SELF-LEARNING BASED SHIP ONBOARD DECISION SUPPORT SYSTEM

H. Cui, C. Banks, I. Lazakis, O. Turan and **A. Incecik**

Department of Naval Architecture and Marine Engineering, University of Strathclyde, 100 Montrose Street, Glasgow G4 0LZ, UK, o.turan@strath.ac.uk

ABSTRACT

Shipping contributes to about 3.3% of global carbon emissions equating to about 1,000 million tons of C02 being emitted into the atmosphere each year. Action therefore needs to be taken to reduce this amount considerably within the coming years. This can be achieved immediately, cost effectively and efficiently by increasing the energy efficiency through crew's every day operations onboard. An Onboard Decision Support tool to aid crew in making the correct energy efficient decisions can significantly contribute towards reducing emissions. Voyage optimization (route, heading, speed, propeller trim, etc) and maintenance optimization of the main energy consuming systems onboard, are all factors that will be addressed by the proposed Decision Support System framework discussed within this paper. Automatic analytical methods, such as artificial intelligence, are developed for analysis of the historic and real time monitoring of data and ship performance data. Predictive methods are also adopted for forecasting future ship performance. The construction of a unique system framework with an Energy Efficiency Knowledge Bank, which will provide innovative experience sharing based on the analysed data, is presented by utilising a distributed database management system (DDBMS). A numerical optimization is required and the HCPSO and NSGAII optimisation methods are considered for application. The Decision Support takes its basis from an in-house integrated fuzzy decision support method. A few essential attributes and their corresponding importance weightings are used to perform the decision support after the optimization has been carried out; thus providing the crew members with clear and informative suggested best operational (voyage and maintenance) practices.

Keywords: Low Carbon, Energy Efficiency, Onboard, Decision Support, Seafarers

NOMENCLATURE

DDBMS	Distributed Database Management System
FMADM	Fuzzy Multiple Attribute Decision-Making
HCPSO	Hybrid Co-evolution based Particle Swarm Optimization
MFMADM	Multi-agent based Fuzzy Multiple Attribute Decision making
NSGAII	Nondominated sorting genetic algorithm II
SVM	Support Vector Machine

1. INTRODUCTION

Ship onboard decision support systems are one of the important tools in assisting the ship crew to carry out their daily activates. This system can help the ship staff to make important decisions. However, the current systems have many common problems. Current decision support systems are pre-built by software developers who do not have an existing data store nor in-service shipping experience. Furthermore, shipping experience, particularly in the field of energy efficient (Low Carbon) operations is a rapidly developing area of increasing focus. At present it generally falls to the shipping company to update the software with this continually generated in-service experience as current systems as of yet do not have this automated capability. Additionally, the gained experience whilst in service is often hard for crew to formally gather and therefore goes unnoticed. Personal experience is generally applied by individual crew in an ad hoc manner. These are some of the main reasons as to why most decision support systems are not found effective and thus not used for emission reduction benefits. With the new requirements for the protection of the marine environment, low carbon shipping as well as more complex ship operation and onboard equipments and hence the use of decision support systems is in more demand than ever before from shipping companies. If a decision support system was able to update itself, both with the international regulations and ship specific in-service experiences, it would greatly improve the effectiveness, applicability and sustainability of decision support systems.

This paper therefore proposes a new self-learning based decision support system that can automatically update itself according to the day-to-day in-service experience onboard the ship. The work is in progress and the paper presents the framework while the program development is in processing.

Single-objective optimization methods consider the full operation and control of a ship and then provide the optimal operational decision solution to effectively assist the crew in reducing carbon emissions. However, most decisions are Multi-objective and therefore multi optimization methods will provide a group of optimum results where the crew themselves have to select one as the final solution. This selection is often hard for the crew when facing many options and many uncertain events. The final decision will be very dependent on the personal experience and capability of the decision maker (crew member, master) and not necessarily reflect the best energy efficient (Low Carbon) decision for that scenario. This is where decision support is needed to assist the crew in change of operations.

The system proposed in this paper uses learning technology to establish a novel expert system which can make a decision based on previous in-service experience of virtual specialists and technology managers. This will greatly improve the practicability and robustness of the decision-making system in finding the optimum solutions for the complex situations at sea.

In this paper, three machine-learning methods (Decision Tree, Q-learning & Support Vector) are integrated together to provide an effective self-learning system. A fuzzy group decision-making system is then developed to satisfy the requirements of real time decision-making, whilst a multi-agent based system is used to improve system robustness. Section 2 of this paper discusses and provides background to the three machine-learning methods selected for use. Section 3 presents the fuzzy group decision-making methodology and Section 4 provides a detailed introduction the integrated system. Conclusions are given in Section 5 and the future work is described in Section 6.

2. MACHINE LEARNING METHODS TO ACHIEVE A SELF-LEARNING SYSTEM

The self-learning characteristic of the decision support system presented within this paper is based on simulating the learning process of a human. Learning science has been developed since the 1990s (Sawyer, 2006). In this study, the popular learning model of Atkinson and Shiffrin is considered in conjunction with the improved working memory, developed by Baddeley (Baddeley, 1986).

There are three basic parts of memory process in Atkinson and Shiffrin model (as shown in Figure 1).

- The first part is the sensory memory, immediate memory, which can last just several seconds.
- The second part is the long-term memory in which ‘rules’ are remembered.
- The third part is the short-term memory; the most important part for consideration within this paper. The short-term memory selects the appropriate sections in the sensory memory to transfer to the long-term memory and then abandons the other sections.

Figure 2 explains the working memory model of Baddeley (Braddeley, 2000). The memory is processed by a central executive. The multi-components, which include a visuospatial sketchpad, episodic buffer and phonological loop, activate the central executive to help remembering. Thus the working memory is activated via all kinds of perception methods.

