DAMOCLES:
AN EXPERT SYSTEM FOR DAMAGE CONTROL MANAGEMENT ABOARD STANDARD FRIGATES

by
Ed W.A. van Leeuwen BSc and Michael Perre MA
TNO Physics and Electronics Laboratory

1. ABSTRACT

Under commission of the Royal Netherlands Navy TNO Physics and Electronics Laboratory is developing Damocles, a Damage Monitoring and Control Expert System. Damocles is an artificial intelligent decision support system to be used by the damage control organisation aboard Standard frigates, in particular the damage control (DC) officer. It offers the DC-officer an integrated collection of tools directed at fire-fighting, damage control and maintenance of ship stability. The aim of developing and introducing Damocles is to improve and quicken the decisions concerning (re-)configuration of technical systems in the case of a calamity.

2. INTRODUCTION

TNO Physics and Electronics Laboratory, in collaboration with the NBCD School of the Royal Netherlands Navy, is developing Damocles, a Damage Monitoring and Control Expert System. The main purpose of the Damocles project is the development of an expert system which supports the damage control (DC) officer aboard Standard frigates in maintaining the operational availability of the vessel by safeguarding it and its crew from the effects of weapons, collisions, extreme weather conditions and other calamities. Basically DC-management includes the classical command and control cycle: status maintenance, situation assessment, planning, tasking and evaluation. An important way of making the total DC-organisation more effective is to improve the quality of the decision making process by providing automated decision aids to the DC-officer in addition to the information processing and presentation facilities already available [1]. This applies especially to damage assessment and planning.

The DC-officer on board navy ships is part of the organisation of nuclear, biological and chemical protection and damage control (NBCD). When a calamity has occurred it is very difficult to collect all relevant information concerning the calamity, to structure and interpret the acquired information in order to find a solution for the problem and to ascertain what influence the calamity has on the operational availability of the vessel.

The experience of the DC-officer is very important in the management of DC-situations. The DC-officer must not only consider actions that provide short-term solutions for problems, but also assess the long-term consequences. Due to the nature of DC-situations NBCD-management has to make decisions in a short time frame and under stressful conditions. These decisions are very often based on incomplete information of a complex technical system (S-frigate). When a wrong decision is being made, the consequences can lead to more serious damage as was initially the case. A decision support system can play a crucial role in this kind of situations. The functioning of such a system is dependent on the availability of (technical) knowledge about the structure of the ship and its subsystems, knowledge about the state the ship is in, knowledge about procedures which have to be followed (laid down in documents) and the experience of DC-officers.

More and more conventional systems contain intelligent modules, without the assurance that these modules satisfy the same rigourous quality measures as the conventional ones do. In comparison with conventional software systems the quality of expert systems is viewed as not being very satisfactory. Some of the more problematical aspects are knowledge acquisition, testing, evaluation and the maintenance of the knowledgebase. As yet there is not much unanimity with regard to the ways in which these problems have to be tackled. This is an objectionable state of affairs, especially when you are dealing with critical applications, e.g. process control systems in industry or nuclear power plants.

The same argument is also valid for military Command, Control, Communications and Intelligence (C^3I) systems [2]. A characteristic of these systems is that they consist of large databases with which the deployment of men and material is coordinated. Starting point of this discussion is the thesis that a knowledgebase can be viewed as a collection of facts which can be manipulated with intelligent rules. These rules are also stored as objects in the knowledgebase [3]. An additional convenience of integrating an inference engine with such a database system is that facilities as integrity, concurrency, security, recovery and distribution can now be used inside what we call a knowledgebase system, or expert system [4]. The main theme of this paper is that the integration of database theory and artificial intelligence signifies a step in the direction of a better quality control of expert systems.

In this paper four topics will be discussed. After the introduction we give an overview of damage control management, followed by a description of the Damocles system architecture, conceptual modelling of the knowledge domain and the integration of artificial intelligence and database technologies.

3. DAMAGE CONTROL MANAGEMENT

3.1 Problems encountered in damage control management

In case a calamity has occurred, the DC-officer has to collect and combine data from different sources (sensors, communication systems, orderlies etc.) in order to assess the situation. On the basis of this, the DC-officer plans actions and looks after the careful execution of these actions. There are a number of problems interfering with the decision process: complexity of the vessel, uncertain and incomplete information, time pressure and catastrophic effects of wrong decisions. The DC-officer can only carry out his duty when he has a lot of experience with the vessel and the procedures which have to be executed.

