From Knowledge-Based to Skill-Based Systems: Sailing as a Machine Learning Challenge

Pieter Adriaans

FNWI / ILLC
University of Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
The Netherlands
pietera@science.uva.nl
http://turing.wins.uva.nl/~pietera/ALS/

Abstract. This paper describes the Robosail project. It started in 1997 with the aim to build a self-learning auto pilot for a single handed sailing yacht. The goal was to make an adaptive system that would help a single handed sailor to go faster on average in a race. Presently, after five years of development and a number of sea trials, we have a commercial system available (www.robosail.com). It is a hybrid system using agent technology, machine learning, data mining and rule-based reasoning. Apart from describing the system we try to generalize our findings, and argue that sailing is an interesting paradigm for a class of hybrid systems that one could call Skill-based Systems.

1 Introduction

Sailing is a difficult sport that requires a lot of training and expert knowledge [1],[9],[6]. Recently the co-operation of crews on a boat has been studied in the domain of cognitive psychology [4]. In this paper we describe the Robosail system that aims at the development of self-learning steering systems for racing yachts [8]. We defend the view that this task is an example of what one could call skill-based systems. The connection between verbal reports of experts performing a certain task and the implementation of ML for those task is an interesting emerging research domain [3],[2], [7]. The system was tested in several real-life race events and is currently commercially available.

2 The Task

Modern single-handed sailing started its history with the organization of the first Observer Single-Handed Transatlantic Race (OSTAR) in 1960. Since that time the sport has known a tremendous development and is the source of many innovations in sailing. A single-handed skipper can only attend the helm for about 20% of his time. The rest is divided between boat-handling, navigation,

N. Lavrač et al. (Eds.): ECML 2003, LNAI 2837, pp. 1–8, 2003.

preparing meals, doing repairs and sleeping. All single-handed races allow the skippers to use some kind of autopilot. In its simplest form such an autopilot is attached to a flux-gate compass and it can only maintain a compass course. More sophisticated autopilots use a variety of sensors (wind, heel, global positioning system etc.) to steer the boat optimally. In all races the use of engines to propel the boat and of electrical winches to operate the sails is forbidden. All boat-handling except steering is to be done with manual power only.

It is clear that a single-handed sailor will be less efficient than a full crew. Given the fact that a single-handed yacht operates on autopilot for more than 80 % of the time a slightly more efficient autopilot would already make a yacht more competitive. In a transatlantic crossing a skipper will alter course maybe once or twice a day based on meteorological data and information and from various other sources like the positions of the competitors. From an economic point of view the automatization of this task has no top priority. It is the optimization of the handling of the helm from second to second that offers the biggest opportunity for improvement. The task to be optimized is then: *steer the ship as fast as possible in a certain direction and give the skipper optimal support in terms of advice on boat-handling, early warnings, alerts etc.*

3 Introduction

Our initial approach to the limited task of maintaining the course of a vessel was to conceive it as a *pure machine learning task*. At any given moment the boat would be in a certain region of a complex state-space defined by the array of sensor inputs. There was a limited set of actions defined in terms of a force exercised on the rudder, and there was a reward defined in terms of the overall speed of the boat. Fairly soon it became clear that it was not possible to solve the problem in terms of simple optimization of a system in a state-space:

- There is no neutral theory-free description of the system. A sailing yacht is a system that exists on the border between two media with strong non-linear behavior, wind and water. The interaction between these media and the boat should ideally be modelled in terms of complex differential equations. A finite set of sensors will never be able to give enough information to analyze the system in all of its relevant aspects. A careful selection of sensors given economical, energy management and other practical constraint is necessary. In order to make this selection one needs a theory about what to measure.
- Furthermore, given the complexity of the mathematical description, there is no guarantee that the system will know regions of relative stability in which it can be controlled efficiently. The only indication we have that efficient control is possible is the fact that human experts do the task well, and the best guess as to select which sensors is the informal judgement of experts on the sort of information they need to perform the task. The array of sensors that 'describes' the system is in essence already anthropomorphic.
- Establishing the correct granularity of the measurements is a problem. Wind and wave information typically comes with the frequency of at least 10 hz.

