is designed to represent reality rather than the data requirements of any software application. In a data centric strategy, all tools see the same data even if the application is limited and specific in its use of that data. If we look at 2D schematic flow solvers like S3D, we recognize that these tools perform analysis on a logical view of the Energy Network. Applications like RSDE are more 3D driven, requiring accurate weight and volume analysis that the logical tools do not need or use. Both S3D and RSDE need to model the same Energy Network and understand the Systems represented in the design. Logical and physical views of a single data representation are possible and desirable in a data centric model, but the data must be synchronized as each application makes changes to the shared data. Consider the arrangement of Components within a Concept in the example shown in Figure 13. The purpose of the example is not to demonstrate a realistic Energy Network, but rather to show the logical and physical relationship between a cable and two connected and arranged Components. In a data centric model, there is only one cable, one laser, and one energy storage device. A very plausible use case may suggest that the logical data like Ports and Connections of the three Components may exist first. It is very likely that these logical designs will exist as patterns or collections of sub-networks persisted for reuse by tools like RSDE or S3D across different designs.

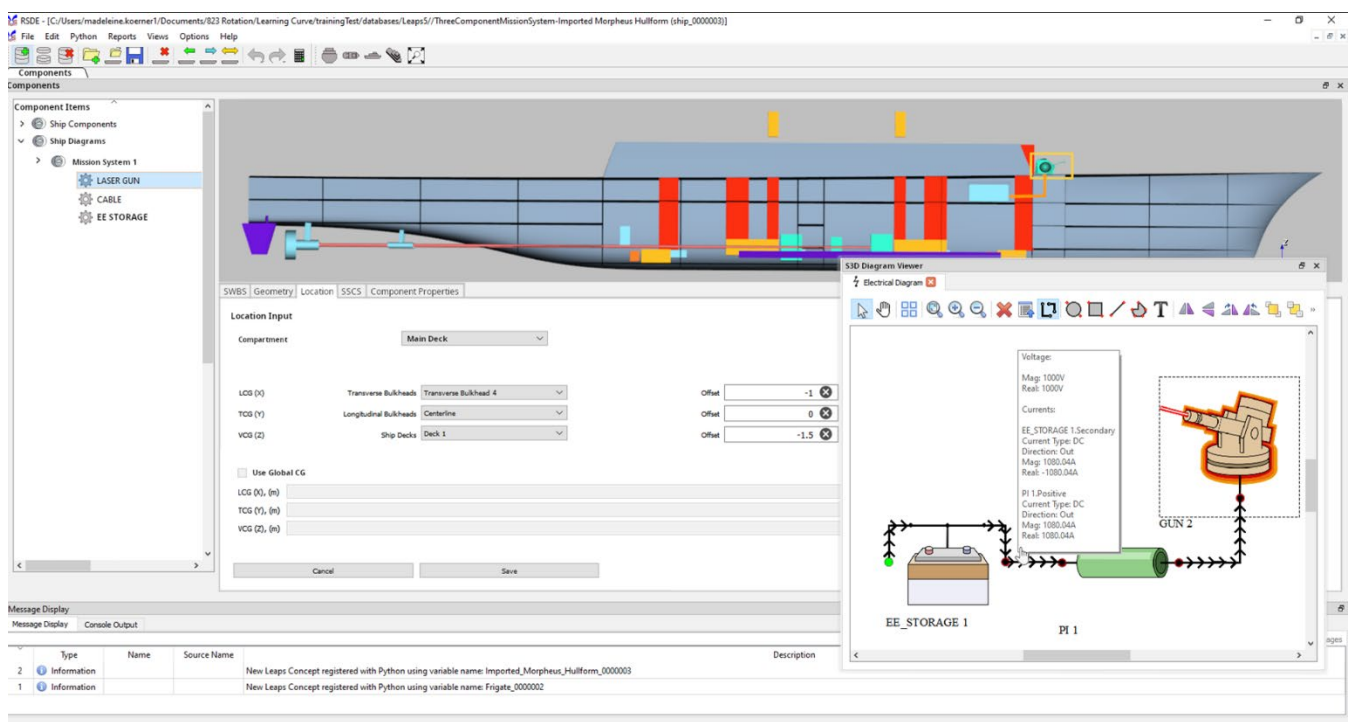


Figure 13- RSDE (physical) and S3D (logical) views of the same data

RSDE may need to add this pattern and rearrange either the battery Component or the laser Component. Within RSDE the cable would need to be routed and the resultant cable length updated in the database. The logical connectivity between the three Components remains unchanged.

