DIAMS Revisited: Taming the Variety of Knowledge in Fault Diagnosis Expert Systems

M. Haziza , S. Ayache, J.-M. Brenot, D. Cayrac, D.-P. Vo

Matra Marconi Space, 31, rue des Cosmonautes, 31077 Toulouse Cedex – FRANCE.

e-mail : haziza-marc@mms.matra-espace.fr

Abstract

Matra Marconi Space (MMS) has been developing spacecraft diagnostic support systems for eight years. The DIAMS program, initiated in 1986, led to the development of a prototype expert system, DIAMS-1, dedicated to the Telecom 1 Attitude and Orbit Control System, and to a near-operational system, DIAMS-2, covering a whole satellite (the Telecom 2 platform and its interfaces with the payload), which was installed in the Satellite Control Center in 1993. The refinement of the knowledge representation and reasoning is now being studied, focusing on the introduction of appropriate handling of incompleteness, uncertainty and time, and keeping in mind operational constraints. For the latest generation of the tool, DIAMS-3, a new architecture has been proposed, that enables the cooperative exploitation of various models and knowledge representations. On the same baseline, new solutions enabling tighter integration of diagnostic systems in the operational environment and cooperation with other knowledge intensive systems such as data analysis, planning or procedure management tools have been introduced.

I. Introduction

Spacecraft (S/C) operations have pioneered the introduction of the Knowledge-Based Systems (KBS) technology in Space. The prototyping activities conducted in the eighties have allowed to demonstrate the potential of KBS to assist in controlling space systems. Knowledge-Based Systems in S/C Control Centers (SCC) have proven to have a high potential for

- assisting spacecraft engineers in monitoring and analyzing S/C data, and in diagnosing on-board failures from the knowledge of the S/C state obtained through the telemetry.
- assisting S/C engineers in complicated operations where the exact sequence of operations is determined by external constraints and by the actual S/C state at each step.

Spacecraft Assembly, Integration and Test (AIT) is also becoming a knowledge intensive activity that requires appropriate knowledge-based assistance. Due to the increasing complexity of space systems, an increasing number of parameters have to be tested before launch through more and more elaborated test procedures. At the same time, the duration of the AIT phases is continuously decreasing. This makes the AIT phase a critical phase in almost all present space projects and increases the pressure on the development teams.

The use of knowledge based systems for emergency management, fault diagnosis, resource management, replanning/rescheduling, etc. and the operational integration of such facilities in future ground infrastructures (SCC's, AIT environments) should help lowering the risks in problem diagnosis and selection of recovery actions, avoiding mis-diagnosis that might endanger the system in-orbit or under test, and eventually reducing the overall cost of the AIT & operation phases.

These general considerations motivated the launch of the DIAMS program by the mid-eighties. DIAMS is a step-wise fault diagnosis expert systems development programma initiated by Matra Marconi Space with support from CNES in 1986.The analysis of the DIAMS programma illustrates the progressive approach adopted by MMS to master the inherent complexity of the knowledge required while delivering successive generations of knowledge-based tools that can actually provide support in spacecraft operations.

II. DIAMS-0: the first steps

First experiments in the domain of diagnosis were conducted in 86. An early mock-up was developed in Smalltalk. It allowed to confirm some basic knowledge representation and reasoning principles

and particularly the importance of model-based approaches and object-oriented knowledge representations.

The Object-Oriented (OO) paradigm was found well-suited to the implementation of knowledge-based systems. In the OO paradigm, each elementary problem-solving competence may be attached as a method to one or several domain object classes.

The Model-Based approach clearly distinguishes on the one hand the application domain which is modelled in terms of functional or behavioral components and on the other hand generic reasoning mechanisms that can interpret such models and work on them. KBS implementing the model-based approach may be decomposed into domain-independent modules - the *KBS shell* - on the one hand and *domain-specific Knowledge Bases (KB)* on the other hand. The KBS shell implements the core of the inference process (basic knowledge representation and reasoning mechanisms, general problem-solving strategy) and the external communication services (user interface, interface with the operational environment). It is generally reusable for other target systems of the same nature, possibly through customizing of the external communication services. The Knowledge Bases are generally specific to the target system (a specific S/C system or subsystem for instance).

III. DIAMS-1: Establishing the founding principles

The development of a first generation of diagnostic tools, DIAMS-1, started in 1986. The project was co-sponsored by the French Space Agency. It led to the delivery of a prototype Expert System dedicated to the TELECOM 1 Attitude and Orbit Control System (AOCS) [7]. The selected implementation platform was the SUN/UNIX environment and an object-oriented dialect on top of Prolog called Emicat. Graphical interfaces were developed on top of Sunview. The prototype was installed in the TELECOM 1 SCC and evaluated by the operation staff in 1989 [8].