The DC-officer takes charge of combating a calamity from within the ship control centre (SCC), which can be compared with an operator room in a power plant. After a missile hit various ship systems could have sustained damage. At this moment the DC-officer has to take a number of decisions, e.g. concerning the high pressure sea-water system (HPSW, or firemain). How should this system consisting of pipes be (re)configured in order to simultaneously support one or more fire and repair parties (FRP), isolate leakages, keep critical systems (sprinkler, cooling) under pressure and remove inflowing water with bilge-pumps? A more general question could be: What are the relative priorities between attacking a fire, drain off water from compartments or even flooding them (to preserve ship stability)? Each of these problems can be solved easily when they occur in isolation. When a DC-officer is confronted with several of these problems at the same time, he has to weigh one counter-action against another [1].

While executing his task a DC-officer encounters the following problems: complexity of technical systems, incomplete information, stress and lack of experience.

a. Complexity of technical systems. Despite extensive documentation, the systems on board an S-frigate are not easily seen through.

b. Incomplete information. The communication on board a ship often takes place under difficult circumstances (noise, smoke, water, confusion) and is therefore not optimal (correct information can get lost, incorrect or redundant information can be created).

c. Stress. The time pressure is high, because a DC-officer has to take a number of decisions (possibly with great consequences) in very little time.

d. Lack of experience. In order to correctly fulfil his duties a DC-officer must not only have at his disposal an extensive knowledge of the ship and procedures, but also knowledge based on experience.

In brief, it requires a certain amount of experience to make correct decisions in a short time frame, based on incomplete information.

3.2 Damage control tasks

The following DC-management tasks can be identified: detection, data fusion, damage assessment, planning and plan monitoring.

a. Detection. Detection of significant events occurring within the ship or one of its subsystems. Detection can be done by automatic means (sensors) or by human observers. Events of interest are the presence of water, smoke (digital signals) or the value of a temperature, flow, pressure (analogue signals). It is important that these events are notified to the DC-officer.

b. Data Fusion. Usually an event is detected by more than one sensor. Also detections at different points in time may refer to the same event or a set of causally related events. Different sensors provide a different view of the event, with different levels of detail and accuracy. It is essential to fuse all this information into one "objective" picture of the situation: the actual system status.

c. Damage assessment. After having established the ship's status, it is very important to realistically assess the damage of the ship, i.e. evaluate the short term as well as longer term consequences for the ship and mission. Various hypotheses have to be formulated and evaluated and the most likely events have to be predicted.

d. Planning. After having assessed the damage and its probable consequences, actions have to be planned to combat the incident. These actions depend on overall directives from the ship's commander, the desired and actual situation. Actions may range from the closing of a valve to the tasking of a fire and repair party (FRP) to undertake some specific activity. Often alternative courses of action have to be formulated and a decision made in a short time frame. This task also includes the allocation of resources to the various actions.

e. Plan monitoring. The evolving situation has to be monitored continuously, to verify that the conditions for the plan still hold and that progress is being made. If not, other actions might be more appropriate. During the execution of combat actions, priorities may change. Input to this task are all the incoming signals and reports. Thus this task is closely related to the data fusion task. To be more precise, the tasks described form a cyclic process.

4. ARCHITECTURE OF DAMOCLES

4.1 Overview

The Damocles project started with a detailed task analysis of the DC-officer in which the following main task areas were identified: stability monitoring and the prevention and repression of fire and damage. The Damocles system assists the DC-officer in these tasks, most notably the evaluation of the situation (system monitoring and diagnosis), determination of the measures to be taken (planning) and monitoring of the execution (plan-monitoring).

With regard to Damocles two kinds of requirements can be distinguished: functional and operational. Functional demands lay down the functionality of the system, operational demands specify aspects which are of importance when the system is in operational use. The following functional requirements did receive much attention:

- To support fire-fighting, damage control and the maintenance of ship stability,
- To simulate technical systems aboard an S-frigate and
- To query an intelligent database.

