Adding Simulation Capability to Early-Stage Ship Design

Matthew Ferrante, Julie Chalfant, Chryssostomos Chryssostomidis
Massachusetts Institute of Technology
Cambridge, MA, USA
chalfant@mit.edu

Blake Langland, Roger Dougal
Department of Electrical Engineering
University of South Carolina
Columbia, SC, USA

***Abstract*— The Navy's early-stage ship design tools do not currently include an inherent simulation capability. Under Navy direction, the Electric Ship Research and Development Consortium (ESRDC) has worked to develop a simulation tool that can be used to determine functionality of ship systems at the early stages of design. This paper describes the current capabilities of the simulation tool and the process and status of the efforts to integrate this tool with the Navy's design tools.**

Keywords—ship design, simulation, product meta-model

I. Introduction

The Electric Ship Research and Development Consortium (ESRDC), the Naval Surface Warfare Center Carderock Division (NSWCCD), and the Office of Naval Research (ONR) have been working together to expand the simulation capability of the Navy's early-stage ship design tools. ESRDC is developing a system simulation and analysis tool, Smart Ship Systems Design (S3D) [1], which provides inherent simulation capability and potentially provides a gateway to higher fidelity simulations. Work is ongoing to allow S3D to store and retrieve ship design data into and out of the Navy's Leading Edge Architecture for Prototyping Systems (LEAPS) database [2], thus providing a much closer integration with other early-stage design tools used by the Navy. This paper describes the current capabilities of S3D and the process for integration of S3D with LEAPS.

II. Smart Ship Systems Design (S3D)

ONR has funded the ESRDC to research and subsequently develop a collaborative, concurrent, web-based environment for the design of Navy ships. S3D is comprised of a suite of tools that support various engineering teams across multiple disciplines with the design and analysis of electrical systems, mechanical systems, and air and liquid cooling systems, as well as the arrangement of equipment in three-dimensional space from the naval architect's perspective. S3D was undertaken with several goals in mind:

- dramatically reduce the time and costs incurred during the conceptual design phase;
- reduce risk and increase the quality and efficiency of the overall ship;
- reduce the time and costs typically associated with supporting, deploying, and maintaining design and simulation software; and
- remove barriers to collaboration, particularly with respect to geographically dispersed teams, permitting users access to the design and the ability to execute simulations regardless of their computing device of choice.

Here we use the term *collaborative* to mean that the environment is open to many users (yet secure), in order that these users may participate in the design process contemporaneously and interactively.

There is a plethora of commercial tools available that provide the ability to penetrate deep into the design of a system within any one particular domain, producing detailed simulations and analyses within that domain. Today, much more so now than in the past, it is essential that such discipline-specific tools be capable of interacting with their equivalent counterparts from other domains and readily incorporating the effects of, and interactions between, these systems in order to ensure the proper integration of components. The complexity of modern systems requires deep expertise from engineers in each domain, all of which must supply the information necessary to create a complete and physically realizable system model. This presents several problems such as identifying what information should be shared between tools, ensuring the integrity of this data, determining in which format the data is to be exchanged, and understanding how such tools need to be integrated in order to ensure that a solution converges for all disciplines. Complicating things further is that these tools need to support an agile design environment that is capable of assimilating rapid design changes and capability enhancements while adhering to reasonable cost and schedule constraints. The ESRDC developed the S3D environment in order to address the issues of interoperability between multiple simulation tools while also improving agility by providing an environment that enables rapid closure of ship system designs.

The S3D environment provides a set of tools to help meet the needs of engineers in diverse fields. The environment includes a three-dimensional visualization tool, several simulation tools and solvers, an equipment catalog, collaboration tools, and document sharing. S3D includes discipline-specific workspaces as well as analytical, modeling, and visualization tools that provide a common vision for the product under design across the disciplines. The design

This work is supported by the Office of Naval Research N00014-14-1-0166 (MIT) and N00014-14-1-0165 (USC), ESRDC – Designing and Powering the Future Fleet, and MIT Sea Grant College Program under NOAA Grant Numbers NA14OAR4170077 and NA10OAR4170086.

environment also includes a database of product information and an expanding library of component models that can be used to assemble complex power, cooling and mechanical system models.

