Adding a Truth Maintenance System to ERA, the Electronic Referee Assistant, to allow backtracking

David Crighton

Master of Science
School of Informatics
University of Edinburgh
2005

(Graduation date: November 2005)

Abstract

ERA, the Electronic Referee Assistant is an expert system which is designed to help novice referees produce reviews of Informatics papers. The most common criticism of the previous version of this system was that users are unable to return to previous sections of the review to amend their results. This project reimplements ERA on a reusable web based logic programming framework which provides this functionality by using a Truth Maintenance System to retain consistency in the underlying knowledge base.

Acknowledgements

First and foremost I thank Professor Alan Bundy and Doctor Stephen Potter for their extensive help, advice and support throughout every stage of this project. I would also like to thank Jonathan Betts, Marc Roberts and John Henry for giving up so much of their time to help me evaluate the system. Finally I would like to thank my girlfriend Amy for her support and last minute proof reading.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*David Crighton*)

Table of Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Objectives 2

1.3 Document Structure 2

2 Background **5**

2.1 Expert Systems 5

2.1.1 Expert System Methods 7

2.2 Expert System Shells 11

2.2.1 CLIPS family 11

2.2.2 JESS 12

2.2.3 Drools 12

2.3 Expert Systems and Informatics Paper Reviewal 12

2.3.1 ERA Version 1 13

2.3.2 ERA Version 2 16

2.3.3 ERA Version 3 16

2.4 Truth Maintenance Systems 18

2.4.1 Justification-based Truth Maintenance System 18

2.4.2 Assumption-based Truth Maintenance System 20

2.4.3 Logic-based Truth Maintenance System 20

2.5 Summary 20

3 Specification **21**

3.1 Functional Specification 21

3.2 Prolog Prototype 22

3.3 Implementation Options for ERA 4 25

3.3.1 The architecture of ERA Version 3 26

3.3.2 The Different Options 28
3.3.3 The Decision to Reimplement 29

4 System Architecture and Implementation 31
4.1 The Foundation Layer 31
4.1.1 The Parser 31
4.1.2 The KnowledgeBase 33
4.1.3 The JTMS 34
4.2 The Application Layer 38
4.2.1 Application Layer architecture 39
4.2.2 Providing a Dynamic Interface 42
4.2.3 Controlling Execution and Knowledge Base State over User Backtracks 43
4.2.4 Supplementing Horn Clauses 44
4.3 Summary 44

5 Evaluation 45
5.1 Evaluation 45
5.1.1 Methodology 46
5.2 Results 48
5.2.1 Analysing Results: Questionnaires 48
5.2.2 Analysing Results: Reviews 51
5.2.3 Analysing Server Logs 53
5.3 Summary 53

6 Conclusion 55
6.1 Further Work 56
6.1.1 Improving the Foundation Layer 56
6.1.2 Examining the Reviewal Process 57
6.1.3 Integrating ERA with conference management software . . . 58
6.2 Related Work 59
6.3 Concluding Remarks 59

Bibliography 61

A Example Screens From ERA 4 65

B ERA Version 4 reviews 69

B.1 1 69

B.2 2 70

B.3 3 72

C Hand Reviews 75

C.1 1 75

C.2 2 76

C.3 3 77

D ERA Version 3 Reports 79

D.1 1 79

D.2 2 80

D.3 3 82

Chapter 1

Introduction

The review of research papers is a vital part of the scientific process, unfortunately there seems to be very little formal guidance available to novice referees on how to produce a good review. Some conferences provide referees with guidelines [5] or ”tip sheets” but very little research seems to have gone into the peer-reviewal process. This need has been identified by Professor Alan Bundy and has led to the development of various versions of the ERA system.

The ERA system or Electronic Referee Assistant is an expert system which assists novice reviewers by presenting a series of guided questions via a web interface. ERA has shown considerable promise so far even being used by students to produce reviews for the Informatics Research Methodologies course at the University of Edinburgh.

