In: Proceedings of the European Safety and Reliability Conference, ESREL '97, June 17-20, 1997, Lisbon, Portugal, Page 2301-2308.

FAILURE DIAGNOSIS AND ANALYSIS FOR AN AUTONOMOUS UNDERWATER VEHICLE

Palle Christensen [1], Kurt Lauridsen [1], Henrik Østergaard Madsen [2]

[1] Risø National Laboratory, System Analysis Dept., DK-4000 Roskilde, Denmark.
[2] MARIDAN Autonomous Underwater Vehicles ApS, DK-2970 Hørsholm., Denmark

ABSTRACT

This paper describes a functional model of an autonomous mini submarine for the inspection of oil- and gas pipelines. Setting up this model was part of the process of creating a mission management system for the submarine. The vehicle control system will contain parts responsible for the manoeuvring of the vehicle, diagnosis, obstacle avoidance and countermeasures to possible malfunctions of systems and components. Preliminary to the design of the control system, an analysis of the functions to be performed by the vehicle during a survey was carried out. The Goal Tree - Success Tree method was used as the primary tool in this analysis. The resulting functional model identified the soft- and hardware necessary for the operation of the vehicle. Furthermore, the model forms the design basis for an automatic diagnosis system for identifying causes of failures during a mission.

KEYWORDS

Autonomous underwater vehicle (AUV), Failure diagnosis, Functional modelling, GTST method, Mission management system, MPLD method.

INTRODUCTION

MARTIN is an unmanned submarine for autonomous inspection of oil- and gas pipelines and electrical cables. It is being developed in a Danish research project by Maridan ApS, the Danish Technical University (DTU), Risø National Laboratory, and the companies Reson A/S, SEAS A/S, and Em. Svitzer A/S. Autonomous operation of the vehicle offers a potential for economical benefits compared to the use of tethered vehicles or divers because of less dependence on weather and less demands to the sophistication of the mother ship. Therefore, it will be of great interest to operators of submarine pipelines and cables which need periodical inspection in order to detect position, damage, corrosion or changes in the surrounding sea bed. Also other types of inspection, e.g. for measurement of environmental factors, can be done using MARTIN.

However, in order to be of interest to potential users the vehicle must be able to perform its autonomous missions reliably and to react correctly to foreseen - and unforeseen - disturbances, such as obstacles not indicated on the charts. Therefore, a large effort has been put into designing the control system, starting with a functional analysis of the tasks to be performed by the vehicle during a mission. The analysis comprised all aspects of a mission, including the identification of potential external hazards which have to be catered for, e.g. various kinds of obstacles in the water or at the sea bed.

THE VEHICLE

The MARTIN vehicle has a 4.5 meter long flatfish shaped, low-drag hull with two main thrusters aft (Bjerrum *et al.*, 1994). It has three vertical and one horizontal thruster for low speed, and fins, rudders and an elevator for normal speed manoeuvring. The hull is open, and all electronics is placed in watertight compartments. Propulsion energy comes from 5 kWh lead-acid batteries. The maximum speed is 2.5 m/s, the range capacity 80 kilometres, and the maximum depth 100 meters. It was based on the MARIUS vehicle developed under the European MAST programme (Egeskov *et al.*, 1994). Existing projects will add a 3 kW Stirling engine to extend the range to 500 km, and the depth rating will be upgraded to 2000 meters in future versions.

