KBE/AI for Ships Construction - A Feasibility Study

M. Tufail Shahzad, Northwestern Polytechnical University, Xi'an/China, ts@mastership.nl
Jacques Hoffmans, MasterShip Software BV, Eindhoven/Netherlands, jh@mastership.nl

Abstract

After about 60 years of CAD in shipbuilding, we are currently entering the 3rd generation of CAD software: The generation that leverages the power of Artificial Intelligence (AI). These topics will strongly increase the future development and the maturity of CAD/CAM. The KBE (Knowledge Based Engineering)/AI (Artificial Intelligence) for Ships project is the development of an intelligent software system which contains a great number of special knowledge and expertise on shipbuilding constructions and assembling. Based on this knowledge and expertise stored in the system, the expert system must apply artificial intelligence and computer technology to simulate the decision-making process of human experts to reason about, judge, and finally solve, complex issues. In the recent times, influence of AI, Cloud Collaboration and VR, has expanded rapidly, where MasterShip's position in Maritime industry can play an important role for these developments. This paper also briefly explains the results of the Innovative approach and detailed technical feasibility performed by MasterShip on different cases.

1. Introduction

Nowadays it is a common problem that the detailed engineering and work preparation for ships is done by inexperienced young engineers in the drawing room where this was done by experienced workers at the shop floor before. At this point, Mastership is ambitious to introduce a virtual system called "Knowledge Based Engineering" for ships (KBE for ships) especially at this stage of the shipbuilding process. This means we want to make virtual use of the practical knowledge of experienced workman and technical constraints during the detailed engineering and the work preparation. Therefore the KBE for Ships system will be a software system that makes decisions by simulating the problem-solving process of human beings. Based on this knowledge, expertise and constraints that will be stored in the system, the expert system must apply artificial intelligence to simulate the decision-making process of human experts to perform reasoning and make judgments, and finally solves complex issues. Design and engineering has always been the interesting and most complex phase for ship construction where to keep the information in order has never be easy. A conventional approach for such phase is presented in Fig.1.

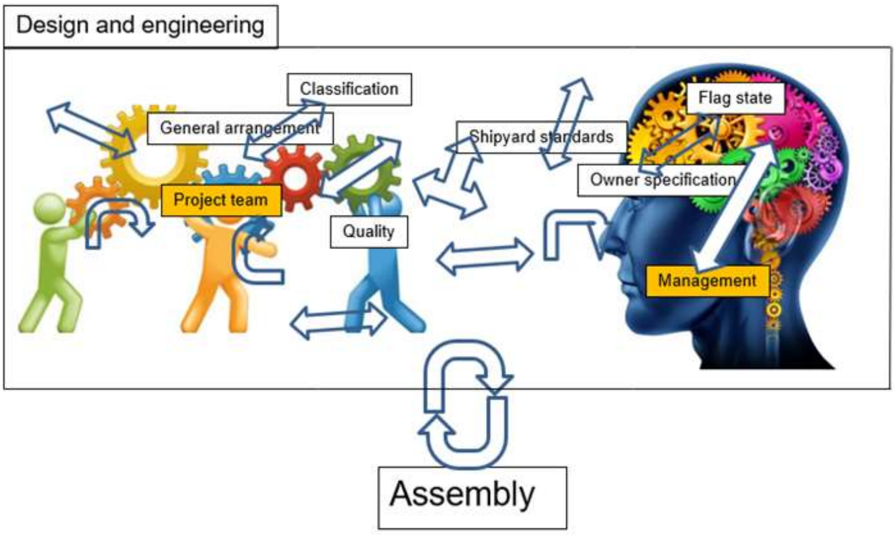


Fig.1: Conventional approach

We, at MasterShip, believe that most of the things done in the design office can be replaced or better managed with a help of knowledge based systems, i.e. expert systems, Fig.2.

