White paper

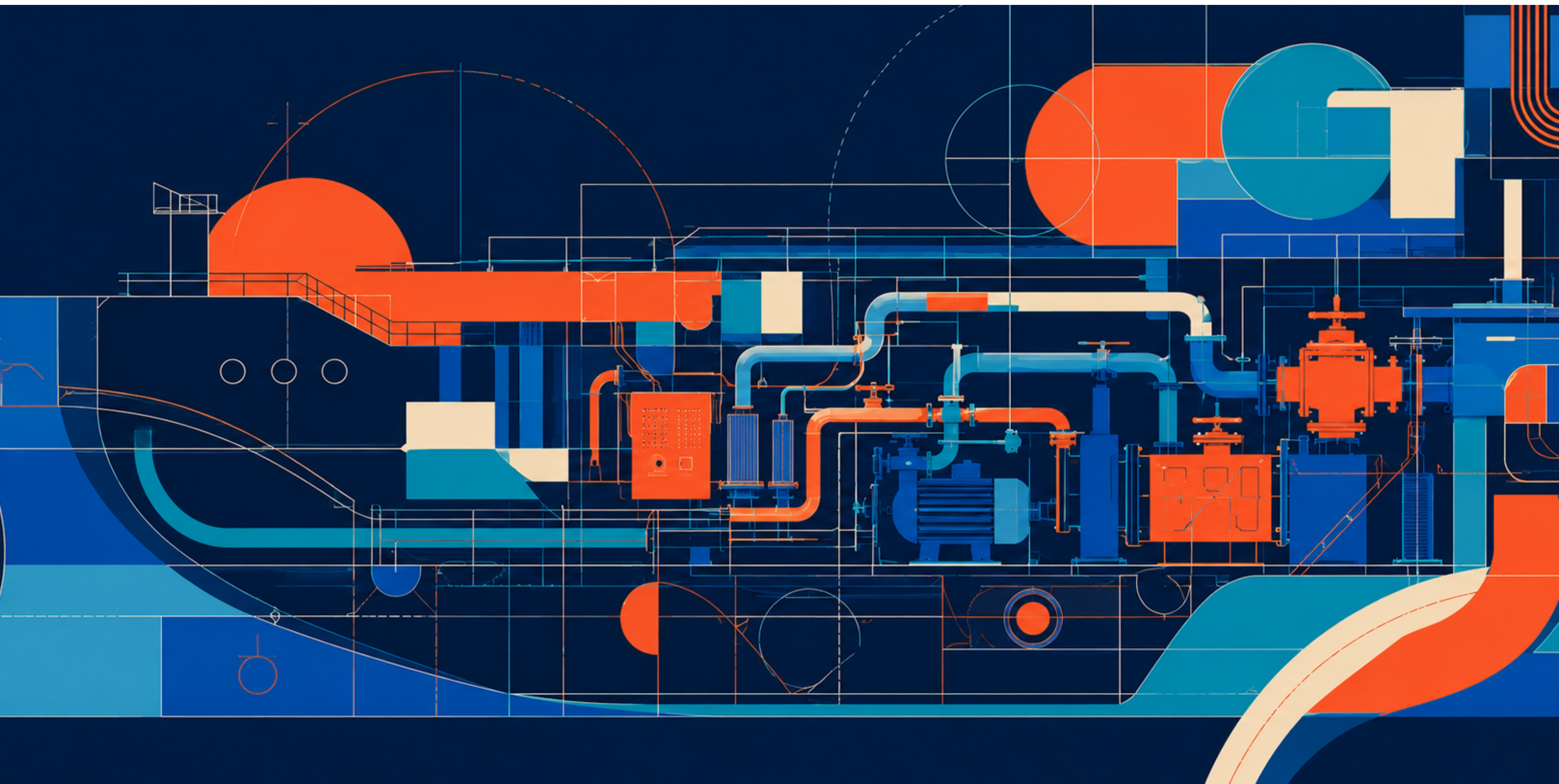
September 23, 2026

Improving Maritime Engineering Efficiency via Design Retrieval

An introduction to finding reusable designs with machine learning

Daniel Jacobsen

danieljacobsen.com



Executive summary

Maritime engineer-to-order (ETO) deliveries require a large engineering effort: a single large delivery can involve approximately 100,000 engineering hours. Early design choices can add downstream detail-design and construction work. Procurement, fabrication and delivery all wait on engineering deliverables.

Relevant previous designs are often already in the archive. When they are reused, there is less repeated work and those later activities can begin earlier. Finding them is the challenge. Conventional archive search depends on project numbers, filenames, classifications and individual memory. It cannot directly compare the geometry, connections and interfaces contained within engineering designs.

Machine learning can compare the content of a new design with historical designs. The choice of representation is critical: each one preserves some aspects of a design and omits others. Metadata, text, images and geometry record different information about the same design. Graphs are distinctive where similarity depends on adjacency, interfaces, containment and connectivity: CAD boundary representations, P&IDs, assemblies and electrical diagrams can then be compared through their topologies, relationships and attributes. A graph's nodes and edges can also carry the other representations as attributes.

Published research demonstrates the underlying methods in mechanical CAD, process-plant matching and ship-piping graph retrieval. This white paper proposes a retrieval workflow for maritime ETO in which the search finds similar designs, checks on duty and interfaces narrow them to suitable ones, and the engineer decides which is worth adapting.

A first pilot should cover one kind of design that the organisation delivers repeatedly, and compare graph retrieval with the way engineers search today. It succeeds if engineers find and reuse precedents faster than they do now.

1 The Cost and Time Problem in Maritime ETO Engineering

Maritime engineer-to-order (ETO) deliveries often represent very large investments for the equipment supplier. Leaders in the industry estimate that one complete delivery can require on the order of 100,000 engineering hours.[1] At a fully burdened engineering cost of $85 per hour, that represents $8.5 million of total engineering effort.