But hidden in these signals are other concepts that exist only on a different timescale eg. gusts (above 10 seconds), veering (10 minutes) and sea-state (hours). A careful analysis of sensor information involved in sailing shows that sensors and the concepts that can be measured with them cluster in different time-frames (hundreds of seconds, minutes, hours). This is a strong indication for a modular architecture. The fact that at each level decisions of a different nature have to be taken strongly suggest an architecture that consists of a hierarchy of agents that operate in different time-frames: lower agents have a higher measurement granularity, higher agents a lower one.

- Even when a careful selection of sensors is made and an adequate agent-architecture is in place the convergence of the learning algorithms is a problem. Tabula rasa learning is in the context of sailing impossible. One has to start with a rough rule-based system that operates the boat reasonably well and use ML techniques to optimize the system.

In the end we developed a hybrid agent based system. It merges traditional AI techniques like rule based reasoning with more recent methods developed in the ML community. Essential for this kind of systems is the link between expert concepts that have a fuzzy nature and learning algorithms. A simple example of an expert rule in the Robosail system is: *If you sail close-hauled then luff in a gust*. This rule contains the concepts ‘close-hauled’, ‘gust’ and ‘luff’. The system contains agents that represent these concepts:

- Course agent: If the apparent wind angle is between A and B then you sail close hauled
- Gust agent: If the average apparent wind increases by a factor D more than E seconds then there is a gust
- Luff agent: Steer Z degrees windward.

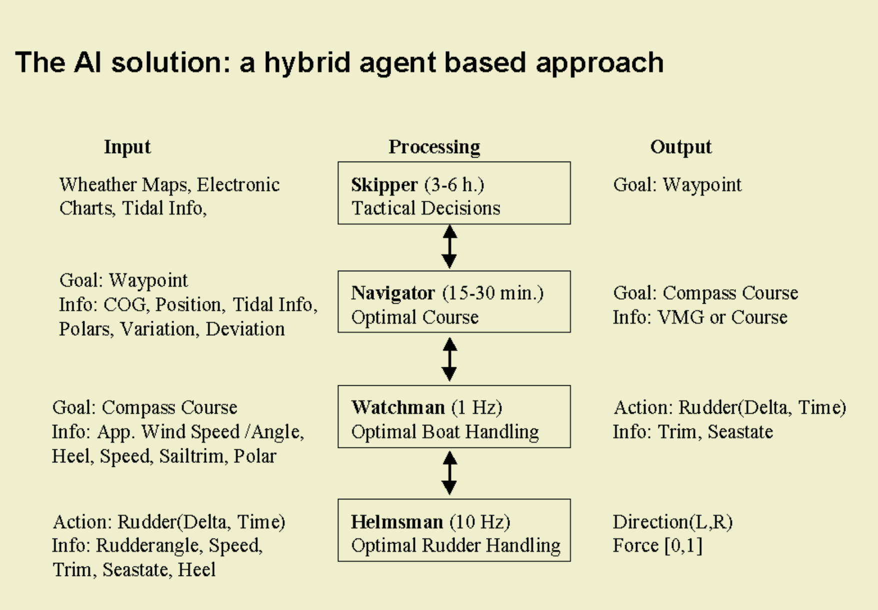
The related learning methodology then is:

- Task: Learn optimal values for A,B,C,D,E,Z
- Start with expert estimates then
- Optimize using ML techniques

This form of *symbol grounding* is an emerging area of research interest that seems to be of vital importance to the kind of skill-based systems like Robosail [8], [2], [7], [5].