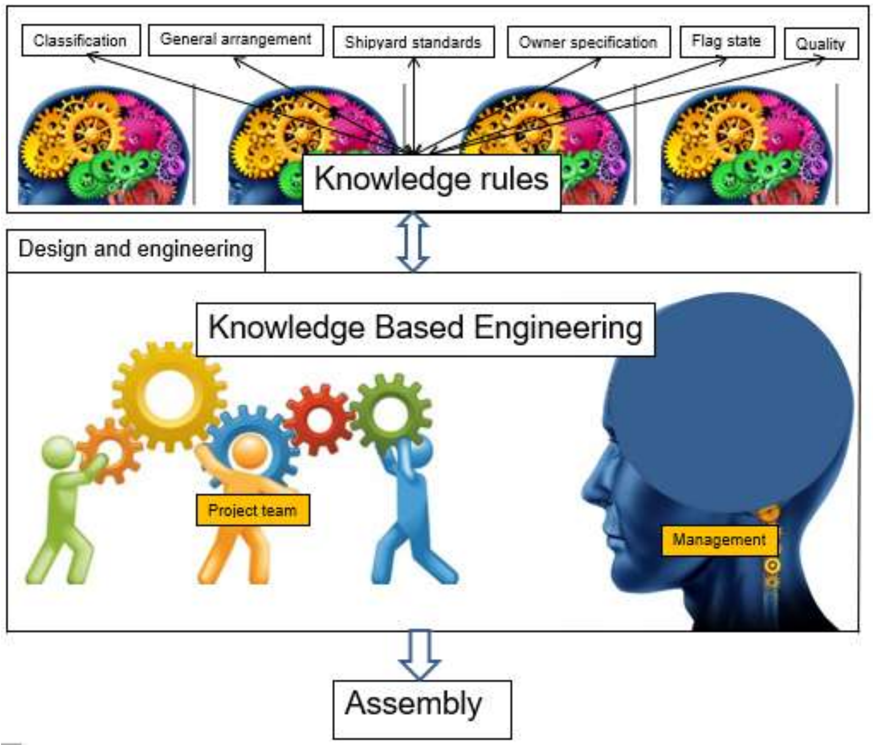


Fig.2: Basic concept of expert system

2. Literature review

Automation in ship design and engineering is introduced since mid-sixties of the 20th century. About 10 years after the first experiments on CAD/CAM in general in the fifties of the 20th century, *Dokken (2012).* Since this introduction of CAD/CAM there were many developments as well as in software as hardware that had great influence on the design and production process of ships. CAD/CAM Was a new industry and today we can conclude that there are still some immature effects. A few remarkable points are:

- Isle of automation: All kind of specific process steps where automated. As well over different involved disciplines as hull, HVAC, electrical, interior and mechanical as over the same discipline but divided over the sequential process steps like concept, basic, detailed and production design.
- Lack of digital communication: Even lack of conventional communication! It appeared to be hard to pass on data to other design and engineering stages.
- Changing processes: The design and building process changed as a consequence on the CAD/CAM development. For instance:
  - 50 years ago, parts of the detailed design and the full production process was done entirely non-digital on the shop floor and lofting floor of the shipyard and is done now fully digital during the last stages of the design and engineering process.
  - Another example is that the design and engineering process stages are growing into each other. The boundaries between the stages concept, basic, detailed and production design the stages are grey and the tasks over these stages are more and more integrated.
  - Simultaneous engineering: work on more disciplines can be performed at the same time and earlier in time than before.
- Earlier need of information and decisions: Because many process stages can be performed earlier and in more detail the information on equipment and details is needed earlier in the

process. Anno 2015 the still appears to be a challenge. It is also noticed that an old rule that was also valid in non-digital design and engineering is still valid: the earlier a mistake is found the cheaper to solve it. Or the other way around: the later a mistake is found the more expensive to solve it. In CAD/CAM it seems obvious that a mistake in the digital phases (concept, basic, detailed and production design) will almost always be cheaper to solve than a mistake at the shop floor.

Above mentioned dots lead to several “Gaps and Overlaps” in the design, engineering and assembly of ships. The costs involved in these Gaps and Overlaps are estimated at 20 % of the total project costs, *Moyst and Das (2005)*.

3. Requirements

Knowledge Based Systems are mostly based on so-called rule-based systems in which heuristic rules are used to encapsulate knowledge from human experts. A knowledge engineer captures these rules during sessions with a human expert, in which think-aloud protocols are used to extract the used knowledge. Subsequently this knowledge is encoded in the form of IF-THEN rules and used by an inference engine, which applies the knowledge to projects in which this knowledge is required.

In some cases, the expert who makes comments while using a CAD system may also propose new rules during the design process and later extend these notes into full rules. Also some systems use machine learning to automatically acquire such knowledge. Because the existing software of MasterShip is based on AutoCAD, this should be target of any possible solution.

So in general, the following functions should be made available to be included in the proposed software:

1. A facility to note and register required KBE facilities, preferably during the design process. This may help in identifying missing functionality. It is mostly used to register where a need exist for expert advice. A first list of known requirements would be a good start.
2. A facility or method to acquire the identified knowledge. This will mostly be based on Knowledge Engineering principles.
3. An open-source inference engine that can be embedded in the AutoCAD environment. We will investigate what libraries are available and how well they fit into the requirements.
4. A facility to encode and test the rules with the proposed system. Because maintenance of a knowledge base can be a daunting task, good verification and testing facilities are a very important requirement. There should be a number of test projects in which all designed rules are tested. There should be a complete regression test environment that allows testing of the entire system and its facilities after every modification.
5. A system to automatically document the acquired rules so the users may get a good idea of what kinds of checks and advice can be expected while using the system.