The architecture of Damocles is set up along the lines of the so called second generation expert systems. These systems can not only suggest solutions on the basis of heuristics, but can also reason with more fundamental knowledge. Therefore Damocles has at its disposal: knowledge about the structure of the vessel and its subsystems, knowledge about the actual state of the vessel, knowledge about the desired state of the vessel, knowledge about the working of subsystems, knowledge about procedures to be followed and knowledge based on experience of DC-officers. The first three types of knowledge could be called declarative, the last three procedural.

Damocles can be divided into two layers. An under-layer consisting of modules that represent specific subsystems and an integrating upper-layer of modules that support the DC-officer in his duty (and uses the modules in the under-layer). The upper-layer contains the following modules:

- Fire-fighting,
- Damage control and
- Maintenance of ship stability.

The subsystem dependent under-layer contains the following modules:

- Ship,
- Sea-water systems:
    - Firemain,
    - Foam systems,
    - Sprinkler systems,
    - Dewatering systems,
    - Trim systems,
    - NBC prewet system,
    - Cooling systems,
    - Sanitary systems,

- Halon systems,
- Ventilation system,
- High pressure air system and
- Fuel load and transport system.

The interface between Damocles and the DC-officer has been given much attention. The DC-officer can make use of a number of technical systems. At present a prototype of Damocles is running on a Sun-workstation equipped with a colour graphic display in order to present images of underlying technical ship systems (e.g. ventilation and HPSW). These images consist of 2-dimensional representations of pipe networks containing all relevant information required for monitoring and control of these systems. In addition to this, a 3-dimensional image of the ship with overlays of technical systems is used to convey spatial information. On-line manipulations can be made to simulate the consequences of decisions. Usually these decisions are made by inspecting complex lay-out schemas and diagrams which show where pipes are running and how they are connected, etc. Damocles offers the opportunity to simulate the earlier mentioned technical systems and it gives concrete solutions for certain problems. This is realized by reasoning on the basis of structural and functional information about the system components.

4.2 Damocles system tasks

The following functions can be identified for the decision support system (comparable with the DC-management tasks): monitoring state of ship, diagnosis, prediction of consequences of problems or causes, planning of actions, execution and monitoring of chosen plan, prediction of consequences of possible orders. In the following paragraphs these tasks are described in more detail.

a. Monitoring state of ship. This task should observe state variables and compare them with their norms. The collection of norms forms the goal state of the ship. The diagnosis task is activated if there are discrepancies between variables and their norms. There is a discrepancy if the norm for the state of a valve is open and the real state of that valve is closed.

b. Diagnosis. This task tries to find a cause for the discrepancy. If there is no pressure in one pipe, the cause can be a malfunction of a pump or a leakage in a pipe.

c. Prediction of consequences of problems. The impact of problems on the state of the ship has to be predicted to give those problems relative priorities. Fire in an ammunition store receives a higher priority than a loss of water pressure in a lavatory.

d. Planning of actions. To solve a problem or minimise the impact of it, actions have to be executed. Thus one has to synthesize a plan, a collection of successive actions. This is a kind of state-space search, in which the goal state is described by a collection of norms. The plan has to be optimised, the effort to achieve it has to be minimised.

e. Execution and monitoring of chosen plan. If there is a plan the actions have to be executed in the right order. This task monitors this process, e.g. a pump cannot be switched on before it is initiated.

f. Prediction of consequences of possible orders. If one changes a norm, one introduces discrepancies, new problems, new consequences and new actions to be executed. So, before changing a norm the consequences have to be predicted and a confirmation of the user is required. This is to prevent the user giving orders to the system with unforeseen consequences.

Artificial intelligence provides the tools and techniques to use effectively the knowledge about the vessel and to accept, combine and fuse the data from sensors and reports [5].

5. CONCEPTUAL MODELLING

Expert systems are often developed with rather exotic methods and implemented in an AI-shell. This causes a number of problems; how can the expert system be embedded in other software, what communication links are possible and where do we get system developers who are acquainted with the used methodology? It is our conviction that these problems can be alleviated by using the relational model. E.g. coupling an expert system with an existing C^3I-system or RDBMS should be easy. Already during the specification of the knowledge, conventional system analysis and design methods can be used. The system can also be implemented with tools of established reputation. A big advantage of the relational model is the fact that very complex models and manipulations of data (and rule) sets can be defined with a limited number of concepts and operators.