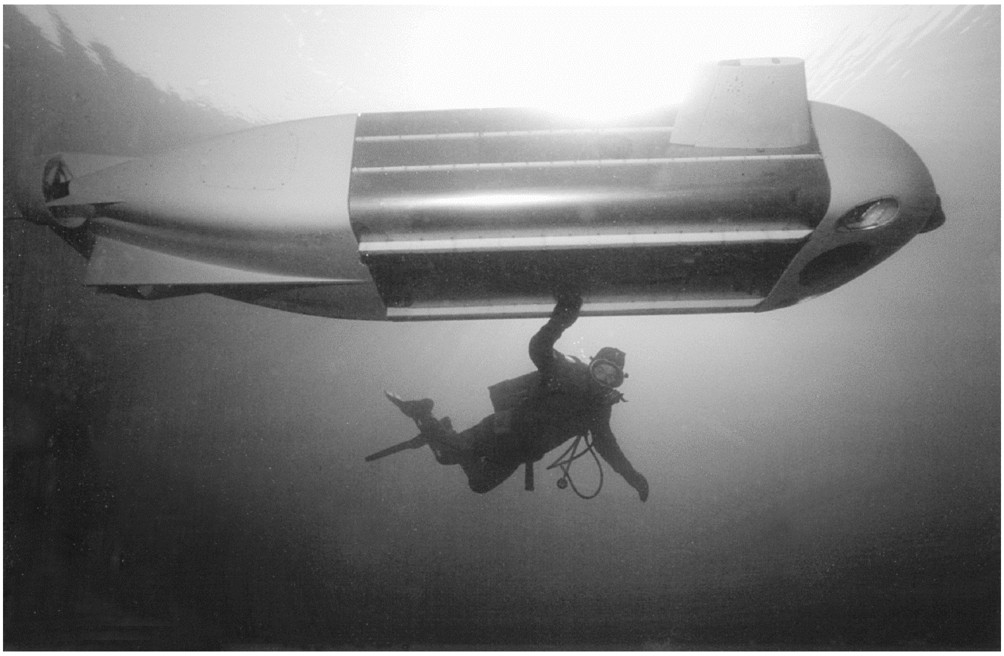

Figure 1: MARTIN

MARTIN is currently being operated in UUV-mode (Unmanned Underwater Vehicle), e.g. supervised by an operator onboard a mother ship using radio or acoustic modem, but will be upgraded to fully autonomous mode (AUV mode). It has a distributed control system centred around four PCs running the OS9000 multitasking operating system for the high level control, and 20 locally placed microcontroller boards for the low level hardware control (Madsen *et al.*, 1996). They are connected by a CAN bus network, which is extended via radio or acoustic modem to an operator on the mother ship. The onboard sensors include inertia platform, DGPS, echo sounder, Doppler log and imaging sonar (Bjerrum, 1996).

The vehicle is aimed at environmental surveys and cable and pipeline inspections, which require high precision positioning and a high degree of data quality assurance. The long range requires a flexible and fault tolerant control system, capable of detecting and compensating for failures. The control system is based on the NIST model (Albus *et al.*, 1989) with multiple levels of modules with increasing level of abstraction. The modules are independent and can be moved between the PCs. They are organised in a control hierarchy structure, which can be restructured on-the-fly in case of failure or reconfiguration. They can be divided into three groups: The Sensor part, including sensor interface and sensor fusion, which obtains the vehicle's position, the Vehicle Control part, mainly consisting of the conning system, which steers the vehicle, and finally the Mission Management System.

In: Proceedings of the European Safety and Reliability Conference, ESREL '97, June 17-20, 1997, Lisbon, Portugal, Page 2301-2308.

THE MISSION MANAGEMENT SYSTEM

The mission management system executes the planned survey, monitors the vehicle, communicates with the human operator and performs emergency actions as necessary.

A very important part of the mission management software is the diagnosis system. All modules include local diagnostics, which monitors the module itself and other modules or hardware it controls. Since this local diagnosis cannot be expected to give a 100% coverage, a global diagnosis function is used to cater for second order effects: a special module monitors the hydrodynamic behaviour of the vehicle by means of a Kalman filter model. It compares the actual vehicle movement with a model based on the actual thruster and rudder movement together with the supposed vehicle state. Discrepancies will indicate a wrong vehicle condition or an error in the data acquisition. Hypotheses about possible errors are then produced by an expert system. Another model calculation, based on each of the hypotheses, thereafter determines whether the hypothesis can be accepted or must be rejected. The diagnostic information is sent to a central diagnosis module, which compiles the status information to get a complete view of the ship and environment condition and determine the possible consequences of a failure. The mission management system will then be able to replan or stop the survey.

