Université d'Ottawa • University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Marvin ZALUSKI

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

Master of Computer Science

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Case Authoring from Text and Historical Experiences

N. Japkowicz

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

S. Matwin

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

L. Peyton

F. Oppacher

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

CASE AUTHORING FROM TEXT AND HISTORICAL EXPERIENCES

Marvin Zaluski

Thesis

submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the degree of Master of Computer Science

November, 2003

Ottawa-Carleton Institute for Computer Science
School of Information Technology and Engineering
University of Ottawa

Your file Votre référence
ISBN: 0-612-90362-1
Our file Notre référence
ISBN: 0-612-90362-1

To my parents for all their support.

Abstract

Knowledge management is the acquisition and distribution of knowledge within organizations. The knowledge acquisition bottleneck problem is an obstacle in implementing knowledge management system. This thesis describes a solution to this problem through a semi-automated approach that sorts through the mass of information that is available within the organization. Our tool was developed using state-of-the-art Case-Based Reasoning and Information Extraction technologies. More specifically, we developed a semi-automated case authoring method that creates a case-base in two steps. It begins by extracting knowledge from readily available resources such as technical documents and follows by complementing those cases using individual experiences in the organization. The case-base developed is a reflection of the knowledge encoded in the technical documentation and an authentication of the cases with real historical instances. Our case authoring approach is applied to the real world in the aerospace domain.

Acknowledgements

My thesis required the interactions between different organizations and people and no single person could have completed this task without help. This collaborative environment supported the research that focussed on the knowledge acquisition for Case-Based Reasoning or case authoring. The completion of the thesis would not have been possible without the help and support from the following people and organizations.

First, I would like to acknowledge the time and guidance of my two thesis supervisors, Dr. Stan Matwin and Dr. Nathalie Japkowicz. Without their assistance, this thesis would not have been possible. Their insight and guidance were invaluable when I reached obstacles that seemed too difficult. They challenged my ideas and identified problems, which either required explanation or resolution. They persuaded me to accept challenges that would have seemed daunting at first appearance. Finally, they helped my writing through their comments and suggestions, which I am grateful for. They provided me with the academic support that I needed to complete my thesis.

Secondly, I would like to recognize the National Research Council for the financial funding and resources that were influential in the completion of my thesis. As a valued employee, the NRC provided the financial support for both my research and education. Also, NRC provided me with machine and knowledge resources that were influential in the realization of my research with knowledge from structured documents.

Thirdly, I would like to thank Air Canada for the technical documentation and data that is used in the maintenance of their aircraft. Also, I would like to thank the staff for their helpful interpretation of the technical documentation and data. These technical documents and data were influential in the development of a proof-of-concept application that was based on the case authoring approach in the thesis.

Finally, I would like to acknowledge Syncrude Canada for technical documentation and data that was used to determine similarities between the aerospace and open-pit maintenance

organizations. This assessment helped estimate the adaptability of the case authoring approach to other problem domains such as open-pit mining. This information was used to identify processes within the case authoring approach that could be reused in another application.

Finally, I would also like to thank the members of my committee:

- Dr. Franz Oppacher, Carleton University
- Dr. Liam Peyton, University of Ottawa
- Dr. Stan Matwin (thesis supervisor) University of Ottawa
- Dr. Nathalie (thesis supervisor), University of Ottawa

Without the help from the above-acknowledged sources, my research into this new case authoring approach would have not become a reality. This type of success requires the interaction between different people and organizations. I am grateful to experience this type of collaboration and the effect it had on the completion of my thesis.

Table of Contents

Figures

Tables

1 Introduction

Knowledge has become not only useful in everyday operations of an organization, but it has become essential to the survival of organizations. Globalization has affected organizations around the world in ways never previously imagined. With the penetration of technology, the cost of doing international business has dropped considerably. The Internet has reduced the cost of communication around the world substantially. Organizations can enter the global marketplace by simply constructing a web page, publishing it, and registering it with an Internet search engine. Amazon.com became a top competitor in the selling books by creating a store in cyberspace that is open 24/7 and adapting itself for each of the individual customer's needs [Freidman 2000]. These lower costs for entering the global marketplace and becoming an instant competitor has allowed the competitive advantages gained from other more traditional methods to disappear [Stewart 2001]. Therefore, organizations are faced with more fierce competition resulting from competitors that may not have been previously considered.

Since traditional sources for competitive advantage are becoming more insignificant and unobtainable, knowledge within the organization is being identified as a source to regain/retain this competitive advantage [Stewart 2001]. In 1999, America's most valuable export was knowledge from licensing and royalty fees from intellectual property [Stewart 2001]. The demand for physical labour is decreasing while knowledge intensive labour is increasing. Peter Drucker was the first person to identify this new type of economy based on knowledge intensive work and defined it as the knowledge-based economy [Drucker 2001]. Since knowledge is considered to be universally accessible, the competitive nature of the knowledge-based society will be the fiercest that the history has ever seen [Drucker 2001]. Therefore organizations must invest and nurture their intellectual capital to increase and sustain their competitive advantage in a global community.

Intellectual capital is the knowledge that transforms raw materials into something more valuable [Stewart 2001]. Sometimes the raw materials are not physical, but conceptual; for

example, information. This realization of intellectual capital has changed organizations' business models and modes of operations. For example, General Electric's CEO identified that his organization could profit from the sale of knowledge intensive services for their products, such as maintenance for aircraft engines [Stewart 2001]. Knowledge increases the value of your product thus knowledge should become more important asset in organizations [Stewart 2001]. Currently, organizations know knowledge is useful and valuable, but no generic method exists for capturing and disseminating knowledge throughout the organization.

In order for knowledge processes to be effective in organizations, two obstacles must be overcome. These two obstacles are knowledge acquisition and knowledge distribution [Stewart 2001]. The first obstacle of knowledge acquisition is the identification and definition of knowledge. Since knowledge and information have a dynamic and interactive relationship with each other, it is sometimes difficult to differentiate between knowledge management and information management systems [Watson 2001]. Therefore, a clear definition of knowledge is needed within the organization. The organization's definition of knowledge is influential in determining the sources of information that produce knowledge. The understanding of these information resources and how they contribute to further knowledge are essential in the acquisition and representation of knowledge in the organization. Once this is accomplished, processes can be developed to manage the distribution the new knowledge.

The second identified challenge is knowledge distribution within the organization. Knowledge is more useful to the organization when it is shared with and accessed by other people in the organization. People learn knowledge through personal experiences or from other people's experiences. This is satisfactory for the individual level, but this is not acceptable for an organizational level. Knowledge must be effectively and efficiently be disseminated throughout the organization in order for its results could be fully realized. Verbal and written expressions of knowledge are common vessels for the distribution of knowledge. These methods are useful when people are the end users for the knowledge, but

in order for machines to understand the knowledge inside the text some other representations must be considered.

Artificial Intelligence (*AI*) helps overcome these two obstacles related to knowledge management. Applications of AI have discovered patterns in data that would not have been possible by a human. The results from such data analysis can be interpreted by a person can be used to modify the organization's practices, thus influencing the organization. AI research into natural language understanding helps transform knowledge encoded inside of text to a more machine-readable format, which can be used by other automated processes. In this way an organization's knowledge can influence the AI application. [Watson 2001] describes the effect of information on knowledge and the effect of knowledge to create new information. This cycle is critical in the discovery and creation of new knowledge. Also, Watson discussed knowledge management systems and the complimentary relationship to CBR applications [Watson 2001].

A variety of *Case-Base Reasoning* (*CBR*) applications have been implemented since the idea of CBR was founded. These applications range from helpdesk [Aha 1998] to tutorial applications [Bruinghaus and Ashley 2001]. CBR is defined in [Reisbeck and Schank 1989]:

"*A case-based reasoner solves problems by using or adapting solutions to old problems.*"

The most important prerequisite in any CBR application is a collection of experiences in the form of a case-base [Bartsch-Spörl et al. 1999]. Case authoring is the acquisition of new experiences that are not represented in the case-base. These experiences are captured during the case authoring process. A majority of CBR applications use manually intensive approaches to create new experiences for their case-bases. There has been little or no research done to facilitate automatic or semi-automatic approaches to authoring of cases for the case-base [Aha 1998]. This thesis will describe a semi-automated approach to case authoring that utilizes resources that are readily available in an organization.

The case-base is a reflection of the experiences that have occurred, but may not be as comprehensive as other knowledge resources such as manufacturers' manuals. Domain

experts rely on these other resources to assist them in solving new problems. Our approach to case authoring does not start at the individual experience, but at the documents that contain domain knowledge. It would therefore be useful to develop a case-base in two steps. The first step is to build a generic, comprehensive case-base from technical documentation. In the second, continuing step the case-base grows incrementally with experiences of the organization that uses the CBR system. This approach eliminates the manual processing of previously documented experiences and allows the domain expert to focus their time on authoring cases from the anomalous ones. Finally, the effectiveness of a case can be determined from the historical statistics that have been compiled from past experiences.

The thesis describes this new approach to case authoring for CBR applications and presents results, discussion, and future work related to this case authoring approach. Chapter 2 reviews the different approaches taken for case authoring and a potential problem domain, maintenance and repair environment, is identified as a focus for applying our new approach to case authoring. The third chapter describes the general architecture for our case authoring approach and Chapter 4 outlines the details of implementing this case authoring approach in the aerospace repair and maintenance domain. Chapter 5 displays the results for applying this new case authoring approach to the problem domain and these results are discussed in Chapter 6. The identified areas for future research and development are listed in Chapter 7. The final chapter concludes by reviewing the implementation, results, and future work for our case authoring approach.

2 Background

2.1 Case-Based Reasoning

CBR is a machine learning technique that adapts historical experiences to solve new problems. The works of Roger Schank on dynamic memory laid the foundations for CBR in Artificial Intelligence [Aamodt and Plaza 1994]. This work was inspired by the other research done in the field of cognitive science [Bartsch-Spörl et al. 1999]. One of the first systems to implement CBR was CYRUS [Watson 1997]. CYRUS was based upon Schank's dynamic memory model and *memory organization packets* (*MOP*) theory of problem solving and learning. This system contained the various meeting information and knowledge about former US Secretary of State, Cyrus Vance. The information and knowledge contained in the system was accessed using a question and answer interface. The case memory model implemented in CYRUS became the basis for other CBR systems [Aamodt and Plaza 1994].

The CBR cycle identifies the processes required to perform CBR. The CBR cycle can be used to introduce CBR to AI researchers [Aha 1998]. The CBR cycle takes in an unknown problem description and produces a proposed solution based on the past cases stored in the Case-Base. Aha proposed five processes in the CBR cycle [Aha 1998]. They are:

1. RETRIEVAL of cases most similar to the problem description presented in the current problem.
2. REUSE of the knowledge and information stored within the case to solve the current problem
3. REVISE the proposed solution according to the current experience
4. REVIEW the result of applying the proposed solution to the current problem
5. RETAIN this current experience by recording the information and knowledge in the case-base

When a new problem occurs, the problem description is composed using a full or partial set of features contained in the case definition. The problem description is then used to retrieve the most similar cases stored in the case-base. The information and knowledge from the retrieved case solution is used to adapt the case solution into a proposed solution. The proposed solution is applied to the problem and the result is recorded in the case-base as a new case. If the result was not favourable then further iterations of the CBR cycle are needed until the expected outcome is observed. This is a variation to the original CBR cycle proposed by [Aamodt and Plaza 1994]. For instance, Aha introduces a fifth step to the CBR cycle called REVIEW. This is new step evaluates the outcome of the proposed solution. Also, there can be iterations performed between REVIEW and REVISE to quickly adapt the proposed solution into the desired outcome. The CBR cycle is done whenever new data is presented and therefore is classified as a lazy learning approach to machine learning [Mitchell 1997]. This approach to machine learning allows the case-base to grow and adapt itself incrementally over time.

CBR is well suited for problem domains that possess the following constraints: the ability to facilitate incremental learning, the ability to perform query-specific reasoning, fast training performance, tolerance for missing values, and the ability to provide precedent explanations [Aha 1998]. Incremental learning tasks do not assume that all the training data is made a priori and further learning is done later when more training data is made available. CBR approaches prevent the early selection of summary generalizations on the presented training data, which makes CBR suited for problem domains that require incremental learning [Aha 1998]. CBR benefits from localized information from specific queries because no single global approximation to estimate a target function is established [Mitchell 1997]. Therefore, problems that possess query-specific reasoning can benefit from CBR implementation [Aha 1998]. Some problem domains are identified by [Aha 1998] and they include feature selection, robotic navigation and interactive troubleshooting. Three examples of CBR systems related to diagnosis and repair are a system developed by General Dynamics Electronic Boat Division, CaseLine, CASSIOPEE, and IDS.

General Dynamics Electric Boat Division developed a CBR application to troubleshoot large-scale manufacturing problems not covered by standard procedures [Brown and Lewis 1991]. Large-scale manufacturing is the development of a few, complex items over a long period of time; for instance, one ship takes approximately 5 years to build. The task faced in this domain is how to effectively solve problems where standard procedures are ineffective. These problems or "nonconformances" are resolved in relative isolation. Domain experts do the research into resolution of the nonconformance and the results are seldom communicated within the organization. The first attempt at acquiring and distributing this knowledge within the organization was done using a rule-based reasoning system. This rule-based system became too complex to maintain as the number of nonconformances increased. The rule-based approach was replaced by a CBR approach because the issues of knowledge acquisition and brittleness were less severe. The domain experts performed case authoring and were more comfortable expressing the problem descriptions and solutions in the case structure than in the form of rules. The CBR application demonstrated more robustness and flexibility compared to the previous rule-base reasoning system.

CaseLine was one of the early applications of CBR for interactive troubleshooting problems on board Boeing 747-400 aircraft at British Airways [Magaldi 1994]. This CBR system was implemented instead of using a rule-based or model-based approach. Since rule-based approaches use general conclusions, context from each individual case is not considered and contextually rich decision making environments use this information when making decisions. CBR can take this information into account when retrieving similar encountered problems and constructing a proposed solution. Model-based reasoning has the capability to represent domain knowledge very well, except a large amount of time is required to create these models. Another problem encountered with model-based reasoning was the decision of what models to construct. For instance, jet engine problems can be modeled in various different ways: gas-path model, vibration model, structural model, etc. It is unclear which model(s) would be successful in providing a solution for a problem on that jet engine. A CBR implementation provided a mechanism for retrieval of past experiences, perhaps with modifications and the ability to collect historical experiences over time. The largest overhead

encountered in the development of CaseLine was the initial problem analysis, case construction, and data entry. CaseLine can be searched using three different modes: ATA chapter structure, EICAS message symptoms, and defect reports. Once a case is retrieved, CaseLine identifies the procedures that have the highest likelihood of solving the problem. The size of the CaseLine case-base was 200 cases at the time of publication. This tool was not used to alleviate the maintenance technician from using already approved resources such as manuals, but used to reduce the amount of time needed for fault finding and analysis.

Another aerospace application of CBR is CASSIOPEE [Heider 1995]. This system was developed for the use of troubleshooting the CFM-65-3 jet engine. Problems that occurred on the jet engine were stored in a maintenance and reliability database where each record contains parametric and textual information about the problem. The information recorded in the textual fields was identified as valuable in the case authoring process, except the interpretation of these fields was done manually. First, decision tree induction was used to determine the relevant slots in the parametric data and use them in the retrieval of cases from the case-base. Next, the information from the free text narratives was manually processed to develop a list of technical terms that could be used in the retrieval of cases. This list of technical terms describes symptom specific parameters found in the text descriptions. This technique resulted in the identification of 16,000 cases from the database. Later evaluation of the case-base resulted in a more precise case-base and more accurate retrieval [Heider et al. 1996]. A set of 100 new technical parameters was introduced and the case-base was manually reconstructed with this new information. Further testing was performed to assess the accuracy, robustness, flexibility, performance, space, and effectiveness. The main conclusion reached was that it is desirable to have a few cases rich in information than many poorly documented cases when developing high quality decision support systems.

The *Integrated Diagnostic System* (*IDS*) is an applied hybrid reasoning decision support tool developed at the National Research Council of Canada [Wylie et al. 1997]. IDS was implemented in the aerospace domain at Air Canada. IDS integrated two different reasoning approaches rule-based and case-based reasoning. The rule sets were developed from

technical documentation and domain expertise and represented domain knowledge that was stable. The undiscovered knowledge not represented within the rule set was captured by the IDS CBR component. Similar to [Brown and Lewis 1991], domain expertise not yet reflected in the technical documentation could be added to the case-base and retrieved to solve problems as they occurred. A similar philosophy of assistance, rather than a recommendation as outlined by [Magaldi 1994], was taken. The maintenance technician is given all the available information for making the decision on solving the problem, but that person has to ultimately decide which action to take. The IDS case structure consisted of symptoms that described the problem, recommended solution, and historical statistics used to evaluate the effectiveness of past applications. Messages that are automatically generated by the aircraft are used to describe the symptoms for the problem, which are used in the case retrieval process. The case solution is displayed to the maintenance technician with its historical statistics. Cases are recorded by manually entering the information from identified problems using a custom designed tool called the *Snag Rectification Object Validation* (*SROV*) tool. This tool allowed the domain expert to validate the data for a problem instance and update the case-base. The data used in the creation of cases comes from the operational data from the aircraft and the repair reports written by the maintenance staff. Similar to [Heider 1995], the text within repair notes contain valuable information describing symptoms and solutions. Since the domain expert must interpret this information, it is considered a time consuming task. The case-base could be updated in two of the following ways. The first was that the problem instance became a new case in the case-base. The second was a modification of the historical statistics field of a retrieved case in the case-base. The problem instance was recorded as either a successful or failed application of the case. Due to the limited availability of domain experts to create cases, the case-base did not capture the domain expertise it was intended for. In order to develop a successful CBR application in an organization, many constraints have to be taken into consideration, one of them being the access to domain expertise. Therefore, a manually intensive case authoring approach is not optimal in the aerospace maintenance domain and other more automated approaches need to be developed.

CBR systems have been useful in many different problem domains. CBR is especially suited for problem domains that rely on past experiences to propose solutions to current problems. This approach is favourable when the cost associated with storing the historical experiences is more important than the query/response performance. The CBR approach also takes advantage of specific localized information from cases to construct solutions to queries. As demonstrated in the above, CBR applications are helpful as a tool for knowledge management and acquisition in the interactive troubleshooting domain.

2.2 Information Extraction and Case Authoring

The most important prerequisite in any CBR system is a collection of experiences in the form of a case-base [Bartsch-Spörl et al. 1999]. Case authoring is the process of capturing new experiences within the case-base. As seen in the above examples, the approaches to case authoring are very labour intensive and require the interaction with the domain expert. These case authoring approaches are typical and start at the individual experiences. As the number of individual experiences used in case authoring increases, the time required from the domain expert also increases and usually the availability of the domain expert is very limited. Besides the domain expert is not interested in recording their knowledge, but applying it. The discovery of new knowledge through manually intensive case authoring is difficult and this obstacle is commonly known as the knowledge acquisition bottleneck. Since little or no research has been done to facilitate automatic or semi-automatic approaches to authoring cases for a case-base [Aha 1998], more effort is needed to explore different possibilities to make case authoring more automated.

One possible solution is to integrate *Information Extraction* (*IE*) techniques into the case authoring process. Information Extraction is the creation of a structured representation of selected information identified from within the text [Grishman 1997]. In contrast to full text understanding, Information Extraction tasks focus only on the text that is relevant to the task. [Grishman 1997] has identified two major parts in the Information Extraction process. The first is the extraction of individual "facts" from the text using local text analysis. The second part is the generation of larger facts through the identified relationships that exist between

facts. A final step is taken to translate the pertinent facts into the desired output, called a template. Since much of the valuable information used for case authoring is in textual form [Wylie et al. 1997; Heider 1995], case authoring approaches could benefit from implementing Information Extraction techniques.

The use of textual documents to extract knowledge is a relatively new idea. Brüninghaus and Ashley were the first authors to discuss it and its implementation in the *SMart Index LEarner* (*SMILE*) [Brüninghaus and Ashley 2001]. SMILE is a CBR application used to tutor law students by establishing legal precedence from litigation. Textual documents related to legal suits are processed to identify parties, products, locations, and other information stated in each legal suit. Abstracted roles (plaintiff, defendant, product) and events (disclosed, discussed) are assigned to this specific information so more general patterns that can be constructed and saved in the SMILE case-base. This automated process authors cases from the legal text after the symbolic learning algorithm, ID3, processes a training set of marked up text. This Information Extraction approach to case authoring reduces the amount of resources needed to author cases from individual experiences.

2.3 Maintenance Domain

Aircraft, cars, trucks, computers, and people have documentation written to allow a person to diagnose and repair problems that may occur. Domain experts use this documentation to assist in making timely decisions on what actions should be taken to resolve a problem. After the solution has been applied, the domain expert may record this experience in textual form for future reference. Domains such as aerospace and open-pit mining maintenance implement computer applications to track maintenance activities. Other domains such as the medical domain may use more traditional methods such as paper to achieve a similar functionality. This documentation and recorded experiences can be instrumental in the acquisition and maintenance of knowledge stored in a case-base.

Aircraft are very complex systems with a variety of sensors, computers, and communication equipment. This makes the troubleshooting of aircraft difficult even with the onboard

diagnostic capabilities and the extensive documentation developed and provided by the aircraft manufacturer. Aircraft technicians must make timely decisions on whether the aircraft being maintained will be operational within limited time-span. These decisions are reached after a variety of knowledge and information resources are consulted and this information is distributed over different computer systems. For instance, British Airways Engineering has 300 legacy systems; most of them were not integrated with each other. An engineer would have to consult four different legacy systems simultaneously to perform a planning task [Careless 2003]. It would be helpful to collect the information from all these different resources so that the maintenance technician can make more timely, accurate decisions. The majority of knowledge about the aircraft is found in the aircraft's manuals written by the aircraft manufacturer (e.g. *Trouble Shooting Manual* (*TSM*), *Illustrated Parts Catalogue* (*IPC*)). Symptoms to problems related to the TSM are identified as *Fault Event Objects* (*FEOs*) within IDS and stored in a database [Wylie et al. 1997]. Information related to the repair and maintenance of the aircraft is found in the *Aircraft Maintenance Tracking And Control* (*AMTAC*) recording system. The information from these four information resources are critical in documenting reoccurring experiences, which have good potential for cases in a case-base.

