Development of an Early-Stage Design Tool for Rapid of Distributed Ship Service Systems Modelling in Paramarine – A Submarine Case Study

Muhammad Mukti, UCL, London/UK, hary.mukti@ucl.ac.uk
Rachel Pawling, UCL, London/UK, r.pawling@ucl.ac.uk
David Andrews, UCL, London/UK, d.andrews@ucl.ac.uk

Abstract

The sophisticated 3D based synthesis that is enabled by the UCL Design Building Block (DBB) approach means the designer can model distributed ship service system(s) (DS3) physical entities to whatever level of detail deemed necessary well beyond the DS3 concept design level. The high flexibility of the Paramarine ship design toolset, particularly the descriptive ability provided by the DBB objects through storing data at different levels of design granularity, enables design exploration to different levels of design hierarchy. However, several drawbacks have been found in implementing such a sophisticated (fully 3-D) modelling tool in Early Stage Ship Design (ESSD). These include the effort to model or create each of the numerous features and placing them individually in the vessel's configuration. The paper presents the development of an ESSD tool that can rapidly generate a submarine early stage design with significant DS3 definition. That definition is sufficiently descriptive but still general enough to allow the level of flexibility in design exploration required at early design stages. The tool aimed to make the 3D based synthesis execution process as simple as possible so that the designer is able to manipulate the 3D architecture of the vessel and focus on important architecturally driven decision making in ESSD. An ocean going conventionally powered submarine case study was undertaken and demonstrated the capability and the flexibility of the tool.

1. Introduction

As a Physically Large and Complex System (PL&C), *Andrews (2012)*, the submarine design process encompasses various design phases and is conducted by different organisations. The design process consists of several concept, assessment or feasibility, followed by contract or project definition to fix price and check out the selected design remains balanced (especially the buoyancy and stability balance which is more demanding in submarine design than surface ship design) before proceeding to detailed design, *Andrews (1994)*. However, in the initial sizing of complex vessels, where recourse to type ship design can be overly restrictive, one crucial set of design features has traditionally been poorly addressed. This is the estimation of the weight and space demands of the various Distributed Ship Services System(s) (DS3), which is "a collection of connected components that provide a service from one or multiple sources to multiple users, via connections throughout the ship, directed towards defined functions, supporting specific operations of the vessel", *Mukti et al. (2021)*. Such an approach inhibits the ability of the concept designer to consider the impact of different DS3 options, *Andrews (2018)*.

Given the advancement of computer graphics, not utilising such technology to better synthesise DS3 in ESSD was seen to be not taking advantage of Computer-Aided Design (CAD) developments. Thus, this paper begins by outlining a proven design method utilising a sophisticated fully three-dimensional (3D) Computer-Aided Ship Design (CASD) software that could potentially accommodate both the synthesis of the whole submarine as well as that for the DS3. This is followed by investigating the modelling issues in using such a CASD tool for DS3 synthesis. Sections 4, 5, and 6 describe a new tool to mitigate the identified issues. After that, a case study demonstrates the applicability of the new tool, followed by conclusions and recommendations.

2. Computer-Aided Ship Design

In this section, a short review of a CASD approach that could potentially accommodate DS3 synthesis is provided and then the potential emergent issues using such CASD are discussed.

2.1. The UCL Design Building Block Approach

The UCL Design Building Block (DBB) approach, *Andrews and Dicks (1997),* is a proven design method and was implemented as the SURface CONcept (SURFCON) module (for both surface ships and submarines as shown in Fig.1) in the sophisticated fully three-dimensional (3D), commercial naval architectural CASD software Paramarine™, https://paramarine.qinetiq.com/products/paramarine/index.aspx, *Andrews and Pawling (2003).*

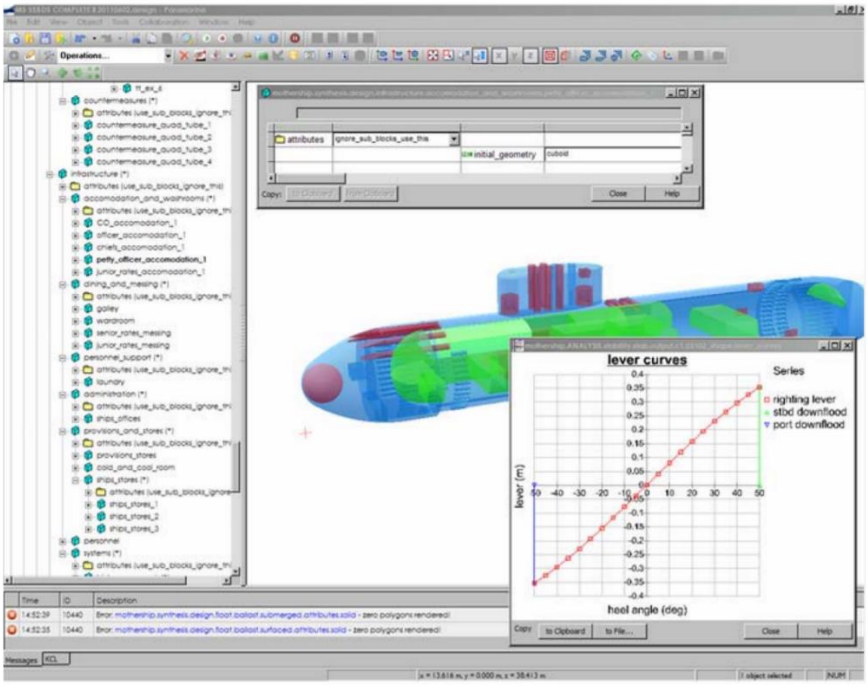


Fig.1: Screenshot of Paramarine showing interactive numerical, tabular, and graphical information in the Design Building Block objects, *Pawling and Andrews (2011)*

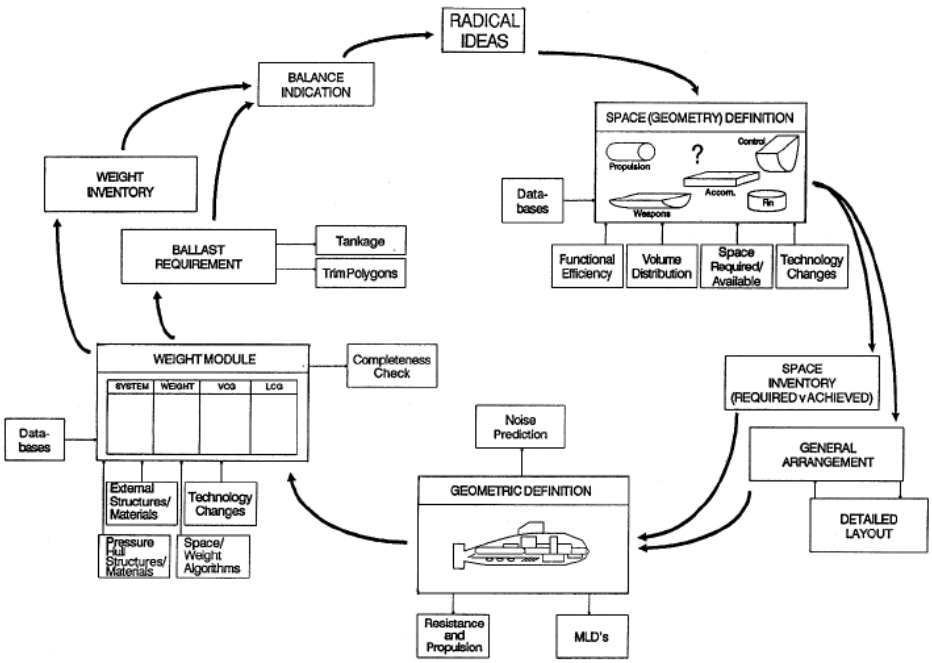


Fig.2: Logic of Design Building Block implementation to submarine design in SUBCON, *Andrews et al. (1996)*

The Implementation of the DBB approach in Paramarine™ provides an object-oriented and top-down approach that allows discrete objects to be modelled and manipulated in different levels of granularity. These objects can attach information in a form of string and numerical data, such as weight and geometry and even can be assigned different sizing algorithms, *Pawling (2007)*. It can start by developing a small number of coarse models as indicated by 'Space (Geometry) Definition' in Fig.2. These models can be based on equipment databases, including new equipment that is under development, reflecting the technology and configurational innovations implicit in commencing the process through fostering 'Radical Ideas' (top of Fig.2). As the design progresses, the coarse model of a few Super Building Blocks (SBB), may not fully populate the enclosed volume. This is then broken down into more detailed blocks as necessary as reflected in the building block design phases for surface ship design (e.g. topside and major feature design phase and super building block -based design phase), *Andrews and Pawling (2008)*. From these assembled blocks the ship design can be manipulated and assessed under a block object called a Master Building Block (MBB) defining the whole vessel characteristics, *Andrews and Pawling (2003)*, until the design is balanced, i.e., reach an acceptable performance, *Andrews (2018)*.

2.2 Automated Approaches

Since the DBB implementation has been designer-led, decisions are made by the designer, as opposed to highly automated approaches. In previous submarine design research at UCL, *Purton et al. (2015)* created an automated design tool that he called Submarine Preliminary Exploration of Requirements by Blocks (SUPERB). This uses high-level input and sizing algorithms provided by the UCL design procedure to arrive at crude numerical syntheses. The numerically balanced Pareto Front solutions are then assessed and the front 'lowered' from more detailed consideration, *Purton (2016)*. Before this UCL work, the US submarine builder, Electric Boat and US Navy's Naval Sea Systems Command (NAVSEA) also developed Submarine Concept Design (SUBCODE) using one hundred Microsoft® Excel® workbooks to automate the early stages of submarine design, *Mahonen et al. (2007)*. More recent work is the application of the packing approach model (pioneered by *van Oers (2011)* and, subsequently, *Duchateau (2016)*) for the conceptual design of submarines, *Cieraad et al. (2017)*.

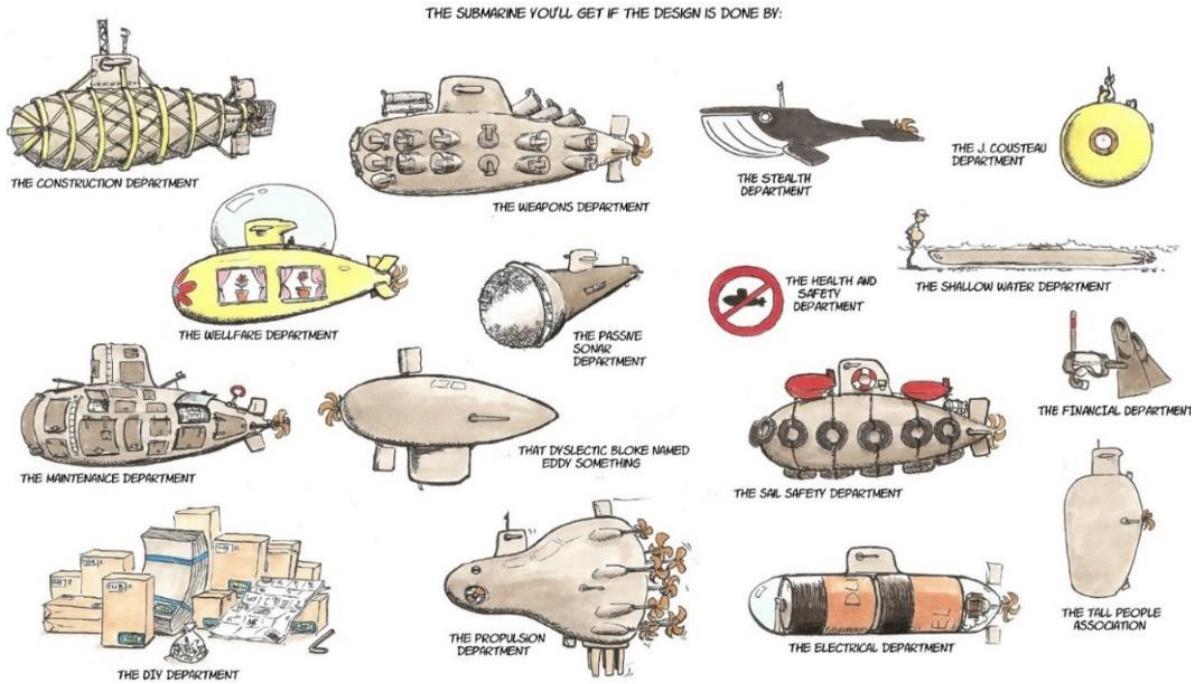


Fig.3: Submarine design for different "objectives" due to CDR Boomstra RNLN, *Duchateau (2016)*

Automated approaches hardcode design steps, many design algorithms, and their assumptions for sizing often implying, but not limited to, how the spaces are arranged within the vessel. This, in turn, makes

the software program follow several design decisions automatically every time an unbalanced condition occurs in the design. This can then allow hundreds of concept designs to be generated quickly but all based on 'hidden' configurational assumptions. Such an automated approach is consequently difficult to be assessed, i.e., is not revealed easily (if at all) to the designer and thus is a 'black-box' synthesis. The danger of such black-box approaches is that not only do they inhibit creativity and the introduction of innovations, but also could constrain the overall ship design size early in the design process. Whereas *Andrews (2011)* has strongly argued, any design solution should emerge from a proper Requirement Elucidation dialogue with requirement owners or stakeholders. Such a dialogue aims to balance different visions or objectives across multiple design stakeholders in the eventual complex vessel design (see Fig.3). This requires an approach like the UCL DBB approach that is human-centred (glass-box) rather than computer-centred (black-box) and thus architecturally driven.