If possible, the inclusion of an explanation facility should be included to explain to the users why a certain rule is being applied.

4. Scope

MasterShip encountered four main reasons to implement this research in the shipbuilding industry as knowledge based engineering:

1. <u>Integrating general arrangement specification and classification</u> - As well as for the primary as secondary scantling and their belonging details it costs a lot of time to design and engineer according to the applicable classification and/or flag rules. Classification Societies released software these days to find the right dimensions. Integration of the classification/flag rules into a knowledge based database for the CAD/CAM software will be a great step forward.

2. Using practical workshop experience at the drawing office - During the non-digital period when the production design of the ship was done at the shop floor the knowledge of how to build the ship was added by the workers at the shop floor. This added knowledge existed out of general shipbuilding knowledge and specific yard-oriented knowledge. The yard-oriented knowledge was based on the available stock, machinery, working space, cranes, storage space and employees' capabilities at the shipyard. Because this work was done in the real physical stage it was visible for everyone who was around. In the modern design and engineering process the work preparation is done in the virtual stage. Sometimes at the drawing room at the shipyard, sometimes at a design or engineering company elsewhere. Because the last decades CAD/CAM was a way of working that was more popular and only educated to younger people the people who did this modern digital and virtual engineering work did not have a lot of experience yet in general shipbuilding and also not in the yard specific shipbuilding. It was and still is a challenge to bring the shop floor knowledge to the drawing room. A knowledge base database can be of great help here.

3. Becoming a learning organization - Shipyard has to take care for their company knowledge. Many times, the so-called company knowledge is only stored in individual knowledge storage systems of employees. Sometimes it is stored in personal digital files or even in small personal non-digital notebooks. When people leave the company, the knowledge is lost. Companies can grow into learning organizations when they use a system that this knowledge is stored in as well as for storage low accessible as well as for retrievable low accessible knowledge database.

4. Automation of quality control - Quality control takes a large amount of time during the design and engineering process. It is evident that this time must be spent because finding mistakes as early as possible is much cheaper than finding mistakes in a later stage as for example in the worst case at the assembly stage of the ship. It is a challenge to automate a part of the quality control via the use of knowledge base engineering. It will enable to do things right at once instead of repairing afterwards which will definitely be will be a great step in the field of shipbuilding and maritime industry.

5. Feasibility study

In order to validate the concept and innovation proposal by MasterShip, a detail technical feasibility study was performed during the year of 2015-16. This technical report gives an introduction to the concept of Knowledge Based Engineer in product development. It discusses applications of KBE/AI and also describe the process of developing KBE-systems. A brief view of commercial KBE/AI software is given and then benefits and drawbacks with KBE is discussed as well. The report is finished off with some statements about the current research topics of KBE.

Some concrete findings of that report are being mentioned below for a better understanding of the previous research on this regard:

1. As described in the conclusions of the preceding sections, we recommend to first design a number of template interfaces between AutoCAD and a RuleBase system, using a simple sequential Rule Execution mechanism. This system can be built around the ShipHull example and form a Proof-Of-Concept application. It can be used directly for demonstrations or even be used as a small implementation of the KBE.
2. At a later stage, this system can be expanded with more knowledge and AutoCAD features and will then also require more facilities from the Inference Engine. This will form an integrated system, wholly proprietary, eliminating the need for a complicated language interface.
3. During the feasibility research, after completion of multiple test using inhouse inference engine, it is concluded that such system will be able to perform the neural network

characteristic which will lead us to machine learning of such data that is self-generated rules based on the decisions throughout the process.

4. We propose a two-phase approach, where in:
    a. first phase the code that is already available will be expanded to make a working prototype, with a mock-up Inference Engine.
    b. The second phase will expand upon this program and based on several tests with users be developed into a functional Proof-Of-Concept.

This will then form the basis for a later actual implementation, in which the other features will gradually be implemented. This system can grow by adding more problem areas and its associated knowledge. There will be many more AutoCAD interfaces and additional facilities in the Knowledge Base and Inference Engine.