Here we describe procedurally a generic maintenance incident, with the focus on the different knowledge sources involved. A maintenance technician may consult with a variety of manuals when fixing an aircraft. The sensor data from the onboard, built-in test equipment is collected and used to identify the fault isolation procedure in the TSM. These fault isolation procedures contain a list of possible causes and recommend further testing procedures to isolate the faulty components. The possible causes list is a list of suspected faulty components from which the maintenance technician must choose. This list is unordered and cannot be used as the only rationale for their solution. This list is key for identifying components that are referenced in the rest of the fault isolation procedure, but these components are listed arbitrarily. The possible causes list could become more influential in the decision making process if a process for ranking the components is developed. The maintenance technician must perform the recommended actions in the fault

confirmation and fault isolation sections of the fault isolation procedure to solve the problem. Once the faulty component has been identified, the maintenance technician may reference other manuals such as the IPC for part numbers and company information for ordering the component. The TSM is a valuable source of knowledge in identifying faulty components of the aircraft and the IPC is useful in further identifying the component. The references between these manuals are not explicitly defined, but can be inferred by the domain expert. The knowledge from the TSM and IPC is critical in authoring cases in the airline's maintenance organization and these manuals are updated quite frequently. One problem the case authoring approach must address is the ability to automatically obtain the knowledge represented in the TSM fault isolation procedures and IPC so newer versions of these manuals can be used to update the case-base. The case creation stage of our case authoring approach will address these identified problems.

Aeronautical maintenance performed on Canadian aircraft is required by law to record all maintenance activities performed on an aircraft [Transport Canada, 2002]. The aircrew and maintenance staff record maintenance incidents in maintenance tracking systems such as AMTAC. Maintenance actions describe the work done to address the maintenance incident. These AMTAC reports contain information such as the date and time of the problem and repair, personnel that found or fixed the problem, and the parts used in the repair. Four data entry fields have been identified as important to the case authoring process. The two of fields contain free text that describes the maintenance incident and the action taken. The other two fields contain information on what parts were installed and removed in the maintenance action. Since this abundant AMTAC data is obviously relevant for case authoring and it was available to us, we have decided to use it in our case authoring process. We have identified three problems that we must address in order to use the maintenance reports in case authoring approach. The first is the understanding of the natural language that is entered in the free text entry fields. The maintenance technician often personalizes the text used to express the problem and solution. For example, the text contains specialized word representations such as 'XFER' for 'transfer'. Since conventional English grammar is not guaranteed, the word order may not be helpful in understanding the text. This makes the task

of understanding the text non-trivial [Farley, 1999]. Also, there is no requirement to fill out all the fields in the AMTAC report. This causes missing data values to occur in the data and information to be entered in other fields. Finally, there is no guaranteed clear correlation from the diagnostic information generated by the aircraft, the fault isolation applied to repair the aircraft, and the maintenance reports composed by the maintenance technicians. The solutions to these problems will be discussed in the case validation stage of the case authoring approach.

3 Case Authoring Methodology

The proposed case authoring methodology from readily available resources such as textual documentation and historical experiences addresses the knowledge acquisition bottleneck for CBR systems. The knowledge acquisition bottleneck is defined as the difficulty in capturing knowledge for automated processes [Rubin et al 1999]. Domain experts express their knowledge using informal methods like natural language, diagrams, and common sense [Boicu et el 2001]. These informal methods often do not capture all the essential details especially ones that are considered obvious. One approach is to employ a knowledge engineer to translate the domain expert's knowledge into a more formal, precise and complete representation. This manual approach to knowledge acquisition is very difficult, error-prone, and time-consuming [Boicu et al. 2001]. Therefore, any case authoring approach needs to address the knowledge acquisition bottleneck problem and this problem can be overcome by automating as much of the case authoring process as possible. This is accomplished by using already available resources such as technical documentation and historical experiences to automatically construct cases for the case-base.

One readily available resource is the individual historical experiences that are captured by the organization. Organizations can capture historical experiences in a number of different ways ranging from high-tech information management systems to more traditional paper based filing systems. These historical experiences contain parametric and textual information used to describe problem and the solution that was applied to resolve the problem. Some historical experiences may contain assessment information on the effect of the applied solution to the problem. This flow of information contains tangible examples of the decision-making process and the domain expert's knowledge used to reach that decision. Traditionally, historical experiences have been the initial starting point for many manual [Heider 1995, Wylie et al. 1997] and automated [Brüninghaus and Ashley 2001] case authoring approaches. The case-base constructed using these approaches reflect the problems that have happened in the past, but cannot predict unforeseen problems that may occur in the

future. These unforeseen problems may be already documented in other resources such as the technical documentation.

Another readily available resource used by domain experts is technical documentation. Technical documentation is a compilation of knowledge written by the domain experts and refined over time. As described by [Boicu et al. 2001], much of this knowledge is encoded in the natural language and visual representations found within the technical documentations. In order for an expert system to benefit from this knowledge, these informal methods of representations must be translated into more formal representations so that the information can be automatically processed. Information Extraction is a useful technique used for translation of text into more formal representations that can be processed by an expert system. These technical documents are processed using Information Extraction techniques to author cases from the knowledge expressed in the text. Even though full understanding of the text is not achieved, Information Extraction is a useful technique to identify specific information within the text and the development of more complex structures from the text [Hobbs et al. 1997]. The results of Information Extraction could be transformed into the case structure outlined by the CBR system and thus, Information Extraction can be used to process technical manuals to author cases. By gleaning the knowledge already documented in the technical texts, relief from some of the difficulties caused by the knowledge acquisition bottleneck can be achieved. For instance, pre-processing of a priori knowledge from documentation can benefit the case authoring process. Case authoring approaches for plan creation found that manually eliciting knowledge from a textual doctrine is critical in establishing planning knowledge in case authoring [Aha et al. 2001].

The proposed methodology for case authoring from readily available resources such as textual documentation and historical data is outlined in Figure 1. This methodology is fundamentally different from other case authoring approaches in that it does not start with historical experiences within the organisation, but domain knowledge encapsulated in available text from technical documentation to author cases. This approach benefits from the readily available resources in two ways. The first is the automatic creation of a case-base

from both technical documentation and historical experiences based on knowledge exhibited within the organization. Since already known knowledge is represented in the case-base, this can be used to filter out historical experiences that are already represented in the case-base. Anomalous historical experiences are identified and can be further processed by other case authoring approaches, thus focusing their efforts on unknown problems that occur.

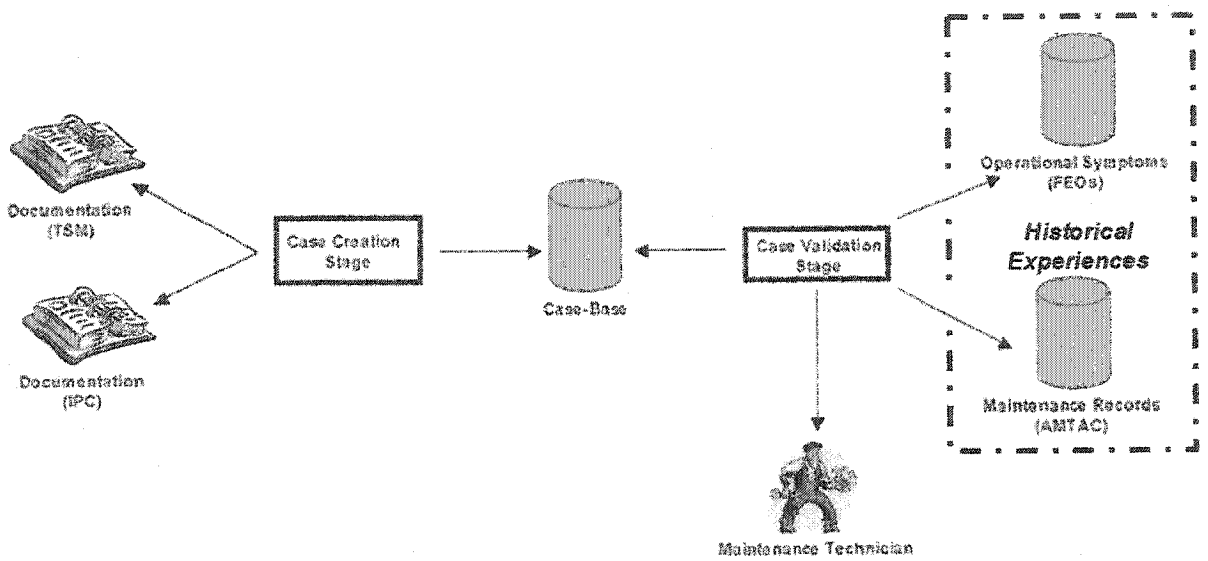

Figure 1: Case Authoring Approach using Text and Historical Experiences.

Many CBR applications assume that this documentation does not exist, but for some domains this information is readily available. We have identified two stages in development of the case-base: case-base creation and case-base validation. The case-base creation stage is a process that extracts cases from the technical documentation and enhances those cases with additional information identified in the technical documentation. Since the technical documentation contains comprehensive knowledge about the problem domain, there must be a way to authenticate which cases have been experienced by the organization. The case validation process uses the historical experiences captured by the organization to determine if the case was used and supplements the case with historical examples. The resulting case-base combines the information from the two different resources, technical documentation

and historical experiences, to create a comprehensive case-base that can be used by the organization.

3.1 Case-Base Creation

3.1.1 Technical Document Identification and Analysis

Domain experts use a variety of technical documentation when solving problems within the organization. The availability of these technical documents and how they are used are critical to our case authoring process. Through examination of the technical documentation and interaction with the domain experts, usage patterns can be established for how these documents are used in the decision-making processes for the organization. Since not all the documentation may be relevant in the construction of the case-base, these usage patterns are important in identifying the technical documentation that may be useful in the case authoring process. Once the technical documentation influential to case authoring has been identified, each document is assessed according to how knowledge can be acquired. Some of the considerations are storage format, structure, and references to other documentation.

The first consideration is the format used to store the documentation. There are a variety of document storage methods from paper-base to electronic. The differences in document storage influence the amount of effort needed in the case authoring process. For instance, electronic documentation requires less processing than paper documentation because of the extra overhead of scanning the paper document is required for paper documentation. Special software, such as an Optical Character Recognition program, and configuration of such a program is required to transform the paper-based document into a more machine-readable electronic format. Once the document is in electronic format, it does not necessarily guarantee case creation because many different electronic formats exist. Examples of some electronic formats are *HyperText Markup Language* (*HTML*), *eXtensible Markup Language* (*XML*), *Standard Generalized Markup Language* (*SGML*) and propriety formats such as Acrobat PDF and Microsoft Word. HTML, XML, and SGML documents are easier to process due to standards established by the [W3C, 2003] in publishing content in this format. Proprietary formats are more difficult process due to the specialized format for document

storage and tools for viewing the text may not allow one to easily export a text-only representation.

The second assessment to consider is the structure of document. The structure of the document outlines the organization and this organization may influence the organizational structure of the case-base. Since different documents contain different organizational structures, this also should be reflected in the organization of the case-base. This multi-dimensional organization of the case-base allows for different views of the cases depending on the context of the query. Other structures may also be identified within the document such as encoded rule structures, which are influential in the case authoring process. These other structures describe problem symptoms and recommended solutions that can be expressed as cases in the case-base. Finally, this structure may be a reflection of the writing style incorporated by the organization and be helpful in processing other similar documents.

References between documents are a consideration that must be assessed to assist in the case authoring process. Documents can contain references between other documents, which can be quite explicit or implicit. The explicit references are easier to establish than implicit ones. Both of these references may be used to gain more information about the problem or solution by processing the referenced document for extra information. Since this extra information may further enhance the cases constructed from one document, it is important to discover these relationships among the collection of documents.

This up front identification and analysis of the technical documentation has two purposes: to gain better understanding of the documentation and how it is used, and to allow for the development of automated tools to handle the processing of documents. Organizations are dynamic in nature and changes in their decision-making processes are reflected in the technical documents. Tools used to manage knowledge should be able to easily reflect the current knowledge by automatically processing these documents. Information Extraction is an influential tool in discovering knowledge encoded within these documents and transforming this knowledge into the form of a case-base.

3.1.2 Development of the Case Structure

Case structure is described as two parts: the problem description and recommended solution [Watson 1997]. The problem description outlines the symptoms that problem exhibits and the symptoms can be defined as different information types from parametric and textual values of complex structured objects. The problem description determines the similarity function for case-base retrieval, which is used in the retrieval of the case. Thus the retrieval mechanisms for CBR are more elaborate than the ones implemented in other instance-based learning systems [Mitchell 1997]. The recommended solution is the actions taken to resolve the problem. They can be adapted to suit the presented problem instance. The structure of the case will determine the template for which the Information Extraction system will use in its extraction task

3.1.3 Development of the Information Extraction system

In typical Information Extraction systems, a template is a collection of tasks and is the end result of the Information Extraction system [Freitag 1998]. Templates contain fields for information called slots. Any instantiation of a slot with information is called a slot fill. Information Extraction systems address multiple-subtasks that are defined as goals for the Information Extraction system such as semantic and syntactic pre-processing, slot filling, and anaphora resolution [Freitag 1998]. Our case authoring approach will implement Information Extraction techniques for the purpose of slot filling. The development of Information Extraction systems can be done using two approaches: knowledge engineering approach and automatic training approach [Appelt and Israel 1999]. The knowledge engineering approach employs a knowledge engineer to develop grammar rules for the Information Extraction systems and is more laborious. The automatic training approach takes annotated text as training data to develop rules for the Information Extraction system using a machine learning algorithm. [Appelt and Israel 1999] outline these approaches in greater detail and reveal advantages and disadvantages with each of their implementations.

3.1.3.1 Identification of Information for Case Authoring

The identification and analysis of the text documents in the previous steps will assist in the development of the rule sets used to identify information and relationships useful in case authoring. These rule sets can be constructed either by the knowledge engineer or from the annotated text used in the machine learning process. The rules defined for slot fills are used to identify problem descriptions and solutions, which can be found in multiple documents. The result of this stage is a set of rules that identify slot fills for the template.

3.1.3.2 Identification of Relationships

As mentioned before, the text documents are prominent in the development of the rules that identify the relationships between the identified problem descriptions and solutions. There are three different relationships that need to be identified for case authoring. The first is the collection of information used to identify the same problem descriptions with different representations. A similar process is also performed to identify information for same solutions with different descriptions. These two information processes result in problem descriptions and solutions that reflect all the information, which can be distributed over different documents. These two relationships construct a more detailed collection of problem descriptions and solutions that are important in the later stages of our case authoring approach; for instance, the identification of problem descriptions and solutions in the historical experiences. Finally, the relationships between the problem descriptions and solutions are represented by rules and are used by the Information Extraction system. This correlation is important in the next step in creating cases for the case-base.

3.1.3.3 Construction of a Case

Once the rules for the Information Extraction system have been developed, the final step is to fill in the template. The rules used by the Information Extraction system identify and correlate all the information related to the identified problem descriptions and recommended solutions. This information is used to fill in the slots within the template, which represents the case structure. Each completed template is used to construct a case and each new case is stored within the case-base. Even though some of the case information is

incomplete, this is left for later stages in the case authoring process. The resulting case-base is a reflection of the knowledge represented in the identified technical documents.

3.1.4 Testing the Information Extraction System

IE systems are evaluated using measures defined in the *Message Understanding Conferences* (*MUCs*) [Grishman and Sundheim 1996]. Precision, recall, and F-measure are three accepted measures for determining the performance of an Information Extraction system. *Precision* (*P*) measures the correctness of the provided information to assess the accuracy of the Information Extraction system [Makhoul et al. 1999]. Precision is expressed as follows:

$$P = \frac{\text{Correct Information Extracted}}{\text{Information Extracted}}$$

Recall (*R*) is a measurement of comprehensiveness of the correct information compared to total amount of correct information identified manually [Makhoul et al. 1999]. Recall also measures the information missed by the Information Extraction. Recall is expressed as follows:

$$R = \frac{\text{Correct Information Extracted}}{\text{Total Possible Correct Information}}$$

Results of precision and recall are combined in F-measure, which is a weighted harmonic mean [Makhoul et al. 1999]. F-measure is expressed as follows: [Jurafsky and Martin 2000]

$$F = \frac{(\beta^2 + 1) * \text{Precision} * \text{Recall}}{\beta^2 * \text{Precision} + \text{Recall}} \quad 0 < \beta < 2$$

β is the similarity weight between precision and recall. β is commonly expressed as 1 when precision and recall are equally weighted. If precision is favoured β is greater than 1, and when recall is favoured, β is less than one. A higher the F-measure score means that the Information Extraction system has a better combination of precision and recall performance.

From the Information Extraction systems entered in the MUCs, domain experts and knowledge engineers define acceptable thresholds for precision, recall, and F-measure. Some examples of tasks performed at the MUCs are extracting information from naval messages and newspaper/newswire articles. In particular, the tasks for MUC-3 and MUC-4 conferences were gathering information about terrorist incidents in Latin America. A template was defined for identifying relevant information from 1,000 pieces of text. The best systems entered in the MUC conferences have resulting precision scores of 50% and recall scores of 40% [Cowie and Lehnert 1996]. A standard set for state of the art Information Extraction systems is an F-measure of 0.6 [Appelt and Israel 1999]. These state of the art benchmarks can be compared to the performance of human analysts using similar measures where human analysts scored between 0.6 and 0.8 [Cowie and Lehnert 1996]. The task of authoring cases from textual documentation is compariable to MUC tasks. The case structure is used as a template for the Information Extraction task and the manuals are the pieces of text that the Information Extraction system must process. The newpaper and newswire articles used in the MUC task are more difficult to process because of the writing styles differ between different publishers. Since the manufacturer publishes manuals for a piece of equipment, there is constitency within these manuals and the task for extracting information should be easier. Therefore, it is reasonable to use these state of the art benchmarks in our Information Extraction task in case authoring.

An iterative development cycle occurs with the Information Extraction system until these thresholds are achieved. Once acceptable thresholds for the Information Extraction system are established, the Information Extraction system can be used to construct cases from the documents.

3.1.5 Build the Case-Base

Now that the Information Extraction system for case extraction has been built, it is ready to process the identified technical documents. These documents are processed by Information Extraction system and a case is created whenever a template is filled with the appropriate information. The technical documentation is seen as a reflection of the manufacturer's

expertise for a piece of equipment and any cases that are authored from the technical documentation are creditable. Other tests are performed on the case-base for case-base management and quality assurance purposes. For instance, a test to check for duplicate cases that exist within the case-base is a plausible test for case-base management purposes. Once the Information Extraction has processed, a case-base is created from the technical documentation. The resulting case-base is used as input into the next stage, case-base validation.

3.2 Case-Base Validation

3.2.1 Historical Experience Resource Identification

As mentioned above, domain experts use different mechanisms for recording problems and the solutions. A case-base constructed from only technical documentation does not reflect the operational knowledge, which is captured in the historical experiences recorded by the organization. It is important for our case authoring approach to identify and analyze these mechanisms for documenting problems and solutions that have happened in the organization. These problem/solution documentation systems range from sophisticated information management systems that use SAP [SAP 2003] to traditional paper-based systems. First, the recording processes useful in providing historical experience for case-base validation must be identified. This can be established through interaction with the domain experts. Once the different processes for recording historical experiences are identified, these processes are analyzed so that one could automatically extract historical experiences related to the case-base.

Similar challenges that were identified in the analysis of the technical documentation can be seen in the recording process analysis. Accessibility, format, and interpretation are all factors that must be considered when constructing an automated approach to extract the historical experiences related to the case-base. Since the information related to the problems and solutions may contain more information than just text, the automated processing must identify the different data types and how to handle them. It is common to store much of the information related to historical experiences in customized databases that contain textual and

parametric information [Heider 1995; Wylie et al. 1997]. Therefore, a better understanding of the database structure can be achieved with the help from the domain experts.

3.2.2 Problem Instance Retrieval

Once the system(s) for recording historical experiences have been identified, a process can be developed for automatically retrieving specific problem instances. After this process is developed, the case-authoring approach can authenticate the cases within the case-base. The first step is to retrieve an arbitrary case from the case-base. The problem description for this case contains symptoms that can identify the problem. These symptoms are used to search the historical experiences for instances of problem descriptions that match these symptoms from the case. Once the problem instances are retrieved, information about specific problems can be used to identify the solutions. Since not all organizations have recording processes that automatically correlate the problem descriptions and solutions, this step is important for establishing identifying information such as a temporal reference to correlate the problem descriptions to solutions automatically.

3.2.3 Case Solution Identification

The problem instances retrieved in the previous step are important in retrieving solutions from the historical experiences within the organization. Since some of the solutions retrieved are not related to the problem instance, further filtering must occur to eliminate the solutions that are not related to the problem instance. This filtering is done by matching the solution information stored within the case to the solution used in the retrieved solution. If there is a match then the problem instance is correlated to the solution else the solution is disregarded. If an organization has already implemented a process that correlates the problem instances to solutions then the above filtering is not required. The case solution identification step resulted in a set of problem instances with solutions that are related to the case both in problem description symptoms and solution information. These specific problem instances are used to provide information for the case's historical experiences and the statistics collected on these historical experiences.

3.2.4 Case-Base Update

Each problem instance in the set of problem instances is evaluated for the effectiveness of the solution. This evaluation can be done using the timeline established within the problem instance. Using the date and time information from the solution, a successful application of the solution can be determined if the symptoms outlined in the problem description disappear after the application of the solution. An unsuccessful solution can be inferred when the problem description symptoms persist after the application of the solution. This evaluation information is added to the problem instance and is used to update the case. The case is then updated with the problem instance reference information and the evaluation information is used to update the historical statistics for the application of the case. Finally, the updated case is saved within the case-base.

The case-base validation process is an iterative process. This process is completed once all the cases in the case-base have been processed. Steps 3.2.2, 3.2.3, and 3.2.4 are repeated until all the cases have been authenticated. The case-base validation process adds the historical information to the information from the technical documentation in the case-base. Also, a better understanding is gained on what amount of the technical documentation is used solving everyday occurring problems and what amount of reoccurring problems are not covered by the technical documentation. This case-base is a knowledge resource that can be updated and distributed within the organization with the most current information and knowledge.

4 Implementation

The aerospace maintenance and repair was identified as the problem domain to implement our case authoring approach and Air Canada was identified as the potential end user. This problem domain possesses aircraft that are becoming more complex with the increased amount of on board computers, sensors, transmitters, and interconnecting networks. This domain is rich in diagnostic information that is increasing in volume. The maintenance technician must take this diagnostic information and other information resources into account when troubleshooting problems on the aircraft. Similar to [Brown and Lewis 1991], the amount of knowledge sharing amongst the maintenance technicians is limited and the maintenance organization staff is physically distributed around the world, thus this type of organization could benefit from sharing knowledge about similar experiences that happened. Some of the resources that are available to the Air Canada maintenance technicians are technical documentation, maintenance action recording facilities, and expert systems.