2.3 Gulfs of Execution and Evaluation

Although the synthesis of the whole submarine design could have been developed using the sophisticated 3D based synthesis UCL DBB approach, there were several drawbacks in implementing such a sophisticated (fully 3D), high-fidelity, high-capability Computer-Aided Design (CAD) modelling tool in ESSD. These included the difficulties due to modelling or creating each of the numerous features and placing them individually. The latter can be considered laborious and demanding, especially if detailed modelling must be carried out after each design change and iteration, *Andrews et al. (2009)*. Such modelling effort can be referred as to the Gulfs of Execution and Evaluation, see Fig.4), which qualifies the overall level of effort required in making a system perform the desired task correctly, *Norman (2013)*. Therefore, the 3D based synthesis was then reduced to what can be called '2.5D' to allow a simpler architecturally oriented design tool to be developed in-house for specifically surface ship research and education referred as to the UCL JavaScript layout exploration tool, *Pawling et al. (2015), Kouriampalis et al. (2021).* In the current paper, an alternative solution was developed without creating a further separate or standalone design tool like the UCL Javascript tool. That tool sacrificed many advantages of using 3D based synthesis and 3D informed dialogue, which Paramarine facilitated and was seen to be necessary for exploring the submarine DS3 in ESSD.

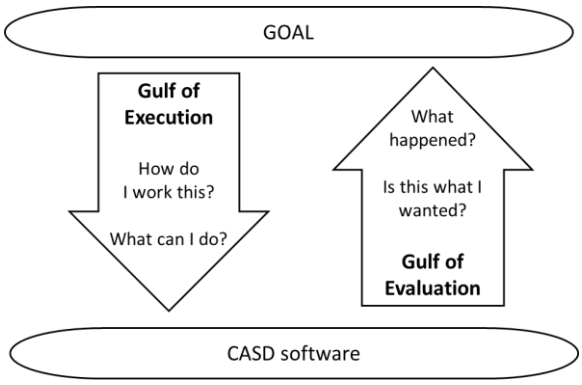


Fig.4: Gulfs of Execution and Evaluation, *Norman (2013)*

2.4 Initial Investigation

The advantages of using the sophisticated 3D based synthesis UCL DBB approach in SURFCON Paramarine for DS3 were investigated, *Mukti et al. (2019),* using an SSK example, which was selected based on a previous study, *Mukti and Randall (2017)*. *Mukti et al. (2021)* presented an early version of DS3 synthesis retaining design flexibility and avoiding bottom out the preferred design. It utilised "Submarine Flow Optimisation" SUBFLOW for DS3 together combined with the UCL Design Building Block approach, *Andrews et al.* (1996), Fig.5. That implementation revealed the technical issues in integrating the network-based sizing approach with the submarine design process using SURFCON Paramarine (i.e., a significant amount of Gulfs of Execution and Evaluation was required when using both approaches), which could inhibit exploring DS3 options in ESSD. This is discussed further in the following section.

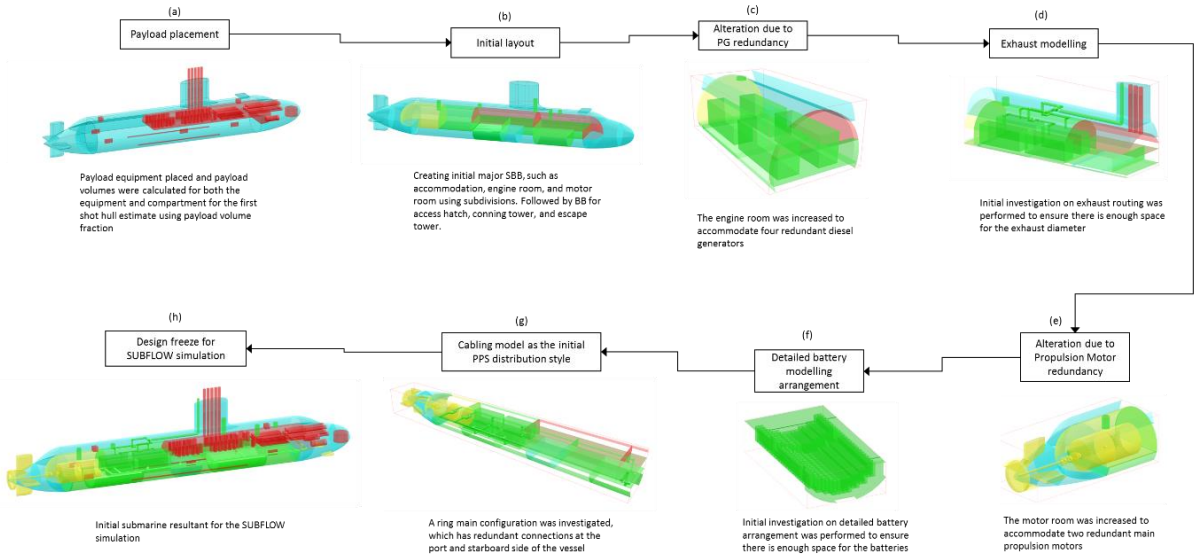


Fig.5: Summary of Power and Propulsion System (PPS) Case Study, *Mukti (2022)*

3. Development of the Approach

Normally, to create an object in Paramarine, the designer requires five steps as illustrated in Fig.6 (left), click object, click insert, click the type of the placeholders, click the rename column, and then click OK. Other possible approaches exist e.g., copy, and paste from a pre-defined template. Still, it required at least three steps (e.g., to rename each of the relevant objects). This process was considered to inhibit the important benefit of the UCL DBB approach, since many clicks would be required if one design consisted of hundreds of objects where design exploration aims to explore multiple designs.

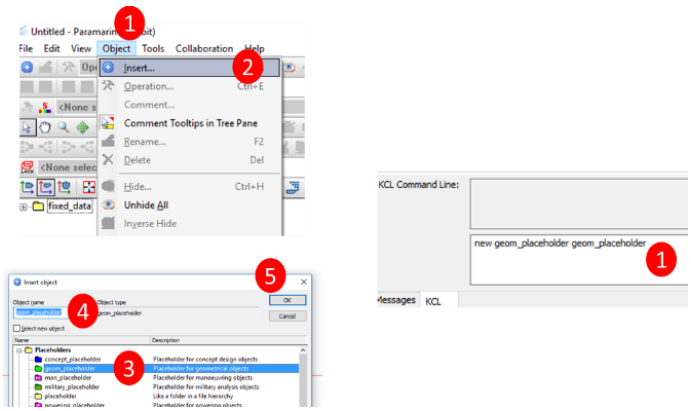


Fig.6: Illustrative modelling effort in Paramarine showing the manual process (left) and the use of a single line of KCL codes (right)

Fortunately, Paramarine has an alternative modelling approach using a KCL line as shown in Fig.6 (right). Only one step, one line of KCL command is required to create an object in Paramarine. This greatly reduced the effort of modelling in Paramarine. Now the question would be how to utilise this feature without constraining the design and retaining the benefits from the UCL DBB approach. Therefore, several programs in Excel were created to automate the modelling effort using KCL lines.

This was first tested to automate the modelling effort of a refined physical model of a submarine case study, *Mukti et al. (2021)*. The comparison is illustrated in Fig.7. Fig.7 (left) shows the theoretical modelling required to model 277 building blocks for DS3 components of that submarine case study. Since each DS3 component would require an equipment object (5 clicks), a geometric object (5 clicks), inserted as a SURFCON building block object and modelled DS3 routings (100 clicks) this suggests if there are 277 DS3 components building blocks the theoretical effort required would be some 30,000

clicks for a single design. Meanwhile, following the steps number in Fig.7 (right), all numerical input data defined in spreadsheet programs could be converted to 12,780 KCL lines within 40-50 s.

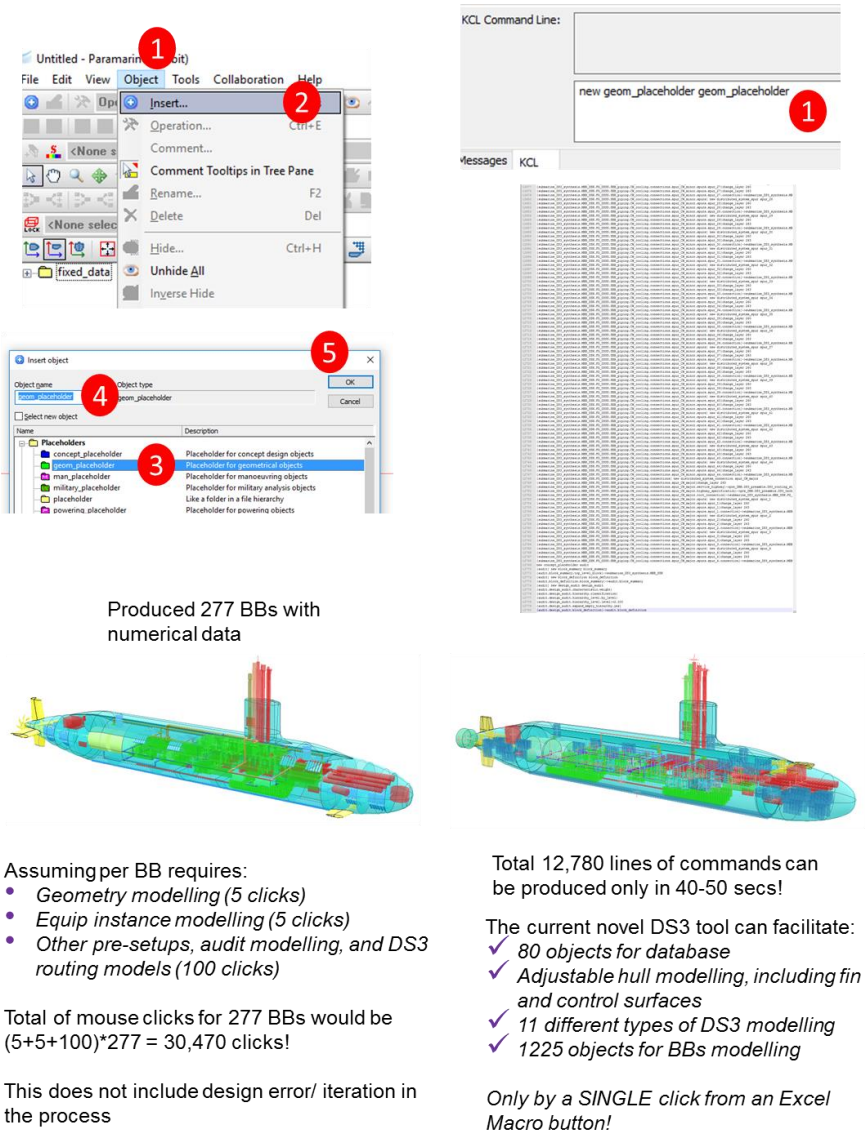


Fig.7: Theoretical modelling effort of a submarine case study, *Mukti et al. (2021)* in Paramarine showing the manual process (left) and the use of KCL macro line (right)

Given the use of Excel and KCL can potentially alter the Gulf of Execution in modelling DS3 in Paramarine, the next section outlined a new approach utilising such tools.

4. The Network Block Approach

As described in the previous section, the procedure to model a DS3 component as a Design Building Block object, including connecting it to another Design Building Block, required at least 40 clicks. This meant, if a design consists of 50 pairs of connected building block objects, the modelling process would require at least 2000 clicks. This would not include any design changes or alterations to the modelling. Such a laborious process is depicted as the "bottleneck" process in red in Fig.8 and considered as the 'repetitive/routine task' for the Gulfs of Execution and Evaluation, Fig.4, in modelling DS3 in ESSD. This could then distract the designer from the benefit of the UCL Design Building Block implementation for DS3 synthesis in Paramarine.