1.1 Motivation

The most consistent criticism of all the ERA systems is not allowing the user enough freedom in the order questions are answered. Users would find it helpful to be able to jump backwards and forwards between sections when writing a review. The current problem with this is that changes made in a previous section may invalidate inferences made by the program. Currently there is no way of dealing with this problem. A nave solution would be to recalculate all the inferences every time any of the sections are changed. While the size of the ERA system means that this is probably feasible it is a highly inelegant solution which would perform a lot of unnecessary calculation. Using a strategy like this would prevent ERA from scaling well if more questions were added or more levels of detail were added to the hierarchy.

A more sophisticated approach would be to keep track of which inferences depend on each other so that only inferences affected by the change would need to be updated. The approach to this problem adopted in this project is to build a Truth Management System into ERA which ensures that inferences made by the ERA system are consistent even if previously believed facts are retracted. This would eliminate the major technical hurdle in providing functionality to navigate back to previous pages or in fact to jump to any arbitrary part of the review.

1.2 Objectives

The primary objective of this project is to enable backtracking functionality in ERA through the exploitation of TMS technology. Taking into account design issues raised by the survey of current expert system shell technology and those raised during the development of a Prolog prototype JTMS, the best approach was to rebuild ERA from scratch.

The first objective is to create a cohesive framework for the ERA system which integrates an inference engine with a TMS. This is accomplished in the web scripting language PHP and provides a mini expert system shell which is uniquely suited to web programming tasks. This allowed the reimplementation of ERA using the framework and state of the art asynchronous web programming to incorporate the ability to back track to previous sections of the review.

Through using this new version of the ERA system and by comparing results collated from previous versions of the system, this paper goes on to examine the extent to which these expert systems improve the quality of review produced by inexperienced academics. The related question of how much of domain experts knowledge has been successfully captured by the expert systems is also considered.

Secondary objectives include enhancing the inference capabilities of ERA where possible and in doing so attempting to improve the usefulness of the system as a whole and improving the interface in order to provide greater usability.

1.3 Document Structure

Chapter 2 gives background information pertinent to the project; this includes a detailed overview of expert system technologies and the various expert system shells which implement them. It describes previous work that has been done in this area in the form of the two previous versions of ERA. Finally Truth Management Systems are described in detail.

Chapter 3 deals with the design specification of ERA version 4. It starts by describing the Prolog prototype which was developed and discusses the design decisions made in light of both the prototype and the material covered in chapter 2. The result is a detailed functional specification of the ERA system.

The fourth chapter describes the overall architecture of the final system particularly focusing on how specific functionalities could be encapsulated into reusable modules such as the knowledgebase module and the JTMS module. The development of ERA on top of this framework is then described including how server side and client side programming interact in order to produce a responsive system.

Chapter 5 describes the methodology used to evaluate the system including how the experiments were designed to maximise reuse of the data gathered from previous versions of ERA. The motivation for examining both the user experience and the quality of reviews produced is discussed before presenting the results.

The final chapter presents the conclusions which can be drawn from this project. The achievements and evidence gathered from evaluation are related to the objectives and hypotheses described in this section. This leads to a discussion of possible further work on ERA as well as detailing some related projects which could be relevant to future development of ERA.

The dissertation is then completed with a bibliography and a selection of appendices containing relevant material which is either too detailed or of too specialised interest for the main body of the work.

Chapter 2

Background

This chapter presents a literature review of work published on expert systems for informatics paper reviewal. The first section gives an introduction to expert systems themselves and the tools available for developing expert systems. The second section documents the progress made so far on developing an expert system for reviewing informatics research papers. The final section presents an overview of Truth Maintenance Systems (TMS) from the perspective of adding TMS to an expert system for informatics paper reviewal.

2.1 Expert Systems

Expert systems are in many ways the champions of the Good Old Fashioned AI (GOFAI) paradigm which was originally coined by Haugeland [26]. In many ways, despite philosophical claims that systems based on manipulation of facts and inference rules can't scale up to intelligent behaviour, expert systems are one of the most successful areas of AI. Expert systems are defined as;

> A computer program that simulates the thought process of a human expert to solve complex decision problems in a specific domain [4].