The technical documentation for the aerospace maintenance and repair domain is authored by different organizations. The original equipment manufacturers, government agencies, and airlines themselves contribute material contained in the different technical documents. Documents related to the maintenance and repair of the aircraft are the TSM, IPC, Aircraft Maintenance Manual, Aircraft Service Manual, and Minimum Equipment List manual. These different manuals contribute different information towards aircraft maintenance and repair. These manuals share a common structure outlined by the *Air Transport Association of America* (*ATA*) under ATA iSpec 2200. ATA reference codes label procedures or diagrams inside these manuals. ATA references are expressed as numeric code groupings separated by a '-'. An example taken from the AMM is '79-21-51-420-050', which identifies a procedure in the manual. The first pair of digits refers to the chapter; the second pair, the section; the third, the subject. The second last grouping refers to the function and the last grouping refers to the sequential number [Fischer 1997]. The first three digits are standardized by the ATA where the first two digits refer to the chapter number information and the next digit the subsystem information. The remainder of the digits are subject to interpretation by the

aircraft manufacturer and similarity of assigned ATA codes by the manufacturers is not guaranteed between the different manufacturers. In the above example, '79' refers to OIL related items on the aircraft and the first digit after the '-', 2 references OIL DISTRIBUTION related subsystems in the OIL chapter. The storage formats for these documents are Corel WordPerfect (MEL), SGML (TSM), and StarDoc (TSM). StarDoc is a proprietary system developed for Air Canada that manages the maintenance and repair documentation for all the Air Canada aircraft.

The AMTAC recording system stores all repair and maintenance activities that happen at Air Canada. Maintenance or aircrew staff record descriptions of problems into a logbook where they are later recorded into AMTAC. The maintenance technician uses these problem descriptions to decide what maintenance and repair activities are to be performed in order for the aircraft to operate. When repair or maintenance activities are completed, special forms are completed by the maintenance technician and later typed into AMTAC. Problem descriptions and maintenance actions are correlated to one another. Some AMTAC reports have manual references entered in the textual field, except that this does not occur quite frequently. Therefore, there is no guaranteed correlation between the maintenance activities and the manuals used in the repair and our case authoring approach has to perform this task in order to authenticate the cases in the case-base.

IDS system was an expert system developed by the National Research Council of Canada that used rule-based and case-based reasoning to diagnose and monitor fleets of equipment [Wylie et al. 1997]. IDS contained the infrastructure to handle the automatic messages transmitted by the aircraft and to transform them into a format that can be used within IDS. IDS used these diagnostic messages to determine if a problem has occurred on the aircraft. Problems are identified either by an activation of a rule within the IDS rule set or a retrieval of a case from the case-base. Finally, IDS provides the functionality to reference different manual documentation using an integrated web browser and allows the maintenance technician to monitor the complete fleet of aircraft and retrieve information about the problem when further investigation is required.

4.1 Case-Base Creation

4.1.1 Technical Document Identification and Analysis

TSM contains the knowledge about troubleshooting most problems that occur on the aircraft. The TSM contains fault isolation procedures that further outline tests and actions taken to resolve problems that occur on the aircraft. A fault isolation procedure contains five parts: list of possible causes, list of referenced procedures, a fault confirmation section, a fault isolation section, and a close-up procedure section. The list of possible causes is a list of components that are referenced in the procedure, but are not ordered in any specific ranking. This list is not to be used as a shotgun approach to troubleshooting. The shotgun troubleshooting approach replaces all the components related to the problem in the hope that one of the components fixes the problem. Since this is a very costly and inefficient approach to troubleshooting, this kind of approach to solving problems is discouraged especially when predefined procedures exist. The referenced procedures list outlines the different procedures mentioned within the fault isolation procedure. The fault confirmation section contains steps for further confirmation of the problem through administration of extra tests. The fault isolation section contains the troubleshooting logic in isolating the component that is at fault. The fault isolation section contains a hierarchy of procedures that structure this troubleshooting logic. Different branches may be taken depending on the outcomes of further tests. Finally, the fault isolation procedure contains references to other aircraft manuals such as the AMM and ASM.

The TSM is distributed in SGML format with a predefined *Document Type Definition* (*DTD*) structure. This DTD structure was used to construct a database so that the SGML could be stored. This approach to storing the SGML in a database allows for easy retrieval of the information related to the different tasks outline by IDS. For instance, the automatically generated messages processed by IDS and the IDS rule sets constructed from the TSM were updated using this TSM SGML database. Also, HTML pages were generated from this TSM SGML database to be used on an IDS website for TSM reference.

IDS rule set is constructed from the implicit rule set encoded in the TSM. Automatically generated messages are used as symptoms to identify the applicable fault isolation procedure. This rule set does not outline specific component information or actions taken on those components. It is up to the maintenance technician to follow the fault isolation procedure to resolve the fault. Also, there is no historical information, which could be useful to the maintenance technician, on what had happened historically and the effectiveness of the different solutions. Finally, there was no guaranteed reference to all maintenance incidents when an IDS rule and solution was applied and the result of applying that solution.

IPC lists all the components that on the aircraft and how they are physically related to one another. Diagrams and table structures in the IPC outline with great detail the composition of larger components from smaller components. The quantities for specific parts that are found on the aircraft are listed the IPC. Also, changes can be tracked for specific part number information, as it is modified or replaced by other parts. The IPC contains *Functional Item Number* (*FIN*) code information that can be correlated to components referenced in other manuals. This FIN code is a generic description code for a function of a component that could be mapped onto multiple parts that perform the same function. The IPC is also distributed in SGML format and stored in a database using a similar approach as performed with the TSM.

4.1.2 Development of the Case Structure

The case structure used in this case authoring approach is the same as the one used in IDS [Wylie et al. 1997]. The features used for case retrieval are indexed by the problem symptoms. These symptoms are the automatically generated messages from the built-in test equipment onboard the aircraft. The case separates these symptoms into different aggregations according to textual similarity, time proximity, TSM reference, and human association grouping. The textual similarity aggregation are diagnostic messages that are textually similar in nature and the time proximity aggregation identifies diagnostic messages that occur at the same time period of each other when the problem occurs. The TSM grouping aggregation is diagnostic messages that have relevance in identifying a TSM

reference for fault resolution. Finally, the human association grouping is a field where the domain expert can arbitrarily enter in additional symptoms that are related to the problem. The component and action taken on the component is stored in the case as the recommended solution. The actions taken are defined in a predetermined list of actions and the components are entered as part information codes. Additional information such as historical statistics and recorded incidents are also stored in the case. The historical statistics and recorded individual experiences are captured from the organization's historical data in the manual validation of case-base.

4.1.3 Development of the Information Extraction system using Regular Expressions

4.1.3.1 Identification of Information for Case Authoring

Reiterating from Chapter 2, Information Extraction does not perform a full understanding of the text and some of the better performing Information Extraction systems use simpler techniques to perform the task [Hobbs et al. 1997]. The first task of our Information Extraction system is to identify TSM components and actions taken on those components in the fault isolation procedures. Our case authoring approach implemented regular expressions to perform the Information Extraction task. Since a knowledge engineering approach to Information Extraction development was taken, a knowledge engineer developed regular expressions using the TSM. A manual text analysis of the TSM fault isolation procedure was performed to determine the most frequently occurring maintenance action. Word frequencies were compiled from the TSM text and the verbs that could be identified as maintenance actions were used to develop the regular expressions. The most frequently occurring maintenance action was replacement using the verb 'replace'. The verb 'replace' was used 15, 013 times in the TSM. A quick study of the examples in the text helped construct regular expressions. Chapter 5 describes these regular expressions and the frequency of their application within the text. These regular expressions are used to identify the actions and components within the TSM fault isolation procedure that represents a solution to a problem.

The problem descriptions were identified in the rule set that IDS extracted from the TSM. This IDS rule set identifies the diagnostic messages that are generated by the aircraft, which

are used in the problem description. These diagnostic messages are located on the *Left Hand Side* (*LHS*) of the IDS rule set. Unique identifiers represent the diagnostic messages encoded in the IDS rule set and these unique identifiers are used to get all the diagnostic message information from the IDS message databases. The interpretation of the IDS rule set establishes the symptoms that describe a problem, which will be used in the problem description for the case.

Finally, the knowledge engineer used the IPC DTD to identify the 'EIN' entity tag that surrounds the FIN code information. This FIN code is then used to identify part number and manufacturer information from the IPC. The IPC SGML database is queried to identify FIN codes and their respective part number and manufacturer information. This information is important to the case because it was used in the case-base validation stage.

4.1.3.2 Identification of Relationships

The interpretation of the IDS rule set not only identifies symptoms for the problem description, but it also correlates the symptoms to applicable fault isolation procedures. The IDS rule set contains a reference to the fault isolation procedure on the *Right Hand Side* (*RHS*) of the rule. As mentioned previously, the fault isolation procedures contain information on the solutions that solve the problem. This relationship between problem descriptions and solution is important in authoring cases for the case-base and this relationship is established through the IDS rule set and the TSM.

A mapping of components referenced in the TSM and the IPC is established through the FIN code. The FIN code is a correlation that can determine the part number and manufacturer information in the IPC to parts in the TSM. The component referenced in the TSM may contain a FIN code following the text description of the component. For example, 'BSCU (10GG)' is a referenced component in the TSM where '10GG' is the FIN code. The FIN code '10GG' is used to establish that two different part numbers from the same manufacturer are found in the IPC. This FIN code relationship is important in further enhancing the case with more information on the component used in the solution.

4.1.3.3 Construction of a Case

The case structure developed in 4.1.2 was used as a template for slot filling task of the Information Extraction system. The IDS rule set, TSM and IPC were processed to extract the information useful in completing the template and establishing larger structures of information from the identified relationships. Once a template was filled, this became a candidate for a case where the problem description and solution information was transferred from the template. Each of the cases in the case-base were constructed using the developed Information Extraction system.

4.1.4 Testing the Information Extraction System

The testing criteria for the Information Extraction system was the same measurements described in Section 3.1.4, which are recall, precision and F-measure. The results of this testing is outlined in Chapter 5 and the thresholds for satisfactory performance of the Information Extraction system were set at the same measurements as [Cowie and Lehnert 1996] and [Appelt and Israel 1999]. Evolution of the Information Extraction system took some iteration to achieve these threshold performance measures. Also, the identified relationships for constructing composite structured information were manually examined by the knowledge engineer for correctness.

4.1.5 Build the Case-Base

4.1.5.1 TSM Fault Isolation Procedure Scanner

The TSM Fault Isolation Procedure Scanner was developed to create the cases from the TSM and the implementation for this stage is outlined in Figure 2. The TSM Fault Isolation Procedure Scanner uses both the IDS rule set and the TSM fault isolation procedures to create the case-base. Each individual IDS rule is processed by the TSM Fault Isolation Procedure Scanner for symptoms located in the LHS of the rule and the corresponding procedure on the RHS. The automatically generated diagnostic messages are extracted from the LHS and then used to create a template case. A template case is created because a symptom set can have more than one recommended solution. The corresponding procedure from the RHS is scanned using the TSM Fault Isolation Procedure Scanner with the regular

expressions developed in the previous step. Once an action and component are identified within the TSM fault isolation procedure, a new case is duplicated from the template case. This new case has its component and action fields populated with the action and component information that was identified from the TSM fault isolation procedure. After the IDS rule set has been processed, a case-base is built from the IDS rule set and TSM documentation.

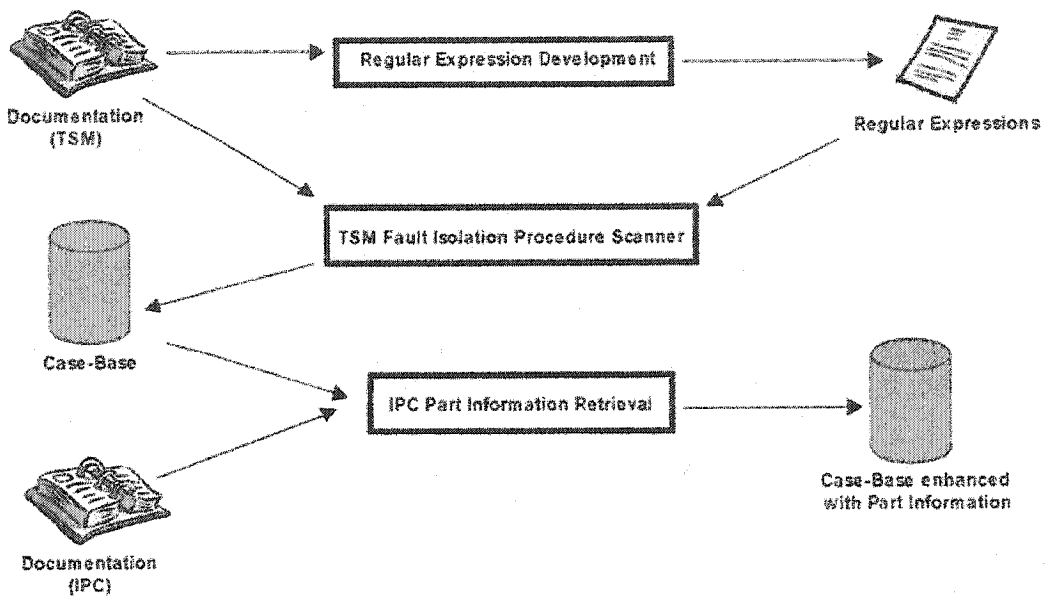

Figure 2: Case-Base Creation Stage

This case-base might be perceived as a duplication of the IDS rule set, but the case-base can be enhanced and updated with supplementary information. Further enhancements can be in the form of additional information gained from other manuals such as the IPC. Another form of enhancement is the recording of individual experiences that validate the case's usefulness. Once the case-base is enhanced with additional information, the cases contain more knowledge and information than the IDS rule set and can be updated easily. This up to date knowledge affects the way the cases are organized and retrieved and represents the current knowledge of the organization.

4.1.5.2 IPC Part Information Retrieval

The first enhancement of the TSM case-base helps identify components in the case validation stage. The IPC contains information about the specific part number and manufacturers. A correlation between components in the TSM and IPC are established through a code called the FIN code. Not all components in the TSM case-base have a FIN number associated with them. If a FIN code is found, it is used to identify IPC part number and manufacturer information and add this component information to the correlated TSM case. Since the identification of components in the AMTAC reports, which will be needed during the case-validation stage (see below), is difficult, any additional part information could be useful in this identification process. The resulting TSM case-base is ready to be validated with related individual historical experiences.

4.2 Case-Base Validation

The case-base validation stage is the process that further enhances the case-base by capturing the organization's maintenance history inside the case-base. Even though a large set of cases may have been extracted from the TSM, the aircraft may not have generated all the problem symptoms described by the TSM. For case-base performance, it is desired to minimize the number of cases, but still achieve the same amount of coverage [Lenz et al. 1998]. Since the TSM contains comprehensive information about problems on board the aircraft, an analysis of the aircraft's maintenance history can help reorganize the cases used in the case-base created from the TSM. In distributed CBR architectures, caching frequently retrieved cases on the client-side is done to improve the overall performance of the system [Smyth 1995]. A similar hierarchy of cache memory can be implemented for the generated case-base to improve performance where the most referenced cases are retrieved first before ones that have never been referenced. This applicability metric is established by validating the cases with historical experience. The case validation stage of case authoring can be broken down into four parts: identification of historical experience resources, problem instance retrieval, solution identification, and case-base update. The steps involved in the case validation stage are outlined in Figure 3.

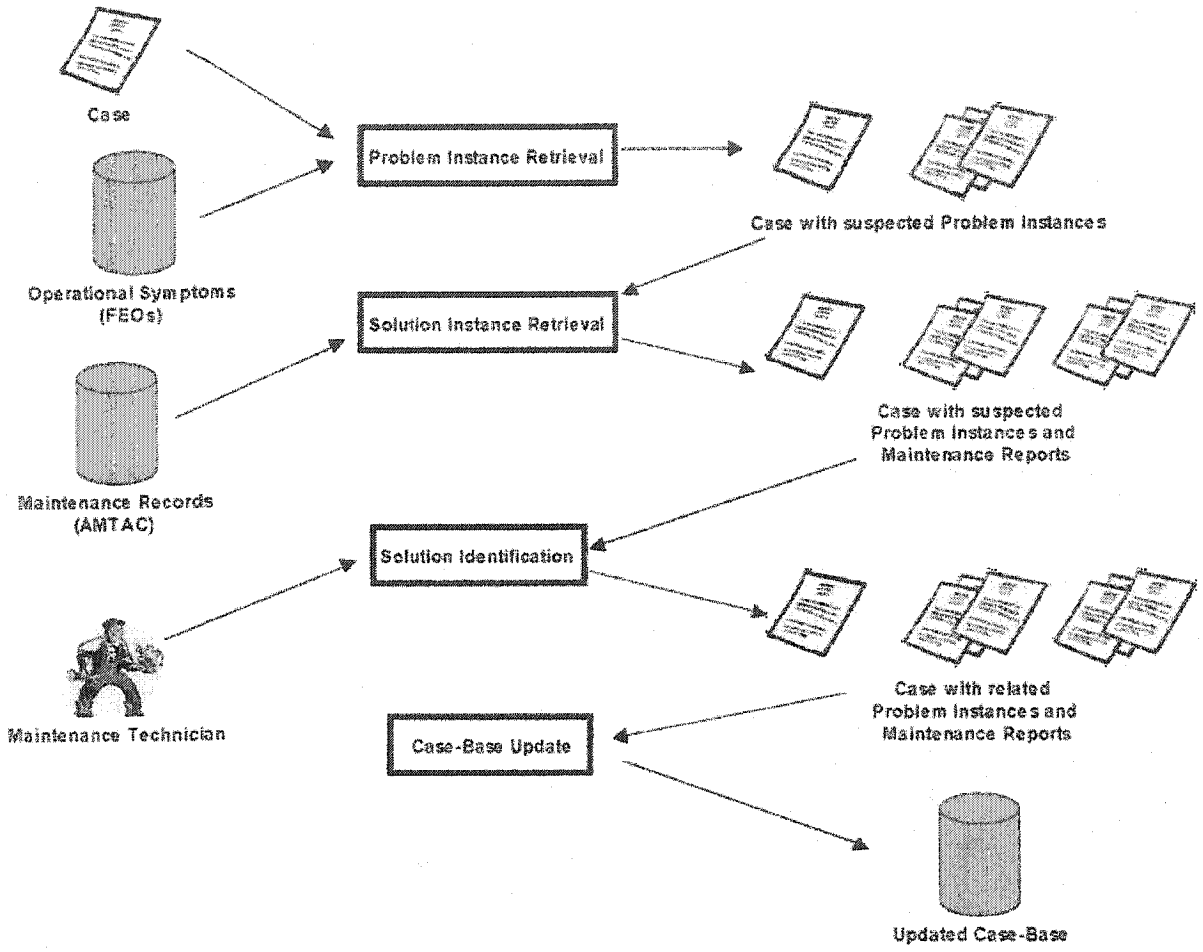

Figure 3: Steps involved in Case-Base Validation Stage

4.2.1 Historical Experience Resource Identification

Identification of historical experience resources is the evaluation of the resources available for case-base validation and is performed when implementing case-base validation. This step is performed once unless a new information resource is identified that is influential to the case-base validation stage. In analysis of the problem domain, two information resources used in case-base validation stage were the FEO database and the AMTAC database.

IDS stored the problem instances as FEOs in the FEO database. The information for these problems is found in a single table, FEO. The FEO table is contains 17 fields, which we will

consider 7 of these fields in the case-base validation stage. Table 1 outlines these fields and their data types.

Field Name	Data Type	Description
DEPARTING_STN	String	flight origination location
DESTINATION_STN	String	flight destination location
FLIGHT_NUMBER	String	flight number
FIN_NUMBER	String	aircraft identification
START_DATETIME	Date/Time	start of flight date and time
END_DATETIME	Date/Time	end of flight date and time
TSM_MESSAGES	String	related TSM messages occurring during the flight

Table 1: FEO table fields used in the case-base validation stage

The AMTAC reporting system is the information management system used by Air Canada to record maintenance activities on their fleets of aircraft. AMTAC uses a database to store information related to these maintenance activities. The case-base validation stage uses four of tables found in the AMTAC database. Four tables were identified as relevant for the case-base validation stage: ACFT_MI, ACFT_MI_ACTN, ACFT_PART_INST, and ACFT_PART_RMVL. The ACFT_MI table stored reported maintenance problems by the maintenance or aircrew staff. In this table there exists a problem description field, which contains the textual description of the problem. Each reported problem is assigned a unique id, EVT_DESC_OCCR_NBR and this id is used to join the information from the ACFT_MI_ACTN, ACFT_PART_INST, and ACFT_PART_RMVL tables. The ACFT_MI_ACTN table stores the information related to the maintenance actions performed to resolve the problem, which the action description field is important in determining the

solution identification in the case-base validation stage. The information stored in this action description field is the recorded text of actions performed by the maintenance technician. The ACFT_PART_INST and ACFT_PART_RMVL tables store the information related to the installation and removal of the parts used in the maintenance action. These tables contain fields that describe the part numbers and manufacturer information used for authenticating the cases in the solution identification step. Part numbers from the manufacturer and specialized Air Canada part codes reflect the different representations for part information. Now that the different information resources have been identified and analyzed, it is time to validate the cases in the case-base.

4.2.2 Problem Instance Retrieval

Problem Instance Retrieval uses the symptom set from the case to identify instances of problems that occur in the operation of the aircraft. First, an arbitrary case is retrieved from the case-base to initiate the validation process. Cases describe symptom sets. The symptom sets are used to identify FEO problem instances created by IDS. The FEO database is searched to obtain FEO problem instances related to a specific case. If no FEO problem instances were found then the case would not have been retrieved at any time during the aircrafts' recorded history. Cases record their associated FEO information for later processing. This FEO information includes the aircraft identification and the period of time when the symptoms occurred. Once the FEOs have been retrieved from the FEO database, they are clustered with respect to aircraft identity and time. Two FEOs that happen on the same aircraft within a 24-hour period of each other are considered to be the same problem to eliminate problems that are sporadic in nature. The result at the end of the problem instance retrieval process is a correlation between a case and suspected problem instances where this case could have potentially been applied.