The hypothesis generation, the consequence assessment and the replanning will be based on AI techniques using the functional model of the vehicle. This is to be directly implemented as a database structure.

MODELLING METHODS

Functional modelling is an upcoming discipline because the functioning of complex systems is not as obvious as e.g. for simple tools. Therefore, as man made systems become more complex, new means for the analysis and description of the behaviour of such systems are needed. The abstract, general task of a system is to fulfil its goal with success and with a minimised risk of failure. The analysis of the fulfilment of goals by means of functions and supporting hardware and software is called functional modelling.

A few formalised methods have been established for this purpose, i.e. the Multilevel Flow Modelling (MFM) and the Goal Tree - Success Tree method (GTST). MFM (Lind, 1994) describes dynamically the achievement of goals by means of flows of mass, energy and information. Flow structures are connected upwards to describe goal achievement and downwards to describe conditions for the flow structures. A special set of symbols have been defined for visualising functional conditions and goal achievements in the MFM diagrams. GTST (Kim and Modarres, 1987a) features a hierarchical breakdown of goals into subgoals, subgoals into functions, and finally connects functions to components and software with the capabilities needed to establish the lowest levels of the functional hierarchy. The success tree may be visualised as a tree where the single components are connected to the lowest level of functions they implement, and the higher levels of the hierarchy proceed downwards. When there is a need to visualise explicitly the dependence of each low level function on the components realising it, the Master Plan Logic Diagram (MPLD) method may be used for drawing the success tree (Hunt and Modarres, 1987). In this format the components are placed vertically, and a matrix showing the connections are directly visible. The MPLD method has been developed further into a dynamic MPLD, featuring the description of time dependent processes and fuzzy state variables (Hu and Modarres, 1996).

Dynamics are not easily handled by the GTST method. Recently, a method has been developed which combines the features of the MFM- and the GTST methods, namely the Hybrid MFM-GTST or HMG method (Jalashgar and Modarres, 1996). This method aims at failure detection, and for this purpose MFM is most adequate for the high level goal descriptions while GTST is more useful at the lower levels of functionality and realising components.

As a first step towards the functional modelling for MARTIN the GTST method was chosen for the following reasons. A functional decomposition is needed as a basis for a design check list, i.e. for checking that all functions have been implemented during the design phase. GTST is well suited for creating an easily understandable model for this purpose. GTST is also a fairly straightforward hierarchical method for which system diagnosis has been demonstrated (Kim and Modarres, 1987b) by reasoning in a knowledge base, a facility which will also be implemented in MARTIN.

THE SETTING UP OF THE GTST MODEL

The GTST method decomposes the main goal of a mission into subgoals and functions until the level of hardware components. A tree is drawn depicting this dependence. This decomposition tree is useful for the design phase failure analysis, but also very applicable as a basis for the reasoning mechanism in a diagnosis system.

A representative scenario was modelled. In this scenario the task is thought to be the performance of a video inspection of a 30 km long section of a 762 mm diameter oil pipeline in the Storebælt (Great Belt, a strait in the Danish waters) at depths varying between 8 and 35 m. The pipeline was laid down at the sea bed in a trench which is up to 2 m deep at places. The inspection result is to be a video recording and a mapping of the pipeline. The scenario has the following main points:

- Downloading of mission information to the vehicle
- Launch of the vehicle
- Precise measurement of position
- Diving and cruising to the starting point for the survey
- Finding the pipeline
- Cruising along the pipeline and making video recording
- Determining when the end of the section to be inspected is reached
- Cruising back to the mother vessel
- Docking
- Retrieval of survey results