In an effort to reduce the time, cost, and complexity of installing and maintaining the software over a geographically diverse clientele, it was determined that the S3D environment would be delivered over the web via the SaaS (Software as a Service) model. This delivery model does not require the direct installation of any software on the client's computing device; therefore, this deployment model requires no IT staff to configure, install, or update the software. As new versions of the S3D design environment are developed and deployed to the cloud, the modifications are available in near real-time to all clients. Deploying S3D to the cloud offers additional advantages from the application developer's standpoint, such as the ability to dynamically scale the backend of the application via the programmatic addition of computational resources as the load increases. In this context, *cloud-based* means that the system data resides in a data storehouse that is independent of any one user's own computer, that the data is accessible by a large number of registered users over arbitrary geographic areas, and that computing power is elastic and adaptable to the computing demand.

While designing a ship there are many opportunities to make improvements to various ship components and to the performance of subsystems; the tradeoffs proposed by the various engineering teams compete with one another. When using domain-specific tools there may not be a clear rationale for pursuing one design over another because the relative impacts to the ship cannot be holistically assessed within a single discipline. An integrated and collaborative design environment can provide a better understanding of how modifications in one subsystem affect all other interconnected systems, hence helping to guide progress on the design to a more globally optimum solution. S3D has, as one of its major features, the ability to allow all the major disciplines to interact and assess in real-time the implications of a change in one subsystem across all the systems under design.

The discipline-specific tools in S3D permit an engineer to create a specific topological view of the current design from the perspective of that discipline, allowing the engineer to focus on the parts of the design that are of the most interest or significance to that discipline. Each discipline-specific tool automatically brings the pertinent set of attributes for a component to the surface while hiding other attributes that are likely not important from the user's current perspective; these hidden values can be viewed upon request. This information hiding brings better clarity and focus to the specific concerns of the particular discipline.

An important aspect of S3D is its ability to be responsive to modifications made by the users and provide appropriate feedback to all participating engineers. Immediate feedback is provided to the engineer within his or her specific discipline in that agreement between connected components at the boundaries of those components is enforced for such properties as temperature, pressure, voltage, frequency, current, etc. Violation of the constraints at the boundary or internal to the model itself will cause appropriate notifications to be raised to all users.

The architecture and implementation of a communication mechanism is important in order to ensure that the environment is capable of moving large amounts of data expediently and in a bidirectional manner. The concurrent and collaborative nature of S3D, its ability to support distributed and eventually massively parallel simulations executing within the cloud infrastructure, and to ensure compatibility with the cloud-based deployment model, led to the decision to implement the communication mechanism over standard http protocols utilizing web sockets.

As an engineer makes modifications to the design, essential information is propagated to all users currently connected to the S3D environment. For instance, if the thermal engineer determines an increase in the flow rate for a cooling pump is required and makes the necessary adjustment, the electrical engineer will immediately see an impact in the form of an increase in the electrical load for the pump. Likewise, if the naval architect determines that it is necessary to relocate coupled electrical equipment from one compartment to another, this might result in lengthening cables and, consequently, a modification to the cable impedance will result. The electrical engineer might then see a significant voltage drop across the bus upon simulation. In this way the S3D environment allows all engineers to work together in real-time and directly see the impact changes to the design have on the performance of all ship systems.