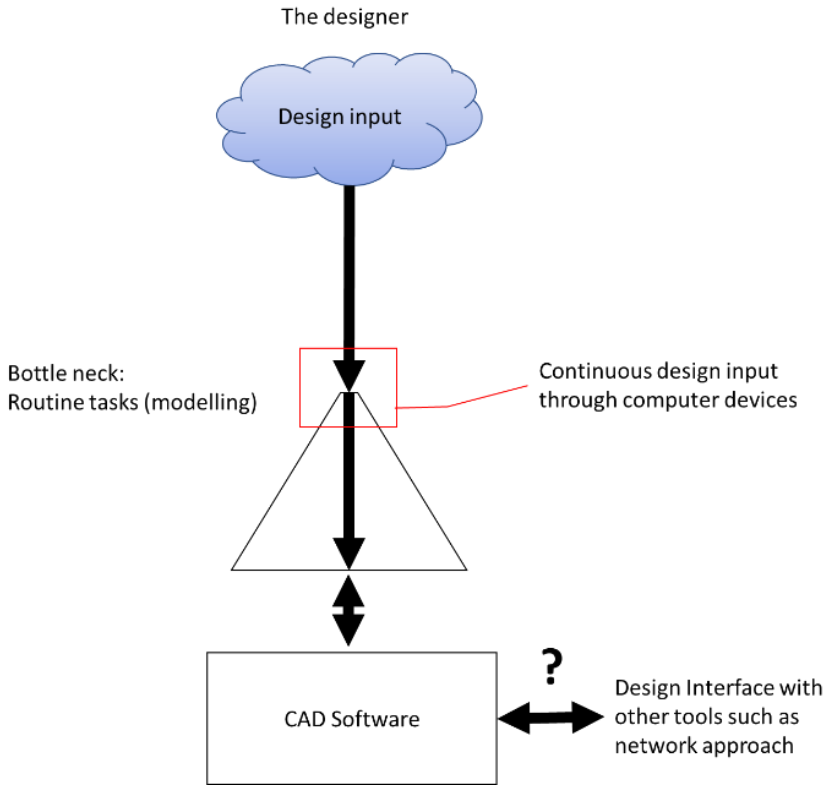


Fig.8: Data flow problem, showing the input and the bottleneck problem in blue and red, respectively

It was found in the design undertaken in this research, the reliance on the software was increased once the design had been developed into a sufficient level of details for DS3 synthesis. This is indicated by the question mark in Fig.8. For example, the need to extract specific design data to other tools, as demonstrated *Mukti et al. (2021)*, Fig.5, was found to be time-consuming. Multiple clicks were required including tracing the location of the relevant DBB objects in a specific DBB hierarchy within hundreds of DBB objects and putting the data manually into MATLAB. This reliance may not be an issue if, for example, the analysis is not directly part of the design synthesis process and thus the process is not iterative, i.e., it would not be necessary to feed the data back to the Paramarine ship synthesis process simultaneously. However, in the proposed approach, the SUBFLOW network activity was significant in the DS3 synthesis process, meaning frequent data transfer and so the speed of data flow between design tools mattered. Such rapidity of transfer was seen to be essential to ensure the designer could perform the many iterations required to design DS3 physically and logically (see DS3 framework, *Brefort et al. (2018)*). Thus, the manual process, as demonstrated in Fig.5, was considered prohibitively long and thus not readily plausible for the design to incorporate sufficient key DS3 components in ESSD without a new approach.

The new approach, termed Network Block Approach (NBA), consisted of frameworks, methods and design tools that employed a strategy to 'intercept' data flow before being inputted to Paramarine and use of Excel spreadsheet input (as shown in green in Fig.9). Although Paramarine already has an interface with Excel as an object, using this Excel object in Paramarine makes the Excel file embedded in the Paramarine file, which complicates the MATLAB to read such an embedded file for network analysis. Using Excel with Paramarine is also not novel, *Fiedel (2011)*, *Thurkins (2012)*, *Jurkiewicz et al. (2013)*, but using Excel to combine the UCL DBB approach with the SUBFLOW simulation for DS3 synthesis has not been done before. The NBA was not just an Excel tool, it comes with extensive frameworks and methods, *Mukti et al. (2022)*, that leverage and sit in the gap between the benefits of the Paramarine 3D based synthesis tool and the SUBFLOW network-based DS3 synthesis.

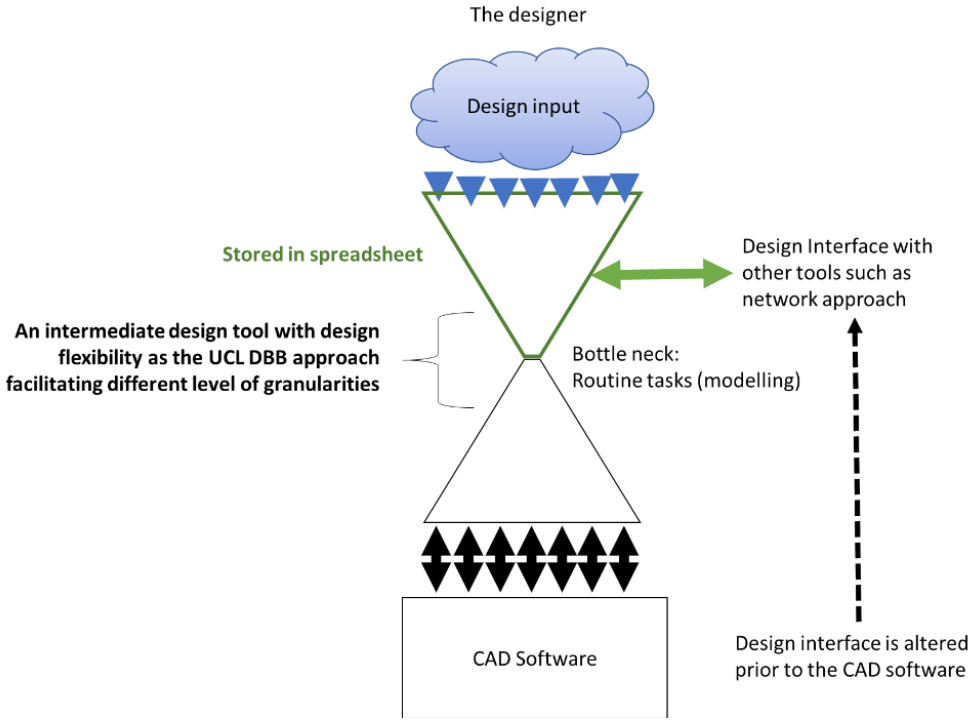


Fig.9: A proposed strategy for data flow adopted by the Network Block Approach, showing the bottleneck issue in Fig.8 can be mitigated by the spreadsheet tool in green

With the proposed approach, the designer can define design data in the spreadsheet program instead of inserting manually into Paramarine. Thus, the necessary data has been converted to thousands of lines of code, which can be more than 20,000 lines of 'Kernel Command Language' (KCL) lines. Paramarine can then automatically produce objects necessary for any DS3 synthesis, based on such KCL lines. This has been shown to save days of laborious modelling in Paramarine and unlocked the possibility for employing a new approach, such as the network-based DS3 synthesis using MATLAB. This was achieved without losing the benefits of a 3D architecturally centred submarine and DS3 synthesis and the 3D informed dialogue that SURFCON Paramarine provides. Since the design data was readily available in the spreadsheet environment, this was transferred to MATLAB with ease, unlike the manual procedure in the first pre-NBA implementation, *Mukti et al. (2021)*.

Table I: Summary of programs in the Network Block Approach

Program	**Description**	**Function**
MMP	Main Menu Program	Execution menu to compile all programs
DPP	Design Preamble Program	Hardcoded design setup
DAP	Design Analysis Program	Hardcoded analysis setup
HGP	Hull Granularity Program	Input for hull size
VGP	Volume Granularity Program	Input for spaces
WGP	Weight Granularity Program	Input for weight
EDP	Equipment Database Program	Input for equipment data
CGP	Component Granularity Program	Input for DS3 components for arrangement and SUBFLOW
SPP	System Preamble Program	Input for DS3 connections
SCP	System Connection Program	Input for DS3 connection and SUBFLOW

The programs in the NBA are listed in Table I. The Main Menu Program (MMP) is a menu to execute all the programs in the NBA with a single 'click'. The MMP was also connected to the Design Preamble Program (DPP) and the Design Analysis Program (DAP). The DPP and DAP were hardcoded KCL

scripts for automatically setting up the analytical capability available in the Paramarine system, including the audit function. The application and description of the programs in Table I within the NBA are discussed in next the section.

5. Submarine Case Study

To test whether the new tool could rapidly capture the style choices of DS3 at component granularity level and could be validated with available data, a case study was developed with the payload and style choices akin to the ocean-going 2500 tonne generic submarine extracted from the database used in the annual UCL submarine design exercise, *UCL-NAME (2014)*. This case study is described in more detail in *Mukti et al. (2022)*. The output of the programs is summarised in Fig.10, which shows how the output of each program is integrated into the whole submarine design. The inputs required for each program in the case study is now described in the following subsections.

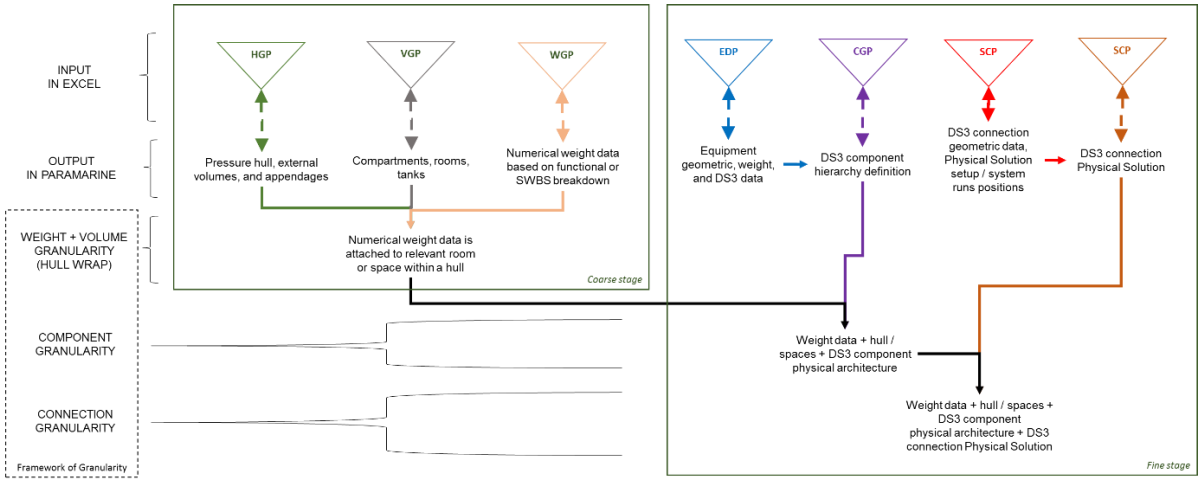


Fig.10: Output summary of the programs in the Network Block Approach; see Table I for acronyms

5.1 Main Menu Program

The Main Menu Program (MMP) was developed based on the macro interface that Paramarine provided. It contains several macro buttons: to open software; to open a Paramarine file; to build a KCL script; and to generate the KCL script from all programs (see Fig.11 for compilation sequence).

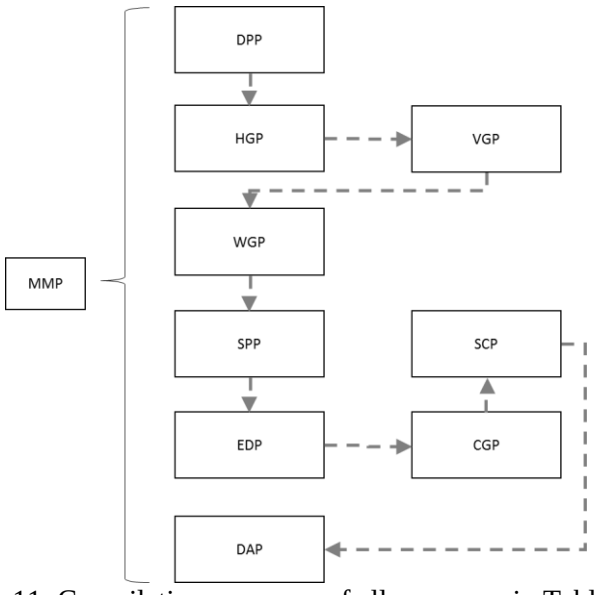


Fig.11: Compilation sequence of all programs in Table I

The MMP works closely with the Design Analysis Program (DAP) and the Design Preamble Program (DPP), which hardcoded the Gulf of Execution for performing necessary naval architectural analyses in Paramarine. The output of DPP is called DPPO (output) consisting of:

- weight group classifications (UCL SUB Weight Groups), *UCL-NAME (2014)*
- consumables (seawater, freshwater/ diesel oil, lube oil, LOX)
- ship conditions (surfaced or submerged)
- crew types (not used)
- other characteristics, e.g., costs

4.2 Hull Granularity Program

The Hull Granularity Program (HGP) provides a scalable submarine hull configuration with a specific chosen style, *Andrews (2021)*, which is a single hull with a casing configuration, Fig.12. Any different major style will require a new HGP. To develop a new HGP, one can first manually model the submarine in Paramarine and then create the macro script based on such models.