The relationship between the electricCable Component and the electricalCableLength Property in the UML model is depicted in Figure 14. The cable owns this length property, but it is the responsibility of the applications to ensure that as a cable changes, its data is updated.

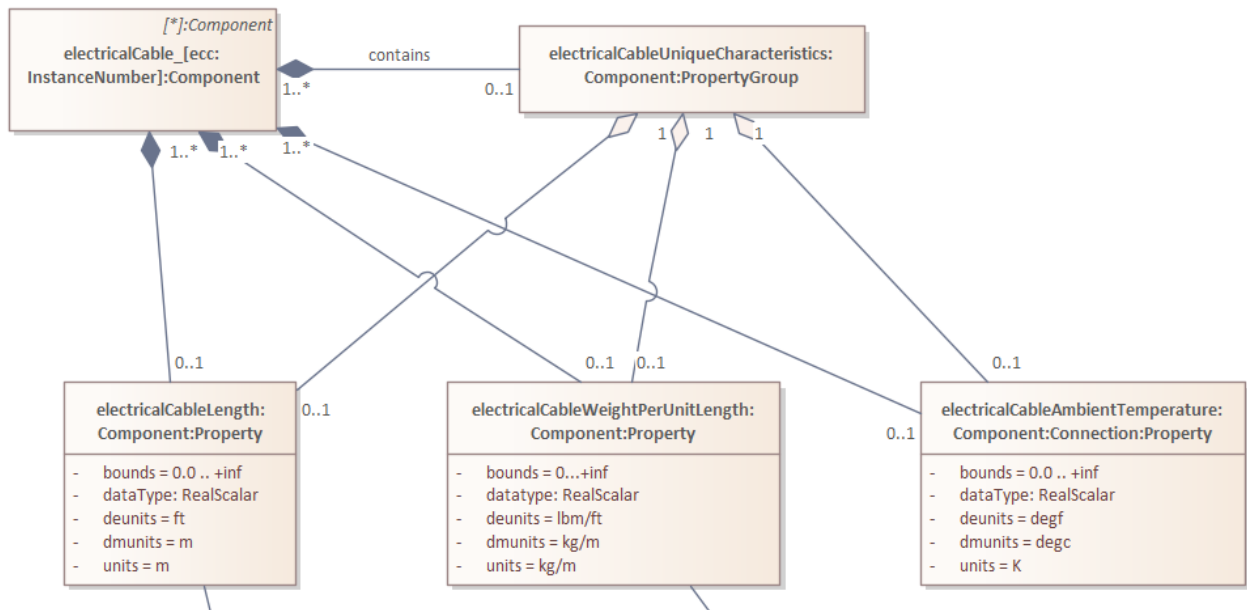


Figure 14- Electrical Cable properties

The electrical resistance of the cable provides another example of the interdependency between the Concept and the Component. This property will change based on the temperature of the compartments that the cable is routed through. The powerflow simulation is therefore affected by the Concept even if the simulation is unaware the Concept exists. Ship design software like RSDE define and update Component Property data, such as cable length, during arrangements. Analysis software such as S3D use these Component property data to analyze and simulate the Concept.

### Proxy Components

During early stages of the design process, not every Component will necessarily be modeled. Compartment lighting is a good example. It is not feasible during concept design to model every light and electrical cable and switch on the ship so estimates of the lighting load are used. These estimates are captured in Proxy Components. Currently tools like RSDE estimate the lighting load based off of the arrangeable area of compartment spaces in the ship. A Proxy Component should have both a knowledge of the Concept it belongs to and a method to solve for its load independent of the power flow simulation. For the lighting example, the proxy lighting Component (or an external application) would get the space volume data from the Concept and would include the equation that relates the volume to the properties, such as requestedElectricalPower, necessary for the power flow simulation. Proxy Components can also be used to simulate a portion, or sub-network, of an overall Energy Network by providing substitute sources and loads. As depicted in Figure 15, the proxy Components provide or sink the energy at the boundary interfaces of the sub-network under study; they establish the sub-network boundary conditions.