The use of 3D CAD tools to perform verification and fit of critical components is needed when laying out systems, and especially when arranging multiple systems simultaneously. The information learned and the CAD models created from these early exercises need to be shared with the larger design team to ensure the team as a whole has a common vision for what is being designed. This process is often hindered due to the distributed nature of design activities that arises within larger groups of engineering teams and due to the adoption of heterogeneous 3D CAD tools. For these reasons, the S3D environment includes a visualization tool which helps to ensure that the implications of the physical arrangement of equipment in a design can be grasped and understood quickly by all parties involved. The visualization environment includes support for readily moving between discipline-specific topological views and the universal 3D view of the ship systems. Because these discipline-specific simulation tools and the 3D visualization tool are integrated within the same design environment, the ability to readily and programmatically extract physical details of the design, such as the approximate length of cable runs, bending radius of cables, pipe runs, elevation changes in a pipe run, etc., can be reflected back into the appropriate logical schematics. Conversely, the various analyses of the simulation tools are able to provide additional information, such as the temperature of components or state and availability of certain devices, and this information can be propagated to the visualization environment. Similarly, if the electrical engineer wishes to change the selection of a gas-turbine generator set in the electrical schematic from an MT30 to an LM2500+, the 3D visualization environment will be notified and the appropriate

CAD model will be loaded. The visualization tool offers a promising medium for the distillation of large amounts of highly technical and discipline-specific information into generalizations that can be easily consumed by less technical stake holders and by those with substantially different technical backgrounds.

The philosophical approach which USC originally took when developing S3D is one in which multiple teams of engineers bring their unique expertise and design experience to bear within a web-based, collaborative, concurrent design environment. The ship design is then iteratively improved with engineers making the primary decisions while being assisted with the analyses provided by all tools, until the design converges across all disciplines.

III. Interface with Navy Design Tools

The very early stages of design within the Navy's design process are generally undertaken with a single designer or a small team, using a single-user toolset. The Advanced Ship and Submarine Evaluation Tool (ASSET) is a modular synthesis tool that produces a low-fidelity ship design based on user inputs and a set of parametric-based algorithms. The design is stored in the Leading Edge Architecture for Prototyping Systems (LEAPS) data repository, which is accessible by other Navy design tools such as the Integrated Hydrodynamics Design Environment (IHDE) for hydrodynamic analysis and Ship Hullform Characteristics Program (SHCP) for stability. None of the current early-stage design tools currently provide a system layout and simulation capability; the Navy design community seeks integration of S3D into the LEAPS toolkit to provide this capability.

Since the early-stage design tools are used on a stand-alone basis and required significant security for classified information, NSWCCD requested a version of S3D that would be operated by a single user on a single stand-alone computer, and that would use LEAPS as the native data repository.

The first step in the integration of these tools sets has already been undertaken. In order for the solvers, models, and schematic editors within the S3D environment to be leveraged with the LEAPS toolset, the dependencies on web based technology needed to be removed and the source code needed to be ported to a traditional Windows desktop application. The S3D environment leveraged SQL Server for its repository and this was also removed in anticipation of utilizing LEAPS as the repository for the ship design. There is commonality between many concepts within the LEAPS database and those developed within the S3D environment and this has helped to ease the development time required to integrate these toolsets. There are obviously differences between the two environments as well; however, LEAPS offers a convenient and nicely abstracted object model which is capable of capturing and storing the metadata and relationships that exist within the S3D environment.

There are particular pieces of S3D which have been removed in this version as these currently do not support the idea of a more automated approach to ship design. In particular, the collaborative features such as document sharing, messaging, and the need to send notifications between multiple users are no longer necessary. The first step in this integration is to produce a tool that allows users in S3D to open a ship design synthesized by ASSET that is stored within the LEAPS database, supplement this design with additional details for all required distributed systems, perform an analysis with S3D, and finally store the design back into the LEAPS database. Eventually, it is envisioned that the S3D simulation models and capabilities would be directly leveraged in a more automated way, removing the need for a more manual design process.

IV. Leading Edge Architecture for Prototyping Systems (LEAPS) Integration

LEAPS is a flexible data repository that can be used to hold data for a wide variety of systems, from ships and submarines to aircraft to bicycles and beyond. The LEAPS architecture defines a "metamodel" (LEAPS/MM), which is a set of generic classes to accommodate complex engineering representations for product modeling. For example, there are classes for geometric representation, performance behaviors, individual part definition, complex system definition, and design and analysis processes such as studies. The metamodel provides a formal hierarchy of classes that allow for the creation and management of LEAPS objects; this hierarchy is shown in Fig. 1. For example, the *Concept* class collects all the LEAPS objects used to formalize the abstraction of a particular configuration of a product or design. The *Concept* class comprises and owns LEAPS *Property*, *PropertyGroup*, *Component*, *System*, *Diagram* and *Connection* objects, and includes a *Structure* object for shapes and geometry.