The idea is to utilize both of the above human memory models and then adapt the mechanisms into a machinery (computer based) system. However, the mechanisms of the human memory are too complex and the detailed processes are still an unsolved problem. Particularly the short-term memory is still controversial due to its complexity

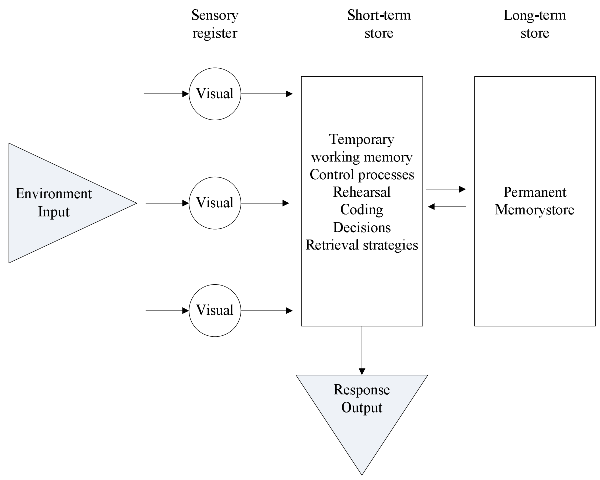

Figure 1: The model of Atkinson and Shiffrin, summarized by Baddeley (Baddeley 1986)

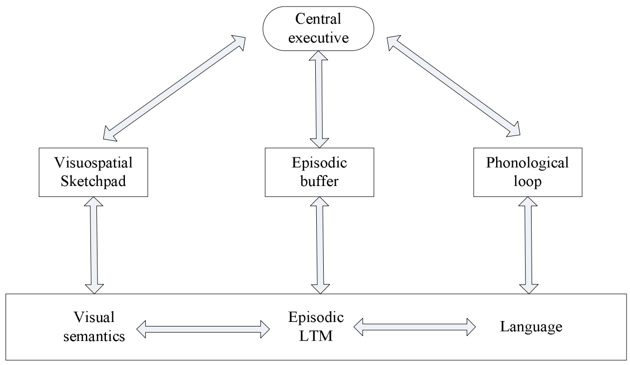

Figure 2: Multi-component working memory model of Baddeley (Baddeley 1986)

A method for modeling the self-learning process of the human mind is therefore needed of which there are several different methods and concepts that are used across many different disciplines; such as engineering, social sciences. Even within engineering alone there are several different understandings and interpretations for the various concepts as the corresponding approaches are decided upon and employed to solve different problems. Within this paper three machine-learning methods are considered and discussed in order to develop an effective self-learning system. Herbert Simon (Simon, 1983) defined machine learning as:

> 'Any changes in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population'.

The concept of self-learning within a decision support system can therefore be considered as when the computer uses a machine learning methods to draw experience from prior actions, to build a database and hence apply the experience to future practical activities directly. There is no single machine-learning method that can simulate the human process of this to the extent that is desired within this paper's objectives. Therefore three methods have been adopted which all have varying strong characteristics and when combined together will form and effective self-learning system. The characteristics and advantages of each of the three methods are discussed along with their role and application within the proposed decision support system.

2.1 DECISION TREE METHOD

The decision tree method was first introduced in 1990 as one of the most popular learning approaches. Decision tree is a tool that utilises a tree-like graph or model to classify instances (scenarios) by sorting them based on feature values (attributes/actions of the scenario). The decision tree is constructed in a top down fashion starting from the root node (the beginning scenario). Each node within the tree represents a different instance (scenario) whilst each branch represents the possible feature values (attributes/actions of the scenario). Each path from the root node, via connected branches and nodes to the selected end node represents a full possible scenario path.

With the development of the decision tree theory different algorithm versions have evolved. Tjen-Sien Lim (Lim and Loh, 2000) made a comparison between decision tree and other learning algorithms and concludes that C4.5 algorithm has a very good combination of error correction and solution convergence speed.

The pseudo code of C4.5 algorithm

1. Check for base cases
2. For each attribute "*a*"
3. Find the normalized information gain from splitting on "*a*"
4. Let "*a_best*" be the attribute with the highest normalized information gain
5. Create a decision node that splits on "*a_best*"

6. Recur on the sub-lists obtained by splitting on "*a_best*" and add those nodes as children of node

There are three main reasons why the decision tree method has been selected as a learning approach in the proposed system.

1. The decision tree method has good ability to operate complex representations whist still being easily explained.

In the decision support system, one important principle is that the analysis process should be powerful and easily understood. The designers not only want to know the calculation results when the system solves a complex problem but also the process to obtain the results. In another word, how to get the results is just as important as the solution itself. Normally, only when the user clearly understand the decision mechanism, they are willing to apply the results obtained in practice. The decision tree method can help the decision makers understand the process of analysis better and hence the solution is much more likely to be used.

2. The decision tree has the ability to treat the discrete data.

In most ship design optimisations, the objectives and limitations are discrete. The learning approach needs the ability to process discrete data together with the ability to deal with continuous data.

3. The fast solution convergence and data catogorisation speed can reduce the computational running time, especially for mass data.

The speed classification of decision tree is relatively fast comparative to other approaches of data mining and thus provides a successfully solution for time consuming problems.

2.2 Q-LEARNING METHOD

Real-time learning is very important to the proposed system and the Q-learning method is employed to realize this function within this paper. The Q-learning method has been developed very quickly in recent years and has extensive application in engineering, business, management *etc*.

Q-learning (Watkins 1989) (Watkins and Dayan 1992) belongs to the reinforcement learning area. The principal behind this method is that it works by learning an action-value function (attributes/actions of the scenario) that gives the expected utility (scenario) given the actions taken, storing this as a fixed policy thereafter and hence applying the action-value function again to the same/similar scenarios.(Cui 2010)

An important advantage of Q-learning is that it is able to compare the expected utility (scenario) of the available actions without requiring a model of the environment. Therefore it can be described as a form of model-free reinforcement learning. Despite the requirements of Q-learning being relatively independent of the environment, it does not mean Q-learning is applicable for all situation (Watkins and Dayan, 1992).

With the development of Q-learning theory, the research in continuous mathematical modeling has made progress but the discrete and finite mathematical environment still remain the main applications for Q-learning. For this reason a discrete environment with finite step change is a better foundation for Q-learning and this well suited to the application of ship onboard decision support system.