Setting up the basic knowledge representation and reasoning mechanisms

Knowledge Islands

One of the main advances realized through DIAMS1 was the decomposition of the knowledge base into different categories of so-called *Knowledge*

Islands (KI) representing the different domains of expertise required for diagnosis
- hierarchical decomposition of the system into functions with identification of basic commands and observables
- qualitative models of behavior
- shallow knowledge required for solving the most common problems or to deal with situations where the expert understanding is not deep enough to include a functional or a behavior model

The notion of knowledge island turned out to be particularly well-suited to the management of the different natures of knowledge. It greatly facilitated the KB maintenance and incremental refinement. It also made easier the local implementation of new types of knowledge, including new or refined knowledge representation paradigms designed to achieve a finer representation.

Functional knowledge

The functional model consists of a set of functional diagrams, grouped into knowledge islands, and describing at the component level:
- the functional elements of the system,
- the functional links, representing possible influences between functional elements,
- the observable parameters (telemetry) associated to some of the functional links, and the available telecommands.

The functional model is hierarchical and its deeper level corresponds to the limits of the satellite commandability and observability. It depicts telecommands and telemetries connections and corresponds to the switching diagrams used in S/C operation engineering activities (figure 1).

Figure 1. Example of functional diagram

698
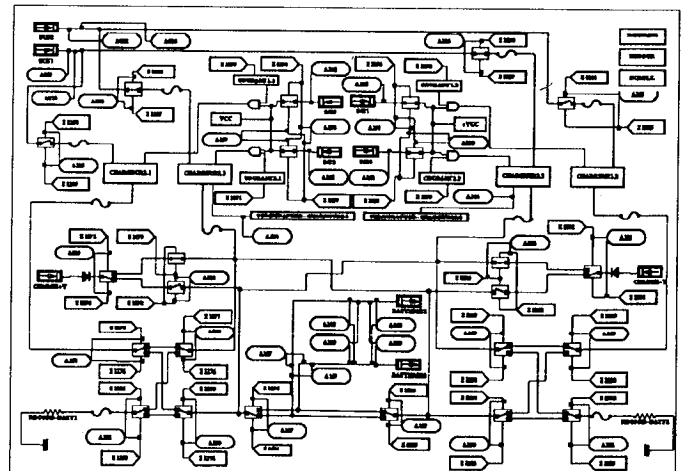
For each functional element, a propagation function defines how abnormal influences received are propagated to other elements, under the assumption that it is nominal (not faulty). It describes how this component responds to abnormal input influences, or how its inputs can be abductively suspected when its outputs are in abnormal states.

The main justification of this hybrid model based approach is that, because the systems modelled are very complex, the functional elements do not have a general description of their behavior. In other words, the model is not built to provide predictions of all the possible behaviors of the modelled system. It is rather a qualitative representation of the possible fault propagation between the components of the system. The fault modes of the suspected unit(s) are defined only by their signatures in terms of abnormal output(s). Fault modes do not need to be systematically identified a priori. Interactions between components can stand for all kinds of physical signals (e.g. electrical, command signals, thermal influences). A very restricted set of states has been shown sufficient in most cases to represent the propagation of faults over the functional layouts.

Diagnostic reasoning in a functional KI may be decomposed into three fundamental tasks which are:
- hypotheses generation: given suspect links pointed out by a behavior analysis or by previous analyses in other functional KI's, find out which functional elements might account for the symptoms. This result is achieved by backward propagation of the anomalies through the links between the functional elements, using the propagation functions abductively.
- hypotheses elaboration: given the set of suspected functional elements given by the reasoning in the previous step, determine what the impact of their fault would be on the observables of the KI currently investigated. This is achieved through forward propagation through the links, using the propagation functions deductively.
- hypotheses discrimination, that is discriminate among the hypotheses coming from the first step by adding more information about other observable parameters generated at the second step. The principle of the diagnosis is then to enter a discrimination loop between the possible causes. The system selects an observable according to various criteria, like the reliability of the measure or the discrimination power of the observable, and

then asks for its qualification. Depending on the nature of the response, some possible causes are discarded (the ones which are incompatible with the qualification of the observable given by the user). If there are still discriminating observable parameters, another step of the loop is entered, otherwise the result of the diagnosis is either a single cause or a set of non discriminated possible causes.

Behavior knowledge

The behavior Knowledge meets the requirement for system level knowledge that allows to rapidly get a partial conclusion about the origin of the problem (reconfiguration criterion, global fault corresponding to some system state variables) and then to focus the attention on some subfunctions of the functional model and so to limit the exploration of the functional model to these subfunctions.

Standard forms were defined to capture the AOCS behavior knowledge. These forms were used to specify in a systematic way all the observables (e.g., the roll angle), system variables (like the nozzle firing command or the nozzle state variable) and the observable manifestations (e.g., the displacement of the S/C nutation center along the roll axis after an actuation sequence) necessary to represent the behavior of the system together with the relationships existing between these different elements. The behavior model also contained a number of causal relationships representing the AOCS automatic reconfiguration logic. Once this information was entered in the KB, the KBS shell could build the causal graphs relating system variables, fault modes, and observable manifestations, and discriminate between them using the same generic inference mechanisms as in the functional model (figure 2).