Essentially an expert system's role is to aid a human expert or someone who at least has familiarity in the field. They are typically designed to interact with the user by for example requesting more information in order to make a more informed decision. This can allow a human expert to complete tasks more quickly or often with a lower rate of failure since expert systems can keep track of things which may have been overlooked by the expert.

Expert systems have come a long way, originating as a variation of the Production System methodology proposed by Post [10]. A production system consists of a database and a set of rules, the system then simply iteratively selects rules and executes them adding the result to the database. The invocation of rules can be viewed as ”a sequence of actions chained by modus ponens” [10].

Expert systems emerged as a more practical results orientated subset of Production Systems. Where many Production Systems (PSG, PASII, VIS etc) [4] were used to model the human cognitive system (including effects of forgetting or making errors), Expert Systems however aim to display competent behaviour in a problem domain.

Perhaps the earliest successful Expert System was DENDRAL; this expert system was designed to generate plausible candidate structures for unknown organic compound based on data from a mass spectrometer. The project started as a series of papers delivered to NASA on an approach to generating all possible chemical structures [23]. However it quickly evolved into a fledgling expert system. The project led to the development of META-DENDRAL which could formulate new rules for DENDRAL and the system enjoyed some success being licensed by Stanford University for commercial use and demonstrating performance comparable to human experts [28].

Further research at Stanford, produced MYCIN, possibly the most famous of any expert systems. MYCIN was developed to ”provide diagnostic and therapeutic advice about a patient with an infection” [3]. MYCIN used many of the lessons learned when creating DENDRAL to produce a much more complex system which provided well over a decade of research at Stanford. Antibiotics were being widely misused at the time of MYCIN with only an estimated 33% of physicians making an effort to separate viral from bacterial infections [3]. Other problems with predicting different interactions between drugs made human diagnosis increasingly difficult. Evaluation of MYCIN suggested that success rates were similar to academic experts in the field and significantly more accurate than the actual treatment prescribed in the test cases or the treatments prescribed by a medical student [29]. MYCIN went on to substantially influence many other expert system based research including TEIRESIAS (a tool for defining knowledge bases), EMYCIN (a domain independent framework for building expert systems and many others.

2.1.1 Expert System Methods

Expert systems typically employ one of two reasoning modes; forward-chaining and backward-chaining. This section provides a brief explanation of these two methods along with the Rete algorithm which is a commonly used algorithm for improving the speed of forward-chaining.

2.1.1.1 Forward Chaining

Forward-chaining is a data-driven method where the facts in the knowledgebase are retained in working memory which is continually updated. Rules in the system are activated when certain conditions are present in working memory [15]. The rules are often referred to as condition-action rules where the left hand side of a rule represents a pattern which must match the facts in the working memory. If a match is present then the action of deleting or adding facts to the working memory is executed. An example of a rule might be expressed by the following psuedo-code:

```
IF student(X) AND has_money(X) AND handed_in_assignment(X)
THEN ADD can_go_to_pub(X)
```

This example rule represents the fact that if X is a student who is not broke and has completed his/her work then they are able to go to the pub. In this case the pattern which is being matched is the top line of the rule and the action is the bottom line of the rule.

In a forward chaining system an interpreter controls the application of the rules based on a recognise-act cycle. The interpreted first examines the working memory to determine if any of the rules match the data. This will typically use some form of conflict-resolver which will prevent the same rule being fired over and over again and also provide a strategy of which rules should be examined first. Other useful conflict-resolution strategies are firing rules on more recent items in working memory first which enables a single train of reasoning to be followed and firing rules with more specific pre-conditions before ones with more general preconditions, this allows exceptions to be dealt with before general cases [**?**]. Once a rule has been selected its actions are executed and the cycle starts again until there are no more matching patterns left.

2.1.1.2 Backward Chaining

Backward-Chaining is a goal driven approach, the system is presented with a goal, rules are then checked to see if they can support this goal. If a rule is found which has the goal in the THEN clause of the rule the facts in the IF clause are added as sub-goals and backward-chaining is performed on them recursively. This continues until the sub-goals can be proved by facts in the knowledgebase in which case the original goal is proved successfully or until no more rules can fire in which case the goal is unproved [8]. This process is essentially the mode of operation which prolog uses and a backward-chaining system can use the same set of rules as a forward-chaining one. In practice however different types of rules are more efficient to execute on one method than another [15].

