

Article

# CAD Sensitization, an Easy Way to Integrate Artificial Intelligence in Shipbuilding

Arturo Benayas-Ayuso <sup>1,2,\*</sup> , Rodrigo Perez Fernandez <sup>1,3,\*</sup>  and Francisco Perez-Arribas <sup>1</sup>

<sup>1</sup> Department of Naval Architecture, Construction, Ocean and Naval Systems, Universidad Politécnica de Madrid, 28040 Madrid, Spain; francisco.perez.arribas@upm.es

<sup>2</sup> Accenture—Industry X—Industrial Division—Aerospace and Defense, 28046 Madrid, Spain

<sup>3</sup> Siemens Digital Industries Software, 28760 Madrid, Spain

\* Correspondence: abenayas@alumnos.upm.es (A.B.-A.); rodrigo.perez.fernandez@upm.es (R.P.F.); Tel.: +34-616-002-979 (A.B.-A.)

**Abstract:** There are two main areas in which the Internet of Ships (IoS) can help: firstly, the production stage, in all its phases, from material bids to manufacture, and secondly, the operation of the ship. Intelligent ship management requires a lot of information, as does the shipbuilding process. In these two phases of the ship's life cycle, IoS acts as a key to the keyhole. IoS tools include sensors, process information and real-time decision-making, fog computing, or delegated processes in the cloud. The key point to address this challenge is the design phase. Getting the design process right will help in both areas, reducing costs and making agile use of technology to achieve a highly efficient and optimal outcome. But this raises a lot of new questions that need to be addressed: At what stage should we start adding control sensors? Which sensors are best suited to our solution? Is there anything that offers more than simple identification? As we begin the process of answering all these questions, we realize that a Computer Aided Design (CAD) tool, as well as Artificial Intelligence (AI), mixed in a single tool, could significantly help in all these processes. AI combined with specialized CAD tools can enhance the sensitization phases in the shipbuilding process to improve results throughout the ship's life cycle. This is the base of the framework developed in this paper.

**Keywords:** shipbuilding; software; technology; ship design; computer aided design; internet of things; machine learning; artificial intelligence; deep learning; CAD



**Citation:** Benayas-Ayuso, A.; Perez Fernandez, R.; Perez-Arribas, F. CAD Sensitization, an Easy Way to Integrate Artificial Intelligence in Shipbuilding. *Computers* **2024**, *13*, 273. <https://doi.org/10.3390/computers13100273>

Academic Editors: Paolo Bellavista and Isidro Calvo

Received: 4 August 2024

Revised: 8 October 2024

Accepted: 18 October 2024

Published: 21 October 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In this paper, the concept Internet of Ships (IoS) tries to cover two applications of the Industrial Internet of Things (IIoT), focusing on the unique use of the sensors in all materials and product life cycle, reducing the ship sensitization process in time and cost [1].

We constructed our approach based on IoS. The first element to check is shipbuilding materials and components. When a material arrives to be stored in the shipyard, some identification needs to be assigned, and it needs to be marked and traced in the system. Depending on the type of element, this mark can be used in future applications on the ship. In the cases where it can be reused, the designer must select a sensor that can get an essential measure, like stress or temperature (other types can be suitable, too) [1].

The acquisition department gets the ship design material orders to place a bid option for their purchase. This action must also place a second working order in the manufacturing system for the material store people to attach a special Passive Radio Frequency Identification (RFID) item for identification or sensitization purposes [1].

We are going to focus on passive RFID tags because they are cheap and can be easily attached to the material, as well as read with a small sensor or even a smartphone (compared with semi-active or active ones with the same capabilities) [1].

The manufacturing process must keep the RFID item intact or with a low impact in the manipulation process, which assures, mainly for RFID sensors, that they are in a perfect state for working in a real ship life cycle [1].

All these steps set a complex path for the materials in the purchase/store process as well as new working methodologies for the ship designers. The software tools or systems used in the shipbuilding must cover the three well-defined and separated periods of the ship process: the design phase, shipbuilding, and the operation life, in which IoS sets an extra relation of the elements and processes included in each phase [1,2].

These elements can be sustained by working methodologies, such as Model-Based Systems Engineering (MBSE). This method is used to solve problems that need to be addressed in high-value-added sectors, such as defense products or aerospace engineering, which are also high-risk sectors that require robust design and construction processes from very early stages to reduce costs and ensure planning and scheduling within budget. Systems engineering requires the use of more specific technologies to be applied effectively. In systems engineering, there are countless applications that can be used to provide solutions at different stages of a ship's life cycle [2].

However, this is only part of the problem. There are major changes that affect us and require new technologies. Among them, the Internet of Things (IoT) stands out for the impact it has directly on the world we see and touch, as does the analysis of the data obtained with this source. At this point, as a collateral effect, Artificial Intelligence (AI) appears to make a heuristic study of the data. This type of study allows us to shed light on advantages, qualities, or problems that, in the linear analysis based on our own knowledge, do not usually come out but with a lot of experience, and not always. How this trend affects or may affect the world of shipbuilding and what the software industry can do to introduce IoT and AI in this sector is the main topic of this article [1].

Porter and Heppelmann [3] described the new technology stack: *“Smart, connected products require companies to build and support an entirely new technology infrastructure. This technology stack is made up of multiple layers, including new product hardware, embedded software, connectivity, a product cloud consisting of software running on remote servers, a suite of security tools, a gateway for external information sources, and integration with enterprise business systems”*. If a company wants to enter this business, it must have products that connect with the products of other companies. So, the necessity of having an open product is increasing. Also, Taylor [4] affirmed, *“The platform is the key to success. The things will get increasingly inexpensive, applications will multiply, and connectivity will cost pennies. The real value will be created in the horizontal platform that ties it all together, i.e., the new OS”*.

From the authors' point of view, the mentioned platform is not exactly an Operating System (OS) [5] but a set of interconnected applications that share the same data about the product. Data that have been appearing since the early design phases feed the single source of truth, not only with the design attributes but also with the desirable values for optimal performance and with the Objective Quality Evidence (OQE) generated during the ship construction and operation processes. Around these data in the single source of truth will appear the applications that control the lifecycle of the components, the sensors that collect the performance information, or the operational life of the same objects and devices [2].

### 1.1. CAD Tools and Simulation Tools

To set the grounds, in a wide manner, based on Aslam et al. definition of IoT and IoS: *“The IoT is composed of the following six basic building blocks (2C3SI): Communication, Computation, Semantics, Services, Sensing, and Identification, which deliver the functionality of the IoT technology. Likewise, based on the maritime application environment. . . . The IoS consists of four building blocks, i.e., sensing, communications, computation, and services”* [6].

Computer-Aided Design (CAD) systems are traditionally used for the design and construction phase processes, the manufacturing process has become the core of the current shipbuilding analysis, and Plant Simulation software has become the next level in production analysis, creating a specific CAD tool for the evaluation of the shipyard production

process based on a virtual mockup. Here is the first point where IoS set a new requirement in the production process analysis.

All these tools generate a lot of information, which needs to be managed, setting relations between the results and establishing evolution paths based on modification history. In this data management, the Product Lifecycle Management (PLM) solutions are helping to provide valuable product management tools in design and a verified link to Enterprise Resource Planning (ERP) for all phases of production, including monitoring and managing design non-conformities in the design or construction phases, or even during the ship's life cycle, in the maintenance phase. However, these tools require an enormous amount of information to be provided during the design and construction phases of the vessel. Therefore, it is necessary to combine CAD, PLM, and ERP so that all the value of the product is kept in the system and can be efficiently transferred to the digital mockup as the base of a real effective digital twin [2].

The single source of information can be a database, a PLM, an ERP, or other tools. But this tool must be established as the master and core of the entire product life cycle, and all other tools, as satellites of its information.

It is key that all devices, sensors, and components are not directly connected to the IoT layer of information but connected to a management core that controls the data to be exchanged to the IoT layer by those devices that are allowed to enter this network. In this way, it is possible to take advantage of the fact that all the existing components in the ship have their respective model created, incorporated, and defined from the CAD system. We could say that the CAD is the cradle of the ship's elements. Therefore, it is the CAD System that knows them and from where they can be assigned the mission and the level of participation in that IoT world [2].

Usually, CAD tools are known as a design screen where items are created, attributed, and placed (location parameters or attributes), but new tools have been developed and integrated with PLM, sharing information and simplifying the integration with simulation tools, which are become more interactive, including a 2D/3D environment (CAD like User Interface) helping in the learning process of their use.