Figure 2. Examples of behavioral relationships
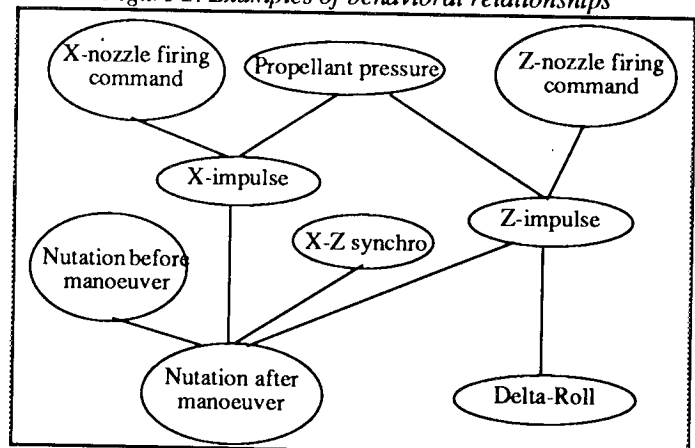
Lessons learned from the experimentation phase

The main results of the experimentation phase were gathered in a document jointly elaborated with the Telecom 1 operations [8]. The experimentation of the prototype was very useful in clarifying the situation and mission of the expert system in the SCC and in refining the operational requirements. It confirmed DIAMS-1 basic knowledge representation and reasoning mechanisms. The general conclusion was that the DIAMS approach improved the communication between the S/C manufacturer and the SCC staff, and that, as a model-based system, DIAMS provided the SCC staff with a better knowledge of the S/C functions and behavior. The experimentation phase also indicated how additional functionalities could be implemented in future versions of the system.

The DIAMS-1 experimentation phase demonstrated that the approach chosen was ripe for being applied in large scale applications. It convinced the French Space Agency to start the development of a full scale diagnostic support system for TELECOM 2 satellites.

Two of the technical lessons learned during the experimentation phase are worth being recalled here:
* An important part of the S/C knowledge is available under graphical form (functional diagrams for instance). The experimentation emphasized the importance of the graphical model edition and animation services. Graphical model editors are needed for instance for building the functional model and checking the graphical consistency of its hierarchical decomposition. Model animators are needed to display and to animate the appropriate diagrams during reasoning. Models editors and animators require a development tool which offers an object-oriented language for modelling the domain semantics (semantic objects) and integrated graphical utilities to manage the interactions between the semantic objects and their graphical representations.
* It was also remarked that some basic mechanisms could be reused in the framework of the S/C project to support a number of design activities. The hypothesis elaboration mechanism could be for instance adapted to perform impact analyses - e.g., to figure out the impact of a given fault or a given telecommand on the system observables. Impact analysis is one of the main techniques used for instance to elaborate the TM/TC plan or to analyze

failure modes effects and criticality (the FMECA) during the S/C design phase. TM/TC Plans and FMECA also are major sources of information for the construction of the KB and the optimization of the diagnostic strategy.

IV. DIAMS-2: Maturing the knowledge modelling and the development process

Through DIAMS-2, MMS addressed the development of a fault isolation tool covering a whole spacecraft: french telecommunication satellite TELECOM 2. This project was the consequence of the very positive results of the development and evaluation of the DIAMS-1 prototype [9][2][3][4].

DIAMS-2 was developed over a period of 4 years from 1989. The selected implementation platform was the KEE/CommonLISP object oriented environment which was considered the reference environment for KBS development when the DIAMS-2 project was started. It also complied with the semantic-graphic integration requirement that resulted from the DIAMS-1 experimentation.

Refining Knowledge Modelling

DIAMS-2 is a hybrid system combining decision tree based symptoms - hypotheses associational reasoning to initiate diagnosis and to focus the reasoning on particular functions and components and the DIAMS-1 model-based techniques to complete diagnostic reasoning on particular functions and to provide the final isolation of the fault.

Investigation Procedures

The decision-tree based knowledge, called Investigation Procedures (IP) in the latest generation of the tool, adds a strategic layer on top of the functional model. It is used to select among pending hypotheses and to focus the attention on definite parts of the functional model (figure 3).

IP modelling starts at the system level, implementing a top-down approach. The used knowledge is elaborated by S/C operation engineers during the mission preparation phase. It corresponds to the Contingency Operations section of the Operations Preparation Handbook. IPs can be enriched on the basis of anomalies experienced during the S/C in-orbit lifetime.The knowledge is represented as decision trees whose nodes are either binary tests (e.g., testing whether a given parameter is abnormal) or actions on the satellite (e.g., sending a

telecommand that will allow to discriminate between candidate hypotheses).

