Information Retrieval/Case-Based Reasoning for Critical Incident and Accident Data

Peter McElroy

Class CS4H

Session 1999/2000

Department of Computing Science
University of Glasgow
17 Lilybank Gardens
Glasgow
G12 8RZ

Contents

- Abstract

When a person witnesses an accident or an incident which places the lives and safety of other people at risk they are encouraged to fill out a report. NASA, the UK Civil Aviation Authority and the Health and Safety Executive maintain huge databases of such incident reports.

Over time, these organisations have accumulated hundreds of thousands of records and data which will, however, only be of use if people can spot similarities amongst many comparable incident reports. This can be accomplished using conventional database queries providing both the data and the query are posed in the correct manner.

Incident reports however, are too diverse, affecting real people using complex equipment and therefore it is difficult to devise appropriate schema. This project will, therefore, focus on the application of case based reasoning or more general information retrieval techniques to help identify common features in a collection of incident reports.

Chapter 1

Introduction

1.1 Introduction

Many large organisations such as NASA (http://www-afo.arc.nasa.gov/ASRS/ASRS.html) and the UK Civil Aviation Authority (see http://www.chirp.dircon.co.uk) maintain large databases containing accident and incident reports. The numbers of these reports has grown rapidly over the years due to the large number of situations which put the safety of people in jeopardy. This makes it hard to analyse and notice similarities in the reports, with a view to a better understanding of how accidents can occur. This report argues therefore, that we need the help of tools such as conventional database queries to simplify the process. It can be extremely difficult to use such tools appropriately however, due to the complexity and diversity of most factors which contribute to such an incident. Therefore we must look to other alternatives such as case based reasoning for the analysis of incident and accident reports, to see if they improve upon existing methods.

1.2 Information Retrieval V Case Based Reasoning

To structure the information that is recorded about an incident, conventional databases use strictly defined data-models. Therefore as incident data changes, due to the identification of new causes, it becomes harder to change the data-model. Alternatives include using free-text retrievals as modelled by internet search engines. Unfortunately, this approach relies on incidents being stored as free text and is therefore subject to the limitations of precision and recall (see section 3.1.4 Structured Query Language for definitions). Case Based Reasoning on the other hand loosens the restrictions on the data enforced by current database techniques making it easier to adapt to ever changing scenarios. Although precision and recall still affect retrieval, it is too a much lesser degree. Indicating that perhaps case based reasoning is the way forward for incident analysis.

1.3 The Aviation Safety Reporting System

> "The Aviation Safety Reporting System (ASRS) was established in 1975 under a Memorandum of Agreement between the Federal Aviation Administration (FAA) and the National Aeronautics and Space Administration (NASA). FAA provides most of the program funding; NASA administers the program and sets its policies in consultation with the FAA and the aviation community. NASA has chosen to operate the program through a contractor selected via competitive bidding. The current contractor is Battelle Memorial Institute."(http://www-afo.arc.nasa.gov/ASRS/Overview.html)

The ASRS collects voluntarily submitted aviation safety incident reports which it analyses and replies to in an attempt to reduce the probability of aviation accidents. The reports submitted are held in the strictest confidence with regards to those involved in the incident. With more than 300,000 reports collated (at the last count) the ASRS assures us that a reporters identity has never been breached. This seems the most important fact that ensures people report aviation incidents. The ASRS use the data to:

> "Single out discrepancies in the National Aviation System (NAS) in an effort to have these deficiencies remedied by the authorities responsible."
>
> "Support the NAS in terms of planning and the construction of policy."
>
> "Strengthen the foundations of aviation human factors and safety research." (http://www-afo.arc.nasa.gov/ASRS/Overview.html)

The ASRS incident reports are ideal to use as data for the construction of different case libraries and databases. The information contained in each of these reports is as follows:

- Accession Number - a unique, sequential number assigned to each report.
- Date of Occurrence - the date of the occurrence/situation in the form of a year and a month; e.g., 9304 represents April 1993.
- Reported by - role of the person who reported the occurrence/situation. Codes used are: FLC - flight crew; PLT - pilot; CRM - crew member; CTLR - Air Traffic Controller; PAX - passenger; OBS - observer; AFC (or AIR) - Air Force; NVY - Navy; UNK - unknown.
- Persons Functions - description of a person's function at the time of the occurrence.
- Flight Conditions - the weather environment at the time of the occurrence or situation in terms of the conventional definition for flight conditions. Codes used are: VMC - visual meteorological conditions; IMC - instrument meteorological conditions; MXD - mixed flight conditions (both VMC and IMC); MVI - marginal VFR; SVF - special VFR.

- Reference Facility ID (or LOC ID) - the standard three-letter (or letter-number combination) location identifier associated with an airport or navigational facility as referenced in the FAA Order 7350.5Z series entitled "Location Identifiers."

- Facility Identifier - the standard three-letter (or letter-number combination) location identifier associated with an ATC facility as referenced in the FAA Order 7350.5Z series entitled "Location Identifiers."

- Aircraft Type - the aircraft type involved in the incident differentiated by arbitrary gross takeoff weight ranges (military aircraft type are differentiated by function).

- Anomaly (Descriptions, Detector, Resolution, Consequences) - short summary of a standard chain of sub-events within a reported incident.

- Situation Report Subjects - description(s) of a static hazard which creates a safety problem.

1.3.1 Report Outline

Chapter 1 supplied a brief introduction into the motivation behind this study, in addition to an initial comparison of incident analysis techniques. Also provided was an introduction to the Aviation Safety Reporting System (ASRS) which supplies all of the incident and accident reports used in this study.

In chapter 2 the use of Case Based Reasoning as an analysis tool will be discussed. To achieve this the U.S. Navy Conversational Decision Aids Environment (NaCoDAE) will be used. Developed at the Navy Centre for Applied Research in Artificial Intelligence by Leonard A. Breslow and David W. Aha, NaCoDAE is a software prototype being developed under the Practical Advance in Case-Based Reasoning project. In addition the Eindhoven Classification Model of System Failure (Van der Schaaf, 1992) which can be used as a checklist to assist the identification of causal factors, will be used to improve the incident analysis process.

In addition the classification of incidents under this model and the validation of the technique used will be covered. This chapter will also discuss the creation of case libraries for use with NaCoDAE, some of which are based on the Eindhoven Classification Model, and their validation in an effort to investigate the feasibility of Case Based Reasoning in accident and incident analysis.

Chapter 3 involves the use of conventional Information Retrieval tools to analyse accident and incident data. This chapter will concentrate on the development of accident and incident databases for the INQUERY probabilistic Information Retrieval (IR) system developed at the University of Massachusetts, Amherst. Further the testing of these databases will be discussed and consideration given to creating an inquery client-server connection for access over the World Wide Web. This chapter also includes precision and recall examples and user experiments using the databases produced. The investigation of IR techniques will provide a counterpoint to the use of case based reasoning, supplying us with the means of evaluating the Case Based Reasoning technique.

Following this chapter 4 provides the conclusions reached.

Chapter 2

Case Based Reasoning

2.1 Case Based Reasoning - an Introduction

The method of Case Based Reasoning originated in the United States (see Agnar Aamodt (1995) *"Case-Based Reasoning"*) and has become popular in recent years. The central issue of such an approach to problem solving is that the reasoner makes judgements based on its previous experiences, unlike other AI approaches which rely upon their knowledge of the problem domain. Essentially the reasoner must be able to remember old situations and adapt their solutions, so that it can compare any new problems to previous cases and determine which ones have the most similarities. Each new situation is then indexed if appropriate and filed away for future use, thus increasing the reasoner's knowledge base. This method is similar to the way humans frequently solve problems (see Kolodner (1993) for results and Gentner (1988) for the roots in child development). The key to the method is remembering i.e. getting the cases into memory and retrieving them when they are needed, this is called the indexing problem. This problem raises two main issues which establish the performance of the reasoner:

- how well the reasoner can foresee the advantages of each case that it stores in memory, and
- how well a reasoner can characterise the new problem and then infer which old cases should be consulted.

In addition the reasoner's usefulness is affected by the amount of experience it has had. It is clear that if there are very few previous cases to choose from then most new problems will be unrecognisable. Therefore, if adaption heuristics cannot be applied the reasoner will fail. If the reasoner can learn from this failure and figure out how the situation can be avoided in future it can insert this situation into the case library so that it can avoid such situations. Or, if the reasoner has the ability to discover what would fix the problem and learn from its mistakes it could become a powerful tool indeed.

2.2 NaCoDAE (the Navy Conversational Decision Aids Environment)

NaCoDAE is based on Conversational Case-Based Reasoning (CCBR) technology. Users interact with the system by providing a problem description which is matched to cases in the case library. A partial description will suffice. The user then can incrementally build on the description and answer related questions which enhances the problem description. With each incremental change the ranked list of best case matches is improved along with their percentage mark according to the users manipulation. This process can continue until the user is happy with a potential match to which the solution can be displayed. It is clear with such a system that to achieve good performance some expertise is needed when designing the case library so as to display the right questions at the right time. To these ends a component of NaCoDAE called CLIRE (Case Library Revisor) follows design guidelines to automatically revise case libraries.

This Conversational CBR technology was pioneered by Inference Corporation, and incorporated in their highly successful line of CBR products (i.e. CBR Express). The decision to use NaCoDAE was based on the question of avaliability and cost. A working copy and permission to use the software was provided free of charge. By using the software it was hoped that a reasonable idea of the suitability of Case Based technology as a tool for incident and accident analysis could be established.

2.3 Encoding a Case Library

Case library files are the texts used by the NaCoDaE tool to define the cases, actions and questions that are browsed by the users. To define a case library the three types of objects which must be specified are cases, questions and actions. The following describes the fields of each object:

To define a case -

- BEGIN CASE case-name (defines a new case where case-name is a one word unique identifier for this case instance)
- TITLE "A succinct title enclosed in quotes"
- DESCRIPTION "A description enclosed in quotes, which is used to match the cases against natural language queries input by the user"
- QUESTIONS
- question-name1 : "question-answer1"
- ...(possibly one or more questions, each on a new line.)
  (where question-name1 is the name of the associated question, and question-answer1 is the appropriate answer to this question leading to this case.)

- ACTIONS action-name1
- ...(possibly one or more actions, each on a new line.)
  (where action-name1 is the name of the associated action.)

To define a question -

- BEGIN QUESTION question-name (which is a unique identifier for this question)
- TITLE "A succinct title, enclosed in quotes"

- TEXT "Descriptive text, enclosed in quotes, which can specify a procedure for answering the question or clarify the question."
- TYPE : NUMBER — YESORNO — LIST (which depicts the type of question this is)

  (Of the three types YESORNO and NUMBER are self explanatory. To specify a list the following should be added:)

- LIST
- "One possible answer, in quotes"
- "A second possible answer, in quotes"
- " ...(possibly additional answers, each on a new line.)

To define an action -

- BEGIN ACTION action-name (single word uniquely identifying the action.)
- TITLE "A succinct title enclosed in quotes, on how to execute the action"
- TEXT "Instructions, enclosed in quotes, on how to execute the action"

2.4 Overview of the Case Libraries

Two different structures of case libraries were created. The first (see fig 2.1: where root cause refers to the basic causes of the incident) organises incident data into the cases themselves and then based on a set of questions you can refine the types of incidents you are viewing. Also, by looking at the actions associated with each case, you can view which anomalies the incident has been associated with.

Example Case, Question and Action entries in the libraries follow:

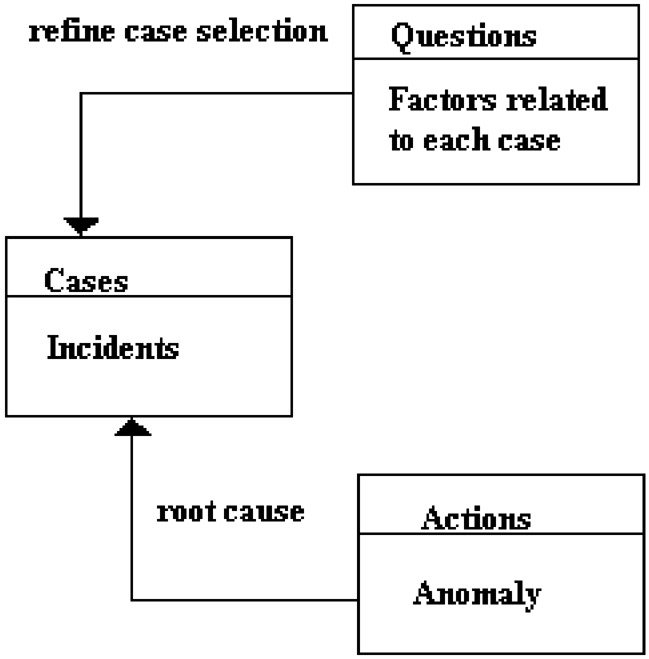

Figure 2.1: Case Library Structure

2.4.1 Example Case

BEGIN CASE 414863

TITLE "AN ATX C207 CLAIMS THAT THE LCL CTLR ON 2 SEPARATE OCCASIONS CLRED A DEPARTING ACFT FOR TKOF OPPOSITE DIRECTION WHEN HE WAS ON BASE LEG."
DESCRIPTION "SEP/XX/98 AT APPROX XA30 HRS. ATX X, A C207, HAD BEEN CLRED TO LAND RWY 33 AT MERRILL FIELD AT ANC. JUST PRIOR TO TURNING ON A CLOSE IN BASE TO FINAL, TWR CLRS ATX Y, A HELI, TO DEPART RWY 15. BOTH ACFT HAD TO MANEUVER TO AVOID A COLLISION. ALSO FORCED ATX X TO BE PLACED IN DANGER OF HELI'S ROTOR WASH. AT APPROX XB35 HRS, ATX Z, A C207, WAS CLRED TO LAND RWY 33 MERRILL FIELD. WHILE ATX Z WAS ON CLOSE IN L BASE, ATX X WAS CLRED TO DEPART RWY 15. BOTH ACFT HAD TO MANEUVER TO AVOID A COLLISION."
QUESTIONS
question1 : "Flight Crew - FLC"

question8 : “unknown”

question3 : “Local controller”

question2a : “PLT Pilot in a single-person crew”

question2b : “PLT Pilot in a single-person crew”

question2c : “PLT Pilot in a single-person crew”

question9b : “unknown”

question9a : “small aircraft (less than 5,000 lbs)”

question9c : “small aircraft (less than 5,000lbs)”

question10a : “Tower”

question10b : “Airport”

question11a : “Cockpit/Flight Crew”

question13 : “None”

question12 : “Flight crew avoidance / evasive action”

ACTIONS action9

action16

action20

TOOLBOOK

END CASE

In the above example ‘414863’ is the unique identifier associated with this case. The incident report synopsis has been used as the title of the case explaining the circumstances. Following this is the list of associated questions (each with the relevent answer enclosed in quotes) and also the actions for this case (where actions represent anomalies identified for the incident). All the data was taken from the ASRS reports and was not changed in anyway. Moreover no new analysis was performed upon it.

2.4.2 Example Question

BEGIN QUESTION question9a

TITLE “What was the type of the 1st aircraft involved in the occurrence?”

ANSWERS

TYPE : LIST

LIST :

“small aircraft (less than 5,000 lbs)”

“small transport (5001 - 14,500 lbs)”

“light transport (14,501 - 30,000 lbs)”

“medium transport (30,001 - 60,000 lbs)”

“medium large transport (60,001 - 150,000 lbs)”

“large transport (150,001 - 300,000 lbs)”

“large transport (over 300,000 lbs)”

“wide-body (over 300,000 lbs)”

“ultralight (including hang gliders)”

“sailplane/glider”

“special purpose”

“fighter”

“bomber”

“military transport”

“military trainer”

“none”

“unknown”

WEIGHT

END QUESTION

In the question above ‘question9a’ represents the unique identifier for this question. The title is the question being asked and following from this is the list of possible answers for the particular question.

The types of questions asked were chosen from the ASRS incident report structure and because all of the fields in the ASRS reports could not have yes and no answers, the list type questions where used when creating the library. In some of the cases there was more than one answer to a particular question (for example what was the type of aircraft in the accident? When there was more than one aircraft involved). For these cases additional questions where added along the lines of what was the 1st type of aircraft involved in the incident? and what was the 2nd type of aircraft involved in the incident? This is an inelegant solution for the application programmer but it is the only solution working within the restrictions of NaCoDAE's case library format. A list of questions used in both the rotary vehicle and air traffic control case libraries follows:

Who was the incident reported by?
What was the 1st flight crews function at the time of the occurrence?
What was the 2nd flight crews function at the time of the occurrence?
What was the 3rd flight crews function at the time of the occurrence?
What was the 4th flight crews function at the time of the occurrence?
What was the function of the TWR at the time of the occurrence?
What was the function of the TRACON at the time of the occurrence?
What was the function of the ARTCC at the time of the occurrence?
What was the function of the MIL at the time of the occurrence?
What was the function of MISC people at the time of the occurrence?
What was the function of the 2nd MISC at the time of the occurrence?
What was the flight conditions at the time of the occurrence?
What was the type of the 1st aircraft involved in the occurrence?
What was the type of the 2nd aircraft involved in the occurrence?
What was the type of the 3rd aircraft involved in the occurrence?
What was the type of the 4th aircraft involved in the occurrence?
What was the type of the 1st facility involved in the occurrence?
What was the type of the 2nd facility involved in the occurrence?
What was the type of the 3rd facility involved in the occurrence?
What was the 1st thing to detect the anomaly?
What was the 2nd thing to detect the anomaly?

What was the 3rd thing to detect the anomaly?

Was the anomaly resolved?

What was the anomaly consequences?

2.4.3 Example Action

BEGIN ACTION action8

TITLE "Conflict/NMAC (near midair collision) "

TEXT "A conflict is defined as the existence of a perceived separation anomaly such that the pilot(s) of one or both aircraft take evasive action; or are advised by ATC to take evasive action; or experience doubt about assurance of continuing separation from the viewpoint of one or more of the pilots or controllers involved. A near midair collision is when the flight crew reports, either directly or as quoted by the controller, that the reported miss distance is less than 500 feet."

TOOLBOOK

END ACTION

Although the above format of an action is self explanatory in an abstract sense an "action" no longer describes an action but an anomaly.

2.5 The Eindhoven Classification Model of System Failure

In addition to the previously created libraries it was decided that there may be some benefit to be gained from pre-processing the data in some way. Clearly, if the data was classified to fit some understandable structure it may be easier for the human mind to analyse. Rasmussen (1976) provided a basic model of human error based on three levels of behaviour. In terms of human error the model represents Skill Based, Rule Based and Knowledge Based behaviour. A description of each follows:

- Skill Based refers to automatic reactions demanding little or no attention.
- Rule Based concerns attentive reactions when one knows how to handle in a certain, well-known, situation.
- Knowledge Based refers to creative, analytical reactions when confronted with new, unknown problems without off-the-shelf solutions.

Human error however cannot be removed totally from the organisational and technical factors which impact upon an incident. The Eindhoven Classification Model (Van der Schaaf, 1992) (fig 2.2) integrates all of these factors and is widely used as a checklist to analyse incident data. Thus the model is perfect for our purposes. The classification of root causes under the model are as follows:

TE - A technical factor of engineering indicating a wrong design.
TC - A technical factor of construction indicating that the design
was not followed properly during the construction phase.
TM - A technical factor indicating unexplainable material defects.
OP - (Operating Procedures) refers to the inadequate quality of procedures.
OM - (Management Priorities) refers to any de facto pressure by top - or middle management to deviate from the formal organisational priorities.
HK1 - The correct status and dynamics of the system to be controlled must be known to the operator.
HK2 - The goal, or priorities of goals, must be known and understood by the operator.
HR1 - The operator in question must be qualified to do the job.
HR2 - If applicable the operator must obtain a temporary permit for activities where extra risk is involved.
HR3 - The preparation of the job itself starts by informing other operators, if necessary, of the work to be done.
HR4 - When arriving at the job location the local system status should be checked to comply with the expected conditions.
HR5 - The job itself should be planned correctly, i.e. the appropriate methods should be chosen and carried out in order.

HR6 - The prescribed equipment and information for a proper job performance should be present and used.

HS1 & HS2 - The execution of the required actions themselves implies successful correct movements; both controlled, i.e. intended, detailed, movements (e.g. to manipulate tools and request information), and maintaining the correct body position in order to make the controlled movements possible.

X - Unclassifiable behaviour

2.6 Overview of Eindhoven Case Libraries

This type of library (shown in fig 2.3) attempts to encompass the Eindhoven Classification model of system failure described previous. In this instance the cases are classified using a more analytical method in terms of factors of the model (e.g. TE, TC ...) and by answering the questions you can restrict the selection of factors. The actions associated with each of the factors represent incident data already classified in the library (the classification will be discussed in the following section). Thus each incident is classified and grouped together in an abstract model as opposed to being loosely coupled to other incidents on the basis of the same number of aircraft being involved in the incident ect..., as was previous. Also if there is a preferred rectifying action associated with the factor included in the classification matrix (matrix of causes and prefered actions) then this is also included as an action. Thus offering a prefered solution to the anomaly if one is avaliable. Two separate incident databases have been classified in the model one being Air-Traffic Control Incidents and Helicopter (Rotary vehicle) Incidents. These data sets were chosen due to the high level of detail and expertise contained in the reports.

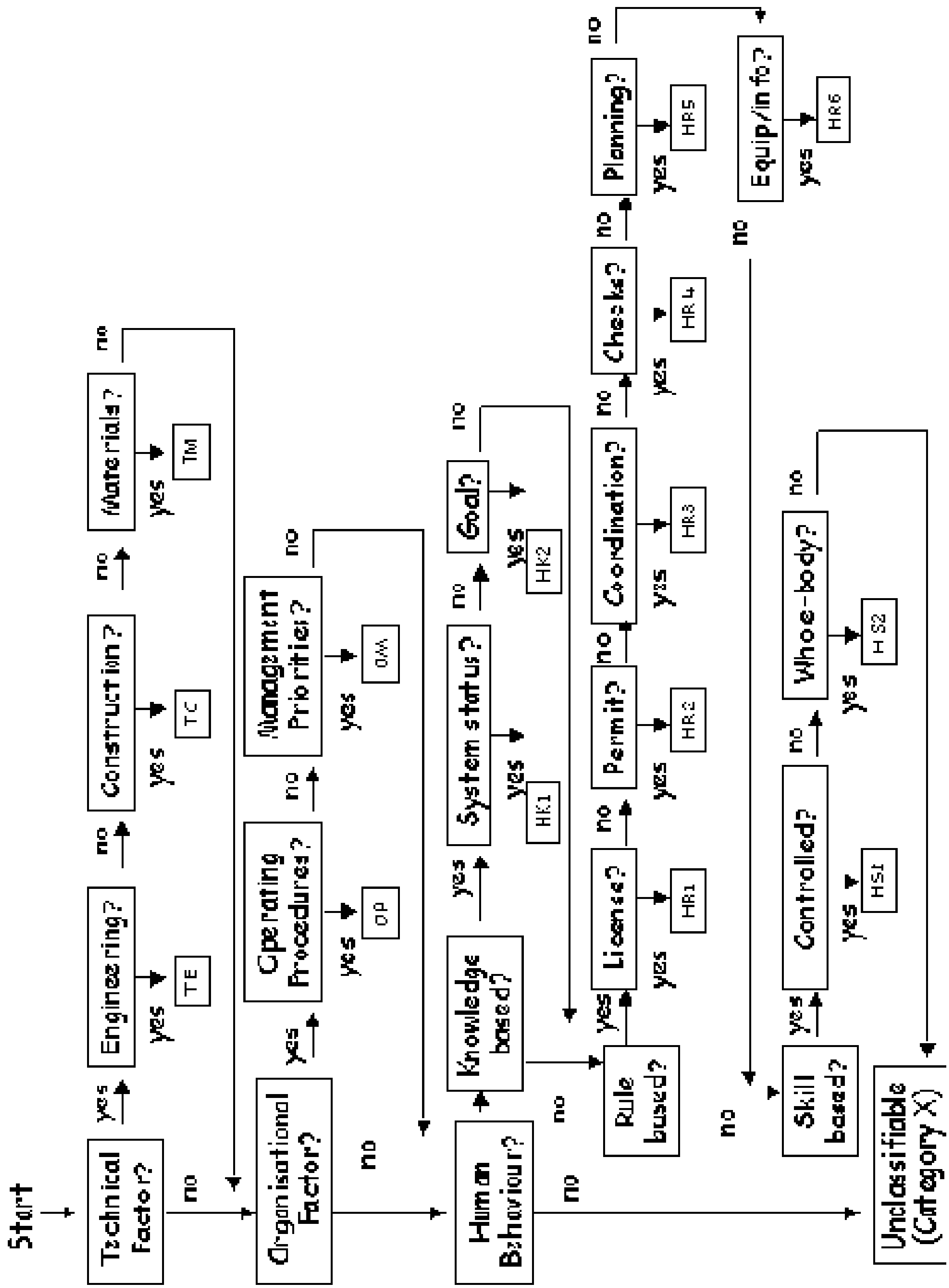

Figure 2.2: Eindhoven Classification Model

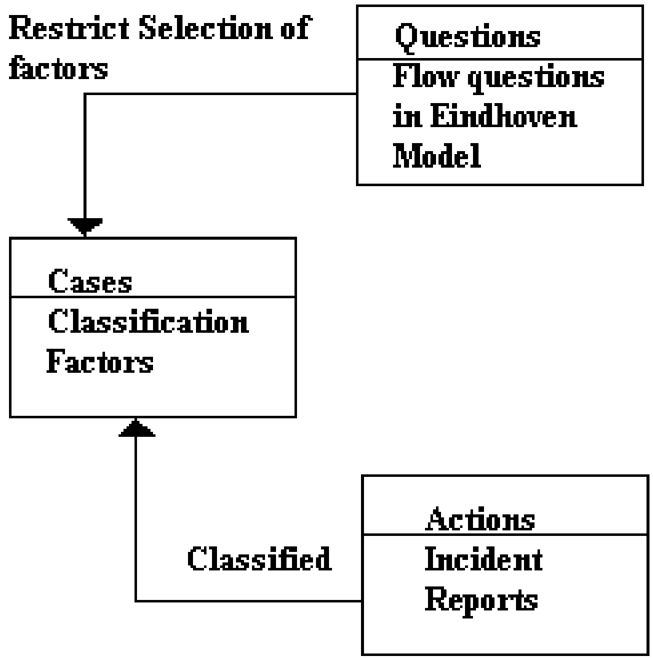

Figure 2.3: Eindhoven Case Library

2.7 The Classification

2.7.1 Causal Tree Incident Description Method

Causal trees (Van Vuuren and Van der Schaaf, 1995), are essentially derived from fault trees. They are used as a means to present, in chronological order, the decisions and critical activities that occur during an incident. Also, they are used to convey how all decisions and activities relate to each other in a logical sense. When representing near misses using causal trees the tree is divided into two sections:

Failure side - which represents the activities which lead to the failure.