The theory of Q-learning is simple and clear but its mathematical proof is complex and involves many disciplines. In order to give a systematic and general understanding, a brief conceptual framework combined with mathematical discussion is presented in this section.

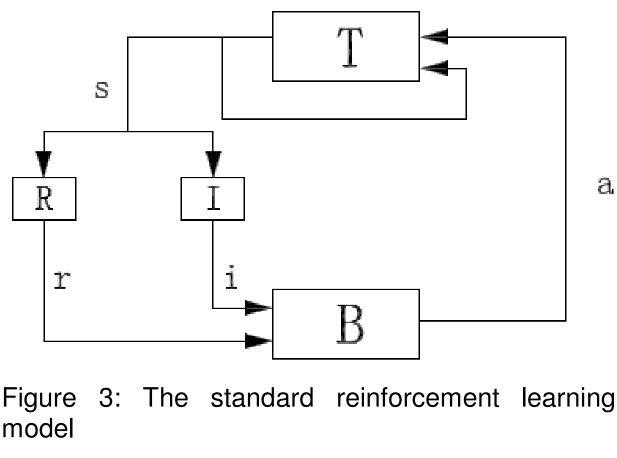

Figure 3: The standard reinforcement learning model

From the model of Figure 3, it can be seen that there are four basic factors of reinforcement learning:

• The policy factor

The mapping from environment to action. This factor is the core of reinforcement learning and there are many ways to implement this policy, for example as a lookup table, functions or an artificial neural network.

• The reward function

This is the goal of reinforcement learning. The aim of the agent in the reinforcement learning is to maximize the total reward in the lifetime run.

• The value of function

A formula to synthesise all the rewards that an agent expects to get over the future.

• Optionally, a model of the environment.

In a typical reinforcement learning model, an agent is connected to the environment via perception and action. The model shown in Figure 4, illustrates the process where B is an agent and T is the environment.

- In the first step, agent B receives an input i;
- In the second step, agent B chooses an action 'a' to generate an output.

This action 'a' changes the environment T

- In the third step, the value of this state transition is communicated to the agent B through a scalar reinforcement signal, 'r'.

The agent's behavior, B, should choose actions that tend to increase the long-run sum of the values of reinforcement signal. This can be learnt over time by systematic trial and error, guided by a wide variety of algorithms.

Q-learning is an excellent approach for reinforcement learning to assist the optimisation algorithm in finding the hidden relationships between the scenarios and actions. In practice, it is computationally impossible to find all the necessary integrals (changes in scenario due to actions dependent on the environment) without additional knowledge or some modification. Q-learning solves this problem by taking the maximum value (biggest change) over a set of integrals. Rather than finding a mapping (value iteration) from state to state (scenario to scenario), Q-learning finds a mapping from state/action pairs (actions that result in particular scenarios). These resultant values are called Q-values.

Q-learning therefore makes use of Q-functions where the function is used to determine which previously defined Q-value to perform given the state, and hence following the given policy thereafter.

The definition of an optimal Q-value is the sum of the reinforcements received when performing the associated action and then following the optimal policy thereafter. Equation (1) is a general expression.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (1)$$

According to Equation (1), Q-learning differs from other value iteration reinforcement learning in that it displays the relationship of given actions and expected values of the successor states. It does not require that each action is performed in a given state and the expected values of the successor states are calculated.

Considering the Q-learning running process, before learning has started, *Q* returns a fixed value, chosen by the designer. Then, each time the agent is given a reward (the state has changed). In the following iteration, the agent is given a reward (the state has changed) every time and *Q* value will be changed according to this reward. New values are calculated for each combination of states from the statement set S and action a from the action set A . It assumes the old value and makes a correction based on the new information as shown in Equation (1).

Equation (1) is equivalent to:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t)(1-\alpha) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a)] \quad (2)$$

Equation (7) is the formula used in this study.

A disadvantage of Q-learning is that it is an unsupervised learning method: the system cannot be taught whether an action it performs is good or bad; i.e. there are no 'teachers'. However, integration with a mathematical method called dynamical programming can solve this problem.

Dynamic programming involves just two basic principles.

- "Firstly, if an action causes something bad to happen, then the system learns not to do that action in that situation again.
- The second principle is that, if all the actions in a certain situation lead to bad results, then that situation should be avoided.

For the dynamic programming, the primary objective of learning is to find the correct mapping from states to state values (scenario to scenario). In other words, the dynamic programming tries to find the relationship between the states and the expression values of state (attributes/actions of the scenario). Q-learning is developed from the theory of dynamic programming and therefore combining these two methods provides an output reinforcement learning technique that can do any number of tasks.

2.3 SUPPORT VECTOR MACHINE METHOD

The most numerical issues in the proposed system require a learning method which can make an accurate prediction according to small sample. At the same time, the learning method should have the excellent ability to control the complex learning environment whilst providing a clear explanation to the users. The Support Vector Machine (SVM) is a relatively new developing machine learning method and it provides a powerful machine learning approach to deal with above learning problems during the decision making stage of the proposed system.

“The Support Vector Machine (SVM) is a supervised learning method that generates input-output mapping functions from a set of labeled training data. The mapping function can be either a classification function, i.e., the category of the input data, or a regression function”, (Wang 2005). Generally, the ‘machine’ in SVM is not a real machine. In machine learning, an algorithm is always called machine, so SVM continues using this custom and ‘machine’ here means algorithm. The word ‘support vector’ comes from the training samples in SVM which are expressed via vectors and SVM strongly focuses on the vectors at the edges which support the seeking of the hyper plane. Usually, a support vector machine is constructed by a hyper-plane or set of hyper-planes in a high or infinite dimensional space. The hyper-plane, which has the largest distance to the nearest training data points of any class, can make a good separation of data.