This scale follows from the nature of the work. Each delivery is configured for a customer's requirements, and maritime products combine many interacting systems, interfaces and regulatory constraints. They are often engineered for one vessel or a short series rather than distributed across a high production volume.[2] A systematic review describes ETO deliveries as taking months or years [Karhunen et al., 2024].

[1] From confidential personal communications with maritime ETO leaders. The figure is an illustrative project-scale estimate, not an industry benchmark.

[2] Some maritime ETO segments build longer series, in which case the engineering effort per vessel can be lower.

Engineering effort also affects project time costs. Later work cannot start until design is issued: class and flag approval, procurement, fabrication, and interface coordination all wait on those deliverables. Early design choices can also directly increase the work that follows. In naval shipbuilding, for example, under-sizing a vessel during concept design packs more equipment into a smaller hull, which adds detail-design and construction hours [Keane et al., 2016]. The cost of engineering is therefore not confined to the design office.

Customer-specific design, verification and approval are core to the maritime ETO value proposition. The opportunity is to reduce avoidable repetition. That would cut both the monetary cost of engineering and the time cost that follows from it. Not all of the 100,000 hours are open to that. Better retrieval can affect the work for which a reusable precedent exists in the archive and is found while the design is still open, less the effort of adapting and verifying it. This white paper does not estimate that share. For one kind of design, the pilot in Section 7 measures how often engineers found and reused a precedent and how much adapting it took.

2 The Opportunity and Challenge of Design Reuse

2.1 The opportunity

A new delivery need not imply that every design needs to be created from scratch. Consider an engineer beginning an HVAC design for a new vessel. The organisation may already have delivered related systems for similar vessels: assemblies, components, interfaces and engineering choices that would be a better starting point than a blank sheet.

If those precedents can be found in time, less work is repeated and design outputs can reach approvals, procurement, fabrication and interface coordination sooner.

Ship piping designers already reuse routes from similar past projects rather than creating them from scratch. That saves design time and starts from layouts that have already been verified [Kong et al., 2026]. Historical ETO projects can also be analysed for reference architectures worth reusing [Løkkegaard et al., 2023]. The ETO literature still identifies a need for more systematic design reuse [Karhunen et al., 2024].

2.2 The challenge

Those relevant designs still have to be found.

For the engineer beginning the HVAC design, the difficulty is to identify the few useful designs among a large archive, most of which will not help.

Finding them with conventional archive search requires knowing how previous work was named, classified and divided between files. A relevant subsystem may sit inside a larger delivery, use different terminology, or belong to another team's project.

Conventional search works when the engineer already knows the project number, filename, document class or exact term used in the earlier project. Those fields describe the record around the design rather than its engineering content. They do not expose the geometry inside a CAD model, the connections in a P&ID, the topology of an electrical diagram or the interfaces within an assembly.

Two designs can therefore be similar even when their descriptions differ. Files with similar names may equally contain different engineering structures. When classification is weak or inconsistent, large design archives are harder to exploit [Hsiao et al., 2016].

Engineers also search through colleagues, because people can supply project context and design rationale that documents do not record [Hertzum and Pejtersen, 2000]. Retrieval then depends on individual memory as well as on the archive itself.

A previous design cannot reduce engineering work if it cannot be found. The practical result is manual searching, recreation, and reliance on who happens to remember a similar project.

Key takeaway

Retrieval is the bridge between stored design knowledge and practical reuse.

3 The Design Retrieval Loop

Retrieval bridges the archive and the new design, and that bridge is crossed more than once. Consider again the engineer beginning the HVAC design for a new vessel. What is known about the job at that moment is the *query*: the duty it must meet, the space it must occupy, the rules it must satisfy. Retrieval searches the archive with that query and returns a shortlist of previous designs to inspect. Each archived design is termed a *source*.

That search has four steps:

1. form a query from the information available in the current design;
2. represent the query and archived designs in a comparable form;
3. compare them and rank the archive by similarity; and
4. present the leading candidates for engineering inspection.

The engineer reviews the most promising candidate, adapts what fits and verifies the result against the requirements. After that work the emerging design is richer than it was, and the richer design becomes the next query. This is the *design retrieval loop* (Figure 1).

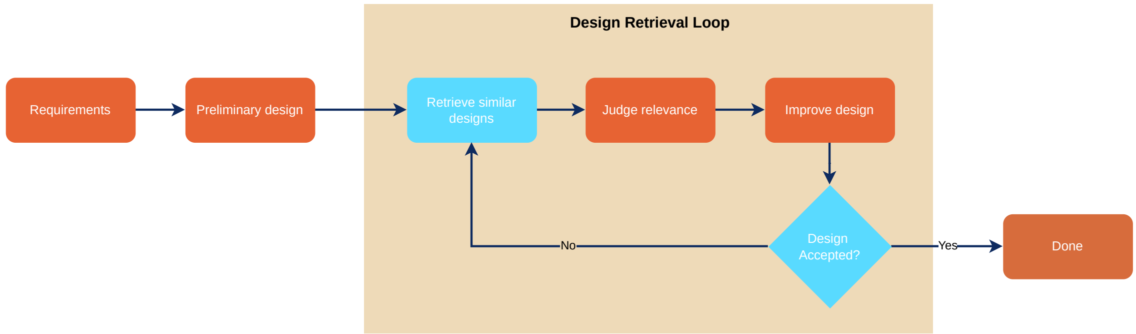


Figure 1: The design retrieval loop. The archive is searched at each pass; adapted work becomes the next query.

3.1 Queries evolve as designs mature

Each turn of the loop searches with more of the emerging design, maturing the query. Table 1 follows that change through an example CAD design; the principle is the same for P&IDs and single-line diagrams.