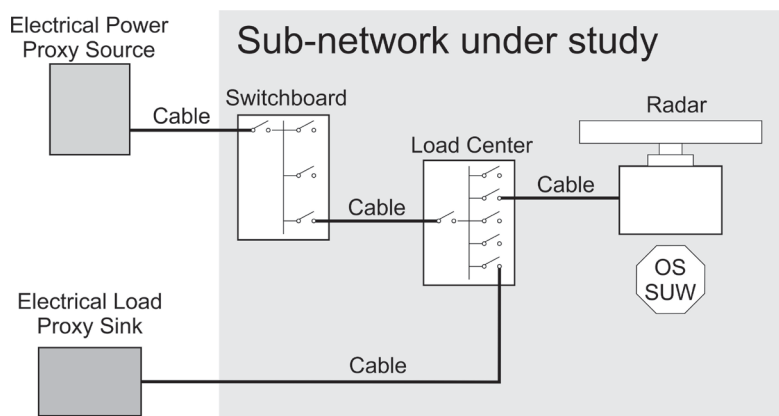


Figure 15-Proxy Source and Sink Components

### Ports

As depicted in Figure 16, Component Ports can be of multiple Port types:

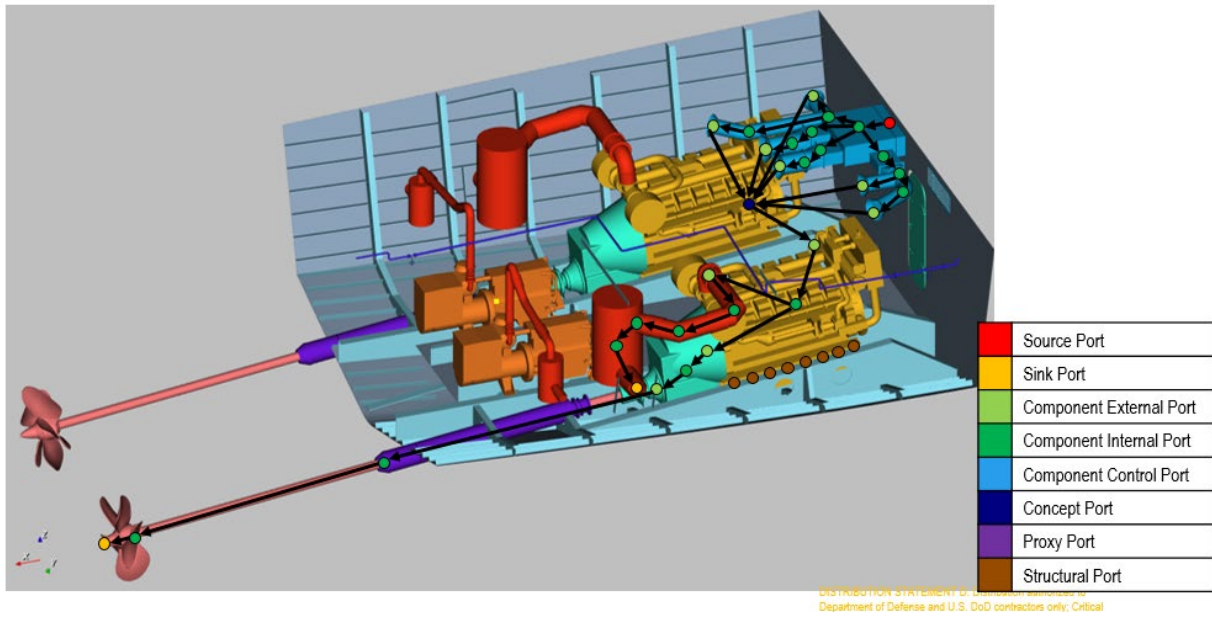


Figure 16- Port examples

**Component External Ports**

External Ports are Ports that allow for input and output into and out of a Component, and provide Connection to other Components via their external ports. For example, the gas turbine from Figure 11 will have multiple external ports. There will be input external ports for fuel and air, there will be output external ports such as power out and exhaust out. These ports have attributes such as Connection type, ratings for inputs such as flowrate for fluids and power ratings for electrical ports. External ports can also be bi-directional.