The LEAPS/MM is implemented and accessible through a C++ Application Programmer's Interface (API), which provides the structure for creating, storing and manipulating instances of the class structure provided. Since the LEAPS API and S3D are written in different languages, an integration library to allow use of the C++ LEAPS API functions by a C# code such as S3D was written.

LEAPS further defines a "product metamodel," an object-oriented schemata that defines a specific category of products. The LEAPS product metamodel (LEAPS/PMM) that formalizes the definition of data for naval surface ships is known as FOCUS. This PMM provides the structure and terminology for objects pertinent to a surface ship so that programs interfacing with a model stored in LEAPS are able to identify, find and use the data in a consistent manner.

In order to integrate fully with LEAPS, the new S3D program must be able to utilize the LEAPS API, consume LEAPS objects and map them to corresponding entities in S3D, and store any necessary data in LEAPS for future use. For this to be possible, some LEAPS/MM classes must be extended or re-defined to accommodate the S3D model, and the FOCUS model must be extended to be able to store S3D data that are relevant to other tools in the LEAPS suite.

A. Object Dictionary

We investigated the baseline structure of the two databases including such things as definition of data types, definition of units, and relationship between metamodels. Table 1 lists the "gaps" identified between the two applications.

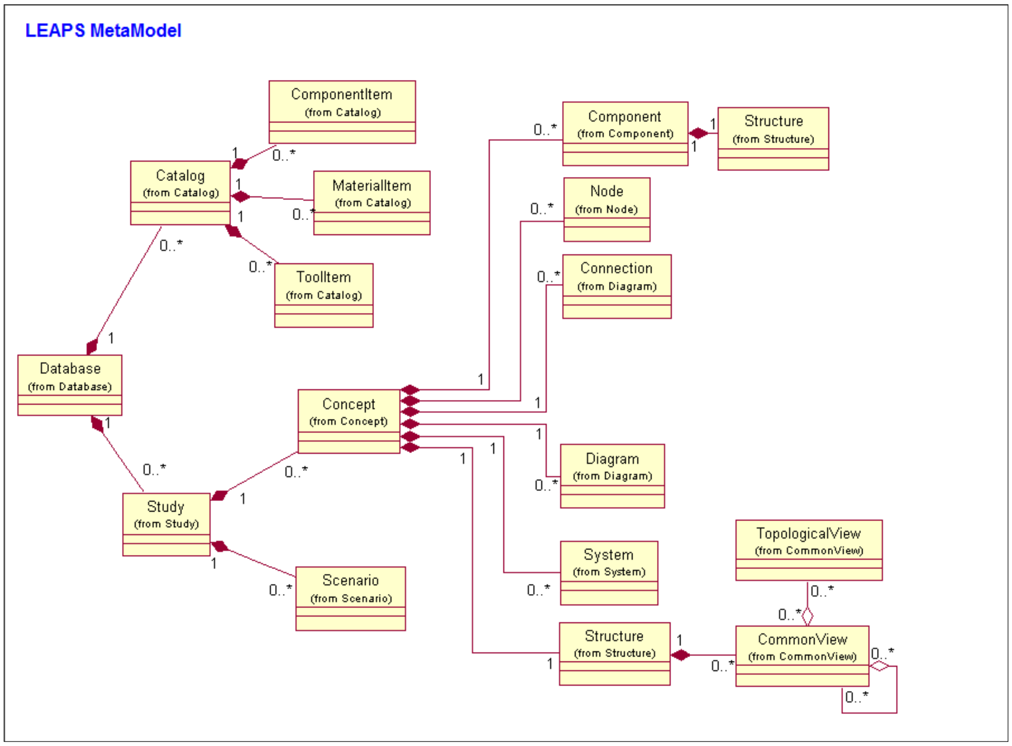

Fig 1. Diagram of the LEAPS metamodel, from [2]