Because information is stored in the same source of truth (PLM), the simulation impact is incorporated into the design seamlessly in any of the phases. CAD tools are at the beginning of the product lifecycle, but they also have a strong relationship with the production cycle (attributes), providing the necessary information for the construction in all aspects. Those CAD tools with a compact and homogeneous database can extend their contribution to both the life cycle management tools and the IoT connectivity management application of ship elements [2].

The basis of the digital twins is in the 1D/2D/2.5D/3D/4D model developed by the CAD, the data generated in the PLM through the study of its life process, and in the simulation tools that are incorporated into the design ecosystem as satellites based on the PLM or CAD data, adding more information to the model as the result of the simulation study; as well as all the data that can be incorporated from the steps taken in production (ERP) to the modeled elements that are added as the basis of this digital mockup. All these attributes, from production processes and simulation, complete the model information.

The 1D/2D/2.5D/3D/4D models, attributes, and relationships between the different elements of our design will allow the use of simulators in the digital twin phase to generate and manage it effectively not only as a well-illustrated model but as a valuable element with very relevant data that will allow us to simulate and anticipate complex situations that may occur in our ship [7].

Ship models have the added problem that it can cost a lot of time and money to transmit the data from the navigation position to the shore location of the twin, contrary to what happens in other sectors. For this reason, this type of digital twins must exist in both places, on board the ship and ashore, in the ship's control and maintenance team.

For all this to be possible, CAD applications must meet certain characteristics. A CAD system must evolve to become a global solution integrated and integrable with all the base elements of a future digital twin, also helping in the future design process [7].

With this in mind, we asked ourselves: What should be those features that make a CAD system an easily integrable design solution? What other features should CAD incorporate to help us in the design phase? We believe the answer is the integration of technologies such as AI in design processes. And since it is impossible to grow fast enough with organic growth, technologies that integrate other technologies are posed as the basis for this type of process. Companies must, therefore, look to the future of their products, thinking in terms of integration and integrability to facilitate the creation of value ecosystems in production.

A CAD-AI system needs to assist in the design phases, not just as a chatbot, which allows us to look for help on command in natural language, but with suggestions based on design lessons learned as well as production process non-conformities.

The question now is how to achieve this kind of integration. To answer this question, it is necessary to look to the future and identify the technological trends that will be developed in this area. We believe that connectivity, artificial intelligence, virtual reality, augmented reality, 3D printing, and Blockchain are emerging as the keys to designing the future in the coming years. We also think about what the application of these trends will look like in the maritime domain, which is undergoing a transformation to adapt to the new paradigm [2].

Any future technology for software solutions, the IoT world, Smart Ships, and any other product technology development will have to rely on one of these trends. They are not unrelated paths; they are converging trends and support each other [2].

### 1.2. Conceptual Framework

The conceptual framework on which the development is based identifies the experience in the field of work, in this case, the design of ships, as the key variable for achieving the desired result (see Figure 1).

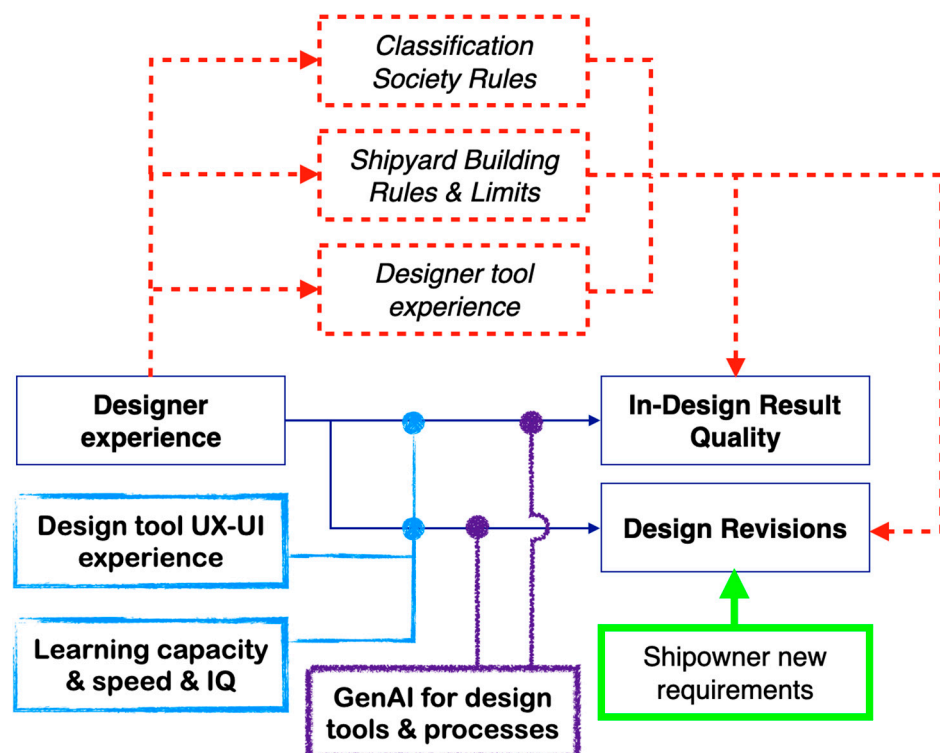


Figure 1. Conceptual framework.

The variables affected by the result are the quality of the designed product and the quantity or number of revisions that must be made to achieve an optimal result.

The fundamental causal relationship to be investigated is that the designer's experience enables the reduction of the number of revisions and the enhancement of the design quality, thereby ensuring a more suitable and effective solution in a shorter time frame. Furthermore, the potential for the same team to address more intricate designs is also a factor that must be considered. However, this invariably demands the involvement of highly experienced teams or teams with a high capacity for learning.

Additionally, the applicable rules of the classification society or the shipyard, as well as the construction limits existing in the facilities, also impact the design quality and the number of changes that must be applied to it.

A third factor is experience with the tools that will be used in the design process. Even if the designer lacks experience in the specific field, having a comprehensive understanding of the available tools can facilitate more efficient and focused design work. Conversely, a highly experienced designer may also experience a decline in performance when switching to a new tool. This can result in a loss of control over the expected outcome or an increase in the required time to achieve it.

Another factor that has an impact is the changes resulting from the owner's requirements; however, this will not be discussed in further detail here. These are typically established at the outset of the design process; however, alterations to the vessel's intended use or the regulatory requirements of the ports or states where it will berth may also necessitate adjustments to the design.

If we now focus on the variables that moderate the result, one factor is the user experience (UX) and user interface (UI) of the design tools. The ease and comfort of use, as well as the intuitiveness of the interface, allow users to become proficient at an earlier stage, thereby reducing the impact of the user experience on the quality of the design.

The subsequent variable of this category is the capacity and velocity of learning, in addition to the intelligence of the designers. These variables exert a considerable influence on the outcome of the designs, encompassing both their quality and the time required to complete them.

The objective at this juncture is to enhance the moderating variables—"Design tool UX/UI experience" and "Learning capacity & speed & IQ"—and to attenuate the effects of the mediating variables—"Classification Society Rules", "Shipyard Building Rules & Limits" and "Designer tool experience"—thereby reducing their impact on the results. To this end, a generative artificial intelligence system is being developed that enables the user to prioritize design tasks, freeing up cognitive resources that would otherwise be devoted to addressing the numerous constraints that impede the optimal solution.

This element, "GenAI for design tools & processes", becomes a new moderating variable that should change the game rules to improve design results.

## **2. Methodology Definition: Ship Design Tools Deep Study**

### *2.1. Which Are the Stages Where CAD Tools Can Help in IoS*

In the context of classical design, the project spiral approach is a common methodology. Nevertheless, the incorporation of CAD at the outset has been shown to diminish the overall duration of the design process. This is because CAD enables more comprehensive visual verification of designs, which is more closely aligned with the user's perspective. This is corroborated by the findings of the studies referenced in [8,9].

The first challenge of the Internet of Ships (IoS) is sensors, sensors that assist in the operation of the ship, sensors that require at least two wires for power and another two for reading the associated signal, sensors that have a mix of mechanical and electronic components, sensors placed in a particular location, regardless of whether they are easily accessible or not, sensors that, based on the IoS approach, must have a unique identifier within the ship as well as in their representation on the digital model, germ of the digital twin that will use such information [10].