Like a forward-chaining system conflict-resolution strategies are used to determine which rule will be tried first where multiple rules match the current goal. Prolog uses a nave method simply selecting the first rule it finds and backtracks to other rules if this line of reasoning fails. More sophisticated search methods can easily be applied to this methodology to prevent some of the typical problems with depth-first backtracking.

2.1.1.3 Forward vs. Backward-Chaining

The most appropriate method to use in an expert system is entirely dependant on the type of problem being approached. If a specific hypothesis is being tested, then backward-chaining will need to fire fewer rules than a forward-chaining system since no conclusions will be drawn from facts and rules which are irrelevant to the problem. Backward-chaining can however be wasteful when there are many possible ways of proving a goal, at worst all of these reasoning chains would need to be explored before one which matched the initial facts was found [**?**].

Forward-chaining is more useful when there are a large number of things that require proving from a small rule-set. This is especially true when there are several different rules which will draw the same conclusion. Another appropriate situation for the use of forward-chaining is when the final conclusion in not known in advance and the focus of interest is on what different facts can be deduced from the knowledgebase. This kind of exploratory work is particularly useful when working with a very large rule-set where the interaction can't be fully predicted by the designer [15].

2.1.1.4 The Rete Algorithm

The Rete algorithm is a method for improving the speed of forward-chaining at the expense of memory. The algorithm itself was developed by Charles Forgy at Carnegie Mellon in 1979 and works by building an acyclic directed graph into which facts are propagated. Any facts which reach the end of the graph cause a rule to be fired [14].

The network consists of two types of nodes, the first type have 1 input and 1 output and are constrictive nodes which only allow matching tuples to propagate. The second nodes type has 2 inputs and 1 output; these nodes connect output arcs from two different nodes merging the tuples from both the left and right incoming arcs into a single tuple of the outgoing arc [1].

The conditional part of a rule can be viewed simply as a pattern which specifies the attributes a tuple must have in order to fire the rule. Each condition becomes a 1-1 node which is attached to the graph below a series of ”entry” nodes which filter tuples by type.

The following example from the Drools manual illustrates the concept. Consider a complex condition such as ”For any person who has a dog that has the same name as that person’s sister’s cat” which could be the left hand side of a rule. The rule can be broken into three conditions all of which must be true for the rule to fire;

```
(1) (person name=person? Sister=sister?)
(2) (person name=person? Dog=petname?)
(3) (person name=sister? Cat=petname?)
```

The first rule specifies that for the rule to be applicable the two people involved must be sisters. The second and third rule express that the cat and dog must have the same petname, the dog must be owned by the person and the cat must be owned by the sister.

A Rete network for this problem can be seen in figure 2.1, the join nodes represent the fact that in the overall rule these three conditions are joined with an ”AND” relationship.

When a tuple from working memory is processed by the network any nodes which match are annotated with the tuple. This allows the inference engine to effectively remember partial matches to rules. This drastically reduces the number of comparisons that need to be made between the working memory and the rules-set reducing the computational complexity to O(RAC) where R is the number of rules, C is the average number of conditions per rule and A is the number of assertions in working memory.

