Delivering Spacecraft Control Centers with embedded Knowledge-Based Systems: the Methodology issue

S. Ayache , M. Haziza , D.Cayrac
Matra Marconi Space, 31, rue des Cosmonautes, 31077 Toulouse Cedex – FRANCE.
Tel: (33) 62 24 77 60, Fax: (33) 62 24 77 80
e-mail : ayache@soleil.matra-espace.fr

Abstract

Matra Marconi Space (MMS) occupies a leading place in Europe in the domain of satellite and space data processing systems. The maturity of the Knowledge-Based Systems (KBS) technology, the theoretical and practical experience acquired in the development of prototype, pre-operational and operational applications, make it possible today to consider the wide operational deployment of KBS's in space applications. In this perspective, MMS has to prepare the introduction of the new methods and support tools that will form the basis of the development of such systems. This paper introduces elements of the MMS methodology initiatives in the domain and the main rationale that motivated the approach. These initiatives develop along two main axes: knowledge engineering methods & tools, and a hybrid method approach for coexisting knowledge-based and conventional developments.

I. Introduction

Matra Marconi Space (MMS) occupies a leading place in Europe in the domain of satellite and space data processing systems. It has a long experience, as architect of both types of systems, in the integration of hardware and software components, man-machine interfaces, knowledge and data management systems, etc.

The development of methods and supporting environments is a part of MMS missions. MMS has a confirmed expertise in the domain of system engineering methods and tools. For instance, MMS has co-authored the HOOD design method (dedicated to the architectural and detailed design of large real-time and embedded Software applications) and is involved in the working group in charge of proposing evolutions of the method.

MMS has also acquired a theoretical and practical experience in the development of Knowledge-Based Systems (KBS) through numerous R&D, pre-operational and operational projects generally sponsored by CNES (the French space agency), ESA or other customers such as ARIANESPACE. The development activities conducted at MMS in the eighties have allowed to demonstrate the benefits of KBS to assist users in operation environments. That experience has also led to a robust in-house KBS development methodology.

It is now possible to consider the wide operational deployment of KBS's in space applications. In this perspective, MMS has to prepare the introduction of new methods and support tools that will form the basis of such systems development as well as their cooperation with more conventional methods [10]. After a brief description of the MMS approach in the field of space diagnostic support systems development, this paper develops the methodology issue that MMS is currently tackling and presents an experimentation of a hybrid method approach in the diagnostic systems field.

II. Space diagnostic support systems: the DIAMS programme

MMS has been investigating and experimenting spacecraft diagnostic support systems for eight years. The DIAMS concept, initiated in 1985, led to the development of a prototype expert system dedicated to the Telecom 1 Attitude and Orbit Control System [7] DIAMS-1, and to the present Telecom 2 Expert System [8], DIAMS-2, covering a whole satellite (platform and interfaces with the payload), which was installed in the Satellite Control Center at the beginning of 1993 [3].

One of the main advances realized through DIAMS-1 was the decomposition of the Knowledge base (KB) into different types of Knowledge Islands (KI) representing different domains of expertise.
DIAMS-1was implemented in Emicat (an object dialect on top of Prolog).

The next generation called DIAMS-2 was a near operational system developed on top of a KEE/ CommonLISP platform. It is a hybrid system combining decision-tree based symptom-hypothesis associational reasoning to initiate and to focus the diagnosis, and the DIAMS-1 model-based techniques to complete the diagnostic reasoning on particular functions and to provide the final isolation of the fault.

In DIAMS-2,comprehensiveness and efficiency was priviledged against fineness of representation and reasoning. Simplified representations well suited to the practical problems faced in space industry were introduced as a first approximation. A progressive refinement of the models and of the reasoning paradigms selected (for instance to include the handling of incompleteness, uncertainty and time) is now being considered in the definition of a new generation of knowledge based systems, DIAMS-3 [4],[5].

DIAMS-3 is being implemented in C++ and uses the ONTOS Object Oriented Database Management System for knowledge storage and retrieval. Beyond the porting into C++ of the DIAMS-2 machinery, DIAMS-3 will provide generic model edition services and C++ libraries of operational standard for handling time, incompleteness and uncertainty. These libraries could also be reused in other KBS development projects.