In the GTST model the main goal is the safe, correct and timely performance of the mission. This goal is subdivided in the subgoal "Perform mission correctly on time" and three functions which fulfil the implicitly given subgoal "Maintain safe operation". At the time of writing this paper most effort had been devoted to the further breakdown of the subgoal mentioned first, i.e. the normal operation of the vehicle. The first steps of this breakdown are shown in figure 2. The notation used in the GTST diagrams is the following: basically all goals and functions are shown as boxes, and the connections downwards in the diagram are implicitly AND gates, i.e. all lower level functions must be fulfilled in order for a given function to be fulfilled. Due to the limits of the paper we have introduced a double-box as the notation for a function which is expanded elsewhere. In figure 2 we also use a "mode switch" which will be explained later.

The fulfilment of the subgoal "Perform mission correctly on time" depends on the success of a number of functions, as indicated in figure 2. Already at this level we see an illustration of a limitation of the GTST method, viz. the difficulty of modelling sequential or dynamic behaviour. The functions "Maintain launch capability" and "Maintain docking capability" in principle need not be fulfilled while the vehicle is cruising, but have to be so in the beginning and at the end of the mission, respectively. What the GTST model tells thus is only that the two functions are necessary for the success of the complete mission and must be fulfilled when called for. This is sufficient information when the purpose of the model is to serve as a "checklist" in designing the control system. In the diagnosis applications the mode switch will be used to circumvent this limitation.

The function "Maintain vehicle support" includes, among other elements, the control hierarchy, communication with the operator and supervision of secondary functions. "Perform mission control" comprises the high-level control of the execution of the functions necessary to carry out the mission. "Maintain payload tasks" is a function which is important for the success of the mission, but not so much for the control and safe operation of the vehicle, which has been of prime importance in the investigations, so far. Furthermore, it will be highly scenario dependent. It has, therefore, not been expanded further yet.