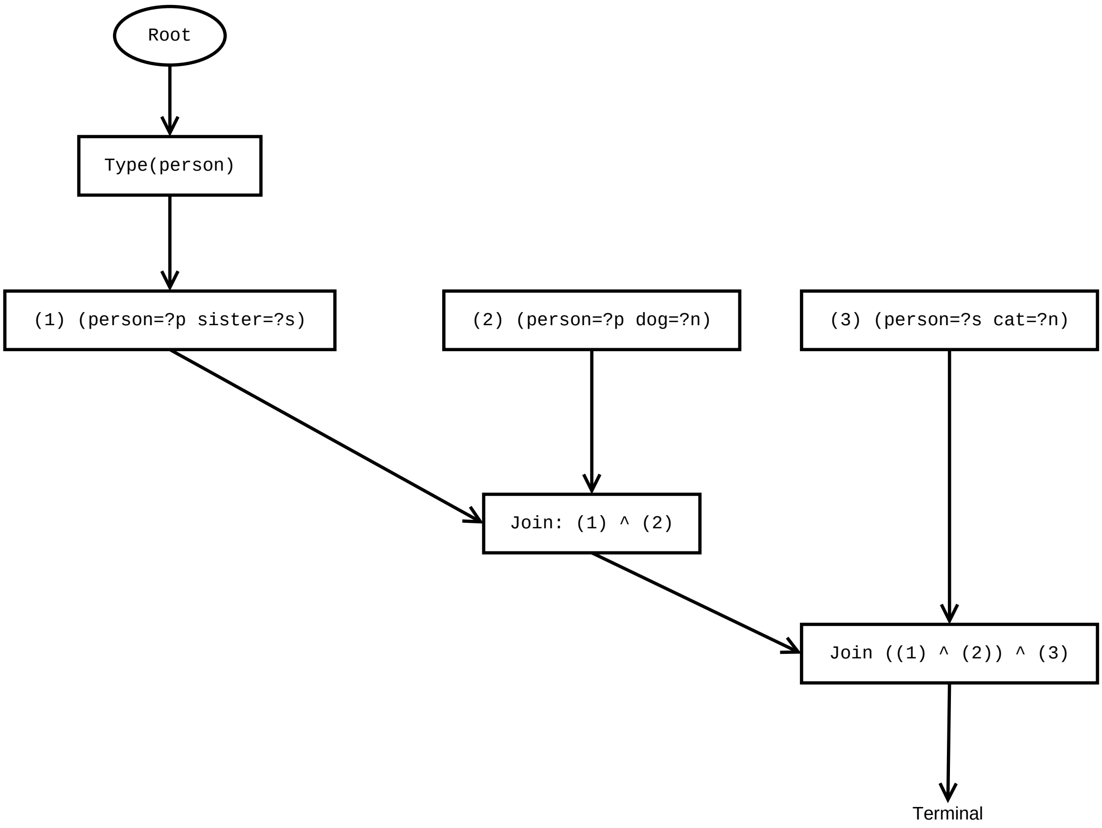


Figure 2.1: An example Rete network

2.2 Expert System Shells

Expert systems have become very popular in many different industries and it was natural for tools to be developed which would aid the rapid development of expert systems. There are now numerous commercial and free Expert System Shells which are normally small, lightweight languages used to describe facts and rules coupled with an inference engine, these shells can be used to very quickly build or prototype Expert Systems in any domain.

2.2.1 CLIPS family

CLIPS (C Language Integrated Production System) is an expert system shell which was developed by NASA's Johnson Space Center in 1985 in response to the high cost of ownership of the LISP based expert system tools available at the time and the difficulty integrating lisp with other languages at the time [9]. CLIPS has since been moved in to the public domain and is now maintained completely independently from NASA. The low-cost and liberal licensing terms of CLIPS has made it a very popular tool in government, industry and academia.

CLIPS supports forward-chaining with a Rete algorithm variant as well as several newer features such as procedural and object orientated programming. CLIPS also has built in procedures for code execution profiling and rule-verification.

Since CLIPS is now public domain software it has been used as the basis for a number of spin-off projects which specialise CLIPS for use in particular environments. WebCLIPS implements CLIPS as a CGI application which allows CLIPS to be used to develop expert systems which operate through a web-page interface [16]. Since the original version of CLIPS uses a text based interface this provides a far more user friendly way of interacting with the underlying expert system.

A new CLIPS based project PHLIPS fills a similar niche market but this time implements CLIPS as an extension to the popular server side scripting language PHP [2]. Although PHLIPS is in early development and only supports a select handful of CLIPS functions, PHP's well developed session and HTML templating support could lead to this being a viable competitor to webCLIPS.