5.1. Direction

There are at least two major parts for the a KEB system, Ship Design Data and a Decision Making algorithm, which may include optimization and machine learning algorithms. Now there are some open sourced tools or algorithms used for machine learning, which may be used in the Expert Systems. The Ship Design Data must be parameterized so that the algorithm can build the connections or rules between those parameters. I think how to parameterize the design data will need a lot of research, for the key parameters must be chosen and feed to the algorithm of machine learning. And it may need tons of data for the machine learning. For the algorithms, we also need to find the best one fit our area:

- Intelligent construction design - This is mainly for stage of ship construction detail design, to combine class-society rules and customized templates in the process of making construction design, with an interactive way. In industry of aviation and automobile, this technology has been widely used for many years.
- Ship hull surface division - That means to combine expert knowledge for the work of making seams/butts in an interactive way. Influence factors of hull thickness, curvature, construction lines arrangement, shipyard-workshop capability, etc. shall be included.
- Welding information add-on In the stage of work preparation, clients will ask for adding welding information in the building kit, such as welding length, material, method and green margin, etc., this kind of information is related to class rules and shipyard-workshop capability.
- Assembly plan - Every shipyard has it unique capability of manufactory and assembly, the best Assembly Plan shall match the feature of the shipyards and optimize the process of cutting, welding, coating, assembly, etc.

5.2. Long-term goal

The long-term goal of AI SH is that companies can become self-learning organizations. Learning by automatically adding new rules to the system. Self-learning by allowing the program to suggest the adaption of new rules. Finally, the goal must be that the AI SH program will not just check on rules but will implement the rules automatically as well. Also, the set-up of a consultancy company for advice on AI in the same domain is a long-term goal.

5.3. Machine Learning in the shipbuilding domain

Almost all rules and interfaces that have been identified so far are logical constructions that do not lend themselves for a machine learning approach. Learning the Rules of a RuleBase would require a large number of examples and associated solutions to allow automatic learning. Such situations may apply to certain other tasks in shipbuilding that were not present in the three examples that we investigated.

6. The rule base architecture

Fig.3 shows a simple flow chart how the rule base is designed in relation with our inference engine. It shows the overall process of how the information will be added and how the deducing of results will lead us to the final conclusions.