Style decisions			
PH	one		
PH configuration	single hull		
description	Input	unit (Default)	Acronym (in paramarine)
cylinder length	41.40	m	lcyl
pressure hull diameter	7.60	m	dph
aft end length	12.08	m	lae
forward end length	11.40	m	lfe
casing height	1	m	hcas
longitudinal fin location	28	%	fin loc x
fin length scale	100	%	lfin
fin beam scale	100	%	bfin
fin height scale	90	%	hfin
longitudinal aft control surfaces location	95	%	acs loc x
aft control surfaces length scale	200	%	lacs
aft control surfaces beam scale	200	%	bacs
aft control surface height scale	350	%	hacs
rudder bottom height scale	250	%	hrb
longitudinal fwd control surfaces location	80	%	fcs loc x
vertical fwd control surfaces location	-40	%	fcs loc z
fwd control surfaces length scale	150	%	lfcs
fwd control surfaces beam scale	275	%	bfcs
fwd control surfaces height scale	150	%	hfcs

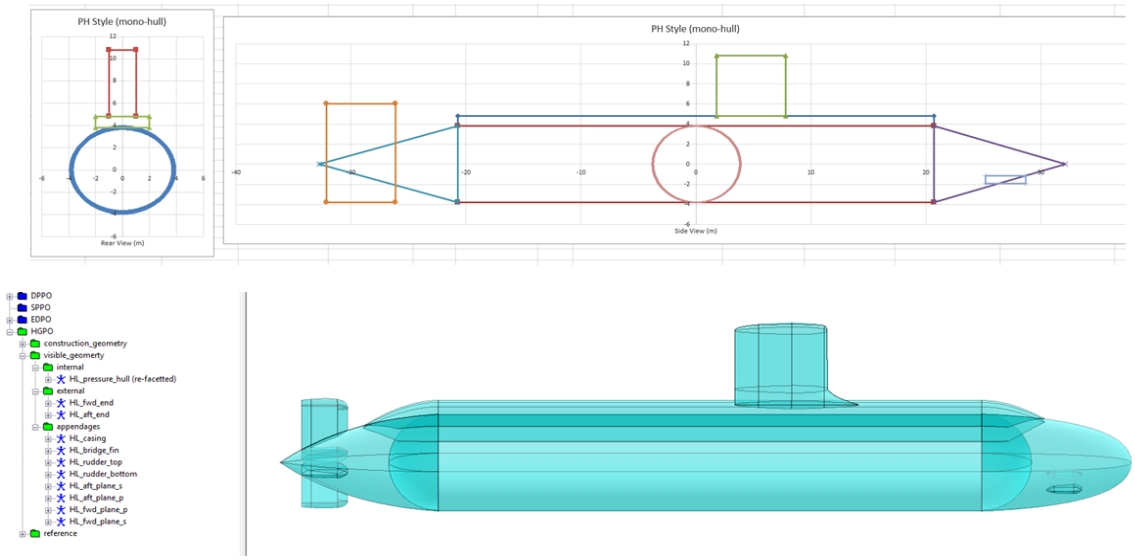
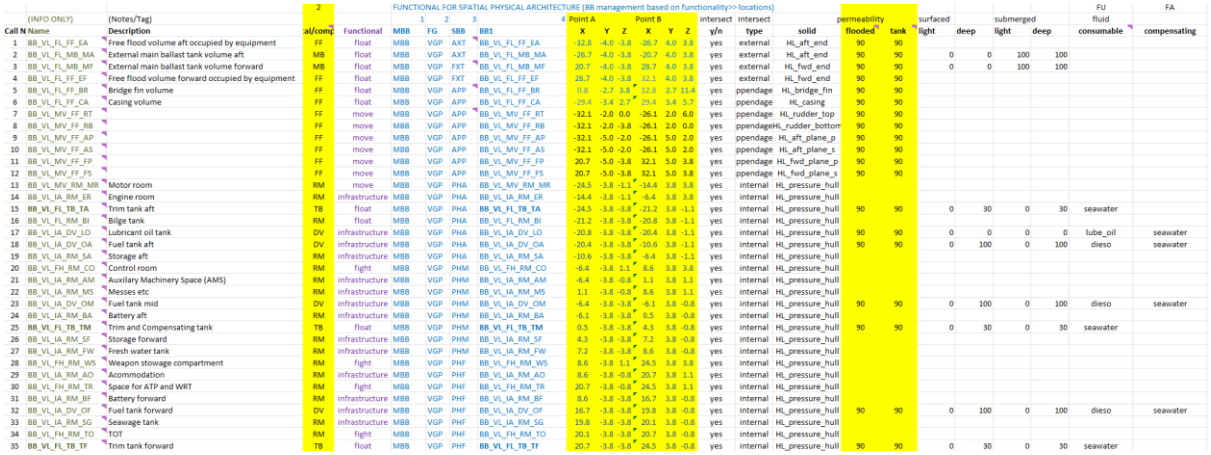


Fig.12: Layout of the HGP showing the input in Excel (top) and the output in Paramarine (bottom)

4.3 Volume Granularity Program

The Volume Granularity Program (VGP) consists of inputs to define spaces on the vessel as given in Fig.13. The building blocks for spaces are defined based on names, BB hierarchy (to level 4), two points (A and B) defining the boundary of the blocks, location of the space relative to the hull model defined in VGP, tank definition. This spreadsheet layout reveals the input of the case study reached up to 800 inputs (35 by 23), which included “string” data input as well as numerical data input.

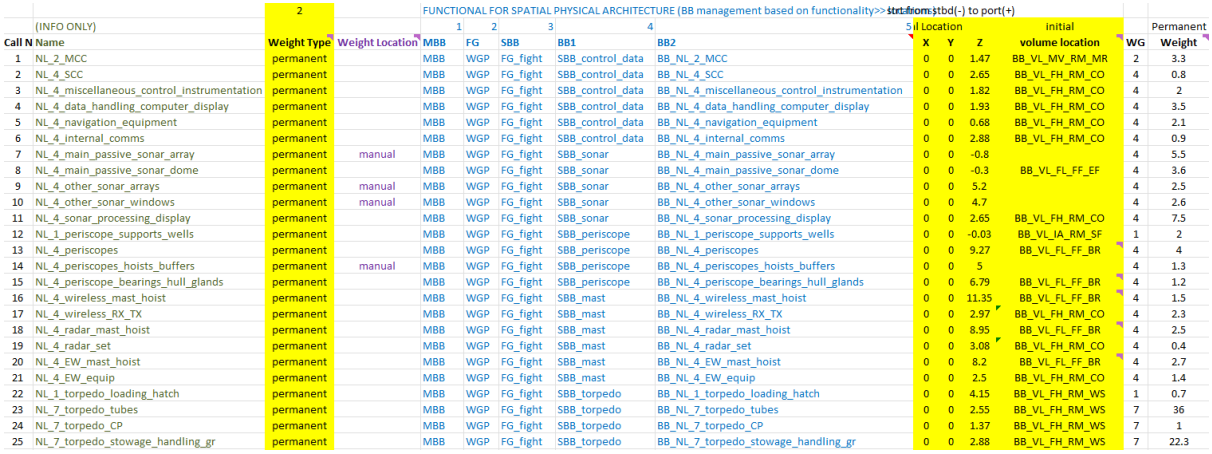


	(INFO ONLY)	(Notes/Tag)	2		FUNCTIONAL FOR SPATIAL PHYSICAL ARCHITECTURE (BB management based on functionality>> locations)																		FU	FA
					1	2	3	4	Point A			Point B						permeability		surfaced		submerged		fluid	
Call N	Name	Description	al/comp	Functional	MBB	FG	SBB	BB1	X	Y	Z	X	Y	Z	y/n	intersect	intersect	solid	flooded	tank	light	deep	light	deep	consumable	compensating
1	BB_VL_FL_FF_EA	Free flood volume aft occupied by equipment	FF	float	MBB	VGP	AXT	BB_VL_FL_FF_EA	-32.8	-4.0	-3.8	-26.7	4.0	3.8	yes	external	HL_aft_end		90	90						
2	BB_VL_FL_MB_MA	External main ballast tank volume aft	MB	float	MBB	VGP	AXT	BB_VL_FL_MB_MA	-26.7	-4.0	-3.8	-20.7	4.0	3.8	yes	external	HL_aft_end		90	90	0	0	100	100		
3	BB_VL_FL_MB_MF	External main ballast tank volume forward	MB	float	MBB	VGP	FXT	BB_VL_FL_MB_MF	20.7	-4.0	-3.8	28.7	4.0	3.8	yes	external	HL_fwd_end		90	90	0	0	100	100		
4	BB_VL_FL_FF_EF	Free flood volume forward occupied by equipment	FF	float	MBB	VGP	FXT	BB_VL_FL_FF_EF	28.7	-4.0	-3.8	32.1	4.0	3.8	yes	external	HL_fwd_end		90	90						
5	BB_VL_FL_FF_BR	Bridge fin volume	FF	float	MBB	VGP	APP	BB_VL_FL_FF_BR	0.8	-2.7	3.8	12.8	-2.7	11.4	yes	ppendage	HL_bridge_fin		90	90						
6	BB_VL_FL_FF_CA	Casing volume	FF	float	MBB	VGP	APP	BB_VL_FL_FF_CA	-29.4	-3.4	2.7	29.4	3.4	5.7	yes	ppendage	HL_casing		90	90						
7	BB_VL_MV_FF_RT		FF	move	MBB	VGP	APP	BB_VL_MV_FF_RT	-32.1	-2.0	0.0	-26.1	2.0	6.0	yes	ppendage	HL_rudder_top		90	90						
8	BB_VL_MV_FF_RB		FF	move	MBB	VGP	APP	BB_VL_MV_FF_RB	-32.1	-2.0	-3.8	-26.1	2.0	0.0	yes	ppendage	HL_rudder_bottom		90	90						
9	BB_VL_MV_FF_AP		FF	move	MBB	VGP	APP	BB_VL_MV_FF_AP	-32.1	-5.0	-2.0	-26.1	5.0	2.0	yes	ppendage	HL_aft_plane_p		90	90						
10	BB_VL_MV_FF_AS		FF	move	MBB	VGP	APP	BB_VL_MV_FF_AS	-32.1	-5.0	-2.0	-26.1	5.0	2.0	yes	ppendage	HL_aft_plane_s		90	90						
11	BB_VL_MV_FF_FP		FF	move	MBB	VGP	APP	BB_VL_MV_FF_FP	20.7	-5.0	-3.8	32.1	5.0	3.8	yes	ppendage	HL_fwd_plane_p		90	90						
12	BB_VL_MV_FF_FS		FF	move	MBB	VGP	APP	BB_VL_MV_FF_FS	20.7	-5.0	-3.8	32.1	5.0	3.8	yes	ppendage	HL_fwd_plane_s		90	90						
13	BB_VL_MV_RM_MR	Motor room	RM	move	MBB	VGP	PHA	BB_VL_MV_RM_MR	-24.5	-3.8	-1.1	-14.4	3.8	3.8	yes	internal	HL_pressure_hull									
14	BB_VL_IA_RM_ER	Engine room	RM	infrastructure	MBB	VGP	PHA	BB_VL_IA_RM_ER	-14.4	-3.8	-1.1	-6.4	3.8	3.8	yes	internal	HL_pressure_hull									
15	BB_VL_FL_TB_TA	Trim tank aft	TB	float	MBB	VGP	PHA	BB_VL_FL_TB_TA	-24.5	-3.8	-3.8	-21.2	3.8	-1.1	yes	internal	HL_pressure_hull		90	90	0	30	0	30	seawater	
16	BB_VL_FL_RM_BI	Bilge tank	RM	float	MBB	VGP	PHA	BB_VL_FL_RM_BI	-21.2	-3.8	-3.8	-20.8	3.8	-1.1	yes	internal	HL_pressure_hull									
17	BB_VL_IA_DV_LO	Lubricant oil tank	DV	infrastructure	MBB	VGP	PHA	BB_VL_IA_DV_LO	-20.8	-3.8	-3.8	-20.4	3.8	-1.1	yes	internal	HL_pressure_hull		90	90	0	0	0	0	lube_oil	seawater
18	BB_VL_IA_DV_OA	Fuel tank aft	DV	infrastructure	MBB	VGP	PHA	BB_VL_IA_DV_OA	-20.4	-3.8	-3.8	-10.0	3.8	-1.1	yes	internal	HL_pressure_hull		90	90	0	100	0	100	dieso	seawater
19	BB_VL_IA_RM_SA	Storage aft	RM	infrastructure	MBB	VGP	PHA	BB_VL_IA_RM_SA	-10.6	-3.8	-3.8	-6.4	3.8	-1.1	yes	internal	HL_pressure_hull									
20	BB_VL_FH_RM_CO	Control room	RM	fight	MBB	VGP	PHM	BB_VL_FH_RM_CO	-6.4	-3.8	1.1	8.6	3.8	3.8	yes	internal	HL_pressure_hull									
21	BB_VL_IA_RM_AM	Auxilary Machinery Space (AMS)	RM	infrastructure	MBB	VGP	PHM	BB_VL_IA_RM_AM	-6.4	-3.8	-0.8	1.1	3.8	1.1	yes	internal	HL_pressure_hull									
22	BB_VL_IA_RM_MS	Messes etc	RM	infrastructure	MBB	VGP	PHM	BB_VL_IA_RM_MS	1.1	-3.8	-0.8	8.6	3.8	1.1	yes	internal	HL_pressure_hull									
23	BB_VL_IA_DV_OM	Fuel tank mid	DV	infrastructure	MBB	VGP	PHM	BB_VL_IA_DV_OM	-6.4	-3.8	-3.8	-6.1	3.8	-0.8	yes	internal	HL_pressure_hull		90	90	0	100	0	100	dieso	seawater
24	BB_VL_IA_RM_BA	Battery aft	RM	infrastructure	MBB	VGP	PHM	BB_VL_IA_RM_BA	-6.1	-3.8	-3.8	0.5	3.8	-0.8	yes	internal	HL_pressure_hull									
25	BB_VL_FL_TB_TM	Trim and Compensating tank	TB	float	MBB	VGP	PHM	BB_VL_FL_TB_TM	0.5	-3.8	-3.8	4.1	3.8	-0.8	yes	internal	HL_pressure_hull		90	90	0	30	0	30	seawater	
26	BB_VL_IA_RM_SF	Storage forward	RM	infrastructure	MBB	VGP	PHM	BB_VL_IA_RM_SF	4.3	-3.8	-3.8	7.2	3.8	-0.8	yes	internal	HL_pressure_hull									
27	BB_VL_IA_RM_FW	Fresh water tank	RM	infrastructure	MBB	VGP	PHM	BB_VL_IA_RM_FW	7.2	-3.8	-3.8	8.6	3.8	-0.8	yes	internal	HL_pressure_hull									
28	BB_VL_FH_RM_WS	Weapon stowage compartment	RM	fight	MBB	VGP	PHF	BB_VL_FH_RM_WS	8.6	-3.8	1.1	24.5	3.8	3.8	yes	internal	HL_pressure_hull									
29	BB_VL_IA_RM_AO	Acommodation	RM	infrastructure	MBB	VGP	PHM	BB_VL_IA_RM_AO	8.6	-3.8	-0.8	20.7	3.8	1.1	yes	internal	HL_pressure_hull									
30	BB_VL_FH_RM_TR	Space for ATP and WRT	RM	fight	MBB	VGP	PHF	BB_VL_FH_RM_TR	20.7	-3.8	-0.8	24.5	3.8	1.1	yes	internal	HL_pressure_hull									
31	BB_VL_IA_RM_BF	Battery forward	RM	infrastructure	MBB	VGP	PHF	BB_VL_IA_RM_BF	8.6	-3.8	-3.8	16.7	3.8	-0.8	yes	internal	HL_pressure_hull									
32	BB_VL_IA_DV_OF	Fuel tank forward	DV	infrastructure	MBB	VGP	PHF	BB_VL_IA_DV_OF	16.7	-3.8	-3.8	19.8	3.8	-0.8	yes	internal	HL_pressure_hull		90	90	0	100	0	100	dieso	seawater
33	BB_VL_IA_RM_SG	Seawage tank	RM	infrastructure	MBB	VGP	PHF	BB_VL_IA_RM_SG	19.8	-3.8	-3.8	20.1	3.8	-0.8	yes	internal	HL_pressure_hull									
34	BB_VL_FH_RM_TO	TOT	RM	fight	MBB	VGP	PHF	BB_VL_FH_RM_TO	20.1	-3.8	-3.8	20.7	3.8	-0.8	yes	internal	HL_pressure_hull									
35	BB_VL_FL_TB_TF	Trim tank forward	TB	float	MBB	VGP	PHF	BB_VL_FL_TB_TF	20.7	-3.8	-3.8	24.5	3.8	-0.8	yes	internal	HL_pressure_hull		90	90	0	30	0	30	seawater	