JClips is an implementation of CLIPS which can be embedded within a JAVA application which is available under a public domain licence [25]. The use of JClips would allow an expert system to be developed using CLIPS which could then be delivered through a webpage via a java applet or distributed as a cross platform java application.

Numerous other specialisations of CLIPS exist allowing the core of CLIPS to be integrated with many other programming languages such as PERL, python and Ada. Since the expert system being studies is currently delivered through a webpage only the variants which would support this have been mentioned.

2.2.2 JESS

Jess is an expert system shell produced at Sandia National Laboratories, JESS was originally intended to be a clone of CLIPS in the java language [13]. Since then many features have been added which differentiates JESS from the CLIPS family most noticeably the addition of backward-chaining support as well as forward-chaining with the Rete algorithm and the ability to operate on and reason about Java objects. One key advantage of JESS is that it retains a degree of syntax similarity with CLIPS to the extent that many CLIPS programs will work in JESS without any modification.

2.2.3 Drools

Drools is a java implementation of the Rete algorithm which supports declarative programming [1]. Like JESS and CLIPS Drools provides support for object oriented programming. Unlike the CLIPS family and JESS Drools is a very lightweight system with far fewer features and a much smaller code-base. Drools also complies with the JSR-94 specification for API's to rule engines in java. This means that any application developed in Drools should be able to be ported to other java rule engines very easily.

2.3 Expert Systems and Informatics Paper Reviewal

There were several pre-cursors to the production of an expert system for paper reviewal. The first, an expert system for project supervision was developed in 1995 by John Tonberg [22] as part of a 4th year project at the University of Edinburgh. Another precursor was an expert system to ”Advise on performing AI Research” which was developed by Varvara Trakkidou as part of her Msc. Project [27].

Both of these projects shared some common ground with the task of paper reviewal in that the user would enter information and the result would be some form of appraisal of the work. In the case of the second project this would be advice on how to improve study habits.

Both of these projects were implemented in CLIPS, a similarity with the other paper reviewal systems mentioned which highlights their nature as pre-cursors to the ERA system.

2.3.1 ERA Version 1

The first expert system for informatics paper reviewal was developed by Massimo Caporale [7] as his final year Bsc project at the University of Edinburgh in 2003. ERA stands for ”Electronic Referee Assistant” and was implemented in CLIPS. The system itself worked by presenting the user with a set of high-level questions about the paper to be reviewed, if the user was unable to answer any of these question the system would decompose the question into a set of more specific questions. This process would continue until a level was arrived at where the user felt comfortable answering the questions. The system would also attempt to evaluate the paper by analysing the user input and generating a score. This score is then used to determine if the paper should be accepted for a conference or not along with a coarse grained scale of confidence (i.e. clear accept, weak accept).

This hierarchical structure of questions was based on Professor Alan Bundy’s Informatics Research Methodologies (IRM) [6] lectures and the notes to CADE-12 referees [5]. In order to finalise the different questions used a large number of referee forms for various informatics conferences were analysed. The final categories used in the ERA system were Validity, Significance, Originality, Relevance, Presentation, Confidence and Overall. An additional comments section was added to the form at the advice of Alan Bundy to allow a more detailed description of the strong and weak points of the paper being reviewed.

Massimo Caporale represented the reviewal process as a decision tree-like structure. Each of the major sections were branches from the Overall section and had child nodes representing finer grained divisions within the category. At each node the user could either give a score, or if the user wasn’t confident enough to give a score could traverse down the tree one level and answer more specific questions. At each node a weighted average of the scores of child nodes was used to calculate the score for levels unanswered by the user. The final score was therefore a weighted average of all of the questions the user answered. The exact weight assigned would depend on different rules for different sections of the review.

The system itself was designed in quite a modular way, presumably to facilitate the addition and removal of different sections to the review. Although this possibility wasn't explored in the work this would allow different conferences to customise ERA so that reviews are conducted along the axis which are most important to the conference. For example the Originality section might not be relevant when all of the papers being looked at are literature reviews.