The second IoS challenge pertains to the production phase, namely the univocal identification of parts. This may be achieved through the use of Barcode, Quick Response Code (QR Code), Radio Frequency Identification (RFID) tags [2,11], or a combination thereof. This type of identifier is utilized to facilitate the aggregation of parts for the Manufacturing Bill of Materials (MBOM), to ascertain the status of each MBOM item during the production process and may also include an indicator of preprocessing requirements or other steps in the production chain.

These tags are generated, used, and, in most cases, discarded or destroyed in the production process, which implies additional costs in the MBOM process are not directly linked with them. This extra cost is fully justified by the savings generated throughout the entire process, reducing the working hours for each MBOM preparation, control, transfer to the workshop, processing, and handover to the next step in the production line [2].

Based on this unique identification, part, if not all, of the process can be performed autonomously, i.e., the processing of the tube coils [12]. This allows certain jobs to be performed at any time during the working day, making it possible for the assembly pallets of each MBOM to be prepared outside working hours and ready for the shipbuilding workers to continue the next working shift.

These challenges have a common link: identification. Modern CAD has a unique way of identifying each element or occurrence in the data store—owned database, individual file-based database, or delegated in PLM—and the user's current working design, regardless of whether it is performed in 3D scene, 2D/1D design or a drawing, or each element in the simulation of a system. Not only the internal identification can be unique, but also a combination of other attributes that can be useful for the selected marking element sequence, making CAD one of the most important parts of the chain to generate this unique code, the seed for unique identification [10].

The generation of the unique identification code can be done on the fly in CAD tools, based on patterns, or, in case of item creation, saved as a special traceable attribute for each item or as a new related element with full life cycle capability and evolution (in PLM). At this point, when we generate a new pipe or divide it in working spools, CAD can populate an attribute with this code or create and link a new item that represents the tag linked with the piece of pipe [11].

This second approach is much better, as it helps us during the life cycle of this label, new entity, knowing when the item is generated is mature enough to be processed and, when it is "printed" and attached to the item, the referred element has a collectible catalog component assigned in the warehouse, or after workshop processing for elements like spools that have two production phases, creation of item and installation of item. After this process, the two elements, the item to be identified and the tag, became a new manufacturing item to be used in production. In addition, control manufacturing process dates can be obtained by synchronizing the warehouse control system based on an ERP [2] with the CAD system or, better, its life cycle control tool, the PLM.

An advantage of generating these tags in the CAD and storing them in the PLM is that relationships and non-repetition controls can be established, as well as viewing evolutions of the tags and their usage, allowing the shipyard to study the process from an item point of view. These new data set a new lake of information to be processed by the AI system connected.

In the manufacturing process, on the workshop or onboard the ship, when items become processed parts, such as pipe spools, the identification and sensor placement can collide or, rather, can be used as the same item, as an identification-sensor tag (RFID sensor tag) [1]. In this case, the cost of MBOM process identification is greatly reduced by using this identification sensor in the future ship operation phase, obtaining a measurement of the parameter by the sensor, and obtaining an identification that can be used in the onboard system to assist the ship's engineer's work [2].

A final idea points to RFID sensor tags, which are unique and can return a measurement of any parameter on the tagged item. The cost is reduced due to the absence of

wires—passive RFID or Battery-Assisted Passive RFID (BAP RFID) [2]—with a measurement always ready on the spot but only read when the process estimates it, being done as many times as necessary to ensure full operation. The cost of the sensor used is reduced, and its replacement is relatively easy. Furthermore, it can be placed in parts of the vessel that are not easily accessible with the ability to obtain the required value. This last idea only refers to passive RFID sensors that do not require maintenance for their operation [7].

An example of the latter idea can be RFID-type stress tags on the external or internal structure of the vessel. These can be applied to the main frame or other main elements subjected to high stresses during navigation. In this way, a study can be made, either in the construction phase of moving the blocks from the workshop to the slipway or study the structure's fatigue in the ship's navigation [2].

From these ideas, CAD software must assist in the entire design process with final solutions for the ship project and tools or aids in the production phase that are not destroyed during the production phase and can be useful afterward. The way to achieve this is the core of new developments based on new technologies that can be used today.

## 2.2. How CAD Can Help with Tag and Sensor Distribution

Based on the previous point, the labeling process helps in the production stages, preserving the traceability of the MBOM materials, as well as the final product obtained from this processed MBOM, which is a new material to be added to the next stage of production controlled in the ERP from the approved PLM design elements. This new product represents a new material to be included in another MBOM.

The ability to easily identify items allows users to check these tags against the shipyard's design system, returning information about the catalog item (component), relative position and final location on the ship, related items, etc., making it easy to integrate this information with a CAD-PLM system check tool [10,11].

In addition, if this information is loaded into the onboard software, it can be accessed during operation, allowing easy verification of the item's condition by comparing actual performance parameters with the baseline reference values in the digital twin loaded into the vessel's system [7].

The data from the digital twin compared to that obtained from an ID tag also allows for insight into recommended maintenance actions, prediction of possible unanticipated failures, and replacement work to be carried out prior to element breakage [7].

In the case of sensors, their distribution throughout the vessel is influenced by information from

1. Classification societies
2. The shipyard's construction experience
3. Original Equipment Manufacturer (OEM) requirements and in-equipment sensing capabilities
4. The shipowner's special reporting requirements

Based on these standards, the designer must locate the sensors in all the ship's systems, which can be wired or wireless, so that they are accessible with the cables or via a radio-frequency reader. In addition, part identification tags must be added to automate the production process, which can be a readable code accompanied by a barcode and/or QR, or even use a mixed tag that supports the visual code system (readable + barcode and/or QR) with an RFID identifier, easier to read when the tag keep in a hidden location in ship system.

These two ideas, labeling and a combination of label and sensor, can overwhelm users with information, making the design and design verification phases a nightmare. These are typical examples where CAD system tools can validate, improve and/or even correct some of the most common errors based on lessons learned from previous designs or, after applying classification society standards, correct the design to meet both.

The software developer must assist the designer with powerful CAD tools that can handle a set of standards (design rules) that apply to the model, allowing users to verify,

understand, and correct the design easily. According to our proposition of labeling, establishing identification tags and, where possible, replacing an identification tag with a combined RFID sensor will help reduce the cost in the overall life cycle [10].

According to our labeling proposal, establishing ID tags and, where possible, replacing an ID tag with a combined RFID sensor will help reduce the overall life cycle cost.

Based on the above requirements, one set of these standards is [2]

1. Tag visible from common aisles or corridors for barcode or QR code.
2. For RFID, BAP tags must be accessible for battery maintenance.
3. The reader antennas must be able to read all assigned elements, i.e., the distance must not exceed 30 m without obstacles and 15 m with obstacles. This establishes the control radius for the automatic readers [1].
4. Avoid shaded areas around RFID tags.

The location of tags and sensors is one of the problems that can be solved by design rules in the CAD system. But if the rulers, controls, and other aids are not well designed, they can overwhelm the designer in the process.

### 2.3. In Design Rules

Shipyards and engineering offices face a major problem in developing new designs, which is magnified by the loss of experience and the significant training requirements for less experienced engineers to produce high-quality designs. To solve these problems, companies have developed design rules that users must learn and apply to their current work. This idea is not new, as a set of design rules has always been included in the shipyard procedure book.

Problems arise when shipyard design authorities must check that the design conforms to this procedure book in an evolving environment, which overwhelms them and generates a lot of delay in the design process, a lot of revisions of the design, and sometimes human errors based on the non-understood process.

A natural solution easily arises: CAD should incorporate them, helping the user to follow and check them and allowing engineering managers (one of these shipyard design authorities) to verify and report non-conformities.

How we afford this change affects the outcome, even from a scalability point of view.

There are too many options to solve this problem, but from authors' experience and point of view, the three most common ways the CAD Developer can face this are Finite-State-Machine, CAD Configurable/Evolutive Software, and a Generative Artificial Intelligence Engine (GenAI) tool.

In the Finite-State-Machine approach, the software developer study the problem, offers a solution to the customer as a system improvement, and includes it in the next release. This was the old traditional approach. The main problems of this method are

1. Each rule programmed in the main core of the application must be predefined, functionally and technically, and included.
2. It offers a fast way to check but sets a rigid scenario for changing rules.
3. CAD developers should contact customers continuously (in an efficient and agile manner) or have a highly trained support team that can translate efficiently to developer changes required in rules.
4. New rules cannot be applied to the model till the next release or update of the software is launched, which sometimes also requires a rebuild of the database or CAD files used in the design.