4 The System

The main systems contains four agents: Skipper, Navigator, Watchman and Helmsman. These roles are more or less modelled after the task division on a modern racing yacht [9]. Each agent lives in a different time frame. the agents are ordered in a subsumption hierarchy. The skipper is intended to take strategical decisions with a time interval of say 3 to 6 hours. He has to take into account weather patterns, currents, seastate, chart info etc. Currently this process is only partly automated. It results in the determination of a waypoint, i.e. a location on the map where we want to arrive as soon as possible. The navigator and the watchman have the responsibility to get to the waypoint. The navigator deals with the more tactic aspects of this process. He knows the so-called polar diagrams of the boat and its behavior in various sea states. He also has a number of agents at his disposal that help him to asses the state of the ship: do we carry too much sail, is there too much current, is our trim correct etc. The reasoning of the navigator results in a compass course. This course could change within minutes. The watchman is responsible for keeping this course with optimal velocity in the constantly changing environment (waves, wind shifts etc.). He gives commands to the helmsman, whose only responsibility it is to make and execute plans to get and keep the rudder in certain positions in time.



The AI solution: a hybrid agent based approach

Input	Processing	Output
Wheather Maps, Electronic Charts, Tidal Info,	**Skipper** (3-6 h.) Tactical Decisions	Goal: Waypoint
Goal: Waypoint Info: COG, Position, Tidal Info, Polars, Variation, Deviation	**Navigator** (15-30 min.) Optimal Course	Goal: Compass Course Info: VMG or Course
Goal: Compass Course Info: App. Wind Speed /Angle, Heel, Speed, Sailtrim, Polar	**Watchman** (1 Hz) Optimal Boat Handling	Action: Rudder(Delta, Time) Info: Trim, Seastate
Action: Rudder(Delta, Time) Info: Rudderangle, Speed, Trim, Seastate, Heel	**Helmsman** (10 Hz) Optimal Rudder Handling	Direction(L,R) Force [0,1]

Fig. 1. The hierarchy of main agents

There are a number of core variables: log speed, apparent wind speed and angle, rudder angle, compass course, current position, course on ground and speed on ground. These are loaded into the kernel system. Apart from these core variables there are a number of other sensors that give information. Amongst others: canting angle mast, swivel angle keel, heel sideways, heel fore-aft, depth, sea state, wave direction, acceleration in various directions. Others will activate agents that warn for certain undesirable situations (i.e. depth, temperature of the water). Others are for the moment only used for human inspection (i.e. radar images). For each sensor we have to consider the balance between the contribution to speed and safety of the boat and the negative aspects like energy consumption, weight, increased complexity of the system.

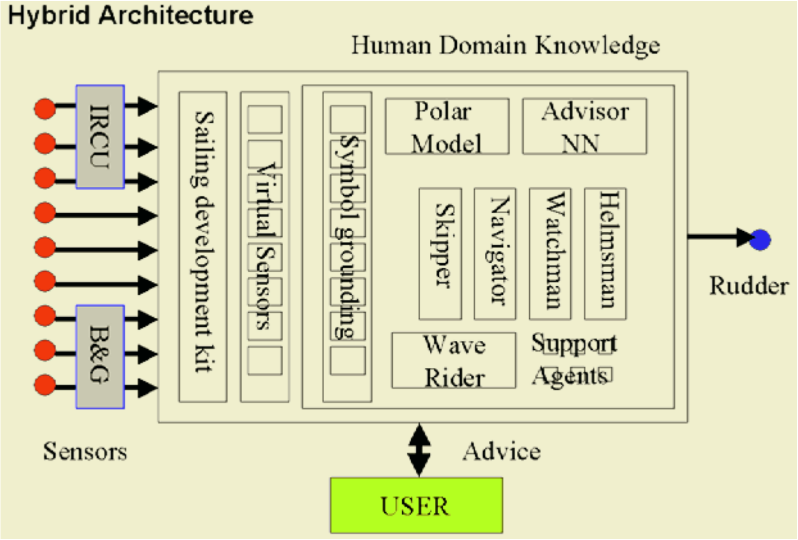


Fig. 2. The main architecture

The final system is a complex interplay between sensor-, agent- and network technology, machine learning and AI techniques brought together in a hybrid architecture. The hardware (CE radiation level requirements, water and shock proof, easy to mount and maintain) consists of:

- CAN bus architecture: Guaranteed delivery
- Odys Intelligent rudder control unit (IRCU): 20 kHz, max. 100 Amp (Extensive functions for self-diagnosis)
- Thetys Solid state digital motion sensor and compass
- Multifunction display
- Standard third party sensors with NMEA interface (e.g. B&G)

The software functionality involves:

- Agent based architecture
- Subsumption architecture
- Model builder: on line visual programming
- Real Time flow charting
- Relational database with third party datamining facility
- Web enabling
- Remote control and reporting

Machine Learning and AI techniques that are used:

- Watch man: Case Based Reasoning
- Helmsman: neural network on top of PID controller
- Advisor: nearest-neighbor search

- Agents and virtual sensors for symbol grounding
- Data-explorer with machine learning suite
- Waverider: 30 dimensional ARMA model
- Off line KDD effort: rule induction on the basis of fuzzy expert concepts

Several protypes of the Robsosail system have been tested over the years: the first version in the Single Handed Transatalantic in 2000, a second prototype was evaluated on board the Kingfisher during a trip from Brazil to the UK. A final evaluation was done on board of the Syllogic Sailing Lab during the Dual Round Britain and Ireland in 2002. In 2003 the first commercial version is available.

5 Lessons Learned

The Robosail application is a hybrid system that can be placed somewhere between pure rule-based systems and pure machine learning systems. The nature of these systems raises some interesting philosophical issues concerning the nature of rules and their linguistic representations. In the course of history people have *discovered* that certain systems can be built and controlled, without really understanding why this is the case. A sailing boat is such a system. It is what it is because of ill-understood hydro- and aerodynamical principles and has a certain form because the human body has to interact with it. It is thoroughly an anthropomorphic machine. Human beings can handle these systems, because they are the result of a long evolutionary process. Their senses are adapted to those regions of reality that are relatively stable and are sensitive to exactly those phase changes that give relevant information about the state of the systems. In a process of co-evolution the language to communicate about these concepts emerged. Specific concepts like 'wave', 'gust' and 'veering' exist because they mark relevant changes of the system. Their cognitive status however is complex, and it appears to be non-trivial to develop automated systems that discover these concepts on the basis of sensor data.

A deeper discussion of these issues would have to incorporate an analysis of the nature of rules that is beyond the scope of this paper. The rules of a game like chess exist independently of their verbal representation. We use the verbal representation to communicate with others about the game and to train young players. A useful distinction is the one between *constitutive rules* and *regulative rules.* The constitutive rules define the game. If they are broken the game stops. An example for chess would be: *You may not move a piece to a square already occupied by one of your own pieces.* Regulative rules define good strategies for the game. If you break them you diminish your chances of winning, but the game does not stop. An example of a regulative rule for chess would be: *When you are considering giving up some of your pieces for some of your opponent's, you should think about the values of the men, and not just how many each player possesses.* Regulative rules represent the experience of expert players. They have a certain fuzzyness and it is difficult to implement them in pure knowledge-based systems. The only way we can communicate about skills is in terms of regulative rules. The rule *If you sail clause hauled then luff in gust* is an example. Verbal reports of experts in terms of regulative rules can play an important role in the design of systems. From a formal point of view they reduce the complexity of the task. They tell us *where to look* in the state space of the system. From a cognitive point of view they play a similar role in teaching skills. They tell the student roughly what to do. The fine tuning of the skill is then a matter of training.