4.2.3 Case Solution Identification

4.2.3.1 Solution Retrieval

The problem instances retrieved in the Problem Instance Retrieval step contain the aircraft's identity and the period of time when the problem occurred. This information is used to

construct a query that will retrieve all the repair documentation for the aircraft during the time of the problem. The repair documentation is in the form of AMTAC reports that are stored in the AMTAC database. The Solution Retrieval process uses the AMTAC database to retrieve suspected AMTAC reports that happened during the time of the problem instance. The query results in a collection of AMTAC reports related to a specific aircraft for the problem instance's period of time. Five fields in the AMTAC report establish its relevance to the problem instance: problem description, solution description, parts installed/removed, and temporal information. These suspected AMTAC reports are correlated to their respective problem instances. Since not all the retrieved AMTAC reports are related to the problem instance, the next step established the relevancy of each of the AMTAC reports to the problem instance.

4.2.3.2 Solution Identification

Solution Identification process reduces the collection of suspected AMTAC reports to those related to the problem instance. The symptoms, solution, and part information stored in the case is used to establish its relevancy. The first step was to compare the AMTAC part information to the IPC part information stored in the case. The information in the parts removed/installed fields of the AMTAC report helps establish the component replacement with confidence, but this field is not always filled in. Therefore, other fields in the AMTAC report must be processed in order to determine the component and action taken in the solution. The text from the AMTAC report was searched for the words established in the criteria for AMTAC report applicability. The text in these fields was processed using a *Bag of Words* (*BOW*) approach to determine similarity to the case's word criteria [Craven et al. 1998]. A BOW approach was used to determine the existence of terms used in both AMTAC report and the case's word criteria. In order to improve the accuracy of this second step, a stop list of words was manually created. This stop list contains frequently occurring words such as 'FAULT' and 'MSG'. The presence of a stop word is disregarded unless a non-stop word is present. Automatic processing of the AMTAC reports identifies the AMTAC report(s) that are relevant to the case's suspected problem instances. A final evaluation from the maintenance technician assures that the AMTAC messages are related to the problem

instance. This manual evaluation consists of the maintenance technician making a binary decision on whether the solution was applicable or not. After the case solution identification is completed, the case's problem instances contains correlated TSM fault isolation procedures to related AMTAC reports and these problem instances can be used to update the case-base.

4.2.4 Case-Base Update

The case-base update implements a strategy for evaluating the effectiveness of the case solution with respect to the case's problem instances. Each of the case's problem instances contains the solution recommended by the case. The related AMTAC report is used to establish the date and time for the repair. If the repair occurred after the presence of symptoms for the case disappeared then the repair was successful. If not then the repair was unsuccessful. The result of this evaluation is used to update the information inside the case. The case is updated in two ways. The first is the AMTAC report identifier and the results of the repair are added as additional information to the case. The second is the statistical information regarding the result of the application of the case is updated by adding one to the applicable success or failure count. Once the case information is complete, the case is modified in the case-base to reflect these historical experiences.

This semi-automated case authoring approach results in a case-base that is constructed from the TSM manual with enhancements from other manuals and historical experiences. The approach facilitates constructing a case-base from scratch or modifying an existing case-base to reflect changes in the aircraft's documentation. When an update to the TSM is issued, our approach is reapplied to update the existing case-base. Also, this method can be used as a complimentary approach to the other case authoring approaches implemented in IDS. A case-base can be created using this case authoring approach and can later be extended with the Automated Case Creation System in IDS [Yang et al. 2003]. This method eliminates problems that have solutions in the technical documentation and gives a computationally intensive focus on the undocumented problems that occur on the aircraft.

4.3 IE System using an NLP approach

Even though use of regular expressions in our Information Extraction system yielded a large amount of cases to authenticate; the regular expressions were not capturing all the cases and case information in the technical documentation. Further analysis of the text showed that tests and results that could be useful in describing symptoms for the case were not being extracted. This information could make the cases constructed from the TSM more precise in the retrieval process. Also, other maintenance actions, such as adjust, required further regular expression development to implement these other maintenance actions in the Information Extraction system. Since the regular expressions are developed manually, more effort is required to develop new regular expressions and integrate these into the Information Extraction system. This approach of using regular expressions within the Information Extraction system would quickly become unmanageable as more regular expressions are developed. Therefore, exploration into alternative broad coverage techniques to easily extract the extra tests, results, and cases from the texts was considered.

Natural Language Processing (*NLP*) techniques are useful in the implementation of Information Extraction approaches. [Brüninghaus and Ashley 2001] outline the limitations of statistical Information Extraction approaches, typically used for Information Retrieval. They provided arguments for applying more sophisticated NLP approaches to interpret the text in the Information Extraction task. Word order and clause negation were a couple of arguments in favour of implementing NLP techniques in Information Extraction [Brüninghaus and Ashley 2001]. One NLP approach we investigated was the use semantic grammars in the case-base creation stage.

Semantic grammars are grammars that identify syntactic and domain specific semantic constructs in a corpus [Allen 1995]. Semantic grammars are constructed to improve efficiency and performance for interpretation of semantic concepts within a specific domain [Allen 1995]. Since the case-base creation stage concentrates on the actions and components used as case solutions, a semantic grammar is developed to identify such constructs and other information related to the case. Similar to construction of syntactic grammars, semantic

grammars share similar challenges: construction of comprehensive lexicon and grammar rules. Semantic grammars possess an additional challenge of identifying domain specific semantic relationships inside the corpus.

Semantic grammars have a better understanding of the text through broad coverage parsing, clause negation, and anaphora resolution. Our case authoring approach can benefit from the implementation of NLP approaches through linguistic analysis. Therefore, semantic grammars should be investigated to determine the viability of such an approach to IE. A semantic grammar is more robust than regular expressions and is easier to maintain than a large group of specialized regular expressions. In order to capture the information in the tests and results, regular expressions would have to be developed for each identified pattern in the text. Because each of these regular expressions is manually developed from the text, linguistic information that could be helpful in generalizing some these regular expression patterns is not considered. Semantic grammar rules would be able to use this linguistic information to represent rules that are more general in nature. This generalization in the semantic grammar rules led to the reduction in the amount of rules needed to cover all the text, thus making the overall Information Extraction system more maintainable. The developed semantic grammar also yields cases that are more precise due to the extra tests and results that can be extracted. Clause negation is when a positive clause's context is made the opposite with the insertion of 'not'. The following two TSM sentences are examples of clause negation.

If the filter is not clogged:

If the recirculation fan (15HG) does not operate:

Clause negation in the context of our case authoring approach represents the absence of symptoms in the problem description. Like in the medical domain, it is important to represent clausal negation accurately in the case authoring process [Bellazzi et al. 1999]. Finally, anaphora resolution is the correlation between pronouns like 'it' to an entity that was referenced earlier in the text [Jurafsky and Martin 2000]. Anaphora occurred quite frequently

inside the TSM corpus in the form of pronouns 'it' and noun phrases 'the filter'. Anaphoric resolution is important in building cases that describe the problem description and solution accurately. The approaches taken to resolve anaphoric reference was take the first aircraft component reference related to the anaphora. This outlines some of the challenges that semantic grammar must address in our case authoring process. The Information Extraction system implementing a semantic grammar used three steps to processing the text from the TSM: pre-processing, grammar development, and post-processing.

4.3.1 Pre-processing

The pre-processing stage is the first step of authoring cases from text documents. The TSM HTML pages are used as input for this stage. This process takes the list of components identified in the Possible Causes section and generated representation as Prolog facts. These components will become an important part of our lexicon used for identifying aircraft components inside the TSM page.

The second part of the Pre-processing stage takes the Fault Confirmation and Fault Isolation sections and lifts the text from these sections. The order of the lines in the text, the grouping of sentences in the lists, and order of the sentences are important features of the corpus that we need to preserve. The lifted text is used to generate Prolog fact representation. Other details about which section the text is from, the order that the text appears, and the level of list indentation of the text is preserved in the Prolog fact. This information is extracted using the HTML tags. Finally, tags that do not add information to the text are filtered out from the text. The result of the Pre-processing phase is a file that contains Prolog facts for the aircraft components and text from the TSM page. This file can be consulted from within a Prolog environment. Below is a sample of the Prolog facts generated.

```
aircraft_component('FILTER-RECIRCULATION (4012HM)').

aircraft_component('FAN RECIRCULATION (15HG)').

aircraft_component('FILTER-RECIRCULATION (4013HM)').

aircraft_component('FAN-RECIRCULATION (14HG)').
```

```
sentence('Do the operational test of the cabin recirculation
      fans 14HG and 15HG AMM TASK 21-21-00-710-001 .', 'Fault
      Confirmation', 1, 1).

sentence('If the test gives the maintenance message RECIRC FAN
      1 OR SPLY:', 'Fault Isolation', 2, 1).

sentence('make sure that the recirculation filter (4012HM)
      housing and the adjacent area is clean.', 'Fault
      Isolation', 3, 2).
```

4.3.2 Grammar Development

The TSM page contains important information about the tests and conditions that isolate the problem. Also, the TSM page recommends repair actions on components that will fix the problem given a set of symptoms. This means that we must represent these semantic concepts inside our semantic grammar. Three major semantic concepts were conditions, actions, and wiring lists. Finally, the semantic grammar identified other grammatical constructs that are useful in case construction.

4.3.2.1 Conditions

The conditional semantic concept contains information about tests and results of tests that isolate the existing problem. Many of the conditions start with a conditional subordinator 'if'. Conditional subordinators are clauses that convey a direct condition and 'if' is a common conditional subordinator [Quirk et al., 15.18, 1990]. [Quirk et al. 1990] states 'if' can be followed by a finite, nonfinite, or verbless clauses. The TSM has examples of conditional clauses using finite and verbless clauses.

A typical example of a conditional clause using a finite clause is shown below:

If the filter is clogged:

A finite clause is a clause that contains a finite verb [Quirk et al., 3.19, 1990]. Finite verbs can take the *-s* or past tense form [Quirk et al., 3.3, 1990]. The verbs in the form of BE are special because it identifies that it is functioning as a copular verb. The verb BE is a copular verb when it connects the subject of a clause to a predicate nominal or an adjective [Jurafsky and Martin, 2000]. The noun phrase 'the filter' is the subject of the clause. 'Clogged' is the past participle of the verb 'clog' so it is an adjective. The semicolon is important in the clause because it marks the end of the condition, therefore the semicolon must be included in the semantic grammar. This leads to the following representation of the semantic grammar rules that apply to the above example.

TSM Sentence → TSM Condition, ':'
TSM Condition → Conditional Subordinator, TSM Noun Phrase, TSM Verb Phrase
Conditional Subordinator → 'if'
TSM Noun Phrase → Determiner, TSM Nominal
Determiner → 'the'
TSM Nominal → TSM Noun
TSM Noun → 'filter'
TSM Verb Phrase → Copular Verb, TSM Adjective
Copular Verb → 'is'
TSM Adjective → Verb Past Participle
Verb Past Participle → 'clogged'

Another example of a conditional clause using a verbless clause is shown below:

... if necessary.

A verbless clause is a clause that has omitted the verb usually in the form of BE [Quirk et al., 14.6, 1990]. The subject is usually recovered from the context of the surrounding clauses. In our example adding the subject and verb 'it is' does not alter the meaning of the clause. 'Necessary' is an adjective describing the pronoun 'it'. The following is the representation of the semantic grammar rules that apply to this TSM condition.

TSM Action → Imperative Verb Phrase, TSM Condition, Manual Reference
TSM Condition → Conditional Subordinator, TSM Adjective
TSM Adjective → 'necessary'

4.3.2.2 Actions

The second major semantic concept of the TSM is the action. The action concept describes the actions performed in the procedure and the aircraft components that are affected by the action. The clauses associated with actions are typically imperative. An imperative clause is a clause with an implied subject and the verb generally is in its base form [Quirk et al., 11.15, 1990]. The following is an example of a TSM action.

remove the FILTER-RECIRCULATION (4012HM) AMM TASK 21-21-41-000-001 .

As we can see the subject is not stated in the above clause. This means that 'you will' is implied as the subject for the clause. Now, looking at the word 'remove', we can identify that this is a verb in its base form. Imperative clauses follow the same grammar rules for the predicate clause as declarative clauses. The predicate clause in the example has two parts to it. The first part is the object that the verb applies to, 'the FILTER-RECIRCULATION (4012HM)'. The second part refers to the AMM reference for the removal procedure. These two predicate parts must be addressed in the TSM semantic grammar. 'The FILTER-RECIRCULATION (4012HM)' is a noun phrase. As mentioned before a noun phrase can start with a determiner followed by a noun. 'FILTER-RECIRCULATION (4012HM)' is identified not only as a noun, but also as an aircraft component identified in the Possible Causes section of the TSM page. From the Pre-processing stage of the project, this string of words will be identified in the lexicon as an aircraft component. Finally, the manual reference, AMM TASK 21-21-00-000-001 must be identified as another semantic concept of the TSM. The first word, AMM, identifies the manual to which the reference refers. In this case the manual is the Aircraft Maintenance Manual. This identification of the manual reference will constrain the remaining words in the reference. The second word, 'TASK', is particular to the AMM reference. The third word is the number reference. This number reference must be a specific format as described by the manual it refers to. Here are the semantic grammar rules that apply to the previous action example.

TSM Action → Imperative Verb Phrase, Manual Reference
Imperative Verb Phrase → Imperative Verb, TSM Noun Phrase
Imperative Verb → Verb Base Form
Verb Base Form → 'remove'
TSM Noun Phrase → Determiner, TSM Nominal
TSM Nominal → TSM Component
TSM Component → 'FILER-RECIRCULATION (4012HM)'
Manual Reference → 'AMM TASK 21-21-41-000-001'

Another example of a TSM action is:

replace it if necessary.

The TSM action contains a TSM condition embedded after the action and our semantic grammar must address this issue. There is a similar structure in that the clause begins with an imperative verb, 'replace'. The pronoun 'it' serves as a TSM placeholder for the aircraft component referenced in earlier context and thus makes it an anaphoric reference [Jurafsky and Martin, 2000]. 'It' is defined as a pronoun that can be used to represent noncount concrete nouns [Quirk et al., 6.9, 1990]. The zero article is used with 'it' [Quirk et al., 11.15, 1990]. The resolution of this anaphoric reference is handled in the Post-processing step of this Information Extraction system. The TSM condition is processed as mentioned in an earlier paragraph. The following TSM semantic grammar rules were developed to apply to this example.

TSM Action → Imperative Verb Phrase, TSM Condition, Manual Reference
Imperative Verb Phrase → Imperative Verb, TSM Noun Phrase
Imperative Verb → 'replace'
TSM Noun Phrase → TSM Nominal
TSM Nominal → TSM Placeholder
TSM Placeholder → 'it'

4.3.2.3 Wiring

The third major semantic category is the TSM clauses that contain the list of wires that connect the components together. Below is a small sample from the TSM

RELAY (16HG) connector A/X and SW (4HG) connector A/C2, RELAY (16HG) connector A/Z and GND ASM 21-21/01 and ASM 21-63/01.

The form of the TSM clause is not that of a normal English grammar, but our TSM semantic grammar identifies the different components and wiring between these components. The connection between the two aircraft components are conjoined by the word 'and'. The phrases before and after the conjunction are noun phrases that describe each end of the connection. According to Quirk et al., noun phrases can be conjoined by a conjunction, which lead to another definition of the noun phrase [Quirk et al., 13.21, 1990]. A noun phrase is composed of an aircraft component, 'RELAY (16HG)', followed by one or more nouns, 'connector A/X'. This piece of information can be useful in determining that the 'RELAY (16HG)' has a connector associated with it labelled 'A/X'. The TSM clause has one or more connections described and commas separate these connections. This comma separator is another punctuation mark that has to be considered when building the TSM semantic grammar. Below is the TSM semantic grammar rules that can be applied to the wiring list concept.

TSM Wiring → TSM Noun Phrase, ',', TSM Wiring
TSM Wiring → TSM Noun Phrase, Manual Reference

TSM Noun Phrase → TSM Nominal, Conjunction, TSM Noun Phrase
TSM Nominal → TSM Component, TSM Nominal
TSM Nominal → TSM Noun, TSM Nominal
TSM Component → 'RELAY (16HG)'
TSM Noun → 'connector'
TSM Noun → 'A/X'

4.3.2.4 Other Grammatical Constructs

Some other grammatical constructs that were addressed were conjunctions, prepositions, negation, and other special cases. As mentioned earlier, noun phrases can be joined together using the conjunction 'and'. Since TSM actions are similar to clauses, they can be conjoined

together using conjunctions [Quirk et al., 13.9, 1990]. The following rule was created in order to address the conjunctions and TSM actions.

TSM Sentence → TSM Action, Conjunction, TSM Sentence

The TSM contains prepositional phrases that must be identified with the TSM semantic grammar. The following is an example of a prepositional phrases found in the TSM procedure.

do a check of the RELAY-L FAN POWER (5HG)

This example demonstrates the post modification of the noun phrase 'a check' using a prepositional phrase with 'of' [Quirk et al., 5.49, 1990]. This situation is sometimes referred to as the 'of-genitive'. We will focus on the prepositional phrase, 'of the RELAY-L FAN POWER (5HG)'. As previously stated the start of the prepositional phrase starts with a preposition. This is followed by a noun phrase. The following rule was added to the TSM semantic grammar to address the issue of prepositional phrases.

Imperative Verb Phrase→ Imperative Verb, TSM Noun Phrase, Prepositional Phrase
Prepositional Phrase → Preposition, TSM Noun Phrase
Preposition → 'of'

Another grammatical construct handled by the TSM semantic grammar is clause negation. Clause negation is when a positive clause's context is made the opposite with the insertion of 'not'. The following two TSM sentences are examples of clause negation.

If the filter is not clogged:
If the recirculation fan (15HG) does not operate:

Clause negation can be achieved by inserting 'not' between the operator and the predicate. The operator in a clause is first or only auxiliary in a finite verb phrase. In the first example

the verb BE functions as the operator and 'not' is inserted after 'is'. In the second example, there is no operator of the positive clause, 'If the recirculation fan (15HG) operates'. This means that the dummy operator DO is inserted before the verb 'operate' and 'not' is inserted after the operator [Quirk et al., 10.33, 1990]. The following rules are used to identify negation in the TSM page.

TSM Verb Phrase→ Copular Verb, Negation, TSM Adjective
TSM Verb Phrase → Verb, Negation, TSM Verb Phrase
Copular Verb → To Be Verb
To Be Verb → 'is'
Negation → 'not'

These rules work with the TSM page, but better rules would incorporate the use of the operator.

Two cases of the TSM sentences were not addressed in [Quirk et al., 1990], but the English grammar rules for these TSM sentences were identified in [Quirk et al, 1985]. The first sentence below is an example of a collocation of two words, 'make sure', which identifies this special unique situation.

make sure that the recirculation filter (4013HM) housing and the adjacent area is clean.

Collocation is a quantifiable position-specific relationship between two words [Jurafsky and Martin, 2000]. The verb phrase 'make sure' is like a bare stem and is followed by a that-clause [Quirk et al., 16.45, 1985]. A that-clause is a clause that starts with 'that'. The following TSM semantic grammar rules were developed to address this type of TSM sentence.

Imperative Verb Phrase → Imperative Verb Phrase Make Sure, That Clause
Imperative Verb Phrase Make Sure → 'make sure'
That Clause → 'that', TSM Noun Phrase, TSM Verb Phrase

The TSM sentence below is another example of a special case that needs special attention. The sentence has a clause embedded inside identified by the parenthesis. The clause inside the parenthesis has two preposition phrases where the first preposition is without a subject.

> *replace the FAN RECIRCULATION (15HG) (referred to as FAN (15HG)) AMM TASK 21-21-51-000-001 and AMM TASK 21-21-51-400-001 .*

At first, the first two words were referenced as a phrasal verb. A phrasal verb is a collocation of two words that would have another grammatical function when separated. For instance, 'I wrote up the report' uses the phrasal verb 'wrote up'. 'Wrote up' when used together describes a creation of the report. In this case 'up' is not used as a preposition to describe the direction to which I wrote. In the above example, 'referred to' can be a suspect for a phrasal verb because its present tense, 'refer to' is listed as a phrasal verb [Quirk et al., 1985]. The simple test of switching the second word in the phrasal verb to the end of the sentence can determine if the word is adverbial [Quirk et al., 7.55 1985]. In our phrasal verb example, this test confirms that the word, 'up', is adverbial, 'I wrote the report up'. Therefore, the word combination 'referred to' is a verb, 'referred', followed by a preposition, 'to'. This leads to the problem that the phrase, 'to as' is two prepositions in sequence. This is correct grammar because the object being referred to, 'it', has been omitted [Quirk et al., 15.29, 1985]. The preposition, 'as' refers to a comparison and in this case the comparison is one of equality [Quirk et al., 9.3, 1990]. The following are the TSM semantic grammar rules that are applied to the above TSM clause.

> TSM Action → Verb Phrase Imperative, '(', TSM Component Reference, ')', Manual Reference
> TSM Component Reference → Verb Past Tense, Prepositional Phrase, Prepositional Phrase
> Prepositional Phrase → 'to'

This concludes the TSM semantic grammar development in the Information Extraction system for case-base creation stage. Even though more development is needed for a robust,

flexible, and complete solution, this is a foundation for continued development as described in the Future Work section.

4.3.3 Post-processing

Once the parse tree has been generated by the semantic grammar in the previous stage, a case-base can be constructed from these generated parse trees. The cases are constructed using the IDS rule set for the diagnostic messages related to the problem description and the tests and results of those tests are added from the semantic grammar parse tree. The identified actions and components in the semantic parse tree are used in describing the solution for the case. The structure in the fault isolation procedure, which was preserved in the Prolog facts, must be taken into consideration when constructing each individual case. When an aircraft component and action are identified, previous statements belonging to the same troubleshooting logic path are processed to identify tests and results that can further justify this solution. This information was previously not encapsulated within the case using the regular expression implementation for IE. Representation of this new information must be accommodated within the case structure and the retrieval process must also take this information into consideration when retrieving cases for a problem. Also, since many of these observable symptoms are not automatically recorded, the maintenance organization must consider a process of collecting this information in order for it to be useful in case-base retrieval. Finally, if the component is an anaphoric reference, this reference must be resolved before the case is saved in the case-base. When an anaphoric reference is encountered, the previous statements in the troubleshooting logic are examined to make a good prediction on an aircraft component. Special logistics must be taken into account when assigning a component to an anaphor. A common anaphor used was 'it' and the technique used to resolve the reference was search for the nearest possible component reference before the anaphor. Another anaphor was a noun phrase, 'filter', where the nearest possible component is a filter component. Any other components that occur in between the anaphor and the filter component are disregarded due to the lack of reference to a filter component.