Design maturity	What the engineer can search with
Requirements	Text, parameters and operating conditions
Early concept	Sketches, envelopes and rough arrangement
Preliminary design	Partial CAD geometry and principal interfaces
Developed design	Detailed CAD geometry, interfaces and assemblies

Table 1: How a query thickens as a design matures, shown here for a CAD design.

The query begins as a description of the need and ends as the design itself.

The query is the least specific at the beginning, when the arrangement is relatively open and the choices being made will drive detail-design and construction work for the rest of the project (Section 1). It is richest at the end, when most of what a precedent might have changed is already fixed. The most difficult search is thus needed when its outcome would be most helpful.

Key takeaway

The most difficult search comes first. At the beginning the query is least specific, yet that is when the design is still open and the choices being made shape the work that follows.

4 Finding Similar Designs with Machine Learning

The engineer beginning the HVAC design now has a query: the duty, the space, the rules. Each archived design is a *source*. Every pass of the design retrieval loop turns on judging how similar that query is to a source in the archive.

There are two ways to estimate that similarity.

The first is to specify the criterion in advance and apply it deterministically. That is the conventional approach. It typically uses metadata such as filenames, tags, classifications and design parameters, for example the width of a pipe, as input to a fixed procedure that returns a similarity score. The same idea can be applied to structured design data: an engineer writes a matching rule, and an algorithm applies it.

A major advantage of specified rules is that they are inspectable. They are also hard to make adequate for complex systems. Similarity then depends on many interacting relations that are not the same from project to project, and that are difficult to capture as a complete rule set. Ordinary archive search has a more basic limit: it compares the record around a design, that is, metadata, rather than the design itself, and nothing guarantees that the record tracks the design reliably.

The second way is to *learn* the similarity criterion from the design archive, via machine learning (ML). On published retrieval tasks, that learned comparison has found more of the right candidates than hand-engineered scoring rules [Li et al., 2019]. A preprint on mechanical CAD retrieval reports that ML-based retrieval searched an archive about a hundred times faster than heuristic (non-ML) graph matching, and with higher human-rated retrieval quality [Quan et al., 2024]. ML methods can be trained without requiring engineers to label similar pairs [Qin et al., 2025, Quan et al., 2024]. Those results are from adjacent technical domains, not from maritime ETO. Whether they appear as faster, cheaper finding of reusable maritime precedents is what a first pilot measures.

4.1 Design representations

A query and a source can only be compared if they are represented using the same representation category. Table 2 summarises the five main, high-level categories. They are complementary rather than stages in a ladder: the same design artefact can be represented via filename, a page image, a geometric solid, and so on. Every representation preserves some aspects of a design and omits others.

Representation	What it provides	What it misses
Metadata	Records and fields around the artefact: project number, tags, dimensions, classification	Shape, interfaces and component relationships
Text	Wording of requirements, specifications and notes	Explicit shapes, connections and topology
Visual	Pixel content of drawings, photographs or rendered CAD views	native CAD model information, explicit shapes, connections and topology
Geometric	Spatial shape of a part or assembly, either sampled (point clouds, meshes, voxels) or CAD-native B-rep	Exact curves and topology when only sampled; relationships between separate components when only a single solid; non-spatial system connectivity
Graph	Entities and relationships: components, connections, interfaces and containment. Node and edge attributes can carry other representations	Page appearance; any relationship or attribute that was not recorded or extracted

Table 2: Each representation records different information, and so leaves out different engineering structure.

4.2 The query–similarity process

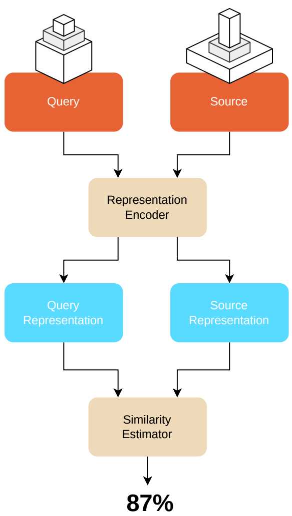


Figure 2: The query–similarity process. The query and a source are mapped into comparable representations and compared to produce a similarity estimate.

Figure 2 is a close-up of the comparison step in the design retrieval loop. The query is formed from the information available in the current design iteration. Each source is an archived design, or a design element of comparable scope to the query: a part, assembly, system module or diagram section. For the HVAC search, that unit might be a previous module rather than a whole-project delivery. Both query and source are mapped into a comparable representation. Those representations are compared to produce a similarity estimate.

The query can consist of one or more representation types. A transformation can convert it into a type already used in the archive, and a source can be converted into the type the query provides.

4.3 Transforming representations

A representation can be converted into another. A drawing can be read as text; a picture of equipment and connections can be turned into a graph; a CAD solid can be sampled as a point cloud or rendered as an image. Each conversion may lose information; even so, such transformations may still be beneficial if they put query and source into a shared type, or if a more accurate similarity method can be applied to the result.

Both query and source representations can be transformed. One case arises when rendered drawings or PDFs are to be searched as graphs: the entities and connections must first be extracted. That extraction is a digitisation step: it produces a graph that can later be indexed and compared. Separate symbol and line detectors can miss how equipment is connected. Recent work on P&IDs therefore extracts symbols and their interconnections together, producing an attributed graph of equipment, instrumentation and connections [Stürmer et al., 2025]. The same extraction is needed for other engineering diagrams stored as images or PDFs. Native CAD and structured diagram files can be mapped to graphs without that raster step.

A mixed archive may need both mappings. Which conversions are required follows from what the query provides and the type each source is already stored in.

4.4 Selecting representations

The first choice in building a similarity system is deciding what to compare. That is, which *representations* to use for the query and for the source designs. Each type in Table 2 retains different content, so each shows a different kind of resemblance.