The ERA system was evaluated along two different axes, the first based on the quality of review produced using the system and the second based on user satisfaction with the system. Although the first dimension is arguably the most important, given that if an expert system does not produce better results than an unaided user it can be viewed to have failed. However in order for any system to be widely used it is important that the users are confident using the system.

To evaluate the quality of the review six papers were reviewed by 5 different referees. Each paper was reviewed twice, once using a conventional paper form and once using ERA. The paper forms were actually submitted to a conference as part of the conference's actual reviewing process. The final decision of the Program Committee of the conference was used to evaluate the performance of ERA.

There are several problems with this experimental methodology. Most obviously the paper reviews performed as part of the experiment would directly influence the results that ERA would be compared to. Each paper in the conference was evaluated using three reviews, one of which was a paper review written as part of the evaluation. This makes the paper reviews used in the experiment inadequate as a control group and highly biases the results of the experiment in the direction of the paper reviews.

Secondly the referees used in the experiment were reviewing papers for a conference. This suggests that the referees are experts in the field and whilst it is useful to see how using ERA affects domain experts the stated target of the ERA system were inexperienced referees. Testing a range of referees from totally inexperienced (1st year students) all the way up to very experienced (lecturing professors) would have given a much better overview of ERA's performance.

ERA's performance was decidedly poor in this experiment. The system predicted the wrong outcome in all of the cases examined. This was compared to the paper form reviews which predicted the correct outcome 66% of the time. This evidence suggests that using ERA was producing worse reviews.

Caporale identified that this problem with the overall score being assigned was most likely due to the weighted averages not being correct. Indeed he even points out that the rules were developed by trial and error, given that Caporale was an inexperienced referee himself it is unlikely that he would be able to hit upon the optimal rule in any case.

Despite the disappointing results of ERA in this experiment it isn't totally representative of the quality of reviews produced by ERA. The experiment focussed solely on the final score ERA assigned to the paper and didn't take into account the actual body of the review itself.

In order to provide some kind of qualitative analysis of the body of the reviews all of the reviews were appraised by Alan Bundy in the role of domain expert. Whilst paper reviews tended to be longer and by that virtue somewhat more significant it was inconclusive as to which method produced the better review.

A blind study using a range of referees with different levels of experience would be an ideal way to get more quantitative results from the actual body of the review. Each referee could produce several reviews some using ERA some without. The reviews could then be allocated to domain experts to evaluate preferably with a score. The results could then be grouped by referee in order to see if that individual person produced better or worse reviews using ERA. Unfortunately experimentation on this scale is difficult to organise, especially as it relies on finding a panel of domain experts who have time to participate.

ERA fared much better in the user satisfaction section of the evaluation. Each reviewer was asked to fill in a questionnaire about ERA. All but one of the reviewers found the system helpful with over half stating they would use it again and recommend it to others. In general users felt that the system needed a GUI (Graphical User Interface) to make it easier to use. Most reviewers though the questions were asked in a sensible order and were appropriate questions although a few expressed concerns that some of the questions were too ”black and white”. The only other real criticism was that the structure of the questions was completely rigid, and that it was impossible to jump backwards and forwards between different sections or tackle them in a different order.

ERA version 1 was an important piece of exploratory work that whilst not quite managing to be a successful system made considerable progress towards this goal. Although very little Expert System technology was used in this version of ERA it laid down some foundations and uncovered some key issues in this field which were addressed in subsequent versions of ERA.

2.3.2 ERA Version 2

In response to user feedback requesting a GUI for ERA Massimo Caporale was able to develop ERA 2 under an ESPRC grant. ERA 2 was developed in the webCLIPS variant of CLIPS which allows the user to interact with the program through forms on a web page. Unfortunately the development of this version of ERA has not been documented in a dissertation so it is unclear if ERA 2 has been experimentally evaluated by Caporale. The interface is certainly far more user-friendly and many of the sections have been altered to give more assistance to the user such as the addition of help buttons in certain sections.

ERA 2 was evaluated as a comparison to ERA version 3 by Brian Hutchinson. This is discussed in more detail in the next section of the document.