Fig.13: Layout of the VGP showing major inputs required in defining spaces on the vessel

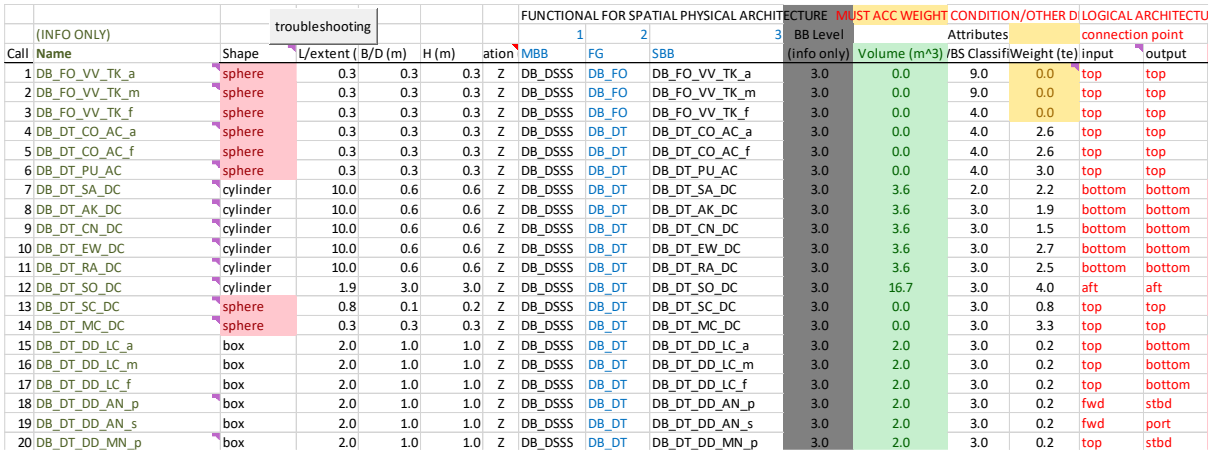
4.4 Weight Granularity Program

The Weight Granularity Program (WGP) defines numerical weight on the vessel, Fig.14. This consists of naming convention to reflect the Weight Group (WG) number, weight location ("manual" if it is defined in x, y, z coordinates), building block hierarchy (to level 5), volume location defined in the Volume Granularity Program (VGP), and the numerical weight data. The number of inputs in the WGP for the submarine case study was about 1800 inputs, assuming there are 10 inputs for each weight.



	(INFO ONLY)		2		FUNCTIONAL FOR SPATIAL PHYSICAL ARCHITECTURE (BB management based on functionality>> starting from stbd(-) to port(+)																			
						1	2	3	4	5	l Location			initial										Permanent
Call N	Name		Weight Type	Weight Location	MBB	FG	SBB	BB1	BB2		X	Y	Z	volume location	WG	Weight								
1	NL_2_MCC		permanent		MBB	WGP	FG_fight	SBB_control_data	BB_NL_2_MCC		0	0	1.47	BB_VL_MV_RM_MR	2	3.3								
2	NL_4_SCC		permanent		MBB	WGP	FG_fight	SBB_control_data	BB_NL_4_SCC		0	0	2.65	BB_VL_FH_RM_CO	4	0.8								
3	NL_4_miscellaneous_control_instrumentation		permanent		MBB	WGP	FG_fight	SBB_control_data	BB_NL_4_miscellaneous_control_instrumentation		0	0	1.82	BB_VL_FH_RM_CO	4	2								
4	NL_4_data_handling_computer_display		permanent		MBB	WGP	FG_fight	SBB_control_data	BB_NL_4_data_handling_computer_display		0	0	1.93	BB_VL_FH_RM_CO	4	3.5								
5	NL_4_navigation_equipment		permanent		MBB	WGP	FG_fight	SBB_control_data	BB_NL_4_navigation_equipment		0	0	0.68	BB_VL_FH_RM_CO	4	2.1								
6	NL_4_internal_comms		permanent		MBB	WGP	FG_fight	SBB_control_data	BB_NL_4_internal_comms		0	0	2.88	BB_VL_FH_RM_CO	4	0.9								
7	NL_4_main_passive_sonar_array		permanent	manual	MBB	WGP	FG_fight	SBB_sonar	BB_NL_4_main_passive_sonar_array		0	0	-0.8		4	5.5								
8	NL_4_main_passive_sonar_dome		permanent		MBB	WGP	FG_fight	SBB_sonar	BB_NL_4_main_passive_sonar_dome		0	0	-0.3	BB_VL_FL_FF_EF	4	3.6								
9	NL_4_other_sonar_arrays		permanent	manual	MBB	WGP	FG_fight	SBB_sonar	BB_NL_4_other_sonar_arrays		0	0	5.2		4	2.5								
10	NL_4_other_sonar_windows		permanent	manual	MBB	WGP	FG_fight	SBB_sonar	BB_NL_4_other_sonar_windows		0	0	4.7		4	2.6								
11	NL_4_sonar_processing_display		permanent		MBB	WGP	FG_fight	SBB_sonar	BB_NL_4_sonar_processing_display		0	0	2.65	BB_VL_FH_RM_CO	4	7.5								
12	NL_1_periscope_supports_wells		permanent		MBB	WGP	FG_fight	SBB_periscope	BB_NL_1_periscope_supports_wells		0	0	-0.03	BB_VL_IA_RM_SF	1	2								
13	NL_4_periscopes		permanent		MBB	WGP	FG_fight	SBB_periscope	BB_NL_4_periscopes		0	0	9.27	BB_VL_FL_FF_BR	4	4								
14	NL_4_periscopes_hoists_buffers		permanent	manual	MBB	WGP	FG_fight	SBB_periscope	BB_NL_4_periscopes_hoists_buffers		0	0	5		4	1.3								
15	NL_4_periscope_bearings_hull_glands		permanent		MBB	WGP	FG_fight	SBB_periscope	BB_NL_4_periscope_bearings_hull_glands		0	0	6.79	BB_VL_FL_FF_BR	4	1.2								
16	NL_4_wireless_mast_hoist		permanent		MBB	WGP	FG_fight	SBB_mast	BB_NL_4_wireless_mast_hoist		0	0	11.35	BB_VL_FL_FF_BR	4	1.5								
17	NL_4_wireless_RX_TX		permanent		MBB	WGP	FG_fight	SBB_mast	BB_NL_4_wireless_RX_TX		0	0	2.97	BB_VL_FH_RM_CO	4	2.3								
18	NL_4_radar_mast_hoist		permanent		MBB	WGP	FG_fight	SBB_mast	BB_NL_4_radar_mast_hoist		0	0	8.95	BB_VL_FL_FF_BR	4	2.5								
19	NL_4_radar_set		permanent		MBB	WGP	FG_fight	SBB_mast	BB_NL_4_radar_set		0	0	3.08	BB_VL_FH_RM_CO	4	0.4								
20	NL_4_EW_mast_hoist		permanent		MBB	WGP	FG_fight	SBB_mast	BB_NL_4_EW_mast_hoist		0	0	8.2	BB_VL_FL_FF_BR	4	2.7								
21	NL_4_EW_equip		permanent		MBB	WGP	FG_fight	SBB_mast	BB_NL_4_EW_equip		0	0	2.5	BB_VL_FH_RM_CO	4	1.4								
22	NL_1_torpedo_loading_hatch		permanent		MBB	WGP	FG_fight	SBB_torpedo	BB_NL_1_torpedo_loading_hatch		0	0	4.15	BB_VL_FH_RM_WS	1	0.7								
23	NL_7_torpedo_tubes		permanent		MBB	WGP	FG_fight	SBB_torpedo	BB_NL_7_torpedo_tubes		0	0	2.55	BB_VL_FH_RM_WS	7	36								
24	NL_7_torpedo_CP		permanent		MBB	WGP	FG_fight	SBB_torpedo	BB_NL_7_torpedo_CP		0	0	1.37	BB_VL_FH_RM_WS	7	1								
25	NL_7_torpedo_stowage_handling_gr		permanent		MBB	WGP	FG_fight	SBB_torpedo	BB_NL_7_torpedo_stowage_handling_gr		0	0	2.88	BB_VL_FH_RM_WS	7	22.3								