**Component Internal Ports**

Internal ports are used to define any changes within a Component that provide necessary information for a simulation. An example of an internal Port is a T-Pipe. Unlike a straight pipe, a T-Pipe will have a single input and two outputs (or vice versa.) At the intersection point of the T, an internal Port will be used to inform the fluid flow simulation that the flow is splitting at this location and to calculate the necessary flowrates for all inputs and outputs.

**Component Control Ports**

Component Control Ports are ports on a Component that describe controllable attributes or measured quantities for simulations. A simple example of a control Port could be a ball valve actuator handle. See Figure 17. The Port would not have an effect on the physical representation of the Component but would contain the information needed for analysis in simulation tools.

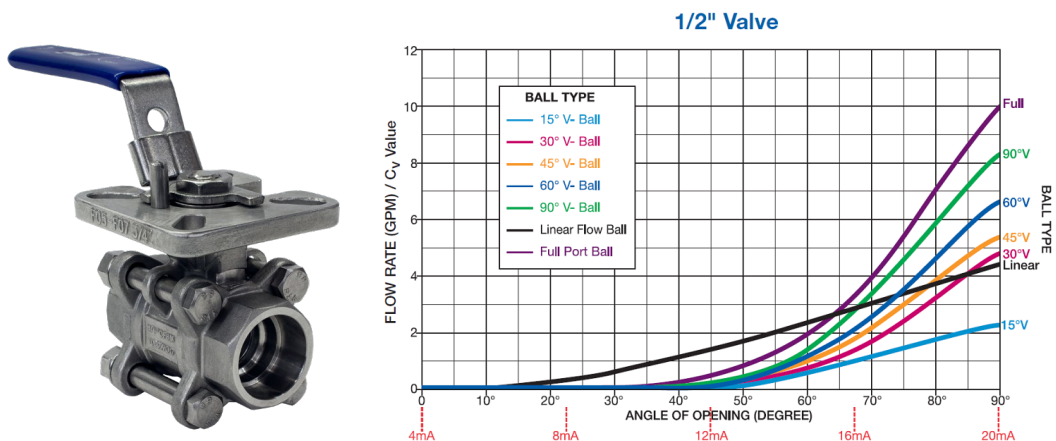


Figure 17: Component Control Port example

**Source Ports**

A Source Port will be the entry point of an Energy Network where an Energy Resource (Natural or Manmade) enters the network from the concept or the environment. An example of this type of Port is the inlet to a fuel line or an air intake to a gas turbine. See Figure 18. A space in the ship used to store fuel is not a Component, but the fuel contained there is a

natural Energy Resource. The external Port of an inlet pipe should connect with and understand that is it connected to a fuel tank if the Energy Network is to connect sources to sinks. It may need to understand the tank's capacity.

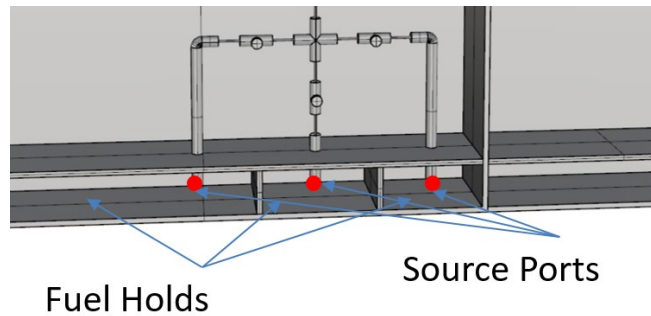


Figure 18- Source Port Example

### ***Sink Ports***

A Sink Port is the last point of an Energy Network where energy is discharged to the Natural Environment. The propellers of a ship would have Sink Ports (Figure 18) where the energy from Fuel and Air for Combustion Sources is finally discharged to the environment to “Move” the ship. (A lot of the energy is also discharged through the exhaust pipes – but not all of the energy discharged through the exhaust pipes is attributable to a given Energy Network)

