Integrated Diagnostic System (IDS) for Aircraft Fleet Maintenance

Michael Lehane, François Dubé, Michael Halasz, Robert Orchard, Rob Wylie, Marvin Zaluski

Integrated Reasoning Group
National Research Council of Canada
Institute for Information Technology
M50 Montreal Road, Ottawa, Ontario, Canada K1A 0R6
http://ai.iit.nrc.ca/IR_public
mlehane@ai.iit.nrc.ca

Abstract

The aim of the Integrated Diagnostic System (IDS) project is to research, develop and test advanced diagnostic and decision support tools for maintenance of complex machinery. This paper provides an overview of the hybrid reasoning conducted within this system with particular reference to Case-Based Reasoning (CBR) and its integration in this environment. The technical development of this system is outlined as well as an operational review of the fielded prototype. Furthermore, the development of a distributed, generic architecture for this system to aid its potential widespread implementation is introduced.

Introduction

The Integrated Diagnostic System (IDS) Prototype, Version 2.11 is an applied AI system used to diagnose problems and help manage repair processes of commercial aircraft fleets. For more information on the motivations for, and methodology followed in this project, as well as technical issues covered in the development of this version of the prototype see (Wylie et al. 1997).

Briefly, IDS refines an asynchronous stream of messages consisting of symptoms and repair actions into descriptions of complete fault-repair episodes. The process exploits many knowledge sources, some allowing messages to aggregate, others allowing messages (or messages clusters) to merge, be modified or discarded. The ideal result is clear, concise, complete descriptions of fault events, which unambiguously associate symptoms with appropriate repair actions (Wylie et al. 1997). IDS was built using ART*Enterprise® (A*E), Version 2.0 and makes extensive use of its rule-based and Case-Based Reasoning (CBR) facilities in order to apply various sources of knowledge (manuals, heuristic, historical data) to this problem.

IDS - Operating Principles

The system starts by cleaning up and classifying (using CBR) the message stream from the aircraft to produce IDS Message Objects (IMOs).

These IMOs are then clustered into Fault Event Objects (FEOs). This clustering is conducted using heuristics gathered from engineers and maintenance technicians. FEOs take input from the Troubleshooting Manual (TSM) objects and Minimum Equipment List (MEL) objects. The TSM objects represent clusters of IMOs, which are identified in the TSM as indicative of particular faults. Similarly, the MEL objects represent clusters of IMOs, which are identified in the MEL manual as indicating that for safety, the operation of the aircraft is restricted in some way.

These symptoms (i.e. message clusters in the FEOs) are then associated with appropriate repair actions. This process, exploits both rule based and case-based reasoning. The resulting Snag[1] Rectification Object (SRO) is then stored.

In the final stage of the process, suggested repair actions are composed and presented to the user. These are derived from historical maintenance events appearing similar to a current FEO (using CBR) and from the Troubleshooting Manual (if the FEO contains a TSM object).

Case-Based Reasoning within IDS

There are two CBR components within IDS v2.11 (Figure 1.0), firstly CBR facilities the retrieval of relevant knowledge from bodies of noisy, poorly structured and incomplete historical information. Secondly, it is being used in the creation of a corporate memory (Leake 1996) of diagnostic repair information for use within IDS and Air Canada.

[1] A snag is a commonly used aviation term for an equipment "problem".