Other important objectives of DIAMS-3 concern tigher integration with other knowledge-based systems like data analysis or procedure management tools and more generally the complete integration of that kind of tools in the operational loop [11].

III. Methodology issues

Spacecraft Control Centers (SCC's) have to process large amounts of data from which the relevant information is generally difficult to extract and may require the use of KBS for instance for data analysis and diagnosis (such as those belonging to the DIAMS family). Knowledge-based planning and scheduling or procedures management tools can also be useful to master the management and execution of complex operational tasks. These different categories of KBS's generally need to communicate with the operational environment, i.e to exchange information with conventional software or databases. In addition, the embedding of the various kind of software components (including KBS's) into hardware and at a higher level into a system with its organizational logic has to be taken into consideration.

An example of typical Satellite Control Center functional architecture is provided in table1

Table 1. Typical SCC functional architecture

Core system and Common services	Databases, data storage and retrieval Time synchronization and distribution Local Area Network(s), communications Distributed environment monitoring and control Operation documentation management.
Procedural applications	Data reconstruction and distribution Flight dynamics monitoring and control Operation procedures construction and execution...
Knowledge-based applications	Data analysis Diagnosis Planning/Scheduling...

Various methods, tools, languages, models, or architectures are used to develop these different kindsof components. To give an example, in many SCC's development projects currently conducted at MMS, SADT and HOOD are used for the analysis and design of conventional software, and the MERISE Information System Design methodology (including Entity-Relationship diagrams) is used for the database components. The operational integration of KBS's in SCC's thus raises two kinds of methodology requirements:

- Knowledge engineering methods & tools:
  Well-suited methods and tools are required for expertise analysis and knowledge modelling, knowledge verification & validation, or KB Administration and Maintenance.
- Cooperation with conventional SW development approaches
  The elaboration of a methodology framework for the cooperation between knowledge engineering and SW engineering methods and tools is an essential requirement to guarantee the safe and efficient cooperation between KBS's and conventional applications within a same operational environment.

Rather than expecting the advent of the ultimate methodology that would allow to develop all types of system components within the same integrated methodology, a pragmatic solution, experimented by MMS, consists in adopting a hybrid method approach. In such an approach, the task of building the integrated application is carried out by developing all the system components within a methodology framework that allows the use of the most suitable existing methods in the successive phases of the development.

This approach of course requires to define correspondences between models for cross validation purposes but it carries a number of very interesting properties. For instance, it allows to benefit from the experience gained with the existing methods, allows to use existing tools supporting the methods, avoid problems such as compatibility with existing models (SCC's HOOD models for instance) or the costly training of a large number of people to a new method.

A hybrid method approach for KBS development grounded on KADS, HOOD and OMT has been successfully experimented by MMS through the development of the new generation of diagnostic support systems (DIAMS-3). This approach is detailed in the next section.

IV. The hybrid method approach experimented in DIAMS-3

1. Selected methods

Knowledge Engineering methods

The CommonKads method [14] which is now a knowledge engineering method rather popular in Europe supported by off-the-shelves tools has been selected as the DIAMS-3 Knowledge Engineering method. Its founding principle is Knowledge Level Modelling. The purpose of the knowledge-level model is to make the organization of knowledge in the system explicit independently of any representational issue (symbolic representation in terms of rules, frames, etc.) and, a fortiori, of any implementation level issue. **The CommonKads model set** is briefly presented in table 2:

Table 2. The CommonKads model set

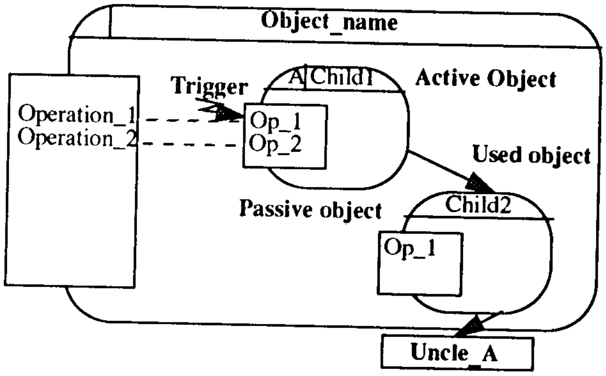
Organizational model	provides an analysis of the organizational environment in which the KBS will run
Task model	Descibes the real-life tasks executed in the organizational environment
Agent model	Describes the properties of agents that perform tasks specified in the task model
Communication model	Describes all transactions between agents
Expertise model	Organizes problem-solving knowledge in four layers: domain, inference, task and strategic knowledge