If they want to improve the previous solution with an evolved approach, the CAD developer needs to generate a configuration capability in the solution, manual or based on a tool, which helps the customer to create its own rules. This is the current approach of most software developers, not only in CAD but also in other software solutions. But, usually, this approach ends up with the customer IT department being overwhelmed. Also,



this set of rules can be hard to apply to the final design because it requires an interpretative time, an application time, and a report time. The main problems of this method are:

1. Time consumption checking if this is done in the design phase: delay.
2. Programming of the rules. They can collide in design processes.
3. Report and automatic actions appliance. There is no knowledge about possible solutions.

The third way, which is based on the conceptual framework exposed, tries to minimize all effects of the elements and even reduce the main variable, the designer's experience in the working field, creating the effect of a one-minded team in the designers' group.

Like previously discussed approaches, some rules need to be defined, but the way we define them is in natural language, setting a change of state in the underlying engine for the rules. CAD should be integrated with a GenAI tool, allowing the shipyard to auto-configure a set of design rules by "reading" the normative assigned design rules book, learned as users work, etc.

Rules are created as a result of the requirements read from applicable normative, creating a set of features to be applied to the final design. The application of the rule can be easily interpreted by the final user, setting this feature as Boolean (match/not match), a list of states (allowing a Finite-State-Machine-like result), a number result of a simulation auto-applied, or even a descriptive report of the problems shown. The weight of each feature can be normalized to reduce the impact of each one in the check scenario. This can be done by adding a performance rules book for the report results.

All these sets of autoconfigured design rules are applied as training examples of the GenAI system results, checking them with finished previous projects and generating reports to be validated by the user (validating GenAI solution). Once GenAI CAD integration is trained and validated to perform tasks based on these rules, we can use it on a real design:

1. Learning about design phase evolution.
2. Learning about solutions applied to the issues discovered.
3. Applying some automatic solutions from lessons learned.
4. Creating its own design rules to improve the final design.

The solution's capacity for autonomous improvement allows for the retraining and reapplication of any alterations to the design rules, ensuring that all cases within the ship design are consistently updated. To reduce inefficiencies, this validation process can be completed overnight and reviewed in the morning by the corresponding design system validator for the affected zone. The process is as straightforward as uploading an updated version of the rules to the system.

Because of the incorporation of the GenAI capability of a Propositional Calculus Engine, each new or modified Classification Society rule can generate new design rules or modify existing ones, which are then reapplied automatically to the design. This process involves verification, the generation of reports, the application of new solutions to existing cases, and, in some instances, the issuance of an MBOM holdup order to prevent rework in the shipbuilding process. Furthermore, this approach facilitates the assessment of the financial implications associated with the implementation of a new rule.

The last approach is not obvious, and it requires extra effort on the CAD development side because it includes some of the advantages of the second solution, applying it in an effortless manner and decreasing overwhelming problems associated with it. In addition, on the customer side, there is no Customer Off the Shelf (COTS) solution. The client should define rules, train the GenAI CAD solution only with shipyard requirements, capabilities, and rules based on previous projects and shipyard physical limits, and then begin to use it. The last step sets a delay time from the implementation of the solution and the advantage that can give the solution; however, when everything is running, it is shown to reduce time-to-market, compliances, and non-conformities.

#### 2.4. CAD/PLM Elements to Be Controlled by IA & Base Tables

GenAI is based on the understanding of the Natural Language and an interaction with people in this way, but design and simulation tools, or PLM/ERP, have a different language to be learned and processed by the GenAI, which implies that understanding CAD as a tool to create, modify, attribute and simulate the different elements involved in modeling process, as well as, the approvals, validations, history management and effectivity applied in PLM tool, or BOM processing in ERP tool, recording resources assignment to the different processes, material bids, material collection, etc. This focuses us on a single direction: we can establish a set of elements to be studied by different IA methodologies based on a new “language”, talked by designers, manufacturers, and all people and tools involved in the process.

The user interacts with the design tool, with commands and the parameters needed to complete all design operations. This is done in a session, following a sequence and generating a set of objects in computer memory, which can be saved to a database, a file, or a PLM element as the final solution. This sets the first element of conflict: not all elements created by the user in a session are saved; some are discarded, and the lesson is in the process, not in the result applied.

To collect all this data, we need to set the use of the CAD as a sensor, getting data from all operations performed by the user. The way to perform it is by establishing a new set of action collectors to acquire all this information and save it (in the same way defined for elements in CAD, a PLM element, or a set of tables in a database).

For this paper, we have selected the database solution approach exploded in Python language with AI libraries like PyTorch and NumPy. The proposed table schema used require three tables to save data generated during the user session:

1. User actions. Each command or action the user has performed in CAD should be saved in the table.
2. In-action steps. Some actions have a simple step, but some others require a form to be filled with a lot of information.
3. In-step, populated attributes/parameters. As well as action steps, each step can require a simple value or a set of them.

After collecting all data from several user design sessions, we can begin to apply deep learning, data classification, and other methodologies to extract new actions for future interactions with users.

This table helps the GenAI CAD layer to learn how to design with CAD or how to interact with simulation tools, but there is a link that needs to be filled by the user, which requirements are performed by the actions done if these elements are from the book of shipyard design rules, the classification society or the customer requirements list. All these documents need to be at the PLM, separated into paragraphs and requirements, and linked to the solution objects generated by CAD/Simulation design tools.

This paper is based on data collected in the design of the cooling system of the main engine, from requirements of power to system limits of the classification society, shipyard design rules, and other mandatory documents applicable to the current design.

#### 2.5. Selection of the Appropriate AI Engine

This part is one of the most complex parts in the full integration process since the final solution’s scalability and performance are highly dependent on this.

A first analysis is performed on which the capabilities of the AI engine spectrum are going to be applied to our solution:

1. Deployment platform: this affects the type of servers where the AI system can run and be deployed. Some of the AI engines are cross-platform, and most of them are supported in the main OS, but there is a small set that only runs in certain OS.
2. Coding language: this is only informative if we are not going to modify the main code but it also needs to be considered as the base to debug certain core problems.

3. Interface: this is the programming language used as the base of the integration. This is important since it is the base of our integration framework and tools. Selecting the right interface can help us achieve our goals in a more efficient way.
4. Deployed libraries: this influences the time to prepare the AI engine to test our integration. There are three bases to deploy:
  - OpenMP is an API that supports multiplatform shared memory multiprocessing programming in C, C++, and Fortran on most platforms, instruction set architectures, and operating systems, including Solaris, AIX, HP-UX, Linux, macOS, and Windows. It consists of a set of compiler directives, library routines, and environment variables that influence runtime behavior [13].
  - OpenCL is a framework for writing programs that execute across heterogeneous platforms consisting of CPU, GPU, DSP, FPGA, and other processors or hardware accelerators. OpenCL specifies programming languages (based on C and C++) for programming these devices and APIs to control the platform and execute programs on the computer devices [14–16].
  - CUDA<sup>®</sup> is a parallel computing platform and API model created by NVIDIA<sup>®</sup>. It allows software developers and software engineers to use a CUDA-enabled GPU for general-purpose processing—an approach termed GPGPU (General-Purpose computing on Graphics Processing Units). The CUDA platform is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements for the execution of computer kernels [17].
5. Automatic Differentiation capabilities: In mathematics and computer algebra, automatic differentiation (AD), also called algorithmic differentiation or computational differentiation, is a set of techniques to numerically evaluate the derivative of a function specified by a computer program [18,19].
6. Pretrained models: this is important to reduce solution preparation time.
7. Kinds of deep learning network support include
  - Recurrent Neural Network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a sequence [20,21].
  - Convolutional Neural Network (CNN, or ConvNet) is a class of deep neural networks wherein connections between the nodes do not form a cycle [21–23].
  - Restricted Boltzmann Machine (RBM) is a generative stochastic artificial neural network that can learn a probability distribution over its set of inputs [24].
  - A Deep Belief Network (DBN) is a generative graphical model or, alternatively, a class of deep neural networks composed of multiple layers of latent variables (hidden units), with connections between the layers but not between units within each layer [25].
8. Actively developed: this is important because the error correction and improvement part of the AI engine relies on a third-party developer. Non-development engines need to be maintained and improved by our company, requiring dedicating resources to it.
9. Open-Source Generative AI Models. Select between the most used models:
10. Generative Adversarial Networks (GAN): these are two Neural Networks (NN), a generator and a discriminator, trained together in a competitive setting. First, NN generates samples, and the second one, the discriminator, tries to evaluate the authenticity of the generated sample [26,27]. In this fully competitive environment, samples become better, and the discriminator improves its capability to distinguish if it is real or fake. The most popular models of this type are Deep Convolutional GAN (DCGAN), StyleGAN, and CycleGAN.
11. Variational Autoencoders (VAE): this model type combines the capability of a generative model with an encoder. This is an unsupervised model type. The capabilities developed by this model are to encode data in a lower dimensional latent space and

decode it back into the original data format. The most popular models are Vector Quantized VAE (VQ-VAE) and  $\beta$ -VAE [28].