Figure 3. Examples of IP components
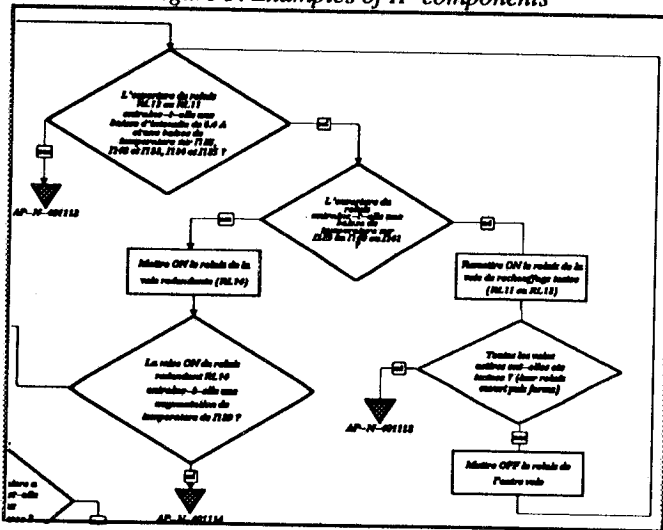
A diagnostic session starts when the user inputs a set of anomalies. The initial tests implement a discrimination strategy at system level. These tests are mainly membership tests which aim at localizing the satellite subsystem where the primary anomalies have occurred. This kind of procedures can often be automated.

At subsystem level, the diagnostic strategy consists in using as far as possible higher level observations and characterizations of the satellite behavior or evolution, in order to simplify or even avoid in-depth analyses involving the functional model. Connections with the functional model are reached when tests involve large numbers of telemetries and need reference states to compare the current situation with.

Figure 4. Investigation of a functional KI with DIAMS-2
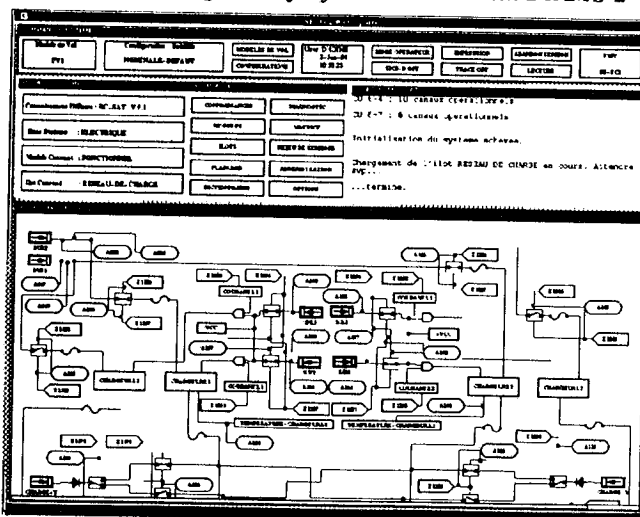
Maturing the Development Process

Moving to a full scale industrial application raises stringent requirements in terms of Knowledge Management and KBS Development Methodology. With support from CNES, MMS elaborated a first set of Software Engineering principles and Quality Assurance rules applicable to KBS projects that benefited from the experience acquired in DIAMS-1.
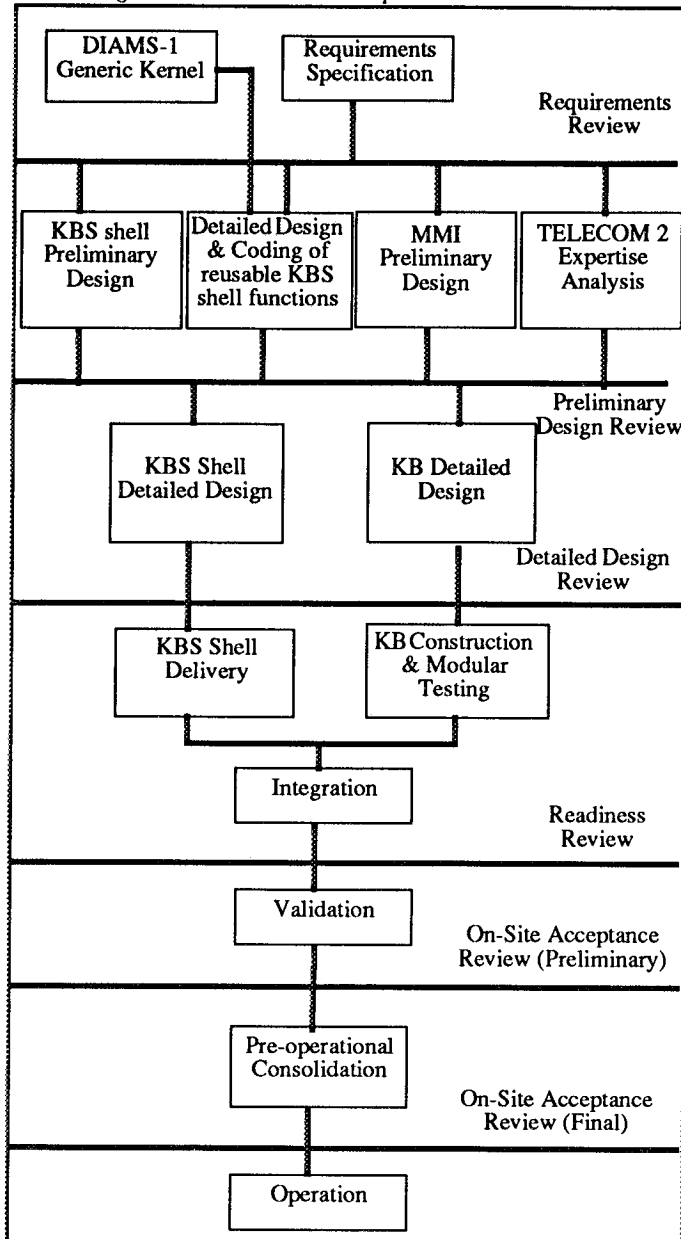
The construction of the Knowledge Base was conducted by a dedicated team independent from the KBS shell development team. The KB development team performed the capture of knowledge and the construction of the KB using well-suited methods and tools in compliance with the representational constraints of the operational environment. It also maintained close relationships with the target system project organization - essentially through cooperation with the TELECOM 2 operation engineering team; the System, Subsystem and Integration specialists of the S/C project did not directly participated in the construction of the KB.