Damocles contains models of several technical ship systems. These models can be used in different ways:
- To simulate the system or train students: What would happen if?
- To search for faults in the system (diagnosis) by comparing the functionality of the model with the reality
- To make statements about partially observable systems; The status of the system is represented in the model. The observed status can be compared with the model status. If there are discrepancies the model status needs to be adjusted. By using a model one can make decisions about a technical system, even when the information on hand is incomplete.

The conceptual model has to be a complete and consistent representation of a knowledge domain in which a distinction is made between a knowledge scheme (definitions of all used facts and relations) and the actual knowledgebase. This distinction can also be seen as a separation of types and instances. Consistency and completeness can be maintained by means of constraints on the actual knowledgebase. A conceptual model has to be constructed with a development method that lays down explicitly the definition of facts and their interrelations. In an AI-methodology like KADS (Knowledge Acquisition and Structuring) four levels of knowledge are being distinguished [6]: domain, inference, task and strategic levels. NIAM, Nijssens Information Analysis Methodology, and ExtendedNIAM, can be used to structure these four levels of knowledge [7,8,9]. On the domain level the basic elements and their interrelations are specified. The inference level contains the inferences which can be initiated based on the domain level. The third level of knowledge is the task level, which specifies process structures and tasks. This level is specified with the use of KADS. The resulting conceptual model is laid down in a relational data structure. The fourth knowledge level is not used in the Damocles system. By using (E)NIAM the quality of extensive data- and knowledgebase specifications is guaranteed.

6. INTEGRATING ARTIFICIAL INTELLIGENCE AND DATABASE TECHNOLOGIES

6.1 From database to knowledgebase

Relational database management systems (RDBMS) are primarily being used for administrative applications. Concepts like data independence, data integrity, controlled redundancy, security and privacy are also very important when you are dealing with knowledgebase management systems (KBMS). Other reasons to adhere to the relational model are its conceptual simplicity and the fact that system developers are familiar with it. Most software producers are more experienced in using Oracle and Ingres rather than Lisp, Prolog or AI-development tools like KEE, ART or Knowledge Craft. Moreover it is not easy to become acquainted with such advanced tools. By using an RDBMS update anomalies and redundant storage can be prevented. It also offers a flexible growth path when operational concepts are changed or data structures are modified.

When a knowledgebase is viewed as a special kind of database, various facilities of DBMSs could be used in KBMSs. Examples are recovery (to restore a knowledgebase after a calamity, fault or power failure), concurrency (simultaneous utilization of a knowledgebase by different users), distribution (physically distribute a knowledgebase over different locations), security (protect a knowledgebase against unauthorized usage) and integrity (guard against inconsistencies of the knowledgebase) [4]. Especially this last point enables a direct relation with analysis and design methods of databases. Consequently, there is a need to build a conceptual model of a knowledge domain. A conceptual model can be placed between the internal model of a knowledgebase (the way in which the relations are physically represented) and the external model (the way the user sees the system).

An obvious choice for the representation of a conceptual model of reality, represented with NIAM, is a relational database. This implies that the domain and inference level can be implemented in a relational database. The procedural level has to be represented preferably in a relational programming language: Prolog is a logical choice. Most RDBMSs can be approached by means of the standard interface and query language SQL (Structured Query Language). This leads to hardware and operating system independent storage systems. Different kinds of software can be integrated when the interfaces between them are implemented with SQL.

One of the programming languages used in artificial intelligence research is Prolog (Programming in Logic). This language bears much resemblance with relational systems and can therefore be used to implement a model which consists of relations. Prolog can be viewed as a relational query language, because it is based on relations and contains relational operators. It differs from an RDBMS with respect to the manner in which data is being stored: Prolog rules and facts are placed in internal computer memory and not in files. Prolog is not only a query language, but also an implementation language that can be used to develop software systems. Prolog has considerably more potential than SQL (the use of recursion) and other programming languages (meta-programming facilities). There are also parallel versions of the Prolog language. Deductive database systems, also called expert database systems or knowledgebase management systems, combine the features of RDBMSs and logic programming languages.

6.2 Architectures for knowledgebase management systems

Several proposals have been made with respect to the architecture of knowledgebase management systems (KBMS). Often a distinction is made between loosely-coupled and tightly-coupled KBMSs [3].