Fig.14: Layout of the WGP showing major inputs required in defining items of weight data on the vessel



		troubleshooting						FUNCTIONAL FOR SPATIAL PHYSICAL ARCHITECTURE			MUST ACC WEIGHT CONDITION/OTHER D			LOGICAL ARCHITECTU	
	(INFO ONLY)							1	2	3	BB Level	Attributes			connection point	
Call	Name	Shape	L/extent (	B/D (m)	H (m)		ation	MBB	FG	SBB	(info only)	Volume (m^3)	/BS Classifi	Weight (te)	input	output
1	DB_FO_VV_TK_a	sphere	0.3	0.3	0.3		Z	DB_DSSS	DB_FO	DB_FO_VV_TK_a	3.0	0.0	9.0	0.0	top	top
2	DB_FO_VV_TK_m	sphere	0.3	0.3	0.3		Z	DB_DSSS	DB_FO	DB_FO_VV_TK_m	3.0	0.0	9.0	0.0	top	top
3	DB_FO_VV_TK_f	sphere	0.3	0.3	0.3		Z	DB_DSSS	DB_FO	DB_FO_VV_TK_f	3.0	0.0	4.0	0.0	top	top
4	DB_DT_CO_AC_a	sphere	0.3	0.3	0.3		Z	DB_DSSS	DB_DT	DB_DT_CO_AC_a	3.0	0.0	4.0	2.6	top	top
5	DB_DT_CO_AC_f	sphere	0.3	0.3	0.3		Z	DB_DSSS	DB_DT	DB_DT_CO_AC_f	3.0	0.0	4.0	2.6	top	top
6	DB_DT_PU_AC	sphere	0.3	0.3	0.3		Z	DB_DSSS	DB_DT	DB_DT_PU_AC	3.0	0.0	4.0	3.0	top	top
7	DB_DT_SA_DC	cylinder	10.0	0.6	0.6		Z	DB_DSSS	DB_DT	DB_DT_SA_DC	3.0	3.6	2.0	2.2	bottom	bottom
8	DB_DT_AK_DC	cylinder	10.0	0.6	0.6		Z	DB_DSSS	DB_DT	DB_DT_AK_DC	3.0	3.6	3.0	1.9	bottom	bottom
9	DB_DT_CN_DC	cylinder	10.0	0.6	0.6		Z	DB_DSSS	DB_DT	DB_DT_CN_DC	3.0	3.6	3.0	1.5	bottom	bottom
10	DB_DT_EW_DC	cylinder	10.0	0.6	0.6		Z	DB_DSSS	DB_DT	DB_DT_EW_DC	3.0	3.6	3.0	2.7	bottom	bottom
11	DB_DT_RA_DC	cylinder	10.0	0.6	0.6		Z	DB_DSSS	DB_DT	DB_DT_RA_DC	3.0	3.6	3.0	2.5	bottom	bottom
12	DB_DT_SO_DC	cylinder	1.9	3.0	3.0		Z	DB_DSSS	DB_DT	DB_DT_SO_DC	3.0	16.7	3.0	4.0	aft	aft
13	DB_DT_SC_DC	sphere	0.8	0.1	0.2		Z	DB_DSSS	DB_DT	DB_DT_SC_DC	3.0	0.0	3.0	0.8	top	top
14	DB_DT_MC_DC	sphere	0.3	0.3	0.3		Z	DB_DSSS	DB_DT	DB_DT_MC_DC	3.0	0.0	3.0	3.3	top	top
15	DB_DT_DD_LC_a	box	2.0	1.0	1.0		Z	DB_DSSS	DB_DT	DB_DT_DD_LC_a	3.0	2.0	3.0	0.2	top	bottom
16	DB_DT_DD_LC_m	box	2.0	1.0	1.0		Z	DB_DSSS	DB_DT	DB_DT_DD_LC_m	3.0	2.0	3.0	0.2	top	bottom
17	DB_DT_DD_LC_f	box	2.0	1.0	1.0		Z	DB_DSSS	DB_DT	DB_DT_DD_LC_f	3.0	2.0	3.0	0.2	top	bottom
18	DB_DT_DD_AN_p	box	2.0	1.0	1.0		Z	DB_DSSS	DB_DT	DB_DT_DD_AN_p	3.0	2.0	3.0	0.2	fwd	stbd
19	DB_DT_DD_AN_s	box	2.0	1.0	1.0		Z	DB_DSSS	DB_DT	DB_DT_DD_AN_s	3.0	2.0	3.0	0.2	fwd	port
20	DB_DT_DD_MN_p	box	2.0	1.0	1.0		Z	DB_DSSS	DB_DT	DB_DT_DD_MN_p	3.0	2.0	3.0	0.2	top	stbd

Fig.15: Layout of the EDP showing major inputs required in defining spaces on the vessel

4.5 Equipment Database Program

The Equipment Database Program (EDP) defines the input necessary to create a physical model of a DS3 component. The input consists of name, shape, dimensions, orientation, BB hierarchy, WG classifications, weight, connection points, Fig.15. For the submarine case study, there were 365 equipment objects, which means 5100 input data, assuming each component requires 14 inputs.

4.6 Component Granularity Program

The Component Granularity Program (CGP) provides input for integrating DS3 components into the whole submarine design. As shown in Fig.16, the inputs for the DS3 components: the type of components, which could be equipment (DB) or numerical (NL), *Mukti et al. (2022)*; equipment data defined in EDP; BB hierarchy (up to level 4), relative position in X-, Y-, Z- axes relative to the space block, space block defined in VGP. Unlike database (DB) components, numerical (NL) components could be used to handle DS3 components that lacked sufficient detail in ESSD.

			troubleshooting	CTIONAL FOR SPATIAL PHYSICAL ARCHITECTURE (BB management based strt from stbd(-) to port(+)									
	(INFO ONLY)			1	2	3	4	BB Level (i	Initial Location			axis ro	initial
Call N	Name	Object Type (nu	Equipment from Database	MBB	FG	SBB	BB1		X%	Y%	Z%	X/Y/Z	compartment
1	BB_DB_FO_VV_TK_a	equipment	DB_FO_VV_TK_a	MBB	CGP	FO	BB_DB_FO_VV_TK_a	4	0.0	0.0	0.0		BB_VL_IA_DV_OA
2	BB_DB_FO_VV_TK_m	equipment	DB_FO_VV_TK_m	MBB	CGP	FO	BB_DB_FO_VV_TK_m	4	0.0	0.0	0.0		BB_VL_IA_DV_OM
3	BB_DB_FO_VV_TK_f	equipment	DB_FO_VV_TK_f	MBB	CGP	FO	BB_DB_FO_VV_TK_f	4	0.0	0.0	0.0		BB_VL_IA_DV_OF
4	BB_DB_DT_CO_AC_a	equipment	DB_DT_CO_AC_a	MBB	CGP	DT	BB_DB_DT_CO_AC_a	4	0.8	0.0	0.0		BB_VL_FH_RM_CO
5	BB_DB_DT_CO_AC_f	equipment	DB_DT_CO_AC_f	MBB	CGP	DT	BB_DB_DT_CO_AC_f	4	-0.4	0.0	0.3		BB_VL_FH_RM_WS
6	BB_DB_DT_PU_AC	equipment	DB_DT_PU_AC	MBB	CGP	DT	BB_DB_DT_PU_AC	4	0.5	0.0	0.0		BB_VL_FH_RM_CO
7	BB_DB_DT_SA_DC	equipment	DB_DT_SA_DC	MBB	CGP	DT	BB_DB_DT_SA_DC	4	0.1	0.0	0.3		BB_VL_FL_FF_BR
8	BB_DB_DT_AK_DC	equipment	DB_DT_AK_DC	MBB	CGP	DT	BB_DB_DT_AK_DC	4	-0.5	0.0	0.3		BB_VL_FL_FF_BR
9	BB_DB_DT_CN_DC	equipment	DB_DT_CN_DC	MBB	CGP	DT	BB_DB_DT_CN_DC	4	-0.3	0.0	0.3		BB_VL_FL_FF_BR
10	BB_DB_DT_EW_DC	equipment	DB_DT_EW_DC	MBB	CGP	DT	BB_DB_DT_EW_DC	4	-0.2	0.0	0.3		BB_VL_FL_FF_BR
11	BB_DB_DT_RA_DC	equipment	DB_DT_RA_DC	MBB	CGP	DT	BB_DB_DT_RA_DC	4	-0.1	0.0	0.3		BB_VL_FL_FF_BR
12	BB_DB_DT_SO_DC	equipment	DB_DT_SO_DC	MBB	CGP	DT	BB_DB_DT_SO_DC	4	-0.2	0.0	0.0		BB_VL_FL_FF_EF
13	BB_DB_DT_SC_DC	equipment	DB_DT_SC_DC	MBB	CGP	DT	BB_DB_DT_SC_DC	4	0.2	0.0	0.0		BB_VL_FH_RM_CO
14	BB_DB_DT_MC_DC	equipment	DB_DT_MC_DC	MBB	CGP	DT	BB_DB_DT_MC_DC	4	0.7	0.0	0.0		BB_VL_MV_RM_MR
15	BB_DB_DT_DD_LC_a	equipment	DB_DT_DD_LC_a	MBB	CGP	DT	BB_DB_DT_DD_LC_a	4	0.1	0.0	0.5		BB_VL_MV_RM_MR
16	BB_DB_DT_DD_LC_m	equipment	DB_DT_DD_LC_m	MBB	CGP	DT	BB_DB_DT_DD_LC_m	4	-0.8	0.0	-0.3		BB_VL_FH_RM_CO
17	BB_DB_DT_DD_LC_f	equipment	DB_DT_DD_LC_f	MBB	CGP	DT	BB_DB_DT_DD_LC_f	4	-0.1	0.0	0.5		BB_VL_FH_RM_WS
18	BB_DB_DT_DD_AN_p	equipment	DB_DT_DD_AN_p	MBB	CGP	DT	BB_DB_DT_DD_AN_p	4	-0.6	0.4	-0.6		BB_VL_MV_RM_MR
19	BB_DB_DT_DD_AN_s	equipment	DB_DT_DD_AN_s	MBB	CGP	DT	BB_DB_DT_DD_AN_s	4	-0.6	-0.4	-0.6		BB_VL_MV_RM_MR
20	BB_DB_DT_DD_MN_p	equipment	DB_DT_DD_MN_p	MBB	CGP	DT	BB_DB_DT_DD_MN_p	4	-0.5	0.2	-0.4		BB_VL_FH_RM_CO

Fig.16: Layout of the CGP showing major inputs required in defining DS3 components on the vessel

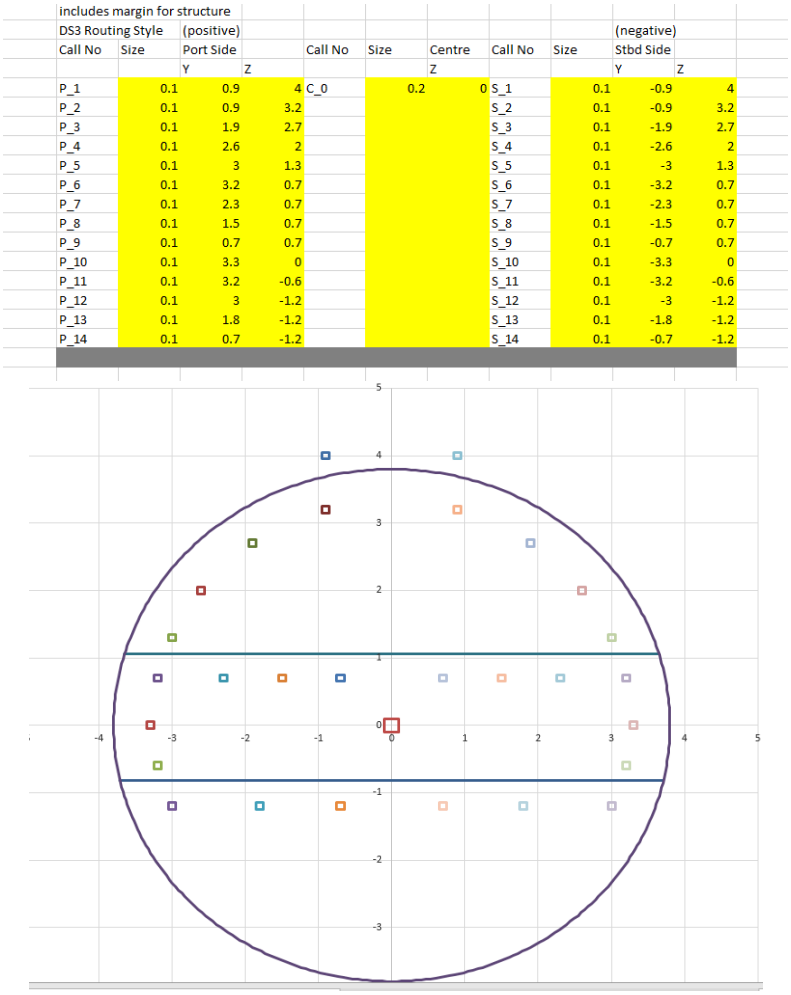
4.7 System Preamble Program

The System Preamble Program (SPP) provides an input menu to physically define DS3 connections. As shown in Fig.17, it consists of the name of the connection, DS3 technology (e.g., cabling, piping, trunking), mitred bend assumption, the shape of the connection (circle or rectangle), cross-sectional dimensions, and UCL submarine weight classification. For the submarine case study, there were more than 400 connections and thus 3600 inputs if each connection requires 9 inputs.