Recovery side - which represents activities that have prevented the incident from becoming a real accident.

The Causal Tree forces us to understand that there is never one single cause of incidents but in fact many factors contribute. These factors are a combination of technical, organisational and human causes and are found at the bottom of the tree on the failure side. Due to the large amount of incident data to be classified it was not feasible to produce a Causal Tree for every incident but an example from incident number 414249 is shown in figure 2.4. This represents our use of a Causal Tree using the Eindhoven Classification Model. The causal tree method will be used later (see section 2.10.2 The Experiment) to validate the classification technique used (see following section 2.7.2 Rough Classification). The incident narrative that the tree was based on is as follows:

"I WAS ON A ROUTINE TRAINING IN AN R22 HELI WITH A HIGH TIME AIRPLANE PLT CAPT. WHILE CONDUCTING NORMAL TFC PATTERNS IN THE PRACTICE AREA AT AND BELOW 500 FT AGL, A FELLOW HELI INSTRUCTOR WARNED ME OF A LOW FLYING FIXED WING IN THE AREA APPEARING TO OPERATE HAZARDOUSLY. AFTER A WHILE THE COMPANY HELI DEPARTED AND I REPOSITIONED TO HIS LOCATION. WHILE ON A TKOF RUN, I SUDDENLY SPOTTED THE SAME ACFT MENTIONED PREVIOUSLY HEADED STRAIGHT FOR US. HE APPEARED TO BE DSNDING INTO OUR CLB. I TOOK THE CTLS AND ENGAGED IN EVASIVE MANEUVERS. THE PLANE STILL APPEARED ON A COLLISION COURSE. SO TO PREVENT COLLISION, I ENTERED AUTOROTATION AS A QUICK DSCNT TACTIC AND COULD HEAR THE AIRPLANE'S ENG DUE TO OUR PROX. I IMMEDIATELY ATTEMPTED COM THROUGH FREQS 123.025, 123.45, 123.30 TO ALERT T HE FIXED WING. ONLY RESPONSE WAS FROM ANOTHER AIRPLANE WHO ALSO OBSERVED THE C152'S OPS. THE AIRPLANE (C152) CONTINUED TO FLY WITHIN DANGEROUS PROX TO OUR PATTERN BTWN 400 FT AND BELOW. WE FINALLY DEPARTED THE AREA IN FEAR OF OUR SAFETY AND OBSERVED HIS SUBSEQUENT DEP TOWARD FXE. I FINALLY GOT HIS IDENT NUMBER OVER TWR 120.9 AND LEARNED THAT IT WAS FBO'S. I MADE A LNDG ON THEIR RAMP, APCHED PLT. HE KNEW WHY I WAS THERE AND JOKINGLY SAID WE GOT A LITTLE CLOSE. I LEARNED THAT HE WAS AN EXAMINER GIVING A COMMERCIAL CHKRIDE. I WANTED TO MAKE HIM AWARE OF OUR ONGOING HELI OPS IN THAT AREA SO HE COULD

KEEP AN EYE OUT. HIS LACK OF CONCERN WARRANTED ME TO CALL THE SCHOOL LATER AND AGAIN MAKE THEM EXTRA AWARE THAT HELIS SHARE THE AREA AND TO PAY ATTN TO OUR POSSIBLE PRESENCE."

From the tree we could identify the root causes as the quality of the operating procedures may not be adequate (OP in the classification model), in addition the operator of the fixed wing aircraft may not have understood the goals (HK2 in the classification model) and also that the job was not planned correctly (HR5 in the classification model). The creation of a causal tree is a subjective process and therefore two different people could create different trees and subsequently have slight differences in the root causes identified.

2.7.2 Rough Classification

As mentioned in the previous section it was not feasible to produce a causal tree for every one of the 100+ incidents that I analysed. Therefore another more general classification technique had to be created. The solution was to use the method outlined below:

> The first stage of the classification was to associate each anomaly which could occur, with a factor(s) in the Eindhoven Model then extend this association to the actual incident reports which contain one or more anomaly cited as the cause of the incident. Thus providing a quick step rough classification. A list of anomalies and their classification can be found in Appendix A.

Using these rough results each incident from the rotary vehicles and air traffic control data was classified under the Eindhoven Model according to which anomaly(s) had caused the incident. The rotary vehicles data consisted of 51 incident

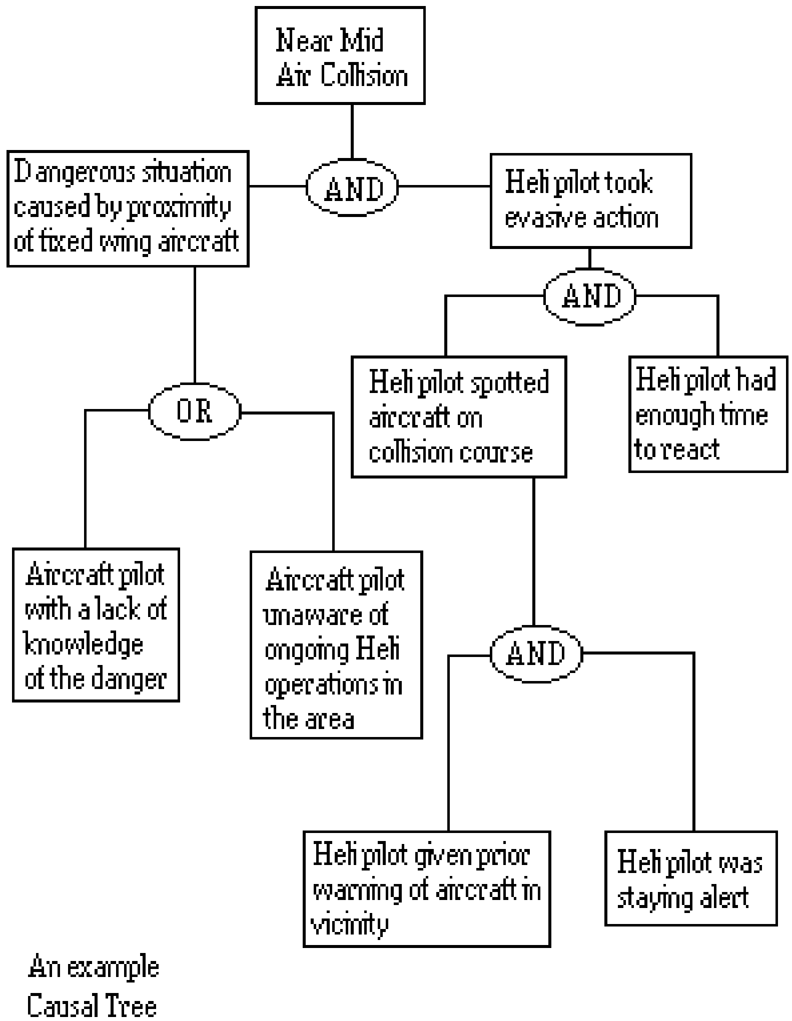

Figure 2.4: Example Causal Tree

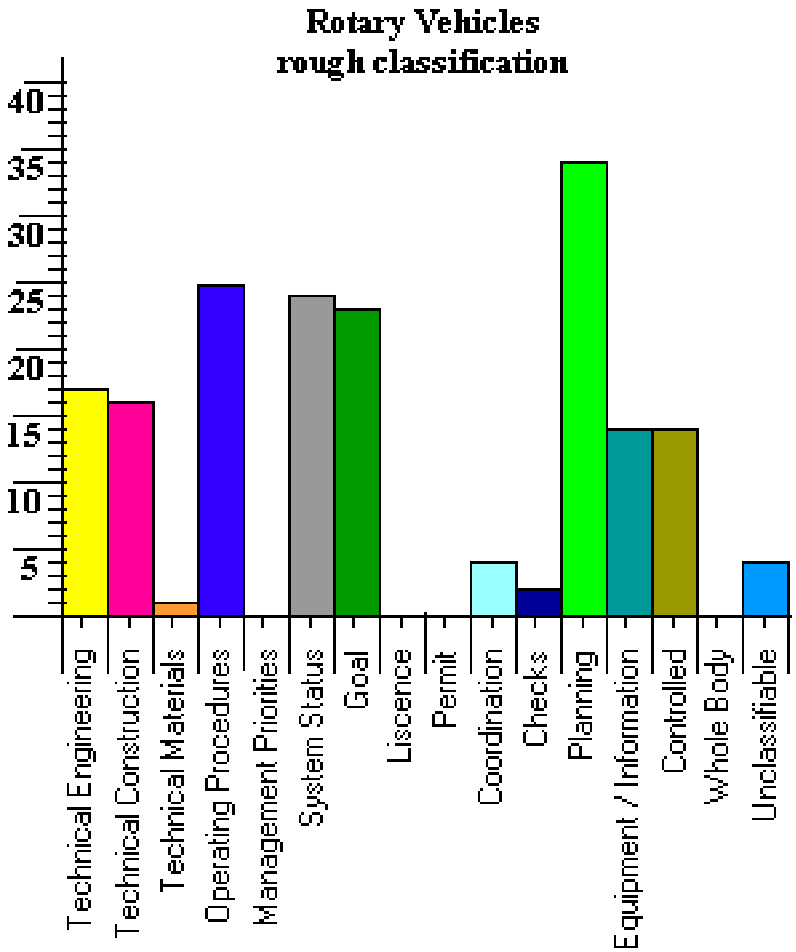

Figure 2.5: Rotary Vehicles Rough Results

reports and the air traffic control data 50 incidents. The results obtained can be found in figure 2.5 : for the Rotary Vehicles and figure 2.6: for the Air Traffic Control results. As with the causal tree approach the method is subjective and therefore can be prone to individual bias. From viewing figures 2.5 and 2.6 Operating Procedures, System Status, Goal and Planning are the most common causes identified consistently in both sets of incident data. This could indicate that the specific anomalies associated with these factors occur more often than any others.

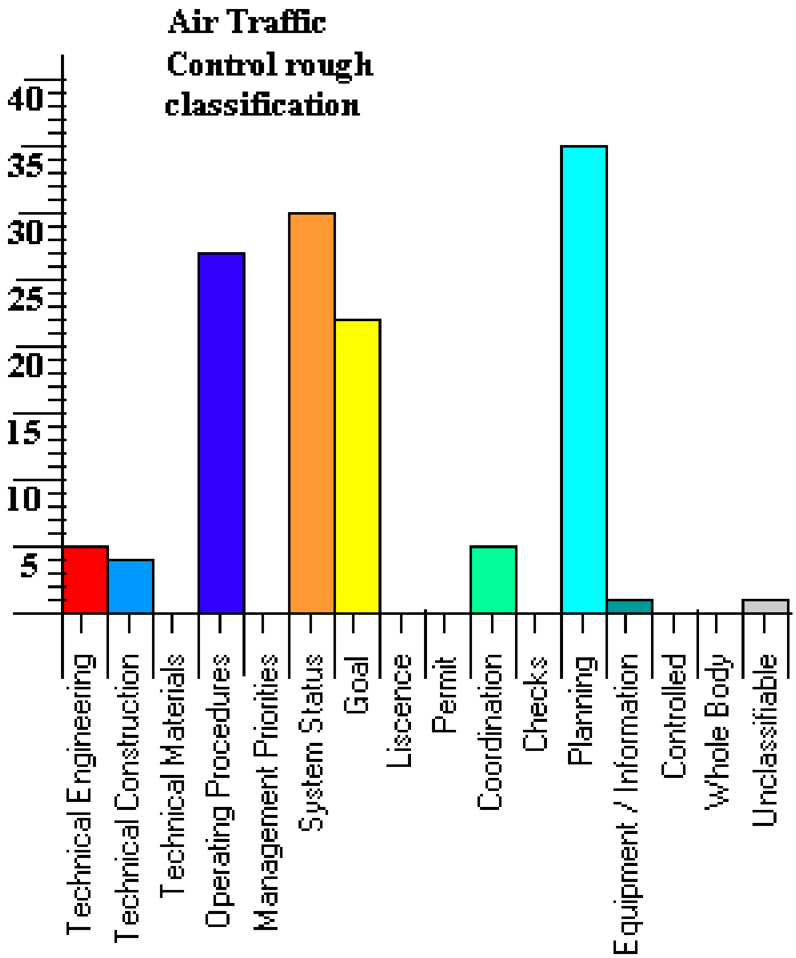

Figure 2.6: Air Traffic Control Rough Results

2.8 Classification Refinement

Once the rough classification was achieved it became clear that some refinement of the results was needed, for example it was noticed (from reading the synopsis of specific reports in the air traffic control incident data) that there was more than one incident which has an unclassifiable root cause. The initial rough classification does well to classify all the incidents but it does not attempt to classify certain specifics of each incident which reflects the individuality of each case. In addition the old menace of "subjective bias" should be reduced as much as possible. Reading the synopsis of each individual incident, the results therefore were refined where it was deemed appropriate, by adding or removing factors from the original classification. For example incident 402434 in the rotary vehicles database was originally classified with root causes HR5, OP and HK2. The narrative and synopsis of this incident is as follows:

"WHILE ENROUTE TO HELI BASE I CALLED JAX APPROACH ON 2 DIFFERENT FREQEUNCIES FOR RADAR ADVISORIES BUT DID NOT RECEIVE ANY REPLY. I CLIMBED ABOVE 2600 FT MSL TO GO OVER THE TOP OF NZC CLASS D AIRSPACE. APPROX 3 NM NORTH OF NZC I HAD A NEAR MISS WITH A FLIGHT OF 2 FA18 HORNETS. THESE AIRCRAFT WERE IN A TIGHT FORMATION IN WHAT APPEARED TO BE A CLIMBING L TURN."

"NMAC BETWEEN THE REPORTER IN A SMALL HELICOPTER IN LEVEL FLIGHT OVER THE TOP, AND APPROX 3 MILES AWAY FROM A CLASS D MILITARY BASE AIRSPACE AND A FLIGHT OF 2 F18 MIL HORNETS IN A CLIMBING L TURN. THERE WAS NO TIME FOR THE REPORTER TO TAKE EVASIVE ACTION."

From the synopsis we can justify the original classification, OP because the procedures used to ensure such near mid air collisions with military aircraft dont occur could be inadequate and HR5 because the job may not have been planned correct-

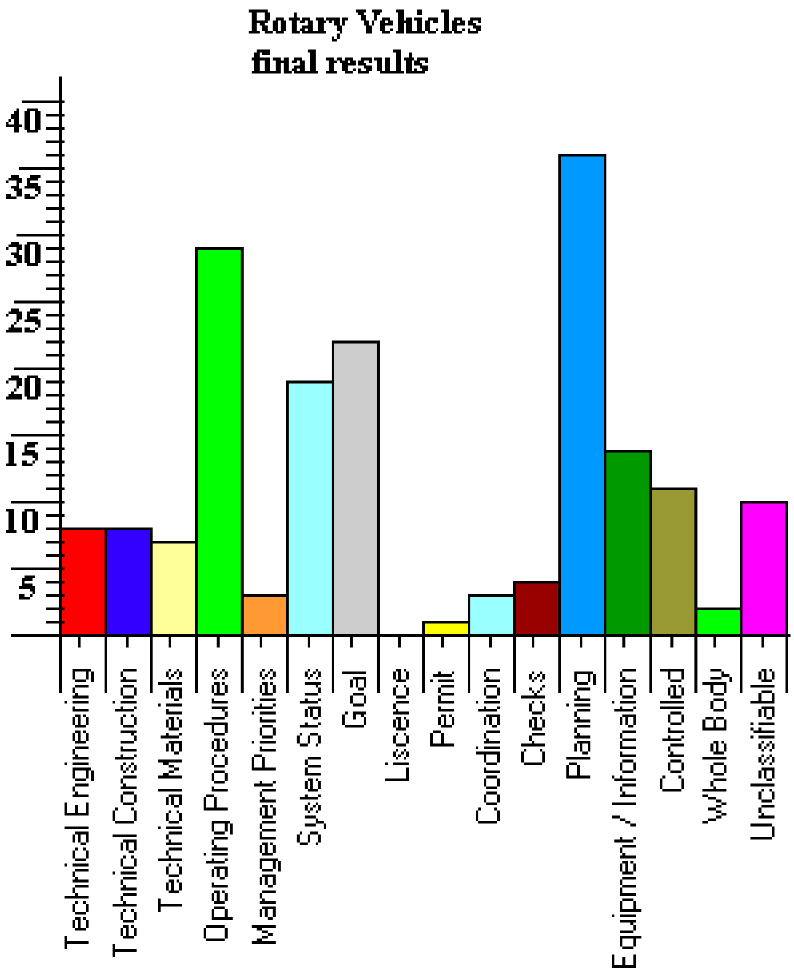

Figure 2.7: Rotary Vehicles Modified Results

ly i.e. was it wise to choose a flight path so close to military airspace? However the original classification of HK2 was wrong. It is clear from the synopsis and narrative that the goals were understood by the operator therefore this root cause should be removed from our original classification. The modified results can be viewed in figure 2.7: for the Rotary Vehicles and figure 2.8: for the Air Traffic Control results. It is to be noted that ideally this refinement should be by an independant analysist but due to the size of the task and the filled schedules of the few people qualified this was not possible.

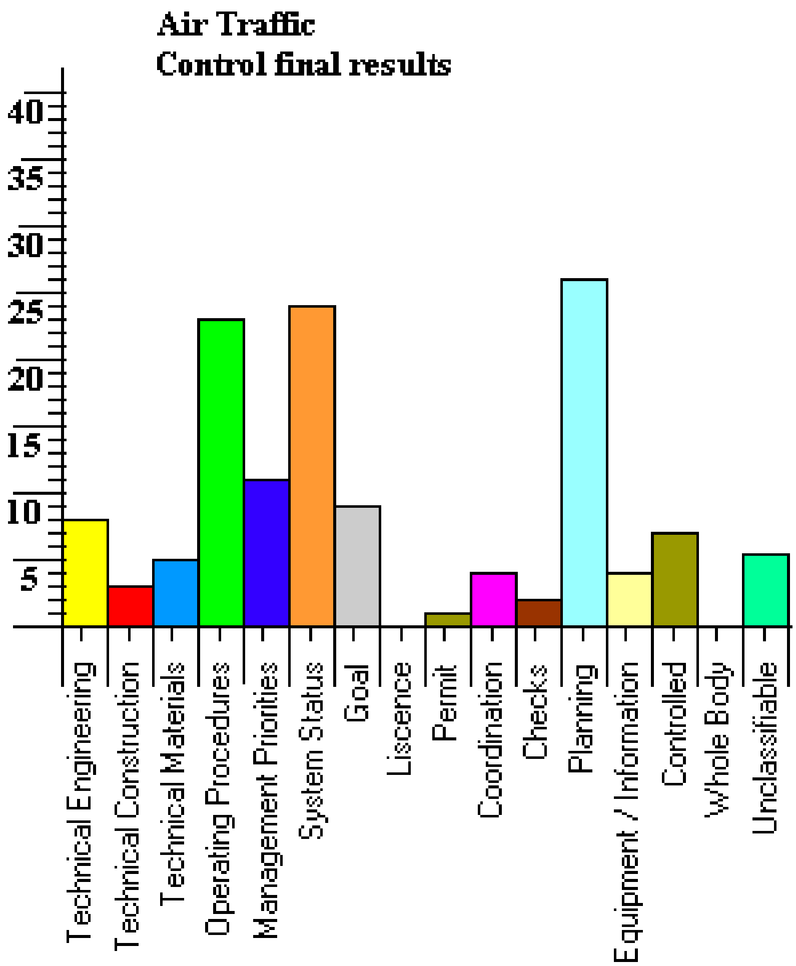

Figure 2.8: Air Traffic Control Modified Results

2.9 Statistical Analysis of Results

The mean of each factor was calcutated to give us an idea of which factors in the Eindhoven model were more common than the average and therefore more likely to be a root cause of a rotary vehicle incident or air traffic control incident. The varience was used to calculate the standard deviation thus giving us an idea of the spread of how much each factor has been identified as a root cause of an incident.

2.9.1 Rotary Vehicle Analysis Rough Results

Calculating the mean of each factor we arrive at; X = 177/16 = 11.0625

Calculating the variance we get; Var = 3859/16 - (11.0625)x(11.0625)

Var = 241.1875 - 122.3789063

Variance = 118.8085938

Standard Deviation = 10.89993549

We can see from the figures (figure 2.9) that the most likely cause tends to be a lack of planning (HR5), i.e. the appropriate methods where not chosen, or carried out in order. This cause scores well over the mean score and 9 incidents above the next cause. The next most likely causes are Operating Procedures (OP), System Status is not known to the operator (HK1) and the goals are not understood by the operator (HK2), scoring more than twice the mean. Also from the figures we can see that unexplainable material defects (TM), the operator was not qualified (HR1), the operator must obtain a permit (HR2) and correct whole body movement (HS2) are not likely causes of rotary vehicle incidents.

Scores	Scores X	Scores squared X x X
TE	17	289
TC	16	256
TM	0	0
OP	25	625
OM	0	0
HK1	24	576
HK2	23	529
HR1	0	0
HR2	0	0
HR3	4	16
HR4	2	4
HR5	34	1156
HR6	14	196
HS1	14	196
HS2	0	0
X	4	16

Figure 2.9: Rotary Vehicle Rough Results

2.9.2 Air Traffic Control Analysis Rough Results

Calculating the mean of each factor we arrive at; X = 140/16 = 8.75

Calculating the variance we get; Var = 3506/16 - (8.75)x(8.75)

Var = 219.125 - 76.5625

Variance = 142.5625

Standard Deviation = 11.939995394

We can see from the figures (figure 2.10) that the most likely cause tends to be a lack of planning (HR5), i.e. the appropriate methods where not chosen, or carried out in order. This cause scores well over the mean score and 5 incidents above the next cause. The next most likely causes are System Status is not known to the operator (HK1), Operating Procedures (OP) and the goals are not understood by the operator (HK2), each scoring more than twice the mean. Also from the figures we can see that unexplainable material defects (TM), Management Priorities (OM), the operator was not qualified (HR1), the operator must obtain a permit

scores	Scores X	Scores squared X x X
TE	5	25
TC	4	16
TM	0	0
OP	27	729
OM	0	0
HK1	30	900
HK2	22	484
HR1	0	0
HR2	0	0
HR3	5	25
HR4	0	0
HR5	35	1225
HR6	1	1
HS1	10	100
HS2	0	0
X	1	1

Figure 2.10: Air Traffic Control Rough Results

(HR2), the operator did not check local system status (HR4) and correct whole body movement (HS2) are not likely causes of the air traffic control reported incidents.

2.9.3 Rotary Vehicle Analysis Final Results

After the second pass, we now see that calculating the mean of each factor yields;

X = 177/16 = 11.0625

Calculating the variance we get; Var = 3615/16 - (11.0625)x(11.0625)

Var = 222.9375 - 122.3789063

Variance = 100.5585938

Standard Deviation = 10.02789079

From the revised results (figure 2.11) we can see there is a better spread across

Scores	Scores X	Scores squared X x X
TE	8	64
TC	8	64
TM	7	49
OP	2	841
OM	3	9
HK1	19	361
HK2	22	484
HR1	0	0
HR2	1	1
HR3	3	9
HR4	4	16
HR5	36	1296
HR6	14	196
HS1	11	121
HS2	2	4
X	10	100

Figure 2.11: Rotary Vehicle Final Results

the mean with a standard deviation of around 8. We still have some exceptional results however, like the largest probable cause tends to be a lack of planning (HR5), i.e. the appropriate methods where not chosen, or carried out in order. This cause scores well over the mean score and 14 incidents above the next cause. The operator was not qualified (HR1) still scores 0 after the revision and therefore seems the least likely cause in event of an incident.

2.9.4 Air Traffic Control Analysis Final Results

Similarly , the results for the second pass with Air Traffic Control incidents yields:

Calculating the mean of each factor we arrive at; X = 132/16 = 8.25

Calculating the variance we get; Var = 2192/16 - (8.25)x(8.25)

Var = 137 - 68.0625

Variance = 68.9375

Standard Deviation = 8.302860953

From the revised results (figure 2.12) we can see there is a better spread across the mean with a standard deviation of around 8. We don't have any exceptional results however, but some causes score around the mid twenty mark like the largest probable cause, a lack of planning (HR5). This cause scores three times the mean score and 2 incidents above the next likely cause. The operator was not qualified (HR1) still scores 0 after the revision along with correct whole body movement (HS2) and therefore both seem the least likely cause in event of an incident. It is to be noted that the affect of the second pass tends to smooth out the original analysis results in both cases. This may not entirely be a good thing as it would have the affect of making cases appear more similar. It is vital therefore, that before changing an original classification the analyst must be completely sure of why they are doing so.

2.10 Validation of the Classification Technique

2.10.1 Rationale for Classification Validation

Due to the large number of incident reports to be classified (101 in all) it was not feasible to carry out a Causal Tree analysis on each. Instead, a rough classification based on the type of anomaly for each specific incident was carried out. Then each incident was analysed subjectively in terms of the 'NARRATIVE' and 'SUMMARY' parts and the initial classifications modified accordingly. Each incident classification was not as detailed as it would have been if a full Causal Tree analysis was used. Therefore, the technique used should be evaluated in some way, too see how accurate the root cause identification was.