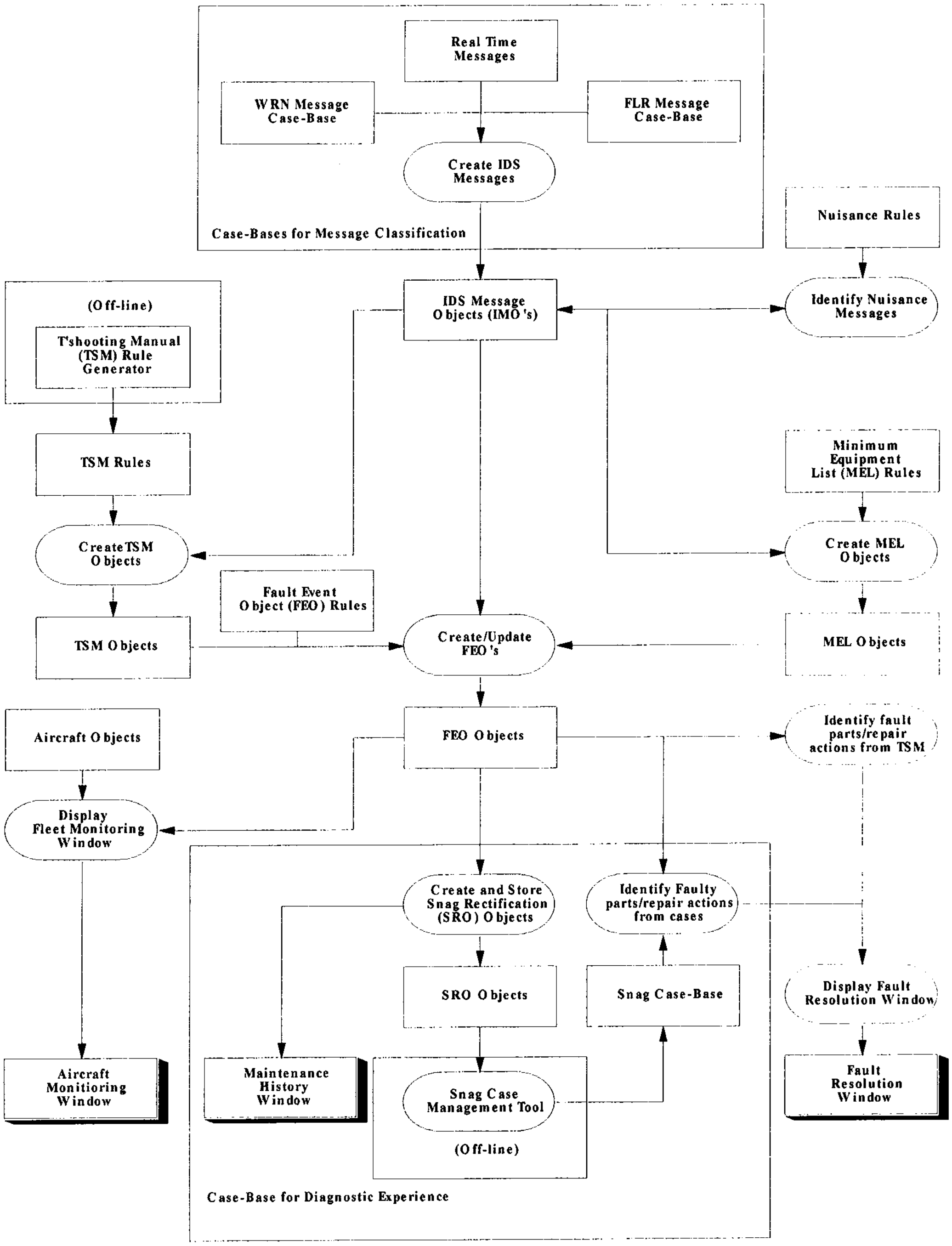

Figure 1.0 IDS V2.11 Data flow diagram.

Case-Bases for Message Classification

The Airbus A320 aircraft on-board diagnostic routines generate two types of messages, namely failures (FLR) and warnings (WRN) messages. In total there are about 3400 FLR and 560 WRN cases representing these messages. Messages consist of text and an ATA number; these describe the aircraft components in a hierarchical manner. Messages received from the aircraft cannot be recognized using simply string matching as they may be distorted during transmission. To overcome this, two case-bases have been created for all the FLR and WRN messages. As messages are received the strings and ATA values are matched using trigram matching against the appropriate case-base and associate a unique identifier with the message object that is created. A threshold for matching allows poor matches to be flagged. Occasionally, messages are received that are not in the case-base; these are investigated to see if this is a valid message that needs to be added to the case base.

A*E's CBR tools are being used here to provide the low-level inexact matching best described as an unusual, implementation of an episodic memory and one where it is difficult to justify the use of the acronym "CBR" (Wylie 1998), (Watson 1997). While the use of a case-base to identify messages and assign unique identifiers is not a standard use of CBR this implementation has proven to be robust in matching messages.

Case-Base for Diagnostic Experience

This case-base is usable as a corporate memory allowing retrieval of historical situations, which appear similar to the current situation, providing a means by which maintenance technical experts can feed their knowledge back into line maintenance system.