3. DECISION MAKING METHOD

The decision-making method accepted here is developed from the fuzzy multiple attribute decision-making (FMADM) method, which can solve both linguistic and numerical attributes. (shown in Figure 4). Ölçer (Ölçer 2001) reviewed and analyzed the most of the known FMADM methods according to their group decision-making abilities. Based on this research, they provide a new FMADM approach that can be utilised in ship design, for example, propulsion/maneuvering system selection or subdivision optimization. Cui and Turan (Cui 2010) improved on this to form a multi-agent based system. In this section, the self-learning (discussed in section 2) is added into the decision making method to improve the ability for complex and uncertain onboard applications.

A fuzzy multiple attribute decision-making (FMADM) method, which can solve both linguistic and numerical attribute, is introduced here to solve the decision-making problem. The linguistic attribute is one of the most difficult aspects in decision-making. In the proposed method, the specialists (e.g. shipping company performance expert) on different topic areas are required to evaluate the decisions. The specialists form the specialists committee and this committee decides the quality of design, which solely depends on individual’s knowledge and experience level. Then the technology manager, allocates the weightings of the specialists’ decision to rank the possible decisions in order of importance; to hence help the human make the best decision. How this process can be simulated within a computational system now must be considered. The study proposes a new learning based virtual specialists committee which can use prior experience to evaluate the solutions. This virtual specialist is built via a software agent and it can obtain and update the knowledge automatically. The relevant virtual technology manager will also be created via a software agent and will then allocate the weighting to every member of the committee (shown in Figure 5)

For better application in ship decision support system, an agent-based framework is utilised to realize this method in computer environment. The method is rebuilt according to module based design principle, which makes the system satisfy the change of designer’s requirement and the expansion of the detail.

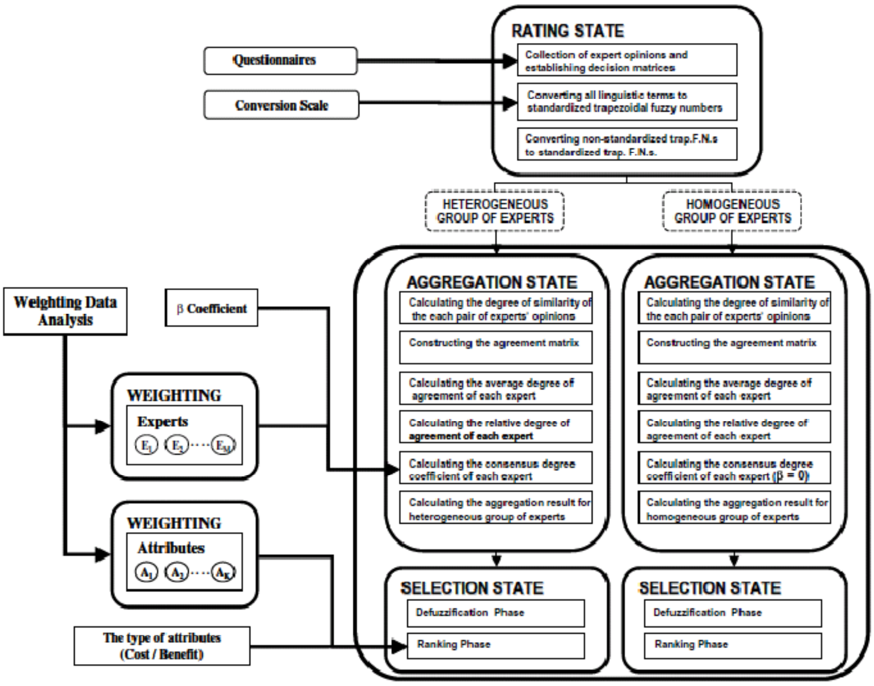

Figure 4: Work flow of FMADM used in this study (Olcer 2011)

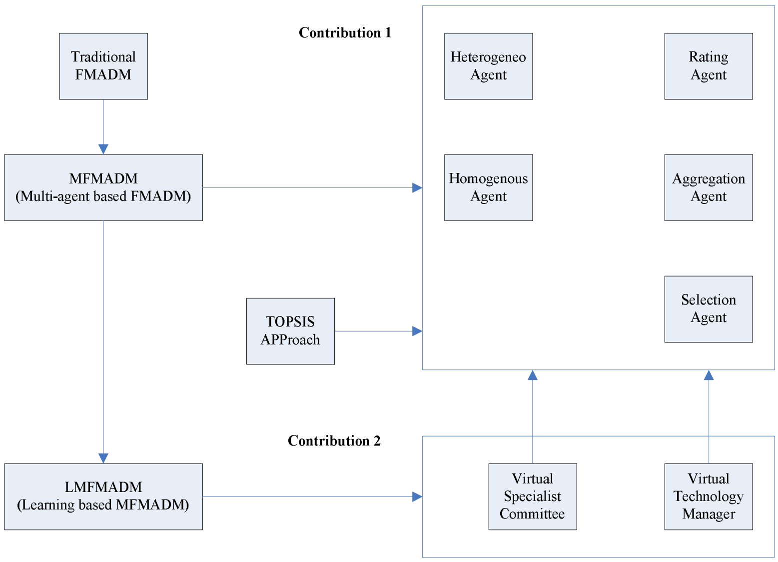

Figure 5: Work flow of MFMADM used in this study.