If the archive is searched by, say, project number or tag, the comparison is of metadata. Metadata is cheap to store and easy to inspect when the relevant fields exist and are consistent: the engineer can see which project numbers, tags or parameter values are shared. However, terminology differs between projects, fields may be incomplete, and a reusable subsystem may be hidden inside a larger delivery.

A text representation records the wording of requirements, specifications and notes. For the HVAC engineer at the start of the job, it is often all that exists: a duty, a capacity or a written constraint. It does not directly record shapes, connections or topology.

A visual representation may use drawings, photographs, rendered CAD views, and so on. It is the right object when the distinction of interest is visible on the page or in a view. Pixels record appearance rather than the native CAD information behind it, so explicit shapes, connections and topology are absent and page layout can be mistaken for engineering structure.

A geometric representation records spatial shape, and it does so in one of two ways: by sampling the outside of a design, or by carrying the CAD model's own definition of the solid.

Sampled geometry approximates external form. Meshes, point clouds and voxels record dimensions, proportions and the relative placement of components at whatever resolution was used. Exact curves and surfaces, engineering topology, features below that resolution, hidden internal geometry and functional meaning do not survive the sampling, and a point cloud does not record which surfaces are joined. Two fan units with the same envelope and proportions therefore resemble each other in sampled form whether or not they are built the same way.

A CAD-native boundary representation, or B-rep, carries the model's own definition of the solid: parametric surfaces and curves bounded into faces, edges and vertices, together with the geometric topology of which faces meet, how edges are oriented and how the solid is closed. Exporting that model as an image, mesh or point cloud discards part of it [Heidari and Iosifidis, 2025].

What a B-rep omits is everything outside the single solid: design history, feature intent, tolerances, materials, functional roles, and how one component relates to another. Those come from the assembly, the engineering record or the diagram.

A graph representation records what a design is made of and how the parts relate: components, connections, interfaces and containment. It is the one category that can carry the others, because the shape of a part, its metadata and its description can all be attached to the node that stands for it.

The query decides where to start. Whatever the emerging design already provides, whether a written duty, a preliminary diagram or a partial CAD model, sets the representation, or representations, used for the search, and the source designs can be transformed to match.

4.5 Estimating similarity

The comparison method follows from the representation. Text can be compared by keywords or by numerical encodings of wording, images by visual descriptors, metadata by exact fields and numerical filters. Each of those compares description, appearance or overall shape more readily than it compares components, connections and interfaces, and each remains a useful complement, most of all while the query is still a name, a paragraph or a picture.

Graph methods are the focus for a narrower reason than a general preference for graphs. Maritime ETO designs are variable-sized assemblies, and whether two of them are similar often depends on component adjacency, interfaces, containment and connectivity, not only on overall shape or local surface detail.

Representations that preserve those relationships retrieve more relevant CAD candidates than alternatives built on local geometry, signed-distance fields or unstructured samples. Learned models that keep native B-rep geometry and topology outperform point-cloud, mesh and voxel models on CAD tasks where parts share topology but vary in shape [Jayaraman et al., 2021]. A preprint on mechanical CAD retrieval represents faces as nodes and their interfaces as edges, keeping local geometry and global topology together; on human-evaluated retrieval that representation outperformed a locally focused B-rep network and a signed-distance method, neither of which captures overall topological structure [Quan et al., 2024].

Key takeaway

Design reuse depends on structure: components, connections and interfaces. When that structure is what must be matched, graph methods compare it directly; other representations remain useful complements.

Choosing a graph representation has five practical consequences:

- connectivity and interfaces are recorded as relationships rather than inferred from a name or an image;
- one abstraction can describe a CAD boundary representation, an assembly, a P&ID and an electrical diagram;
- other representations of the same design can be stored as attributes on nodes and edges, so one graph can combine sampled shape, B-rep properties and interface data;
- an incomplete preliminary design can be compared with a relevant part of a historical design rather than with two complete models (Section 4.9); and
- a result can be shown as matched equipment, connections or interfaces, not only as a distance score.

On graphs, similarity can still be specified as a matching rule or learned from the archive. Specified matching expresses interface constraints directly, stays inspectable and needs no training examples [Rantala et al., 2019]. It examines one pair at a time, so the archive cannot be ranked from representations calculated before the query exists. Learning the criterion is the focus here, because of the accuracy, speed and labelling-cost results already noted, but specified matching remains a practical option.

4.6 Representing engineering designs as graphs

In Figure 3, *nodes* stand for components and *edges* for the relationships between them; both can carry attributes. The pattern of connection is the design's topology, recorded separately from geometry such as position, dimensions or surface shape.