Presently, the case-base is managed through an off-line facility (the Snag Case Management Tool). This application allows the user to browse the SRO database, clean up the contents of an SRO, convert SROs into cases, test a new case against the existing case-base (for redundancy and consistency) and add it to the case-base. On-line, the snag case-base is searched each time a FEO is selected by the user. If a case is found which has similar symptoms (clusters of IMOs) then it is retrieved. From the retrieved cases, repair actions are extracted and used to suggest courses of action to the user. The tight integration of this type of CBR mechanism into a large hybrid reasoning system, makes IDS interesting. At the moment, effort is focused on precisely what diagnostic role the Snag case-base should be playing in IDS, for example should it be refining diagnoses made using TSM knowledge or should it be catching faults missed by the TSM. In addition, how should the CBR system work from an organizational perspective with respect to case creation and validation policies (Kitano and Shimazu 1996) and what impact does the CBR system have on communications between the technical maintenance staff and the line maintenance staff?

Evaluation of IDS

The IDS v2.11 prototype has been on extended field trials at Air Canada for 9 months. This has lead to considerable use of the prototype and generated valuable feedback. During this time the prototype development has been reviewed with the aim of assessing the execution of IDS v2.11 and highlighting future research issues (Dubé & Wylie 1997). Technically, IDS v2.11 demonstrated that hybrid AI reasoning systems are practical and can be built using currently available tools, but development can require a degree of customization to provide acceptable results. The use of appropriate tools is critical in building such applied AI systems. A*E reduced the effort involved in the development of IDS but because of the time constrained, asynchronous nature of the system, it is not a typical application of A*E, this stretched the envelope of the development environment.

The development of IDS has confirmed the belief that to be useful as a decision making tool, systems must be closely coupled to the organization's underlying information flows. A corollary to this is that development tools must provide good data integration to existing applications.

Meanwhile, feedback from Air Canada has indicated that an "intelligent" application automatically collecting, grouping and assessing sets of fault symptoms and automatically alerting maintenance personnel is an asset. The ability to automatically refer the user to the pertinent maintenance manual pages to support the fault resolution process is also extremely useful.

Overall, recommendations were for a system incorporating more intelligence and one that integrated with most of the information systems related to the maintenance operation, this is one of the goals of IDS-98.

IDS-98: A Distributed, Generic Maintenance Management System

The goal of IDS-98 is to create a new version of the IDS software, which builds in an "evolutionary way" (McConnell 1996) on the functionality of IDS v2.11, but has a distributed architecture and reflects a generic maintenance management system that could be applied to other application fields. In addition, IDS-98 will constitute a flexible infrastructure for integrated reasoning research including "operationalizing" of structured technical documents, CBR and data mining (Wylie 1998).

Initial attempts at distributing IDS v2.11 involved duplicating the asynchronous message feeds as well as the entire IDS application for each instance of the application. This meant that each IDS application did the same inferencing and database manipulation, this was not an ideal solution for a system that will potentially be installed across Air Canada's maintenance operations. To overcome this IDS-98 has a distributed architecture making use of ActiveWeb® a Message-Oriented Middleware (MOM) product from Active Systems. MOM refers to the process of distributing data and control through the exchange of messages (Orchard 1997). ActiveWeb does this defining a standard unit for capturing and exchanging information called an "event". At the heart of the ActiveWeb system is a "broker" which handles the distribution of events. The new architecture of IDS-98 presently consists of an IDS v2.11 application connected via a Java® interface to an ActiveWeb broker. Messages from the aircraft are sent to the IDS application where the inference is conducted and then the resulting information is distributed to lightweight remote Java applications. This work is currently in development, but the goal of IDS–98 is to adopt this architecture, to which other modules of the IDS-98 system will publish and subscribe to events.

New Modules in IDS-98

The basic functionality of IDS-98 will build on IDS v2.11 but will include several additions, integration of multiple reasoning techniques, mobile implementation, extension to other fleets, additional CBR functionality and a data mining and trending module (Létourneau, Matwin and Famili 1998). From a CBR perspective the main focus will be the enhancement of CBR functionality. This work will initially focus on the two aspects of the current tool, firstly user requirements and a task analysis exercise, and secondly case attribute definition and representation. The re-evaluation of the case attributes is in response to suggestions from Air Canada that a richer set of attributes may be needed for case representation, this is a critical area of CBR implementation (Lehane 1997).

In addition, the functionality of the existing case base management tool will be enchanced by providing several additional features. Presently, reoccurring symptom sets, which lead to the creation of FEOs, are not highlighted, within IDS-98 such FEOs will be highlighted graphically within the case creation tool. This will allow the user to track reoccurring problems more easily. The second addition will allow case creators to assess the benefit of adding cases to the case base. At the moment, before a case is added to the case-base a user can check for redundancy and consistency. An instrumentation module has been developed which tracks the frequency of case retrieval allowing an assessment of case usefulness. The integration of this module within the case base management tool would allow evaluation of a new case against a historical subset of data to check for instances of retrieval. It is also intended that a version of this instrumentation will run in the background and track the frequency of case retrieval to facilitate evaluation of the case-base. The final development component is the creation of a case-base browser to allow viewing of stored cases.