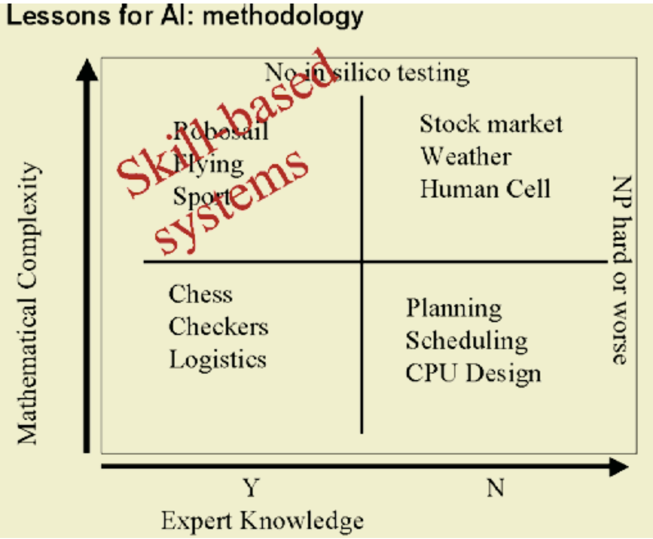


Fig. 3. A taxonomy of systems

This discussion suggests that we can classify tasks in two dimensions: 1) The expert dimension: Do human agents perform well on the task and can they report verbally on their actions and 2) The formal dimension: do we have adequate formal models of the task that allow us to perform tests in silico? For chess and a number of other tasks that were analyzed in the early stages of AI research the answer to both questions is yes. Operations research studies systems for which the first answer is no and the second answer is yes. For sailing the answer to the first question is positive, the answer to the second question negative. This is typical for skill-based systems. This situation has a number of interesting methodological consequences: we need to incorporate the knowledge of human experts into our system, but this knowledge in itself is fundamentally incomplete and needs to be embedded in an adaptive environment. Naturally this leads to issues concerning symbol grounding, modelling human judgements, hybrid architectures and many other fundamental questions relevant for the construction of ML applications in this domain.

A simple sketch of a methodology to develop skill-based systems would be:

- Select sensor type and range based on expert input
- Develop partial model based on expert terminology

- Create agents that emulate expert judgements
- Refine model using machine learning techniques
- Evaluate model with expert

6 Conclusion and Further Research

In this paper we have sketched our experiences creating an integrated system for steering a sailing yacht. The value of such practical projects can hardly be overestimated. Building real life systems is 80% engineering and 20% science. One of the insights we developed is the notion of the existence of a special class of skill-based systems. Issues in constructing these systems are: the need for a hybrid architecture, the interplay between discursive rules (expert system, rule induction)and senso-motoric skills (pid-controllers, neural networks), a learning approach, agent technology, the importance of semantics and symbol grounding and the importance of jargon. The nature of skill-based systems raises interesting philosophical issues concerning the nature of rules and their verbal representations.

In the near future we intend to develop more advanced systems. The current autopilot is optimized to sail as fast as possible from A to B. A next generation would also address tactical and strategic tasks, tactical: win the race (modelling your opponents), strategic: bring the crew safely to the other side of the ocean. Other interesting ambitions are: the construction of better autopilots for multihulls, the design an ultra-safe autonomous cruising yacht, establish an official speed record for autonomous sailing yachts and deploy the Robosail technology in other areas like the Automotive industry and aviation industry.

References

1. Frank Bethwaite. *High Performance Sailing.* Thomas Reed Publications, 1996.
2. Xia Chris Harris, Hong and Qiang Gan. *Adaptive Modelling, Estimation and Fusion from Data.* Springer, 2002.
3. Ross Garret. *The Symmetry of sailing.* Sheridan House, 1987.
4. Edwin Hutchins. *Cognition in the wild.* MIT Press, 1995.
5. N.K. Poulsen M. Nørgaard, O. Ravn and L.K. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems.* Springer, 2000.
6. C.A. Marchaj. *Sail Performance, theory and practice.* Adlard Coles Nautical, 1996.
7. Tano Shun'Ichi Takeshi Furuhashi and Hans-Arno Jacobson. *Deep Fusion of Computational and Symbolic Processing.* Physica Verlag, 2001.
8. Martijn van Aartrijk, Claudio Tagliola, and Pieter Adriaans. Ai on the ocean: The robosail project. In Frank van Harmelen (Editor), editor, *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 653–657. IOS Press, 2002.
9. Cornelis van Rietschoten and Barry Pickthall. *Blue Water Racing.* Dodd, Mead & Company, 1985.