Regardless of the Information Extraction implementation, a case-base exists that represents knowledge from within the different textual resources, but no historical information can be inferred from these resources. The Case-Base Validation stage authenticates the case-base with historical information from the maintenance organization.

5 Experimentation Results

The case authoring approach was used to create a case-base for the Airbus A320/A319 aircraft. The latest TSM and IPC manuals were used in the case creation stage. The case-base validation stage used over six years of historical data from the AMTAC reporting system to authenticate cases in the case-base. This approach resulted in a case-base that could be used by an airline's maintenance technician in IDS. We show three results: two from the stages in the approach and an example using a specific TSM fault isolation procedure.

Our case authoring approach was evaluated using two collections of TSM fault isolation procedures. The first collection of TSM fault isolation procedures was constructed from TSM fault isolation procedures that contain many cases and these identified cases had significant historical evidence of their problem symptoms. Significant evidence is defined as case symptoms that had over 1000 occurrences in the FEO database. The second collection of TSM fault isolation procedures contained cases that exhibit the five most frequently occurring case symptoms determined from the operation of the aircraft. This collection is representative of the five most reoccurring problem symptoms for the aircraft in the fleet. Identification of TSM fault isolation procedures for both collections was based on the results of the case-base creation and validation stages. Once the identification of the TSM fault isolation procedures for each of the collections was done, the text was manually processed to extract cases. This was used as an answer template for later evaluation of the case-base creation Information Extraction systems. Finally, these two case collections were used to evaluate the performance of both case-base creation Information Extraction systems. Using measurements outlined in Section 3.1.4, precision, recall, and F-measure are calculated for each Information Extraction system used in the case-base creation stage.

5.1 Case Creation Results

5.1.1 Regular Expression Information Extraction Implementation

The first experiment was done with the Information Extraction system implementing regular expressions. The two collections of cases were processed to determine precision, recall, and

F-measure. The answer template obtained earlier in Section 5 was used to construct the cases that were manually identified in the text. The results of performing case-base creation on the two collections of fault isolation procedures are outlined in Table 2. The precision for the first collection of TSM fault isolation procedures was 0.96, the second, 1.0. The combined precision from the two collections was used to determine the overall performance of the Information Extraction system; the result was 0.97. Similarly, recall was evaluated for the first collection and the result was 0.27. The second collection yielded a recall measure of 0.48 and the overall recall of this Information Extraction system was 0.30. F-measure was used to combine precision and recall and the overall F-measure for the Information Extraction system with regular expression was 0.46.

Collection	Manually Identified Cases	Regular Expression extracted Cases	Precision	Recall	F-measure
1	164	44	0.956522	0.268293	0.419047619
2	29	14	1.0	0.482759	0.651162791
Total	193	58	0.966667	0.300518	0.458498024

Table 2: Case-base creations stage results for regular expression Information Extraction implementation

The overall precision of this Information Extraction system was quite high compared to results from state of the art Information Extraction systems used in the MUC conferences [Cowie and Lehnert 1996]. The overall recall of the Information Extraction system was lower than the best Information Extraction systems used in the MUC conferences [Cowie and Lehnert 1996]. The F-measure was lower than the state of the art F-measure value of 0.6 established by [Applet and Israel 1999]. This lower F-measure was the result of a low recall measurement; even though the precision measurement was almost perfect.

The case creation stage resulted in a case-base from the TSM and IPC with over 10,000 cases and the cases extracted from the two collections of TSM fault isolation procedures are shown

in Appendix B. The distribution of the three regular expression frequencies can be seen previously in Table 3. The most frequent applied regular expression was the regular expression '(replace) the *.' with 10,608 occurrences. The other two regular expressions are specialized for specific situations and do not yield the same amount of coverage as the first. These regular expressions capture a large number of cases from the TSM, but this scanning approach does not provide complete coverage for extracting cases represented in the TSM. The word 'replace' has 15, 013 occurrences in the TSM, in which 4,332 are not captured using these regular expressions. These specialized instances are outlier situations and additional regular expressions are required to capture these cases.

ID	Regular Expression	Application Frequency
1	/(replace) the (.*)/i.	10,608
2	/do a check of the (.*) and (replace) it/i.	26
3	/make sure that the (.*) is not clogged. If necessary, (replace) it/I	11

Table 3: Regular expressions used in the case-base creation stage

Recall could be improved through development of regular expressions representing other regularly used action terms. Since regular expressions would have to be developed for every usage pattern for each repair action in the TSM fault isolation procedure, the maintenance and adaptation of such an Information Extraction system implementing only regular expressions would be unfeasible. A more versatile approach to Information Extraction would be to implement NLP techniques to provide more robust and broader coverage when processing a problem domain. An investigation into the development of an Information Extraction system implementing a semantic grammar was done and the results presented in the following section.

5.1.2 Semantic Grammar Information Extraction Implementation

The second approach to Information Extraction for the case extraction stage implemented a semantic grammar and this section will describe the results from such an approach. The two

collections of TSM fault isolation procedures that were used in the previous evaluation of the Information Extraction system with regular expressions were used to evaluate the Information Extraction system with the semantic grammar. Two results are presented, the first is the coverage of the semantic grammar using these pages as input and the second result is the performance of this Information Extraction system with respect to precision, recall, and F-measure.

TSM Fault Isolation Procedure	Number of Sentences	Number of Parse Trees Generated	Sentences Missing Words in Lexicon
21-21-00-810-801	50	49 (98%)	1 (2%)
21-26-00-810-803	32	16 (50%)	16 (50%)
21-26-00-810-804	32	16(50%)	16 (50%)
22-83-00-810-817	6	2 (33%)	4 (67%)
22-83-00-810-818	6	2 (33%)	4 (67%)
22-83-00-810-863	27	9 (33%)	18 (67%)
28-21-00-810-807	56	7 (13%)	48 (86%)
28-21-00-810-813	49	8 (16%)	38 (76%)
28-21-00-810-814	49	8 (16%)	38 (76%)
28-21-00-810-817	12	3 (25%)	7 (58%)
28-21-00-810-818	12	3 (25%)	8 (67%)
28-21-00-810-819	26	1 (4%)	25 (96%)
28-21-00-810-823	165	10 (6%)	144 (87%)
31-32-00-810-822	9	4 (44%)	5 (56%)
34-43-00-810-801	35	2 (6%)	32 (91%)
Total	566	140 (25%)	404 (71%)

Table 4: Semantic grammar coverage

First, the results from the evaluation of the coverage of the semantic grammar on the collections of TSM fault isolation procedures is outlined in Table 4. The highlighted row in Table 4 is the TSM fault isolation procedure used to create the semantic grammar. As expected, the coverage is very high, 98%. The TSM fault isolation procedures were used to

further develop the semantic grammar both in lexicon and grammar rules. The development of the semantic grammar focussed on the extraction of TSM actions and components in the TSM fault isolation procedure. As exhibited in Table 4, some TSM fault isolation procedures, 28-21-00-810-819 and 34-43-00-810-801, had relatively low coverage. The semantic grammar parser is unable to parse a sentence in the TSM fault isolation procedure when no sematic grammar rules apply or when the parser encounters a word in the sentence that is not represented in the lexicon. In the above table, the TSM fault isolation procedures that have poor coverage also have words that are not represented in the lexicon. In order to better assess the performance of the semantic grammar rules, we must provide the sematic grammar parser with a more complete lexicon. Section 7.1.1 describes some potential solutions for constructing a more complete lexicon that could be used by the semantic grammar parser. Further development of the semantic grammar is needed to accomplish broader coverage and better understanding of the text. Even though full understanding of the text is not accomplished, [Hobbs et al. 1997] state that full text understanding is not required for Information Extraction and the next results show that this coverage is sufficient to construct cases from the text.

Table 5 shows the second result from evaluating the semantic grammar Information Extraction system with precision, recall, and F-measure criteria. The semantic grammar Information Extraction system constructed 74 cases from the two collections of TSM fault isolation procedures. From this data, all cases created were legitimate, thus the overall precision was calculated at 1.0. The recall measure for the two collections varied slightly where Collection 1 was 0.384 and Collection 2, 0.379. The overall recall for this Information Extraction system was calculated and the result was 0.383. These precision and recall measurements were used to calculate the overall F-measure for the semantic grammar Information Extraction system. The F-measure for the semantic grammar Information Extraction system was 0.554. This F-measure was slightly lower than the measurement for state of the art F-measure outlined by [Appelt and Israel 1999], but the semantic grammar Information Extraction system resulted in a higher F-measure than the regular expression

Information Extraction system. Therefore, the Information Extraction system that uses the semantic grammar was the better system to implement.

Collection	Manually Identified Cases	Semantic Grammar extracted Cases	Precision	Recall	F-measure
1	164	63	1.0	0.384146	0.555066079
2	29	11	1.0	0.37931	0.55
Total	193	74	1.0	0.38342	0.554307116

Table 5: Case-base creation stage results for semantic grammar Information Extraction implementation

5.2 Case Validation Results

The case-base from the regular expression Information Extraction system was used to produce the results for the case-base validation stage. Three results are observed from the following steps in the case-base validation stage: problem identification, solution identification, and case-base update.

5.2.1 Problem Identification Results

The first result from the problem identification step describes the distribution of the case symptoms with respect to the problems experienced by the maintenance organization. One issue is to determine what amount of the TSM is used in the everyday troubleshooting of aircraft. Table 6 displays the distribution of the number of problem symptom set occurrences with respect to the number of cases with that occurrence. Table 6 shows that a majority of the cases represented by the TSM have never occurred and these cases make up 81% of the case-base. The cases that have their symptoms present in the operational data of the maintenance organization are fewer than 19%. Further investigation is needed to confirm if these cases were actually applied. This result is not surprising because the TSM should possess more comprehensive knowledge than what is experienced in the maintenance organization. Some of these unused cases may be more applicable in different situations, for

instance when the aircraft gets older and these constraints must be taken into consideration in case retrieval.

Case Symptoms Frequency	Number of Cases
0	10085 (81.01%)
> 0 and < 100	1859 (14.97%)
>=100 and < 1000	404 (3.25%)
>=1000	95 (0.77%)
Only ~19% of Case Symptoms have Frequency > 0	

Table 6: Distribution of Experience with respect to TSM Case Coverage

Collection	Problem Instances	FEOs	AMTAC Messages
1	21326	32355	74570
2	4928	13432	37161
Total	26254	45787	111731

Table 7: Identified problem instances for each case collection

The second result from the problem identification step is the identification of frequently occurring problems on the aircraft. Out of the 19% of cases that have operational experience, 4% of those have symptoms that occur more than a thousand times in a six year period. One could quickly construct a list of problems that have occurred frequently in the past and document these problems for future reference. Also, maintenance technicians and data mining can better focus investigative efforts by using this list of frequently occurring problems. Finally, if no extensive domain expertise exists, this list could be used as a benchmark for the maintenance organization. This list of frequently occurring problems was used to evaluate the solution identification process in the case-base validation stage. The five most frequently reoccurring problems were used to identify cases that could be authenticated with historical data. This information on frequently occurring problems was used to determine the cases that would be included in each of the collections of cases.

The final result from the problem identification step is the identification of problems that occurred with respect to the two collections of TSM fault isolation procedures. Table 7 shows a summary of the number of problem instances identified for each collection. More detailed information for specific case retrieval symptoms are outlined in Appendix C.

5.2.2 Solution Identification Results

The problem instances retrieved from the problem identification step are used to retrieve the AMTAC messages related to the problem instance with respect to time. Table 7 shows that 74,570 AMTAC messages were retrieved for the problem instances related to Collection 1 and 37,161 for Collection 2. The results of the automated solution identification using the BOW approach for filtering out unrelated AMTAC messages are seen in Table 8. This automated filtering resulted in a reduction of approximately 90% of the AMTAC messages. Further manual validation by the domain expert resulted in another 98.5% of the messages being eliminated. This left 138 AMTAC messages to be used to update the cases in the case-base.

Collection	Automatic Solution Identification	Manual Validation	Precision	Recall	F-measure
1	2313	68	0.029399049	1.0	0.057118858
2	27561	173	0.006276986	1.0	0.012475662
Total	29874	241	0.008167637	1.0	0.016202935

Table 8: Case-base solution identification results

The solution identification step was seen as an *Information Retrieval* (*IR*) task. An Information Retrieval task is defined as storage of media in a way that it can be efficiently and accurately retrieved based on user queries [Jurafsky and Martin 2000]. Two measures for accuracy and relevance of Information Retrieval are precision and recall [Jurafsky and Martin 2000]. Since measures for Information Extraction and Information Retrieval are similar, we will calculate the precision, recall, and F-measure for the automated solution identification process. The precision was 0.008 and recall, 1.0. Event though the overall

recall was 1.0, the precision was very low and needs to be improved to reduce the amount of time needed in manually validating the retrieved AMTAC messages. Since the identification and interpretation of a solution within an AMTAC reports is a difficult task, this difficult challenge is reflected in the low F-measure and precision scores for the overall system. Using the data from Table 7 and Table 8, the total number of initially retrieved maintenance messages from the AMTAC database was 111,731 records. Using part information in the case and AMTAC messages, the Automated Solution Identification step reduced the amount of AMTAC messages to 29,874. 74% of the AMTAC messages were automatically filtered out using the Automated Solution Identification step and this reduced the amount of AMTAC messages that would need to be processed manually by the expert. Section 7.1.2 outlines further improvements needed to achieve a more precise retrieval mechanism for the case-base solution identification step.

5.2.3 Case-Base Update Results

All the cases from the two collections of TSM fault isolation procedures were updated. The case-base update step enhances the case with references to experiences and the results of applying the solution in the case to solve the problem instance. A distribution of historical examples for each TSM fault isolation procedure collection is outlined in Table 9. The table displays number of cases represented in each of the different collections and the number of cases in that collection with successful/unsuccessful applications of the case solution. Finally, the last two columns show the number of successful/unsuccessful examples of applied case solutions that were used to solve identified problem instances in the fault isolation procedure collections. Table 9 shows that a majority of the examples used for authenticating the cases were successful and specific details for each case is outlined in Appendix D.

Collection	Cases	Cases with Successful Application	Cases with Unsuccessful Application	Successful Application	Unsuccessful Application
1	68	9	5	61	7
2	173	24	10	161	12
Total	241	33	15	222	19

Table 9: Case-base validation results

5.3 Example

5.3.1 Case-Base Creation

This section presents a detailed example of the extraction procedure. The TSM fault isolation procedure 22-83-00-810-849 is used to demonstrate the viability of our case authoring approach. This TSM fault isolation procedure was used to demonstrate the viability of the case authoring approach before the two collections of TSM fault isolation procedures was established.

According to the IDS rule set, this fault isolation procedure must be applied whenever the aircraft generates the failure message "AFS BSCU2". Below is sample text from the fault isolation procedure where the actions and components are highlighted by the applicable regular expression patterns.

A. If the test gives the maintenance message AFS: BSCU2 (ISSUED BY: FG1):
   - **replace** the **FMGC-1 (1CA1)**
     AMM TASK 22-83-34-000-001 and AMM TASK 22-83-34-400-001 .
     1. If the fault continues:
        - **replace** the **BSCU (10GG)**
          AMM TASK 32-42-34-000-001 and AMM TASK 32-42-34-400-001.
     2. If the fault continues:
        - do a check and repair the wiring of the BSCU OPP VALID COM and BSCU OPP VALID MON discretes (from the FMGC 1 (1CA1) to the BSCU (10GG)) ASM 22-85/04 .

B. If the test gives the maintenance message AFS: BSCU2 (ISSUED BY: FG2):
   - **replace** the **FMGC-2 (1CA2)**

In this example, three cases were constructed from the text. The TSM fault isolation procedure describes four different solutions and this Information Extraction system missed one. The precision for the example is 1.0 and the recall, 0.75. The calculated F-measure was 0.428 and this was sufficient for proof of concept.

5.3.2 Case-Base Validation

Symptoms	Action	Component	Successful Application	Unsuccessful Application
"AFS BSCU2"	Remove/Install	FMGC-1 (1CA1)	1	0
"AFS BSCU2"	Remove/Install	BSCU (10GG)	11	5
"AFS BSCU2"	Remove/Install	FMGC-2 (1CA2)	2	0

Table 10: Results of cases after case-base validation stage

Table 10 displays the three cases that were extracted from the fault isolation procedure 22-83-00-810-849 after the completion of the case extraction process. The symptoms for these cases were used to retrieve 931 problem instances from the FEO database. The problem instance information was used to retrieve 2990 AMTAC reports from the AMTAC database. The case validation stage identified 19 of these AMTAC messages as solutions to 19 problem instances. Table 10 outlines the results after each of the 19 solutions were evaluated against the occurrence of its related problem instance. Even though replacement of the FMGC-2 is referenced after the other two cases in the text, the structure of the troubleshooting logic suggests that this solution occurs at the same level as the replacement of the FMGC-1. The maintenance message that results from the test differentiates between the two different troubleshooting paths and any historical statistics recorded from these cases would indicate the troubleshooting path frequency. Currently, the troubleshooting logic recommends the replacement of the FMGC-1 before the replacement of the BSCU, but our case authoring approach uncovers that this historically does not happen. We were thus expecting to see 16 unsuccessful replacements of the FMGC-1, but the results show no unsuccessful applications of that case. Therefore, there must be additional knowledge besides

the manuals that the maintenance organization uses to make decisions. This additional information is not reflected within the IDS rule set, but it can be captured in the case-base. The historical statistics on the past applications of a case is additional information that can be used by the maintenance organization to help make decisions on current problems.

The first method for ranking the retrieved cases was based only on the accuracy of previous performances, but cases with few successful applications were ranked higher than more frequently applied cases with some unsuccessful results. Table 10 exemplifies this situation where the replacement of FMGC-1 and FMGC-2 would be considered before a more frequently occurring case with the BSCU replacement. This first method did not take into account the frequency in which each case occurs. Therefore, confidence and support measures for association rules were used to re-evaluate the ranking of the results in Table 10 [Agrawal et al. 1993]. The confidence measure for the cases with replacement of FMGC-1 and FMGC-2 is 100% and better than the case with BSCU replacement, 68.75%. Nonetheless, we should also consider the frequency with which each case occurs. The frequency is reflected in the support measure, which is of 78.57% in the BSCU replacement case, 7.15% in the FMGC-1 and 14.28% in the FMGC-2 cases. We combine confidence and support into an F-like measure and the values are 73.33%, 13.33%, and 25% for the BSCU, FMGC-1, and FMGC-2. This F-like measure suggests that BSCU replacement should be ranked higher than the other two cases, despite its lower confidence. This historical information was influential in the results of the retrieval process by ranking the cases differently than recommended by the TSM. Our case authoring approach compiles statistics on the applicability of the case and adjusts the ranking of similar cases accordingly.

This is reflected in the presentation of information in IDS seen in Figure 4. Figure 4 is a screen shot of the Fault Resolution window in IDS. The Fault Resolution window allows for the maintenance technician to query about more information about a problem on a particular aircraft. Previous to our case authoring approach, the Historical Cases area of the window was empty because no cases were retrieved from the case-base. After our case authoring approach was performed, the Historical Cases area was populated by retrieved cases that

were created from the technical documentation and authenticated with historical experience. This allows for the maintenance technician to make better-informed decisions based upon historical experiences outlined by the technical documentation. Ultimately, the maintenance technician makes the decision once all the information for the problem is evaluated.

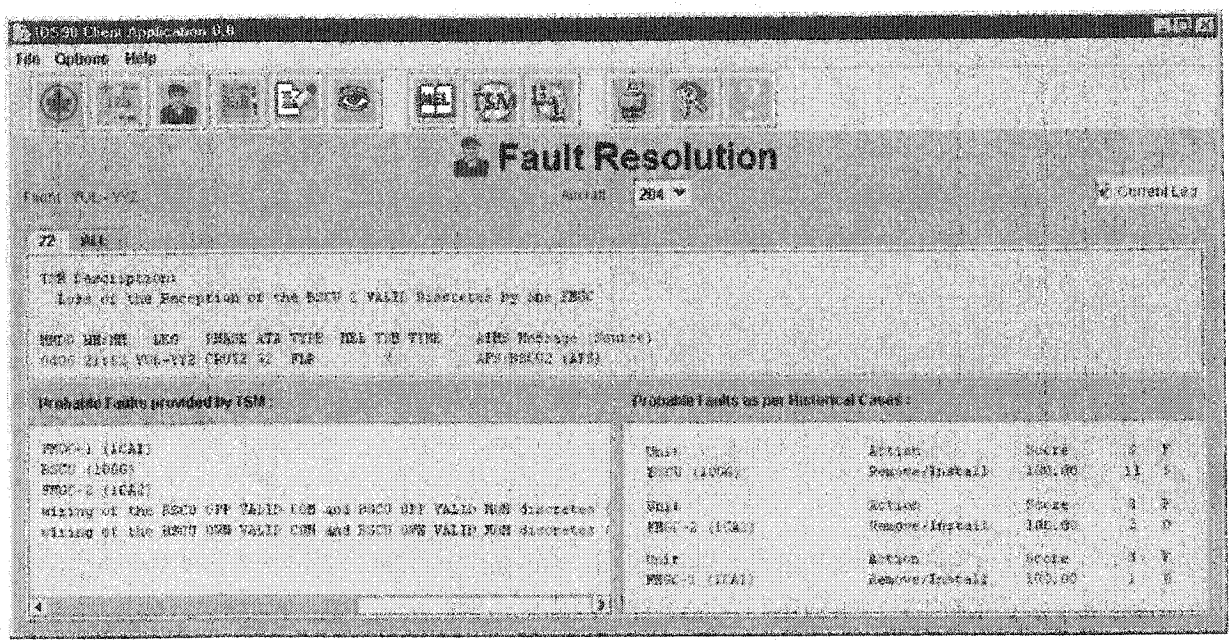

Figure 4: IDS Fault Resolution Screen with retrieved historical experiences

The coverage of the solutions with respect to the total number of problem instances is very low and is less than we anticipated. Despite the small coverage, the solutions give a better idea of how the manuals are used in everyday operation. This allows the maintenance organization to focus their attention on the problem instances where the solutions are unknown. In the results, problem symptoms were present and then disappeared without any solution from the TSM been recorded. This can be explained in different ways, but two possibilities have been identified. The first possibility is that the problem was resolved as a result of an additional test outlined in the TSM. An example of this is a case where a computer was reset, the symptoms were subsequently tested for persistence, and the repair action information recorded in the AMTAC report was "CHECKED OK". The second

possibility is that there may exist relationships between components and symptoms that have not been documented. For instance, a set of symptoms may disappear because an unrelated component has been replaced. Given the complexity of the equipment and the close relationships between components, this possibility is realistic. These unresolved problem instances, identified by the case authoring approach, could be used as training data for other knowledge discovery techniques to discover new undocumented relationships among different aircraft components. It is unclear and not the purpose of this thesis to determine what percentage of cases is related to the two different possibilities.