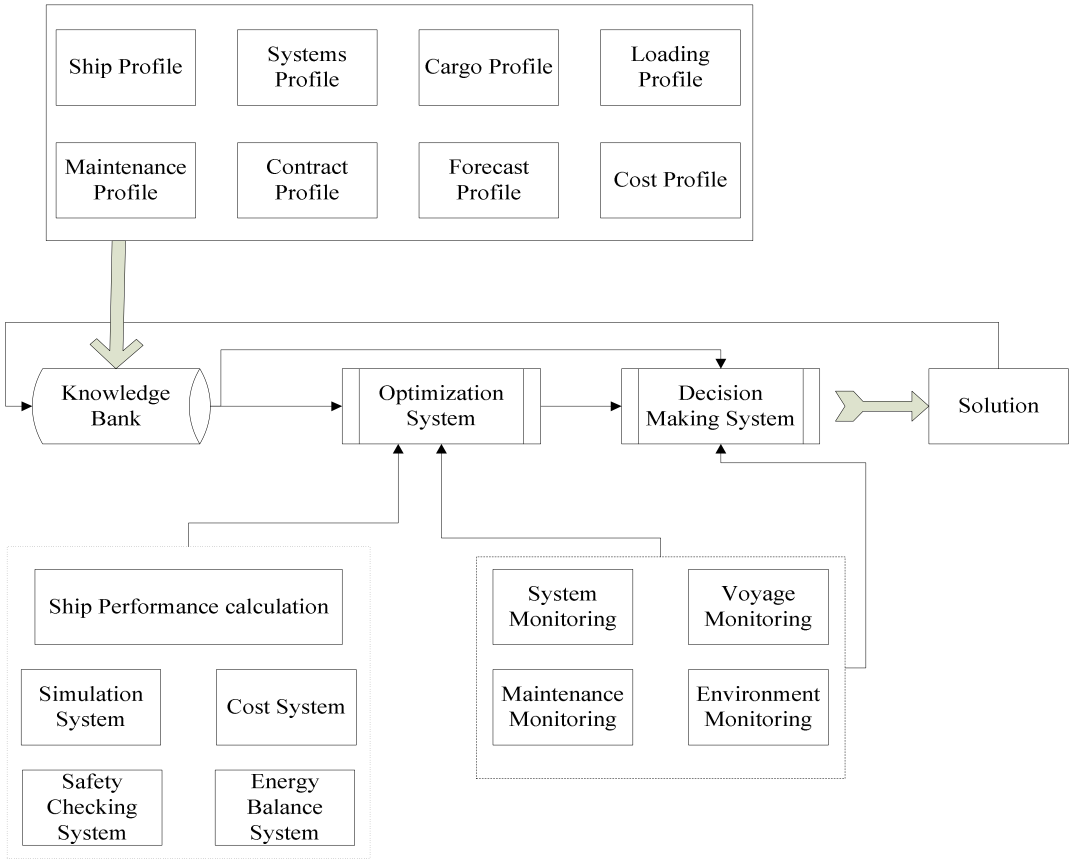

Figure 6: Structure of proposed system

4. A NEW INTEGRATED SELF-LEARNING BASED REAL TIME DECISION MAKING METHOD

The new self-learning based ship onboard decision making system is an integrated system including the discussed machine learning methods, decision making method, optimization methods and a data base management system etc. Figure 6 presents the structure of proposed system.
In Figure 6, the knowledge bank (a database where the experience/knowledge data can be stored for future use) supports the whole system as the system database. This knowledge bank draws the information from eight parts of a particular ship together with the previous experience from learning process and rules from both international organizations and company. The eight parts are shown in Figure 6 sketch out the full situation of the ship. It is noteworthy that updating of knowledge bank is dynamic process which means it will be continually renewed. The optimization method will collect the information from five simulating and calculating system with the assistance from four monitoring system. Two optimization methods, HCPSO (Cui, Ölçer et al. 2009) and NSGAII, are employed to process the multi-objective optimization and both of these two methods are improved with learning functions proposed by Cui and Turan (Cui 2010). Figure 7 provides the basic workflow of two optimization methods. (Cui and Turan 2010)

After the optimization, the system will provide a group of results that are the same ‘good’ for the multi-objective optimization method. So the decision-making system will use prior experience to select one from these results as the final solution. The process of decision making is the process of self-learning and the system will apply the learning independent of human interaction.