The development of a KBS shell is rather similar to a conventional SW development, and requires the same kind of methods and tools for design, coding and testing. The design of the DIAMS-2 KBS shell inherited most of the basic knowledge representation and reasoning mechanisms already implemented in the DIAMS-1 prototype and validated during the experimentation phase. A dedicated team assumed the design, coding and testing of the tool basic functionalities. A third team, independent from the development teams, was in charge of the quality control and of the integration and final validation of the KBS.

A pre-operational consolidation phase was scheduled in the continuation of the KBS development phase. Its goals were
- to familiarize the SCC staff with the KBS
- to experiment and eventually to enact the KBS utilization and maintenance procedures
- to consolidate and validate the external interfaces with the SCC information system, including the S/C and Simulator data access procedures.
- to calibrate tests and explanations on-site with the end-users.
- to refine some knowledge islands to account for the in-orbit experience (e.g., the S/C in-orbit thermal behavior).

701

Figure 5. DIAMS-2 Development Plan Overview

Integrating the end-user in the development process

Cooperation between the KB development team and the SCC staff is needed, during the construction of the KB, to ensure consistency between the knowledge representation formalisms used in the SCC and those used in the KB. A close cooperation is also needed when the system is transferred from the development site to the operation site.

In DIAMS-2, the integration of the end-user in the development cycle was founded on the following principles.

- The S/C User's Manual (UM) remained the reference document for the transfer of information between the S/C manufacturer and the SCC. The level of decomposition of the models was the UM's one, and the same graphical representation modes, and variable identifiers were used.

- Operation Engineers from the S/C project were involved in the development process to continuously maintain consistency between the DIAMS-2 KB and the S/C User's Manual.

- A TELECOM 2 SCC representative was included in the KB development team. His mission was to check that the knowledge representation used (symbology, nomenclature) was consistent with the one used in the SCC, that the functional model was compatible with the hierarchical view of the S/C and the monitoring sets defined in the SCC, and that the observables used were actually accessible through the SCC. Conversely the KB was developed in such a way that the SCC engineer could draw benefit from the KB design and development activity.

Remark: The TELECOM 2A/2B launch campaigns took place during the DIAMS-2 KB Detailed Design phase. This resulted in a lack of availability from both the S/C operation engineering team and the SCC personnel. A first consequence was that an important effort had to be devoted to the refinement of the KB during the pre-operational consolidation phase. This again confirmed the crucial importance of a right phasing with the S/C and SCC development activities, and more generally of a tighter integration between the KBS, SCC and S/C development processes.

V. DIAMS-3: the Integration Age

In DIAMS-2, comprehensiveness and efficiency was privileged against fineness of representation and reasoning. Simplified representations of knowledge, generally well-suited to the practical problems faced in spacecraft operations were introduced as a first approximation. However, in some specific knowledge islands, refined representation and reasoning techniques are required to appropriately handle time, incompleteness and uncertainty. This last refinement step is now being considered through the development of a new generation of diagnostic tools called DIAMS-3 that started in 1992 [5].

Other important objectives of DIAMS-3 concern the reduction of the knowledge acquisition efforts, tighter integration with other knowledge-based tools like data analysis or procedure management tools, and more generally the complete integration of the diagnostic system in the operational loop [10].

C++ is the implementation language retained for DIAMS-3. Beyond porting the DIAMS-2 machinery into C++, DIAMS-3 provides generic model edition services and a set of libraries of operational standard for handling time, incompleteness and uncertainty and for cooperation with other knowledge-based tools (knowledge interchange format and protocol, mapping engine, exchange monitor, etc.). These libraries and basic services, all developed in C++, will be reused in other KBS development projects.

Integration Issues

The different integration issues raised by the operational integration of the diagnostic tool in SCC's or AIT environments have been addressed through a European project called UNITE, co-sponsored by the Commission of the European Communities. They are illustrated hereafter (figure 6).