Software Engineering methods

Having assessed that the association of KADS with object-oriented analysis and design approaches could provide a suitable basis for developments of systems such as DIAMS-3, two complementary methods HOOD and OMT were selected:

- HOOD [12] is a design and development method for large technical and real time software systems. It resulted from the merging of Booch's Object oriented design approach and Abstract Machines methods. The definition of the method was sponsored by ESA and started in 86. Since its birth in 1986, HOOD has become the most commonly used design method in the european space industry. It is now the reference design method for the SW projects sponsored by the European Space Agency. HOOD is a hierarchical design method offering two kinds of interesting relations between objects: the "use" relation to express that one object requires the services of other objects and the "include" relation to express that one object, the parent, is fully implemented by the child objects it contains (cf Figure 1.)

Figure 1. HOOD object: graphical representation



- OMT (Object Modelling Technique) is an object-oriented software development method which extends from problem formulation and requirements analysis, to design and implementation. It has been defined by James Rumbaugh & al. [13] from the General Electric Research center (USA). This method proposes three kinds of models to describe the different views of a system (cf Table3)

Table 3. The OMT model set

Object model	Static, structural view of the system showing objects structure and relationships between them
Dynamic model	Temporal, behavioral view of the system
Functional model	Transformational, functional view of the system

The evaluation work has been focused on the object modelling technique from which the methods draws its name.

2. DIAMS-3 Specification

Two main kinds of output have been provided at the end of this phase:

- Software requirements (following a template close to the Software Requirements Document template recommended in the ESA PSS-05 standard [6]) including both functional and non functional requirements for the overall diagnostic tool.
- A CommonKADS Expertise model for the cognitive parts built with the support of the KadsTool tool. This model is briefly described in the next paragraphs:

Strategic knowledge

The KB is partitioned into knowledge islands (KI's). A KI contains all the knowledge items needed to investigate (i.e. confirm or infirm) some global hypotheses. A strategic-level Investigation Procedure is used to select a path among pending hypotheses and to navigate from KI to KI.

Domain knowledge

Domain knowledge is generally represented by hierarchies of concepts and relations between concepts. A *domain ontology* describes the terms that will be used to formulate statements about the application domain. Domain knowledge may further be specified with the help of some meta-descriptions - *model ontology* - that specify the type and structure of the domain models.

The diagnostic tool model ontology has been mainly represented by two "consist-of" hierarchies structuring:

- the satellite FDIR (Fault Detection and Isolation Recovery) static knowledge and
- the diagnostic session dynamic knowledge introduced as an example in Figure 2.

A complete description of domain knowledge may be found in [1].

Inference & task knowledge

The *inference knowledge* specifies the basic inferences that can be made with the domain knowledge.

The task knowledge describes the problem-solving tasks. Tasks are specified through a task definition and a task body. The task body decomposes the task recursively in terms of activities (other tasks) needed to achieve the task goal. A task description is generally associated to an inference structure and expresses a control flow on the inference structure.

The top-most inference structure and task description of the diagnostic tool Expertise Model are represented in Figure 3. and Figure 4.

3. DIAMS-3 Preliminary Design

HOOD and OMT have been used in a complementary way for preliminary design in the sense that:

- HOOD has mainly been used for the top down decomposition of the application into abstract machines and for an easy representation of interactions of the diagnostic system with external resources such as reasoning schemes. It supported the preliminary design of the diagnostic system shell.