Scores	Scores X	Scores squared X x X
TE	8	64
TC	3	9
TM	5	25
OP	23	529
OM	11	121
HK1	24	576
HK2	9	81
HR1	0	0
HR2	1	1
HR3	4	16
HR4	2	4
HR5	26	676
HR6	4	16
HS1	7	49
HS2	0	0
X	5	25

Figure 2.12: Air Traffic Control Final Results

2.10.2 The Experiment

It was decided that a Causal Tree evaluation should be carried out on a sample incident from both the air-traffic controller's and rotary vehicle's incident database, by two test subjects (to eliminate the effect of investigators bias) with experience in accident analysis. The results could then be compared to my original classifications, therefore, giving an idea of how accurate the original root causes were. The two subjects involved in the experiment were both in the final stages of their PhD's with experience in Pshycology and Computing Science and also major experience in accident and incident analysis. Therefore the feedback gained would be of the highest quality. The two incidents used are as follows, from the rotary vehicle's incident database:

ACCESSION NUMBER : 412640

DATE OF OCCURRENCE : 9808

REPORTED BY : FLC; ;

PERSONS FUNCTIONS : FLC,PLT; FLC,PLT;

FLIGHT CONDITIONS : SVC

REFERENCE FACILITY ID : ONP

FACILITY STATE : OR

FACILITY TYPE : ARPT;

FACILITY IDENTIFIER : ONP;

AIRCRAFT TYPE : SMA; ;

ANOMALY DESCRIPTIONS : VFR IN IMC; OTHER; NON ADHERENCE LEGAL RQMT/FAR; ERRONEOUS PENETRATION OR EXIT AIRSPACE;

ANOMALY DETECTOR : COCKPIT/FLC;

ANOMALY RESOLUTION : NOT RESOLVED/ANOMALY ACCEPTED;

ANOMALY CONSEQUENCES : NONE;

NARRATIVE : DEPARTING NEWPORT ARPT, AT THE TIME OF DEP, THE W HALF OF THE ARPT WAS STARTING TO FOG IN. I HOVER-TAXIED TO THE FAR E END OF THE ARPT AND WAS ABLE TO TAKE OFF IN BLUE SKIES AND UNLIMITED VISIBILITY. THIS ARPT IS SET UP FOR A CTL ZONE WHEN THE VISIBILITY IS LESS THAN 3 MI AND A 1000 FT CEILING. THERE WAS ANOTHER HELI IN THE PATTERN WHOM I WAS IN RADIO CONTACT WITH. HE GAVE ME PERMISSION TO TAKE OFF FIRST AND THEN HE WENT IN AND LANDED. ALL OF THIS WAS DONE VFR ON THE E END OF THE FIELD WHILE THE W END WAS FOGGED IN. THE STANDARD FOR THE OTHER ARPTS WITH CTL TWRS HAS BEEN IF I WAS INSIDE OF THEIR CTL ZONE AND IT WAS IN EFFECT, THEY HAVE ALLOWED ME TO WORK INSIDE THE CTL ZONE WITHOUT A SPECIAL VFR IF I WAS IN THE STANDARD VFR CONDITIONS. ALL I NEEDED TO DO WAS MAKE RPTS OF MY LOCATIONS WHILE WORKING IN THEIR AIRSPACE. AS LONG AS I WAS VFR, I DID NOT NEED A SPECIAL VFR TO BE INSIDE THE AIRSPACE. MY POINT TO ALL OF THIS IS THAT IT IS NOT TAUGHT TO NEW STUDENTS THIS WAY SO IT BECOMES MORE LIKE JUST A STORY WHEN AN OLDER PLT DOES SOMETHING LIKE THIS. IT IS LEGAL TO DO BUT NOT GOOD FOR STUDENTS TO SEE. NOT SURE OF HOW OR WHERE TO MAKE A POINT OF

THIS, OR IF MAYBE IT IS NOT A RELATIVE POINT TO MAKE AT AL-L. HOPE THIS IS NOT TOO CONFUSING, AND THANK YOU FOR YOUR TIME.

MAKE-MODEL NAME : HELICOPTER; HELICOPTER;

FAR PART NUMBER : 91; UNK;

SYNOPSIS : A COMMERCIAL PLT TOOK OFF IN AN SMA HELI FROM A DESIGNATED CLASS E NON TWR ARPT WHEN WX CONDITIONS REQUIRED A SVFR CLRNC. THE PLT DID NOT HAVE A CLRNC, OR BELIEVED THAT HE NEEDED ONE, SINCE HE WAS OPERATING A HELI WHEN THE FLT VISIBILITY MET THE HELI FLT VISIBILITY REQUIREMENT.

REFERENCE FACILITY ID : ONP

FACILITY STATE : OR

DISTANCE & BEARING FROM REF. : 0

AGL ALTITUDE : 0,1200

From the rotary vehicle's incident database:

ACCESSION NUMBER : 425120

DATE OF OCCURRENCE : 9901

REPORTED BY : CTLR; ; ;

PERSONS FUNCTIONS : TWR,SUPVR; FLC,PLT; TWR,LC;

FLIGHT CONDITIONS : VMC

REFERENCE FACILITY ID : PWM

FACILITY STATE : ME

FACILITY TYPE : ARPT; TWR;

FACILITY IDENTIFIER : PWM; PWM;

AIRCRAFT TYPE : LTT;

ANOMALY DESCRIPTIONS : OTHER; NON ADHERENCE LEGAL RQMT/FAR;

ANOMALY DETECTOR : COCKPIT/FLC; ATC/CTLR;

ANOMALY RESOLUTION : CTLR INTERVENED;

ANOMALY CONSEQUENCES : NONE;

NARRATIVE : WX WAS SUNNY BUT COLD, A DAY OR 2 AFTER A S-NOW/ICE STORM. SABRELINER WAS TAXIING OUT FOR IFR DEP. ATC OBSERVED THE FUSELAGE WAS COVERED WITH SNOW AND ICE. ATC ADVISED THE PLT 'IT APPEARS THERE'S A LARGE AMOUNT OF SNOW AND ICE ON THE TOP OF YOUR ACFT.' THE PLT STATED 'IT'S NOT A LOT, IT'S A LITTLE, AND IT WILL BLOW OFF WHEN WE DEPART.' ON TKOF ROLL, ICE WAS OBSERVED PEELING OFF THE FUSELAGE. THIS CONTINUED AS THE ACFT CLBED OUT. ICE WAS OBSERVED FALLING ON OR NEAR A HWY JUST OFF THE DEP END OF THE RWY. THE ACFT WAS SWITCHED TO DEP, BUT A FEW MINS LATER RETURNED FOR LNDG. AS THE ACFT TAXIED IN, SIGNIFICANT ICE FORMATION WAS OBSERVED ON THE ELEVATORS. THE ACFT TAXIED TO AN FBO AND WAS DEICED BEFORE TAXIING BACK OUT FOR DEP. I SPOKE WITH THE FBO LATER. THEY SAID THEY HAD SEEN THE PLT CLRING SNOW AND ICE OFF THE ACFT BEFORE HE FIRST DEPARTED. HOWEVER, THE UPPER SURFACE OF THE ELEVATORS WAS TOO HIGH FOR THE PLT TO SEE FROM THE GND.
SYNOPSIS : SABRELINER PLT RETURN LAND TO PWM DUE TO RESID-UAL ICE ON ACFT.
REFERENCE FACILITY ID : PWM
FACILITY STATE : ME
DISTANCE & BEARING FROM REF. : 2
AGL ALTITUDE : 0,1000

In addition a sample evaluation sheet can found in Appendix G.

2.10.3 Results

Each analysis took approximately one hour to complete. For the rotary vehicles incident subject 1 produced the Causal Tree shown in figures 2.13, 2.14 and 2.15.

Subject 1 derived the following classification (built on elements of the Eindhoven

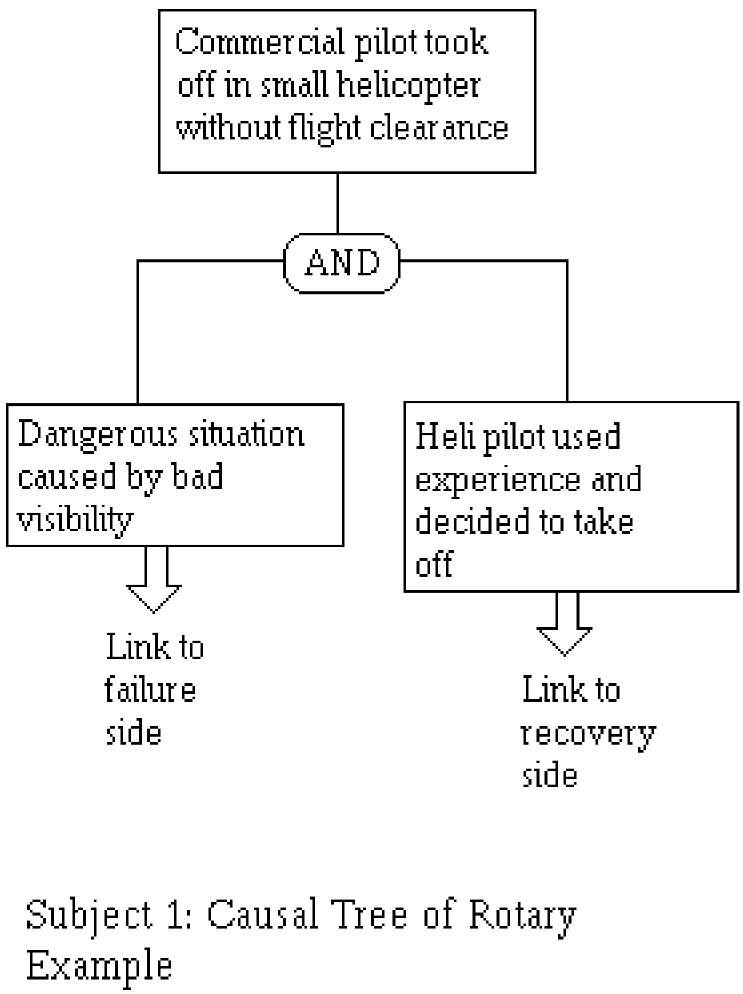

Figure 2.13: Causal Tree 1

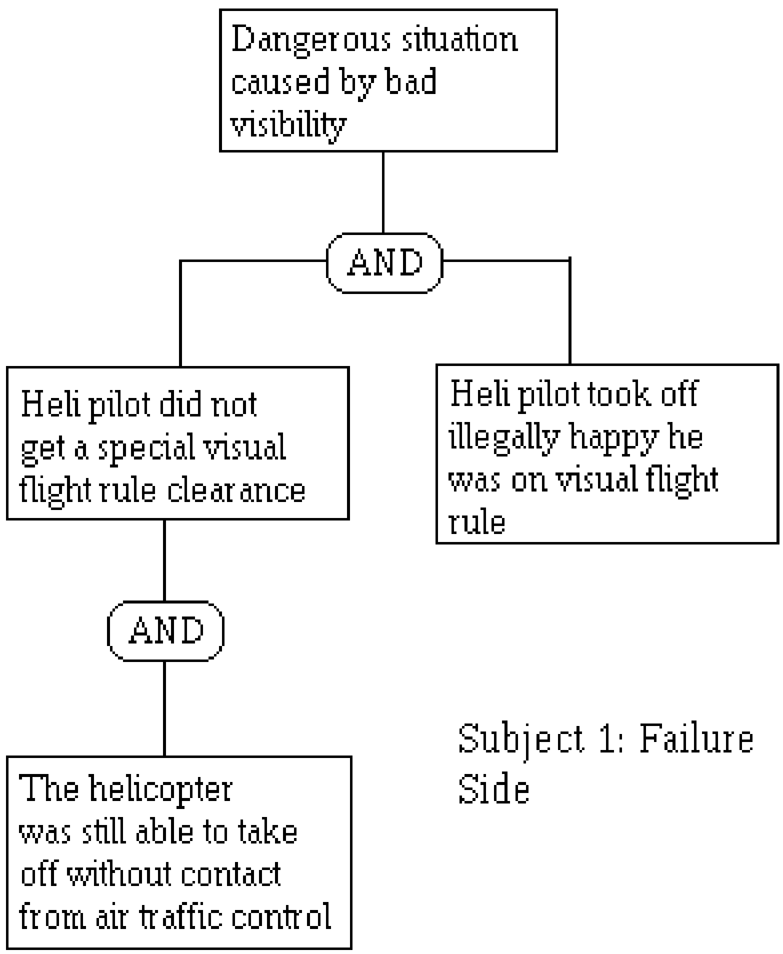

Figure 2.14: Failure Side

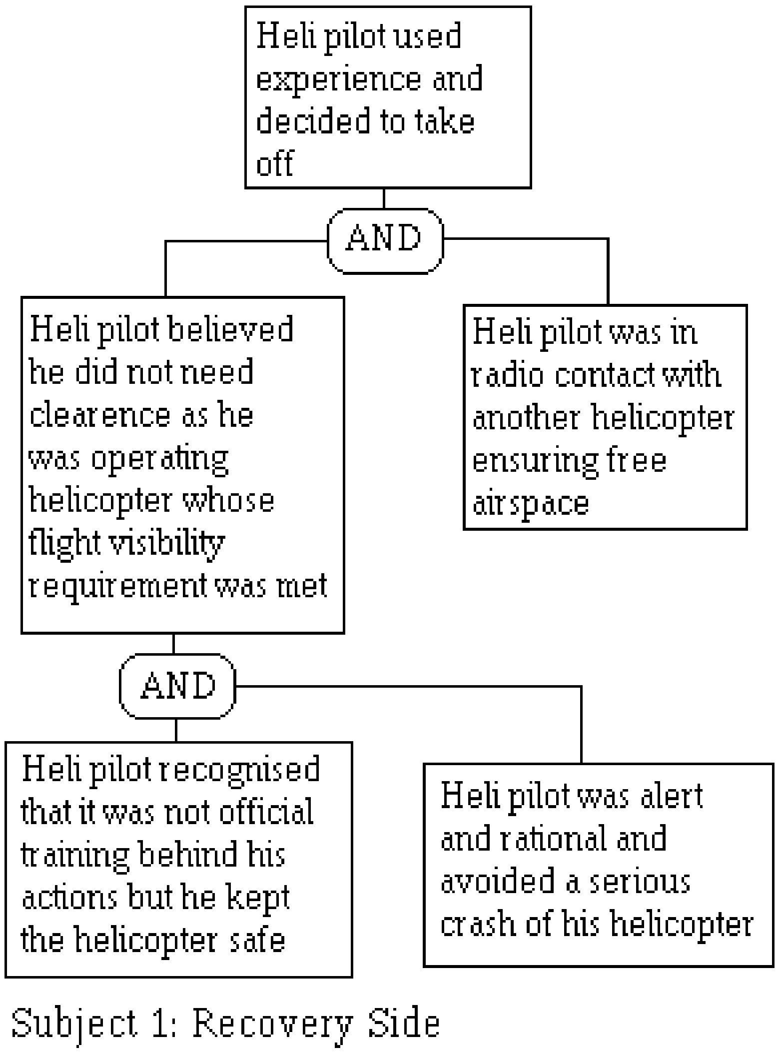

Figure 2.15: Recovery Side

Model introduced earlier) from the Causal Tree producing:

OP - (Operating procedures) refers to the inadequate quality of procedures
OM - (Management Priorities) refers to any de facto pressure by top - or middle management to deviate from the formal organisational priorities.
HR2 - If applicable the operator must obtain a temporary permit for activities where extra risk is involved.
HR5 - The job itself should be planned correctly, i.e. the appropriate methods should be chosen and carried out in order.
X - Unclassifiable behaviour

This contrasts with my original classification in which OP and HR5 were identified as above, but the other three causes that the subject identified, were not originally identified by myself. In addition, HK2 (The goal, or priorities of goals, must be known and understood by the operator.) was also identified in my original classification for this incident.

The Causal Tree produced by subject 2 for the air-traffic controller incident can be viewed in figures 2.16, 2.17, 2.18 and 2.19.

The following classification was derived from the tree by subject 2:

OP - (Operating procedures) refers to the inadequate quality of procedures
HK1 - The correct status and dynamics of the system to be controlled must be known to the operator.
HR2 - If applicable the operator must obtain a temporary permit for activities where extra risk is involved.
HR4 - When arriving at the job location the local system status should be checked to comply with the expected conditions.
HR5 - The job itself should be planned correctly, i.e. the appropriate methods should be chosen and carried out in order.
X - Unclassified behaviour.

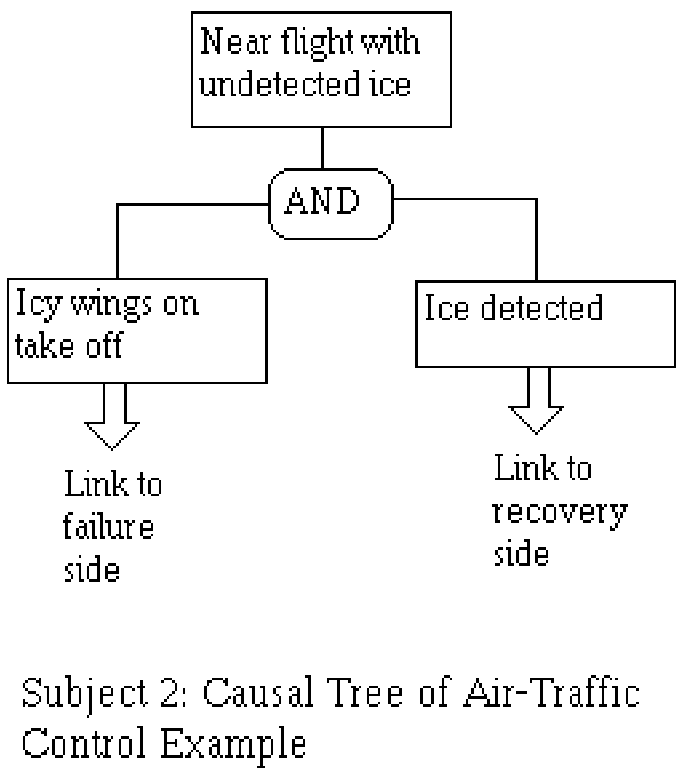

Figure 2.16: Causal Tree 2

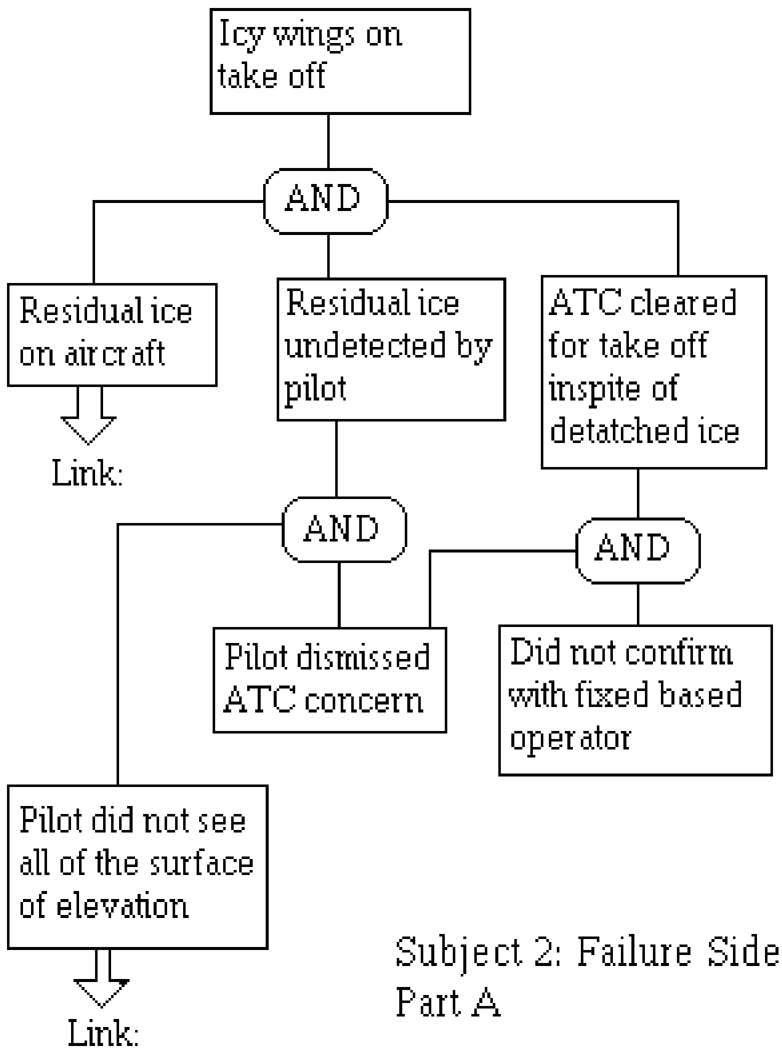

Figure 2.17: Failure Side Part A

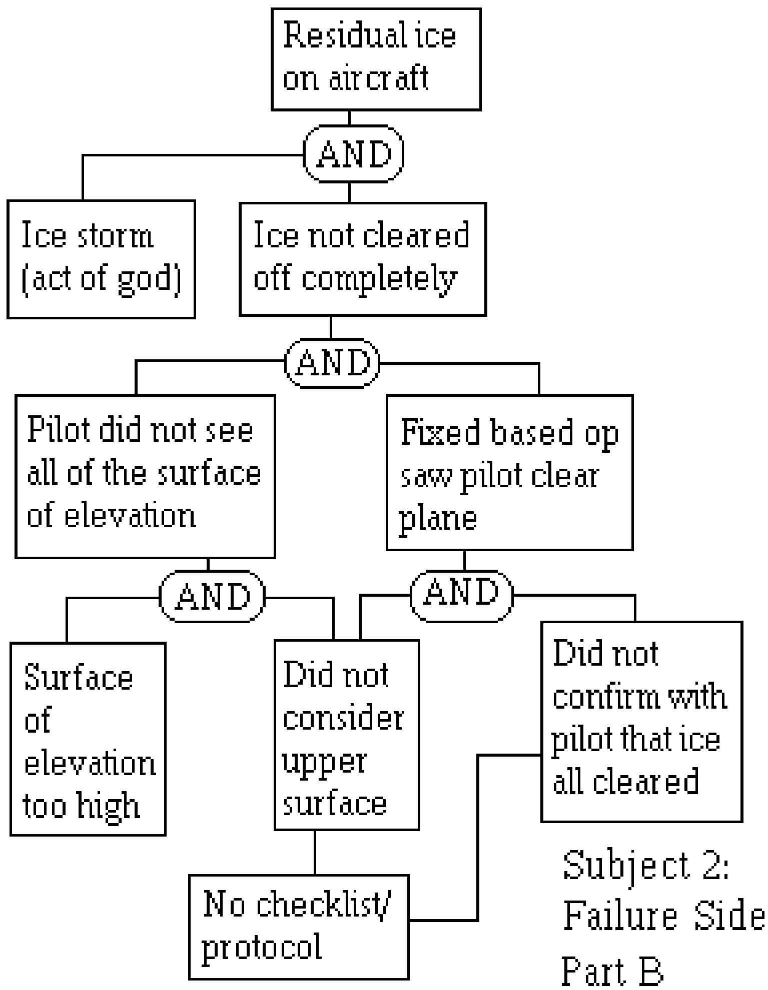

Figure 2.18: Failure Side Part B

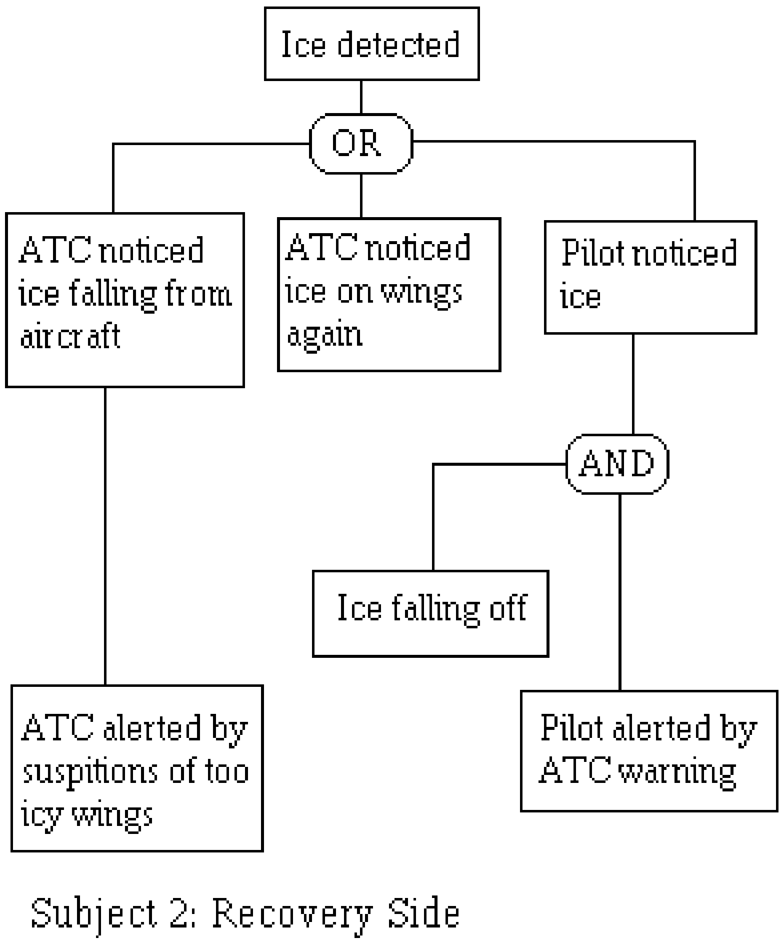

Figure 2.19: Recovery Side

In my original classification HR4, HR5 and X were identified as the root causes, but causes OP and HK1 were missed. In addition to the causes identified above HR1 (The operator in question must be qualified to do the job.) and OM ((Management Priorities) refers to any de facto pressure by top - or middle management to deviate from the formal organisational priorities.) were identified as possibilities but the subject could not be completely certain due to a lack of information in the incident report.

2.10.4 Conclusions

From the results obtained we can conclude that the method used to initially classify the incidents gives a good approximation of the root causes. The only draw back, however, is that the method can miss some of the less obvious causes that can be identified using more detailed analysis. This is clearly shown by the fact that the more in-depth analysis using causal trees produced 2-3 root causes missed by the original method. In addition, the incident reports themselves could be improved by the addition of necessary contextual information. This is highlighted by the fact two possible causes identified by the subject using the Causal Tree method could not be confirmed or rejected on the basis of the information presented in the incident report. Thus we can conclude that unless there is full access to all possible information or sources a proper analysis cannot be complete.

2.11 Testing The Case Libraries

2.11.1 General Case Libraries

To test the general case library's a series of test queries were used. The ranked results were then modified by selecting answers to the relevant questions until a total match was found. The total matches were viewed to ensure that the incidents retrieved were valid. The following is a walk through example of one of the initial tests carried out:

The following query was entered into the air traffic control case library:

EXHAUST FUMES

In plain English we are looking for incidents that involved exhaust fumes causing the anomaly. After running the query two incidents, in the library were displayed as possible matches. Each incident was viewed to check that the incidents returned were accurate and the rest of the case library viewed to make sure that no exhaust fumes incidents had been left out. (An example of the rankings found and one of the incidents returned and viewed with NaCoDAE can be seen in figure 2.20 and figure 2.21 respectively). Figure 2.20 shows the initial results of the query providing the relevent questions asscociated with the matched cases and the ranked score of the incidents matched. Answering the question would refine the query further updating the score of each matched case. Figure 2.21 shows one of the matched incidents viewed. This includes the name, title, description, questions and actions associated with the specific case.