Figure 6. Main Integration Issues explored in UNITE
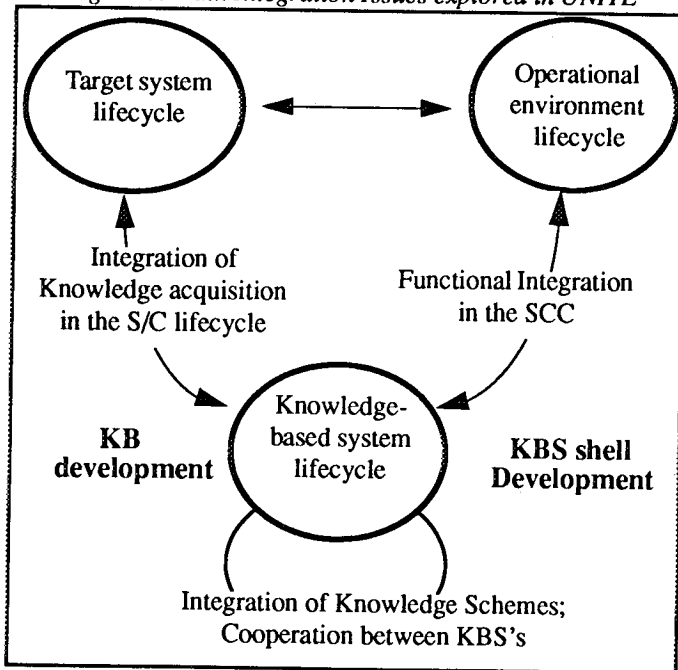
1) A first issue concerns the *integration of different knowledge schemes within a given KBS*. Diagnostic systems in Space indeed require the implementation

and integration of different knowledge representation and reasoning paradigms:

- they need to handle different domain models representing different views of the satellite system (e.g., thermal view, mechanical view, electrical view, etc.).
- the input information, be it provided by human users or by SCC monitoring facilities, is sometimes numeric but more often symbolic, intrinsically uncertain and imprecise, with a validity time frame.
- the basic inference mechanisms are themselves, e.g., exploiting uncertain and imprecise symbolic transfer functions (such as qualitative fault propagation functions) which may need to handle time to reflect the variation of dynamics between different views of the system.
- diagnostic reasoning deals with qualitative temporal propositions with a start, an end and a persistence.
- dependency tracking and maintenance of consistency between different reasoning contexts, or the management of the assumptions and time-constraints under which statements are valid, may require the parallel handling of several uncertain and time-dependent alternative hypotheses.

One of the goals is to give the knowledge engineer the flexibility to choose the most appropriate knowledge representation for some aspects of the problem (e.g., various representations of time and uncertainty), and yet process them in an integrated manner.

2) A second kind of need is concerned with the *sharing and exchange of knowledge between KBS's* that need to cooperate to achieve some global problem solving task. For instance monitoring, diagnostic and data analysis tools need to cooperate to detect and then locate the origin of anomalies. They may need to exchange knowledge or complex information. As the formalisms used to represent this information may vary from KBS to KBS, it is necessary to set up translation mechanisms, from the formalisms of each KBS to a common Knowledge Interchange Format and vice-versa. The approach followed by MMS in that domain is experimental. The goal being to assess the level of maturity and the applicability of existing solutions like those elaborated within the Knowledge Sharing Effort [12].

703

3) *Functional Integration* regards cooperation between the KBS and conventional software modules or database management systems for the construction of fully integrated operational applications. The methodology issues raised by the operational integration of the diagnostic tool in the SCC are investigated in [1]. Functional integration requires a hybrid methodology framework for co-existing conventional / knowledge-based developments.

4) Finally the DIAMS experience feedback has emphasized the importance of a better *integration of the knowledge capture tasks in the S/C lifecycle.*

Integration of knowledge models

The following figure provides a synthetic view of the different types of knowledge models explored through DIAMS-1 and DIAMS-2 and further refined and integrated in DIAMS-3 (figure 7).

Figure 7. Overview of DIAMS-3 Knowledge Models
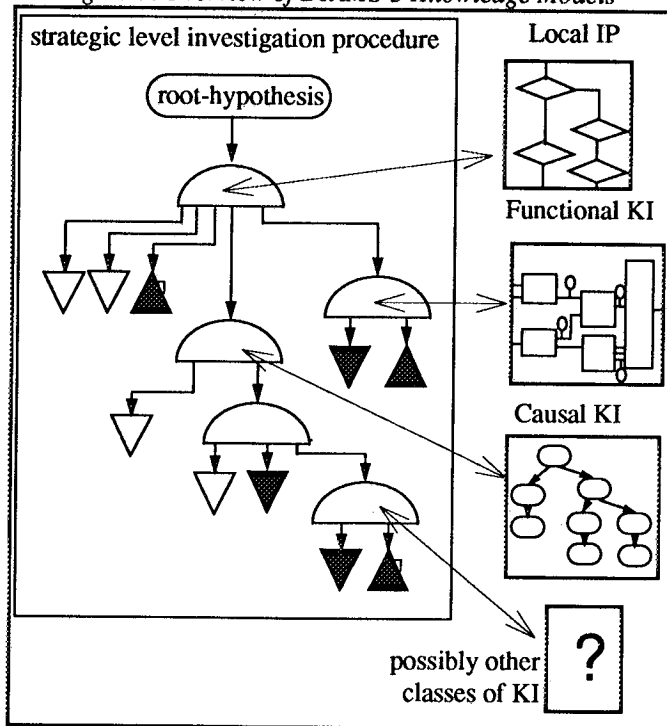
Causal KI

In the latest version of the tool, behavioral knowledge (also called *causal knowledge*) is composed of a reduced set of FMECA related to a family of symptoms, that allows to explore and refine some higher level hypothesis. This is a natural extension of the notion of behavior model explored in DIAMS-1.