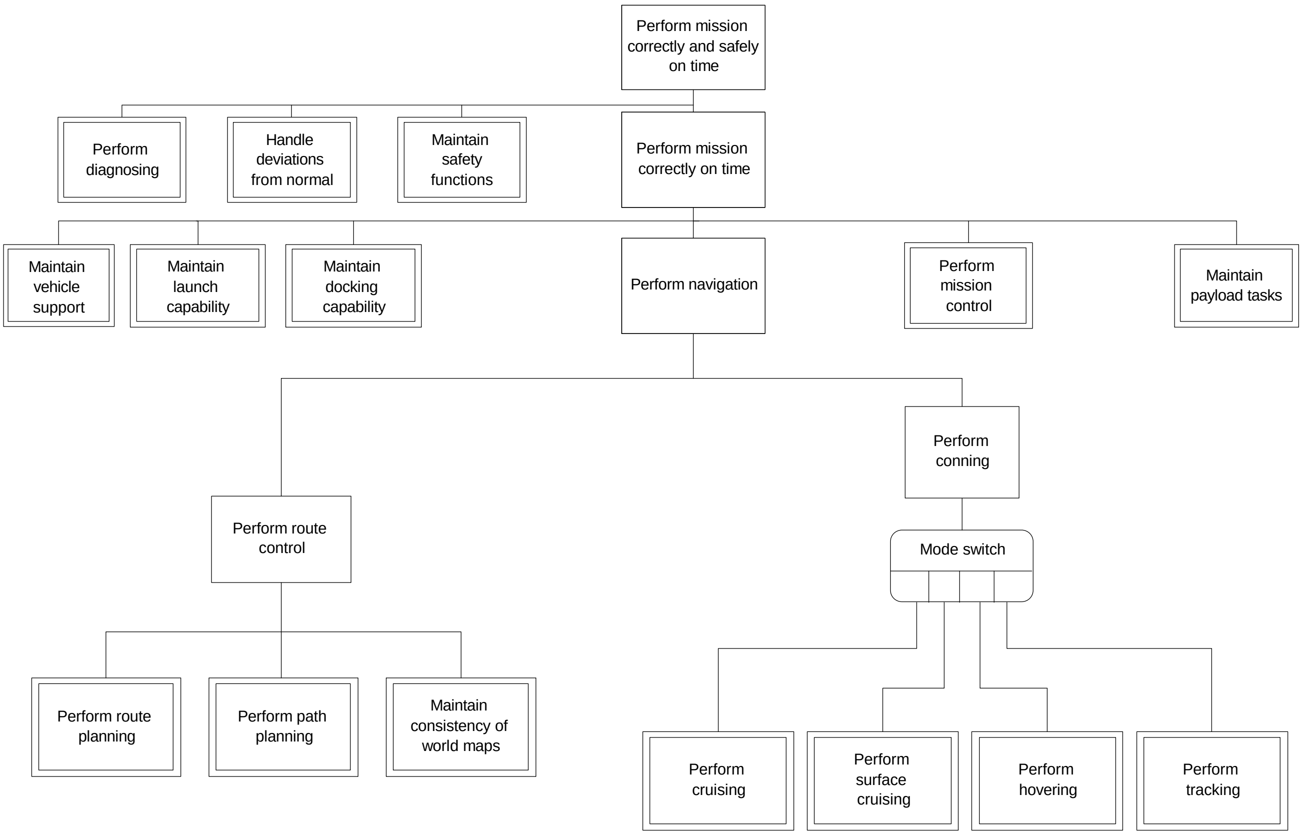

Figure 2: First steps of the GTST model

The function of most interest is "Perform navigation" which has two main parts, i.e. the planning of the route to follow, taking into account obstacles, and the execution of the plans by manoeuvring the vehicle along the prescribed path. The first part is represented by the function "Perform route control" and the second by "Perform conning". In the latter branch we have introduced a "mode switch" in order to represent the various modes of operation of the vehicle during a mission in the same model. Only one of the modes (cruising, surface cruising etc.) can be active at a time, but most of the functions needed further down the tree will be the same for the four modes, although in different combinations. As an example, all the vehicle's thrusters will be needed while hovering whereas only the main thrusters aft will be used for cruising. Control of the shifting between modes will be taken care of by the function "Perform mission control". While the mode switch is a useful tool when drawing the diagram, care should be taken, however, not to extend its interpretation too far at a later stage, e.g. by uncritically converting it into a logical OR-gate. For the execution of the entire mission *all* inputs to the switch are needed, whereas at a given moment only one - but a *specific one* - is active.