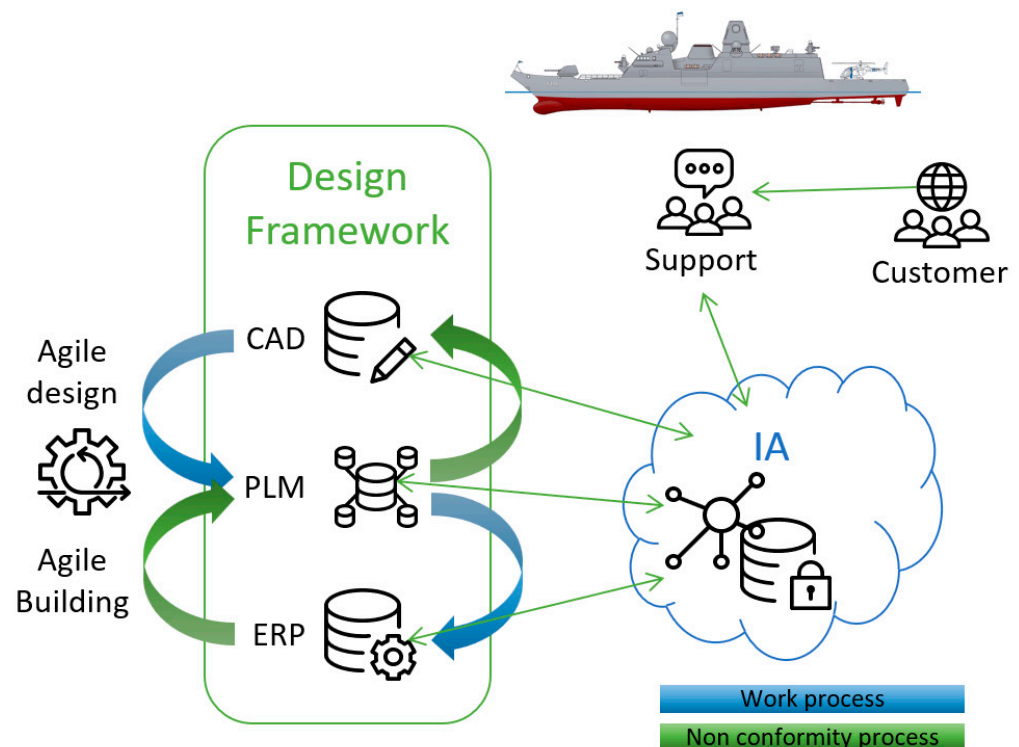
With all these AI requirements, capabilities, and models, we need to select one or a set that helps us prepare our system for the full integration with data generated by the CAD/PLM sensors, which is our training data, and preparing our system to be an actuator based on the action-outputs of our system.

### 3. Application

#### 3.1. GenAI CAD Solution Architecture

The GenAI CAD-based solution developed has required several phases of application on the Software Development side: analyzing your CAD System from the point of view of customer requirements of process automatization, support team information requirements, development team needs for new developments, issue investigation, solution recode and agile deployment as well as include some verification test [29,30].

After the analysis, the selection of the most appropriate GenAI engine forced part of the architecture shown in Figure 2, based on the integration in the product of a set of information interchange methods which allows to generate autonomous learning algorithms to produce background knowledge, as well as generate autonomous design check tools to apply them during night shift, or even, with powerful workstations, during working hours, and finally, generate tools for an autonomous solution application [31].



**Figure 2.** Proposed solution schema.

To implement this, the CAD system and GenAI engine should be programmed, creating two separate tools. The first one is the standard CAD with modified capabilities to inform the GenAI layer in the new interchange language to be understood, but the second one can be a COTS solution, or a new system created by the software provider, depending on provider capabilities. The most important matter for the GenAI system solution is that it is based on an open-source GenAI engine, or a set of them, which matches our needs and allows us quite easy integration [31,32].

### 3.2. Converting a CAD/PLM System in a Set of IoT Sensors

The integration of a computer-aided design (CAD) and product life cycle management (PLM) system with a generative design (GenAI) solution requires the establishment of a series of control points. The incorporation of CAD actions, steps, and parameters into comprehensive design processes allows users to arrive at solutions for given sets of design requirements. The methodology employed to examine this issue allows us to achieve our objective, namely, to transform our CAD into an IoT sensor capable of collecting usage data for subsequent processing in our AI system [31].

Furthermore, additional tools within the process provide information that must be considered. This information pertains to the relationships between elements. The relationships between design elements constitute the foundation of product life cycle management (PLM). Product life cycle management (PLM) enables the identification of direct and hierarchical relationships, which can be studied as the second sensor point to collect data in the design system.

Additionally, simulation is a CAD tool with output values and reports. In this instance, GenAI is capable of reading and understanding the applicable data and translating it into in-design CAD processes, thereby enabling the application of the requisite simulation correction. These two features collectively encompass most of the information that can be extracted from a CAD/PLM system.

However, once the data has been obtained and processed, it is essential to disseminate the findings to the user in a manner that facilitates the enhancement of the design process. To accomplish this, the GenAI system must also be configured to:

- An information system (i.e., a report)
- An autonomous actuator

The information component of the system must be straightforward and interactive with the user interface during the work session. This will enable the user to interact in a guided manner by launching the elements in which they need to intervene, using a step-by-step document report for this purpose. This establishes a feedback system for the proposed actions through step-by-step user validation, which is also stored in the GenAI system.

The CAD system, functioning as an actuator, represents a crucial stage in the process. It must interpret a set of instructions generated by the GenAI system and autonomously modify the design based on the validation and verification of the steps executed in the information phase of the system.

This section will focus on the requirements for transforming a CAD system into an effective IoT sensor/actuator, ensuring the generation of a sufficient quantity of data to construct a valid set. This necessitates the establishment of certain functional specifications.

1. The initial step is to create a CAD sensor tool. It is essential that each command or dialogue within the system generates an entry to be collected by the AI system. The signal contains the following information for processing: a unique identification reference number, a universally unique identifier (UUID) or object identifier (OID), the user who has performed the action, the working mode/module where the action has been performed, the start action date-time, command/dialogue data (e.g., name and information requested), saved data pointing to the second sensor information (point 2), the end action date-time, the parent command/dialogue, and the possible error type.
2. Secondly, the PLM sensor tool: Each item in the working design must generate an entry containing a unique identification reference number, a UUID or OID, the item's unique identification in the PLM system, the date and time of creation or the last modification, attributes (creation requires all, but modification manages only the delta, only changed ones), and 3D geometry in a neutral, interchangeable, and interpretable format.

The combination of these two active sensors enables the AI system to obtain sufficient information during a working user session to create a number of examples. Depending



on the capabilities of the engineering office, the solution can be trained and tested in a relatively short time.

The sensors facilitate the generation of a relation between objects, as well as a historical database of modifications. The AI engine is trained through the analysis of this historical data, which enables the creation of a learning model.

### 3.3. GenAI Models Used in Under-Study Usecase

Since ship design is a long process, the study has been applied to the main engine cooling system of a non-commercial ship. The requirements of the classification society, the shipyard, and the ship owner have been followed with models and design data.

Once the data acquisition structure and the tools implemented in CAD and PLM have been established, the next step is to focus on feeding the training model system. The data is then input into a database, which is subsequently processed using PYTORCH as the AI processing language. The selected GenAI model is a mixture that draws upon multiple methodologies, following Exemplar's VAE system [33].