2.11.2 Eindhoven Model Case Library

To test the Eindhoven Model case library a series of specific language queries were used to identify incidents with the same classifications and the results viewed to ensure the incidents retrieved were valid. The following is a walk through example of one of the tests carried out:

The following query was entered into the rotary vehicle case library:

Technical

The relevant question was answered to yield the resulting total matched case

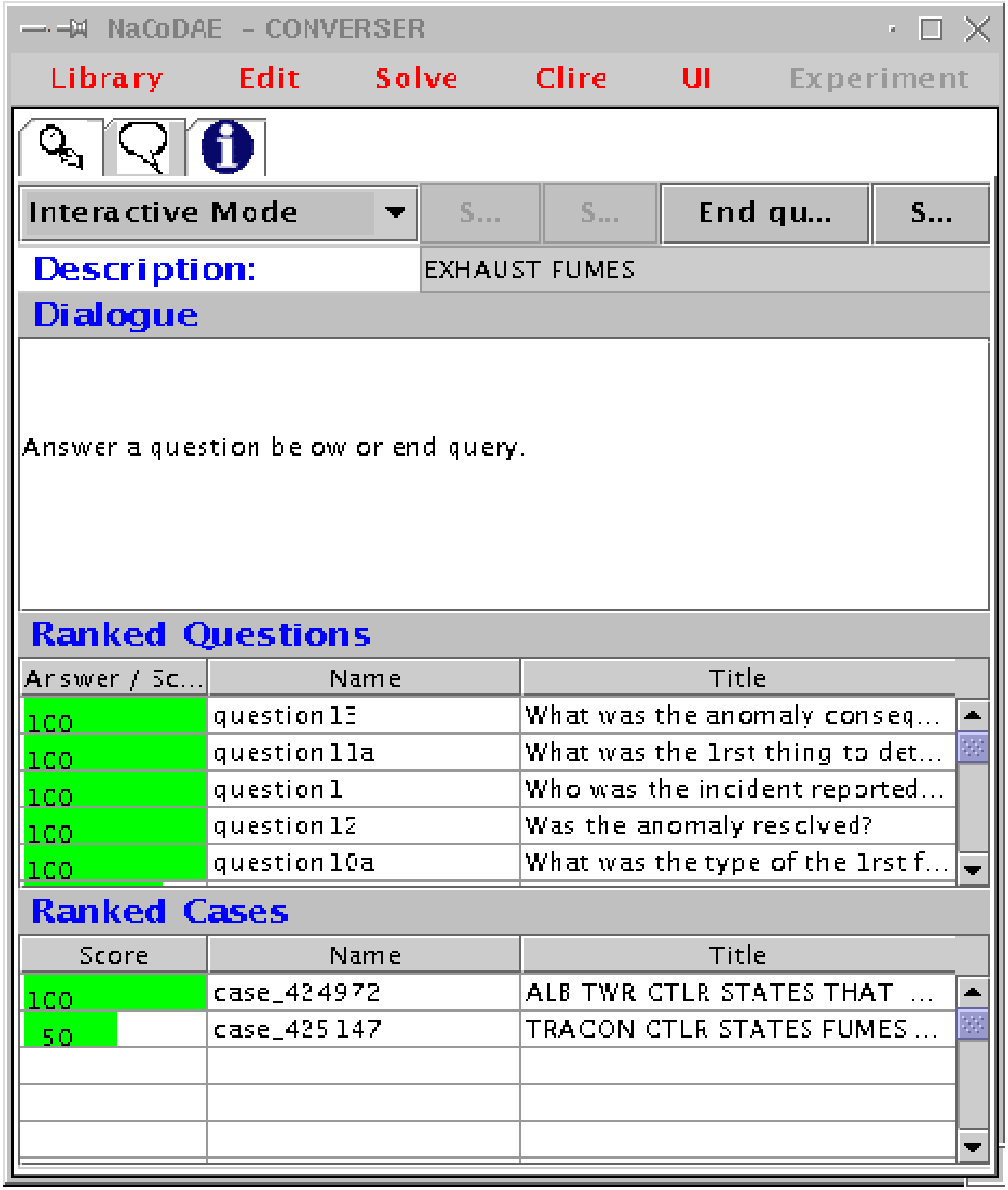

Figure 2.20: NaCoDAE Rankings

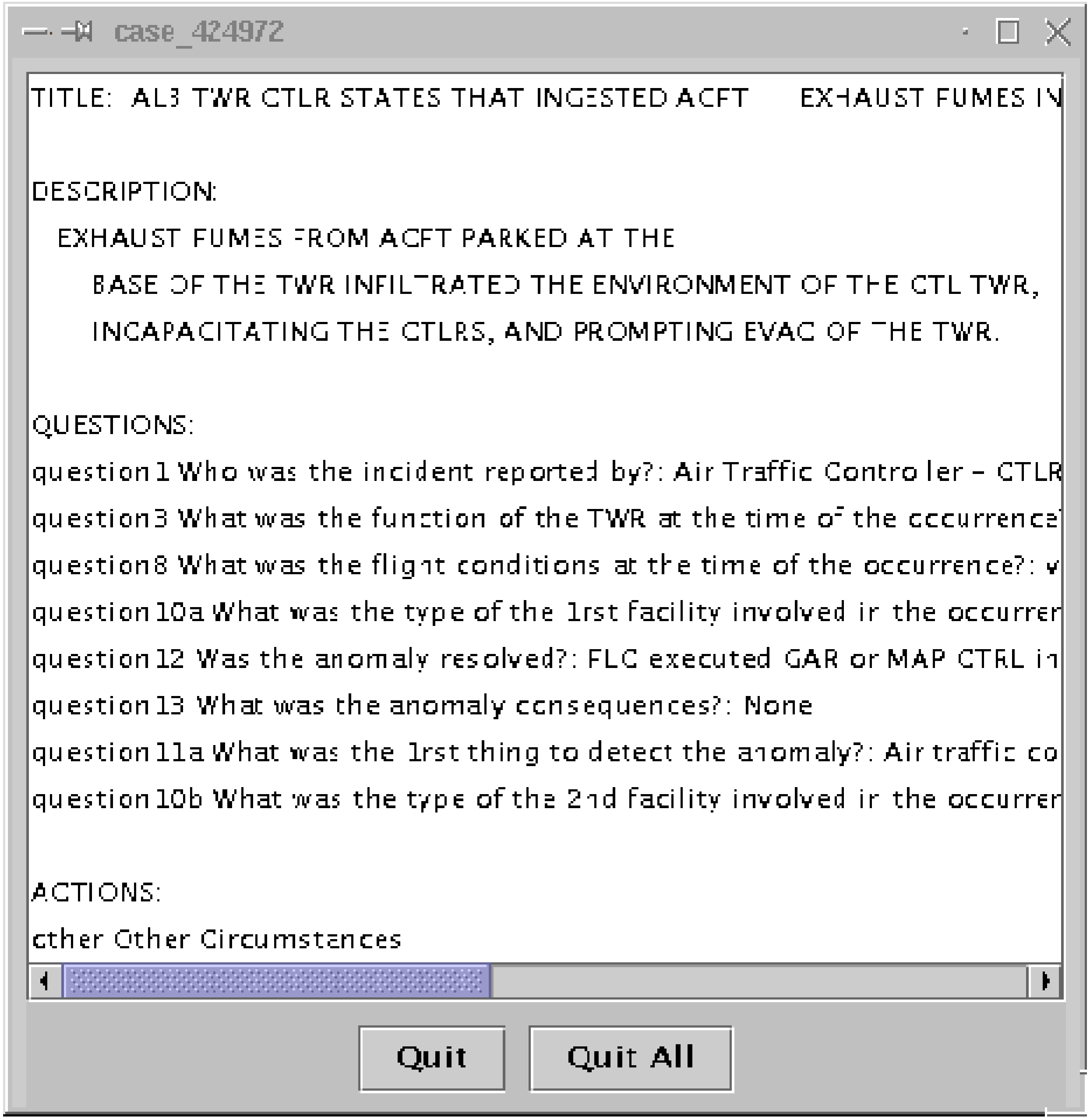

Figure 2.21: Incident Viewed

(As seen in figure 2.22). The actions (i.e. the incidents) were viewed to make sure they matched the classification (see figure 2.23). This query can be refined to select human and organisational factors also, by answering "yes" to the relevant ranked questions. Moreover certain factors may be negated by answering "no" to the specific question thus eliminating them from the procedings. Figure 2.22 shows the initial results of the query providing the relevent questions asscociated with the matched cases and the ranked score of the incidents matched. Figure 2.23 shows one of the matched incidents viewed. This includes the name, title, description, questions and actions associated with the specific case.

2.12 Validation Of The Case Libraries

To evaluate the case libraries a sample of test users were taken and how to use the NaCoDAE tool was explained to them for 5 minutes or so. They were then left to use the software and comment upon it using Think Aloud style techniques. The comments were taped and then all the tapes reviewed at a later date.

With the exception of having to explain the Eindhoven Classification model to two of the users who had never come across it before (both 4rth year Computing Science students at the time), all the users found the method of interaction with the case libraries simple and intuitive. In addition most users found the retrieval of the incident data very effective however, it was suggested that this could be improved (in the non Eindhoven Model library) if the system actually learned through experience i.e. the user telling the tool whether a retrieved document is relevant (relevence feedback). The earmarked documents rankings for the particular query could be then adjusted to be less important by the system and remembered for future cases.

Overall the relevance of the retrieved documents was preferred in the Eindhoven Classification Libraries as expected, due to the pre classification work on each incident beforehand. Further it was suggested that the Eindhoven Classification

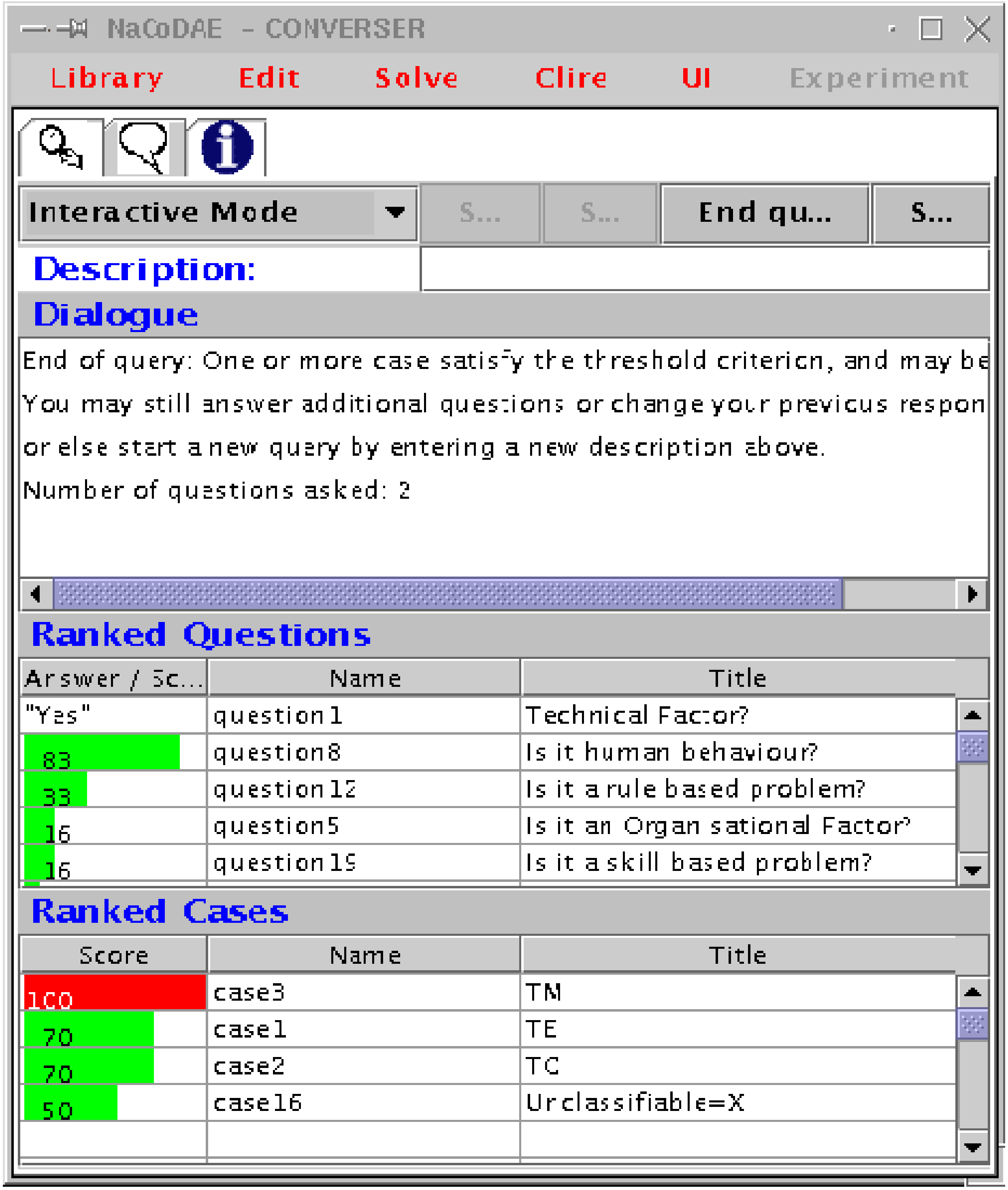

Figure 2.22: Matched Case

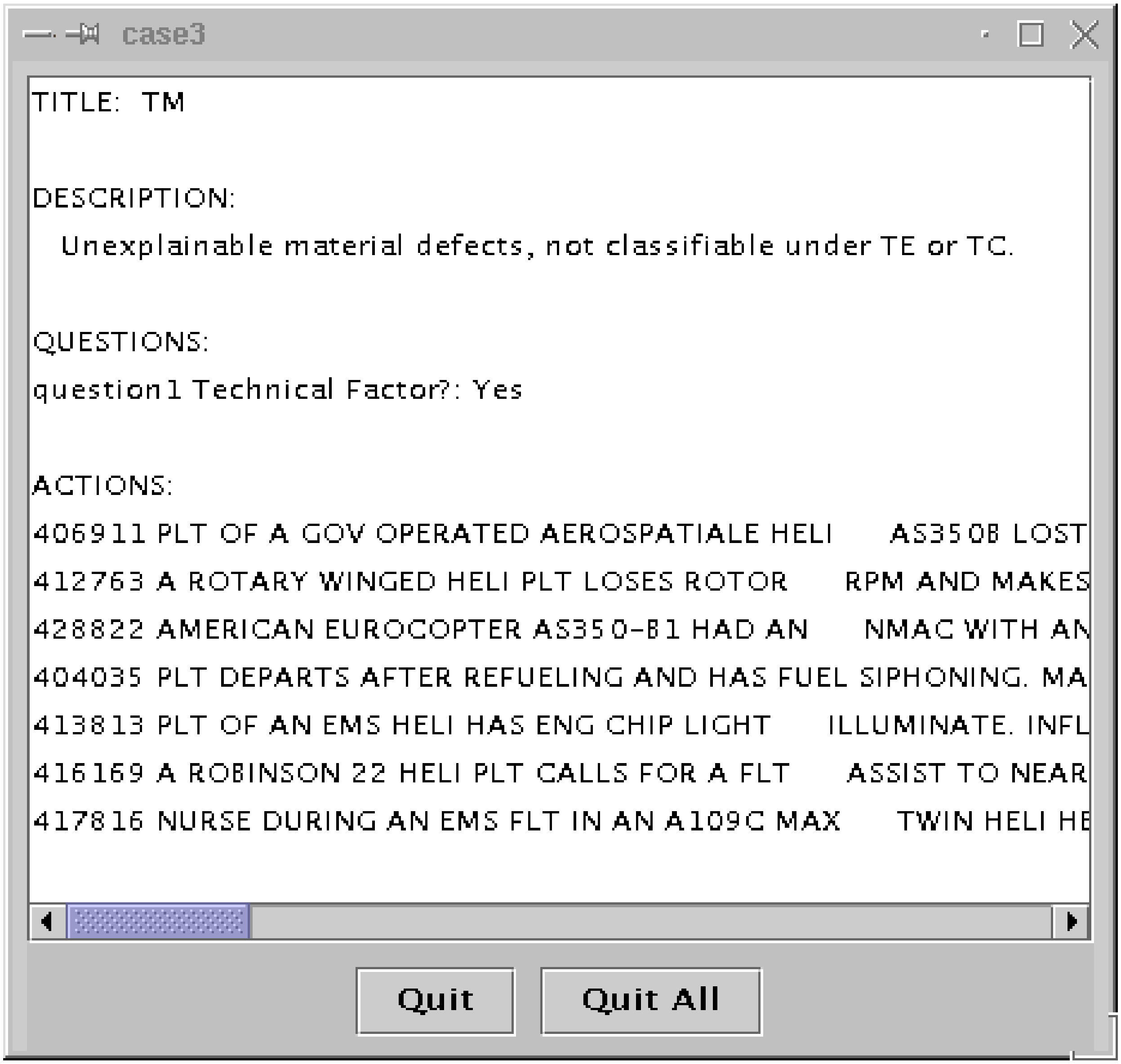

Figure 2.23: Actions

libraries could be improved if the Air Traffic Control incident data and the Rotary Vehicle data were joined together into the one library. This tactic would be most effective if all the incident data (i.e. Maintenance (Mechanic) and Cabin Crew incidents) from the ASRS web site were classified and joined to the same library.

Chapter 3

Information Retrieval

3.1 Inquery

3.1.1 Introduction

Following from the Case Based Reasoning method it is clear that we should look at existing Information Retrieval techniques to give us a counterpoint to the technology. Also, it may be useful to adapt these IR techniques to benefit the less general area of retrieving incident reports and decide how much benefit is really to be gained. These questions will be discussed in the following sections.

3.1.2 Overview

INQUERY is a probabilistic Information Retrieval (IR) system developed at the University of Massachusetts, Amherst. Inquery can retrieve documents relevant to a user's query when given a database of textual documents. These documents themselves need to be indexed by INQUERY's parsing system, although they do not need to be pre-classified by hand. The queries can be constructed in two ways firstly by natural language (i.e. a regular English sentence), and secondly in a more exact structured query language a description of which follows in section Structured Query Language.

3.1.3 xinquery

Xinquery is an X interface to the INQUERY retrieval engine using Tcl/tk X application prototyping tools. The actual retrieval engine application is contained within the 'tclinq' program. Xinquery is merely the front end to this program. ('tclinq' must be on your path when you run xinquery). The tclinq program allows for customised startup files and will also accept database retrieval commands. The xinquery interface includes pull-down menus which allow for definitions of collections, formulation of queries and browsing of terms, as well as buttons defining how searches are to proceed and selection of documents for display. Overall the interface provides access to almost all of the functionality available from the INQUERY retrieval engine (a view of the interface can be seen in figure 3.1). The example view shows a query for "ACFT" which is a common abbreviation for "aircraft" used in the ASRS reports. In this example xinquery returns all documents containing "ACFT" ranked in order of number of occurrances.

3.1.4 Structured Query Language

Queries may be of two types, natural language or structured. Using the natural language queries the user can provide an English sentence to represent the information request. The query is then transformed into a structured form by the Inquery query-processor so that it can then be processed by the query engine. The user however, can also provide a precise definition of the term relationships in the query by putting the query in structured form directly, which may provide improved performance as a result. The performance of an Information Retrieval system is measured by:

- Recall - The number of relevant items retrieved / Number of relevant items in database.
- Precision - The number of relevant items retrieved / Number of items retrieved.

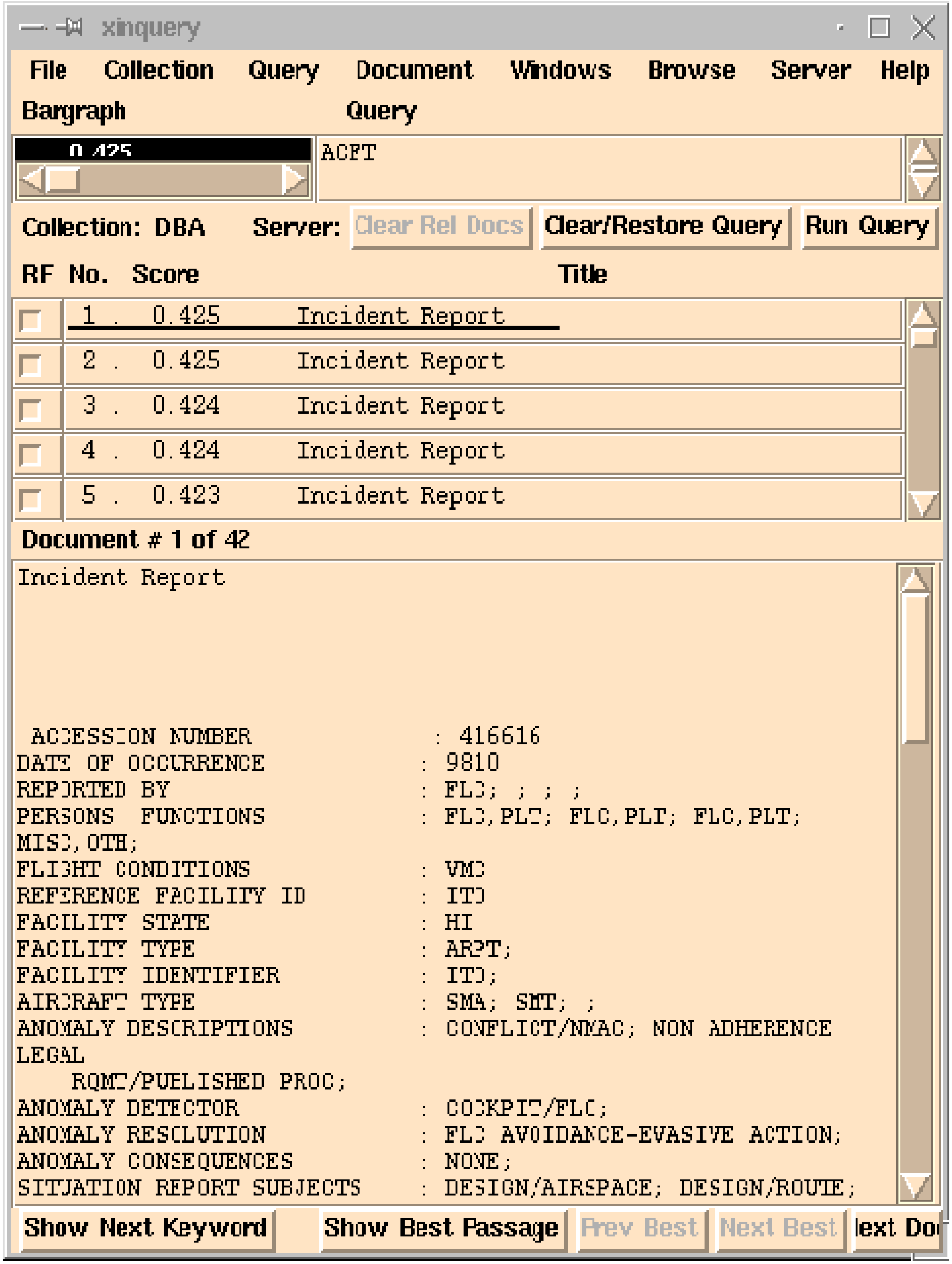

Figure 3.1: Sample Xinquery Screenshot

These can only be calculated with help from a domain expert who can assess relevance. They are nonetheless used as a key means of performance. The desired performance of each of these two factors is driven by the type of search required for example, if the user wishes to find everything, not caring about how many irrelevant documents are returned then recall is important. Alternatively if a quick search for the best items is desired then the precision is more important. As queries become more general we expect the precision to drop as the recall increases, and vice versa as query formulation becomes narrow and specific. Purely in terms of time and data, Inquery can read in parse, and index 400 MB of text per hour (see Inquery technical summary document). Writing structured queries requires some knowledge of the operators provided if the user wishes to formulate the query properly. A description of these operators from the Inquery help files can be found in Appendix B.

In addition examples of these structured queries can be found in section 3.3.2 Testing the General Databases and section 3.5.2 Testing the Fields Databases.

3.2 Creating an Inquery Database

3.2.1 Introduction

Before a collection of documents can be queried to retrieve specific matches the Inquery system requires that the collection must first be indexed to form an Inquery database. The documents desired to be in the database must be in one of four formats: (Each of which is explained in the following section 3.2.2 Document Format)

1. SGML
2. HTML
3. email
4. cacm_dots

When the data has been changed into one of these document formats the collection can be used to create an Inquery database in one step using the 'inbuild' parsing and indexing program (which will be discussed in section 3.2.3 inbuild). However if storage required to index exceeds avaliable disk space the database can be created using incremental steps, by using the 'inparse' along with the 'merge_btl' programs.

3.2.2 Document Format

As stated the Inquery parsing-indexing programs, (inbuild, inparse and merge_btl), recognise four document formats. These formats are SGML, HTML, email and cacm_dots. Each of the document formats implement the same text tokenisation definitions, which are used to determine the types of 'concepts' that are noticed in the text. These 'concepts' are normally just single words but they can also be made up by more than one word. The following is a description of each of the formats supported:

- The SGML (Standard Generalised Markup Language) document format, although it is only a subset of the 'real' SGML format, can recognise a diverse number of different field tag types. SGML is fast becoming common as a document format but the format required by Inquery does not have to meet any of the official SGML standards.

- HTML (HyperText Markup Language) format is in actuality a subset of SGML found commonly in network browser documents.

- The email document format differs slightly from the appearance of a few mail interfaces but essentially recognises the message text from mail as it is displayed in a systems mailbox e.g.

  From aaa@aaa Mon Jan 01 00:00:00 1997

  Return-path: $< gilmouew@dcs.gla.ac.uk >$

  Envelope-to: mcelropd@dcs.gla.ac.uk

Delivery-date: Tue, 07 Mar 2000 13:33:43 +0000

Received: from bo620-22u.dcs.gla.ac.uk ([130.209.245.232] helo=bo620-22u ident=gilmouew) by iona.dcs.gla.ac.uk with esmtp (Exim 3.13 #1) id 12SK7L-00070B-00 for mcelropd@dcs.gla.ac.uk; Tue, 07 Mar 2000 13:33:43 +0000

Date: Tue, 7 Mar 2000 13:33:43 +0000 (GMT)

From: Ewen W Gilmour < *gilmouew@dcs.gla.ac.uk* >

To: Peter McElroy < *mcelropd@dcs.gla.ac.uk* >

Subject: aea

Message-ID: < *Pine.LNX*.4.10.10003071333180.19589 − 100000@*bo*620 − 22*u.dcs.gla.ac.uk* >

MIME-Version: 1.0

Content-Type: TEXT/PLAIN; charset=US-ASCII

Status: RO

X-Status: R

X-Keywords:

X-UID: 363

hi

this is the email text.