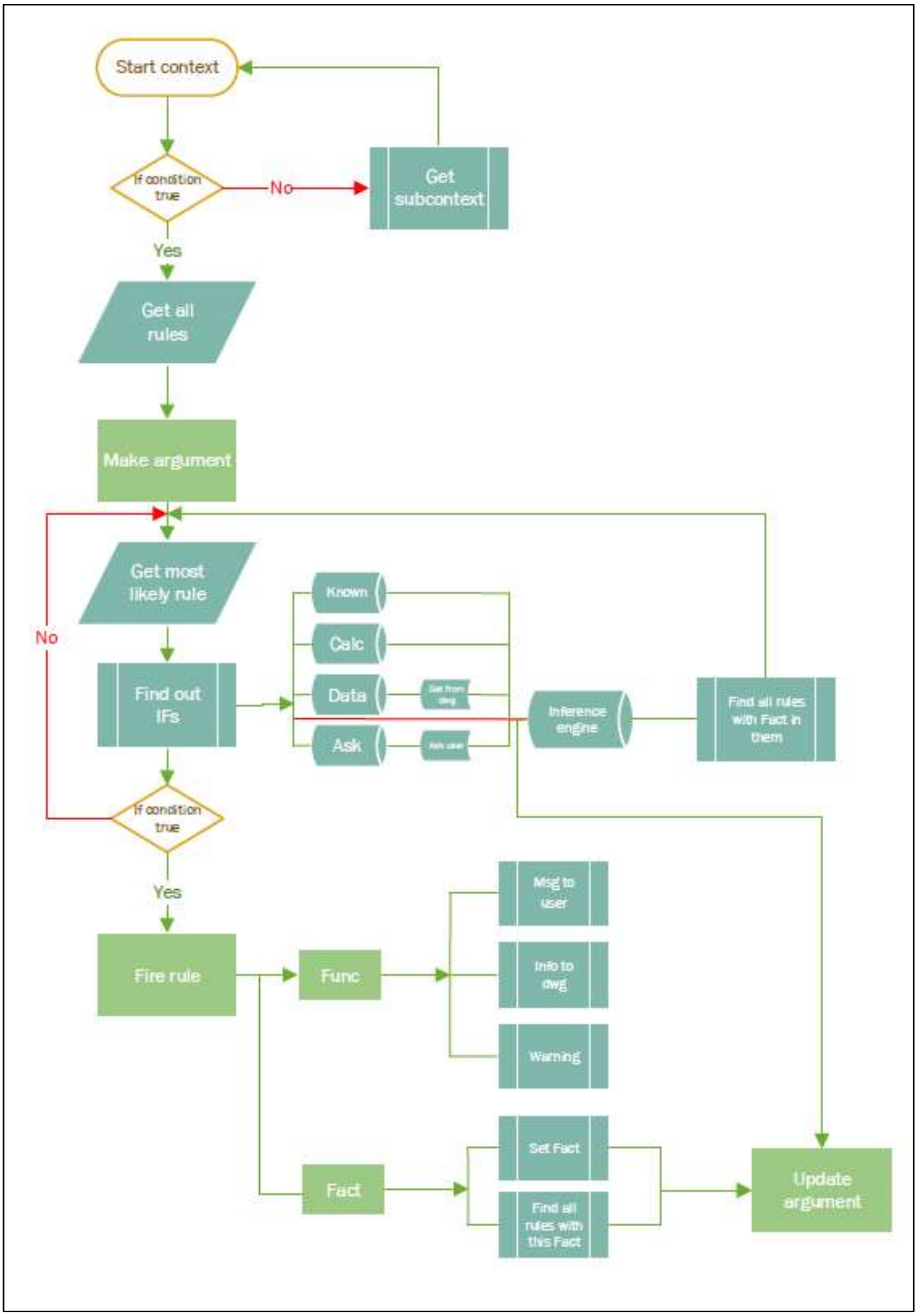


Fig.1: Proposed rule base and inference engine

6.1. The rules

Each rule consists of a number of parts:

```
DefRule R=Plate-MaxSize COMMENT: Check if max plate size is within
Yard limits
        IF: $OR Length > Yard.Plate-Max-Length
              Width > Yard.Plate-Max-Width
              THEN: 100 (Error Size in Plate $Id)
```

Name: Is a unique name that the Rule is referred by.
Comment: Is a short description of what this Rule does. It is used later on when an explanation for a certain decision is requested.
If: Is the collection of Facts that need to be True to make the Rule fire.
Cond: Is the relationship between the Facts of a Rule. This can be an $AND or $OR condition. If more complex situations are needed different Conditions need to be made part of the Inference Engine.
Then: Is the Conclusion of the Rule. In most cases it will conclude another Fact, like the state of a Plate, which indicates if it contains an Error. A conclusion has an associated Certainty Factor.
Facts: Are the Data elements of a Rule and can be part of the Premises (If) or the Conclusions (Then) or both. Since all Facts have an associated Certainty Factor, the Inference Engine needs a facility to combine the certainties of all its Facts.
Funct: The Conclusion of a Rule may also contain several Functions. In the example a text is written to a Log file and also to the Console. There may be a number of Functions on the System Level, like Messages, but also functions, associated with the Context level.

6.2. Certainty factors

To make reasoning with uncertainty possible, the Inference Engine needs some facility to handle this. There are several approaches to this; the most important ones are Bayesian Logic and Certainty Factors. They both combine the probability/certainty of all Facts of a Rule to a total Certainty when the Rule fires. For instance, if one of the Conditions of a rule has a very low certainty this should decrease the overall certainty of the conclusion. On the other hand, if there are several rules, that all can draw conclusions of a certain Fact, then each instance of such a Rule should increase or decrease the overall certainty. It should be some kind of majority ruling, that when two rules both conclude that something is True but another one that it is False, the total should be somewhat True but not False. This process is generally referred to as Truth-Maintenance.

6.3. Functions

In addition to drawing conclusions and determining the certainty of a Fact, Rules may also execute Functions. There can be a Message or a Log entry, but a Rule may also ask Questions or make Suggestions. In addition, a Rule might highlight the Plate on the screen, to indicate that something is wrong. If the system is used as a design assistant, there may be functions to redraw or resize one or more Plates automatically. Functions that otherwise would be done manually by the user could be automated by the rule base.

6.4. Knowledge base and inference engine

Knowledge base is implemented as an in-memory structure of all contexts, rules and concepts as defined in the source code for a certain Rule Base. It is entered with a standard text editor and compiled by Acquaint when loading the Rule Base. errors and warnings are generated for syntax errors, which need to be fixed before running the knowledge base.

7. Development environment

7.1. Development of instantiated Rule Base

Instantiation was originally thought to be realized by having many instances of a single rule. In the new Acquaint system, this has been implemented as Indexed Contexts, which act as a collection of Rules and Concepts, that may be re-used on different sets of input values. So there will always only be one set of active Rules in a given Context, but they can be re-used with different inputs and thus generated different outputs. Each case is handled completely throughout the entire Rule Base and when the final conclusions are drawn, a new case is started. If in the future, we will encounter situations that require several instances, we will set up a new mechanism to deal with that.

For instance, the current implementation will reason about a single plate. As explained earlier a plate is a construct, based on the boundaries of defined hull-lines on four sides. When one plate has been analyzed, the next one will be checked. However, when situations occur, where we need to reason about adjacent plates, there will be a need for two, or more plates. This can be solved initially by defining a number of separate contexts, one for each plate. If however there is a need for a variable or unlimited number of plates to co-exist, true instantiation will be required and we will need to adapt the Inference Engine.