6 Discussion

The case authoring approach was successful in demonstrating that knowledge from technical documentation and information from historical experience are influential in the case authoring process. This approach to authoring cases requires improvements to performance according to the measurements outlined in the above chapter. The discussion of the implementation and results from the case authoring approach using technical documentation and historical experience is broken in two parts: case-base creation and case-base validation.

6.1 Case-Base Creation

The case-base creation stage had two different implementation approaches to Information Extraction and each of these implementations were evaluated with respect to precision, recall, and F-measure. The results showed that the performance was less than state of the art when compared to the measures set by [Cowie and Lehnert 1996] and [Appelt and Israel 1999]. In this section, the results from the two different approaches to Information Extraction and suggested areas of improvements are discussed.

6.1.1 Regular Expression Information Extraction Implementation

The first implementation of Information Extraction for case-base creation was regular expressions. This approach was simple and yielded a case-base that was used in the case-base validation stage. This approach quickly demonstrated that knowledge from the technical documentation was useful in the case authoring process. An initial case-base could be constructed and further enhanced with historical experiences. The regular expression implementation used only three regular expressions to extract a large majority of the cases from the text. From the previous chapter, this Information Extraction implementation had a high precision measure, but low recall measure. The first regular expression, '[replace] the *', yielded the most cases, but also was the reason for the misidentified cases. The other two regular expression were much too specialized and their results reflected this. The majority of cases that were not identified in the text were due to different maintenance actions and

linguistic patterns. Therefore, further investigation is needed to develop more regular expressions to improve recall.

Further investigation into complete coverage for 'replace' and other maintenance actions is needed. This incomplete coverage and confidence is the result of outlier situations within the fault isolation procedure text, the coverage of the IDS rule set with respect to the TSM, and understanding of more linguistic concepts in the text. These three issues need to be addressed in order for the coverage of the regular expressions to improve.

The first investigation could be the study of n-gram frequencies to establish frequently occurring usage patterns. Once frequently identified patterns are discovered, the process can focus on the development of new regular expressions from these frequencies. These new patterns are incorporated into the Information Extraction system and the system is re-evaluated.

The second investigation is in the coverage of the IDS rule set with respect to the fault isolation procedures in the TSM. When calculating the word frequency for 'replace' in the TSM, some references to 'replace' occurred in chapter and section pages that are not part of a fault isolation procedure and are not used in the troubleshooting of problems on the aircraft. The number of occurrences that happened in these situations amounted to 436 and these instances of 'replace' should not be considered in the evaluation of the coverage for the regular expressions. IDS rules are constructed based on the availability of automatically generated information from the aircraft. Some of the fault isolation procedures in the TSM are referenced by messages that are not transmitted from the aircraft and these fault isolation procedures are not represented within the IDS rule set. Therefore, not all the TSM fault isolation procedures have a corresponding IDS rule. Some fault isolation procedures are intentionally left out of the IDS rule set and these fault isolation procedures amount to 3303 occurrences of 'replace'. This makes the number of possible occurrences for 'replace' actually, 11,247 and the overall coverage of the regular expressions for the TSM fault isolation procedure text better than first expected. The recalculated amount of coverage achieved by the regular expression is 94.4% and the amount of 'replace' not covered by the

regular expressions actually 629. This investigation into the word frequency 'replace' proved important in the evaluation of the actual coverage of the regular expressions in the TSM. The second investigation resulted in a more accurate number of occurrences of 'replace', which could be identified by the regular expression Information Extraction system. The subsequent re-evaluation of the coverage with respect to the new number of 'replace' occurrences yielded a higher percentage, 94.4%. Even though this higher coverage was achieved, the other measurements of precision, recall, and F-measure remain unaffected because the fault isolation procedures used in the evaluation of the Information Extraction measures were already being processed.

The third investigation would be to accommodate more linguistic concepts. Two of the regular expressions handle anaphora resolution through encoding it inside the regular expression. The reference for the pronoun 'it' is the first noun phrase before the pronoun. These two regular expressions are considered specialized due to the small amount of occurrences that they relate to. Other anaphoric references are considered to be noun phrases such as 'check' and 'filter'. These noun phrases are generic references for a specific instance similar to the use of a pronoun. For instance in TSM fault isolation procedure 21-21-00-810-801, 'filter' is used to refer to the 'FILTER-RECIRCULATION (4012HM)'. Many different patterns of anaphora reference are written within the TSM fault isolation procedures and each of these patterns requires a regular expression to cover them. This is another reason for more regular expression development.

Another form of linguistic processing is order of words within the noun phrases. The central noun in the noun phrase is called the head noun [Jurafski and Martin 2000]. Adjectives are properties and qualities that describe a head noun. A majority of the aircraft components are described with the head noun first followed by adjectives [Jurafski and Martin 2000]. For instance, CMPT-TCAS refers to the TCAS computer. There are references to 'TCAS computer' where the base noun follows the adjectives. Each of these combinations must be considered in the development of regular expressions that cover the TSM fault isolation procedures.

The last linguistic concept referenced in further investigation is semantic references. Since 'replace' was used to develop the regular expressions, WordNet has four different usages for 'replace' and the TSM exhibits two of those [WordNet 2003]. The majority of occurrences of 'replace' have the verb usage 'to substitute' [WordNet 2003]. Other occurrences of 'replace' have another definition 'to reposition' [WordNet 2003]. The distinction between the two definitions is useful in case authoring because the first definition suggests a repair with another component while the second is simply reinstalling the same component. Another semantic concept is the use of multiple components being replaced in the same sentence. For instance, 'or' is a conjunction that joins two noun phrases of equal status [Jurafsky and Martin 2000]. An example of a sentence using 'or' to separate two different components is:

replace the SFCC-1 (21CV) or SFCC-2 (22CV).

The above sentence contains two different components that may be replaced. The current regular expressions cannot establish this as such and another regular expression(s) are required to cover this situation. These are only a few linguistic techniques that need to be considered in the development of regular expressions.

These extra regular expressions would be specialized for the problem domain and these regular expressions would not be robust to adapt to updates to the manuals or other problem domains. Therefore, if an effort into Information Extraction system were to continue, it should have better understanding of the text and the semantic concepts used.

6.1.2 Semantic Grammar Information Extraction Implementation

The second implementation of Information Extraction for case-base creation used a semantic grammar to identify case information within the TSM fault isolation procedure. Since the number of grammar rules incorporated in the semantic grammar Information Extraction was more than the number of regular expressions used by the regular expression Information Extraction system, the semantic grammar Information Extraction implementation resulted in a better precision, recall, and F-measure scores than the Information Extraction implementation with regular expressions. The semantic grammar Information Extraction

system required more time and resources to develop because the set of grammar rules was more comprehensive than the developed regular expressions. Some of the performance increase was attributed to the increase in the number of rules, but these rules are based on general grammar rules established by [Quirk et al. 1999]. These more general grammar rules allow for more maintenance actions to be covered due to the identification of linguistic constructs such as imperative verbs. These imperative verbs are interpreted as maintenance actions in the TSM and cases can be constructed with the identified information. The semantic grammar Information Extraction implementation took longer to construct, but the linguistic information represented inside the semantic grammar Information Extraction system makes it a more flexible and robust Information Extraction implementation.

This linguistic information encoded within the grammar rules also may lead to an Information Extraction system that could be more easily transferable to other problem domains. Currently, the case-base creation semantic grammar implementation of Information Extraction is specialized for the aerospace problem domain. Other problem domains could be implemented by replacing the lexicon with a lexicon developed from concepts from the new problem domain. Another modification to the semantic grammar rules would be to replace the aerospace specific maintenance concepts with the maintenance concepts for the new problem domain. Since the semantic grammar is based from more general grammar rules, some of the aerospace maintenance concepts may be applicable to other problem domains. For instance, focussing on if-then conditional statements may lead to general patterns that can be transferred between problem domains. The study of such linguistic constructs can be useful in making the Information Extraction system perform have better precision, recall, and F-measure scores.

The semantic grammar implementation processed the two samples of TSM fault isolation procedures and results will need to be obtained from the complete TSM. The development of a complete lexicon to cover the text inside the TSM was a difficult task. Three problems encountered is the spelling of similar words, word order in noun phrases representing TSM, and assignment of semantic meaning to each word. The first problem addresses the spelling

used within the TSM to represent different components. Sometimes shorthand representations are used in the text for components. Words like 'C/B', 'TEMP', and 'CMPTR' refer to 'circuit breaker', 'temperature', and 'computer'. These shorthand representations must be entered into the lexicon in order to be recognized by the semantic grammar. The second problem encountered is the word order in noun phrases representing TSM components. As mentioned previously in Section 6.1.1, the base noun location within the TSM component noun phrase varies from the start to the end. Any grammar must accommodate for this in order to parse the sentence. Finally, the third problem is recognition of noun phrases with similar meaning. The solution to this problem may assist in the resolution of the other two problems previously mentioned. A knowledge base containing correlations between short hand representations and their expanded versions would be useful in development of the lexicon identifying components. Similarly, that knowledge base may contain the different word order representations for the noun phrases that represent TSM components. This knowledge base would also be useful in constructing a comprehensive lexicon for the semantic grammar.

The development of the semantic grammar Information Extraction system resulted in an implementation that was more flexible and robust than the regular expression implementation. The investment of time and resources into the development of the grammar and lexicon not only resulted in increased Information Extraction performance, but increased the potential for reuse in other problem domains because the grammar is based on more general linguistic concepts established by [Quirk et al. 1990] and [Quirk et al. 1985]. Section 7.3 discusses the future work required to adapt the semantic grammar Information Extraction implementation to other problem domains. An Information Extraction system's ability to be easily adapted to other problem domains is important when considering a more general solution to the case-base creation stage. Therefore, the approach taken in the development of Information Extraction system implementing a semantic grammar was a step towards that generalized solution for the case-base creations stage.

6.1.3 Observable symptoms in the Problem Description

Observable symptoms in the problem description were identified as valuable information in the case authoring process in Section 4.3. Observable symptoms are problem symptoms that are not automatically generated by the aircraft and usually are observations by a human observer. The current approaches used did not construct cases with observable symptoms because the current IDS case structure does not accommodate this. The semantic grammar Information Extraction system was able to identify these observable conditions, but in post processing of the generated parse tree these symptoms were ignored. Both Information Extraction systems constructed cases that contained the same information, but would have been different if these symptoms were encoded in the case. One challenge encountered is the labelling of tests and results of tests. The tests and some results are capitalized with specific information while other results are adjectives hidden in the text. Examples of tests are: 'BITE test', 'Operational Test of the Main Fuel Pump System with Fuel Flow by Transfer', and 'Functional Test of the Center Tank Fuel Control'. Many of the tests in the TSM fault isolation procedure are found in the fault confirmation section. Some examples of results of tests are; 'TCAS (1SG)', 'NO DMU DATA', and 'clogged'. Many of the results of the tests are found in the finite clause in the conditional phrase. Regardless of the Information Extraction system used in the case-base creation stage, these identified observable problem symptoms must be incorporated into the case structure. The IDS case structure would have to be modified to accommodate this information in its retrieval process.

6.2 Case-Base Validation

6.2.1 Case-Base Problem Identification

The problem identification was straightforward because the automatically generated symptoms are easily identified in the IDS FEO database. Since an individual problem can occur over multiple flight legs and generate multiple FEOs, the task of grouping these multiple FEOs into individual problem instances is a challenging task. Time is one characteristic that can be used to group these FEOs into individual problem instances. In Section 4.2.2, a period of 24 hours is used to group FEOs with similar TSM related

diagnostic messages. The spurious problems that intermittently generate problem related diagnostic messages might stop producing these symptoms for a period of time before regenerating these messages again. This period of time is arbitrary and may differ between problems. A process might be established to more accurately determine the appropriate time period for each problem.

6.2.2 Case-Base Solution Identification

The case-base solution identification step was the most difficult process to automate due to the inconsistent format and availability of the information provided in the AMTAC reports. Since the process of identifying solutions within the AMTAC reports is a nontrivial task, any automated process must be robust and flexible to handle the varying formats of the information [Farley 1999]. The implementation of the case-base solution identification step must recall all the potential AMTAC reports related to the problem instance. This was done by first retrieving AMTAC reports related to the problem instance with respect to aircraft identity and time. Since the number of AMTAC reports was large for all the problem instances, a second process was used to further filter out irrelevant AMTAC messages using a BOW approach and stop list of words explained in Section 4.2.3.2. The intention of this approach was to recall all possible AMTAC messages related to the problem and let the user filter the unrelated ones. This approach resulted in a low precision and high recall measurements. The requirement that all related AMTAC messages to the problem instances are identified is reflected in these results.

The most accurate information in the AMTAC message was the parts-installed-and-removed field in the AMTAC message. Since case-base creation stage integrated the component information from the TSM and specific part information from the IPC, this allowed for the most reliable identification of components in the case-base solution identification process. The part installation and removal fields accounted for 65.15% of the relevant validated AMTAC messages and the remaining 34.85% were established through the identification of components in the AMTAC report text fields. Some challenges interpreting both types of

information fields were encountered in the case-base solution identification step and these challenges are discussed below.

The first identified challenge was identifying unrelated case solutions in the text fields in the AMTAC messages. Even though a stop list of words was established, some words that were important in solution identification were still frequent in the AMTAC messages. This resulted in a large amount of misidentified case solutions specific to one or more cases. An alternative approach could develop individual stop lists for the cases to filter out irrelevant AMTAC messages. These stop lists could be interactively constructed with the help of end user. Autonomous agents use reinforcement learning for optimally predicting outcomes that satisfy their goals [Mitchell 1997]. A trainer, who evaluates outcome from the agent's prediction, provides feedback to the outcome by either giving the agent a reward or penalty. The agent adjusts accordingly its prediction to maximize the rewards and minimize penalty feedback from the trainer. Robot control and board game participation are examples of problems that implement reinforcement learning, but this type of learning can also be applied to the problem of case-base solution identification. The case-base solution identification process can process a group of AMTAC messages predicting the relevancy of each. After the user validates this group of AMTAC messages, this feedback is used to adjust the individual case's stop list by adding or removing the words. After the stop lists have been re-evaluated, the remaining AMTAC messages are processed again to determine their relevance. This iterative approach could build stop lists that can improve the precision of identifying related AMTAC messages and hopefully recall will remain the same.

The second identified challenge was interpreting the text in the related AMTAC messages. As previously mentioned in the above reasons for performance of the Information Extraction systems, the word order for the components in the AMTAC messages varies. The case-base solution identification must adapt itself to identify components with head nouns preceded by adjectives describing it and vive versa. Since the grammar and spelling used by the repair staff is very specific to each person, the case-base solution identification process must adjust to these individualistic tendencies. One such tendency is the removal of vowels in the words.

Some examples of such are 'rplc' and 'xceiver', which are the word representation for 'replace' and 'transceiver'. Also, character substitution is a problem that is encountered when processing the AMTAC messages. One example is the 'FMGC-1' where some maintenance staff refers to it as 'FMGC #1'. Finally, sometimes the word order is changed in the AMTAC messages. Using the same example with the 'FMGC #1', it is sometimes expressed as '#1 FMGC'. These types of linguistic rules are not common in more formal languages like English and the case-base solution identification process must handle these rules in order to improve precision and recall. One possible solution to this could be to further develop the semantic grammar used in the case-base creation stage to include these AMTAC specific linguistic rules.

A majority of AMTAC messages identify a single action and component used in the repair. Some special instances multiple actions and components are identified inside the AMTAC report. Since identification of which particular action and component that actually fixed the problem is difficult to evaluate in this situation, all actions and components referenced in the AMTAC message are identified as solutions. This can lead to solutions that were not influential in the solution of the problem and it would be more accurate to eliminate those actions and components that were not influential. This problem cannot be addressed by case-base solution identification process and this problem can only be resolved through the domain expert intervention.

The third identified challenge was interpreting linkages between AMTAC reports. Some of the AMTAC reports refer to other AMTAC reports for solutions and problem descriptions. Below is an example of such an AMTAC message:

SEE L1120630 FOR FIX

This example demonstrates that this AMTAC message is related to another AMTAC message, 'L1120630', for the actual repair. This becomes a problem when both AMTAC messages reference the same repair action in both. Even though both AMTAC messages are

relevant, only one maintenance action should be used in the case-base validation stage. This is done to eliminate noise within the data so that only one instance of the repair is recorded.

The last discussed challenge was to differentiate between the different components that have the same part number and manufacturer information. The FMGC-1 and FMGC-2 have the same part information; it is difficult to distinguish between the two in AMTAC reports where their part information matches. Sometimes the text within the AMTAC message does not offer any information for differentiating between the two components. This was resolved by selecting both components in the repair even though this can be seen as an introduction of noise into the case-base.

In conclusion, this discussion outlines the improvements and challenges faced in using this case authoring approach. The improvements in the case-base creation stage focus on the improvements in recall with the inclusion of more linguistic processing of the technical documents. Case-base validation stage improvements focused on improvements that would improve precision in the retrieval of AMTAC messages. This was also achieved through better understanding of the text within the AMTAC messages. Finally, the case-base constructed from both technical documentation and historical experiences contains maintenance information, which can be reused and distributed knowledge within the maintenance organization.

7 Future Work

In this chapter, we outline future work identified for our case authoring approach. This future work includes optimization of the different processes in the two stages of case authoring and it has the potential for additional research. This future work focuses on the improvements to precision and recall measures in the case-base creation stage and similar improvements are identified in the retrieval of historical experiences in the case-base validation stage. Other future work is the integration of the knowledge encapsulated in the case-base with other knowledge acquisition approaches, such as data mining. The final area of future work discussed is the potential application into different problem domains such as the open-pit mining industry. This chapter is divided according to these three areas of future work: further development of the TSM case-base authoring system, integration with other knowledge acquisition systems, and other problem domain applications.

7.1 Further Development of the TSM Case-Base Authoring System

Further development of the TSM case-base authoring system focuses on improvements to the precision and recall measurements used to evaluate the case-base creation and case-base validation stages. Improvements in one or both of these measures will help achieve better accuracy and support in creating and authenticating cases. Other identified future work areas are in the maintenance of the case-base when the existing technical documentation is revised or new technical documents are introduced. The use of the case-base for other AI reasoning techniques is the last area of future development related to the TSM case-base authoring system.

7.1.1 Future Development of the Semantic Grammar

The semantic grammar Information Extraction implementations resulted in better Information Extraction performance measurements than the regular expression Information Extraction implementation. Even though full understanding of the text was not achieved in either approach, the semantic grammar Information Extraction approach incorporated a better understanding of the text, which helped construct more accurate cases. Also, since the

grammar rules used in the semantic grammar were based on more general rules from [Quirk et al. 1985] and [Quirk et al. 1990], the semantic grammar Information Extraction system is more easily modified to other problem domains. It is therefore, our recommendation that any further development done in the application of the case-base creation stage focus on improvements to the semantic grammar Information Extraction implementation. Three areas of future development are automated process of lexicon development, additional TSM semantic grammar rules, and integration with other knowledge bases.

The first identified future development of the semantic grammar is to automate the process of lexicon development. In Table 4, a majority of the sentences in the TSM were not parsed due to the lack of word representation in the lexicon. This lack of word representation could be addressed by manually entering in the missing words into the lexicon whenever the parser identifies a word not in the lexicon. Another approach to lexicon development would be to automate the process by making predictions on the missing words based on the surrounding context or development of predictive models based on marked up training data from the user. This was identified as the major effort needed to help improve the identification of information for creating cases in the case-base creation stage.

Once a lexicon is developed, the sentences not covered by the semantic grammar rules are discovered. These sentences are studied to determine additional semantic grammar rules for the parser. The construction of these grammar rules can be done manually by incorporating rules based more general grammar rules from resources such as [Quirk et al. 1990]. These two areas of future work would improve the recall of the semantic grammar Information Extraction system and an Information Extraction system that would construct a more comprehensive case-base.

The last identified work for future development in the semantic grammar Information Extraction system is to integrate it with other knowledge and information resources. Additional knowledge from other knowledge bases could be useful in improving Information Extraction precision and recall. WordNet is an online literary reference resource that is organized by lexical reference. WordNet contains usage information for words and

relationship information between words [WordNet, 2003]. Some categories that a word can be used as are nouns, verbs, adjectives, adverbs, and function words. Synonymy, antonymy, and meronymy are example of relationships between words that are expressed in WordNet. As mentioned previously, 'replace' has four different word usages where two of them are used in the TSM. WordNet could be used to identify these different word usages and disambiguate the current usage for the word. Also, other knowledge bases could be used for anaphora resolution to determine noun phrase characteristics like number agreement.

In conclusion, the development of the semantic grammar is not complete, but it is satisfactory for the proof of concept of our case authoring approach. Some of the areas of future work can lead to research in automatic lexicon and grammar development, which is useful in many NLP applications. The above future work must be completed in order to create a more robust, broad coverage parser to handle the text in the TSM and be acceptable for use in a real world application. One area of reuse for this semantic grammar could be in the case-base validation stage of the case authoring approach, which could be influential in improving the accuracy in the case-base validation stage.

7.1.2 Less Interaction by the Maintenance Technician

The second identified area of future development in the TSM case authoring approach is in the case-base validation stage. The results reflect the poor precision of the current implementation for the case-base solution identification and the number of AMTAC messages validated manually was large. Even though this is a difficult task to automate, better precision is needed to reduce the amount of effort required to manually validate these case-base solutions in the historical experiences. This reduction in the number of manually validated AMTAC messages is essential if this case authoring approach is to be accepted in a maintenance organization. Two possible suggestions could help resolve this lack of precision; the first is automated learning of keywords related to case solutions in the AMTAC messages. The second is to adapt the semantic grammar developed in the case-base creation stage of the case authoring approach.