ble size from industry that can be used for multiple services)						must larger than Width			(select from column B	
Call N	Name	DS3 Technology	Mitred Bends (yes/no)	Section Shape (Circle/Rectangle)	Width (mm)	Height (mm) (only for rectangle)	nd Radius (mi	Weight Classification (1 to 9 UCL)	Valid Service 1	Va
1	DT_1	cabling	yes	circle	89	89	98	3	DT	
2	DT_2	cabling	yes	circle	89	89	98	3	DT	
3	DT_3	cabling	yes	circle	89	89	98	3	DT	
4	DT_4	cabling	yes	circle	89	89	98	3	DT	
5	DT_5	cabling	yes	circle	89	89	98	3	DT	
6	DT_6	cabling	yes	circle	89	89	98	3	DT	
7	DT_7	cabling	yes	circle	89	89	98	3	DT	
8	DT_8	cabling	yes	circle	89	89	98	3	DT	
9	DT_9	cabling	yes	circle	89	89	98	3	DT	
10	DT_10	cabling	yes	circle	89	89	98	3	DT	
11	DT_11	cabling	yes	circle	89	89	98	3	DT	
12	DT_12	cabling	yes	circle	89	89	98	3	DT	
13	DT_13	cabling	yes	circle	89	89	98	3	DT	
14	DT_14	cabling	yes	circle	89	89	98	3	DT	
15	DT_15	cabling	yes	circle	89	89	98	3	DT	
16	DT_16	cabling	yes	circle	89	89	98	3	DT	
17	DT_17	cabling	yes	circle	89	89	98	3	DT	
18	DT_18	cabling	yes	circle	89	89	98	3	DT	
19	DT_19	cabling	yes	circle	89	89	98	3	DT	
20	DT_20	cabling	yes	circle	89	89	98	3	DT	

Fig.17: Layout of the SPP showing major inputs required in defining physical DS3 connections

In the SPP, the location of system highways can also be adjusted as is shown in Fig.18. This provides identifications to be used in the System Connection Program (SCP). System highways consisted of some pre-defined longitudinal lines from forward to aft of the vessel and could be modelled as a highway object in Paramarine.



includes margin for structure											
DS3 Routing Style		(positive)								(negative)	
Call No	Size	Port Side		Call No	Size	Centre	Call No	Size		Stbd Side	
		Y	Z			Z				Y	Z
P_1	0.1	0.9	4	C_0	0.2	0	S_1	0.1		-0.9	4
P_2	0.1	0.9	3.2				S_2	0.1		-0.9	3.2
P_3	0.1	1.9	2.7				S_3	0.1		-1.9	2.7
P_4	0.1	2.6	2				S_4	0.1		-2.6	2
P_5	0.1	3	1.3				S_5	0.1		-3	1.3
P_6	0.1	3.2	0.7				S_6	0.1		-3.2	0.7
P_7	0.1	2.3	0.7				S_7	0.1		-2.3	0.7
P_8	0.1	1.5	0.7				S_8	0.1		-1.5	0.7
P_9	0.1	0.7	0.7				S_9	0.1		-0.7	0.7
P_10	0.1	3.3	0				S_10	0.1		-3.3	0
P_11	0.1	3.2	-0.6				S_11	0.1		-3.2	-0.6
P_12	0.1	3	-1.2				S_12	0.1		-3	-1.2
P_13	0.1	1.8	-1.2				S_13	0.1		-1.8	-1.2
P_14	0.1	0.7	-1.2				S_14	0.1		-0.7	-1.2

Fig.18: System highways setup in the SPP showing an initial highways visualisation (bottom) and major inputs required in defining system highways on the vessel (top)

4.8 System Connection Program

Like CGP, the SCP also provides necessary inputs for integrating DS3 connections into the whole submarine design. The inputs consist of connection name, physical connection, type of connections, highway defined in SPP, BB hierarchy (up to level 4), the connected DS3 components (source and sink), Fig.19.

						spur must us	FUNCTIONAL FOR SPATIAL PHYSICAL ARCHITECTURE					LOGICAL ARCHITECTURE (li
				Object T	Type of Connections	Use Se Highw	1	2	3	4	SELLECT FROM ARRANGEMENT	TO
ID	Medium (m	Notes	PL ID	Physicall	(system_connection/dis	(yes/no)	MBB	FG	SBB	BB1	Object Source (from)	Dependency 1
1	DT_1		PL0	yes	system_connection	yes P_9	MBB	SCP	DT	BB_DT_1	BB_DB_DT_CO_AC_a	BB_DB_DT_DD_LC_m
2	DT_2		PL0	yes	system_connection	yes P_9	MBB	SCP	DT	BB_DT_2	BB_DB_DT_CO_AC_f	BB_DB_DT_DD_LC_f
3	DT_3		PL0	yes	system_connection	yes P_9	MBB	SCP	DT	BB_DT_3	BB_DB_DT_PU_AC	BB_DB_DT_DD_LC_m
4	DT_4		PL0	yes	system_connection	yes P_9	MBB	SCP	DT	BB_DT_4	BB_DB_DT_SC_DC	BB_DB_DT_DD_LC_m
5	DT_5		PL0	yes	system_connection	yes P_9	MBB	SCP	DT	BB_DT_5	BB_DB_DT_MC_DC	BB_DB_DT_DD_LC_a
6	DT_6		PL0	yes	system_connection	yes P_9	MBB	SCP	DT	BB_DT_6	BB_DB_DT_DD_LC_a	BB_DB_DT_DD_AN_p
7	DT_7		PL0	yes	system_connection	yes S_9	MBB	SCP	DT	BB_DT_7	BB_DB_DT_DD_LC_a	BB_DB_DT_DD_AN_s
8	DT_8		PL0	yes	system_connection	yes P_9	MBB	SCP	DT	BB_DT_8	BB_DB_DT_DD_LC_m	BB_DB_DT_DD_MN_p
9	DT_9		PL0	yes	system_connection	yes S_9	MBB	SCP	DT	BB_DT_9	BB_DB_DT_DD_LC_m	BB_DB_DT_DD_MN_s
10	DT_10		PL0	yes	system_connection	yes P_9	MBB	SCP	DT	BB_DT_10	BB_DB_DT_DD_LC_f	BB_DB_DT_DD_FN_p
11	DT_11		PL0	yes	system_connection	yes S_9	MBB	SCP	DT	BB_DT_11	BB_DB_DT_DD_LC_f	BB_DB_DT_DD_FN_s
12	DT_12		PL0	yes	system_connection	yes P_9	MBB	SCP	DT	BB_DT_12	BB_DB_DT_DD_MN_p	BB_DB_DT_DD_AN_p

Fig.19: Layout of SCP showing major inputs required in defining DS3 connections on the vessel

The number of inputs of the SCP for the submarine case study was 4700 as each connection required 10 inputs and there were 470 connections.

4.9. Summary of the Programs

Although the lines of codes are not necessarily a metric of goodness, the summary of codes of each program is shown in Table II and the output summary in Paramarine is shown in Fig.20. Improvements were made to the proposed programs for performing the modelling task in Paramarine. The proposed programs could convert within a minute on a standard PC machine the input data provided in the sub-marine case study, which consisted of some volume objects, more than 150 numerical weight objects, 200 component objects, and 400 connection objects, to 20,000 lines of KCLs. Therefore, the execution time of the programs, for sending macros to Paramarine, was driven by the quality of the code and there remains scope for this to be further improved. The actual code is over 8000 lines long.

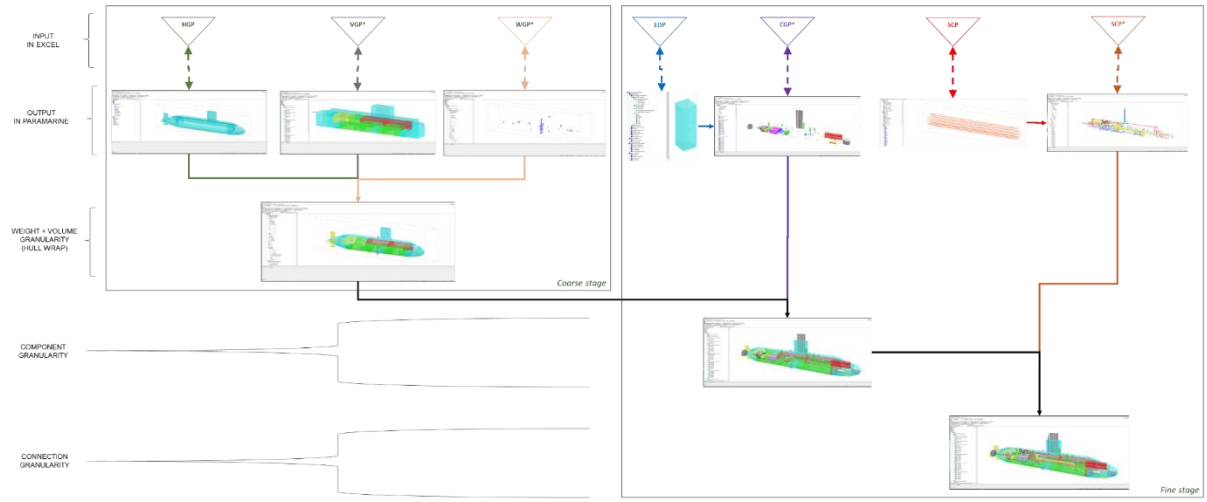


Fig.20: Output summary of NBA programs in SURFCON Paramarine (see Fig.10, *Mukti et al. (2022)*)

Table II: Summary of codes in the Input Data Centre

Program	**Description**	**Script Identifier**	**Size (Lines)**
MMP	Main Menu Program	A_A_MMP	42
DPP	Design Preamble Program	A_B_DPP	238
DAP	Design Analysis Program	A_C_DAP	537
HGP	Hull Granularity Program	C_A_HGP	1460
VGP	Volume Granularity Program	C_B_VGP	910
		C_C_VGP	254
		C_D_VGP	148
WGP	Weight Granularity Program	B_A_WGP	710
		B_B_WGP	191
EDP	Equipment Database Program	D_A_EDP	556
		D_B_EDP	1200
CGP	Component Granularity Program	D_C_CGP	555
		D_D_CGP	684
		D_E_CGP	81
SPP	System Preamble Program	E_A_SPP	369
SCP	System Connection Program	E_B_SCP	300
		E_C_SCP	756
KCL Output			>25000

6. Critique of the New Design Tool Applied to a Submarine Study

As discussed in Section 2, most automated approaches hardcode design steps, many design algorithms and their assumptions for sizing often implying, but are not limited to, how the spaces are arranged within the vessel. This, in turn, makes the software program follow several design decisions automatically every time an unbalanced condition occurs in the design. This can then allow hundreds of concept designs to be generated quickly by the computer(s) but all based on 'hidden' configurational assumptions. Such an automated approach is consequently difficult to be assessed, i.e., is not revealed easily (if at all) to the designer and thus is a 'black-box' synthesis. The implementation of the UCL DBB approach in Paramarine for DS3 was intended to commence a new ship design from a blank sheet. It must be emphasised that although the Paramarine has some hardcoded sizing algorithms as objects (e.g., "generator_sizing" object), the designer still can choose whether to use such objects without the need to modify the main codes of the software, which is the opposite of the black-box system. What makes modern automation have black-box characteristics is not just their inaccessible algorithms or data but also the difficulty in determining the causal link between input databases or design rules and the resulting options generated.

Assuming the development of the tool is before commencing a given design study there would seem to be a trade-off between the level of design automation and the transparency of the tool. Fig.21 shows the more choices, decisions, or design algorithms hardcoded into the tool means the less design effort to generate more design concepts. However, this then reduces the flexibility of the design tool and makes the tool highly opaque as those hardcoded inputs are not revealed easily to the designer using the system, i.e., a black-box tool. Conversely, the glass-box, SURFCON Paramarine design tool with the intent to be able to explore radical solutions, starts the design ab initio, to be highly flexible, without any step-by-step menu (or any dialogue box) for commencing a new submarine design study, which means require more designer inputs, i.e., more design effort than the black-box tool. Therefore, the solution space produced by a glass-box approach will be less populated than the myriad design solutions produced by a black-box approach, however as *Purton (2016)* showed each solution may not be practical and the solution space is likely to be much more restricted, *Andrews (2018)*.