- Lastly, cacm_dots format is used in information retrieval research by standard evaluation collections. These collections include CACM (Communications of the Association for Computing Machinery) and NPL (National Physics Laboratory). There are only a few fields that are recognised by this format and they take the form of a tag made up of a single dot '.' and subsequently a single capital letter at the beginning of a line.

For the use of incident reports in inquery we would have to convert the reports into one of the above formats. With the cacm_dots, SGML and the email formats being too specific for their relevant purposes, changing the reports into HTML format would clearly be the option to choose. With the addition of a simple

HTML structure and the separation of each incident report into an individual file we can create the sort of database we need by using the inbuild function.

3.2.3 inbuild

The function of the inbuild command is to create an Inquery database. The collections and database_name arguments are required, also the -type argument which specifies the document format of the collection. The remaining optional arguments may occur in any order. A description of the relevant arguments we will use taken from the Unix man page follows:

- database_name - name of the database to be created.
- collection - pathname to collection file.
- -fields - create field transactions for field retrieval.
- -type format_type - All files following will be parsed by the specified parser. More than one -type flag may be used. ‘sgml’ is the default.

3.3 The General Databases

3.3.1 Creating the General Databases

To begin with a test database was produced using a selection of incident reports from the ASRS data. This was done to get a feel for the inquery database building procedure and to notice any mistakes which may be made during the stages of construction. Pinpointing these errors was effective at minimising the time taken when dealing with creating databases using a full set of incident data. Each incident was converted into html format and saved in separate files all in the one directory. Then the collection was converted into an Inquery database using the inbuild command, specifying the type as ‘html’. The results were then viewed

using the xinquery database. Two things were noticed at this stage, the construction of the database did not require each individual entry in the incident report to be specified as a paragraph by using the 'p' tag in html, the 'body' tag was sufficient to specify this. Also Xinquery queries would not work unless the Query mode was set to 'passage' rather than 'document' this is due to the fact that on the current platform some essential tcl libraries are missing.
Once the test database had been built, the two general databases (one using the Rotary vehicle incidents and the other using the Air Traffic Controller incidents) were produced and tested using simple queries in Xinquery.

3.3.2 Testing the General Databases

To test the general databases a series of natural language queries and simple structured queries were used. The results were then viewed to ensure the documents retrieved were valid. The following is a walk through example of one of the tests carried out. A query was entered into the air traffic control fields database:

#phrase(CTLR INTERVENED)

In plain English we are looking for incidents that involved air traffic controller intervention (CTLR being the standard abbreviation for controller in the ASRS datasets). After running the query, all the incidents in the database are displayed as possible matches (due to the fact that all documents contain the text "CTL-R"). However, the first fifteen documents are more relevant than the rest (as they contain the phrase "CTLR INTERVENED") and therefore have a higher ranking. The fact that all documents were returned when not all are relevent indicates a lack of precision within the database. Each document however, was viewed to check that the incidents returned were accurate (An example of one of the incidents returned and viewed with Xinquery can be seen in figure 3.2).

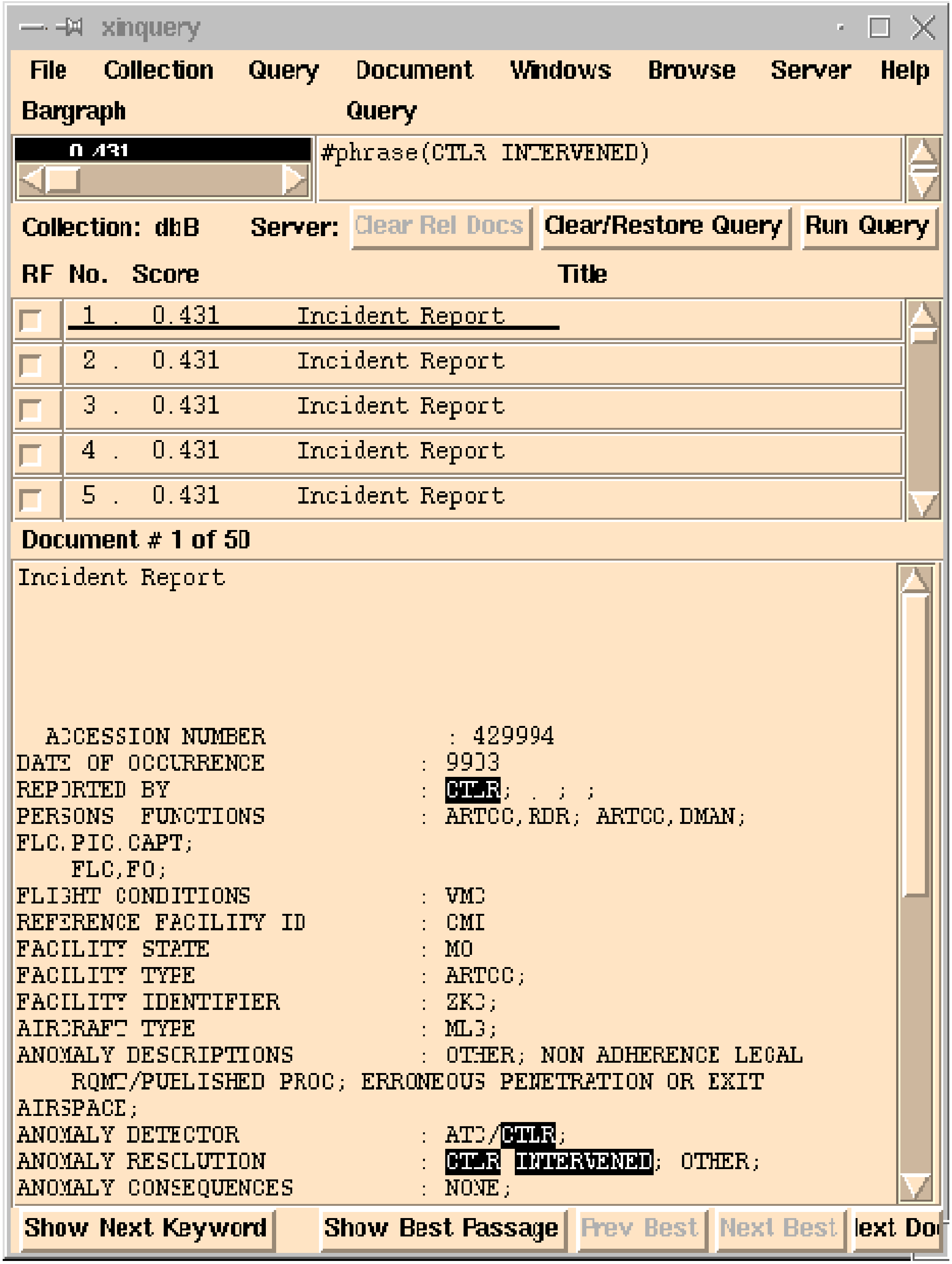

Figure 3.2: Example Incident

3.4 Modifying Inquery

3.4.1 Overview

After testing the general databases produced, it became clear that a greater level of precision was needed. After investigation of the inquery documentation it became apparant that the best way to accomplish this would be to create a database containing fields. With this method specific entries in the ASRS datasets could be mapped in a more structured way into a database which would in turn improve precision. The modification of inquery source code, to allow a document type to specify fields can be done in one of two ways:

- Create a new document translator and add a new document type entry into the dm_trans_tab.c file.
- Modify an existing document translator and its corresponding entry in the dm_trans_tab.c file.

After investigation into both methods, it was decided that modifying an existing translator would be the best approach. This was essentially due to the fact that creating a new document translator would not be feasible in terms of time and also to a lesser extent due to a lack of experience using the lexical language Flex. In contrast modifying an existing document translator would be quicker. The existing translator chosen was the HTML translator. This was due to the fact that the incident reports had already been changed previously into HTML format. The HTML translator was the least complex of the other three avaliable, i.e. SGML, cacm_dots and email.

3.4.2 The dm_trans_tab.c File

Each new field in the document translator function must be entered in the Document Translator Table entry for the HTML translator found at the end of the dm_trans_tab.c file. An entry in the table for our purposes could be the following:

"ReportedBy" "Reported By", ON, INQ_NORMAL, INQ_TEXT, ON, INQ_COOP,

Where each field represents:

FIELD-NAME, DESCRIPTION, NORM-INDEX, FIELD-INDEX, DISPLAY, RUN-RECOGS,
COOP MODES

In addition the number of fields count, for the document type, should be modified to include the new fields i.e. where each entry is of the form:

DOCFORMAT-NAME, TRANSLATOR-FUNCT, NUM-FIELDS, FIELD-TABLE

we change the html entry to:

"html", html_trans, 12, html_fields ,

For a listing of the modified dm_trans_tab.c file see Appendix C.

3.4.3 Modifying The Document Translator

To add a new field to the existing document translator function for HTML, we edit the appropriate flex translator file by following the format. We must add an entry for the beginning tag for the field i.e. < *ReportedBy* > and the end tag, < /*ReportedBy* > as follows.

Note: Field names cannot contain underscores (_).

< *ReportedBy* >
{/* start tag for ReportedBy field */

```
BEGIN TXT;
skip_text(yytext, yyleng);
status = begin_field(“ReportedBy”, yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}

</ReportedBy>
{ /* end tag for ReportedBy field */
BEGIN 0;
status = finish_field(“ReportedBy”, yytext);
if(status != INQ_SUCCESS)
return status;
skip_text(yytext, yyleng);
}
```

For a complete list of the modified document translator see Appendix D.

3.5 The Fields Databases

3.5.1 Creating the Fields Databases

As with the previous general databases a test database was produced for the same reasons. This helped pinpoint minor errors in the document translator where the syntax did not match that of the modified incident reports and helped minimise the time taken dealing with errors when creating the fields databases using a full set of incident data. To create the fields databases each incident was taken from the previously converted html format and field tags were added as specified in the document translator to the appropriate parts of each incident. Then the collection was converted into an Inquery database using the inbuild command, specifying the type as 'html' and using the '-fields' tag. The results were then viewed using Inquery and a set of test query's which will be discussed in section 3.5.2 Testing the Fields Databases. The list of fields chosen from the ASRS data set are as follows:

- REPORTED BY
- PERSONS FUNCTIONS
- FLIGHT CONDITIONS
- AIRCRAFT TYPE
- ANOMALY DESCRIPTIONS
- ANOMALY DETECTOR
- ANOMALY RESOLUTION
- ANOMALY CONSEQUENCES

3.5.2 Testing the Fields Databases

To test the fields databases a series of specific field queries were used and the results viewed to ensure the documents retrieved were valid. The following is a walk through example of one of the tests carried out:

The following query was entered into the rotary vehicle fields database:

#band(#field(AircraftType SMT) #field(FlightConditions VMC))

In plain English we are looking for all incidents involving a small transport in which the flight conditions were VMC. After running the query fifteen incidents in which this was the case were returned (see figure 3.3). Each document returned contains a ranking followed by its score and its number in the database. For example from figure 3.3 the document ranked 1st has a score of 0.200 and it is incident

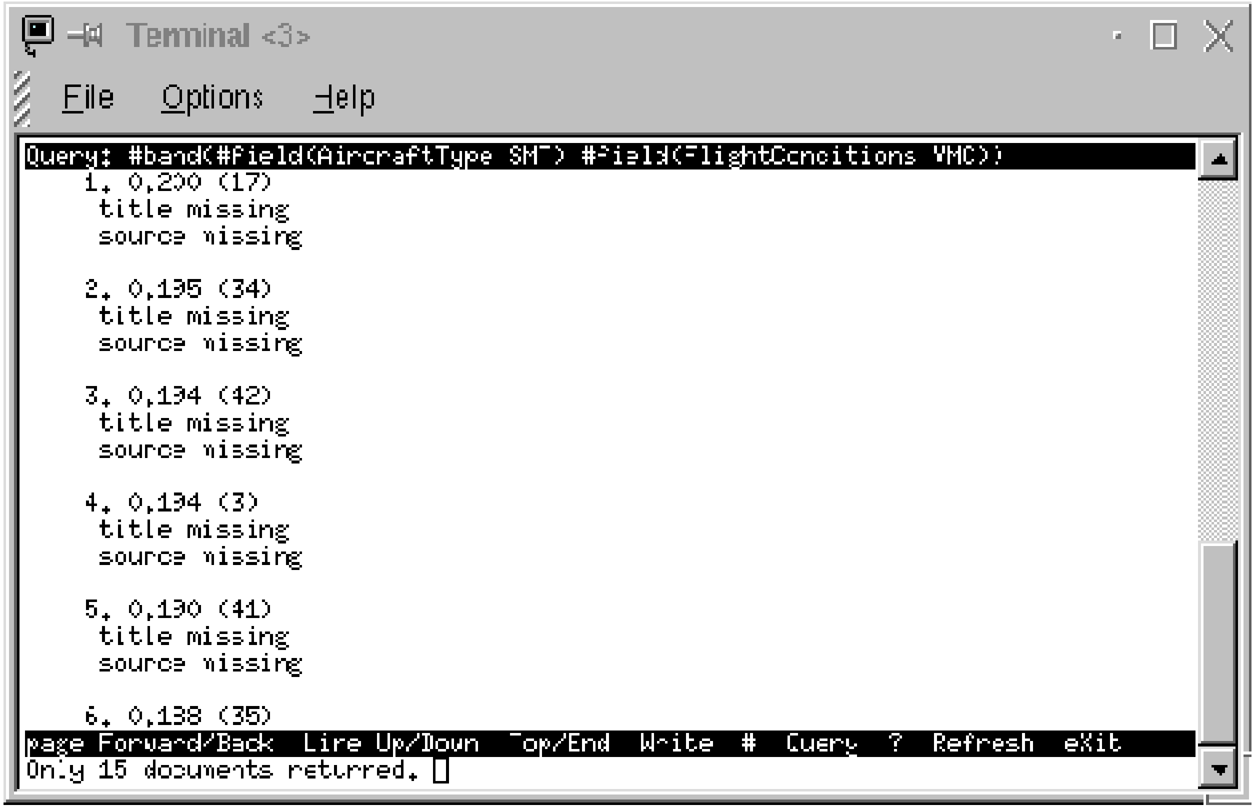

Figure 3.3: Returned Incidents

number 17 in the database. Also included is a note that the documents title and source are missing which have no bearing on our tests. Each document was viewed to check that the incidents returned were accurate, and the overall data set of incidents was checked to make sure that no relevant documents had been left out (An example of one of the incidents returned and viewed with inquery can be seen in figure 3.4). The example shown in figure 3.4 shows the document ranked 5th with a score of 0.190 and contains the incident report in full. Again references to the missing id and title have no bearing on our tests.

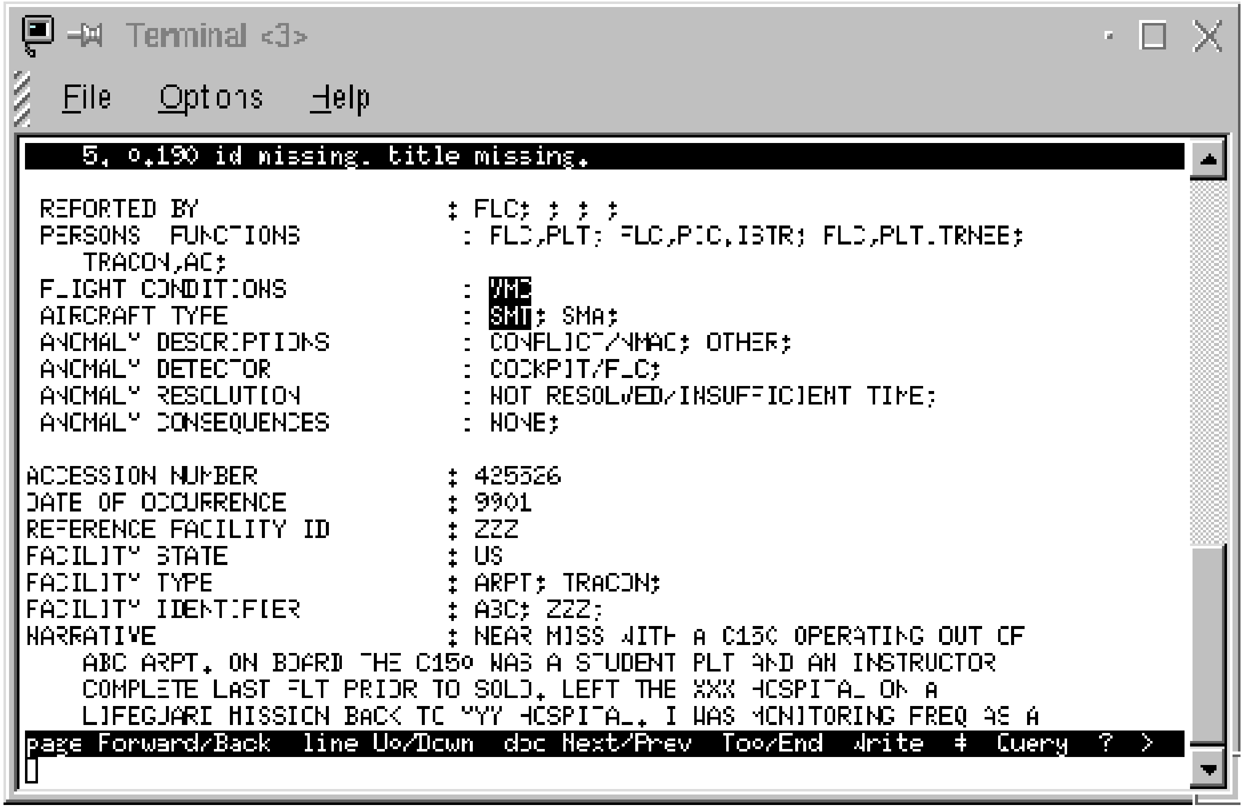

Figure 3.4: Single Incident

3.6 Creating The HTML Front End

3.6.1 Creating the Web Pages

In the interest of providing a more accessible interface to the inquery fields database it was decided that a web based interface should be created. This was designed around a main query page, a help page which tells the user how to formulate standard inquery queries, a page decribing the fields and their possible values (as a help for the user) and also a questionnaire page which hopefully should give reasonable feedback on the database (see figure 3.5 and figure 3.6 for two sample screenshots). The created web site can be found at (see Appendix E for a full listing of HTML source code):

- http://www.dcs.gla.ac.uk/ mcelropd/inquery/

The pages use standard cgi scripting to process site evaluation forms.

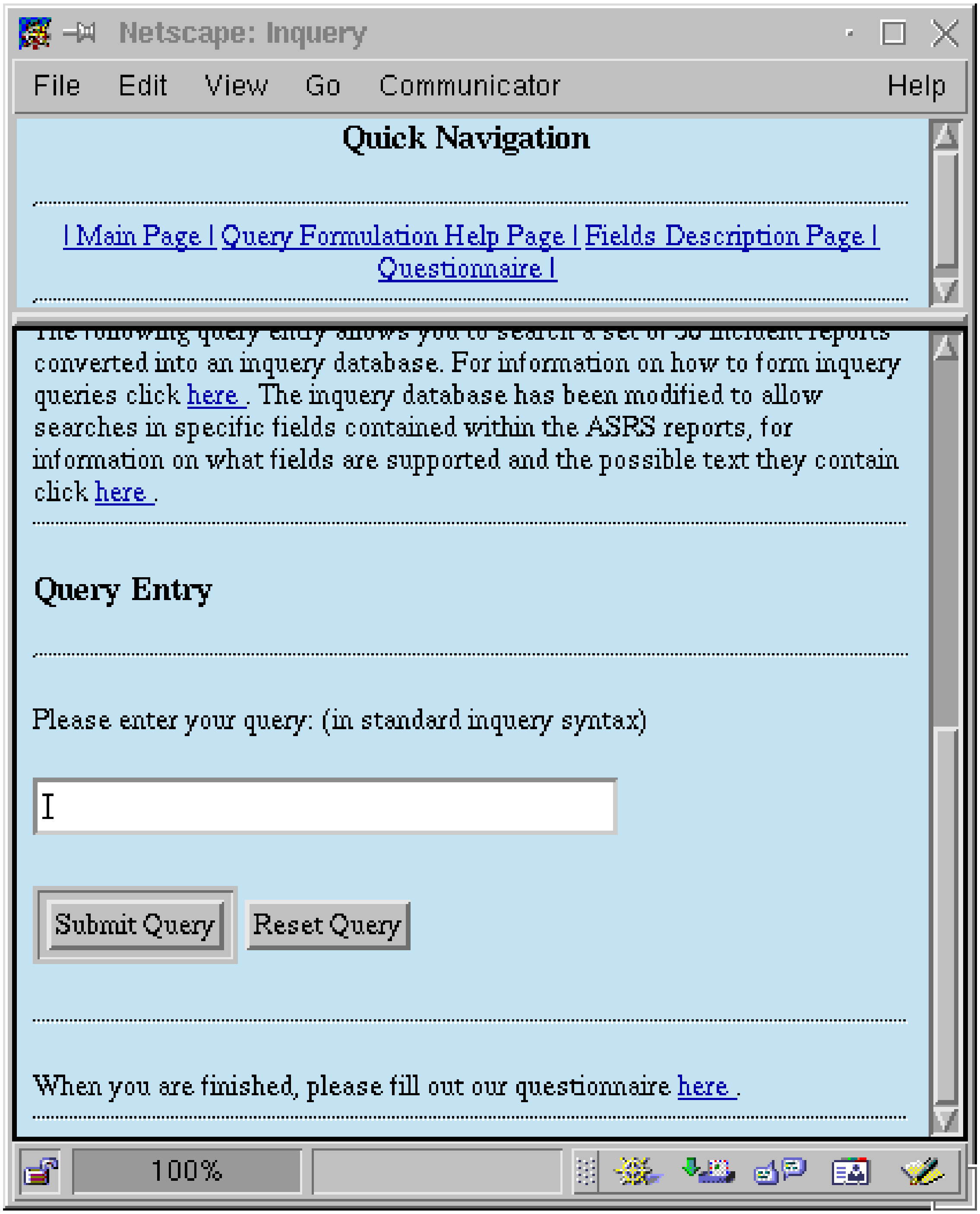

Figure 3.5: Main Query Page

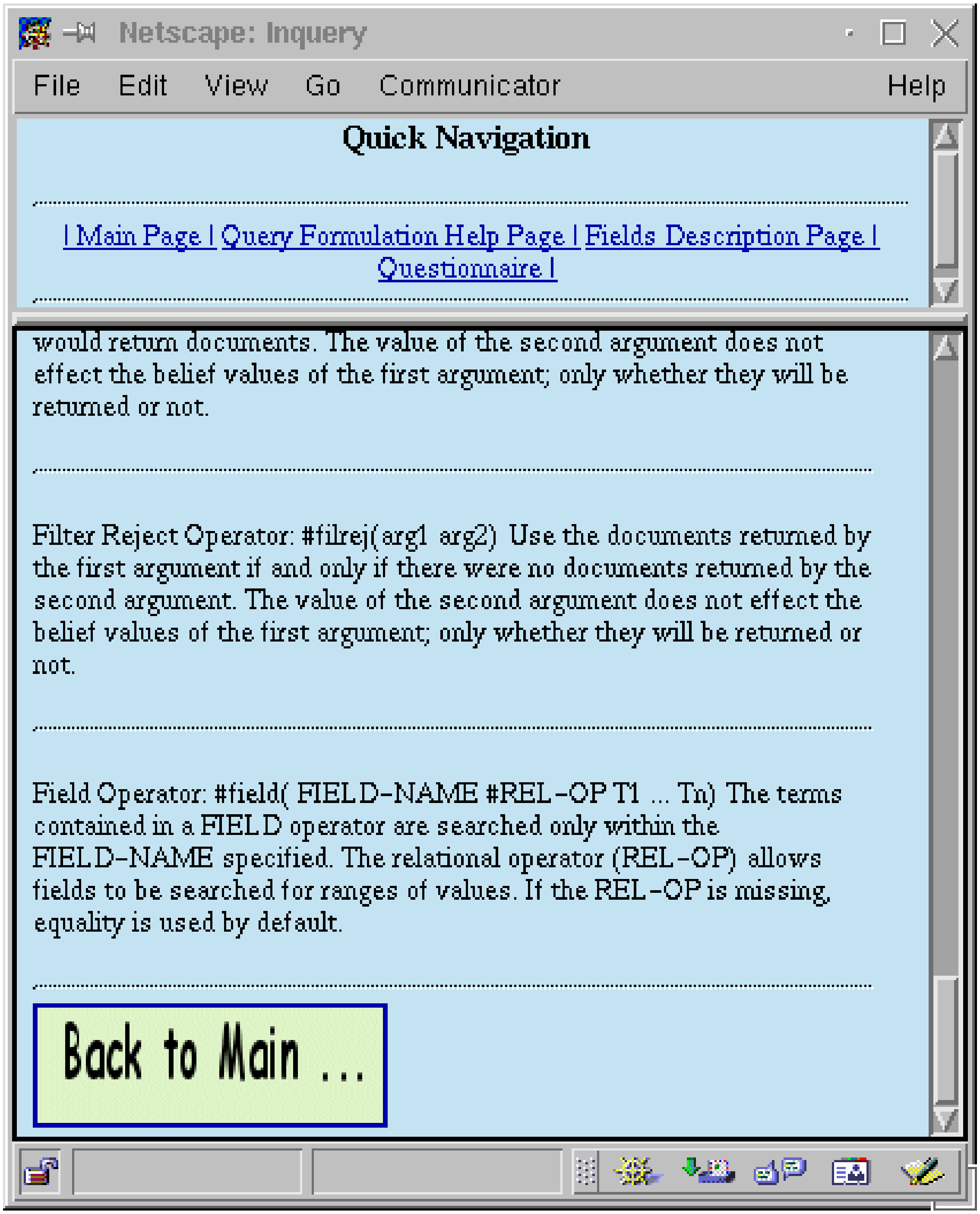

Figure 3.6: HTML Screenshot

3.6.2 Questionnaire CGI Script - using Perl

The questionnaire cgi script was written in Perl due to the languages ease of text manipulation. A Perl program is made up of an ordinary text file containing a series of Perl statements. These statements are written in what looks like an amalgam of English, UNIX shell script, and C. Essentially the code takes the input variables from the "POSTED" FORM and splits them up into an array. Then, not forgetting the user, HTML is printed directly to the browser to give some useful feedback (see Appendix F).