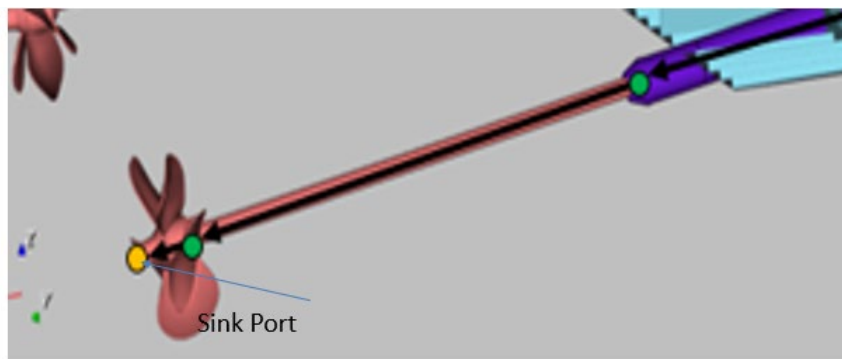


Figure 18: Sink Port Example

### ***Concept Ports***

Concept ports are the subset of Component Ports that connect to the environment at the boundaries of the Concept but are neither Source Ports nor Sink Ports. In LEAPS, openings can have properties that define them as doorways, hatches, ducts, and other open boundaries in the ship's geometry where the knowledge of that opening is important. Knowing a compartment has an opening with a door associated with it allows for the opening to have a state (open, closed). Concept Ports also provide for energy transfer between Components that are not physically connected.

## **ONGOING WORK**

This paper has described the ontological work completed to date. We are currently exploring the incorporation of Elements, Patterns, and Templates into the ontology.

An Element (Figure 19) is a type of Interface that provides for a specific and possibly formalized function. Elements are composed of other Interfaces that provide logical Connections to proxy Components, Component Blocks, Patterns, or other Elements. Elements can be owned by either the Concept, Component, or Catalog Component Item.