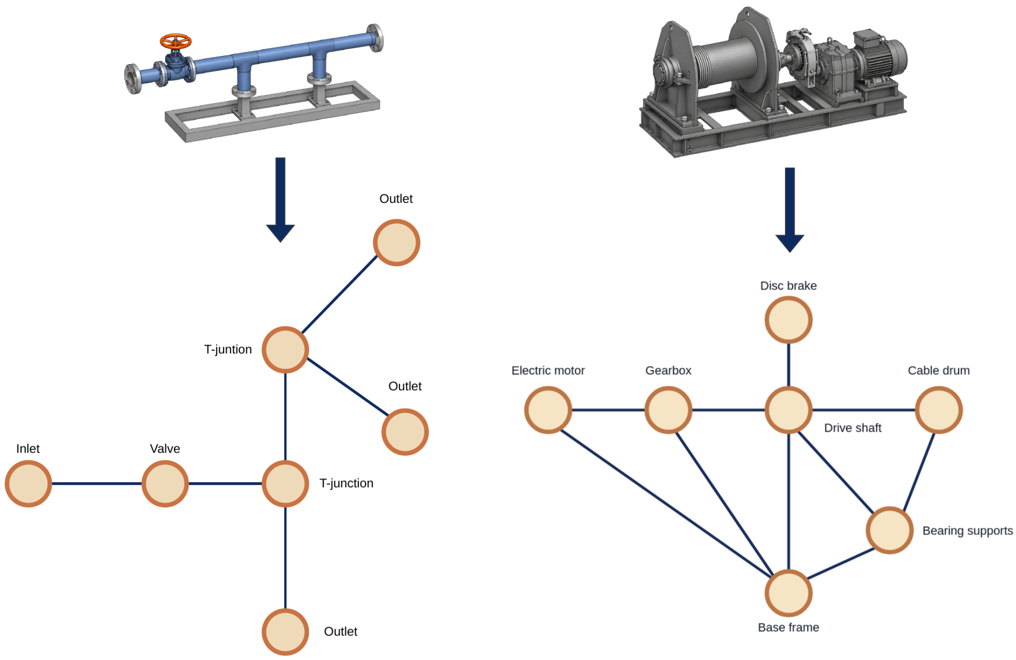


Figure 3: Engineering designs as graphs. Left: a piping manifold mapped to nodes for the inlet, valve, T-junctions and outlets, with edges for flow connections. Right: a winch assembly mapped to motor, gearbox, shaft, brake, drum, bearings and base frame, with edges for mechanical interfaces. Not shown: nodes and edges (links) can have attributes for a richer presentation. For example, a node for a 3D part could have attributes describing its shape.

There is no single graph inherent in a design. The representation follows the retrieval question. For the HVAC search, the same assembly might be represented by components and their physical mates, or by equipment and the flows between them. Table 3 gives illustrative choices.

Design object	Nodes	Relationships
CAD B-rep	Faces	Shared boundaries and adjacency
Assembly	Components	Connections, interfaces or mates
P&ID	Equipment, valves, instruments	Pipes, flow and signals
Electrical diagram	Buses, breakers and loads	Electrical connections

Table 3: Illustrative graph representations of engineering designs. These graphs can nest: a B-rep graph inside a part, an assembly graph among parts, and an interface or system graph for functional connections.

For a P&ID, the graph records equipment and connections and ignores where symbols lie on the page. For a B-rep, faces become nodes and shared boundaries become edges, with surface type, dimensions and orientation as attributes, so the graph keeps both the geometry and the topology.

Those graphs are not alternatives to one another; they nest. Within each part, a B-rep graph represents faces, edges and their connectivity. Among parts, an assembly graph represents mechanical or spatial relationships. Across the wider design, an interface or system graph represents functional connections such as piping, electrical or data flow. Sampled shape from meshes or point clouds, and appearance from rendered images, attach to the relevant part nodes. What none of them supplies by itself is functional meaning: spatial proximity does not identify the function of a connection, which has to come from the source systems or from preprocessing.

Key takeaway

A graph can merge several representations of the same design. Nodes and edges record entities and relationships; attributes on those nodes and edges can carry sampled shape, native B-rep properties or other engineering information.

Approximate shape tells the model what components look like, B-rep describes their precise geometry and internal boundary topology, and assembly and interface graphs describe how they fit together and function within the wider design.

4.7 Learning similarity

A model that learns similarity needs something to learn from. The usual supervised route is to show it examples of designs that should count as similar. However, the archive typically does not contain those labels, and creating them would mean asking engineers to mark up pairs of past designs before the system could be tried at all. Semi-supervised methods mix a few such labels with unlabelled archive data.

Self-supervised methods reduce the labelling burden by generating the training signal from the archive itself, for example by treating two views of one design as a pair that should match. That is the practical route here, because it needs little manual labelling, and the archive itself becomes the training data. A small set of designs already known to be related is still needed to check what the model has learned (Section 7). A pretrained model of another company's designs is unlikely to be available, and large pretrained graph models, sometimes called graph foundation models, are probably a later option rather than a shortcut past this step.

A graph neural network learns by *message passing*: each node repeatedly combines its own attributes with information from its neighbours. After a few rounds, the model describes the piping manifold and the winch in Figure 3 as connected equipment. Two designs can share component types and still differ in how those components are linked, and the model can then detect that difference.

4.8 Estimating similarity through learned encodings

Figure 2 shows how one query is compared with one source. When similarity is learned, the encoder in that figure is the graph neural network just described. It turns each design, on its own, into an *encoding*: a short list of numbers that summarises the whole design, also called an embedding. Designs the model has learned to treat as similar receive similar numbers. The similarity estimator then measures how close the query encoding is to the source encoding, and returns the estimate shown in the figure.

Because each source passes through the encoder separately, its encoding does not depend on the query. Encodings for the whole archive can therefore be calculated once, in advance, and stored. When the engineer searches, only the query is encoded, and it is compared with every stored encoding in turn. The closest sources become the search results. On its own data, the CAD preprint already cited reports search of this kind as about a hundred times faster than a specified graph-matching procedure that took more than twelve hours per query, and as modestly better in the designs people judged it had found [Quan et al., 2024].

An alternative is *pairwise matching*, in which a model takes the query and one source together and matches the components and connections of one against the other [Li et al., 2019]. It can find correspondences that two separate encodings may blur. Nothing can be calculated in advance, however: every source compared this way is processed again with the query each time the engineer searches.

4.9 Partial queries

For most of the HVAC job, the query is incomplete: a preliminary diagram or a partial CAD model, not a finished design. Most of the evidence cited so far compares complete designs with complete designs, which is the easier case.

An encoding summarises a whole design. If much of the query is still missing, its summary differs from that of an archived design that contains the same equipment and connections alongside much more. The two can therefore end up far apart, even though the archived design is exactly the precedent the engineer wants. Search by stored encodings alone works least well when the query is at its least complete.