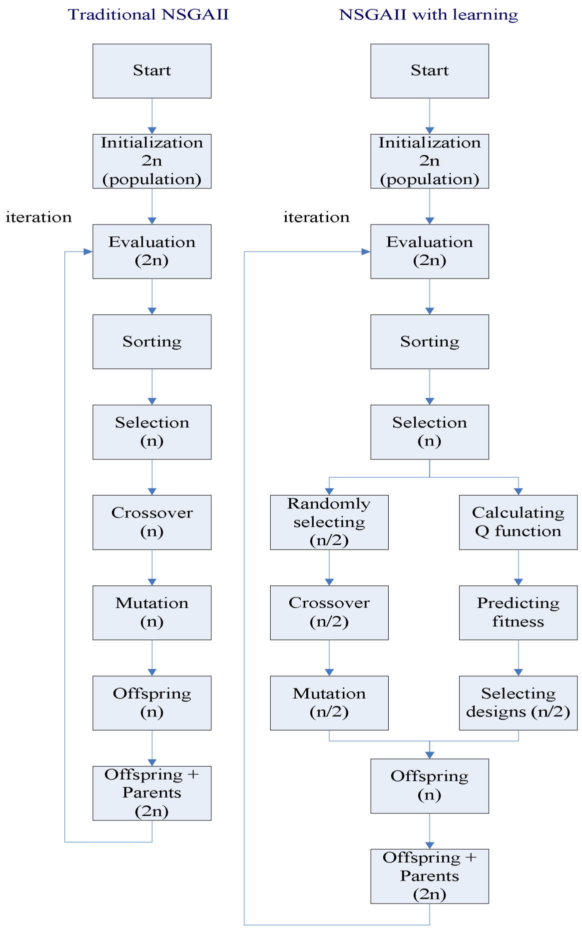

Figure 7-1: Learning based NSGAII

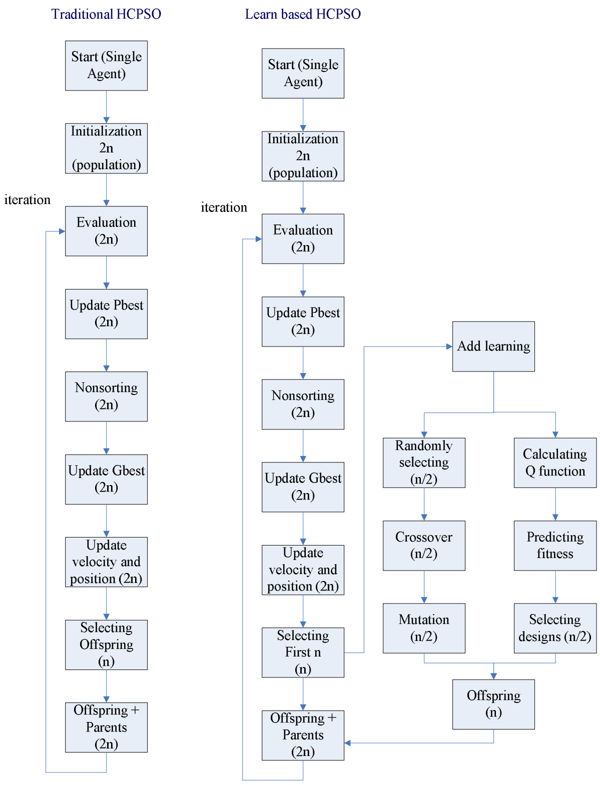

Figure 7-2: Learning based HCPSO

The self-learning decision support system is operated by three parts and divided into two running stages. Figure 8 provides an overview of the operation of this support system. The three parts of this system include the software developer, shipping company and ship fleet. The software developer provides the software and general rules for decision-making including the rules and regulations from IMO, flag nations and classification society etc. The software developer will be responsible for updating new learning methods and decision-making methods together with the new up-coming rules and regulations. Meanwhile, they will also assist the shipping company in ‘training’ the decision-making system. The shipping company will provide the company rules and general experience.

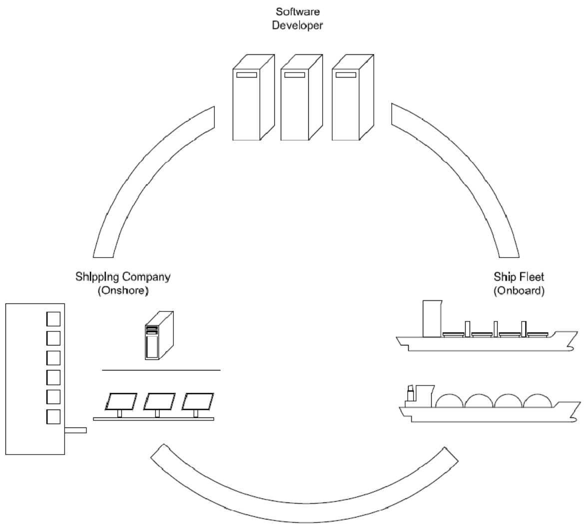

Figure 8: Overview of the operation of this support system

Here, general experience means the experience of running previous shipping cases and this can be provided by the experts within the company. With experience and over time the ship fleet will use the proposed system and give the feed back to the shipping company based on gained experience on how to improve the performance.

In this system, self-learning is the core part. Figure 9 provides the brief learning process of the system. From Figure 9, it can be seen that the whole learning process is divided into two running stages: training and running. The data in Figure 9 is composed by both the previous cases and the current case. First of all, the system will use prior experience to train. This will require the company to organize the specialists of every area to make a full evaluation of the previous cases. These cases will be refined according to two different types of attributes: linguistic and numerical attributes. The linguistic attributes will be operated by decision tree method and the numerical attributes will be analyzed by the support vector machine method. These will form the virtual specialists and technology manager, which will assist the decision making in future.

There are three parts of knowledge. The first one is the general rules and regulations from software developer predetermined. The second one is the experience from learning system when the third one is the inside rules and regulations from the shipping company.

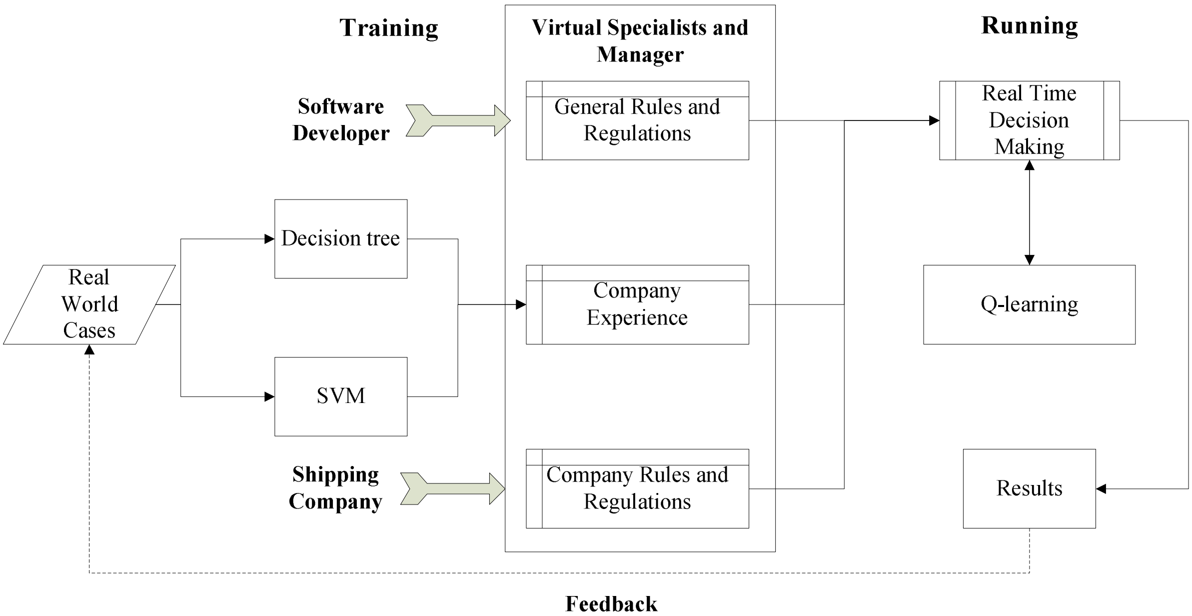

Figure 9: Brief learning processes

In the running stage the third machine learning method, Q-learning, will assist the decision making in processing real-time learning. The result of every decision-making will be sent to the knowledge bank. From the knowledge bank the data can be used for new training cases to improve and revise the virtual specialists and technology manager. This is therefore an iteration improving process.