A loosely-coupled KBMS is an external database management system (DBMS) that is interfaced with a logic programming language: e.g. the Oracle DBMS coupled with the logic language Prolog. Prolog-rules in this configuration can activate queries on the database.

The possibilities of a combination of a relational database and Prolog surpasses those of conventional AI engineering environments. TNO Physics and Electronics Laboratory uses an integrated development environment consisting of Oracle, Quintus Prolog and ProWindows (a Prolog based tool for making graphical interfaces) to realize the Damocles project. The following extras can be envisaged:

- Check on the consistency of knowledge,
- Secure the knowledge against unauthorized usage,
- Distributed storage of knowledge,
- Multitasking and distributed processing,

- Recovery facilities and
- Availability of many development tools.

In a tightly-coupled KBMS there is no distinction between a database system and a logic language, i.e. the strict distinction between database operations and the inference mechanism is abandoned. This can be realized in two ways: Firstly a logic language can be extended with database facilities like integrity, concurrency, security, recovery and distribution. Secondly a DBMS can be extended with deductive (Prolog-like) facilities. An interesting example of this architecture is Postgres, a further development of the DBMS Ingres (Post Ingres) [10,11].

Postgres is a tightly-coupled KBMS developed at the University of Southern California, Berkeley. The main aims of the project are to uphold the relational model and to provide facilities for "active" databases and inference, including forward and backward reasoning. In many applications it is very convenient to use triggers and alerters. Triggers are small pieces Structured Query Language (SQL) program which can be activated when changes are being made in the database (e.g. insert, delete or update). Alerters are comparable with triggers, but are activated by time or date.

The most revolutionary aspect of Postgres is the use of rules and procedures as if they were plain data items. Nijssen views an expert system as a system that contains human expertise and consists of a collection of related facts [7]. These facts can be inserted by a user, or can be derived by the system itself on the basis of other facts and inference rules. Rules in Postgres can perform forward and backward chaining. This can be achieved by "early" and "late" evaluation. In the case of early evaluation a change in a data item that is contained in a rule will directly lead to activation of this rule. In the case of late evaluation the change only becomes obvious when a user queries that particular data item.

When inference rules are used in a KBMS, it is possible to perform a run-time "computation" of a relation. In other words the system has at its disposal an intension of the application (definition of tables, mutual dependencies and inference rules) and when necessary computes the extension (the actual facts in tables). When simulations are being executed (planning task) it is also possible to store different system states in separate databases.

7. CONCLUSIONS

A relational database enhanced with Prolog offers good opportunities for the transparent development of extensive and highly qualitative expert systems. The usefulness of artificial intelligence and expert systems in DC-management is demonstrated by the development process of the Damocles system.

Damocles offers the following advantages:
- Quicker and better decisions in the case of a calamity
- Easy to extend with other types of technical system (e.g. electrical system or weapons system)
- Modular structure eases the incorporation of other types of vessel (e.g. Multi-purpose frigate, Guided missile frigate, container ships) or even another type application (e.g. power plants, management of communication networks)
- Optimal portability to other types of hardware as a result of the chosen software tools (Oracle, Prolog)
- Not only intended to be used aboard navy frigates, but also in training surroundings at the NBCD-School
- Not susceptible to (battle)stress.

For the sake of completeness a few words have to be said about the limitations of Damocles. The operational requirements were not important during development of the prototype system. That is why the system is not mil-spec or real-time. At the moment there is also no direct coupling with the various sensor systems aboard the S-frigate. However, these and other aspects of developing an operational system to be installed on every S-frigate are of later consideration. The first step in this direction will be the incorporation of Damocles in a simulator of the S-frigates SCC, which is placed in the NBCD-School in Den Helder, The Netherlands. After an extensive evaluation and test of the system (beginning in October 1990) a decision has to be made whether Damocles will "go to sea".

8. REFERENCES

[1] Geer, D.W., "Integrated Damage Control System, a Survivable Shipwide Architecture Applying Total Ship Resources Based on Informed Decision Making", In: "Marineblad: Issue of the Royal Society of Naval Officers", Vol. 98, Nr. 10, 1988, pp. 429-437.