Figure 2. Diagnostic session knowledge "consist_of" hierarchy

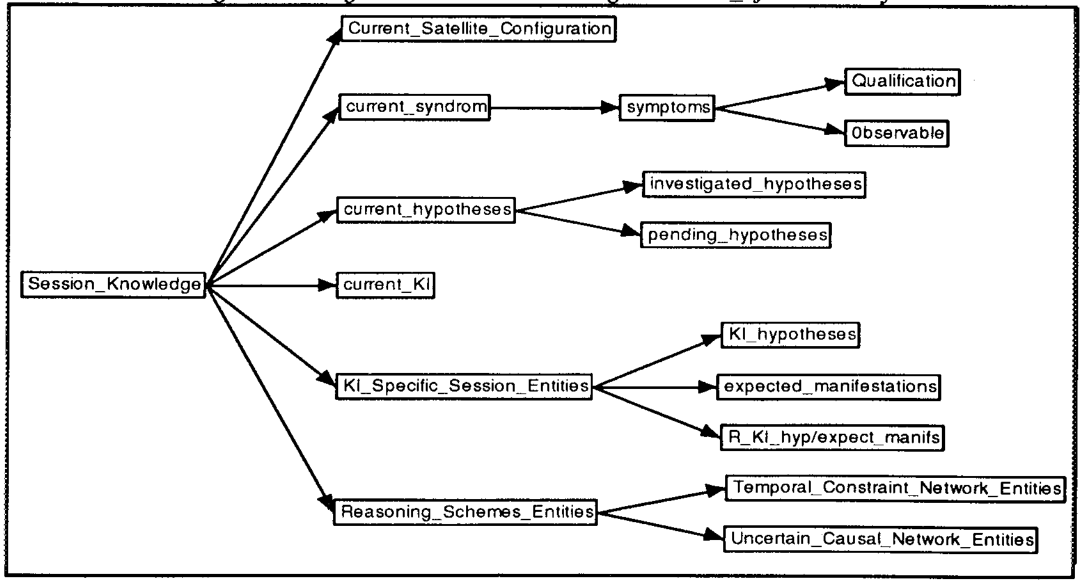


Figure 3. Diagnose Inference structure

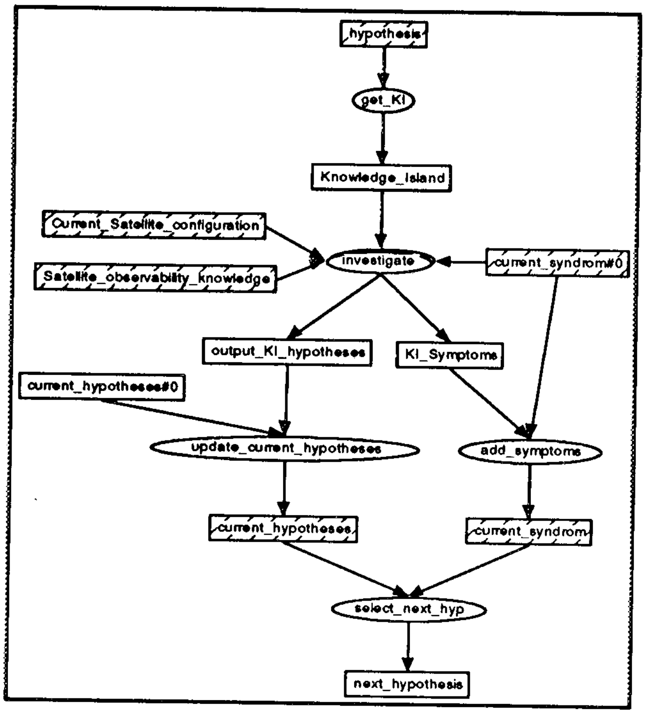


Figure 4. Diagnose Task description

Roles
Input
hi : Initial_hypothesis
Si : Initial Syndrom
Conf: Current_Satellite_Configuration
Obs : Satellite_observability_knowledge
Output
C : diagnostic conclusions
Sf : final syndrom
Control roles
KI : Knowledge_Island – in current investigation
H : Current Hypotheses
S : Current Syndrom
KI_hyp : output_KI_hypotheses – deduced from KI Investigation
KI_sym : KI_Symptoms – observed during KI_investigation
h : next_hypothesis to be investigated
Body
DIAGNOSE(hi,Si,Conf,Obs –> (C,Sf))=
S = Si, h = hi, H = { }
WHILE "select_next_hyp" returns an hypothesis
get_KI(h –> KI) – returns the KI associated to hypothesis h
INVESTIGATE(KI,S,Conf,Obs –> (KI_hyp,KI_sym))
Update_current_hypotheses(H,KI_hyp –> H) – add KI_hyp in H and update hypotheses plausibilities
Add_symptoms(S,KI_sym –> S)
Select_next_hyp (H,S –> h) – select next pending hypothesis h according to diagnostic focusing rules and set h status to "not pending";
END WHILE
C = H, Sf = S
Realization INVESTIGATE
Activates inference diagnose

- The OMT design process is not hierarchical but OMT offers a very powerful object modelling technique including of course modelling of inheritance. OMT has mainly been used to design the domain objects classes and relationships between these classes.