Conclusions

The development of IDS has demonstrated both the importance and viability of an integrated decision making system for the maintenance of complex fleets such as aircraft. The evolution of IDS from v2.11, through to IDS-98 underlines the intent within the Integrated Reasoning group to continue this development in an evolutionary way adding technical functionality that provides measurable utility and advances the use of integrated hybrid reasoning systems.

Acknowledgements

Many people are involved in this project, but special thanks go to the following for their continued support. From Air Canada: Robert Ferland, Robert Parker, Dave Clarke, Marie-Jo Leconte (IDS Production Team), Joe Mallory (Information Technology), Ron Blue (Fleet Management/MTS), Colin Drennan, Henry Dungan, Gerry Richardson (MTS), and Chris Sowerby (Propulsion Engineering). From NRC: George Forester (Business Development).

References

Wylie, R., Orchard, R., Halasz, M., and Dubé, F. 1997. IDS: Improving Aircraft Fleet Maintenance. In Proceedings of the Fourteenth National Conference on Artificial Intelligence, 1078-1085. Menlo Park, Calif.: American Association for Artificial Intelligence.

Leake, D. B. (ed.) 1996. *Case-Based Reasoning: Experiences, Lessons & Future Directions.* Cambridge, MA: AAAI Press/MIT Press.

Wylie, R. 1998. Integrated Diagnostic System (IDS) Strategic Review, Technical Report, Institute for Information Technology, National Research Council of Canada.

Watson, I. 1997. *Applying Case-Based Reasoning: Techniques for Enterprise Systems.* San Mateo, Calif.: Morgan Kaufmann Publishers.

Kitano, H., and Shimazu, H. 1996. The Experience-Sharing Architecture: A Case Study in Corporate-Wide Case-Based Software Quality Control. In *Case-Based Reasoning: Experiences, Lessons & Future Directions, 235-268,* edited by D. B. Leake. Cambridge, MA: AAAI Press/MIT Press.

Dubé, F., and Wylie, R. 1997. Integrated Diagnostic System (IDS) Prototype Evaluation Report, Technical Report, Institute for Information Technology, National Research Council of Canada.

McConnell, S. 1996. Rapid Development: Taming Wild Software Schedules. Microsoft Press.

Orchard, R. 1997. Distributed Computing: Current Directions - The Middleware Explosion, Technical Report, Institute for Information Technology, National Research Council of Canada.

Létourneau, S., Matwin, S. and Famili, F. 1998. A Normalization Method for Conceptual Data: Experience From a Large-Scale Application. Forthcoming in the Proceedings of the Tenth European Conference on Machine Learning.

Lehane, M. S. 1997. The use of Case-Based Reasoning for Decision Support in Engineering Design. Ph.D. diss., Dept. of Civil Engineering, Cardiff University, Wales, UK.

Appendix A: Case-Based Reasoning Integration

1. Integration name/category: Integrated Diagnostic System (IDS). There are two CBR components within this system, Case-Bases for Message Classification & Case-Base for Diagnostic Experience.

2. Performance Task: Case-Bases for Message Classification – Retrieval and classification of messages to control system input. Case-Base for Diagnostic Experience – Retrieval for past experiential knowledge to aid decision-making.

3. Integration Objective: The objective of this system was to create an integrated solution, which closely aligned itself to the existing organizational information flows. The Case-Base for Diagnostic Experience provides feedback and learning within an otherwise continuous stream of messages, which are classified by the two message classification case-bases.

4. Reasoning Components: Rule Base Reasoning (RBR) and CBR.

5. Control Architecture: Sequential, system architecture is built around the information flow, which uses both RBR and CBR.

6. CBR Cycle Step(s) Supported: Case-Bases for Message Classification – Pre-processing and Retrieval. Case-Base for Diagnostic Experience – Retrieval, Reuse, Revision and Retention.

7. Representations: Cases and rules are used extensively within the system.

8. Additional Reasoning Components: User Interaction for case creation and addition to case-base of diagnostic experience.

9. Integration Status: Applied.

10. Priority future work: Development of additional functionality and on-going system evaluation in applied environment.