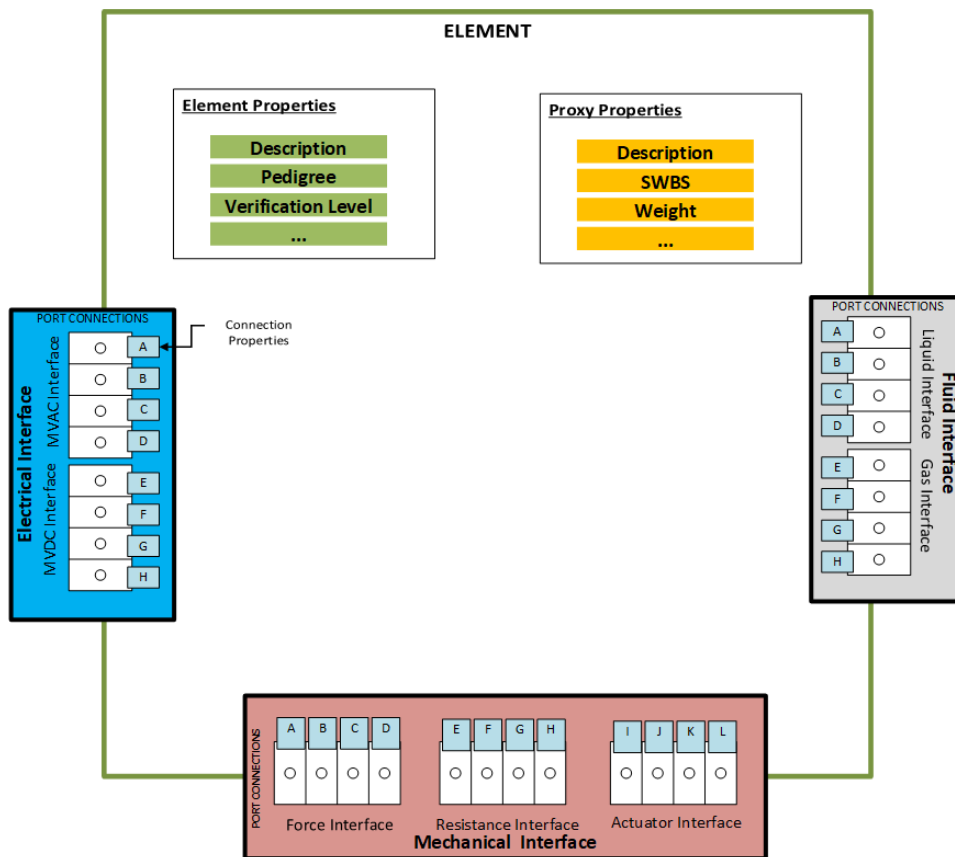
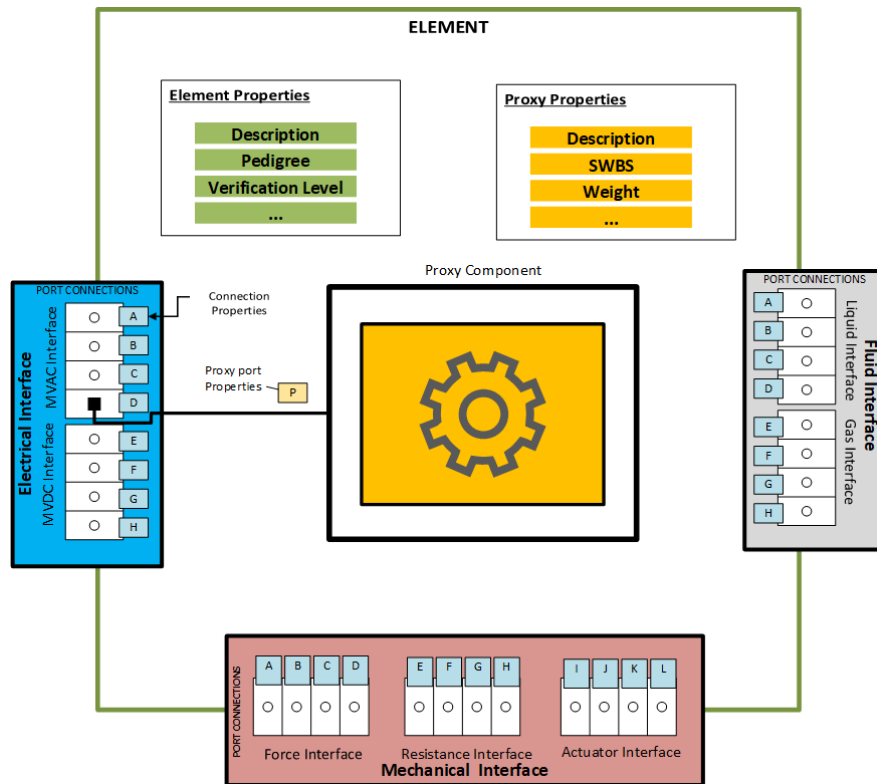
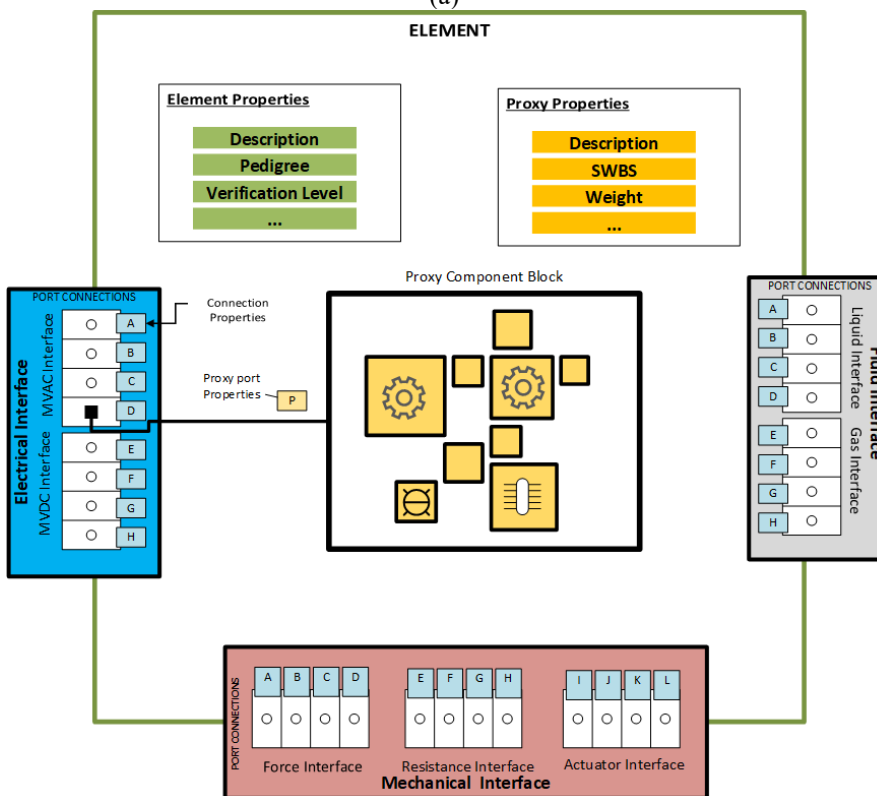