The training data has been derived from two distinct sources and has been processed as a variety of training data:

1. The requirements for Product Lifecycle Management (PLM) are based on a number of factors, including business rules and production limits established by the shipyard itself, regulations set forth by classification societies, and the specific terms outlined in contractual agreements. The data are presented in textual format and can be classified into two main categories: paragraphs comprising contextual information and requirements, which are text strings that must be fulfilled. These are based on natural language and thus require translation into the GenAI model's PLM/CAD actions language.
2. The PLM/CAD actions, relations, and elements include simulation pre-actions, actions, and results. This is the new language in which our GenAI model is trained.
3. PLM/CAD action timings represent the temporal organization of working processes. This is the actual output of our system, which can assist the user during the design sequence, thereby reducing the time required for development and obtaining a verified final solution. The terminology and grammatical structures employed in this GenAI system proposal are derived from the recently developed language.

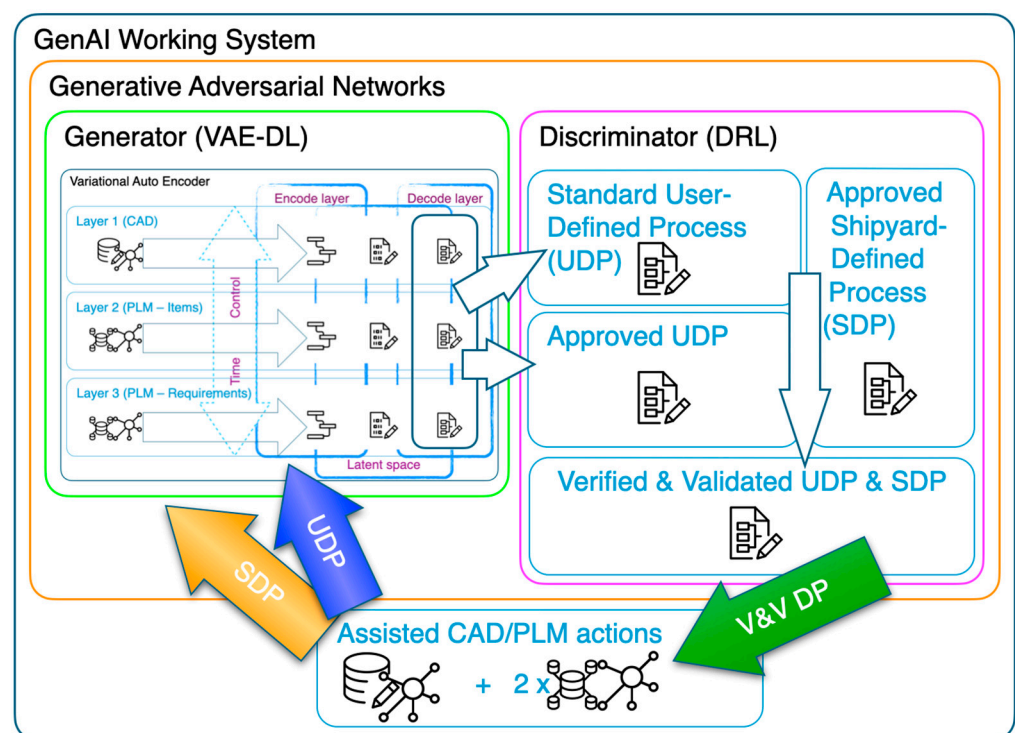
The capabilities of the GenAI VAE model permit the implementation of an encoder that produces the same result as that obtained with curated training data from user procedures. To validate the outputs of the VAE model, the VAE model is employed as the generator of a GAN model, wherein the discriminator is trained using the user's decisions regarding the proposed steps for the processes.

The GenAI system, as illustrated in Figure 3, can be described as follows:

1. The generator is responsible for generating the output based on the input. A VAE model with a deep learning path is employed to train the encode and decode layers. This model handles all known information from the design and requirement processes and manages it through the three layers defined in the previous section. The primary function of this component is to generate a novel set of processes derived from selected requirements for subsequent validation. Initially, this is conducted by the discriminator component and subsequently by the Verification and Validation Committee (V&VC). The V&VC is constituted by processes previously suggested and applied by the user, either partially or in their entirety, or rejected as a non-valid solution for the user. If a process is rejected, it is also necessary for the V&VC to conduct a review.
2. The discriminator is the system that has a set of pre-trained routines that must be updated to include the shipyard, key user, and product owner's main processes that have been approved and rejected by the V&VC. Once a proposed process has been approved, the user is provided with step-by-step guidance, including the suggestion of parameters and instructions on post-validation actions. Substituted actions are stored

in the system as potential alternatives to the proposed design process (illustrated by the purple arrow pointing to the generator), which then feeds the Generator VAE model as new training routines. Rejected actions are identified within the system as invalid alternatives within the proposed design process following the V&VC revision. This facilitates the discriminator's deep reinforcement learning process, enabling it to learn the V&VC methodology through experience.

- The discriminator elements are as follows. As illustrated in the figure, the User-Defined Process (UDP) represents a process that is applied by the user for a specific task to address a particular requirement. The Shipyard-Defined Process (SDP), on the other hand, is a component of the design rules book that is intended to satisfy a particular building standard. The aforementioned processes must be approved by product owners and key users in each discipline (reinforcement). However, for relatively straightforward processes, the approval of a key user is sufficient for implementation. Consequently, the figure depicts both approved user processes and standard processes, which require a lower level of approval. The standard process should be straightforward and serve as the foundation for design and simulation tasks. Once approved, processes must undergo a value and viability check (V&VC), during which key users and product owners determine whether to approve or reject the process value in the discriminator. This decision determines whether the process is certified or not. Only certified processes can be recommended to end users.



**Figure 3.** Continuous training model from design and requirements.

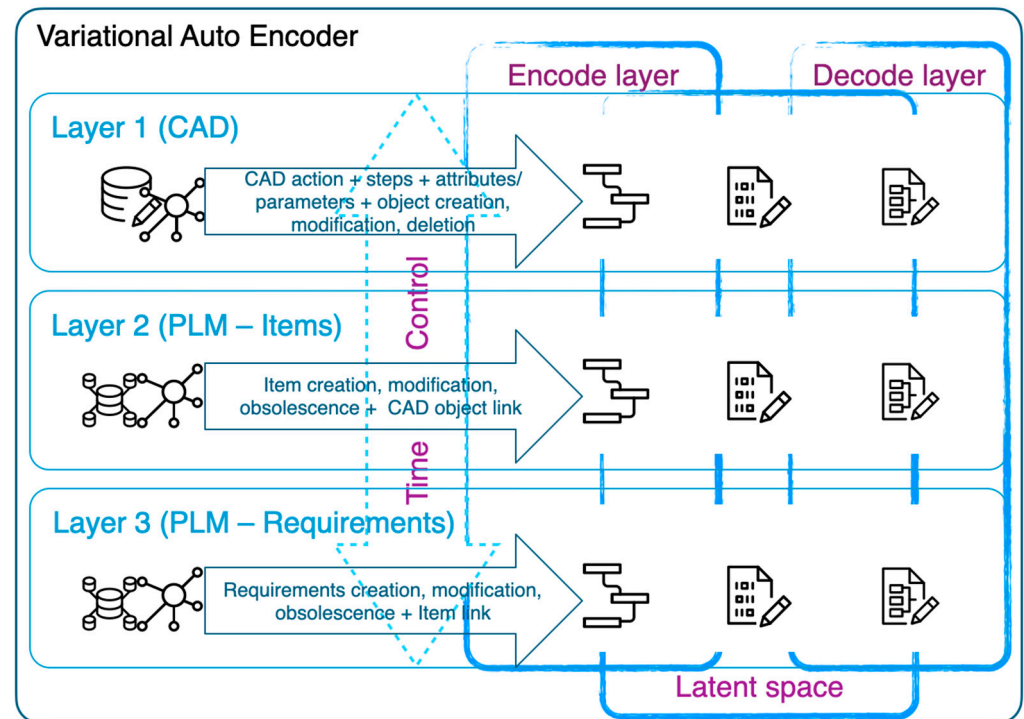
These processes normally come from the experience of users, in all phases of ship-building, but they are also fed by the daily work, although these will be the germ of our Variational Auto Encoder (VAE) process, daily work, which will allow the generator part to “imagine” new processes to solve the same design, shipyard or shipowner requirement.

When the system detects a design routine, it checks if a similar one exists in the system and, based on it, suggests the same, if adaptable, or a new one quite similar and with adapted values. This is verified by the discriminator layer before being suggested to the user, checking not only the process but also the values suggested.

This kind of process must be done very fast to help the designer in a timely manner. This has set a great requirement for the design of the solution.