Incompatibilities in data representation include the following:

- No complex number, currency or C# DateTime representations are available in LEAPS. Support for complex numbers is being added to a future version of LEAPS by NSWCCD. Currencies can be stored as RealScalars in LEAPS, and DateTimes can be stored in a string representation.
- LEAPS uses NURBS splines to represent continuous data and geometry. S3D can represent spline data as an array and interpolate between points in the array for information such as a speed-power curve or an efficiency-load curve. A LEAPS API function exists to extract spline data and represent it in such an array, and to take array data and store it in LEAPS as an object.
- Booleans are explicitly defined in S3D. They can be represented in LEAPS as a string enumeration (TRUE, FALSE).
- Units are handled very differently. In S3D, there is a metadata table of possible units. When an attribute is created and put in the list of Attribute Types, a pointer to a unit is assigned. This unit can only be changed by changing the unit pointer in the attribute type. In FOCUS, units are handled at the PMM level: each property is assigned a unit within the PMM file, and the units cannot be changed.

B. Data Dictionary

Once the baseline data types were identified and compared, the next step was to investigate the representation of those data

TABLE 1. NOTIONAL SHIP PERFORMANCE REQUIREMENTS AND DATA

LEAPS	S3D
Integer	Integer
Floating Point Number	Double
String	String
Spline	(matrices available for extracted spline data)
Url	Url
Enumeration	Enumeration
Digital (1,0) representation	Boolean
Xml	Xml
(no LEAPS equivalent)	Complex
(no LEAPS equivalent)	Currency
(no LEAPS equivalent)	DateTime
Url	FileName

types and the specific examples to achieve compatibility. We address the concepts beginning at the most basic and extending to the most complex, in the following order:

- Object comparison
- Components
- Systems
- Non-system data
- Scenario definition
- Simulation state data and results

We performed a comparison between the generic classes that make up the two application metamodels as shown in Table 2. Certain classes in the two systems have an obvious equivalence; for example, a LEAPS *Component* (an instance of a part) maps to an S3D *Equipment* object, and a LEAPS *ComponentItem* (a reusable part residing in a LEAPS *Catalog*, from which many *Component* objects may be instantiated) maps to an S3D *EquipmentType*. Other types, such as the LEAPS *CommonView*, have no direct correspondence in S3D, but can serve as containers for important S3D data to reside in the LEAPS domain. S3D terms in Table 2 denoted with an asterisk are similar but not exact matches to the corresponding LEAPS term. More detail regarding the structure of LEAPS can be found in [3].

It is important that the two applications be in agreement on the definition of these objects. While it may be obvious that an S3D Component maps to a LEAPS Equipment, the underlying data associated with a particular part may differ between the two applications. Information of this nature – names, definitions, and units of well-known data members – is captured in FOCUS. An example of the PMM description of an exhaust duct is shown in Fig. 2. In this example, the exhaust duct is the EquipmentItem, the Properties include Application, Gas Temperature, Gas Density, Mass Flow, Cross Section Area, and Design Temperature. The properties are defined with a data type, units, bounds, and a description (not shown).

TABLE 2. CORRELATION MATRIX FOR S3D/LEAPS TERMINOLOGY

LEAPS	S3D
Study	Project
Concept	Design
Catalog	Equipment Catalog*
(no LEAPS equivalent)	Equipment Type Category
Component Item	Equipment Type*
Component	Equipment Item*
Property	Attribute
Property Item	Attribute Item
System	Discipline View*
Common View	(no S3D equivalent)
Property Group	(no S3D equivalent)
Material / Material Item	(no S3D equivalent)

A Data Dictionary was created to delineate all Attributes of S3D EquipmentTypes and compare them to the Properties of LEAPS ComponentItems in order to determine what essential data are not currently captured by FOCUS, and to find inconsistencies between similarly-named characteristics which may have different uses in the two domains.