An example of HOOD object graphical description extracted from the documentation generated by the HOODNice tool is provided in Figure 5.

This description shows the decomposition of the object "Diagnoser" which is itself included (with other objects such as "KB_administrator" or "KB_interface") in the decomposition of the top level object called "Diagnostic_System". This figure shows "use" relations between Diagnoser internal objects and external objects (e.g., KB_interface) or objects belonging to the Diagnostic System Software environment (e.g., Temporal Constraint Propagator - TCP- and Valuation Based System -VBS- handling temporal and uncertain reasoning)

An example of OMT sheet extracted from the documentation generated by the OMTool tool is provided in Figure 6. This example shows a preliminary design model for KI_hypothesis and Knowledge_Island domain objects.

4. DIAMS-3 detailed design

Only OMT has been used to support the detailed design activity. This allowed a direct mapping to C++ object classes. OMT has also been used to maintain an up-to-date view of the detailed design model during the coding activity.

Classes identified in OMT preliminary design appear as ONTOS persistent classes in the detailed design model and methods corresponding either to administration methods or to basic inference mechanisms have been attached to these classes. An example of such a persistent class is provided in Figure 7.

Objects identified in the Diagnostic system shell HOOD preliminary design model appear as non-persistent classes in the detailed design model . An example of such a class is provided in Figure 8. In this case, services provided by the "Hypotheses manager" in preliminary design are dispatched in two classes: a semantic class used in the diagnostic process and a graphical class used to manage the Man-System dialog (its content has been masked to simplify the figure).

Figure 5. Diagnoser Hood Object graphical description

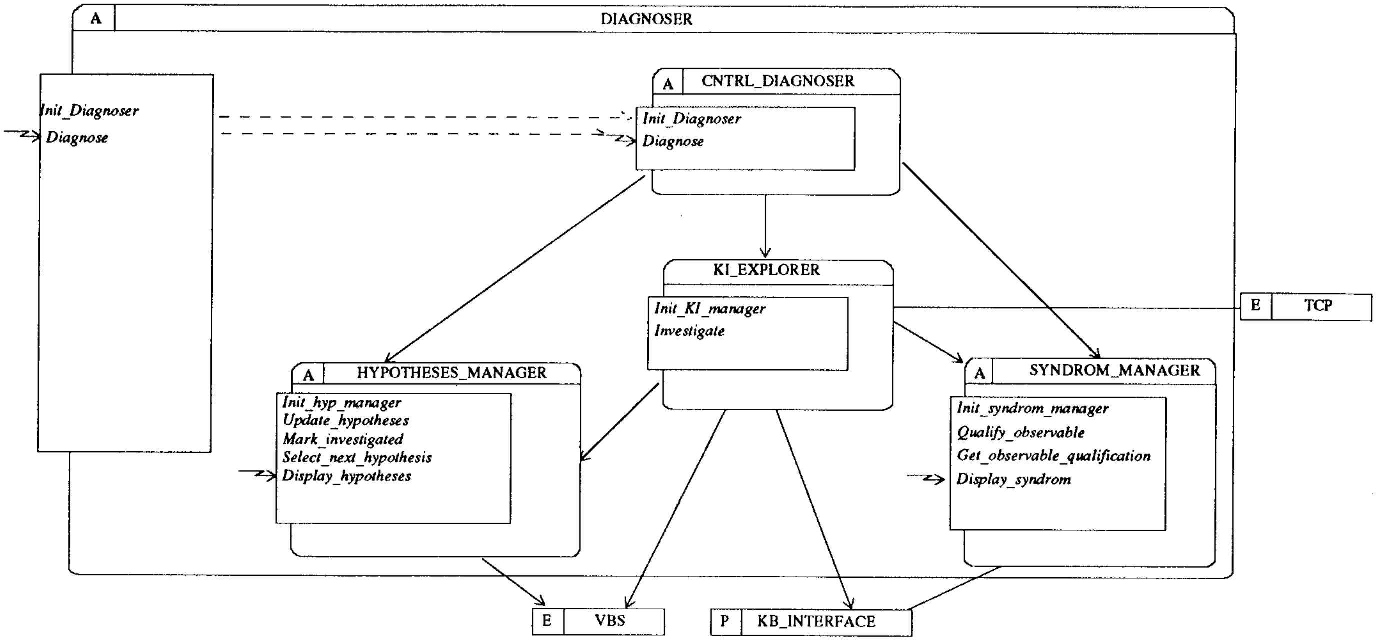


Figure 6. OMT sheet including KI_hypothesis subclasses and KI_hypothesis-KI relationships