Data generated, which apparently is not included in any routine during the user design session, feeds the training model of the VAE layer. Training has been performed in all design phases: conceptual, functional, and detail. These are complemented in the manufacturing process, in-design non-conformities, manufacturing process issues, and details.

Each layer in the VAE model for the generator has its own information, which has been populated in the GenAI system through the specific tools developed for it. See Figure 4 to see more details.



**Figure 4.** Layers required for the VAE trainer.

The Variational Auto Encoder processes all layers of information (Figure 4). The Encode layer uses a mix of non-parametric Gaussian mixture, like the Exemplar VAE [33], with a Complex Simultaneous Perturbation Stochastic Approximation (CSPSA) [34] because some of the actions admit a huge number of values for attributes and parameters and needs to be considered as unknown in the training process.

All models have been developed to help users, not to substitute them. As a result, the user always has control over GenAI system, and no data is autogenerated in the final design except fully verified, validated, and curated procedures.

The start of the system requires deactivating suggestions during the first steps. After the design data is mature enough, outputs from VAE begin to interact with GAN discriminator. The discriminator learns from the V&VC decision and is also affected by issues and non-conformities raised by design reviewers. The actions taken have been included in the system as elements of the engineering change type and linked with the elements, getting the three item types: problem, affected, and solution. This information is processed in the VAE second layer to generate specific actions based on corrections done (Figure 4).

Shipyard design rules and limits are incorporated into the system using a special requirement PLM data model object, which allows it to feed the system in a different way. This is a Natural Language Processing (NLP) that converts them into design routines to be performed and checked in the model. When translation is done, from NLP to design routines, the requirements VAE layer can be filled with all this information, and any change in these rules can report its impact on the ongoing design.

#### 4. Discussion

This model helps the user in the daily work process, offering new solutions to the design being carried out, checking that the shipowner and classification society requirements are followed, helping to use the shipyard rules as well as checking design MBOM against building limits, and coordinating design-simulation-requirement checking in a transparent method for the user.

When there is a change in the regulations or the shipowner's requirements, the shipyard or engineering company can upload it into the system to evaluate a priori the changes to be made with the creation, modification, and elimination/obsolescence of items, checking the impact on the existing elements, and, depending on the maturity of them, make decisions regarding the implementation of the change, or the realization of a configuration change, which also affects to the production line.

In this way, the processes are easily implemented in the shipyard and the engineering office, as the system helps to follow them during the change process. The system acts as another user inside the shipyard, creating the non-conformities and following the problem review flow from Problem Request (PR) to Engineering Change Notice (ECN) in the PLM layer.

In addition, this integrated solution offers clear advantages to both CAD/PLM customers and the software supplier. From the CAD customer's point of view, it helps to create a knowledge base for future engineering developments and assists in repetitive tasks by autonomously creating macros to apply to the design. It automatically detects design sequences and offers them to the user as a solution of pre-recorded macros and actions with their attributes and parameters filled in so that it only confirms the validity of each step of the solution and compares and offers alternative sequences to obtain the same design reducing the process time. In this step, compare the users' approach to obtain a result and offer the user the most time-efficient set of commands without showing which user has developed the most efficient design process.

Other advantages obtained by CAD user are: system can auto-detect design anomalies, based on elements with equivalent applied conditions that obtain different information; design rules can be checked and self-applied to the current design without any additional effort, in a fully curated system; it provides a metric to rank users based on their proficiency in using CAD/PLM and the results obtained, allowing the shipyard/technical office to assign jobs more efficiently, in this way system can help to rank jobs and suggest a pre-assignment of tasks to users based on their skills; it also helps new users follow business rules and apply the most efficient sequences in the engineering process, and when an error occurs in any of the CAD or PLM tools, the session steps can be fully recovered and the cause of the error effectively determined, reducing the time required to analyze the error causes, and the required to get the work status before error situation, all steps are saved in the system.

One of the previous points needs to be highlighted: the auto-check and auto-application of the design rules without additional effort because it sets an evolutionary step. Initially, the user has confirmed more options based on a Deep Reinforced Learning (DRL) process for the discriminator; however, the system has learned, based on V&VC answers, which solutions are most applicable to a design process and offers based on curated generator rules, close solution to the previously approved ones.

From a software provider's point of view, the advantages of CAD System testing can be enhanced based on work sequences on the customer side, allowing the validation of new developments, the validation of improvements, black box testing, and out-of-range testing. From the CAD System, error feedback from the customer is more complete, maintaining information on user actions, steps, attributes, and parameters. The system has a way to export the error data to a developer environment, neutralizing the sensitive data using an SVG data transfer matrix, which allows the user to send neutralized data with an accuracy of 99% for issue reproduction. Another capability generated is the improvement of the

“Deadman” analysis for customer issues, errors, and problems, which also helps to inform the CAD developer, in a neutralized report, of the solution usability statistics.

## 5. Conclusions

The incorporation of sophisticated information systems, particularly in the domain of naval design—both civilian and military—has resulted in significant advancements in relational databases, facilitating the identification of diverse groups within the shipyard. This advancement permits the generation of comprehensive datasets that reveal patterns and relationships among disparate entities, thereby enhancing the design process.

Nevertheless, this integration also extends the verification phase of each design. In most cases, auxiliary engineering firms assume responsibility for managing these designs, employing teams of inexperienced personnel who are guided by experts who have received training in the use of the new tools. Although this represents a significant improvement in the process itself, it also increases the number of required revisions. The methodology described has the effect of creating a series of modifications that address defects that may have been overlooked in the initial phases of the process. This ensures that the quality of the final product is maintained without any loss of capability in the digital models or high-fidelity digital twins.

In the development of the GenAI solution, it has been observed that the system is capable of independently verifying information and applying consistent criteria across numerous elements. This feature serves to enhance quality control, as insights from quality gates are provided to users at an early stage of the design phase. Although the initial design scope was constrained, the intricacy of the overall design process has demonstrated the possibility of significant advancements in the learning process. The realization of this potential is contingent upon users adhering to established regulations and the shipyard’s capacity to fulfill the requisite functions.

The implementation of the integrated GenAI system provides customers with augmented capabilities, more accurate designs, and a reduction in human error as they transition from the design phase to production. Furthermore, the AI model database, which logs customer issues and allows for the efficient retrieval of relevant sequences used to generate data, indirectly benefits CAD developers. The efficiency of verification and validation tests can be enhanced, as these sequences can be autonomously executed by the AI engine, with results systematically recorded.

Lastly, updates to system tools can be validated by comparing previous results with customer data, thereby eliminating the need to redo design and construction steps. Consequently, this results in a reduction in the time required for design, verification, and validation processes. The overall design quality improves, allowing personnel with less experience to gain proficiency more quickly as the system provides integrated assistance throughout the CAD tool. This adaptability empowers shipyards to modify design tools swiftly, facilitating upgrades and iterations as GenAI learns to recognize the varying methodologies needed to achieve desired outcomes without compromising the experiential aspects of design within the shipyard.

## 6. Future Developments

Training models need to be improved to enable greater adaptability and early learning, so stochastic optimization methods will continue to be tested to improve the quality of VAE model outputs to generate better requirements-based processes.

Natural language encoding of requirements in CAD/PM language of actions, steps, attributes, and parameters, using complex processes for application in design, simulation, and PLM tools, also requires improvement, so processes to increase capability will continue to be investigated.

This type of behavior also expands the field outside the shipbuilding domain so that it can be applied to others, which can also take advantage of these capabilities.



**Author Contributions:** Conceptualization, A.B.-A. and R.P.F.; methodology, A.B.-A.; software, A.B.-A.; validation, A.B.-A., R.P.F. and F.P.-A.; formal analysis, A.B.-A.; investigation, A.B.-A.; resources, A.B.-A.; data curation, A.B.-A.; writing—original draft preparation, A.B.-A.; writing—review and editing, A.B.-A.; visualization, A.B.-A.; supervision, R.P.F. and F.P.-A.; project administration, R.P.F.; funding acquisition, R.P.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The datasets presented in this article are not currently available for distribution due to their inclusion in an ongoing research project, which requires the permission of the relevant clients before they can be shared. Any requests for access to the datasets should be directed to Arturo Benayas-Ayuso (abenayas@alumnos.upm.es).