C. Additions to FOCUS

The LEAPS product meta-model (PMM) for surface vessels, FOCUS, is the defined format and relational structure for storage of ship data that enables a software tool to find needed data and to store data in a location that all other software tools operating with LEAPS can find as well. Data that meet the requirements of the FOCUS model are termed "**FOCUS-compliant**." We must formalize in FOCUS a space for the S3D data to reside so that users of other tools may safely access the data.

The LEAPS data repository and the FOCUS model are guaranteed to be backwards compatible – they may be expanded, but will not be changed in a way that makes terms obsolete or data stored in previous versions inaccessible or unusable. In order to maintain LEAPS as a streamlined, functional tool, any recommended changes to the FOCUS model or expansions to LEAPS must be fully vetted before being incorporated.

1) New Component Types

One result of the Data Dictionary effort was the identification of S3D EquipmentTypes for which no equivalent Component exists in FOCUS. The equipment in the S3D equipment catalog can be categorized as electrical, cooling and piping, HVAC, mechanical, and loads including weapons and sensors. In the first expansion of FOCUS, only electrical equipment and the loads will be added to the PMM.

2) New Properties for Existing Components

In S3D, all EquipmentTypes that are electrical in nature share a common set of Attributes pertaining to the electrical characteristics of that part – Rated Voltage, Rated Current, Current Type, etc. – to provide a complete representation of that part to the electrical simulation engine. A set of Attributes also exists for all cooling components, as does a general set of Attributes associated with all components. Our intent is to create several FOCUS-compliant PropertyGroups to formalize these Attributes in the LEAPS domain. We limited the scope of the first FOCUS expansion to electrical Attributes. A new PropertyGroup was formally defined in FOCUS, and populated with S3D Electrical Attributes. This PropertyGroup will appear in the definition of every Component which has a corresponding electrical EquipmentType in S3D, so that these objects will remain FOCUS-compliant after S3D simulations are complete.

There are other electrical Attributes that are associated with particular types of electrical equipment but are not common to all; in LEAPS these become unique Properties for those component types. For example, a motor may have an rpm curve, and a power converter may have primary and secondary side voltage types; these properties may not be applicable to other electrical equipment.

3) Tool-Specific Data Format

There will be information that a specific software tool needs to store that is not used by any other software tool; this is information necessary for the functioning of the software during operation. This information can be stored in LEAPS without being a formal part of the FOCUS model, and is termed "**tool-specific**" data. While this data needs to be stored in an organized manner so that the individual tool can locate it, it does not need to be added to the FOCUS meta-model because other tools will not need to find the data. This helps prevent cluttering the FOCUS model.

In S3D, tool-specific data include information such as the underlying simulation model associated with a particular EquipmentType object in S3D, various metadata attributes needed by the various user interface components of S3D to properly display the Equipment in the view, and metadata pertaining to individual Attributes, such as Read-Only