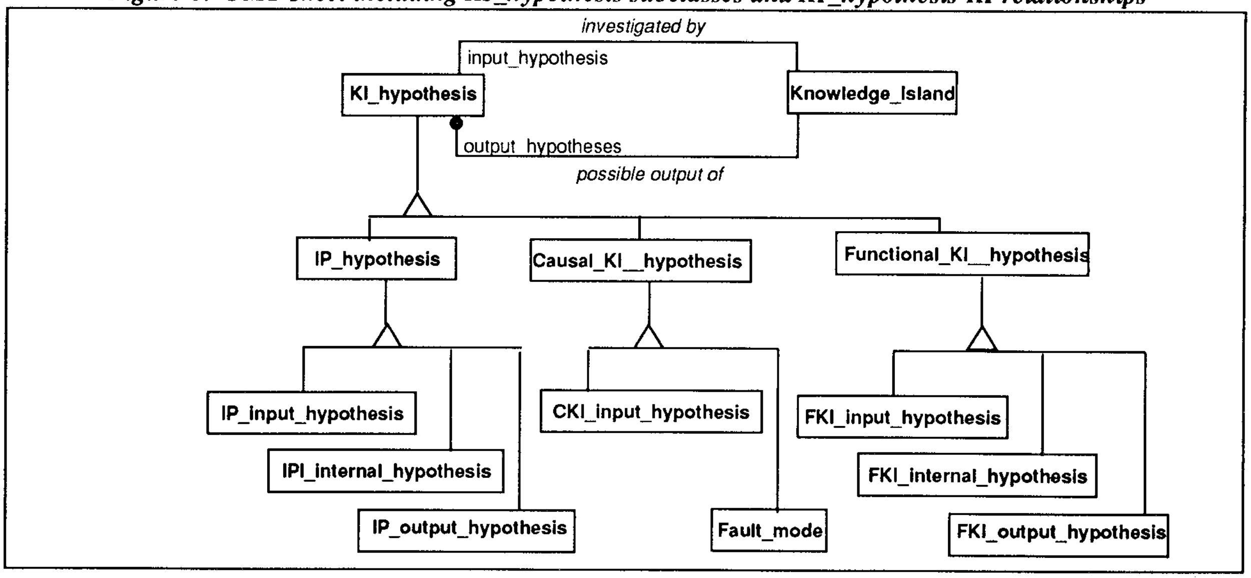
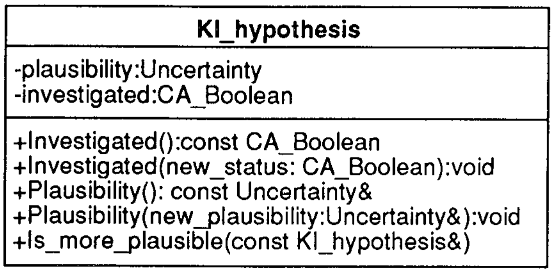


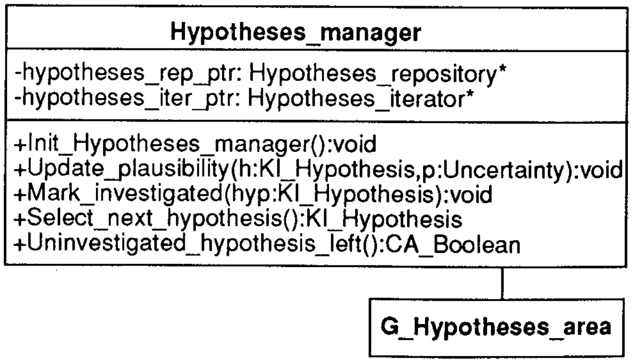
Figure 7. The KI_hypothesis class


KI_hypothesis

-plausibility:Uncertainty
-investigated:CA_Boolean

+Investigated():const CA_Boolean
+Investigated(new_status: CA_Boolean):void
+Plausibility(): const Uncertainty&
+Plausibility(new_plausibility:Uncertainty&):void
+Is_more_plausible(const KI_hypothesis&)