Archived designs can be decomposed so that a query meets items of its own size (Section 4.2). A preliminary HVAC module is then compared with previous HVAC modules, rather than with whole-project deliveries in which it would be a small part of a much larger summary. Much of that division is often already recorded: CAD assemblies and product structures show what each module contains, and diagrams group equipment by system tags. Where only drawings or PDFs exist, the boundaries have to be extracted and checked along with the rest of the design.

With units of comparable size, the search can run in two steps. Stored encodings quickly produce a shortlist of candidate modules, and pairwise matching examines only that shortlist, looking for the part of each candidate that corresponds to the partial query. The second step can only re-examine what the first returned: a precedent missed by the shortlist never reaches pairwise matching. That is why the choice of unit matters most while the query is incomplete.

5 From similarity to reuse

A design that closely resembles the query can still take a great deal of work to adapt. Two designs may look almost identical and need very different amounts of change before either fits a new job, and the amount of change decides whether reuse pays [Udoyen and Rosen, 2009].

On the HVAC engineer's shortlist, the module at the top has nearly the same fans, coils and duct connections as the preliminary diagram. It was built for a much larger cooling load, however, and its control interfaces follow a standard the current customer does not use. The third module resembles the query less. It was built for a similar load, to the same interface standard, and would need a different fan and little else. Nothing in the ranking says so. The engineer finds out only by opening both designs and comparing them.

A ranked list that shows only names and scores tends to be read from the top. The engineer opens the first design, finds it plausible and starts adapting it, even when another candidate, or a new design, would have been better or safer. If several candidates are shown together, each with the components, connections and interfaces that matched the query and the archived files behind it, the engineer can compare them before committing to one.

Whether a candidate is suitable depends on checks, and some of them can be built into the search (Figure 4). Had the archive recorded each module's cooling load and interface standard as attributes, both differences would have counted against the first module, and the third could have ranked above it. Other checks stay with the engineer, such as whether a design would pass the current classification rules. Which checks belong where depends on the kind of design: a process module is judged by its duty, interfaces and fire-safety requirements, a pump skid by its duty, base frame and connection points, a switchboard by its voltage and protection scheme, and so on.

Key takeaway

A similar design is not automatically reusable. The search finds similar designs. Checks on duty, interfaces and rules decide which are suitable, and some of those checks can be built into the search. The engineer decides whether adapting a suitable design is worth the work.

Figure 4: From similarity to reuse. A similar design is reused only after it passes the checks for suitability and the engineer judges it worth adapting.

6 Three maritime ETO examples

The HVAC engineer's module is one of many kinds of design a maritime ETO supplier could search for in this way. Deck machinery, ballast water treatment systems, cargo handling equipment and ship automation panels are others. The three examples below are the HVAC module, a mechanical equipment assembly and an electrical distribution package. Each starts from a different kind of design record: a process diagram, a CAD model and a single-line diagram. Between them they show what changes from one kind of design to the next: the query, the reasons for rejecting a candidate, and the split between checks the search can make and checks left to the engineer. Table 4 shows each of the three as a graph.

6.1 Process-system module

The HVAC engineer starts from a preliminary diagram of the module: the fans and coils already chosen, the duct and pipe connections drawn so far, and the cooling duty the system must meet. The search looks for previous HVAC modules with related equipment and connections. A candidate is worth adapting if it gives a starting arrangement of equipment and connections for the current duty. Different operating conditions rule it out, and so do incompatible piping or control interfaces, different fire-safety requirements for ducts and dampers, or equipment that is no longer made.

Something close to this search has been done for ship piping: similar piping systems from past ships were retrieved by comparing their P&IDs as graphs [Kong et al., 2026]. Those queries were finished diagrams. The closest test of an unfinished one comes from process plants. Starting from an early diagram in which the main equipment and connections were known but not yet sized, graph matching found similar components in earlier plants whose parameters could be reused [Rantala et al., 2019]. Neither study covered HVAC. Between them, though, they tested both conditions the HVAC engineer's search has: a ship system, and a diagram that is still incomplete.

6.2 Mechanical equipment assembly

Pump packages and skids are delivered as assemblies on a base frame, with a fixed footprint and fixed points where pipes and cables connect. The engineer designing one starts from a partial CAD model and the duty the package must meet. The search compares the model's geometry and component structure with those of previous packages. A match is useful when its base frame, envelope and connection points could suit the new installation. A different duty rules it out, and so do incompatible connections or components that are no longer available.

Retrieval of this kind, learned without anyone labelling which parts are similar, has been shown for mechanical CAD parts [Jayaraman et al., 2021, Qin et al., 2025]. A preprint adds an evaluation in which people with mechanical training judged the retrieved parts, and a simple test on assemblies that retrieved each part separately and counted the matches [Quan et al., 2024]. These methods compare the geometry and boundary topology of individual parts. For a pump package, the search would also have to compare duty and connection points, and start from a partial model.

6.3 Electrical distribution package

For an electrical distribution package, the query is a preliminary single-line diagram and the loads it must supply, and the search looks for comparable switchboard sections or distribution architectures. The engineer wants a bus arrangement, protection scheme and set of interfaces that could serve the new installation. A different voltage or earthing system rules a candidate out, as does switchgear no longer available or protection discrimination that class would not approve for this installation. A single-line diagram records the buses, breakers and feeders and how they are connected, so it can be turned into a graph and searched in the same way as the HVAC engineer's diagram. Of the three examples, this one has the least evidence behind it: none of the studies cited here searched single-line diagrams, so the table marks it as proposed rather than documented.

The three also differ in how much work the archive needs before it can be searched. Parts and assemblies stored as native CAD can be mapped to graphs directly. Older P&IDs may exist only as PDFs or scanned drawings, and their equipment and connections must first be extracted as graphs [Stürmer et al., 2025]. That difference is one reason to start with a single kind of design, chosen partly for the state of its archive.