7.2. Inference engine class

The actual Inference Engine and development environment is now based on Acquaint 2020, which is distributed as an executable and the Test Data Source, distributed as a Python program, later to be extended with the AutoCad interface, based on PyAutoCad.

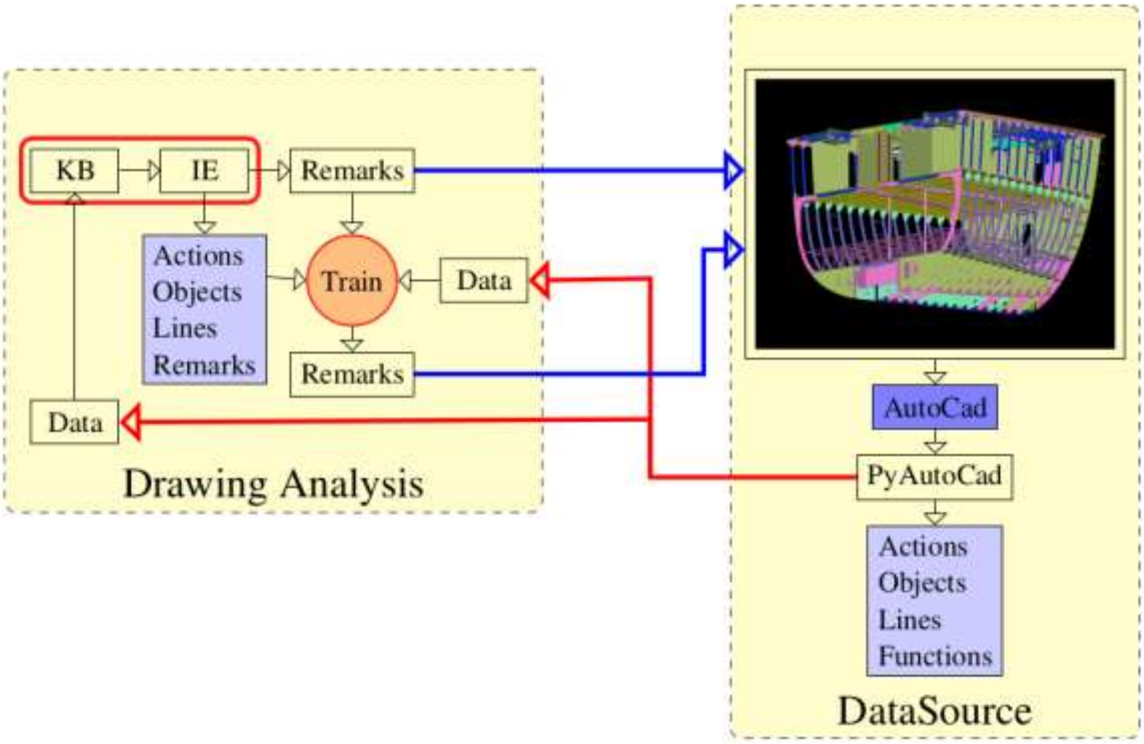


Fig.4: Inference engine class

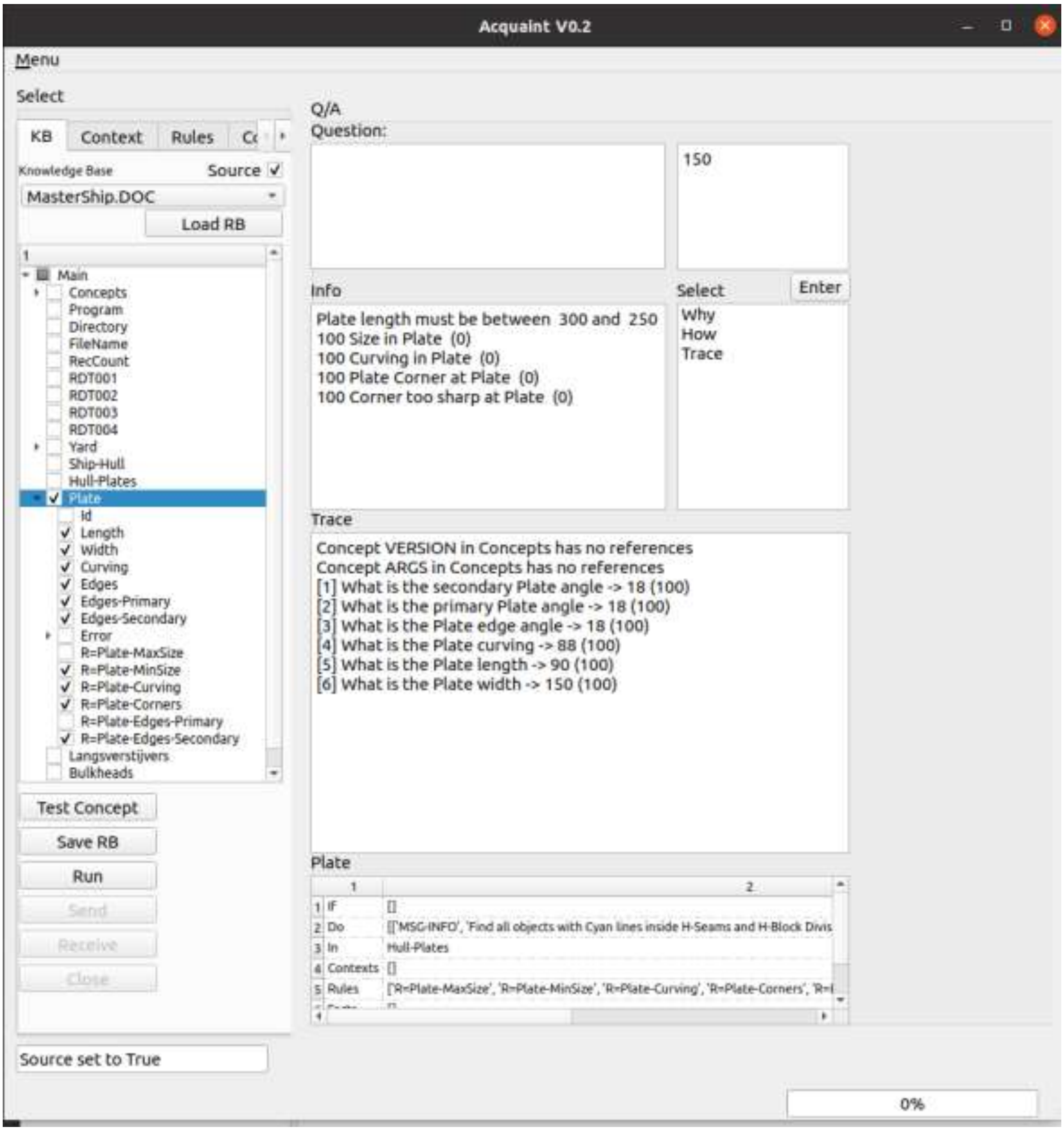
7.3. Rule collection

Three example problems have been defined before and one of them has been implemented as a demo, domain, however, is still rather limited. What it currently does, is checking if a plate fits in the restrictions of a certain Yard. This is an example of relatively simple rules. They could be made more challenging, as can be seen in the Acquaint Demo system Spock, which is a simple Medical Diagnosis demo, dealing with children's diseases. The HullPlates demo could be extended with more complicated rules, if desired.

7.4. Acquaint: In-house testing facility