The case-base solution identification process uses a stop list to filter out AMTAC messages that only contain these frequently occurring words. Currently, one stop word list is implemented for all the cases, but one possibility for better precision is to keep a stop word and keyword list for each case. The words in the stop list could be used to filter out AMTAC messages and the keyword list could be used to identify AMTAC messages. Since the words in each list reflect the individual characteristics of the case, each case should be represented by their own list of words. Further investigation is needed to determine the viability of such approach and the effect on the precision and recall of the case-base solution identification process.

Another possibility is to reuse the semantic grammar and lexicon developed from the case-base creation stage. Further development is needed to adapt the semantic grammar to handle the specialized grammar reflected in the AMTAC reports. Since it unrealistic to parse all the messages retrieved with respect to time and aircraft identity, it might better performance to integrate the previous approach to filtering out irrelevant AMTAC messages using stop word and keyword lists. This approach would be similar to the approach taken by the FASTUS system developed by [Hobbs et al. 1997]. FASTUS takes advantage of cascading different text filtering and understanding techniques in IE. Once the irrelevant AMTAC messages are filtered out, the semantic grammar can focus its processing on the relevant AMTAC messages with better text understanding techniques. This experimentation is left for future consideration to determine the effects of combining the stop word and keyword retrieval with the semantic grammar approach to NLP. Any future developments in this area will help reduce the effort in manually validating the solutions related to the cases in the case-base.

7.1.3 Automated maintenance of the case-base

In Chapter 2, the maintenance of the knowledge inside the knowledgebase is essential to its creditability. The knowledge base reflects the knowledge in the problem domain and real-world problem domains are very dynamic in nature. The knowledge base must continually update this information to represent the current knowledge in the problem domain.

Therefore, the acquisition and maintenance of this knowledge must be automated in order to reduce the manual effort in achieving this.

Currently, the case-base was automatically constructed from the technical documents and the case-base was constructed solely from the current versions of the technical documentations. Since our case-base creation stage is automated, the processing of technical document revisions is the same as the original documentation, but the integration of the discrepancies between the case-base and the new version of documentation need to be handled by some case-base maintenance strategy. Strategies for maintaining the case-base are required to accommodate the revision of the technical documentation and the introduction of new technical documents. Case-base maintenance strategies are identified as an area of future investigation. Case-base maintenance is a focus of many research activities in CBR [Yang et al. 2003, Smyth 1998, Portinale and Torasso 2000]. Some future experimentation is required to identify and implement the best strategy to maintain the case-base.

7.1.4 Integration with other Information Systems

In the case-base creation stage, the addition of part number and manufacturer information was influential in identifying relevant AMTAC messages in the case-base validation stage. Additional study into the problem domain is needed to identify other potential information sources that could be influential in the case authoring process. Some potential information systems are aircraft parametric data, part inventories, flight planning and scheduling, and weather information systems. Each of these and any other information systems used by the maintenance staff must be identified and evaluated for their contribution to the decision making process of the maintenance staff. These studies may lead into the identification and integration of information systems that make our case authoring approach more reliable and accurate.

7.1.5 Conversational Case-Base Reasoning

Conversational Case-Base Reasoning (*CCBR*) is a form of CBR where the end user interacts with the CBR system through a conversation [Aha et al. 2001]. The user can start the

interaction through a brief description of the problem where the additional questions and solutions are retrieved from the case-base. If the user selects a solution, the conversation is over between the end user and the case-base. The user can also select a question, where the answer to the question will give additional information to query the case-base. New questions and solutions may be retrieved from the case-base. This interactive conversation continues until a solution is reached.

In Section 6.1.3, the post processing of observable symptoms in the problem description was not implemented due to the inability of representing these symptoms in the case structure. If the IDS case structure was able to represent these observable symptoms, CCBR could be used as a way to elicit this information from the maintenance staff through interactive conversations with the case-base. The questions to be asked by the CCBR system are ones that are constructed from the observable symptoms outlined in the technical documentation and can be presented in a display similar to [Aha et al. 2001]. One possible ranking of the unanswered questions can reflect the order that the observable symptoms appear in the technical documentation. As the end user interacts with the CCBR system, they are refining their query to the case-base and increasing the accuracy of the retrieved possible solutions. This improvement in accuracy demonstrates the value of including the observable symptoms as part of the case's problem description. The answers to the questions could serve as a record of observable symptoms that can be recorded in the specific problem instance. Until there is an automated process of recording these observable symptoms, interaction between the CCBR system and the maintenance staff is the only viable solution and more work is required to implement such a CCBR system.

7.2 Integration into Other Knowledge Acquisition Systems

7.2.1 Automated Case Creation System

The second approach to case authoring implemented in IDS is the *Automated Case Creation System* (*ACCS*) [Yang et al. 2003]. This case authoring approach is more traditional in starting the process of creating cases from historical experiences. The ACCS automates the process of case authoring by correlating FEOs and AMTAC messages by searching for

automated diagnostic messages contained in both the FEO and AMTAC message. Once this correlation is established, a NLP parser is used to identify the component in the AMTAC message. The ACCS stores the information gained from the FEO and AMTAC message and this newly constructed case is added to the case-base as a new or updated case. The ACCS does not capture some cases where no automatic diagnostic messages was found in the AMTAC message. From case-base validation stage, 22% of the authenticated identified cases had the automatically generated diagnostic message mentioned in the AMTAC message. These other 78% of the AMTAC messages identified by our case authoring approach would have been disregarded because no diagnostic message references were present.

These two case authoring approaches could be integrated to develop a more comprehensive case-base. This could be accomplished by using each approach in succession. The initial case-base could be created using the TSM case-base authoring approach. This would identify problems and problem instances with respect to the technical documentation and historical experiences. The historical experiences used in the TSM case authoring approach would be filtered out of the historical experiences to be processed by the ACCS. The ACCS could process the remaining historical experiences capturing problems and solutions not represented by the technical documentation. This makes the ACCS more efficient by allowing it to focus its processing on the unknown solutions to problems. The integration of these two case authoring approaches and evaluation of the accuracy of the case-base is left for future work.

7.2.2 Mining for Additional Problem Descriptions

Data mining is the process of discovering patterns in data using automated or semi-automated approaches [Witten and Frank 2002]. The amount of data is usually very large quantity and stored in databases. The patterns identified by data mining techniques are useful to the problem domain and results in some advantage [Witten and Frank 2002]. The advantages can be the ability to predict future outcomes accurately through better

understanding of the data. Data mining could be useful in determining additional symptoms in the problem description for the cases extracted from the technical documentation.

After the case-base is authenticated with historical experience, individual cases could be identified for further investigation. This investigation is the discovery of new symptoms that were previously overlooked due to the volume of the data. Two information resources that could be used in this discovery process are the automated messages generated by the aircraft and the parametric data recorded by the aircraft. The first information resource, the automatically generated messages, was previously described in Section 2.3. The cases were constructed by identifying these diagnostic messages in the IDS rule set. The FEO history data could contain additional diagnostic messages that potentially could be identified as additional problem symptoms for the case problem description. These messages could be established through processing the data and discovering patterns of diagnostic messages that are related to time proximity of the problem symptoms. The IDS case definition allows for time-related diagnostic messages to be added in the case's time-proximity field. Time-related diagnostic messages could be discovered in the FEO history using the case information and data mining techniques. These identified time-proximity diagnostic messages could be added to the case as additional problem description systems and used in the case-base retrieval process.

The other mentioned additional problem description symptoms could be the parametric data that is recorded by the aircraft. An aircraft can generate up to 11 types of reports containing parametric data information [Létourneau et al. 1997]. The cases contain problem instances that have recorded date and times and outcomes, which can be used as training data for data mining approaches. These data mining techniques could be useful in identifying specific parameters and values that could be beneficial in describing the problem symptoms for a particular case. Before the case-base could benefit from this parametric information, the IDS case structure would have to be modified to include such information. Once this is achieved, this parametric information can then be recorded in the case as another problem description symptom.

Another benefit that could be obtained through the use of the case-base is the discovery of new solutions to problems. These new solutions could be identified through the analysis of the AMTAC reports associated with the case's problem instances. Text mining is similar to data mining, except that text mining is looking for patterns in text [Witten and Frank 2002]. Text mining has been used to identify keyword phrases in text and for Information Extraction tasks. In the Section 5.3.2, a majority of the problem instances identified were resolved without the application of any retrieved cases. Therefore, there must have been some repair actions taken that solved the problem. These problem instances could be used to identify potential solution patterns in the AMTAC messages, but the amount of text recorded about these problem instances is very large. Since this makes the manual processing of this text infeasible, text mining approaches could be useful in the identification of new case solutions based on the patterns found within in the text. Text mining could be used to identify verb and noun phrases that may be interpreted as maintenance actions and components, which can be transformed into a new case solution. These newly identified solutions could be used to create new cases, which are retained in the updated case-base.

This process of identifying additional problem description symptoms and new problem solutions uses the case-base as training data for the data mining and text mining. The case-base allows the data and text mining techniques to identify problems with recorded outcomes and focus their efforts on these problems. This ability to identify the training data does not directly involve the maintenance staff, but uses the historical incidents recorded in the case-base to be used as training data. The accumulation of training instances in the case-base happens as a side effect of maintaining and repairing aircraft and is transparent to the maintenance staff.

7.2.3 Case-Base as training data for Data Mining

Another identified area of future research is the use of a case-base to assist in the labelling of training set used for building predictive models in data mining applications. Data mining applications already use background knowledge from technical documentation in data pre-processing and data analysis to identify class definitions, relevant parameters, and out-of-

range parameters [Létourneau et al. 1997]. Since the validated case-base has a combined knowledge based on technical documentation and historical experiences, it could be used as a source for labelled training examples, which could be used in the machine learning process of data mining approaches.

The successful application of machine learning is dependent on the type of training experiences available to it [Mitchell 1997]. [Létourneau et al. 1997] identify labelling of the training data as an obstacle that successful data mining applications must overcome. [Létourneau et al. 1997] propose automated or semi-automated approaches to labelling training data as a solution. One overlooked resource is the case-base created from the technical documentation and validated with historical experience. Similar to the future work described in previous section, the case-base can be searched for specific components and the case information can be used to construct a set of training experiences. The most important information in constructing these training experiences is the individual problem instances and results of applying the case to each problem instance. These individual instances contain information regarding temporal information when the problem started and ended. Also, the results could be evaluated and transformed into the classification representation used by the machine learning algorithm. Future research is required to determine the exact effects of using the case-base as a source of training data in data mining applications.

7.3 Other Problem Domain Implementations

The last area of future development is the application of our case authoring approach to other maintenance problem domains. Many maintenance organizations face similar and different obstacles when performing their maintenance activities. The equipment that they are required to maintain and repair is becoming increasing complex like aircraft with the increasing amount of computers and sensors monitoring the equipment. One particular industry that showed similarities to the aerospace maintenance industry and good potential for future application is the open-pit mining industry. The open-pit mining uses equipment that is becoming more sophisticated difficult to repair and maintain efficiently and effectively.

An open-pit mining company uses a variety of different equipment to perform its everyday operation and this equipment is distributed in different locations around the mine. Regular maintenance and unscheduled breakdowns are part of the everyday activities performed by the maintenance organization. Since mining equipment is purchased from different manufacturers, the collection of equipment is very heterogeneous. Therefore, domain experts specialize in maintenance and repair of specific pieces of equipment. Since many similarities can be established between the aerospace and open-pit mining problem domains, our case authoring approach could acquire and distribute this type of specialized knowledge on repair and maintenance of this equipment.

The first step in the application of our case authoring approach would be implement the applications necessary for the case-base creation stage. A common resource for background knowledge is the technical documentation associated with each of the equipment. The type of information found in the technical documentation is maintenance procedures, troubleshooting procedures, parts lists, assembly diagram, and wiring diagrams. The format and structure of the technical documentation between different equipment manufactures is not consistent and any Information Extraction implementation for the case-base creation stage must be flexible and robust to handle this varying structure and format. The semantic grammar Information Extraction implementation could be adapted by removing the semantic concepts related to aircraft specific maintenance and replaced with mining equipment specific semantic concepts. Any relationships between the technical documentation for a single piece of equipment must be identified as outlined in Section 3.1.1. The resulting Information Extraction system could be used to construct a case-base from the technical documents and be authenticated in the following case-base validation stage.

The implementation of the case-base validation stage would be the next required step in using this case authoring approach in the open-pit mining maintenance domain. Much of the equipment in the mine is equipped with sensors that collect diagnostic and parametric information. This diagnostic and parametric information is stored in the *Vehicle Information Management System* (*VIMS*) database. The maintenance personnel use the VIMS database to

identify problems and create *Planned Job Packages* (*PJP*s) for the problem's repair. A PJP is a work order that identifies the labour, parts, and procedures needed to resolve the problem. The problem instance identification could be done using the diagnostic information stored in the VIMS database. The VIMS database identifies a PJP related to the problem incident and the solutions can be identified from the cases retrieved for the problem instance. Since the case-base solution identification uses a BOW approach, no specialized adaptation is required and only the stop list of frequently occurring words would need to be established. The VIMS database contains the time frame for the problem and the PJP contains the information when the repair was completed. A similar strategy could be reused to determine the effectiveness of the repair using the information from the VIMS database and PJP. The case-base could then be updated using this information.

Open-pit mining organizations have all the information resources needed to implement our case authoring approach, but they do not take advantage of these information resources to their fullest potential. Any implementation in a different problem domain is a very difficult task to undertake. This effort could be reduced by automation or reuse from previous implementations. It would be of interest to establish the amount of modification needed to the semantic grammar in order for the semantic grammar Information Extraction system to be useful in the case-base creation stage for open-pit mining equipment. Our case authoring approach could take advantage of this diagnostic data being collected and use it to acquire knowledge about repairs done within the open-pit mining organization. Once this knowledge is represented in the case-base, maintenance staff could access this case-base and use it to make more efficient and accurate repair decisions.

8 Conclusion

The case authoring process does not start with individual historical experiences, but with the background knowledge encapsulated in available documentation. Traditionally, CBR applications use the CBR cycle to store new cases from individual experiences as they occur. The adaptation of similar experiences retrieved from the case-base is used to develop new solutions that are later retained inside the case-base for future reference. The case-base evolves over time as more experiences are captured to reflect the knowledge for the problem domain. The goal of CBR is to use the past to predict the future by retrieving and adapting stored experiences in the case-base. A challenge for CBR implementation is the case authoring process for retaining new experiences in the case-base. Case authoring has been identified as a CBR knowledge acquisition bottleneck and must be overcome in order for CBR to be successfully applied in the real world. Case authoring has had little or no research on automating and semi-automating the process [Aha 1998].

The extraction of knowledge from text documents for case authoring is a relatively new idea. [Brüninghaus and Ashley 2001] were the first authors to discuss it, but they do not address the automation of the process of the creating and validating the cases from textual documentation (ie. manuals). They implemented NLP techniques in an Information Extraction approach to illicit knowledge from litigation documents. Another case authoring approach, ACCS, uses NLP to identify diagnostic messages and solutions in maintenance reports, but cannot identify any cases that do not mention diagnostic messages in the maintenance report [Yang et al. 2003]. Since both these case authoring approaches use the individual experience as the starting point, they cannot speculate beyond the historical experiences except when a new problem presents itself. Background knowledge from technical documentation could help solve this issue and build a case-base that is comprehensive and robust to adapt to the dynamic environments of maintenance organizations.

Our approach to case authoring is fundamentally different than other traditional approaches in that the domain specific background knowledge from technical documentation is used to construct the case-base before individual experiences are considered. This approach is more comprehensive because both background knowledge from the problem domain and historical experiences are considered in the construction of the case-base. This is accomplished in a two-stage process: case-base creation and case-base validation stages. The case-base creation stage constructs an initial case-base that represents the domain knowledge encoded in the text. This knowledge within the case-base is authenticated with historical experiences in the case-base validation stage. These two available resources, technical documentation and historical experiences, are essential in the identification and interpretation knowledge used in the maintenance organization.

The first stage involves the creation of the case-base from the available documentation using Information Extraction techniques. The first implemented Information Extraction system for the case-base creation stage used regular expressions and rule sets to construct cases from the text. This approach to Information Extraction yielded a large case-base, which was authenticated in the case-base validation phase. The results from using the regular expression Information Extraction system were lower scores for recall and F-measure than state of the art system developed for the MUC [Cowie and Lehnert 1996; Appelt and Israel 1999]. Further development of regular expressions would be needed to improve the recall and F-measure scores, but the maintainability and robustness of the Information Extraction system would be reduced. Therefore, an approach to Information Extraction with better text understanding was implemented using a semantic grammar.

Ease of maintenance and more flexibility are some reasons for the development of the second Information Extraction system for the case-base creation stage. The second Information Extraction implementation used a semantic grammar to identify and construct cases within the text and rules. The result of switching to an Information Extraction implementation with NLP techniques was an improvement in recall and F-measure. These improvements were 27% better for recall and 21% better for F-measure. This was still lower

than the state of the art measures outlined in [Cowie and Lehnert 1996] and [Appelt and Israel 1999], but with better coverage by the semantic grammar would achieve better results. This Information Extraction system would be more robust and maintainable than the regular expression implementation because the semantic grammar was based on grammar rules from [Quirk et al. 1990] and [Quirk et al.1985]. These more general grammar rules allow for identification of the aerospace specific concepts so they could be easily replaced with some other semantic concepts related to another problem domain. Some identified areas for future development is a more automated approach to semantic grammar and lexicon development for the Information Extraction application. Once the case-base creation stage is completed, the cases in the case-base must be validated using historical experience.

The second stage in our case authoring approach is the case-base validation stage. This stage further enhances the case-base by populating the cases with historical experience and this results in a better understanding of the different repairs that were performed in the past. Problem instances are identified using the symptoms for case retrieval. Repair actions are retrieved using the time and equipment identification from the problem instance information. These repair actions are further filtered down using the case solution information to determine the relevance of the repair action with respect to the case solution. A BOW approach was implemented in the case solution identification step to filter out irrelevant repair actions. The domain expert would provide quality assurance by making a binary decision whether the repair action is relevant to a specific case. The resulting implementation for case solution identification had high recall, but was not very precise. This resulted in more interaction with the domain expert than was acceptable. When only the relevant repair actions remained, the repair action is evaluated to determine its effectiveness. The problem instance, repair action, and evaluation of effectiveness are used to update the affected case in the case-base. Future research was identified into the improvement of the precision of the case solution identification system.

Both technical documentation and historical experience are influential in the case retrieval process. Technical documentation outlined the problem's symptom and solution information,

but no information could be ascertained with respect to the organization's maintenance and repair history. The historical experience can provide two different pieces of information essential for the case-base. The first information is an evaluation to determine the amount of technical documentation that is actually used in maintaining and repairing equipment. This information could be used to reorganize the case-base so that frequently retrieved cases are retrieved quickly while cases with no historical experience are stored for contingency. The second information resulting from historical experience is the evaluation of which cases retrieved are frequently used to solve the problem. This information determines the effectiveness of previous case applications and cases with more successful applications result in a ranking better than other retrieved cases. Even though the ultimate decision is left to the maintenance personnel, the case-base contains the combined knowledge acquired from both the technical documentation and historical experiences.

Our case authoring approach resulted in a case-base that integrates knowledge from documents and how they have been applied within the organization. This knowledge identified which solutions worked well on certain problems and identified problem instances that had solutions that were not covered by the documentation. Due to the complexity of the repaired equipment, it is not unreasonable that the technical documentation does not cover all the problems that can occur and the maintenance staff use new solutions to remedy the problem. These new solutions are difficult to identify when all the repair histories contain both known and unknown solutions. Therefore, new solutions are easier to identify when the already documented knowledge is filtered. Other automated case authoring techniques like ACCS could focus their efforts on these unknown individual experiences to construct cases that are not represented in the case-base constructed from technical documents. Our case authoring approach is influential in the knowledge acquisition and knowledge maintenance in aerospace maintenance CBR applications.

The future work that is required can be divided into three areas: application improvements for better recall, precision, and F-measure, the integration with other knowledge acquisition approaches, and other problem domain implementations. One of the challenges encountered

is increasing recall without penalizing precision in the case-base creation stage and the improvement to precision is needed in the case-base validation stage. Due to the effect of over-fitting, the development of a perfect system with respect to recall and precision is difficult to achieve [Mitchell 1997]. A reasonable balance between the two measures is needed in order for both stages to be effective in a real world environment.

Another identified area of future research is the effect of our case authoring approach on other knowledge acquisition techniques. The case-base developed from the technical documentation and historical experience can be used to identify new problem symptoms and solutions that could be used to create new cases or update existing ones. Also, the resulting case-base could be used to develop training data for KDD approaches for learning new patterns in the data. These two areas of research reuse the case-base created with our case authoring approach as information to discover new unknown knowledge about the maintenance organization and the equipment it maintains.

The last identified area of future work is the application of our case authoring approach to other problem domains. Many similarities exist between different maintenance organizations and these similarities make reuse of approaches and techniques possible. Some identified similarities are existence of technical documentation that contains background knowledge about the equipment, this equipment generates diagnostic information to identify problems, and maintenance actions are recorded by the maintenance organization. This information is common to most large maintenance organizations, but this information is not used to its fullest potential. This information could be used to provide better insight and interpretation of the maintenance done by the organization by integrating the different types of information with each other. This increase in knowledge achieved a better understanding of the maintenance organization and the equipment it monitors. We have demonstrated one possible approach using CBR and case authoring to gain a better understanding of an aerospace maintenance organization by correlating the different sources of information with each other. Our case authoring approach created a portable knowledge artefact that could be easily distributed across the maintenance organization and accessible to all the maintenance

staff. Since this case-base easily distributed, our case authoring approach promotes knowledge sharing within the maintenance organization.

The knowledge acquisition bottleneck is an obstacle that any application of AI must overcome. Not only the acquisition of knowledge, but also the maintenance of knowledge within the AI application must be automated in order for the system to keep its credibility. Our case authoring approach uses automated techniques to extract cases from the available technical documentation and a semi-automated approach to validate those cases with historical experience. As revisions or new technical documentation becomes available, they could be processed in a similar way using our two-stage case authoring approach to update the knowledge within the case-base. This case-base would then contain the most up-to-date knowledge from both the technical documentation and historical experiences, thus satisfying both constraints for the acquisition and maintenance of knowledge within a knowledge management system.

The knowledge of converting raw materials into something more valuable is called intellectual capital [Stewart 2001]. The raw materials could be traditional raw materials, but in most knowledge intensive environments, the raw material is information. Even though knowledge is important and useful in any organization, it is difficult to measure this new type of asset called intellectual capital. One potential way of measuring intellectual capital is through knowledge acquisition and sharing in AI applications. Our case authoring approach builds a case-base that could be seen as part of an organization's intellectual capital. Because this case-base provides a better understanding of the maintenance done in the organization, this collection of intellectual capital is vital to the operation and competitiveness of the organization.