	Process module	Mechanical assembly	Electrical package
Design record	P&ID of the module	Native CAD assembly	Single-line diagram
Nodes	Fans, coils, dampers, pumps, valves	Parts: pump, motor, valves, base frame	Buses, breakers, transformers, feeders
Edges	Duct and pipe connections	Mounted on, bolted to, piped to	Electrical connections
Attributes the search compares	Cooling duty, operating conditions, interface standard	Shape of each part, duty, envelope, position of connection points	Voltage, earthing system, switchgear type
The query	The diagram so far: equipment chosen, some connections drawn	A partial CAD model	A preliminary single-line diagram
Engineer checks afterwards	Fire-safety requirements for ducts and dampers	Strength of the base frame for the new equipment	Whether class would approve the protection discrimination
Evidence	Documented for ship piping; proposed for HVAC	Documented for CAD parts; simple assembly test in a preprint; proposed for maritime	Proposed

Table 4: The three examples as graphs: what the nodes and edges are, which checks are recorded as attributes so the search can apply them, and which are left to the engineer.

7 A practical first pilot

A first pilot should cover one kind of design: one the organisation delivers repeatedly, and whose archive can be turned into graphs without a long digitisation effort. It asks a single question. When engineers search with graph retrieval, do they find and reuse a precedent faster than they do today? The pilot is kept to one kind of design so that it can finish with a clear answer. Figure 5 shows its three stages.

7.1 Preparing the archive

Preparation comes down to four decisions.

1. The kind of design. It should be one the organisation delivers repeatedly, with enough past examples for a search to matter.
2. The designs to include. Only designs the organisation is free to reuse should be included. Access rules (customer ownership, non-disclosure, export control) and a few facts about each design's history (delivered, class-approved, known defects) are recorded alongside it, so the search can filter on them.
3. The retrieval unit. Archived items should be parts, assemblies or system modules of the same scope as the query, including when the query is still incomplete (Section 4.9).

4. The graphs. Each design is turned into a graph of the kind shown in Table 4. Where a design exists only as a PDF or a scan, its equipment and connections must be extracted and checked before it can be searched.

Building the graphs is part of the pilot's work, and the archive decides how much of it there is. Native CAD generally needs much less preparation than scanned drawings, which can need substantial digitisation.

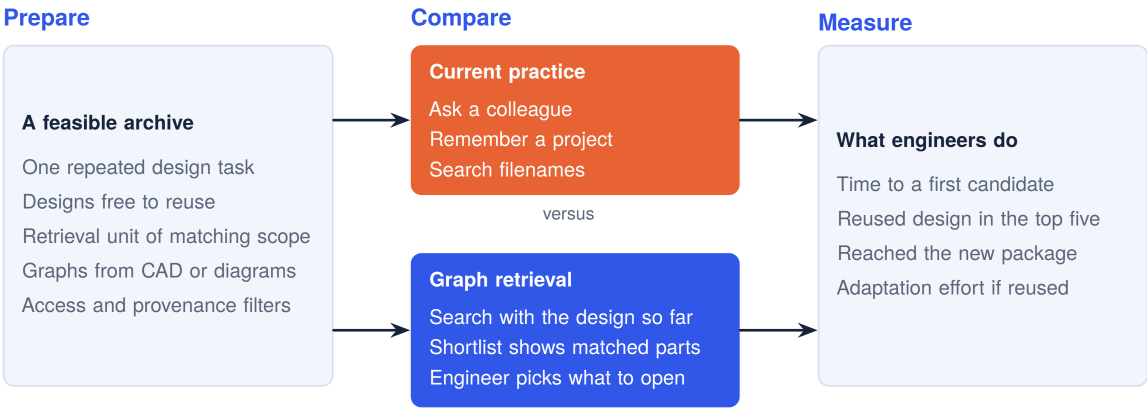


Figure 5: A first retrieval pilot. A feasible archive is searched in two ways; what is measured is whether engineers reuse a precedent faster than they do today.

7.2 Comparing with current practice

The pilot compares graph retrieval with the way engineers find precedents today: asking a colleague, remembering a project, searching filenames. That practice is recorded first, before anyone uses the new search, so the comparison is with what the organisation actually does.

With graph retrieval, the engineer searches with the design as it stands. At first that is a preliminary diagram or a partial CAD model; it becomes more complete as the design goes round the loop in Figure 1. Each candidate on the shortlist shows the components, connections and interfaces that matched the query, next to the archived drawings and models, so the engineer can see why it was ranked.

Each engineer searches both ways, alternating from one task to the next. Some engineers remember more past projects than others, and some tasks are harder than others; alternating within each person evens out both.

7.3 Measuring reuse

The pilot measures four things:

- the time to a first plausible candidate, with each way of searching;

- of the designs engineers went on to reuse, how often graph retrieval had placed one among the first five results;
- the share of searches in which a retrieved design reached the new package; and
- the adaptation and verification effort on the designs that were reused.

To ensure the result can be trusted, a few precautions should be taken. The measures come from what engineers do in their normal work: opening a candidate, exporting it, using it in the new package. A top-five result is reported with the size of the archive, because five from two hundred designs is a different achievement from five from two hundred thousand. Where packages are already known to have been derived from earlier ones, those pairs make a check set: before any engineer sees the tool, the search should find the earlier package from the later one. What result counts as failure, and when the pilot stops, are decided before it starts. Any estimate of hours saved includes the searches that found nothing.

At the end, the organisation has a measured answer for one kind of design, and an archive already prepared and turned into graphs. If engineers found and reused precedents faster, the next step is a second kind of design from Section 6. If they did not, the measures show where the search fell short: in finding the right candidate, in candidates that engineers opened but did not use, or in the effort needed to adapt them.