Figure 19: Element example

Elements can have many applications. The initial application of interest is the ability to represent a single interface from which multiple Components, Component Blocks, and Patterns can be substituted during design and analysis. As a design matures, the Element remains as an architectural feature of the network, but the model structure connecting its interfaces can become more detailed as required. Figure 20 depicts an Element where the contained Component evolves from a proxy Component to a diagram of Components. Ongoing work is determining how to represent elements within the ontology.



(a)-



(b)

Figure 20: Element evolution (a) early stages of design (b) later stages of design

As described by Riggerink et al. 2016, Chalfant and C. Chryssostomidis 2017, and Chalfant et al. 2019, patterns are diagrams of sub-networks or networks where the Component properties are set at default values. Patterns enable reuse of architectures across multiple designs; the Components serve as functional entities. A template is a pattern where the Component properties have been defined to represent a specific concept and sufficient sources and sinks (perhaps proxy sources and sinks) are incorporated to conduct a simulation to answer a specific question. Ongoing work is identifying how to represent patterns and templates within the ontology.

## CONCLUSION

A well-defined and extendable ontology is a key enabler for integrating multiple ship design and analysis tools. This paper focuses on describing ongoing efforts to define such an ontology to support ship design and the analysis of Energy Networks. This ontology is being implemented in LEAPS and FOCUS to enable interoperability of RSDE and S3D in the near term. More tools are anticipated to be integrated in the future.

## REFERENCES

- Ames, R., (Fall 2016). *Shifting Ship Design into High Gear: A New Power and Energy Future Is Coming*, Naval Science and Technology, Future Force Magazine,
- Ames, R., & Cooper, K., (2018). *A Perspective on US Navy Early Stage Design Tools and Future Challenges*, SNAME SMC
- Chalfant J., & Chryssostomidis, C. (Aug 2017) *Application of templates to early stage ship design,* IEEE Electric Ship Technologies Symposium (ESTS), pp. 111–117.
- Chalfant, J., Z. Wang, M. & Triantafyllou M., (August 14-16, 2019). *Expanding the Design Space Explored by S3D*, presented at IEEE Electric Ship Technologies Symposium 2019, Westin Crystal City Hotel, Arlington VA,
- Fowler, M., (September 15, 2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3<sup>rd</sup>), ISBN 9780321193681
- McComb, D., (2018) *Software Wasteland and The Data-Centric Revolution*
- Noy, N. F. & McGuinness, D.L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*, Retrieved from URL [https://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](https://protege.stanford.edu/publications/ontology_development/ontology101.pdf)
- Rigterink, Dr. D. T., Ames, R., Gray, Dr. A., & Doerry, Dr. N. (May 25-26, 2016). *Early-Stage Assessment of the Impacts of Next Generation Combat Power and Energy Systems on Navy Ships*, ASNE Advanced Machinery Technology Symposium, Villanova University Connelly Center, PA,
- Rumbaugh, James, Jacobson, Ivar, Gooch, & Grady, *The Unified Modeling Language Reference Manual*", ISBN 0-321-24562-8
- Snyder, D. J., Parsons, M, Brown, Dr. A., & Chalfant, J., (August 14-16, 2019 ). *Network Architecture Framework Applications with FOCUS-Compliant Ship Designs*. IEEE Electric Ship Technologies Symposium 2019, Westin Crystal City Hotel, Arlington VA,.
- Unified Modeling Language Specification*, (Jan. 2005). Version 1.4.2. ISO/IEC 19501:2005(E), formal/05-04-01, Retrieved from URL <https://www.iso.org/standard/32620.html>