Incompleteness is inherent to FMECA. A more flexible representation of the effects of fault modes has been proposed that eases expression of knowledge, down to the relevant level of detail (i.e., events chronologies), and that does not make any assumption about what is not said explicitly [6].

Handling of time, incompleteness and uncertainty

Some improvements brought by DIAMS-3 should allow to better handle time, incompleteness and uncertainty. Different techniques have been proposed for handling incompleteness, uncertainty or time-dependency. The investigation of the current practice shows that many difficulties in terms of performance or complexity have been experienced in deploying these techniques in industrial contexts and that ad hoc adaptations or simplifications are generally done by the development teams to match the industrial constraints. Beyond adequation to the specific knowledge representation and reasoning needs of the diagnostic tool, performance and complexity thus shall be the main criteria for the assessment of candidate solutions in that domain.

For instance, the information available about the symptoms is incomplete: many observables are not fully monitored in real time. Allowing the users to express their uncertainty about the interpretation of the observable was also recognized as a need. Indeed, some observations involve complex combination and abstraction of elementary pieces of data, followed by a high level interpretation of the result. Adequate formalisms are needed to handle incompleteness and allow expression of uncertainty about the presence/absence of a manifestation.

From a discrimination point of view, graduality in the uncertainty of the fault effects and in the characterization of the observables has been introduced. It allows a ranking of the solutions given by the system. As the diagnostic process is iterative, it was also found useful to have advice with respect to the selection of the next observables to be tested. This is achieved through a utility function that assesses the impact of the test of a manifestation on the possibility of fault mode.

Application developers will be provided with libraries of basic knowledge representation and reasoning mechanisms that can be easily included into application programs without imposing the use of

any particular development tool for the implementation phase. Considering the current trends in Information Technology, libraries of C++ objects seemed to be the best possible choice for DIAMS-3.

A first set of libraries of reasoning schemes have been selected, developed or re-developed in C++, and appropriately encapsulated to answer DIAMS needs:
- A new reasoning scheme which allows to represent and process incomplete and uncertain relations between faults and manifestations (such as FMECA) in a diagnostic context. The core model, based on the possibility theory, includes consistency-based and abductive diagnostic algorithms eploiting uncertain observations, as well as additional tools to measure the utility of tests and the discriminability of a set of fault modes [6]. Extensions of this model to the processing of functional knowledge are being developed.
- A Valuation Based System (VBS) which allows uncertain reasoning in a causal graph with various formalisms, e.g. bayesian, possibilistic, Dempster-Shafer's Theory of Belief, etc.
- A Time Constraint Propagator (TCP) which enables the comparison of an actually observed chronology of events with an a priori knowledge about the causal relationships between events. An hypothesis is confirmed by the TCP when all observed events occur at scheduled dates. If any of the observed events occurs outside the expected time window then the hypothesis is inconsistent and therefore is discarded. When the hypothesis-related events have not yet occurred - the hypothesis can be neither confirmed nor discarded - the hypothesis is said incomplete and TCP provides the validity interval for that hypothesis.

### Integration of reasoning schemes

The joint utilization of the TCP and VBS in a diagnostic context is illustrated by figure 8.

Sometimes such a (weak) integration approach may not be sufficient. Reasoning threads may be too intertwined to be processed efficiently in a separate way. A prototype has been developed to tackle this kind of problem and to evaluate the candidate technology. It addresses the so-called "strong integration" of temporal and uncertain reasoning in a model based diagnostic context. The computational approach consists in generating an ATMS network - Assumption-based Truth Maintenance System - to

compute explanations for symptoms. A possibilistic, temporal, cost-bounded ATMS machinery is used. The cost-bounded feature allows to focus of the reasoning process and to limit computational costs.

The main risk identified for strong integration is performance. The strong integration approach is currently considered as experimental and is not included in the DIAMS technical baseline.