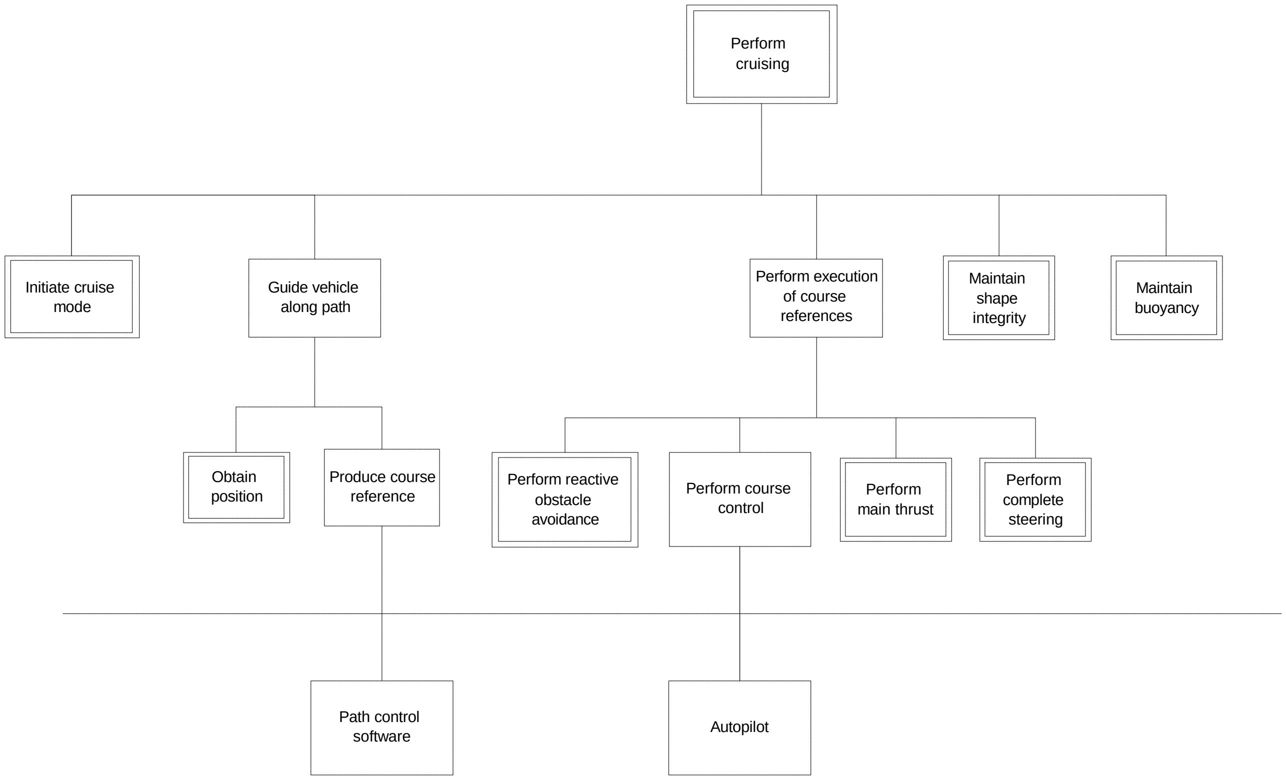

Figure 3 Breakdown to the "hardware level"

As an example, the further breakdown of the function "Perform cruising" is shown in figure 3. For two functions this breakdown reaches the "hardware level" (which also includes software); the transition to the "hardware level" is marked by a dotted line in the diagram. When expanded fully, all functions, of course, will reach the "hardware level". The two main functions under "Perform cruising" are "Guide vehicle along path" and "Perform execution of course references". The first one comprises the calculation of the course to follow based on information about position and the planned path. The latter comprises the following of the course by means of the autopilot controlling the thrusters and rudders etc. and includes the emergency avoidance of late detected obstacles.

The obstacle avoidance is one of the more challenging parts in designing the control system, involving the detection of obstacles by interpretation of sensor signals (primarily sonar signals) and the calculation of a free path or, in case of late detected obstacles, the emergency course change to perform. Based on the experience of the designers of MARTIN as well as other experts in submarine inspections, a number of potential obstacles have been identified, such as big stones, wrecks, anchor chains, timber, fishing nets, fish shoals, seaweed and strong currents. Some of these give good sonar signals and will be fairly easy to identify, whereas others, such as fishing nets, may give weak signals not easy to interpret. In such cases the sonar signal may be combined with other sensor signals in order to detect the nature of the obstacle, either before it is hit or after - for instance seaweed may not be detected until it has been entangled in the propellers; it can then be detected by comparison of information about engine power applied, propeller speed, and the speed of the vehicle. The way to avoid a given obstacle will depend on the kind of obstacle and its position, i.e. whether it is resting at the sea bed, floating in the bulk water or floating at the surface.