The development environment consists of a user interface, the compiler and the inference engine. They are provided as a installed python application that connects with the data source, when starting up. The data source can work independently, providing test cases as input, or be connected to autocad, once the pyautocad interface has been implemented.

When the system is disconnected from the data source, all data has to be entered by the user, providing an easy way to test the rulebase during development, as all information to be retrieved from the data source will be asked from the user. Underneath is an example of the user interface, operating without a data source. User face of such facility is presented in Fig.5.



Knowledge Base: MasterShip.DOC

Source ☑

Load RB

1
- ▪ Main
  - ☐ Concepts
  - ☐ Program
  - ☐ Directory
  - ☐ FileName
  - ☐ RecCount
  - ☐ RDT001
  - ☐ RDT002
  - ☐ RDT003
  - ☐ RDT004
  - ☐ Yard
  - ☐ Ship-Hull
  - ☐ Hull-Plates
  - ☑ Plate
    - ☐ Id
    - ☑ Length
    - ☑ Width
    - ☑ Curving
    - ☑ Edges
    - ☑ Edges-Primary
    - ☑ Edges-Secondary
    - ☐ Error
    - ☐ R=Plate-MaxSize
    - ☑ R=Plate-MinSize
    - ☑ R=Plate-Curving
    - ☑ R=Plate-Corners
    - ☐ R=Plate-Edges-Primary
    - ☑ R=Plate-Edges-Secondary
  - ☐ Langsverstijvers
  - ☐ Bulkheads