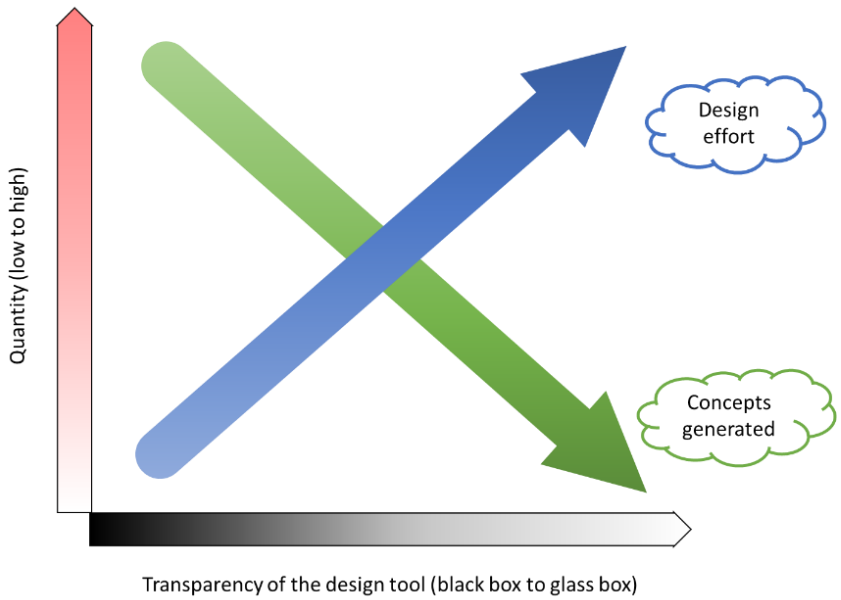


Fig.21: A simplified nature of the Computer-Aided Ship Design tool, with the X-axis as the indicator of design transparency, i.e., it is getting darker on the left-hand side (for black-box approach) and getting brighter on the right-hand side (for glass-box approach) while the Y-axis indicates the level of quantities from low (white) to high (red), which corresponds to the design effort in blue and the number of design solution(s) produced by the tool in green

This then raises questions as to what should be automated in the design tool and what should not. Fig.22 summarises important decision making, such as choice of design algorithms, which should be kept as the input and not hardcoded into design tools. This will not constrain the overall ship design solution space size early in the design process and so retain design flexibility. This is because in ESSD any design study should rapidly evolve as part of the Requirement Elucidation dialogue, *Andrews (2018)*. Thus, in the initial case study (see Section 2.4), the engine room was quickly resized due to the need to fit additional diesel generators for necessary redundancy. Therefore, the proposed Network Block approach allowed design flexibility and only automated routine tasks - Gulf of Execution. Thus, only the routine could be simplified while acting within the design tool should not resort to hardcoding design steps or choices and for the selection of such design algorithms the choice must be with the designer.

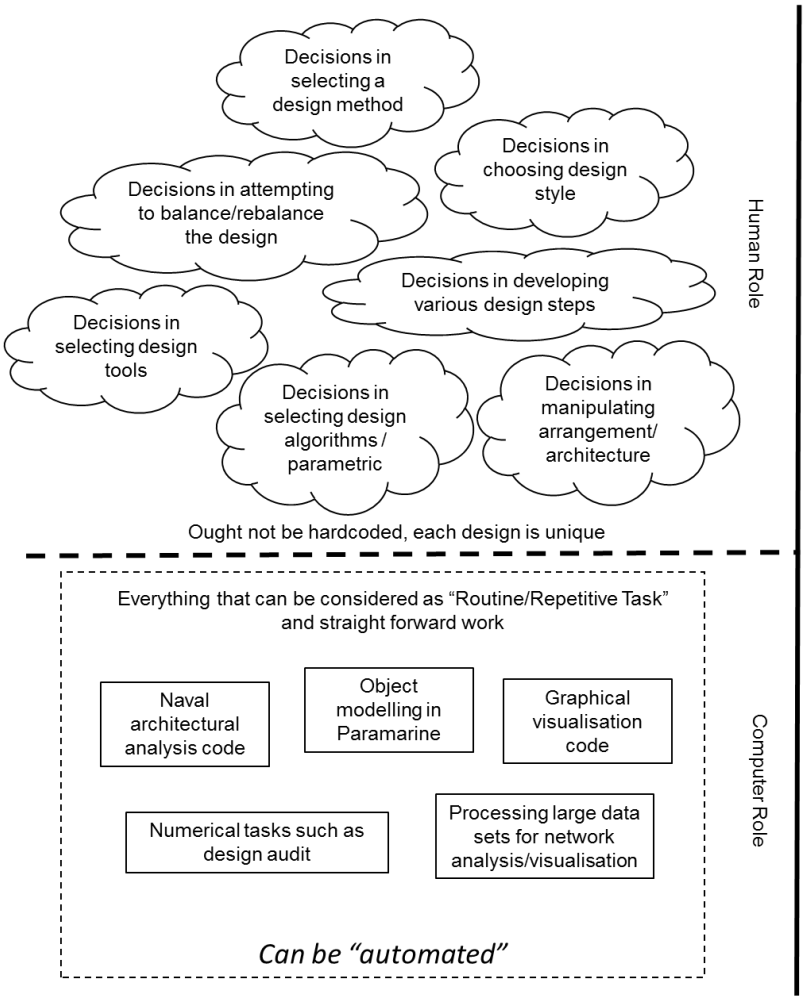


Fig.22: Decision making CAD processes vs human designer showing what ought to be automated and what ought not

7. Conclusions on the New Tool

This paper has outlined a new tool to mitigate modelling issues for DS3 in ESSD using a sophisticated 3D CASD system. A more plausible submarine design than 2.5D definition could now be produced more quickly, enabling a 3D informed dialogue and more realistic space reservation for DS3 routing. Highly automated 3D modelling of DS3, the transparent approach is now possible, which mitigates the demanding modelling task in implementing the UCL DBB approach in SURFCON Paramarine. However, as part of the justification of the ability of the Network Block Approach (NBA) to assist in the DS3 synthesis of submarines, it was necessary to test its sensitivity to different design decisions. This next step is to be addressed in future papers.

Acknowledgements

This research was supported by the Indonesia Presidential Scholarship and the UCL Joint Scholarship awarded to the first author.

References

ANDREWS, D.J. (1994), *Preliminary Warship Design*, Trans. RINA 136

ANDREWS, D.J. (2011), *Marine Requirements Elucidation and the Nature of Preliminary Ship Design*, IJME 153

ANDREWS, D.J. (2012), *Art and science in the design of physically large and complex systems*, Proc. Royal Society A: Mathematical, Physical and Eng. Sciences 468 (2139), pp.891-912

ANDREWS, D.J. (2018), *The Sophistication of Early Stage Design for Complex Vessels*, IJME Special Edition, 472 Part A

ANDREWS, D.J. (2021), *Who Says There Are No Real Choices in Submarine Design?* WARSHIP 2021, RINA

ANDREWS, D.J.; CASAROSA, L.; PAWLING, R. (2009), *Integrating Simulation and Computer Aided Ship Design Software and Processes*, RINA Int. Conf. Computer Applications in Shipbuilding (ICCAS)

ANDREWS, D.J.; CUDMORE, A.C.; HUMBLE, P. ; WILSON, D. (1996), *SUBCON - A New Approach to Submarine Concept Design*, RINA Warships Proc. Symp. Naval Submarines 5: The Total Weapons System, London

ANDREWS, D.J.; DICKS, C.A. (1997), *The Building Block Design Methodology Applied to Advanced Naval Ship Design*, Proc. Int. Marine Design Conf. (IMDC)

ANDREWS, D.J.; PAWLING, R.J. (2003), *SURFCON A 21st Century Ship Design Tool*, 8th Int. Marine Design Conf. (IMDC), Athens

ANDREWS, D.J.; PAWLING, R.J. (2008), *A Case Study in Preliminary Ship Design*, RINA IJME 139

BREFORT, D.; SHIELDS, C.; HABBEN JANSEN, A., DUCHATEAU, E.; PAWLING, R.; DROSTE, K.; JASPER, T.; SYPNIEWSKI, M.; GOODRUM, C.; PARSONS, M.A.; KARA, M.Y.; ROTH, M.; SINGER, D.J.; ANDREWS, D.; HOPMAN, H.; BROWN, A.; KANA, A.A. (2018), *An architectural framework for distributed naval ship systems*, Ocean Eng. 147, pp.375-385

CIERAAD, S.; DUCHATEAU, E.; ZANDSTRA, R.; DE BROEK-DE BRUIN, W.H. van (2017), *A Packing Approach Model in Support of the Conceptual Design of Naval Submarines*, 16th COMPIT Conf., Cardiff

DUCHATEAU, E.A.E. (2016), *Interactive evolutionary concept exploration in preliminary ship design*, Ph.D. thesis, TU Delft, https://doi.org/10.4233/uuid:27ff1635-2626-4958-bcdb-8aee282865c8

FIEDEL, E.R. (2011), *Cooling System Early-Stage Design Tool for Naval Applications*, Master's thesis, MIT

JURKIEWICZ, D.J.; CHALFANT, J.; CHRYSSOSTOMIDIS, C. (2013), *Modular IPS machinery arrangement in early-stage naval ship design*, 2013 IEEE Electric Ship Technologies Symp., Arlington

KOURIAMPALIS, N., PAWLING, R.J.; ANDREWS, D.J. (2021), *Modelling the operational effects of deploying and retrieving a fleet of uninhabited vehicles on the design of dedicated naval surface ships*, Ocean Eng. 219: 108274

MAHONEN, C.; SPRADLEY, W.; GERDON, M. (2007), *Automating Early Stage Submarine Design: Development of the Submarine Concept Design (SUBCODE) Program*, ASNE Day, Arlington

MUKTI, M.H. (2022), *A Network-Based Design Synthesis of Distributed Ship Services Systems for a Non Nuclear Powered Submarine in Early Stage Design*, Ph.D. thesis, University College London

MUKTI, M.H.; PAWLING, R.J.; ANDREWS, D.J. (2021), *Distributed Ship Service Systems Architecture in The Early Stages of Designing Physically Large and Complex Vessels: The Submarine Case*, IJME 163

MUKTI, M.H.; PAWLING, R.J.; ANDREWS, D.J. (2022), *The Network Block Approach Applied to the Initial Design of Submarine Distributed Ship Service Systems*, 14th Annual Int. Marine Design Conf., Vancouver

MUKTI, M.H.; PAWLING, R.J.; SAVAGE, C.; ANDREWS, D.J. (2019), *Distributed Ship Service Systems Architecture in the Early Stages of Physically Large and Complex Products*, ICCAS Int. Conf. Computer Applications in Shipbuilding, Rotterdam

MUKTI, M.H.; RANDALL, R.E. (2017), *Graphic Method for Improved Indonesian Navy Submarine Design Acquisition; the Investigation of Small (Midget) Naval Submarine Development*, RINA Warship: Naval Submarines & UUVs, Bath

NORMAN, D. (2013), *The design of everyday things*, MIT Press

OERS, B.J. van (2011), *A packing approach for the early stage design of service vessels*, Ph.D. thesis, TU Delft

PAWLING, R.J. (2007), *The Application of the Design Building Block Approach to Innovative Ship Design*, Ph.D. thesis, University College London

PAWLING, R.J.; ANDREWS, D.J. (2011), *A Submarine Concept Design - The Submarine as an UXV Motherhip*, Warship 2011: Naval Submarines and UUVs, Bath

PAWLING, R.J.; PIPERAKIS, A.; ANDREWS, D. (2015), *Developing Architecturally Oriented Concept Ship Design Tools for Research and Education*, 12th Int. Marine Design Conf. (IMDC), Tokyo

PURTON, I.M. (2016), *Concept Exploration for a Novel Submarine Concept Using Innovative Computer-Based Research Approaches and Tools*, Ph.D. thesis, University College London

PURTON, I.M.; PAWLING, R.J.; ANDREWS, D.J. (2015), *The Use of Computer Tools in Early Stage Design Concept Exploration to Explore a Novel Submarine Concept*, 12th Int. Marine Design Conf. (IMDC), Tokyo

THURKINS, E.J. (2012), *Development of an early stage ship design tool for rapid modeling in Paramarine*, Master's thesis, MIT, http://dspace.mit.edu/handle/1721.1/74992

UCL-NAME (2014), *Submarine Data Book*, University College London