3.6.3 Inquery client-server connections

To use the web browser interface described above we must first have the inquery-server program running on a host and port. This establishes an inquery database server on the specified machine, with the specified database. Then, whenever a query is passed to the relevent cgi script, a call would be made to the server which performs the query. The results are then returned in the form of a document tag which points to the retrieved documents, these documents are then sent to the relevant HTML page using a simple 'print' command. The source code and cgi script (both written in C) to accomplish this are provided in full in the "Inquery System Administrators Guide" unfortunately due to network access constraints and also a bug in the version of the inquery server being used this was only partially accomplished during the final stages of this project.

3.7 Validation Of The Inquery Databases

3.7.1 Precision and Recall of General Database

To test the recall and precision of the general databases the following query was carried out on the Air-Traffic Controller's database:

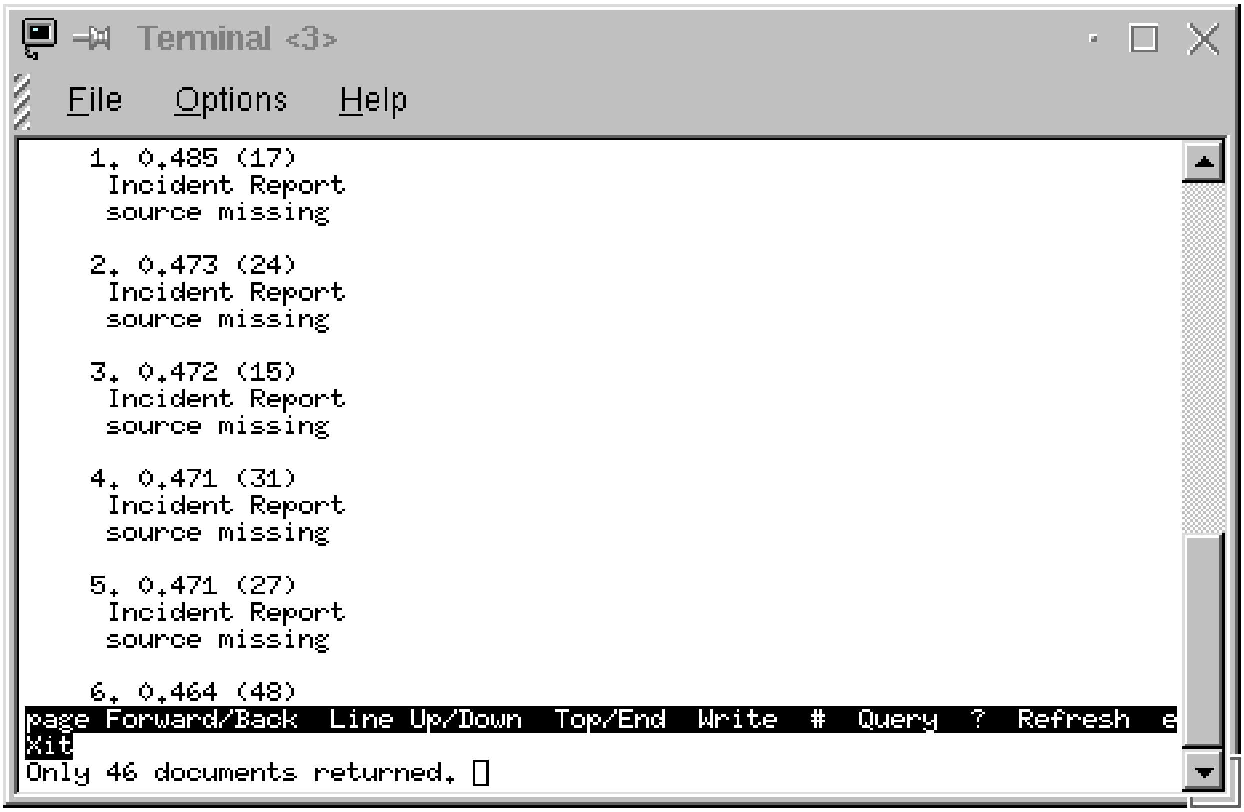

Figure 3.7: Documents Returned

#phrase(ACFT LOST SEPARATION)

Nine incidents from the database had been earmarked previously as being relevant to this query. This was done by myself although ideally an expert would have selected those incidents relevant. We are looking for incidents in which two aircraft lost their standard separation distance. After running the query 46 documents were returned (see figure 3.7). After viewing the results it was clear that the query had returned all 9 documents earmarked as relevant (see figure 3.8).

Using the formulas (stated previously) for recall and precision we find that:

- Recall - The number of relevant items retrieved / Number of relevant items in database.
- Therefore Recall = $9/9 = 1$

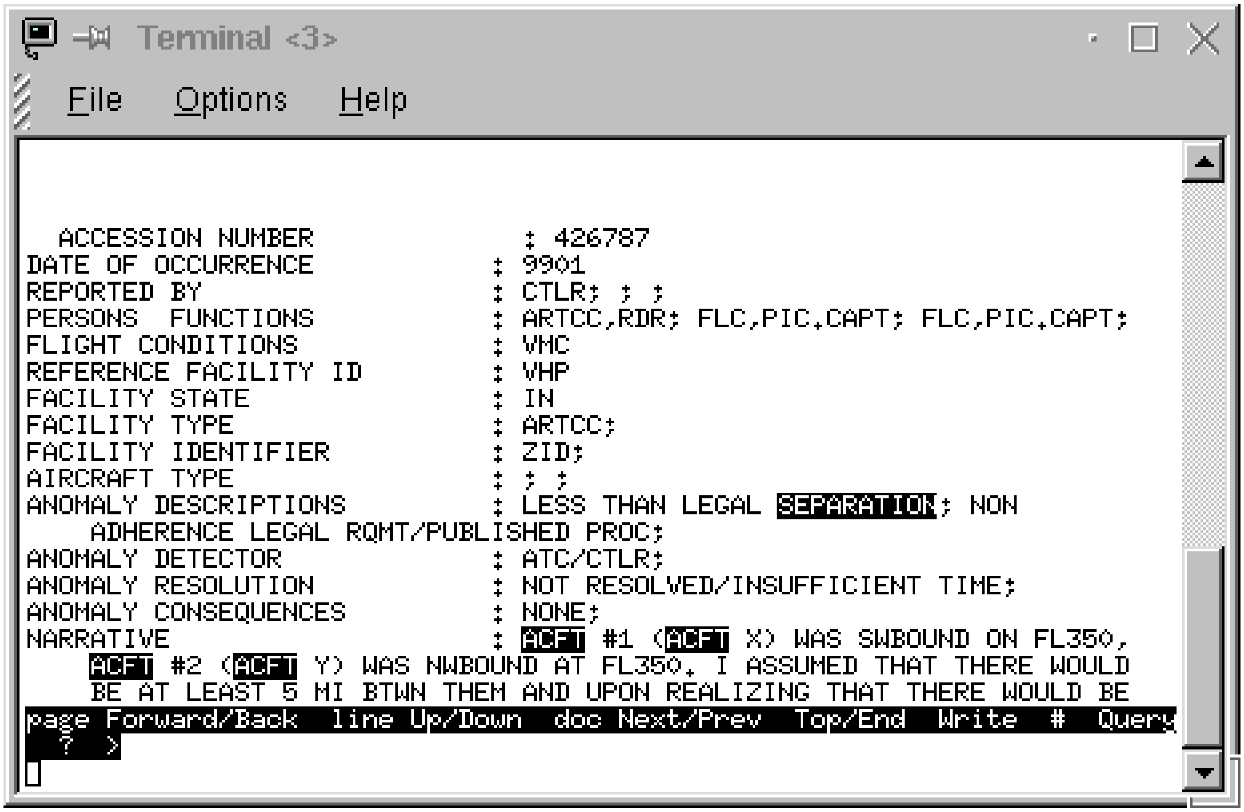

Figure 3.8: Sample Incident

- Precision - The number of relevant items retrieved / Number of items retrieved.
- Therefore Precision = 12/46 = 0.26087 (to 5 decimal places)

The general database has scored perfect for recall, and has also scored poorly for precision as expected. This is due to the lack of structure of the database and also the inability to perform exact queries as is possible with field transactions.

3.7.2 Precision and Recall of the Fields Databases

To test the recall and precision of the fields databases the following query was carried out on the Rotary Vehicles database:

#band (#field(AnomalyDesc NMAC) #field(AnomalyDetector FLC))

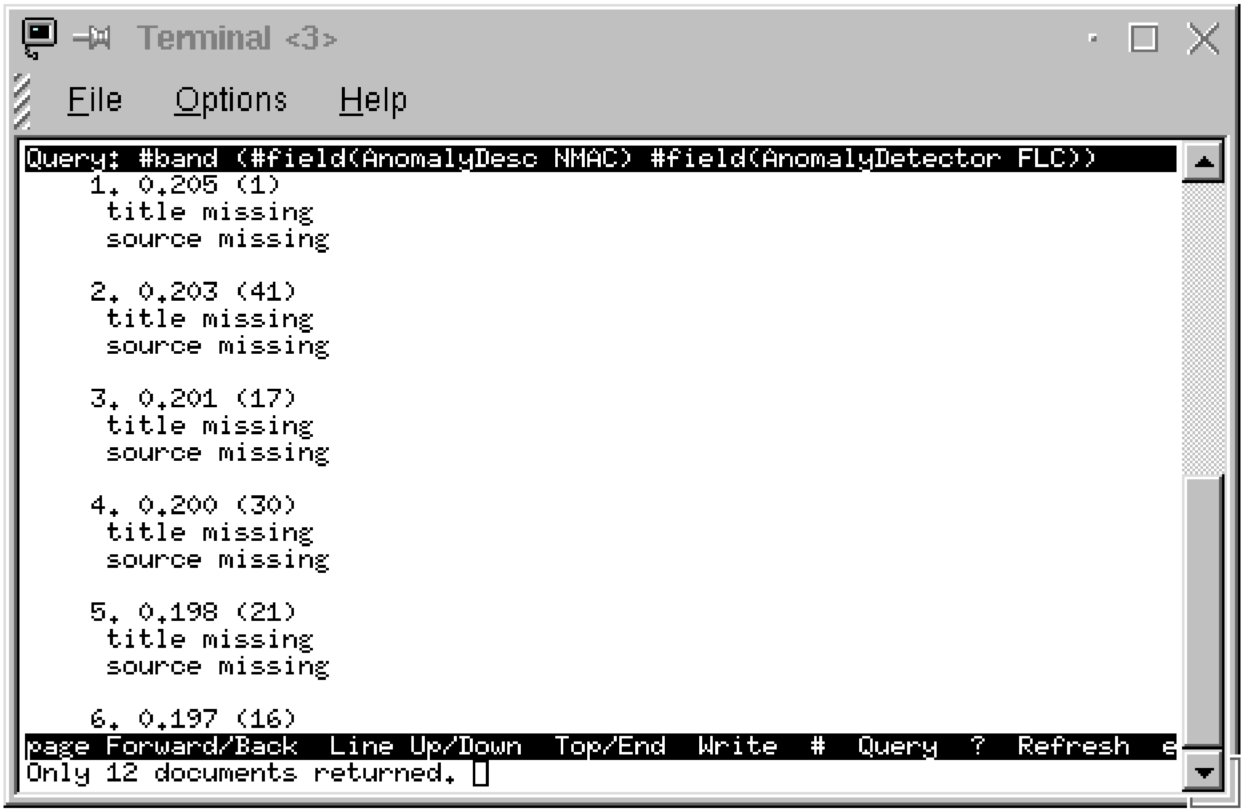

Figure 3.9: Documents Returned

Twelve incidents from the database had been earmarked previously as being relevant to this query (as before by myself). We are looking for all near mid-air collisions in which the pilot discovered there was a problem. After running the query 12 documents were returned (see figure 3.9). After checking each document it was clear that each of the 12 documents were the ones previously earmarked as relevant (see figure 3.10 for an example of a returned document).

Using the formulas (stated previously) for recall and precision we find that:

- Recall - The number of relevant items retrieved / Number of relevant items in database.
- Therefore Recall = 12/12 = 1
- Precision - The number of relevant items retrieved / Number of items retrieved.
- Therefore Precision = 12/12 = 1

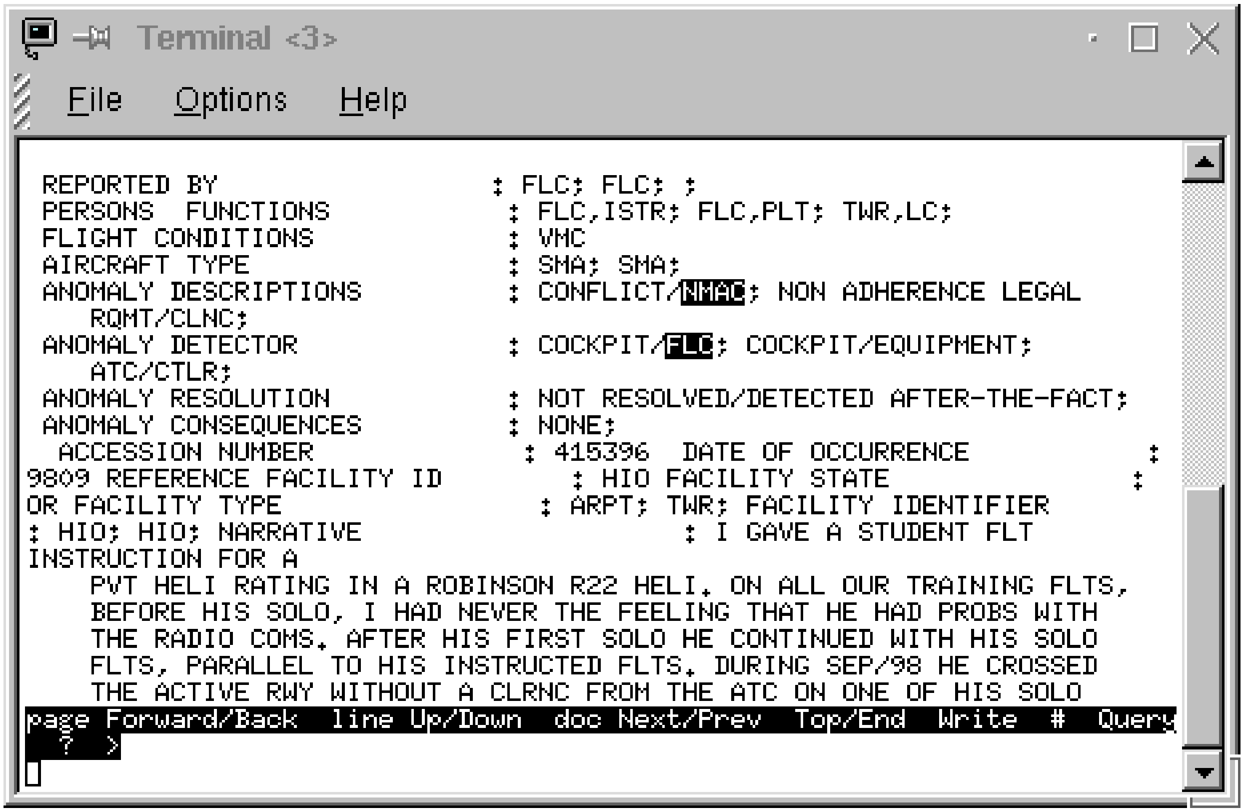

Figure 3.10: Sample Incident

The fields database has scored perfect for precision as expected but has also scored perfect for recall. This is because the database has a limited number documents (only 51 entries) and also, the structure of the incident reports is very precise. Therefore we should expect a high recall score. With the same type of database containing a much larger number of entries these scores would be no where near as high.

3.7.3 User Evaluation

In addition to the precision recall examples some user evaluations were also carried out on the fields databases and a questionnaire (see Appendix H) distributed at the end. The evaluations were run along the lines of an informal Think Aloud with the users comments on the system noted down. The recurring theme throughtout the evaluations was that the database was powerful when looking for specifics but the inability to match like terms made general purpose retrieval poor. Therefore the best improvement would be to provide the ability to match like terms in the database to the query, for example if the query contained the word "smoke" we would also wish to match words such as "fumes". Further it should be noted that the abbreviated terms used by the ASRS and pilots caused some confusion with the test subjects even when provided with a glossary covering the most important terms. To combat this I believe removing the abbreviations and replacing them with the correct words is the only solution as the glossary generally proved ineffectual.

3.7.4 Questionnaire Results

The questionnaires provided fielded the following questions, with the following results (see sample questionnaire in Appendix H):

- How useful would you rate the database? Which scored on average 'Useful'.
- In terms of ease of use how would you rate using the database? Which scored 'Not Very Easy' every time.
- How powerful do feel the database retrieval is? Which scored 'Powerful' every time.
- On average, How relevant did you think the documents returned (to your queries) were? Which scored on average 'Relevant'
- How useful did you find the fields transactions? Which scored 'Very Useful' every time.

The comments included in the questionnaires contained comments to the effect that knowledge of the database (i.e. incident report structure) is essential if you wish to formulate efficient queries.

Chapter 4

Conclusions

The analysis of accident and incident reports is an important area of study as such work can be used to prevent future accidents which may result in injury or the loss of human life. The complexity of these reports however, is such that, using software tools appropriately as an aid to the analysis can be very difficult. This is not due in any way to an inability of the user but down to the limitations of the software tools avaliable, mainly current database schemes. Such schemes may be acceptable (see precision and recall scores in section 3.7 Validation of the Inquery Databases) when dealing with the number of reports used in this study (100+), but the reality of the matter is however, that a far greater number of reports are needed for any analysis to be effective. This in turn causes the precision and recall values of the retrieved results to plummet. Creating incident report specific databases using the field technology in Inquery improves matters slightly, but what is really needed is an alternative technique for the retrieval of similar incidents in large-scale incident reporting schemes.

This study would argue that a Case Based environment for incident_accident retrieval shows signs of being the way forward in this field. Whether the data is pre classified in some way (i.e. using Eindhoven Classification) or simply collated into a case library as is, the technique has shown to be very effective in identifying similar incidents. In addition there are clear areas in which further development would improve the existing case based reasoning approach. With the addition of relevence feedback from the user, and the ability of the reasoner to intrinsically learn from such input adapting not only the earmarked case but also those similar (knock-on effect), the performance of the reasoner would be dramatically improved. Moreover case based reasoning systems were developed to provide guidance on solving problems and making decisions. This would indicate that future developments could be geared towards including the outcomes of particular actions and importantly, identifying those actions which did not resolve the anomaly. Thus such actions could be avoided in future.

4.1 Acknowledgements

Thanks are due to D.W. Aha at the US Naval Research Laboratory, Washington DC who provided access to the NaCoDAE system. Also I would like to thank Ian Ruthven for his vital help with the Inquery Information Retrieval system. Thanks are also due to Alan Stevenson, Ewen Gilmour, Colin Paterson for their willingness to act, and their effort, as test subjects for the evaluations of the NaCoDAE case libraries and Inquery databases created.

I would also like to thank T. van der Schaaf for taking the time to discuss and explain to me the Eindhoven Model in its fullness.

Special thanks are due to Chris Johnson for the help and support provided throughout the duration of this study. Special thanks are also due to Daniela Busse and Shelly Jeffcott for their effort as expert test subjects and also their continued support throughout the past 4 months.

4.2 References

Agnar Aamodt, *Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches.* University of Trondheim, College of Arts and Science, Department of Informatics, N7055 Dragvoll, Norway , 1994.

Klaus-Dieter Althoff, *Evaluating Case Based Reasoning Systems.* Centre for Learning Systems and Applications (LSA) Department of Computer Science, University of Kaiserslautern, 67653 Kaiserslautern, Germany, 1995.

David B. Leake, *Combining Rules and Cases to Learn Case Adaptation.* Computer Science Department, Indiana University, Bloomington, IN 47405, 1995.

J.L. Kolodner (editor),*Case Based Learning.* Kluwer Academic Publishers, London, 1993.

Applied Computing Systems Institute of Massachusetts, Inc (ACSIOM), *Inquery System Administration Manual.* University of Massachusetts, Amherst Computer Science Department within the National Center for Intelligent Information Retrieval (CIIR), 1990 - 1996.

L.A. Breslow and D.W. Aha, *NaCoDAE: Navy Conversational Decision Aids Environment.* Navy Centre for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington DC, Technical Report: AIC-97-018, 1997.

T. van der Schaaf, *Near Miss Reporting in the Chemical Process Industry an Overview.* Eindhoven University of Technology, Graduate school of Industrial Engineering and Management Science, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, 1991.

Floor Koornneef and John Kingston-Howlett, *Accident Data in Organisational Learning, or what makes Accident Databases Useful?* Delft University of Technology, Safety Science Group Kanaalweg 2628 EB DELFT, Netherlands, and John Kingston Associates, Oxleasowes Farm, Brockhill Lane, Tardebigge, Worcestershire B60 1LU, UK, 1998.

C.W. Johnson, *Using Case-Based reasoning to Support the Indexing and Retrieval* *of Incident Reports.* Department of Computing Science, University of Glasgow, Glasgow, 1999.

T. van der Schaaf and L. Kanse, *Errors and Error Recovery.* Safety Management Group, Eindhoven University of Technology, The Netherlands, and Human Reliability Associates, HRA-NL:Arenberglaan 280, 4822 ZR Breda, The Netherlands, 1999.

T. van der Schaaf, *PRISMA; A Risk Management Tool Based on Incident Analysis.* Proceedings of the International Workshop on Process Safety Management and Inherently Safer Processes, October 8-11, Orlando, Florida, 242-251, 1996.

W. van Vuuren, *Organisational Failure: An Exploratory Study in the Steel Industry and the Medical domain, PhD thesis.* Technical University of Eindhoven, The Netherlands, 1998.

Appendix A

List of Anomaly Classifications

Aircraft Equipment Problem/Critical - Aircraft equipment problem that is vital to the safety of the flight. HR6 (an inability to operate the equipment). TE (The equipment was not designed properly). TC (the equipment was not constructed properly).

Aircraft Equipment Problem/Less Severe - Not qualifying as a critical aircraft equipment problem. HR6 (an inability to operate the equipment). TE (The equipment was not designed properly). TC (the equipment was not constructed properly).

Altitude Deviation - A departure from or failure to attain or failure to maintain an ATC assigned altitude. It does not include an injudicious or illegal altitude in VFR flight where no altitude has been assigned by ATC or specified in pertinent charts. An aircraft climbs or descends through the assigned altitude. HS1 (Pilot performs the wrong movement). HR4 (Failure to check aircraft altitude).

Altitude Deviation/Undershoot on Climb or Descent - An aircraft fails to reach an assigned altitude during climb or descent. HS1 (Pilot performs the wrong movement). HR4 (Failure to check aircraft altitude). HK1 (Pilot does not know of the assigned altitude).

Alt Deviation/Excursion From Assigned - An aircraft departs from level flight at an assigned altitude. HS1 (Pilot performs the wrong movement). HK1 (Pilot does not know of the assigned altitude). HR6 (a misunderstanding of the instrumentation). TE (The design of the equipment maybe wrong creating usability problems).

Alt Deviation/Xing Restriction Not Met - Charted or assigned altitude crossing restriction not met. HK1 (Unaware of the altitude restriction). HR4 (failure to check the position of aircraft).

Altitude-Hdg Rule Deviation - Cruise flight contrary to the altitudes specified in FAR 91.159. HR6 (failure to understand equipment). HR2 (failure to get the appropriate permission if avaliable)

Conflict/NMAC (near midair collision) - A conflict is defined as the existence of a perceived separation anomaly such that the pilot(s) of one or both aircraft take evasive action; or are advised by ATC to take evasive action; or experience doubt about assurance of continuing separation from the viewpoint of one or more of the pilots or controllers involved. A near midair collision is when the flight crew reports, either directly or as quoted by the controller, that the reported miss distance is less than 500 feet. OP (the procedure in question was wrong). HK2 (no understanding of the goal or priorities). HR5 (inadequate planning of either of the involved parties).

Conflict/Airborne Less Severe - A conflict not qualifying as an NMAC. OP (the procedure in question was wrong). HK2 (no understanding of the goal or priorities).HR5 (inadequate planning of either of the involved parties).

Conflict/Ground Critical - A ground occurrence that involves (1) two or more aircraft, at least one of which is on the ground at the time of the occurrence or (2) one or more aircraft conflicting with a ground vehicle. The flight crew reports, either directly or as quoted by a controller, that they took evasive action to avoid a collision (emergency action go around, veering on runway or taxiway, take off abort, or emergency braking), and the balance of the report, including the narrative is judged consistent with a critical occurrence. OP (the procedure in question was wrong). HK2 (no understanding of the goal or priorities).HR5 (inadequate planning of either of the involved parties).

Conflict/Ground Less Severe - A ground conflict not qualifying as critical. OP (the procedure in question was wrong). HK2 (no understanding of the goal or priorities).HR5 (inadequate planning of either of the involved parties).

Controlled Flight Toward Terrain - Flying at an altitude that would, if continued, result in contact with terrain. HK2 (no understanding of the goal or priorities).HR5 (inadequate planning of either of the involved parties).

Erroneous Penetration or Exit Airspace - Self-explanatory. OP (failure of the operating procedures). HS1 (an uncontrolled movement by the pilot). HR3 (no coordination between tower, aircraft or flight crew)

In-Flight Encounter/Other - In-flight encounter (e.g., bird strikes, weather balloons). X (Un classifiable anomaly).

In-Flight Encounter/WX - In-flight encounter with weather (e.g., windshear, turbulence, clouds, high winds, storms). X (Un classifiable anomaly).

Less Than Legal Separation - Less than standard separation between two airborne aircraft (as standard separation is defined for the airspace involved). HR5 (misunderstanding of equipment or information provided). HK1 (operator does not know the status of the system).

Loss of Aircraft Control - Self-explanatory. TM (defect in aircraft materials). TE (aircraft design is wrong). TC (construction of the aircraft was not carried out accurate to the design). HS1 (pilot cannot perform the necessary controlled movements).