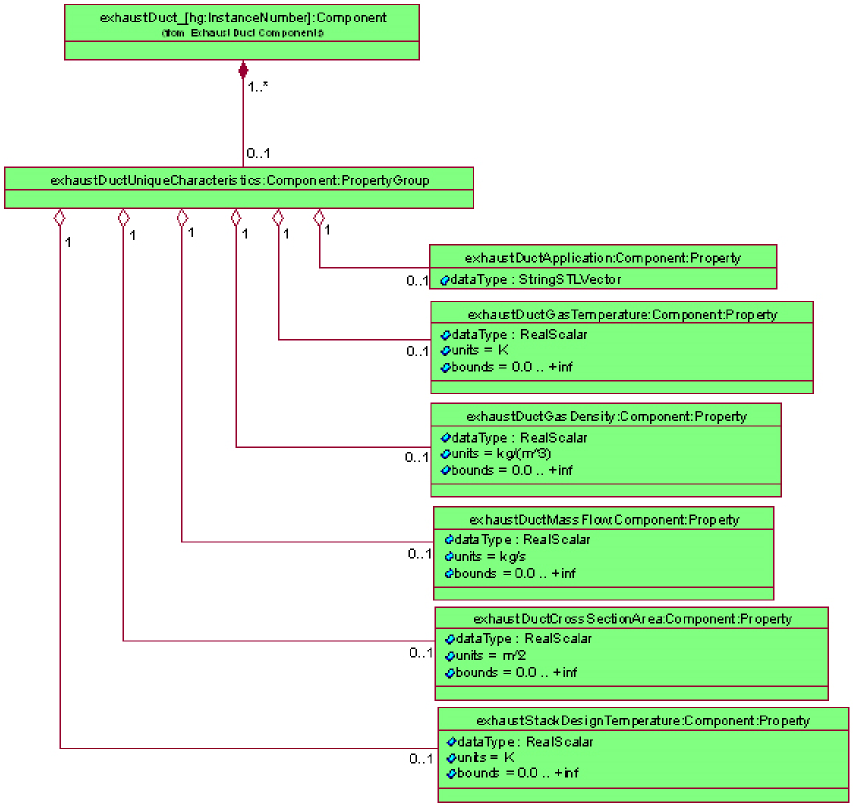

Fig. 2. Diagram of a FOCUS-defined LEAPS Component [2]

designation. Although these data are insignificant to non-S3D users, they are necessary to persist in order for S3D to properly initialize and run simulations.

4) Systems Definition Expansion

The basic philosophy of the Focus System Model is to create a standard way of describing a **system** to any tool which seeks to analyze systems. This methodology needs to be equally useful for any system analysis tool regardless of level of fidelity or focus of analysis. The Focus System Model must be sufficiently general to be able to describe systems across a wide range of disciplines, and sufficiently elastic to allow for varying levels of detail, if necessary. The model was designed to adhere to multiplex, multipartite, multislice network theory [4]. The network connection methodology between two components, using ports, terminals, exchange points, and edges, is shown in Fig. 3.

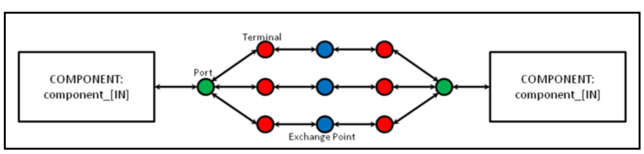

Fig. 3: Schematic showing the relationships of the various node objects, represented by solid shapes, and edges, represented by arrows, from [4].

V. Conclusions and Future Work

The Navy early-stage design tool community desired the addition of simulation capability in the early stages of design. S3D was separately funded to provide such a simulation capability in a concurrent, collaborative format. Work is ongoing to tightly couple S3D to the Navy's current design tools through replacement of the S3D database with the LEAPS data repository. Initial efforts have included documentation of properties in a data dictionary, establishment of methodology for handling systems in LEAPS, and writing a C# integration library to allow use of the C++ LEAPS API functions by a C# code. Future work will include exploration of the LEAPS geometry and views representation, documentation and persistence of simulation results, and replacement of code to allow S3D to draw directly from LEAPS.

References

[1] J. Chalfant, B. Langland, S. Abdelwahed, C. Chryssostomidis, R. Dougal, A. Dubey, T. El Mezyani, J. Herbst, T. Kiehne, J. Ordonez, S. Pish, S. Srivastava and E. Zivi, "A Collaborative Early-Stage Ship Design Environment", ESRDC Technical Reports, 2012.

[2] "LEAPS Version 4.5 Application Programmer's Interface User's Manual." Carderock Division – NSWC Design Tool Development Branch.

[3] "LEAPS Version 4.5 LEAPS Editor User's Manual." Carderock Division – NSWC Design Tool Development Branch.

[4] R. Dellsy, M. Parker, and D. Rigterink, "Multi-Scale, Interdisciplinary Systems Analysis for Naval Platforms," *ASNE Day,* 2015