150

Plate length must be between 300 and 250
100 Size in Plate (0)
100 Curving in Plate (0)
100 Plate Corner at Plate (0)
100 Corner too sharp at Plate (0)

```
Concept VERSION in Concepts has no references
Concept ARGS in Concepts has no references
[1] What is the secondary Plate angle -> 18 (100)
[2] What is the primary Plate angle -> 18 (100)
[3] What is the Plate edge angle -> 18 (100)
[4] What is the Plate curving -> 88 (100)
[5] What is the Plate length -> 90 (100)
[6] What is the Plate width -> 150 (100)
```

	1	2
1	IF	[]
2	Do	[['MSG-INFO', 'Find all objects with Cyan lines inside H-Seams and H-Block Divis
3	In	Hull-Plates
4	Contexts	[]
5	Rules	['R=Plate-MaxSize', 'R=Plate-MinSize', 'R=Plate-Curving', 'R=Plate-Corners', 'R=I

Source set to True

0%

Fig.5: Acquaint user interface

7.5. Why acquaint?

Acquaint comes equipped with a learning facility in the form of Neural Networks. We intend to use this facility in the future as well. In general, learning requires a large number of examples and that is something, that is not available in our problem domain. That is the reason that a Rule Based system is much more appropriate for this problem domain than Machine Learning.

However, when using the system, many examples are gradually being collected. For the Ship-Hull example, for instance, each Plate represents a case, where the RuleBase provides an analysis, but it also offers an opportunity to collect data about each Hull-Plate that is being investigated. The learning interface can collect a number of properties about each hull-plate seen, and associate these with conclusions, drawn from it and feed this information into a Neural Network. That way we could also collect data about other elements, for which there exist no Rules yet. The learning mechanism will be able to collect information in this way and ‘learns’ new rules, that are not based on the If-Then format, but on the association between properties, thus forming a new ‘Neural Rule’, that collects more knowledge as the system is being used.

This learning facility will be part of a future version of Acquaint and it will be to our advantage to think about the possibilities in advance, in order to make it part of the total approach.

8. Conclusion

During this feasibility we have implemented our previous findings in-compliance with our testing facility environment and we succeeded to deduce many interesting facts. Based on these facts we can further expand our project scope area. The main findings of the study are the following:

1. Artificial intelligence in the field of ship construction will not only solve the long-standing issues of mankind in the engineering process but also expand the scope into new heights.
2. From technical perspective, we realized that there are several open-source and commercial Inference Engines available, the ones most useful for this project are written in Java which implements some dialect of Lisp.
3. Using such a system requires the development of a two-way interface between Auto-CAD and the Inference Engine. After trying to use one of the existing interfaces it became clear that the amount of work to make such an interface work seems larger than the effort to build the Inference Engine itself. None of the available interfaces we found, were in use, nor were they maintained and that is not without a reason.
4. The complexity of the Inference Engine for this project is actually not too large, because most of the problems need to be solved in interfaces between the Rule Base and AutoCAD. Certainly in the beginning a simple forward chaining approach is sufficient and we found some open-source projects that could serve a good starting point.

The kind of Rules that would be required for the desired KBE were investigated by building a mock-up Rule Base for three small example projects. In some cases, it proves that a more tightly integration between the required AutoCAD functions and the Rule Base could benefit, which is another reason why a dedicated Inference Engine would be more appropriate

9. Recommendations

Based on these findings, backed by some implementations of parts that seemed most crucial we recommend the following plan of action:

1. Finish the demo project and develop a working Rule Base for the ShipHull example. This requires completing the AutoCAD interfaces.
2. Build a very small first Inference Engine, which does nothing else but execute the Rules sequentially. If forms the basis for further development.

3. Make this first demonstration work completely and use as a basis for the development of a Proof-Of-Concept project.
4. Set up a plan to extend this demonstration into a full POC in which the three main components, the Rule Base, the Inference Engine and the AutoCAD interfaces are implemented and form a basis for further development.

We can overall conclude that at this stage we may further need to inspect the rule base tree by expanding the main items of the test facility. It is unstable and selecting things may crash the system Please note that there is a timeout on answering questions, Currently if you do not answer in time, it crashes which leads us to investigate further to this testing environment to make it stable and effective.

References

CHAPMAN, C.B.; PINFOLD, M. (2001), *The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure,* Warwick manufacturing group, Advanced Technology Center, University of Warwick

DOKKEN, T. (2012), *The History of CAD*, The SAGA Project

MARTIN, J.L. (2009), *Data Modelling, Its Use and Contribution to Ship design, Manufacture, Build and Operations*, ICCAS Conf., Shanghai

MOYST, H.; DAS, M.B. (2005), *Factors Affecting Ship Design and Construction Lead Time and Cost*, J. Ship Production 21/3, pp.186-194