8 Conclusion

Maritime ETO suppliers may already have many of the design solutions they need. Their archives are hard to search because engineering similarity lies in geometry, connectivity, interfaces and context, while archive search looks at filenames and classifications. A precedent that cannot be found saves no engineering hours, and procurement, fabrication and approval start no earlier.

This white paper proposes searching the designs themselves. Each design is represented as a graph of its components and the connections and interfaces between them, so that a new design can be compared with archived ones by their components and how those components are connected. The engineer searches with the design as it stands, from the first preliminary diagram or partial model, and searches again as it grows. The similarity criterion can be learned largely from the archive itself, so a first system can be tested without first building a large set of hand-labelled similar pairs.

The search finds similar designs; checks and the engineer decide whether one is suitable and worth adapting. Some checks, such as duty or voltage, can be built into the search; others, such as fire safety or class approval, stay with the engineer. The search supports that decision by showing, for each candidate, the components, connections and interfaces that matched, next to the archived files.

The methods have been shown in neighbouring work, on mechanical CAD parts, process plants and ship piping. The studies cited here do not test the complete workflow proposed for maritime ETO archives, with an engineer in the loop, so the next step is a pilot rather than a procurement. It covers one kind of design, chosen partly for the state of its archive, and compares graph retrieval with the way engineers search today. It is judged by what engineers do with the results.

Key takeaway

A pilot succeeds if engineers find and reuse precedents faster than they do today.

Continue the conversation

Connect with me on LinkedIn, or write to daniel@danieljacobsen.com to discuss where these approaches could apply in an engineering organisation. I also advise maritime equipment suppliers on AI more generally.

References

Negar Heidari and Alexandros Iosifidis. Geometric Deep Learning for Computer-Aided Design: A Survey. *IEEE Access*, 13:119305–119334, 2025. ISSN 2169-3536. doi: 10.1109/ACCESS.2025.3587121.

Morten Hertzum and Annelise Mark Pejtersen. The information-seeking practices of engineers: Searching for documents as well as for people. *Information Processing & Management*, 36(5): 761–778, September 2000. ISSN 03064573. doi: 10.1016/S0306-4573(00)00011-X.

Chuck Hsiao, Michael Ruffino, Richard Malak, Irem Y. Tumer, and Toni Doolen. Discovering taxonomic structure in design archives with application to risk-mitigating actions in a large engineering organisation. *Journal of Engineering Design*, 27(1-3):146–169, March 2016. ISSN 0954-4828, 1466-1837. doi: 10.1080/09544828.2015.1126813.

Pradeep Kumar Jayaraman, Aditya Sanghi, Joseph G. Lambourne, Karl D. D. Willis, Thomas Davies, Hooman Shayani, and Nigel Morris. UV-Net: Learning from Boundary Representations. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11698–11707, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-6654-4509-2. doi: 10.1109/CVPR46437.2021.01153.

Vaino Valtteri Karhunen, Tero Sakari Juuti, and Eeva Jarvenpaa. Supporting Design Reuse in Engineer-to-Order Context: A Systematic Literature Review. In *Proceedings of NordDesign 2024*, pages 521–529. The Design Society, 2024. ISBN 978-1-912254-21-7. doi: 10.35199/NORDDESIGN2024.56.

Robert G. Keane, Laurent Deschamps, and Steve Maguire. Reducing Detail Design and Construction Work Content by Cost-Effective Decisions in Early-Stage Naval Ship Design. *Journal* *of Ship Production and Design*, 32(02):110–123, May 2016. ISSN 2158-2866, 2158-2874. doi: 10.5957/jspd.2016.32.2.110.

Min-Chul Kong, Myung-Il Roh, In-Su Han, Seong-Won Choi, and Jeongyeon Kim. A method for transferring piping designs using past project data in shipbuilding. *International Journal of Naval Architecture and Ocean Engineering*, 18:100755, 2026. ISSN 20926782. doi: 10.1016/j.ijnaoe.2026.100755.

Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph Matching Networks for Learning the Similarity of Graph Structured Objects. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3835–3845. PMLR, May 2019.

Martin Løkkegaard, Christian Alexander Bertram, Niels Henrik Mortensen, Lars Hvam, and Anders Haug. Identifying profitable reference architectures in an engineer-to-order context. *International Journal of Production Research*, 61(4):1358–1372, February 2023. ISSN 0020-7543, 1366-588X. doi: 10.1080/00207543.2022.2036850.

Feiwei Qin, Liangzhe Zhu, Zijian Xu, Meie Fang, and Ping Li. CADGCL: Unsupervised retrieval of CAD models via boundary representations. *The Visual Computer*, 41(9):6493–6505, July 2025. ISSN 0178-2789, 1432-2315. doi: 10.1007/s00371-025-03949-y.

Yuhan Quan, Huan Zhao, Jinfeng Yi, and Yuqiang Chen. Self-supervised Graph Neural Network for Mechanical CAD Retrieval, June 2024.

Miia Rantala, Hannu Niemistö, Tommi Karhela, Seppo Sierla, and Valeriy Vyatkin. Applying graph matching techniques to enhance reuse of plant design information. *Computers in Industry*, 107:81–98, May 2019. ISSN 01663615. doi: 10.1016/j.compind.2019.01.005.

Jan Marius Stürmer, Marius Graumann, and Tobias Koch. From Engineering Diagrams to Graphs: Digitizing P&IDs with Transformers. In *2025 IEEE 12th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–11, Birmingham, United Kingdom, October 2025. IEEE. ISBN 979-8-3315-1179-1. doi: 10.1109/DSAA65442.2025.11248012.

Nsikan Udoyen and David W. Rosen. Reusability-based selection of parametric finite element analysis models. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 23 (2):197–214, May 2009. ISSN 0890-0604, 1469-1760. doi: 10.1017/S0890060409000134.