Figure 8. The Hypotheses_manager semantic and graphical classes


Hypotheses_manager

-hypotheses_rep_ptr: Hypotheses_repository*
-hypotheses_iter_ptr: Hypotheses_iterator*

+Init_Hypotheses_manager():void
+Update_plausibility(h:KI_Hypothesis,p:Uncertainty):void
+Mark_investigated(hyp:KI_Hypothesis):void
+Select_next_hypothesis():KI_Hypothesis
+Uninvestigated_hypothesis_left():CA_Boolean

G_Hypotheses_area

5. Experience Feedback

Each of the selected methods carries advantages and drawbacks. Taken as a whole, the set of selected methods exhibits complementary features allowing to progress in the elaboration of guidelines for selecting a lifecycle model and a combination of methods well-suited to a particular application project. This is further detailed hereafter.

Methods evaluation summary

CommonKADS

The CommonKADS modelling approach is mainly focused on the analysis phase and cannot be considered as a comprehensive methodology that provides guidance and support in all phases of operational KBS development projects.The application development experience showed that people with a practical experience in SW engineering got acquainted rather rapidly with the KADS approach.

The use of KADS allowed to establish a common universe of discourse over the project. KADS models were found very useful by the newcomers and eased their integration in the project team.

HOOD

The use of the HOOD method allowed the top-down decomposition of the application into modules. This provided a convenient basis for the specification of the man-system interfaces and the modelling of interactions with external resources (other KBS's, database systems or procedural applications). The HOOD modelling approach has been designed to facilitate the structuration of large projects. In the early phase of the application development, its use indeed simplified the task sharing between team members

However the main drawback of the method resides in its lack of support for the modelling of inheritance, which is a critical requirement when developing KBS, and, correlatively, the absence of C++ code generator in the tools that support the method. This feature prevented the selection of HOOD as the application detailed design method.

OMT

OMT offers a powerful object modelling technique which turned out to be well adapted for the preliminary design of classes corresponding to domain objects and for the detailed design of the whole application. In addition the support tool used allowed to generate C++ code skeletons based on the OMT object model components.

Among the methods investigated, OMT is probably the one which is the closest to the ideal comprehensive methodology that could be applied to all kind of system components - KBS's, conventional applications, database applications, etc. - in all phases of integrated systems lifecycle. Notice for example that MMS is using OMT for two KBS projects: "Architectural concept for Spacecraft Operations Automation" (sponsored by ESA/ESOC) which aims at integrationg various KBS(procedures management, data analysis, planning/scheduling) within the current ESOC control center (SCOS) and "Ogre", a KBS for ARIANE5 tests data analysis and reports generation (sponsored by CNES). However the method is still rather young - support tools of industrial standard are only emerging - and not widely used for operational system developments in space. Notice also that in Europe, ADA remains the reference language for real-time systems developments and that HOOD will probably remain the reference method for such developments for a few years still.

V. A hybrid methodology framework for co-existing conventional/knowledge-based developments

The method cooperation approach straightforwardly derives from the operational continuity principle. This requirement states that as organizations are hard to change, and as old applications and organizations have to be maintained while introducing new system capabilities, it is important that applications be developed on a modular basis to enable an incremental development and maintenance strategy.

This principle at the application level translates into a dual principle at the methodological level that could express as follows: when people have a good working knowledge of a given method that has proved to be well-suited to a given class of system components it is preferable to let them use the known methods and to limit the enforcement of new methods to system components and development phases which are not well covered with the existing methods.

Rather than developing a comprehensive methodology, the proposed approach is thus to define a framework that supports the cooperation between methods.

Table 4 introduces a first instance of such an hybrid approach that synthesizes the main results of the method evaluation work as well as other results coming from a comparison of KADS, MERISE, SADT and OMT methods [9]. This table associates a set of methods or languages to each lifecycle phase. Such sets of methods can be interpreted either as alternatives methods (e.g., KADS/ OMT for domain objects modelling) or complementary methods (e.g., HOOD/OMT for preliminary design) or as possible mappings between models for cross_validation purposes (e.g., KADS/MERISE where KADS is used for Knowledge_based components and MERISE for SCC operational databases).