Non Adherence Legal Requirement/Clearance - Non-adherence to an ATC clearance. HR5 (lack of planning causing the offence). HK1 (misunderstanding of the legal requirement)

Non Adherence Legal Requirement/FAR - Non-adherence to a Federal Aviation Regulation. HR5 (lack of planning causing the offence). HK1 (misunderstanding of the legal requirement)

Non Adherence Legal Requirement/Published Procedure - Non-adherence to approach procedure, standard instrument departure, STAR, profile descent, or operational procedure as described in the AIM or ATC facility handbook. HR5 (lack of planning causing the offence). HK1 (misunderstanding of the legal requirement)

Non Adherence Legal Requirement/Other - Non-adherence to SOPs for aircraft, company SOPs, etc. HR5 (lack of planning causing the offence). HK1 (misunderstanding of the legal requirement)

Runway or Taxiway Excursion - An aircraft exits the runway or taxiway pavement. HS1 (pilot is unable to perform the controlled action). TE (the aircraft design was wrong). TC (construction deviated from the design)

Runway Transgress/Other - The erroneous or improper occupation of a runway or its immediate environs by an aircraft or other vehicle so as to pose a potential collision hazard to other aircraft using the runway, even if no such aircraft were actually present. HS1 (pilot is unable to perform the controlled action). TE (the aircraft design was wrong). TC (construction deviated from the design)

Runway Transgress/Unauthorised Landing - A runway transgression specifically involving landing without a landing clearance or landing on the wrong runway. HS1 (pilot is unable to perform the controlled action). TE (the aircraft design was wrong). TC (construction deviated from the design)

Speed Deviation - Aircraft speed contrary to FARs or controller instruction. HS1 (pilot is unable to perform the controlled action). TE (the aircraft design was wrong). TC (construction deviated from the design)

Track or Hdg Deviation - self explanatory HS1 (pilot is unable to perform the controlled action). TE (the aircraft design was wrong). TC (construction deviated from the design)

Uncontrolled Airport Traffic Pattern Deviation - Failure to fly the prescribed rectangular pattern or failure to enter on a 45 degree angle to the downwind leg. HR6 (misunderstanding of equipment or all of the information avaliable). HK2 (misunderstanding of the overall goal).

VFR in IMC - Flight into IMC when not on an instrument flight plan and/or when not qualified. HR6 (misunderstanding of equipment or all of the information avaliable). HK2 (misunderstanding of the overall goal).

Appendix B

Inquery Structured Queries

Sum Operator: #sum (T1 ...Tn)

The terms or nodes contained in the sum operator are treated as having equal influence on the final result. The belief values provided by the arguments of the sum are averaged to produce the belief value of the #sum node.

Weighted Sum Operator: #wsum (Ws W1 T1 ... Wn Tn)

The terms or nodes contained in the wsum operator contribute unequally to the final result according to the weight associated with each (Wx). The final belief value is scaled by Ws, the weight associated with the #wsum itself.

Ordered Distance Operator: #N (T1 ... Tn) or #odN (T1 ... Tn)

The terms within an ODN operator must be found within N words of each other in the text in order to contribute to the document's belief value. The '#N' version is an abbreviation of #ODN, thus #3(health care) is equivalent to #od3(health care).

And Operator: #and(T1 ... Tn)

The more terms contained in the AND operator which are found in a document, the higher the belief value of that document.

Boolean And Operator: #band(T1 ... Tn)

All of the terms within a BAND operator must be found in a document in order for this operator to contribute to the belief value of that document.

Boolean And Not Operator: #bandnot (T N)
Search for document matching the first argument but not the second.

Or Operator: #or(T1 ... Tn)
One of terms within the OR operator must be found in a document for that document to get credit for this operator.

Negation Operator: #not(T1 ... Tn)
The terms or nodes contained in this operator are negated so that documents which do not contain them are rewarded.

Un-ordered Window Operator: #uwN(T1 ... Tn)
The terms contained in a UWN operator must be found in any order within a window of N words in order for this operator to contribute to the belief value of the document.

Phrase Operator: #phrase(T1 ... Tn)
Terms within this operator are evaluated to determine if they occur together frequently in the collection. If they do, the operator is treated as an ordered distance operator of 3 (#od3). If the arguments are not found to co-occur in the database, the phrase operator is turned into a SUM operator. In ambiguous cases the phrase becomes the MAX of the SUM and the OD3 operators.

Passage Operator: #passageN(T1 ... Tn)
The passage operator looks for the terms or nodes within the operator to be found in a passage window of N words. The document is rated based upon the score of it's best passage.

Synonym Operator: #syn(T1 ... Tn)

The terms of the operator are treated as instances of the same term.

Maximum Operator: #max(T1 ... Tn)

The maximum belief value of all the terms or nodes contained in the MAX operator is taken to be the belief value of this operator.

Weight Plus Operator: #+ T1

The effect of the following term or node is increased relative to the rest of the query.

Weight Minus Operator: #- T1

The effect of the following term or node is decreased relative to the rest of the query.

Literal Operator: #lit(T1 ... Tn)

This operator preserves the original forms of the terms contained within it. No stemming or stopping is performed and capitalisation is preserved.

Filter Require Operator: $filreq(arg1 arg2)

Use the documents returned (belief list) of the first argument if and only if the second argument would return documents. The value of the second argument does not effect the belief values of the first argument; only whether they will be returned or not.

Filter Reject Operator: #filrej(arg1 arg2)

Use the documents returned by the first argument if and only if there were no documents returned by the second argument. The value of the second argument does not effect the belief values of the first argument; only whether they will be returned or not.

Field Operator: #field(FIELD-NAME #REL-OP T1 ... Tn)

The terms contained in a FIELD operator are searched only within the FIELD-

NAME specified. The relational operator (REL-OP) allows fields to be searched for ranges of values. If the REL-OP is missing, equality is used by default. This operator is discussed in a section below.

Appendix C

Modified dm_trans_tab.c

/***
Copyright (c) *

Applied Computing Systems Institute of Massachusetts, Inc. (ACSIOM) *

1990 - 1996. *

*

All Rights Reserved. *

*

The INQUERY SOFTWARE was developed at the University of Massachusetts *

at Amherst Computer Science Department within the National Center for *

Intelligent Information Retrieval (CIIR). For more information, contact *

413-545-0463 or http://ciir.cs.umass.edu. *

*

This SOFTWARE was developed in significant part with private funding. *

The INQUERY SOFTWARE is proprietary to ACSIOM and may not be distributed *

in whole or in part or used in any way without authorized permission in *

writing from ACSIOM. Possession, use, duplication or dissemination of *

the INQUERY SOFTWARE and media is authorized only pursuant to a valid *

written license from ACSIOM. *

*

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the U.S. *

Government is subject to restrictions as set forth in subparagraph *

(c)(l)(ii) of the Rights in Technical Data and Computer Software clause *

at DFARS 252.227-7013, or in FAR 52.227-19, as applicable. *

*

Contact ACSIOM at (413)545-6311 for more information *

**/

/**

* "DM_TRANS_TAB_C" is a C file that contains the document manager's

* document format and translator table. Each entry describes one

* recognized document format.

*/

/*

* RCS identifier for module:

*/

static char dm_trans_tab_c_rcsid[]="$RCSfile: dm_trans_tab.c,v $, $Revision: 2.27.2.2 $, $Date: 1996/05/10 16:35:31 $";

#include "docman.h"

extern Int_t email_trans(Char_t *buff, Int_t num, in_Document doc, Int_t mode);

```
extern Int_t sgml_trans(Char_t *buff, Int_t num, in_Document doc, Int_t mode);
extern Int_t cacm_dots_trans(Char_t *buff, Int_t num, in_Document doc, Int_t
mode);
extern Int_t html_trans(Char_t *buff, Int_t num, in_Document doc, Int_t mode);


/* The field description table has the following format:
*
* FIELD-NAME, DESCRIPTION, NORM-INDEX, FIELD-INDEX, DISPLAY,
RUN-RECOGS, COOP MODES
*/
FieldDesc email_fields[] = {
{"DOCID", "Message ID", OFF, INQ_NORMAL, INQ_ID,
OFF, INQ_COOP },
{"FROM", "FROM field", ON, INQ_NORMAL, INQ_TEXT,
OFF, INQ_COOP },
{"TO", "TO field", ON, INQ_NORMAL, INQ_TEXT,
OFF, INQ_COOP },
{"CC", "CC field", ON, INQ_NORMAL, INQ_TEXT,
OFF, INQ_COOP },
{"SUBJECT", "SUBJECT field", ON, INQ_NORMAL, INQ_TITLE,
OFF, INQ_COOP },
{"DATE", "DATE field", ON, INQ_NORMAL, INQ_SOURCE,
ON, INQ_COOP },
{"BODY", "Message Text", ON, INQ_NONE, INQ_TEXT,
OFF, INQ_COOP },
{ NULL, NULL, 0, 0, 0, 0, 0 }
};

FieldDesc sgml_fields[] =
{ {"TITLE", "Document Title", ON, INQ_NORMAL, INQ_TITLE, ON, INQ_COOP
},
```

{“SOURCE”, “Document Source”, OFF, INQ_NORMAL, INQ_SOURCE, OFF, INQ_COOP },
{“DOCID”, “External Doc Id”, OFF, INQ_NORMAL, INQ_ID, OFF, INQ_COOP },
{“DDATE”, “Document Date”, OFF, INQ_NORMAL, INQ_SOURCE, ON, INQ_COOP },
{“CDATE”, “Content Date”, OFF, INQ_NORMAL, INQ_NULL_TYPE, OFF, INQ_COOP },
{“AUTHOR”, “Document Author”, ON, INQ_NORMAL, INQ_SOURCE, ON, INQ_COOP },
{“TEXT”, “Body of Text”, ON, INQ_NONE, INQ_TEXT, ON, INQ_COOP },
{“MESH”, “Medical Mesh Field”, ON, INQ_NORMAL, INQ_TEXT, OFF, INQ_COOP },
{“NOINDEX”, “Non-indexed Field”, OFF, INQ_NONE, INQ_NULL_TYPE, OFF, INQ_OVERRIDE },
{ NULL, NULL, 0, 0, 0, 0, 0 }
};

FieldDesc cdots_fields[] =
{ {“TITLE”, “Article Title”, ON, INQ_NORMAL, INQ_TITLE, OFF, INQ_COOP },
{“DOCID”, “External Doc Id”, OFF, INQ_NORMAL, INQ_ID, OFF, INQ_COOP },
{“SOURCE”, “Journal Source”, OFF, INQ_NORMAL, INQ_SOURCE, OFF, INQ_COOP },
{“AUTHOR”, “Article Author”, OFF, INQ_NORMAL, INQ_SOURCE, OFF, INQ_COOP },
{“ABSTRACT”, “Article Abstract”, ON, INQ_NONE, INQ_TEXT, OFF, INQ_COOP },
{“KEYWORDS”, “Keywords”, ON, INQ_NONE, INQ_TEXT, OFF, INQ_COOP },
{“CATEGORY”, “Category”, OFF, INQ_NORMAL, INQ_NULL_TYPE, OFF,

INQ_COOP },

{“DATA-ENTRY”, “Data Entry Info”, OFF, INQ_NONE, INQ_NULL_TYPE,

OFF, INQ_COOP },

{“IGNORED”, “Non-indexed Field”, OFF, INQ_NONE, INQ_NULL_TYPE, OF-

F, INQ_OVERRIDE},

{ NULL, NULL, 0, 0, 0, 0, 0 }

};

FieldDesc html_fields[] =

{ {“TITLE”, “Document Title”, ON, INQ_NORMAL, INQ_TITLE, ON, INQ_COOP},

{“TEXT”, “Body of Text”, ON, INQ_NONE, INQ_TEXT, ON, INQ_COOP},

{“SOURCE”, “URL of doc”, OFF, INQ_NONE, INQ_SOURCE, OFF, INQ_OVERRIDE},

{“NOINDEX”, “Non-indexed Field”, OFF, INQ_NONE, INQ_NULL_TYPE, OF-

F, INQ_OVERRIDE},

{“ReportedBy”, “Reported By ”, ON, INQ_NORMAL, INQ_TEXT, OFF, IN-

Q_COOP },

{“PersonsFunctions”, “Persons Functions ”, ON, INQ_NORMAL, INQ_TEXT,

OFF, INQ_COOP },

{“FlightConditions”, “Flight Conditions ”, ON, INQ_NORMAL, INQ_TEXT, OF-

F, INQ_COOP },

{“AircraftType”, “Aircraft Type ”, ON, INQ_NORMAL, INQ_TEXT, OFF, IN-

Q_COOP },

{“AnomalyDesc”, “Anomaly Description”, ON, INQ_NORMAL, INQ_TEXT, OF-

F, INQ_COOP },

{“AnomalyDetector”, “Anomaly Detector ”, ON, INQ_NORMAL, INQ_TEXT,

OFF, INQ_COOP },

{“AnomalyResolution”, “Anomaly Resolution ”, ON, INQ_NORMAL, INQ_TEXT,

OFF, INQ_COOP },

{“AnomalyConsequences”, “Anomaly Consequences ”, ON, INQ_NORMAL, IN-

Q_TEXT, OFF, INQ_COOP },

{ NULL, NULL, 0, 0, 0, 0, 0 }

```
};

/* The DocFormat table has the following format:
*
* DOCFORMAT-NAME, TRANSLATOR-FUNCT, NUM-FIELDS, FIELD-TABLE
*
*/
DocFormat_t doc_types[] =
{
{ "email", email_trans, 7, email_fields },
{ "sgml", sgml_trans, 9, sgml_fields },
{ "cacm_dots", cacm_dots_trans, 9, cdots_fields },
{ "html", html_trans, 12, html_fields },
{ NULL, NULL, 0, NULL }
};
```

Appendix D

Modified Document Translator

```
/*************************************************************************\
Copyright (c) 1990-1995 by the *
Applied Computing Systems Institute of Massachusetts, Inc. (ACSIOM) *
All rights reserved. *
The INQUERY Software was provided by the *
Center for Intelligent Information Retrieval (CIIR), *
University of Massachusetts Computer Science Department, *
Amherst, Massachusetts. *
For more information, contact ACSIOM at 413-545-6311 *
\*************************************************************************/
%{


#include <string.h >
#ifdef _MSC_VER
#include <io.h >
#endif /* _MSC_VER **/
#include “docman.h”
#include “inerror.h”

#include “flexbuff.h” /* Flex no-copy buffering **/
```

```
/**********/
/* MACROS **/
/**********/

#define BLANK ' '
#define MAX_STACK_DEPTH 10
#define MAX_FNAME 128

#define YY_INPUT(buf, result, max_size) \
{\
old_buff = YY_CURRENT_BUFFER;\
yy_switch_to_buffer(new_buff);\
\
if(old_buff)\
free(old_buff);\
result=new_buff- >yy_buf_size;\
}

/*
* Module variables:
*/
static char rcsid_l[]="$ID$";

static in_Document curr_doc = (in_Document)NULL;
static char *doc_begin_pos = (char *)NULL;
static int at_eof, in_title = 0;
static int status;
static int stack_depth = 0;
static int save_depth = 0;
static struct
{
```

```
Char_t f_name[MAX_FNAME];
char *begin;
} field_stack[MAX_STACK_DEPTH], save_stack[MAX_STACK_DEPTH];


/*
* Function Prototypes:
*/
static int skip_tag(char *text, int leng);
static int end_doc(char *doc_end);
static int begin_doc(char *doc_end);
static int begin_field(Char_t *field_name, char *beg_pos);
static int finish_field(Char_t *field_name, char *end_pos);

static int first_head = 1;

%}

/* Start conditions **/
%p 4000
%a 4000
%o 5000
%n 1000

%Start HEAD BODY

/* Regular definitions **/

w1 ([a-zA-Z][a-zA-Z0-9]*)
w2 ([a-zA-Z]+(\'[a-zA-Z]*)*)
w3 ([0-9]+[A-Za-z][a-zA-Z0-9]*)
```

```
s ([0-9]+\.[0-9a-zA-Z]+(\.[0-9a-zA-Z]+)+)
a (([a-zA-Z]+\.)([a-zA-Z]+\.)+)
h ([A-Za-z]+\-\n[A-Za-z0-9]+)
wordobj ({w1}—{w2}—{w3}—{s}—{a})


n1 ([0-9]+(\.[0-9]+)?)
n2 ([0-9]{1,3}(\,[0-9][0-9][0-9])+(\.[0-9]+)?)
n3 (\.[0-9]+)
numobj ({n1}—{n2}—{n3})

textobj ({wordobj}—{numobj})

title ([tT][iI][tT][lL][eE]([ ][< >]+)?)
url ([uU][Rr][lL])
html ([Hh][Tt][Mm][Ll])
head ([Hh][Ee][Aa][Dd])
body ([Bb][Oo][Dd][Yy])
base ([Bb][Aa][Ss][Ee])

%%

\<{head}\ > { /** Begining of the header **/
if (!first_head)
skip_tag(yytext, yyleng);
else {
first_head = 0;
BEGIN HEAD;
status = finish_field("TEXT", yytext);
if(status != INQ_SUCCESS)
return status;
}
}
```

```
<HEAD >\<\/{url}\ > { /** process end source fields **/
status = finish_field("SOURCE", yytext);
if(status != INQ_SUCCESS)
return status;
}

<HEAD >\<{url}\ > { /** process source fields **/
status = begin_field("SOURCE", yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}

<BODY,HEAD >\<{title}\ > { /** start tag for title field **/
skip_tag(yytext, yyleng);
status = begin_field("TITLE", yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
in_title = 1;
}

<BODY,HEAD >\<\/{title}\ > { /** end tag for title field **/
if (in_title) {
status = finish_field("TITLE", yytext);
if(status != INQ_SUCCESS)
return status;
}
skip_tag(yytext, yyleng);
in_title = 0;
}

<BODY,HEAD >\<ReportedBy\ > { /** start tag for title field **/
```

```
skip_tag(yytext, yyleng);
status = begin_field(“ReportedBy”, yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}

<BODY,HEAD >\<\/ReportedBy\ > { /** end tag for title field **/
status = finish_field(“ReportedBy”, yytext);
if(status != INQ_SUCCESS)
return status;
skip_tag(yytext, yyleng);
}
<BODY,HEAD >\<PersonsFunctions\ > { /** start tag for title field **/
skip_tag(yytext, yyleng);
status = begin_field(“PersonsFunctions”, yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}

<BODY,HEAD >\<\/PersonsFunctions\ > { /** end tag for title field **/
status = finish_field(“PersonsFunctions”, yytext);
if(status != INQ_SUCCESS)
return status;
skip_tag(yytext, yyleng);
}
<BODY,HEAD >\<FlightConditions\ > { /** start tag for title field **/
skip_tag(yytext, yyleng);
status = begin_field(“FlightConditions”, yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}
```

```
<BODY,HEAD >\<\/FlightConditions\ > { /** end tag for title field **/
status = finish_field("FlightConditions", yytext);
if(status != INQ_SUCCESS)
return status;
skip_tag(yytext, yyleng);
}
<BODY,HEAD >\<AircraftType\ > { /** start tag for title field **/
skip_tag(yytext, yyleng);
status = begin_field("AircraftType", yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}

<BODY,HEAD >\<\/AircraftType\ > { /** end tag for title field **/
status = finish_field("AircraftType", yytext);
if(status != INQ_SUCCESS)
return status;
skip_tag(yytext, yyleng);
}
<BODY,HEAD >\<AnomalyDesc\ > { /** start tag for title field **/
skip_tag(yytext, yyleng);
status = begin_field("AnomalyDesc", yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}

<BODY,HEAD >\<\/AnomalyDesc\ > { /** end tag for title field **/
status = finish_field("AnomalyDesc", yytext);
if(status != INQ_SUCCESS)
return status;
skip_tag(yytext, yyleng);
}
```

```
<BODY,HEAD >\<AnomalyDetector\ > { /** start tag for title field **/
skip_tag(yytext, yyleng);
status = begin_field("AnomalyDetector", yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}

<BODY,HEAD >\<\/AnomalyDetector\ > { /** end tag for title field **/
status = finish_field("AnomalyDetector", yytext);
if(status != INQ_SUCCESS)
return status;
skip_tag(yytext, yyleng);
}
<BODY,HEAD >\<AnomalyResolution\ > { /** start tag for title field **/
skip_tag(yytext, yyleng);
status = begin_field("AnomalyResolution", yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}


<BODY,HEAD >\<\/AnomalyResolution\ > { /** end tag for title field **/
status = finish_field("AnomalyResolution", yytext);
if(status != INQ_SUCCESS)
return status;
skip_tag(yytext, yyleng);
}
<BODY,HEAD >\<AnomalyConsequences\ > { /** start tag for title field **/
skip_tag(yytext, yyleng);
status = begin_field("AnomalyConsequences", yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}
```

```
<BODY,HEAD >\<\/AnomalyConsequences\ > { /** end tag for title field
**/
status = finish_field(“AnomalyConsequences”, yytext);
if(status != INQ_SUCCESS)
return status;
skip_tag(yytext, yyleng);
}
<HEAD >\<\/{head}\ > { /** end of header **/
BEGIN 0;
}


<BODY >\<{body}\ > { /** begining of the body **/
skip_tag(yytext, yyleng);
}

\<{body}\ > { /** begining of the body **/
BEGIN BODY;
status = begin_field(“TEXT”, yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
}

<BODY >\<\/{body} > { /** end of body **/
BEGIN 0;
status = finish_field(“TEXT”, yytext);
if(status != INQ_SUCCESS)
return status;
}
<BODY >\<\/{html}\ > { /** end of doc tag if there was no <body > **/
BEGIN 0;
status = finish_field(“TEXT”, yytext);
```

```
if(status != INQ_SUCCESS)
return status;
status = end_doc(yytext + yyleng);
return status;
}

<BODY,HEAD >\<([\/]?[a-zA-Z][< >]*)(\n([< >]*))*\ > —
<BODY,HEAD >\<!–[< >]*–\ > { /** erase the tag and return nothing **/
skip_tag(yytext, yyleng);
}


\<\/{html}\ > { /** end of doc tag if ther was a <body > **/
status = end_doc(yytext + yyleng);
return status;
}

\<{html}([ ]+[< >]*)?\ > { /** begin tag for document **/
BEGIN BODY;
first_head = 1;
status = begin_doc(yytext);
if(status != INQ_SUCCESS)
return status;
status = begin_field("TEXT", yytext + yyleng);
if(status != INQ_SUCCESS)
return status;
} [\ t\n] { /** Ignore whitespace **/
}

. { /** Ignore unexpected stuff **/
}
```

```
<<EOF > > {
/** Didn't find a complete document **/
BEGIN 0;
return INQ_END_NOT_FOUND;
}
%%

#include "flexbuff.c" /** Flex no-copy buffering **/

Int_t
html_trans(Char_t *inBuff, Int_t numChars, in_Document doc, int eof, Int_t mode)
{
int status;

if(numChars < 6 ) /** A minimum document must have a <DOC > tag **/
return INQ_END_NOT_FOUND;

fill_flex_buff(inBuff, numChars);
curr_doc = doc;
at_eof = eof;
stack_depth = 0;



status = html_trans_lex();

flush_flex_buff();
return(status);
}


static int
```

```
begin_doc(char *beg_pos)
{
doc_begin_pos = beg_pos;
return INQ_SUCCESS;
}

static int
end_doc(char *end_pos)
{
in_ByteSequence doc_bs;

if (doc_begin_pos == (char *)NULL)
{
Q_warning("Begin doc not found before this end doc");
return INQ_SYNTAX_ERR;
}

if(stack_depth != 0)
{
Q_warning("%d field(s) unended at end of document. (last is %s)",
stack_depth,field_stack[stack_depth].f_name);
return INQ_SYNTAX_ERR;
}

doc_bs.bytes = doc_begin_pos;
doc_bs.length = end_pos - doc_begin_pos;

dm_document_set_raw_data(curr_doc, doc_bs);

doc_begin_pos = (char *)NULL;
return INQ_SUCCESS;
}
```

```
static int
begin_field(Char_t *field_name, char *beg_pos)
{
if(stack_depth == MAX_STACK_DEPTH)
{
Q_warning("Fields stacked too deep.");
return INQ_SYNTAX_ERR;
}
printf("starting field %s\n",field_name);
strcpy(field_stack[stack_depth].f_name, field_name);
field_stack[stack_depth].begin = beg_pos;
stack_depth++;

return INQ_SUCCESS;
}

static int
finish_field(Char_t *field_name, char *end_pos)
{

if(stack_depth == 0)
{
Q_warning("End of %s field found before beginning.", field_name);
return INQ_SYNTAX_ERR;
}

stack_depth--;
if(strcmp(field_name, field_stack[stack_depth].f_name) != 0)
{
Q_warning("End of field %s not found. Can't finish %s.",
field_stack[stack_depth].f_name,field_name);
```

```
return INQ_SYNTAX_ERR;
}

/** printf("finishing field %s\n",field_name); **/
if(dm_document_mark_span(curr_doc, field_name,
field_stack[stack_depth].begin - doc_begin_pos,
end_pos - field_stack[stack_depth].begin)
!= INQ_SUCCESS)
{
Q_warning("dm_document_mark_span returned an error.");
return INQ_FAILED;
}

return INQ_SUCCESS;
}

int
html_trans_wrap()
{
return 1;
}

static int
skip_tag(char *text, int leng)
{
int status,i;


/** save this one in the NOINDEX field **/
if ((status = begin_field("NOINDEX",text)) != INQ_SUCCESS)
return(status);
if ((status = finish_field("NOINDEX",text+leng)) != INQ_SUCCESS)
```

```
return(status);

return (INQ_SUCCESS);
}
```

Appendix E

HTML Source Code

E.1 default.html

```
<HTML >
<HEAD >
<TITLE >
Inquery </TITLE >
</HEAD >


<FRAMESET ROWS=“20%,*” >
<FRAME NAME=“Title” SRC=“inqt.html ” SCROLLING=“auto” NORESIZE
MARGINHEIGHT=“1” >
<FRAMESET COLS=“80%” >
<FRAME SRC=“inq1.html” NAME=“Main” SCROLLING=“auto” NORESIZE
MARGINHEIGHT=“1” >
</FRAMESET >
</FRAMESET >


<NOFRAMES >
```

To use this file, you need to have Netscape Navigator 3.0 installed on your computer.