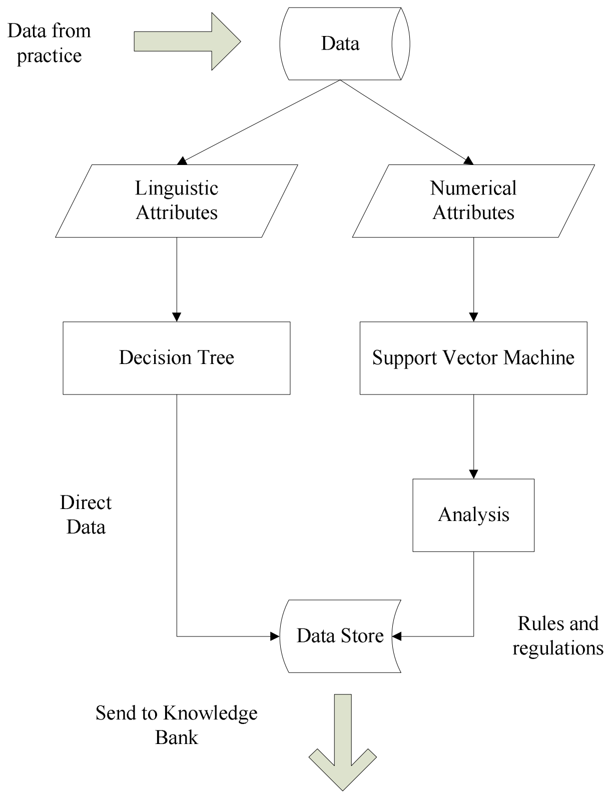

Figure 10: Data Flow of Learning System

Figure 10 introduces the data flow of proposed system. The practical data will be divided into two types and finally, the knowledge bank will store two types of data: original data and analyzed rules.

Figure 11 shows the workflow of new self-learning based decision making method. The virtual specialist and technology manager who are formed by learning methods replace the human roles in this new system. The whole method is reformed according to multi-agent based system and consists of nine parts. These nine parts can be updated or changed independently.

When the optimization stage has been finished, the results are sent to the decision making stage. The interface agent will accept these results and reform their format to different attributes. Meanwhile, the interface agent will automatically draw the relative experience from knowledge bank and information from monitoring system. The ship performance calculation results will be sent to the interface agent accompanying with the optimization results. The interface agent will synthesize all of this information and send them to other agents. The rating agent is to integrate fuzzy data into standardized positive trapezoidal fuzzy numbers and establish the decision matrix. This agent has a strong connection of virtual specialist agent. At the beginning, the virtual specialist agent will make an evaluation on the optimization results (as decision making candidates) sent by the interface agent and will sent the evaluation values to the rating agent to transfer. The aim of aggregation agent is to combine the opinion of single or multidiscipline specialist to form a group consensus opinion. There is a close linkage between this agent and virtual technology manager. Heterogeneous and Homogeneous agent are to give the result of the fuzzy opinions and needs to measure the degree of similarity between trapezoidal fuzzy numbers. Selection agent will select the best solutions according to the suggestions of specialists. It will employ a ranking approach to give the rank of suggestions and this approach is TOPSIS Approach.

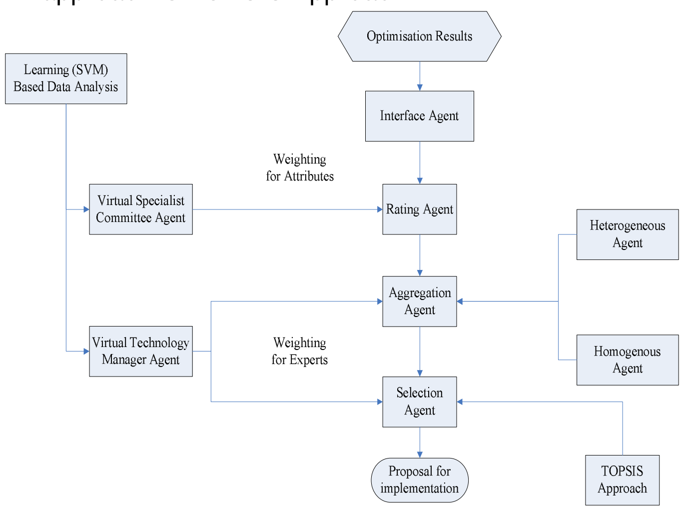

Figure 11: Workflow of new self-learning based decision making method

Here virtual specialist and technology manager is dynamic improving with the adding of cases via knowledge bank. But they have been fixed before the application which means the experience of them is fixed before the next running. If the situation is changed, those virtual specialist and technology manager should have the ability to make a reaction on current statement. For example, in the voyage planning, when the weather conditions suddenly change, re-optimization will take plenty of time. In this situation, if the proposed system can not only begin the re-optimising but also use the current optimization result to make a decision taking account of the factor of weather changing, it will greatly improve the efficiency. These require the ability of real time learning in dynamic environment. Q-learning as a real time learning tool is introduced into this system to realize this task.

When the situation changes, which is watched by the monitoring system, the decision-making method will automatically collect the changed conditions and related attributes. Then the virtual specialists and technology manager will re-evaluate these attributes again. The Q-learning method is used in this time. The system will use the related attributes of prior cases to set the marks. Then the Q-learning method will learn them and give the closest assessment considering other attributes. The decision-making system will use these new evaluating values to make a decision until the new optimization finished. Figure 12 shows real time learning in decision-making process.