The components and software at the "hardware level" can be modelled further, for instance with a view to analysing the effects of component- and software failures to the ability to perform given functions. For this purpose the MPLD method offers a convenient way to get an overview of the dependencies in complex systems. This is illustrated in figure 4, showing a very small part of the MPLD diagram related to the tree shown in figure 3. Components and software are now listed vertically, to the left side of the diagram and

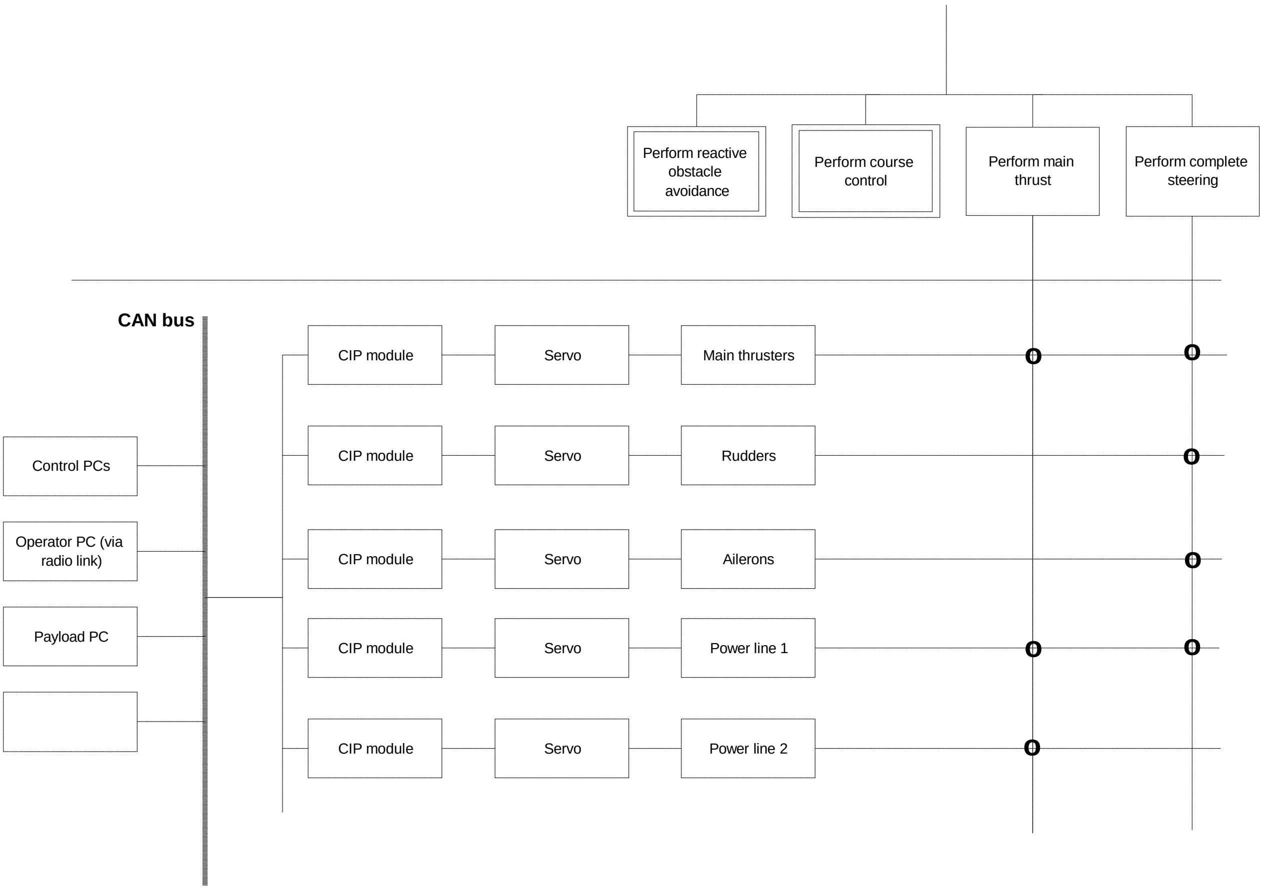

Figure 4 Excerpt of MPLD diagram

connected to the lowest functional level of the goal tree by crossing lines where an "o" in a crossing indicates that the component in question is necessary for the fulfilment of the function above. To the far left in the diagram, the CAN bus is indicated, as well as the high level computers controlling the whole operation of the vehicle.