The increase in the collection and distribution of knowledge within the organization makes it more competitive in the global community [Stewart 2001]. Since the traditional physical competitive advantages are slowly eroding, intangible advantages like knowledge are becoming more important in the organization's ability to compete. The difference in successful companies is that they are able to lever their knowledge to produce their products

more efficiently and effectively. As profit margins shrink in a competitive global marketplace, organizations must adapt their operations to become more efficient [Stewart 2001]. This efficiency can be achieved through knowledge management tools that provide the knowledge worker with easier to access and distribution of knowledge within the organization. In a knowledge-based economy, knowledge and creation of knowledge is considered as important as the physical assets in the organization [Drucker 2001].

References

Aamodt, A. and Plaza, E., (1994). Case-based reasoning; Foundational issues, methodological variations, and system approaches *AI Communications*, Vol.7, No.1, March 1994, pp. 39-59.

Agrawal, R., Imielinski, T., and Swami, A. (1993) Mining Associations between Sets of Items in Massive Database. In *Proceedings of the ACM-SIGMOD 1993 Int'l Conference on Management of Data*, Washington D.C., May 1993

Aha, D. (1998). The Omnipresence of Case-Based Reasoning in Science and Application. *Knowledge-Based Systems*, 11(5-6), 261-273.

Aha, D., Breslow, L., and Muñoz-Avila, H. (2001). Conversational case-based reasoning. *Applied Intelligence*, 14, 9-32.

Allen, J. (1995). *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, Inc.

Appelt, D. and Israel, D. (1999). Introduction to Information Extraction Technology. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence(IJCAI '99)*. Stockholm, Sweden, July 31 – August 6, 1999.

Bartsch-Spörl, B., Lenz, M., and Hübner, A. (1999). Case-Based Reasoning – Survey and Future Directions. In *Proceedings of the Fifth Biannual German Conference on Knowledge-Based Systems (XPS)*, Würtzburg, Germany, March 3-5, 1999.

Bellazzi, R., Montani, S., Portinale, L., and Riva, A. (1999). Integrating Case-Based and Rule-Based Decision Making in Diabetic Patient Management. In *Proceedings of the 3rd International Conference on Case-Based Reasoning (ICCBR-99)*, Seeon Monastery, Germany, July 27-30, 1999.

Boicu, M., Tecuci, G., Stanescu, B., Marcu, D., and Cascaval, C. (2001) Automatic Knowledge Acquisition from Subject Matter Experts. In *Proceedings of the 13th IEEE*

International Conference on Tools with Artificial Intelligence (ICTAI'01), Dallas, Texas, November 7-9 2001.

Brown, S. and Lewis, L. (1991). A Case-Based Reasoning Solution to the Problem of Redundant Resolutions of Nonconformances in Large-Scale Manufacturing. In *Proceedings of the Third Innovative Applications in Artificial Intelligence (IAAI '91)*, Anaheim, USA, July 14-19, 1991.

Brüninghaus, S. and Ashley, K. (2001). The Role of Information Extraction for Textual CBR. In *Proceedings of 4th International Conference on Case-Based Reasoning (ICCBR-01)*. Vancouver, Canada, July 30 – August 2 2001.

Careless, J. (2003). SAP: Ending the IT Madness? *Aviation Maintenance, Aviation Today*, Issue February 2003.

Cowie, J. and Lehnert, W. (1996). Information Extraction. *Communications of the ACM*, Vol. 39, No. 1, January 1996. New York, ACM Press.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. (1998). Learning to Extract Symbolic Knowledge from the World Wide Web. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, Madison, WI, July 26 – 30, 1998.

Drucker, P. (2001). The Essential Drucker: *In One Volume the Best of Sixty Years of Peter Drucker's Essential Writings on Management*. Harper Collins, New York.

Farley, B. (1999) From free-text repair action messages to automated case generation. In *Proceedings of the AAAI 1999 Spring Symposium on AI in Equipment Maintenance Server & Support*. Stanford University, Stanford, California, U.S.A. March 22-24, 1999.

Fischer, D. (1997). A Theory Of Presentation and Its Implications For The Design Of Online Technical Documentation. Ph.D. Thesis. VIDe (Visual and Information Design) Centre, Coventry University, Coverntry, UK.

Freidman, T. (2000). *The Lexus And The Olive Tree: Understanding Globalization.* Anchor Books, New York.

Freitag, D. (1998). Information Extraction from HTML: Application of a General Machine Learning Approach. In the Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98), Madison, WI, July 26 – 30, 1998.

Grishman, R. (1997). Information Extraction: Techniques and Challenges. *Challenges Information Extraction (International Summer School SCIE-97)*, ed. Maria Teresa Pazienza, Springer-Verlag, 1997.

Grishman, R. and Sundheim, B. (1996). Message Understanding Conference – 6: A brief History. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, June 1996.

Heider, R. (1995), Troubleshooting CFM-56-3 Engines for the Boeing 737 - Using CBR and Data Mining. In *Proceedings of the Third European Workshop, EWCBR-96*, Advances in Case-Based Reasoning, Lausanne, Switzerland, November 14-16, 1996, pp512-518.

Heider, R., Auriol, E., Tartarin, E., and Manago, M. (1997). Improving the Quality of Case Bases for Building Better Decision Support Systems. In *Proceedings of the 5th German Workshop on Case-Based Reasoning (GWCBR'97)*, Bad Honnef, March 4-5 1997, p.85

Hobbs, J., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., and Tyson, M., (1997). FASTUS: Extracing Information from Natural Language Texts. *Finite State Devices for Natural Language Processing*, eds. Roche and Schabes, MIT Press, Cambridge MA, 1997.

Jurafsky, D. and Martin, J. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall.

Lenz, M., Bartsch-Spörl, B., Burkhard, H., and Wess, S. (1998). *Case-Based Reasoning Technology: From Foundations to Applications*. Berlin, Springer.

Létourneau, S., Famili, F., and Matwin, S. (1997). Discovering Useful Knowledge from Aircraft Operation/Maintenance Data. In *Proceedings of the workshop on Machine Learning Applications in the Real World, 14th International Conference on Machine Learning*, Nashville, TN, July 8-12, 1997.

Magaldi, M. (1994). CBR for Troubleshooting Aircraft on the Flight Line. In *Proceedings of the IEE Coloquium on Case-Based Reasoning: Prospects for Applications*, Digest No. 1994/057, 6/1-6/9.

Makhoul, J., Kubala, F., Schwartz, R., and Weischedel, R. (1999). Performance Measures for Information Extraction. In *Proceedings of the DARPA Broadcast News Workshop*, Herndon, USA, February 28 – March 3, 1999.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill Companies, Inc. New York.

Portinale, L. and Torasso, P. (2000). Automated Case Base Management in Multi-modal Reasoning System, In *Proceedings of Advances in Case-Based Reasoning: 5th European Workshop (EWCBR 2000)*, Trento, Italy, September 6-9 2000.

Quirk, R., Greenbaum, S., Leech, G., and Svartvik, J. (1985). *A Comprehensive Grammar of the English Language*. Longman.

Quirk, R. and Greenbaum, S. (1990). *A Student's Grammar of the English Language*. Longman.

Reisbeck, C. and Schank, R. (1989). *Inside Case-Based Reasoning*, Erlbaum, Northvale, NJ, USA.

Rubin, S., Smith, M., and Trajkovic, L. (1999). Randomizing The Knowledge Acquisition Bottleneck. In *Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, Tokyo, Japan, October 12-15, 1999.

SAP (2003). http://www.sap.com

Smyth, B. (1998). Case-Based Maintenance, In *Proceedings of the 11th International Conference on Industry and Engineering Applications of AI and Expert Systems*, Castellon, Spain, June 1-4, 1998.

Stewart, T. (2001). *The Wealth of Knowledge: Intellectual Capital and the Twenty-first Century Organization*. Doubleday, New York.

Transport Canada (2002). Canadian Aviation Regulations. http://www.tc.gc.ca/aviation/regserv/carac/cars/html_e/doc/nav-20.htm

Watson I. (1997). *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann Publishers, Inc., San Francisco.

Watson, I. (2001). "Knowledge Management and Case-Based Reasoning: a Perfect Match?", In *Proceedings of the 14th International Florida Artificial Intelligence Research Society Conference (FLAIRS-2000)*, May 2001, Key West FL.

Witten, I. and Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, New York.

WordNet (2003). http://www.cogsci.princeton.edu/~wn/

W3C (2003). World Wide Web Consortium. http://www.w3c.org/

Wylie, R., Orchard, R., Halasz, M., and Dubé, F. (1997). IDS: Improving Aircraft Fleet Maintenance. In *Proceedings of the 14th National Conference on Artificial Intelligence and Innovative Applications of Artificial Intelligence (IAAI-97)*, Providence, Rhode Island.: pp. 1078-1085. July 27-31, 1997.

Yang, C., Orchard, R., Farley, B., and Zaluski, M. (2003). Automated Case Base Creation and Management. To Appear in the *16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 2003)*, Loughborough, UK, June 23-26, 2003.

Appendix A Table of Abbreviations

Abbreviation	Definition
ACCS	Automated Case Creation System
AI	Artificial Intelligence
AMM	Aircraft Maintenance Manual
AMTAC	Aircraft Maintenance Tracking And Control
ASM	Aircraft Service Manual
ATA	Air Transport Association of America
BOW	Bag Of Words
BSCU	Braking and Steering Control Unit
CBR	Case-Base Reasoning
CCBR	Conversational Case-Base Reasoning
DTD	Document Type Definition
FASTUS	find this abbreviation
FEO	Fault Event Object
FIN	Functional Item Number
FMGC	Flight Management and Guidance Computer (check)
HTML	HyperText Markup Language
IDS	Integrated Diagnostic System
IE	Information Extraction
IPC	Illustrated Parts Catalogue
IR	Information Retrieval
KDD	Knowledge Discovery in Databases
LHS	Left Hand Side
MEL	Minimum Equipment List
MOP	Memory Organization Packets
MUC	Message Understanding Conference
NLP	Natural Language Processing
PJP	Planned Job Packages

Abbreviation	Definition
RHS	Right Hand Side
SGML	Standardized General Markup Language
SMILE	SMart Index LEarner
SRM	Service Repair Manual
SRO	Snag Rectification Object
SROV	Snag Rectification Object Validation
TCAS	find this abbreviation
TSM	Trouble Shooting Manual
VIMS	Vehicle Information Management System
XML	eXtensible Markup Language

Appendix B Case Creation Results

Case-Base Creation Results:

TSM Fault Isolation Procedure	Number of Cases in the Key	IE Regular Expression Cases	IE Semantic Grammar Cases
21-61-00-810-803	13	7	12
21-61-00-810-804	13	7	12
22-83-00-810-817	2	1	1
22-83-00-810-818	2	1	1
22-83-00-810-863	13	5	5
28-21-00-810-801	25	2	7
28-21-00-810-807	11	5	5
28-21-00-810-813	28	3	9
28-21-00-810-814	27	3	9
28-21-00-810-817	7	3	3
28-21-00-810-818	7	3	3
28-21-00-810-819	7	4	1
28-21-00-810-823	26	7	2
31-32-00-810-822	3	2	2
34-43-00-810-801	9	5	2
Total	193	58	74
Total Incorrect		2	0

Collection of Cases #1:

Case ID	TSM Message Symptoms	Action	Component
1.1.1	FLR: NO DMU DATA [313634]	Remove/Install	CFDIU (1TW)
1.1.2	FLR: NO DMU DATA [313634]	Remove/Install	DMU (1TV)
1.2.1	FLR: AFS FMGC1 [228334]	Remove/Install	FAC-1 (1CC1)
1.3.1	FLR: AFS FMGC2 [228334]	Remove/Install	FAC-1 (1CC1)
1.4.1	FLR: AFS 28V PWR 11XU1 [242255]	Remove/Install	FMGC-1 (1CA1)
1.4.2	FLR: AFS 28V PWR 11XU1 [242255]	Remove/Install	FMGC-2 (1CA2)
1.4.3	FLR: AFS 28V PWR 11XU1 [242255]	Remove/Install	BTC-1 (11XU1)
1.4.4	FLR: AFS 28V PWR 11XU1 [242255]	Remove/Install	BTC-2 (11XU2)
1.4.5	FLR: AFS 28V PWR 11XU1 [242255]	Remove/Install	CNTOR-DC NORM BUS 2 SWITCHING (1PC2)
1.5.1	FLR: TCAS (1SG) [344334] WRN: NAV TCAS FAULT [34]	Remove/Install	CMPTR-TCAS (1SG)
1.5.2	FLR: TCAS (1SG) [344334] WRN: NAV TCAS FAULT [34]	Remove/Install	C/B-NAV/TCAS (4SG)
1.5.3	FLR: TCAS (1SG) [344334] WRN: NAV TCAS FAULT [34]	Remove/Install	DATA LOADER RECEPTACLE
1.5.4	FLR: TCAS (1SG) [344334] WRN: NAV TCAS FAULT [34]	Remove/Install	TCAS computer
1.5.5	FLR: TCAS (1SG) [344334] WRN: NAV TCAS FAULT [34]	Remove/Install	C/B-NAV/TCAS

Collection of Cases #2:

Case ID	TSM Message Symptoms	Action	Component
2.1.1	WRN: AIR PACK1 FAULT [21]	Remove/Install	VALVE-FLOW CTL (11HB)
2.1.2	WRN: AIR PACK1 FAULT [21]	Remove/Install	SENSOR-COMPRESSOR PNEUMATIC OVERHEAT (10HM9)
2.1.3	WRN: AIR PACK1 FAULT [21]	Remove/Install	SENSOR-COMPRESSOR TEMPERATURE (12HH)
2.1.4	WRN: AIR PACK1 FAULT [21]	Remove/Install	SENSOR-COMPRESSOR OVERHEAT (15HH)
2.1.5	WRN: AIR PACK1 FAULT [21]	Remove/Install	CONTROLLER-PACK 1 TEMP (7HH)
2.1.6	WRN: AIR PACK1 FAULT [21]	Remove/Install	PUSHBUTTON SWITCH-AIR COND/PACK 1 (7HB)
2.1.7	WRN: AIR PACK1 FAULT [21]	Remove/Install	BOARD-ANN LT TEST & INTFC (8LP)
2.2.1	WRN: AIR PACK2 FAULT [21]	Remove/Install	VALVE-FLOW CTL (8HB)
2.2.2	WRN: AIR PACK2 FAULT [21]	Remove/Install	SENSOR-COMPRESSOR PNEUMATIC OVERHEAT (11HM9)
2.2.3	WRN: AIR PACK2 FAULT [21]	Remove/Install	SENSOR-COMPRESSOR TEMPERATURE (32HH)
2.2.4	WRN: AIR PACK2 FAULT [21]	Remove/Install	SENSOR-COMPRESSOR OVERHEAT (35HH)
2.2.5	WRN: AIR PACK2 FAULT [21]	Remove/Install	CONTROLLER-PACK 2 TEMP (27HH)
2.2.6	WRN: AIR PACK2 FAULT [21]	Remove/Install	PUSHBUTTON SWITCH-AIR COND/PACK 2 (6HB)
2.2.7	WRN: AIR PACK2 FAULT [21]	Remove/Install	BOARD-ANN LT TEST & INTFC (20LP)

Collection of Cases #2 (cont'd):

Case ID	TSM Message Symptoms	Action	Component
2.3.1	WRN: FUEL AUTO FEED FAULT [28]	Remove/Install	SFCC-1 (21CV)
2.3.2	WRN: FUEL AUTO FEED FAULT [28]	Remove/Install	original SFCC-1
2.3.3	WRN: FUEL AUTO FEED FAULT [28]	Remove/Install	SFCC-2 (22CV)
2.3.4	WRN: FUEL AUTO FEED FAULT [28]	Remove/Install	diodes.
2.3.5	WRN: FUEL AUTO FEED FAULT [28]	Remove/Install	SFCC-1 (21CV) or SFCC-2 (22CV)
2.3.6	WRN: FUEL AUTO FEED FAULT [28]	Remove/Install	FLSCU-1 (7QJ)
2.3.7	WRN: FUEL AUTO FEED FAULT [28]	Remove/Install	FLSCU-2 (9QJ)
2.3.8	WRN: FUEL AUTO FEED FAULT [28]	Remove/Install	related SEQUENCE VALVE
2.3.9	WRN: FUEL AUTO FEED FAULT [28]	Remove/Install	PUMP-FUEL NO 1 CTR TK (37QA) (PUMP-FUEL NO 2 CTR TK (38QA))
2.4.1	WRN: FUEL CTR TK PUMP1 LO PR [28]	Remove/Install	unserviceable supply cable and connectors
2.4.2	WRN: FUEL CTR TK PUMP1 LO PR [28]	Remove/Install	PUMP-FUEL NO 1 CTR TK (37QA)
2.4.3	WRN: FUEL CTR TK PUMP1 LO PR [28]	Remove/Install	PUMP-FUEL NO 1 CTR TK (37QA)
2.4.4	WRN: FUEL CTR TK PUMP1 LO PR [28]	Remove/Install	components that are unserviceable

Collection of Cases #2 (cont'd):

Case ID	TSM Message Symptoms	Action	Component
2.4.5	WRN: FUEL CTR TK PUMP1 LO PR [28]	Remove/Install	CNTOR-CTR TK PUMP 1 SPLY (35QA) 107VU
2.4.6	WRN: FUEL CTR TK PUMP1 LO PR [28]	Remove/Install	RELAY-CTR TK AUTO CONTROL PUMP 1 (41QA) 92VU
2.4.7	WRN: FUEL CTR TK PUMP1 LO PR [28]	Remove/Install	P/BSW-FUEL/CTR TK/PUMP 1 (33QA) 40VU
2.5.1	WRN: FUEL CTR TK PUMP2 LO PR [28]	Remove/Install	unserviceable supply cable and connectors
2.5.2	WRN: FUEL CTR TK PUMP2 LO PR [28]	Remove/Install	PUMP-FUEL NO 2 CTR TK (38QA)
2.5.3	WRN: FUEL CTR TK PUMP2 LO PR [28]	Remove/Install	PUMP-FUEL NO 2 CTR TK (38QA)
2.5.4	WRN: FUEL CTR TK PUMP2 LO PR [28]	Remove/Install	components that are unserviceable
2.5.5	WRN: FUEL CTR TK PUMP2 LO PR [28]	Remove/Install	CNTOR-CTR TK PUMP 2 SPLY (36QA) 107VU
2.5.6	WRN: FUEL CTR TK PUMP2 LO PR [28]	Remove/Install	RELAY-CTR TK AUTO CONTROL PUMP 2 (42QA) 92VU
2.5.7	WRN: FUEL CTR TK PUMP2 LO PR [28]	Remove/Install	P/BSW-FUEL/CTR TK/PUMP 2 (34QA) 40VU

Appendix C Case-Base Validation Results

Case-Base Validation Collection #1:

TSM Message Symptoms	Problem Instances	FEOs	AMTAC reports	Automatic Identified Related Cases	Manually Validated Related Cases
FLR: NO DMU DATA [313634]	5861	13045	27443	396	14
FLR: AFS FMGC1 [228334]	3682	4851	12195	9	2
FLR: AFS FMGC2 [228334]	3470	5197	12299	14	1
FLR: AFS 28V PWR 11XU1 [242255]	4166	4632	11378	1323	19
FLR: TCAS (1SG) [344334] WRN: NAV TCAS FAULT [34]	4147	4630	11255	571	32
Total	21326	32355	74570	2313	68

Case-Base Validation Collection #2:

TSM Message Symptoms	Problem Instances	FEOs	AMTAC reports	Automatic Identified Related Cases	Manually Validated Related Cases
WRN: AIR PACK1 FAULT [21]	1928	2641	7333	1380	68
WRN: AIR PACK2 FAULT [21]	1824	2515	6965	1629	37
WRN: FUEL AUTO FEED FAULT [28]	1176	1546	4289	4423	49
WRN: FUEL CTR TK PUMP1 LO PR [28]	2823	4196	11292	11726	27
WRN: FUEL CTR TK PUMP2 LO PR [28]	2214	2534	7282	8403	38
Total	9965	13432	37161	27561	173

Appendix D Case-Base Update Results

Case-Base Update Results Collection #1:

Case ID	Successful Application	Unsuccessful Application
1.1.1	0	1
1.1.2	11	2
1.2.1	1	1
1.3.1	1	0
1.4.1	8	1
1.4.2	4	2
1.4.3	3	0
1.4.4	1	0
1.4.5	0	0
1.5.1	16	0
1.5.2	0	0
1.5.3	0	0
1.5.4	16	0
1.5.5	0	0
Total	61	7

Case-Base Update Results Collection #2:

Case ID	Successful Application	Unsuccessful Application
2.1.1	11	0
2.1.2	1	0
2.1.3	7	0
2.1.4	7	1
2.1.5	7	2
2.1.6	0	0
2.1.7	0	1
2.2.1	21	1
2.2.2	0	1
2.2.3	10	0
2.2.4	6	0
2.2.5	8	2
2.2.6	0	0
2.2.7	0	0
2.3.1	0	1
2.3.2	0	0
2.3.3	2	0
2.3.4	0	0
2.3.5	2	1
2.3.6	8	0
2.3.7	7	0
2.3.8	0	0
2.3.9	5	1
2.4.1	0	0
2.4.2	10	0
2.4.3	10	0
2.4.4	0	0

Case-Base Update Results Collection #2 (cont'd):

Case ID	Successful Application	Unsuccessful Application
2.4.5	2	0
2.4.6	7	1
2.4.7	8	0
2.5.1	0	0
2.5.2	6	0
2.5.3	6	0
2.5.4	0	0
2.5.5	1	0
2.5.6	2	0
2.5.7	7	0
Total	161	12

Statistics for Validated Cases:

Collection	Validated AMTAC Messages	AMTAC Messages with Part Information	AMTAC Messages without Part Information	AMTAC Messages with Diagnostic Message Reference	AMTAC Messages without Diagnostic Message Reference
1	68	53 (77.94%)	15 (22.06%)	25 (36.76%)	43 (63.24%)
2	173	104 (60.12%)	69 (39.88%)	28 (16.18%)	145 (83.82%)
Total	241	157 (65.15%)	84 (34.85%)	53(29.01%)	188(78.01%)