**Conflicts of Interest:** Author Dr. Rodrigo Perez Fernandez is employed by the company Siemens Digital Industries Software. Author Mr. Arturo Benayas-Ayuso is employed by the company Accenture—Industry X. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Benayas-Ayuso, A.; Pérez, R. Automated/controlled storage for an efficient MBOM process in the shipbuilding managing the IoT technology. In Proceedings of the Smart Ship Technology 2018, London, UK, 23–24 January 2018; ISBN 978-1-5108-8321-5.
2. Pérez, R.; Benayas-Ayuso, A. Data management for smart ship or how to reduce machine learning cost in IoS applications. In Proceedings of the Smart Ship Technology 2018, London, UK, 23–24 January 2018; ISBN 978-1-5108-8321-5.
3. Porter, M.E.; Heppelmann, J.E. How Smart, Connected Products are Transforming Companies. *Harvard Business Review*, 1 October 2015; ISBN 978-8895181981.
4. Taylor, S. 10 Predictions for the Future of IoT. *Big Data Quarterly*, 31 March 2016. Available online: <https://www.dbta.com/BigDataQuarterly/Articles/10-Predictions-for-the-Future-of-IoT-109996.aspx> (accessed on 8 October 2024).
5. Muñoz, J.A.; Fernández, R.P. CAD tools for designing smart ships in the world of the internet of things (IoT). In Proceedings of the Smart Ship Technology 2017, London, UK, 24–25 January 2017; ISBN 978-1-5108-8299-7.
6. Aslam, S.; Herodotou, H.; Garro, E.; Romero, A.; Burgos, M.; Cassera, A.; Papas, G.; Dias, P.; Michaelides, M. Internet of Ships: A Survey on Architectures, Emerging Applications, and Challenges. *IEEE Internet Things J.* **2020**, *7*, 9714–9727. [[CrossRef](#)]
7. Fernández, R.P.; Regueira, F. How to improve the shipbuilding industry with the Internet of ships concept. *Cienc. Tecnol. Buques* **2022**, *15*, 37–46. [[CrossRef](#)]
8. Perez-Martinez, J.; Fernandez, R.P. Material and Production Optimization of the Ship Design Process by Introducing CADs from Early Design Stages. *J. Mar. Sci. Eng.* **2023**, *11*, 233. [[CrossRef](#)]
9. Pérez-Martinez, J.; Fernández, R.P. Shipbuilding 3D CAD Tools as an Integrated Solution from Concept to Product. *Cienc. Tecnol. Buques* **2021**, *15*, 45–57. [[CrossRef](#)]
10. Liu, G.; Fernández, R.P.; Munoz, J.A.; Regueira, F. Internet of Ships: The Future Ahead. *World J. Eng. Technol.* **2016**, *4*, 220–227. [[CrossRef](#)]
11. Sokolov, S.; Antonova, A.; Knysh, T.; Li, I. Automation of shipbuilding and ship repairing using Internet of Ships technology. *E3S Web Conf.* **2022**, *363*, 01033. [[CrossRef](#)]
12. Fraga-Lamas, P.; Noceda-Davila, D.; Fernández-Caramés, T.M.; Díaz-Bouza, M.A.; Vilar-Montesinos, M. Smart Pipe System for a Shipyard 4.0. *Sensors* **2016**, *16*, 2186. [[CrossRef](#)]
13. Barbosa, C.A.M.; Feijo, B.; Marcelo, D.; Melo, R.; Sérgio, S. Distributed object model for collaborative CAD environments based on design history. *Adv. Eng. Softw.* **2003**, *34*, 621–631. [[CrossRef](#)]
14. Dávila Guzmán, M.A.; Nozal, R.; Gran Tejero, R.; Villarroya-Gaudó, M.; Suárez Gracia, D.; Bosque, J.L. Cooperative CPU, GPU, and FPGA heterogeneous execution with EngineCL. *J. Supercomput.* **2019**, *75*, 1732–1746. [[CrossRef](#)]
15. Faict, T. Exploring OpenCL on a CPU-FPGA Heterogeneous Architecture. Master's Thesis, Ghent University, Ghent, Belgium, 2018. Available online: <https://lib.ugent.be/catalog/rug01:002495122> (accessed on 6 April 2019).
16. Fang, J.; Huang, C.; Tang, T.; Wang, Z. Parallel programming models for heterogeneous many cores: A comprehensive survey. *CCF Trans. High Perform. Comput.* **2020**, *2*, 382–400. [[CrossRef](#)]
17. Ruetsch, G.; Fatica, M. *CUDA Fortran for Scientists and Engineers: Best Practices for Efficient CUDA Fortran Programming*, 2nd ed.; Elsevier: Amsterdam, The Netherlands, 2024; ISBN 978-0-443-21977-1. [[CrossRef](#)]
18. Neidinger, R.D. Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming. *SIAM Rev.* **2010**, *52*, 545–563. [[CrossRef](#)]
19. Baydin, A.G.; Pearlmutter, B.A.; Radul, A.A.; Siskind, J.M. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.* **2018**, *18*, 5595–5637.
20. Qamar, R.; Zardari, B.A. Artificial neural networks: An overview. *Mesopotamian J. Comput. Sci.* **2023**, *2023*, 124–133. [[CrossRef](#)]

21. Bhatti, U.A.; Tang, H.; Wu, G.; Marjan, S.; Hussain, A. Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence. *Int. J. Intell. Syst.* **2023**, *2023*, 8342104. [CrossRef]
22. Zaychenko, Y.; Hamidov, G.; Chapaliuk, B. *The Application of CNN and Hybrid Networks in Medical Images Processing and Cancer Classification*; Cambridge Scholars Publishing: Newcastle upon Tyne, UK, 2023; ISBN 978-1-5275-1539-0.
23. Moutsis, S.N.; Tsintotas, K.A.; Kansizoglou, I.; Gasteratos, A. Evaluating the Performance of Mobile-Convolutional Neural Networks for Spatial and Temporal Human Action Recognition Analysis. *Robotics* **2023**, *12*, 167. [CrossRef]
24. Smolensky, P. Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*; Rumelhart, D.E., McClelland, J.L., Eds.; MIT Press: Cambridge, MA, USA, 1986; pp. 194–281, ISBN 0-262-68053-X.
25. Hinton, G. Deep belief networks. *Scholarpedia* **2009**, *4*, 5947. [CrossRef]
26. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. *arXiv* **2014**. [CrossRef]
27. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2016**. [CrossRef]
28. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2022**. [CrossRef]
29. Harjamäki, J.; Rantanen, P.; Lahtinen, D.; Sillberg, P.; Saari, M.; Grönman, J.; Rasheed, Z.; Sami, A.M.; Abrahamsson, P. *The Report of 85 AI Tools: GenAI Content Production: Enhancing Repeatability and Automation with ChatGPT*; Tampere University: Tampere, Finland, 2024; ISBN 978-952-03-3502-1. Available online: <https://urn.fi/URN:ISBN:978-952-03-3502-1> (accessed on 8 October 2024).
30. Taiwo, R.; Bello, I.T.; Abdulai, S.F.; Yussif, A.M.; Salami, B.A.; Saka, A.; Zayed, T. Generative AI in the Construction Industry: A State-of-the-art Analysis. *arXiv* **2024**. [CrossRef]
31. Daase, C.; Haertel, C.; Nahhas, A.; Zeier, A.; Ramesohl, A.; Turowski, K. On the Current State of Generative Artificial Intelligence: A Conceptual Model of Potentials and Challenges. In Proceedings of the 26th International Conference on Enterprise Information Systems—Volume 1: ICEIS, Angers, France, 27–29 April 2024; pp. 845–856. [CrossRef]
32. Santos LD, S.; Ribeiro Filho PR, C.F.; Macêdo, E.N. Belt rotation in pipe conveyors: Development of an overlap monitoring system using digital twins, industrial Internet of things, and autoregressive language models. *Measurement* **2024**, *230*, 114546. [CrossRef]
33. Norouzi, S.; Fleet, D.J.; Norouzi, M. Exemplar VAE: Linking Generative Models, Nearest Neighbor, and Data Augmentation. Neural Information Processing Systems. In *Advances in Neural Information Processing Systems 33, Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada, 6–12 December 2020*; MIT Press: Cambridge, MA, USA, 2020; Volume 11, pp. 8753–8764, ISBN 9781713829546.
34. Gidi, J.; Candia, B.; Muñoz-Moller, A.D.; Rojas, A.; Pereira, L.; Muñoz, M.; Zambrano, L.; Delgado, A. Stochastic optimization algorithms for quantum applications. *Phys. Rev. A* **2023**, *108*, 032409. [CrossRef]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.