If necessary, a failure analysis can be performed on the single components in order to assign probabilities to the failure to fulfil the functions and, eventually, the goals in the goal tree.

APPLICATIONS FOR THE GTST MODEL

When all the relevant branches of the functional model have been decomposed until the "hardware level" the model is complete as far as its role in the design of control software is concerned, because at that stage the software modules necessary have been identified and the writing of the software can be performed.

Taking the GTST model further by making a failure analysis of the items at the "hardware level" will provide a basis for an automatic high-level diagnosis system. The system will be able to identify which functions cannot be performed when one or more components have failed, and vice versa point to potential causes when it is detected that a given function cannot be performed.

FURTHER WORK

The automatic high-level diagnosis system will be programmed, probably by means of available expert system tools. Within the frame of the present project, which runs through 1998, part of the GTST model will be converted in order to demonstrate the principle. Testing of the system will be carried out by means of simulations and on the actual submarine in the laboratory and as part of the sea trials planned for testing other systems.

ACKNOWLEDGEMENTS

Funding for the work reported in this paper has been provided by the Danish Energy Agency, the Danish Technical Research Council, and the Danish Academy of Technical Sciences.

Professor Morten Lind, M.Sc. Mikael B. Larsen and M.Sc. student Jakob Rasmussen are gratefully acknowledged for valuable discussions and contributions to the modelling work.

REFERENCES

Albus, J.S., McCain, H.G. and Lumia, R. (1989). *NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)*. National Institute of Standards and Technology, Gaithersburg, USA.

Bjerrum, A. (1996). AUV MARTIN - Update. *Sea Technology* **37:12**, 19-22.

Bjerrum, A., Krogh, B. and Henriksen, L. (1995). Unmanned Mini-Submarine for Offshore Inspections. *Proceedings of the Subtech ’95 Conference,* Society for Underwater Technology, Aberdeen, UK. 91-105.

Egeskov, P., Bjerrum, A., Pascoal, A., Silvestre, C., Aage, C. and Smitt, L.W. (1994). Design, Construction and Hydrodynamic Testing of the AUV MARIUS. *Proceedings of the AUV ’94 Conference*. IEEE Oceanic Engineering Society, Cambridge, USA. 199-207.

Hunt, R.N., Modarres, M. (1987). Performing a Plant Specific PRA by Hand - A Practical Reality. *14th Inter-RAM Conference, Minneapolis, 1987.*

Hu, Y-S., Modarres, M. (1996). Functional Modelling Using Dynamic Master Logic Diagram (DMLD). *Probabilistic Safety Assessment and Management '96. ESREL'96 - PSAM-III. June 24-28 1996, Crete, Greece. Edited by Carlo Cacciabue and Ioannis A. Papazoglou.* Springer-Verlag London Limited. 1073-1078.

Jalahsgar, A., Modarres, M. (1996). Identification of hidden failures in control systems: A functional modelling approach. *Intelligent Systems and Soft Computing for Nuclear Science and Industry. Proceedings of the 2nd International FLINS Workshop. Mol, Belgium, September 25-27, 1996. Edited by D. Ruan et al.,* World Scientific Publishing Co., Singapore.

Kim, I.S., Modarres, M. (1987a). *MOAS: A real-time operator advisory system*. University of Maryland, College Park, USA.

Kim, I.S., Modarres, M. (1987b). Application of Goal Tree - Success Tree Model as the Knowledge-Base of Operator Advisory Systems. *Nuclear Engineering and Design* **104**, 67-81

Lind, M. (1994). Modelling goals and functions of complex industrial plants. *Applied Artificial Intelligence* **8:2**, 259-283.

Madsen, H.Ø., Bjerrum, A. and Krogh, B. (1996). MARTIN - an AUV for Offshore Surveys. *Underwater Systems Design* **18:3**, 21-25.