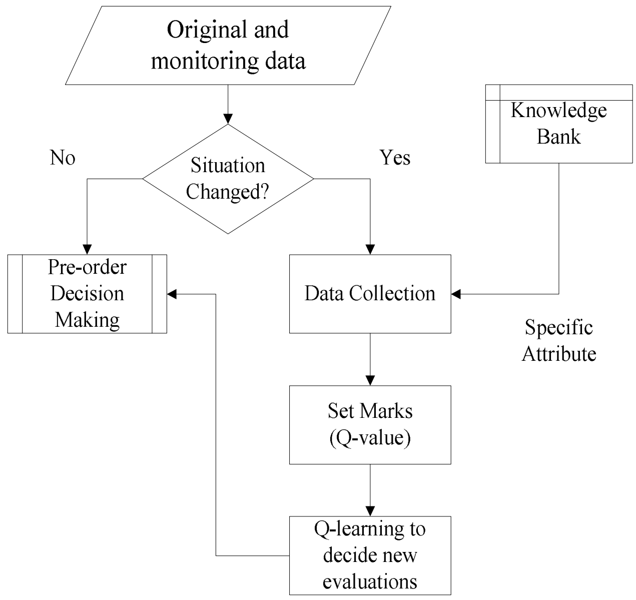

Figure 12: Real time learning in decision-making process

The steps of self-learning based ship onboard decision support system

Training part (I)

The main aim of the train part is to build knowledge bank.

- I-1: Input the general and company rules and regulations
- I-2: Collect previous cases and divide it into two types of attributes: linguistic and numerical
- I-3: The linguistic attributes will be stored directly
- I-4: The numerical attributes will be analyzed by SVM method to find relationships
- I-5: Integrate the knowledge to form the virtual specialist and technology manager

Running part (II)

- II-1: Obtain the ship performance and monitoring information
- II-2: Set optimization objectives
- II-3: Take knowledge from the knowledge Bank as constraints and parameters of algorithm.
- II-4: Process the optimization and use Q-learning to reduce the time of optimization
- II-5: Obtain the optimization results and remove off the unfeasible ones
- II-6: Employ the decision-making method
- II-7: Get the virtual specialist and manger from knowledge bank
- II-8: Check whether the situation has been changed
- II-9: If not, use the decision making method directly
- II-10: If changed, draw more information from knowledge bank to assist the virtual specialist and manger to make a re-evaluation via Q-learning
- II-11: Obtain the final solution
- II-12: Revise this solution and send it to the training part

5. CONCLUDING REMARKS

The proposed ship onboard decision making support system can greatly improve the ability of controlling the situation. It can assist the crew in improving the energy utilization ratio and reducing emissions. The real time learning method in this paper can self-adapt to the complex sea states giving the best operational (voyage and maintenance) decisions. From using the system, the crew can easily realize low carbon targets.

In the proposed system, the Q-learning approach is to be applied for the first time to the real time ship design decision support system. This new method improves the ability to deal with complex situations. This study also presents the examples of combining the Q-learning with NSGAII and HCPSO. This study also successfully improves the traditional decision making method. It solves the automatic updating and the algorithm, which no longer depends on manual calculation, is hence calculated automatically. The Support Vector Machine is successfully imported into the decision-making method in order to solve the issue of a lack/unavailability of experts.

6. FUTURE WORK

The future work of this ship onboard decision support system will focus on three aspects. Firstly, the accuracy of learning to different cases should form a further study. Especially, the linguistic attribute operations need further study for finding a more effective approach for better classification.

In this study, the method is employed to avoid useless treatments for the linguistic and leaves it to operate until the new case needs this information. If a more effective method can be found to treat the linguistic before it is used; this will further reduce the system's run time. At the same time, the automatic calibration of proposed system is also necessary.

The current learning is purely from prior experience and approved rules but does not consider the operating crew factor. How to improve the accuracy and make it comfortable for crew is an important aspect.

Real time learning needs a further study to improve the efficiency of the algorithm. Q-learning approach in this study uses the look-up table that is effective for finite and discrete environment. When a more complex design environment is developed, a more effective approach should be utilized.

Lastly, the conflict solving is an important research field for decision support system. This is also the important and very popular research area of multi-agent system. In this study, the conflict solving is simplified according to specific decision-making situation. However for more complex decision statements and society learning, an advance method needs to be developed and applied to this system. Currently, the research of society learning is in the initial stages and when new approaches become mature or developed, further study should be carried out to improve the performance of the system.

REFERENCES

Baddeley, A. D. (1986). 'Working memory', Oxford University Press.

Baddeley, A. D. (2000). 'The episodic buffer: A new component of working memory?', Trends in Cognitive Science, Vol.4, pp. 417-423.

Cui, h., A. I. Ölçer, et al. (2009). 'An Improved PSO Approach in a Multi Criteria Decision Making Environment', Ship Technology Research, Vol. 56 No 1, pp.13-22.

Cui, h. (2010). 'Learning based multi-agent conceptual ship design decision support system', Naval Architecture and Marine Engineering, Glasgow, University of Glasgow and Strathclyde, PhD thesis.

Cui, h. and O. Turan (2010). 'Application of a new multi-agent based particle swarm optimisation methodology in ship design', Journal Computer-Aided Design, Vol. 42, No 11, pp. 1013-1027.

Lim, T.-S. and W.-Y. Loh (2000). 'A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms', Machine Learning, Vol. 40, No 3, pp. 203-229.

Ölçer, A. I. (2001). 'Development of a new fuzzy multiple attribute decision making approach and its application to decision making in ship design and shipbuilding', Institute of Science and Technology, Istanbul, Turkey, Istanbul Technical University, PhD theis.

Quinlan, J. R. (1993). 'C4.5: programs for machine learning', San Francisco, CA, USA Morgan Kaufmann Publishers Inc.

Sawyer, R. K., Ed. (2006). 'The Cambridge Handbook of the Learning Sciences', Cambridge University Press.

Simon, A. H. (1983). 'Search and reasoning in problem solving', Artificial Intelligence, Vol. 21, No 1-2, pp. 9-29.

Wang, L., Ed. (2005). 'Support Vector Machines: Theory and Applications', Studies in fuzziness and soft computing, Springer-Verlag GmbH.

Watkins, C. (1989). 'Learning from delayed rewards', University of Cambridge.

Watkins, C. and P. Dayan (1992). 'Q-learning.' Machine Learning, Vol. 8, No 3-4, pp. 279–292