</NOFRAMES >

</HTML >

E.2 inq1.html

<HTML >
<HEAD >
<BODY BACKGROUND="bluestone.jpg" TEXT="#000000" BGCOLOR="#3D74EF" link="#0000ff" vlink="#0000aa" >
<TITLE >
INQUERY </TITLE >
</HEAD >
<BODY >

<h1 >
Main Page</h1 >
<h3 >
Overview </h3 >
<p >
"The Aviation Safety Reporting System (ASRS) was established in 1975 under a Memorandum of Agreement between the Federal Aviation Administration (FAA) and the National Aeronautics and Space Administration (NASA). FAA provides most of the program funding; NASA administers the program and sets its policies in consultation with the FAA and the aviation community. NASA has chosen to operate the program through a contractor selected via competitive bidding. The current contractor is Battelle Memorial Institute."</p >

<p >

From the ASRS database at <A HREF=“http://www-afo.arc.nasa.gov/ASRS” target=”_top” >

“http://www-afo.arc.nasa.gov/ASRS” </A >

</p >

<p >

The ASRS collects voluntarily submitted aviation safety incident reports which it analyses and replies to in an attempt to reduce the probability of aviation accidents. The reports submitted are held in the strictest confidence with regards to those involved in the incident with more than 300,000 reports collate the ASRS assures us that a reporters identity has never been breached. This seems the most important fact that ensures people report aviation incidents.</p >

<p >

The following query entry allows you to search a set of 50 incident reports converted into an inquery database. For information on how to form inquery queries click <A HREF=“inq2.html” target=“Main” >

here </A >

. The inquery database has been modified to allow searches in specific fields contained within the ASRS reports, for information on what fields are supported and the possible text they contain click <A HREF=“inq3.html” target=“Main” >

here </A >

. <hr >

<h3 >

Query Entry </h3 >

<hr >

<p >

Please enter your query: (in standard inquery syntax) </p >

<FORM action=“http://students.dcs.gla.ac.uk/scripts/personal/mcelropd/inq.pl” METHOD=“POST” >

<input name=“Query” size=“40” type=“text” value=“” >

<p >

</p >

<INPUT TYPE="submit" VALUE="Submit Query" NAME="Submit Button"

>

<INPUT TYPE="RESET" VALUE="Reset Query" name="Reset Button" >

</FORM >

<hr >

<p >

When you are finished, please fill out our questionnaire <A HREF="quest.html"

target="Main" >

here </A >

. <hr >

</BODY >

</HTML >

E.3 inq2.html

<HTML >

<HEAD >

<BODY BACKGROUND="bluestone.jpg" TEXT="#000000" BGCOLOR="#3D74EF"

link="#0000ff" vlink="#0000aa" >

<TITLE >

INQUERY QUERY FORMULATION HELP </TITLE >

</HEAD >

<BODY >

<h1 >

Inquery Query Formulation Help </h1 >

<p >

Queries may be of two types, natural language or structured. The natural language queries allow the user to simply type the information request as an English sentence. The Inquery query-processor transforms these queries into a structured form which can then be processed by the query engine. By putting the query in structured form directly, the user is able to provide a precise definition of the term relationships in the query, possibly resulting in improved performance. However it requires a knowledgeable user to properly formulate a query using the special operators provided. The available INQUERY operators are described in the following sections. </p >

<hr >

<p >

Sum Operator: #sum (T1 ...Tn) The terms or nodes contained in the sum operator are treated as having equal influence on the final result. The belief values provided by the arguments of the sum are averaged to produce the belief value of the #sum node. </p >

<hr >

<p >

Weighted Sum Operator: #wsum (Ws W1 T1 ... Wn Tn) The terms or nodes contained in the wsum operator contribute unequally to the final result according to the weight associated with each (Wx). The final belief value is scaled by Ws, the weight associated with the #wsum itself. </p >

<hr >

<p >

Ordered Distance Operator: #N (T1 ... Tn) or #odN (T1 ... Tn) The terms within an ODN operator must be found within N words of each other in the text in order to contribute to the document's belief value. The "#N" version is an abbreviation of #ODN, thus #3(health care) is equivalent to #od3(health care). </p >

<hr >

<p >

And Operator: #and(T1 ... Tn) The more terms contained in the AND operator which are found in a document, the higher the belief value of that document. </p >

<hr >

<p >

Boolean And Operator: #band(T1 ... Tn) All of the terms within a BAND operator must be found in a document in order for this operator to contribute to the belief value of that document. </p >

<hr >

<p >

Boolean And Not Operator: #bandnot (T N) Search for document matching the first argument but not the second. </p >

<hr >

<p >

Or Operator: #or(T1 ... Tn) One of terms within the OR operator must be found in a document for that document to get credit for this operator. </p >

<hr >

<p >

Negation Operator: #not(T1 ... Tn) The terms or nodes contained in this operator are negated so that documents which do not contain them are rewarded. </p >

<hr >

<p >

Un-ordered Window Operator: #uwN(T1 ... Tn) The terms contained in a UWN operator must be found in any order within a window of N words in order for this operator to contribute to the belief value of the document. </p >

<hr >

<p >

Phrase Operator: #phrase(T1 ... Tn) Terms within this operator are evaluated to determine if they occur together frequently in the collection. If they do, the operator is treated as an ordered distance operator of 3 (#od3). If the arguments are not found to co-occur in the database, the phrase operator is turned into a SUM operator. In ambiguous cases the phrase becomes the MAX of the SUM and the OD3 operators. </p >

<hr >

<p >

Passage Operator: #passageN(T1 ... Tn) The passage operator looks for the terms or nodes within the operator to be found in a passage window of N words.

The document is rated based upon the score of it's best passage. </p >

<hr >

<p >

Synonym Operator: #syn(T1 ... Tn) The terms of the operator are treated as instances of the same term. </p >

<hr >

<p >

Maximum Operator: #max(T1 ... Tn) The maximum belief value of all the terms or nodes contained in the MAX operator is taken to be the belief value of this operator. </p >

<hr >

<p >

Weight Plus Operator: #+ T1 - The effect of the following term or node is increased relative to the rest of the query. </p >

<hr >

<p >

Weight Minus Operator: #- T1 - The effect of the following term or node is decreased relative to the rest of the query. </p >

<hr >

<p >

Literal Operator: #lit(T1 ... Tn) This operator preserves the original forms of the terms contained within it. No stemming or stopping is performed and capitalization is preserved. </p >

<hr >

<p >

Filter Require Operator: $filreq(arg1 arg2) Use the documents returned (belief list) of the first argument if and only if the second argument would return documents. The value of the second argument does not effect the belief values of the first argument; only whether they will be returned or not. </p >

<hr >

<p >

Filter Reject Operator: #filrej(arg1 arg2) Use the documents returned by the first argument if and only if there were no documents returned by the second argument. The value of the second argument does not effect the belief values of the first argument; only whether they will be returned or not. </p >

<hr >

<p >

Field Operator: #field(FIELD-NAME #REL-OP T1 ... Tn) The terms contained in a FIELD operator are searched only within the FIELD-NAME specified. The relational operator (REL-OP) allows fields to be searched for ranges of values. If the REL-OP is missing, equality is used by default. </p >

<hr >

<A HREF="inq1.html" target="Main" >

<IMAGE SRC="Back.gif" >

</A >

</BODY >

</HTML >

E.4 inq3.html

<HTML >

<HEAD >

<BODY BACKGROUND="bluestone.jpg" TEXT="#000000" BGCOLOR="#3D74EF" link="#0000ff" vlink="#0000aa" >

<TITLE >

FIELDS DESCRIPTION </TITLE >

</HEAD >

<BODY >

<h1 >

Fields Description </h1 >

<p >

A description of the searchable fields and their possible values follows:</p >

<p >

(Please note each entry in a category is separated by a semicolon (e.g., two SMAs in one incident would be coded as “SMA;SMA;” in the AircraftType category.)</p >

<hr >

<p >

ReportedBy - role of the person who reported the occurrence/situation. Codes used are: </p >

FLC - flight crew

PLT - pilot

CRM - crew member

CTLR - Air Traffic Controller

PAX - passenger

OBS - observer

AFC (or AIR) - Air Force

NVY - Navy

UNK - unknown.

<hr >

<p >

PersonsFunctions - description of a person’s function at the time of the occurrence. Codes are built by entering the person followed by ’.’ and then what they were doing, for example FLC.PLT (i.e. the flight crew was a pilot in a single person crew) or TWR.LC (i.e. The tower was a local controller). The codes used are:</p >

<hr >

<p >

FLC </p >

<hr >

PIC - Pilot in command as determined by official designation, prior consensus, or actually controlling the aircraft

CAPT - Captain role in a multi-person flight crew

FO - First Officer/Copilot role in a multi-person flight crew

SO - Second Officer/Flight Engineer role in a multi-person flight crew

OTH - Additional crew member (e.g., navigator) in a multi-person flight crew

CKP - Check pilot (essential flight crew member occupying a crew position/role)

ISTR - Legally qualified flight instructor who is giving instruction at the time of the occurrence/situation

PLT - Pilot in a single-person crew

TRNEE - Flight crew member in training.

<hr >

<p >

TWR</p >

<hr >

LC - Local controller

GC - Ground controller

FD - Flight data position

OTH - Other

COORD - Coordinator position

CD - Clearance delivery

SUPVR - Supervisor

TRNEE - Trainee

<hr >

<p >

TRACON</p >

<hr >

AC - Approach controller

DC - Departure controller

RHO - Radar hand-off position

FD - Flight data position

TRNEE - Trainee

COORD - Coordinator position

SUPVR - Supervisor

OTH - Other

<hr >

<p >

ARTCC</p >

<hr >

M - Manual controller

R - Radar controller

H - Hand-off position

D - Assistant or data man

COORD - Coordinator position

SUPVR - Supervisor

OTH - Other

TRNEE - Trainee

<hr >

<p >

MIL</p >

<hr >

PAR - Precision approach radar

RSU - Runway supervisory unit

OTH - Other

<hr >

<p >

MISC<p >

<hr >

FSS - Flt service station specialist

ACI - Air carrier inspector

UNI - Unicom operator

FBO - Fixed base operator/employee

CAB - Cabin attendant

VD - Vehicle driver

PAX - Passenger

CGP - Company ground personnel

DISP - Dispatcher

CENR - Company enroute check personnel

TADV - Tower advisory

AMGR - Airport manager

OBS - Observer

SUPVR - Supervisor

OTH - Other

<hr >

<p >

FlightConditions - the weather environment at the time of the occurrence or situation in terms of the conventional definition for flight conditions. Codes used are:

</p >

VMC - visual meteorological conditions

IMC - instrument meteorological conditions

MXD - mixed flight conditions (both VMC and IMC)

MVI - marginal VFR

SVF - special VFR.

<hr >

<p >

AircraftType - the aircraft type involved in the incident differentiated by arbitrary gross takeoff weight ranges (military aircraft type are differentiated by function). Codes used are:</p >

SMA - small aircraft (less than 5,000 lbs)

SMT - small transport (5001 - 14,500 lbs)

LTT - light transport (14,501 - 30,000 lbs)

MDT - medium transport (30,001 - 60,000 lbs)

MLG - medium large transport (60,001 - 150,000 lbs)

LRG - large transport (150,001 - 300,000 lbs)

HVT - large transport (over 300,000 lbs)

WDB - wide-body (over 300,000 lbs)

ULT - ultralight (including hang gliders)

SPN - sailplane/glider

SPC - special purpose

FGT - fighter

BMB - bomber

MLT - military transport

MTR - military trainer

<hr >

<p >

AnomalyDesc </p >

<Ul >

Acft Equipment Problem/Critical Aircraft

Acft Equipment Problem/Less Severe

Alt Deviation

Alt Dev/Undershoot on Clb or Des

Alt Dev/Excursion From Assigned

Alt Dev/Xing Restriction

Alt-Hdg Rule Deviation

Conflict/NMAC (near midair collision)

Conflict/Airborne Less Severe

Conflict/Ground Critical

Conflict/Ground Less Severe

Controlled Flt Toward Terrain

Erroneous Penetration or Exit Airspace

In-Flt Encounter/Other

In-Flt Encounter/WX

Less Than Legal Separation

Loss of Acft Control

Non Adherence Legal Rqmt/Clnc

Non Adherence Legal Rqmt/FAR

Non Adherence Legal Rqmt/Published Proc

Non Adherence Legal Rqmt/Other

Rwy or Txwy Excursion

Rwy Transgress/Other

Rwy Transgress/Unauth Lndg

Speed Deviation

Track or Hdg Deviation

Unctrl Arpt Traffic Pattern Deviation

<hr >

<p >

AnomalyDetector</p >

COCKPIT/FLC;

COCKPIT/EQUIPMENT;

OTHER

ATC/CTLR;

ATC/EQUIPMENT

<hr >

<p >

AnomalyResolution</p >

NOT RESOLVED/ANOMALY ACCEPTED;

FLC RETURNED ACFT TO ORIGINAL CLNC OR INTENDED COURSE;

OTHER;

NOT RESOLVED/OTHER

FLC OVERCAME EQUIP PROBLEM

CTLR INTERVENED

FLC BECAME REORIENTED

NOT RESOLVED/DETECTED AFTER-THE-FACT;

NOT RESOLVED/INSUFFICIENT TIME

FLC AVOIDANCE-EVASIVE ACTION;

NOT RESOLVED/UNABLE; OTHER

ACFT EXITED ADVERSE

CTLR ISSUED NEW CLNC

<hr >

<p >

AnomalyConsequences </p >

NONE

FAA INVESTIGATORY FOLLOW-UP;

OTHER;

INJURY; EMOTIONAL TRAUMA;

ACFT DAMAGED;

FLC/ATC REVIEW;

```
</ul >
<hr >
<A HREF=“inq1.html” target=“Main” >
<IMAGE SRC=“Back.gif” >
</A >
</BODY >
</HTML >
```

E.5 quest.html

```
<HTML >
<HEAD >
<BODY BACKGROUND=“bluestone.jpg” TEXT=“#000000” BGCOLOR=“#3D74EF”
link=“#0000ff” vlink=“#0000aa” >
<TITLE >
INQUERY </TITLE >
</HEAD >
<BODY >
<h1 >
Questionnaire </h1 >
<hr >
<FORM action=“http://students.dcs.gla.ac.uk/scripts/personal/mcelropd/quest.pl”
METHOD=“POST” >
<p >
How useful would you rate the database? </p >
<SELECT name=“Db_Useful” >
<OPTION SELECTED >
Very Useful <OPTION >
Useful <OPTION >
```

Indifferent <OPTION >

Not Very Useful </SELECT >

<p >

In terms of ease of use how would you rate using the database? </p >

<SELECT name="Db_Power" >

<OPTION SELECTED >

Very Easy <OPTION >

Easy <OPTION >

Indifferent <OPTION >

Not Very Easy </SELECT >

<p >

How powerful do feel the database retrieval is? </p >

<SELECT name="Db_Power" >

<OPTION SELECTED >

Very Powerful <OPTION >

Powerful <OPTION >

Indifferent <OPTION >

Not Very Powerful </SELECT >

<p >

On average, How relevant did you think the documents returned (to your query's) were? </p >

<SELECT name="Relevant_docs" >

<OPTION SELECTED >

Very Relevant <OPTION >

Relevant <OPTION >

Indifferent <OPTION >

Not Very Relevant </SELECT >

<p >

How useful did you find the fields transactions? </p >

<SELECT name="Fields_useful" >

<OPTION SELECTED >

Very Useful <OPTION >

```
Useful <OPTION >
Indifferent <OPTION >
Not Very Useful </SELECT >
<p >
Please enter any additional comments you may have. </p >
<TEXTAREA name=“comments” rows=14 cols=100 >
</TEXTAREA >
<p >
</p >
<INPUT TYPE=“submit” VALUE=“Submit Questionnaire” NAME=“Submit
Button” >
<INPUT TYPE=“RESET” VALUE=“Reset Questionnaire” name=“Reset But-
ton” >
</FORM >
<hr >
<A HREF=“inq1.html” target=“Main” >
<IMAGE SRC=“Back.gif” >
</A >
</BODY >
</HTML >
```

E.6 inqt.html

```
<HTML >
<BODY BACKGROUND=“bluestone.jpg” TEXT=“#000000” BGCOLOR=“#3D74EF”
link=“#0000ff” vlink=“#0000aa” >
<BODY >
<CENTER >
<h3 >
Quick Navigation </h3 >
<hr >
```

<A HREF=“inq1.html” target=“Main” >

— Main Page —</A >

<A HREF=“inq2.html” target=“Main” >

Query Formulation Help Page —</A >

<A HREF=“inq3.html” target=“Main” >

Fields Description Page —</A >

<A HREF=“quest.html” target=“Main” >

Questionnaire —</A >

<hr >

</CENTER >

</BODY >

</HTML >

Appendix F

Questionnaire CGI Script

```
#!/usr/local/bin/perl

print “Content-type:text/html\n\n”;

read(STDIN, $buffer, $ENV{’CONTENT_LENGTH’});

@pairs = split(/&/, $buffer);

foreach $pair (@pairs) {

($name, $value) = split(/=/, $pair);

$value =~tr/+/ /;

$value =~s/%([a-fA-F0-9][a-fA-F0-9])/pack(“C”, hex($1))/eg;

$FORM{$name} = $value;
```

```
}

print "<HTML > <HEAD > ";
print "<TITLE > Questionnaire Recieved</TITLE > ";
print "</HEAD > ";
print "<BODY > <h1 > Questionnaire </h1 > ";
print "<p > Your questionnaire has been recieved</p > ";
print "<p > Thanks for your support </p > ";
print "<A HREF=\"http://www.dcs.gla.ac.uk/m̃celropd/inquery/inq1.html\" target=\"Main\"
> Return to main page ... </A > ";
print "</BODY > </HTML > ";

open(QDT, " > > /local/www.stud/students/mcelropd/scripts/ques.txt");
print QDT "**************************************************\n";
foreach $key (keys(%FORM)) {

print QDT "$key = $FORM{$key}\n";

}
print QDT "**************************************************\n";
print QDT "\n\n";
close(QDT);

print "</body > </html > ";
```

Appendix G

Sample Evaluation Sheet

G.1 Causal Tree Incident Description Method

Causal trees (Van Vuuren and Van der Schaaf, 1995), are essentially derived from fault trees. They are used as a means to present, in chronological order, the decisions and critical activities that occur during an incident. Also, they are used to convey how all decisions and activities relate to each other in a logical sense. When representing near misses using causal trees the tree is divided into two sections:

Failure side - which represents the activities which lead to the failure.

Recovery side - which represents activities that have prevented the incident from becoming a real accident.

The causal tree forces us to understand that there is never one single cause of incidents but in fact many factors contribute. These factors are a combination of technical, organisational and human causes and are found at the bottom of the tree on the failure side. As with fault trees factors are combined with AND & OR logic leading up to the description of the overall incident at the root of the tree (An example is shown below).

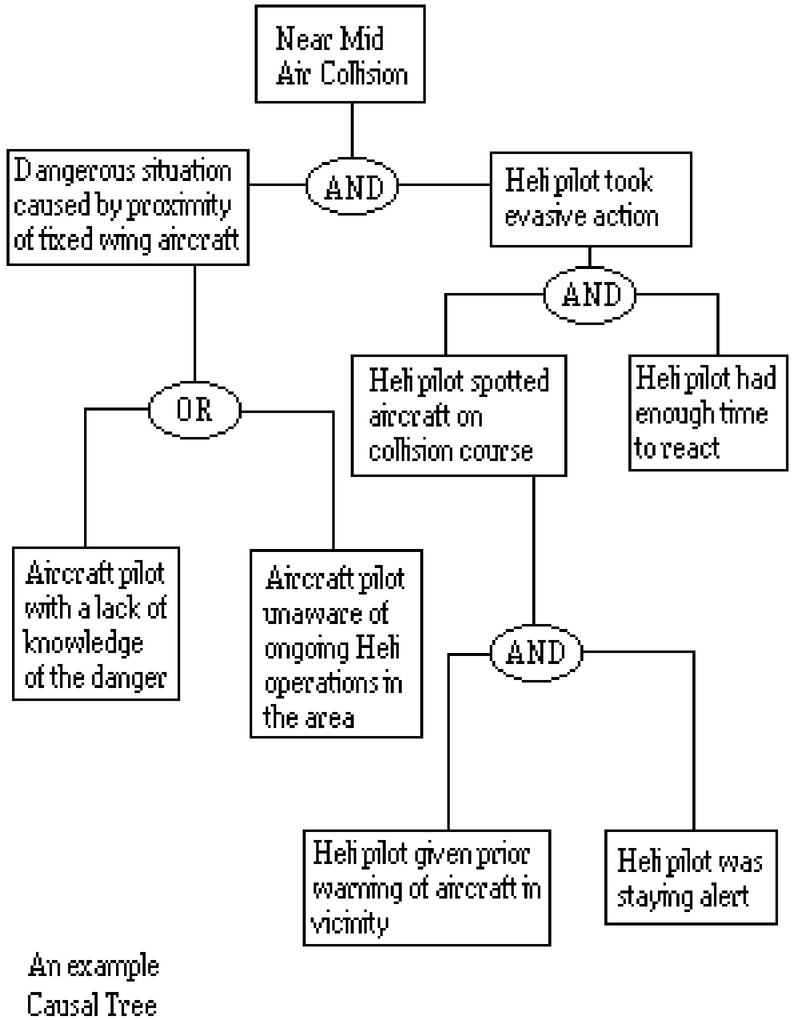

Figure G.1: Example Causal Tree

Create a Causal tree for the following incident: (Please ask if anything is unclear.)

"DEPARTING NEWPORT AIRPORT, AT THE TIME OF DEPARTURE, THE WEST HALF OF THE AIRPORT WAS STARTING TO FOG IN. I HOVER-TAXIED TO THE FAR EAST END OF THE AIRPORT AND WAS ABLE TO TAKE OFF IN BLUE SKIES AND UNLIMITED VISIBILITY. THIS AIRPORT IS SET UP FOR A CONTROL ZONE WHEN THE VISIBILITY IS LESS THAN 3 MILES AND A 1000 FT CEILING. THERE WAS ANOTHER HELICOPTER IN THE PATTERN WHOM I WAS IN RADIO CONTACT WITH. HE GAVE

ME PERMISSION TO TAKE OFF FIRST AND THEN HE WENT IN AND LANDED. ALL OF THIS WAS DONE BY VISUAL FLIGHT RULE ON THE EAST END OF THE FIELD WHILE THE WEST END WAS FOGGED IN. THE STANDARD FOR THE OTHER AIRPORTS WITH CONTROL TOWERS HAS BEEN IF I WAS INSIDE OF THEIR CONTROL ZONE AND IT WAS IN EFFECT, THEY HAVE ALLOWED ME TO WORK INSIDE THE CONTROL ZONE WITHOUT A SPECIAL VISUAL FLIGHT RULE IF I WAS IN THE STANDARD VISUAL FLIGHT RULE CONDITIONS. ALL I NEEDED TO DO WAS MAKE REPORTS OF MY LOCATIONS WHILE WORKING IN THEIR AIRSPACE. AS LONG AS I WAS ON VISUAL FLIGHT RULE, I DID NOT NEED A SPECIAL VISUAL FLIGHT RULE TO BE INSIDE THE AIRSPACE. MY POINT TO ALL OF THIS IS THAT IT IS NOT TAUGHT TO NEW STUDENTS THIS WAY SO IT BECOMES MORE LIKE JUST A STORY WHEN AN OLDER PILOT DOES SOMETHING LIKE THIS. IT IS LEGAL TO DO BUT NOT GOOD FOR STUDENTS TO SEE. NOT SURE OF HOW OR WHERE TO MAKE A POINT OF THIS, OR IF MAYBE IT IS NOT A RELATIVE POINT TO MAKE AT ALL. HOPE THIS IS NOT TOO CONFUSING, AND THANK YOU FOR YOUR TIME."

"SYNOPSIS - A COMMERCIAL PILOT TOOK OFF IN AN SMALL AIRCRAFT HELICOPTER FROM A DESIGNATED CLASS EAST OF NON TOWER AIRPORT WHEN WEATHER CONDITIONS REQUIRED A SPECIAL VISUAL FLIGHT RULE CLEARANCE. THE PILOT DID NOT HAVE A CLEARANCE, OR BELIEVED THAT HE NEEDED ONE, SINCE HE WAS OPERATING A HELICOPTER WHEN THE FLIGHT VISIBILITY MET THE HELICOPTER FLIGHT VISIBILITY REQUIREMENT."

G.2 Eindhoven Model

Once the tree is created, classification takes place by subjectively analysing the tree and running through the Eindhoven model assigning appropriate root causes. The classification model is used as a checklist to analyse incident data. In terms of human error the model represents Skill Based, Rule Based and Knowledge based behaviour represented in the model as S-B, R-B and K-B respectively. The classification of root causes under the model are as follows:

TE - A technical factor of engineering indicating a wrong design.
TC - A technical factor of construction indicating that the design
was not followed properly during the construction phase.
TM - A technical factor indicating unexplainable material defects.
OP - (Operating Procedures) refers to the inadequate quality of procedures.
OM - (Management Priorities) refers to any de facto pressure by top - or middle management to deviate from the formal organisational priorities.
HK1 - The correct status and dynamics of the system to be controlled must be known to the operator.
HK2 - The goal, or priorities of goals, must be known and understood by the operator.
HR1 - The operator in question must be qualified to do the job.
HR2 - If applicable the operator must obtain a temporary permit for activities where extra risk is involved.
HR3 - The preparation of the job itself starts by informing other operators, if necessary, of the work to be done.
HR4 - When arriving at the job location the local system status should be checked to comply with the expected conditions.
HR5 - The job itself should be planned correctly, i.e. the appropriate methods should be chosen and carried out in order.
HR6 - The prescribed equipment and information for a proper job performance should be present and used.
HS1 & HS2 - The execution of the required actions themselves implies successful correct movements; both controlled, i.e. intended, detailed, movements (e.g. to manipulate tools and request information), and maintaining the correct body position in order to make the controlled movements possible.
X - Unclassifiable behaviour

Using the Causal tree you have just created classify under the Eindhoven model the previous incident. (Please ask if anything is unclear.)

Appendix H

Sample Questionnaire

Please circle the relevant answer:

How useful would you rate the database?

- Very Useful
- Useful
- Indifferent
- Not Very Useful

In terms of ease of use how would you rate using the database?

- Very Easy
- Easy
- Indifferent
- Not Very Easy

How powerful do feel the database retrieval is?

- Very Powerful

- Powerful
- Indifferent
- Not Very Powerful

On average, How relevant did you think the documents returned (to your query's) were?

- Very Relevant
- Relevant
- Indifferent
- Not Very Relevant

How useful did you find the fields transactions?

- Very Useful
- Useful
- Indifferent
- Not Very Useful

Please enter any additional comments you may have.