Figure 8. Weak Integration of Reasoning Schemes
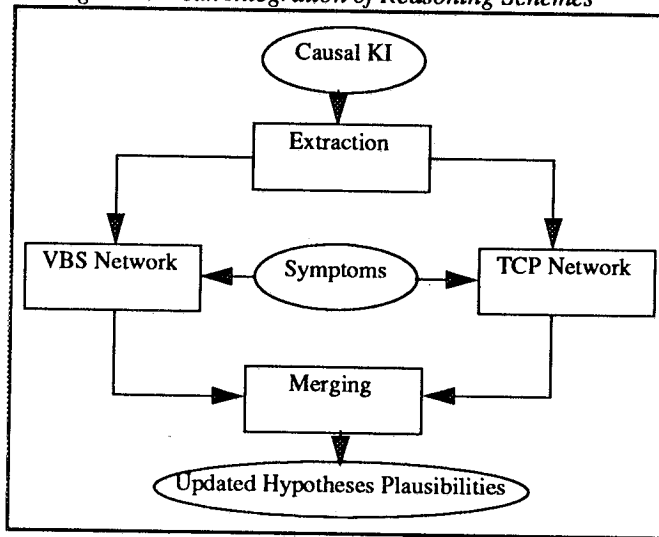
Integration of knowledge acquisition in the S/C lifecycle

The reduction of the knowledge acquisition costs was a permanent concern in each phase of the DIAMS program. A first conclusion was that, in order to improve the interactions with S/C specialists, the knowledge modelling activity should benefit to the S/C project tasks. The goal in DIAMS-3 is now to reach a level of expressiveness and genericity such that the DIAMS knowledge bases could be built and reused throughout the satellite lifecycle. This should contribute to significantly reduce the knowledge acquisition costs.

Current Projects   **Future Projects**

705

VI. Concluding Remarks

The DIAMS program followed a spiral approach, each cycle partially or fully implementing a reference development cycle. The DIAMS spiral lifecycle model is summarized in table 1. Matra Marconi Space is now involved in a tool improvement cycle (DIAMS-3) that would enable a tighter integration of the diagnostic system in ground infrastructures. A more general objective is to set up the techniques, methods and tools that will allow to consider the KBS technology as a baseline technology for the development of future S/C Control Centers or AIT Environments.

The knowledge acquisition issue remains pivotal. It comes down to the following two questions

- How to maximize the reuse of already formalized and managed knowledge?
- How to adapt the S/C project tasks and deliverables so that knowledge could be acquired 'on the fly' during S/C developments ?

A number of solutions have been proposed to proceed in this direction. The on-going experiments should prove that these solutions are ripe for introduction in S/C projects.

Acknowledgments

References

[1] AYACHE S., HAZIZA M., CAYRAC D.: "Delivering Spacecraft Control Centers with Embedded Knowledge-Based Systems: the Methodology Issue", Intl. Symposium on Space Mission Operations and Ground Data Systems, Greenbelt (USA), 1994

[2] BASTIEN-THIRY C., MAURIZE J.C.: "SE-TC2 : The first expert system in a CNES Satellite Control Center", ESA Workshop on Artificial Intelligence and KBS for Space, Noordwijk (NL), 1993.

[3] BRENOT J.M., CALOUD P., VALLUY L.: "The development of an operational expert system for the Telecom2 satellite control centre", ESA Workshop on Artificial Intelligence and KBS for Space, Noordwijk(NL), 1991.

[4] BRENOT J.M., CALOUD P., VALLUY L., GASQUET A. : "On the design and development choices to bring to operation a diagnostic expert system for the Telecom 2 satellite", Intl. Conf. on Fault Diagnosis, Toulouse (F),1993.

[5] CAYRAC D., HAZIZA H. : "Management of Uncertainty and Temporal Dependencies in Real World Diagnostic Systems, Application to the Space Domain", Intl. Conf. on Fault Diagnosis, Toulouse (F), 1993.

[6] CAYRAC D., DUBOIS D., HAZIZA M., PRADE H.: "Possibility Theory in Fault Mode Effects Analyses", IEEE World Congress on Computational Intelligence, Orlando (USA), 1994

[7] HAZIZA M.: "An Expert-System shell for satellite fault isolation based on structure and behaviour", Workshop on AI Applications to Space Projects, Noordwijk (NL), 1988.

[8] HAZIZA M., GIBET L., DEBAY P.: "DIAMS: An Expert-System shell for satellite fault isolation - The User Feedback", 3rd Intl. Conf. on Human-Machine Interactions and AI in Aeronautics and Space, Toulouse (F), 1990

[9] HAZIZA M.: "Towards an operational fault isolation expert system for french telecommunication satellite TELECOM 2", Intl. Symposium on Ground Data Systems for Spacecraft Control, Darmstadt (G), 1990

[10] HAZIZA M.: "Delivering Operational Space Applications with Embedded Knowledge-Based Systems", World Congress on Expert Systems, Lisbon (P), 1994

[11] NECHES R., FIKES R., FININ T. GRUBER T., PATIL R., SENATOR T., SWARTOUT W.: "Enabling Technology for Knowledge Sharing", AI Magazine, Fall 1991

Table 1. The DIAMS Spiral Lifecycle Model

Phase	DIAMS-0	DIAMS-1	DIAMS-2	DIAMS-3
Characterization in the large	X	(X)		(X)
Characterization in the small	X	(X)		(X)
Analysis	Test-Bed Implementation (Smalltalk)	X	(X)	(X)
Architectural Design		X	(X)	(X)
Detailed Design and Coding		Prototype Implementation Experimentation Phase	X	X
Verification & Validation			X	X
Operation & Maintenance			X	X