The method cooperation approach also requires to manage the correspondence between different representations of the same objects at each step of the development process. This is particularly needed for objects encapsulating knowledge & data exchange services between different subsystems and to perform the cross-validation of models. This question has also been investigated in [9]

Table 4. Method components for operational integration of KBS's in space environments

Lifecycle phase	Data / Domain Objects	Functions / Tasks / Inferences
Analysis	KADS, MERISE, OMT	KADS, SADT, OMT
Preliminary design	OMT	HOOD, OMT
Detailed Design	OMT	OMT
Coding	C++	C++

Conclusion

In this paper, we have presented a hybrid methodology framework that could contribute to the operational integration of KBS's in SCC's as this has been demonstrated on the example of diagnostic support systems.

Experience feedback coming from MMS current KBS projects using OMT for the whole lifecycle will also provide valuable inputs for assessing this hybrid methodology framework.

Further goals for MMS in this area are to refine the proposed hybrid approach through elaboration of rules for the maintenance and updating of hybrid models in the coding phase (including the management of traceability links). The situation of prototyping and V&V activities wrt. the proposed hybrid approach are also being investigated.

Acknowledgments

Work related to some of the topics developed in this paper was partially funded by CEC through UNITE project (ESPRIT project 6083). Other partners involved are: Cap Gemini Innovation(France), Queen Mary Westfield College(UK), Sintef Delab(Norway), Eritel (Spain), ITMI (France).

References

[1] S.Ayache, P.Caloud, D.Cayrac, M.Haziza "MMS Component Application Specification and Design"- UNITE deliverable, November 1993

[2] S.Ayache, P. Caloud, D.Cayrac, M. Haziza "MMS Integrated Application Specification and Design"- UNITE deliverable, November 1993

[3] J.M.Brenot, P.Caloud., L.Valluy, A.Gasquet: "On the design and development choices to bring to operation a diagnostic expert system for the Telecom 2 satellite", Proc. Tooldiag Int. Conf. on Fault Diagnosis, Toulouse, France, 1993.

[4] D. Cayrac, M. Haziza: "Management of Uncertainty and Temporal Dependencies in Real World Diagnostic Systems, Application to the Space Domain", Proc. Tooldiag Int. Conf. on Fault Diagnosis, Toulouse, France, 1993.

[5] D. Cayrac, D.Dubois, M.Haziza, H. Prade.: "Relational Diagnosis Based on a Functional Model", Proc. AIENG conf., Toulouse, France, 1993.

[6] ESA PSS-O5-0 standard Issue 2 (February 1991)

[7] M.Haziza "An Expert System Shell for Satellite Fault Isolation based on Structure and Behavior", in Proc. ESTEC Workshop on Artificial Intelligence and Knowledge Based Systems for Space, Noodwijk, Netherlands, 1988.

[8] M.Haziza "Towards an Operational Fault Isolation Expert System for French Telecommunication Satellite Telecom 2": Proc. ESA Symposium "Ground data systems for spacecraft control", Darmstadt, FRG, 1990.

[9] M.Haziza, S.Ayache, D.Cayrac, P.Y.Lambolez: "MMS first report on the use of selected methods for analysis and design" - UNITE deliverable, November 1993

[10] M.Haziza: "Delivering operational space applications with embedded Knowledge_based systems", World congress on Expert Systems, Lisbon, Portugal, 1994.

[11] M.Haziza, S.Ayache, J.M. Brenot, D.Cayrac, D.-P Vo: "Diams Revisited: Taming the Variety of Knowledge in Fault Diagnosis Expert Systems", to appear in proceedings of SPACEOPS'94 (International Symposium on Space Mission Operations and Ground Data Systems), USA, November 1994.

[12] B. Labreuille, JF. Muller, HOOD: a design method for the space industry, AIAA/NASA Symposium on Space Information Systems, 1990

[13] J. Rumbaugh et al: "*Object Oriented Modelling and Design*", Prentice Hall International, 1991.

[14] G. Schreiber et al: "*KADS: a principled approach to Knowledge Based System Development*", Academic Press, 1993