[2] Napheys, B., D. Herkimer, "A Look at Loosely-Coupled Prolog Database Systems",In: Kerschberg, L. (ed.), "Expert Database Systems, Proceedings from the Second International Conference", Benjamin/Cummings, 1989 , pp. 257-271.

[3] Stonebraker, M., M. Hearst, "Future Trends in Expert Database Systems", In: Kerschberg, L. (ed.), "Expert Database Systems, Proceedings from the Second International Conference", Benjamin/Cummings, 1989, pp. 3-20.

[4] Date, C.J., "An Introduction to Database Systems: Volume II", Addison-Wesley, 1983.

[5] Brodie, M.L. (et al.),"Future Artificial Intelligence Requirements for Intelligent Database Systems", In: Kerschberg, L. (ed.), "Expert Database Systems, Proceedings from the Second International Conference", Benjamin/Cummings, 1989, pp. 45-62.

[6] Breuker, J. (ed.), "Model-Driven Knowledge Acquisition: Interpretation Models", University of Amsterdam, 1987.

[7] Nijssen, G.M., "Knowledge Engineering, Conceptual Schemas, SQL and Expert Systems: A Unifying Point of View", In: "Relationele Database Software, 5e Generatie Expertsystemen en Informatie-analyse", Congress-syllabus NOVI, 1986, pp. 1-38.

[8] Nijssen, G.M., Halpin, T.A., "Conceptual Schema and Relational Database Design: A Fact Oriented Approach", Prentice Hall, 1989.

[9] Creasy, P.N., "ENIAM: A More Complete Conceptual Schema Language", In: "Proceedings of the Fifteenth International Conference on Very Large Databases", 1989, pp. 107-114.

[10] Stonebraker, M. (et al.), "The POSTGRES Rule Manager", In: "IEEE Transactions on Software Engineering", Vol. 14, Nr. 7, July 1988, pp. 897-907.

[11] Stonebraker, M. (et al.), "The Implementation of POSTGRES", In: "IEEE Transactions on Knowledge and Data Engineering", Vol. 2, Nr. 1, March 1990, pp. 125-142.

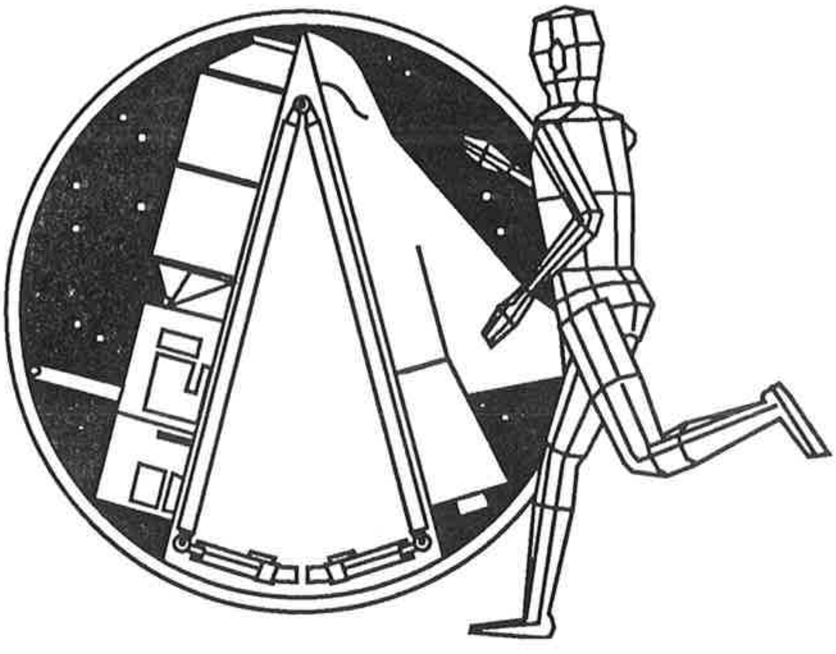
PROCEEDINGS

ARTIFICIAL INTELLIGENCE AND KNOWLEDGE-BASED SYSTEMS FOR SPACE

- Table of Contents -

22-23-24 May 1991

ESTEC, Noordwijk, The Netherlands



Organised by:

THE EUROPEAN SPACE AGENCY Simulation & Electrical Facilities Division - Technical Directorate