2.3.3 ERA Version 3

The third version of the ERA system was developed by Brian Hutchinson [20] as his Msc project at the University of Edinburgh in 2004. It had been identified that ERA although being developed in an expert system shell didn't really take advantage of the inference capabilities provided. In fact the only inference being performed was essentially weighted averages used to calculated section scores and overall scores. ERA 3 was developed to explore how adding inference capabilities would benefit the system.

The approach taken in ERA version 3 was to tailor the questions the system asked to the type of paper being reviewed. To this end an extended summary section was added which determined the content of the paper asking questions such as:

```
"Does this paper describe a new technique?"
"Does this paper describe an adaptation of an old technique
to a different domain?"
"Is this paper a review of work in a domain?"
```

As well as this ”type” classification the nature of the hypothesis of the work was examined. There is a big difference between work targeted at a specific hypothesis and exploratory work where the goal is the identification of a hypothesis.

These results were used to infer what sorts of questions to ask in the different stages of the review as well as tailoring the questions to include references to the paper itself. For example if a paper was identified in the summary section as being about ”Foo's technique” a question in the significance section might read ”How widely applicable is Foo’s technique?”.

The idea is that tailoring the questions would produce a better review by only asking questions which are relevant to the type of work being reviewed as well as making the system easier to use by including information that could help the user within the question itself.

The resulting system contained approximately three times as many inference rules as the ERA 2 system taking the total number of rules up to 148. This added complexity makes the use of expert system technology a lot more justified. Within the ERA1 and ERA2 systems it seemed that using an expert system shell was overly complicated in many ways since the same effects could be produced more simply using conventional software engineering. With the added inference capabilities the development time saved by using an expert system shell was likely to have been significant.

Again ERA 3 was evaluated along two dimensions, the quality of review produced and the user’s satisfaction with the system. In order to test the quality of reviews ERA 3 was used as a tool in Professor Alan Bundy’s Informatics Research Methodologies [6] course. The students taking the course were required to submit at least four reviews which contributed to the final mark of the module. The students were given the option of either using ERA version 3 or using a traditional paper form.

Statistical analysis of the grades achieved by the different students using ERA version 3 and the traditional form yielded a 95% confidence that the students using ERA achieved better grades than those using the traditional form. This is a very good result for the system and provides concrete evidence that using the system improves the quality of reviews produced by inexperienced reviewers.

In order to evaluate the user satisfaction of ERA version 3 a questionnaire was designed which asked the users to grade different aspects of the system. This was compared with a similar questionnaire collected from ERA version 2 users. In all of the criteria ERA version 3 was rated by the users as at least as good as ERA 2 and in some significant categories it scored much higher. Two of the most significant differences were that users of ERA 3 strongly agreed that the ratings recommended by the system were satisfactory whereas users of the ERA 2 system were undecided. Similarly users of ERA 3 agreed that the system provided useful answers to the questions they had which was also undecided amongst ERA 2 users.

In all ERA version 3 seems to be a much stronger candidate for an informatics paper reviewal expert system. It outperforms ERA 2 across both dimensions of analysis consistently and evidence suggests that it improves the quality of reviews of inexperienced reviewers. Several users commented that it would be helpful to be able to answer the questions in a different order, or to return to previously answered questions and edit the responses, a facility which ERA currently doesn't support.

Another improvement identified by Hutchinson is that currently certain characters are not allowed to be entered into the web forms since they interfere with the programs operation. This is a minor software engineering issue which could easily be added in to the current system.

2.4 Truth Maintenance Systems

In any kind of inference system it is possible for long chains of reasoning to be established with many intermediate facts depending on other facts and rules within the knowledgebase. One potential problem is that when any part of the knowledgebase is refuted or retracted there is no way to know how this will affect the rest of the knowledgebase. The brute force solution would be to re-derive all the facts in the knowledgebase based only on what is known to be true. Clearly this is a very inefficient approach to the problem. In order to address this, Truth Maintenance Systems were developed as an augmentation to standard inference systems.

2.4.1 Justification-based Truth Maintenance System

The first Truth Maintenance System (TMS) was proposed by Jon Doyle at MIT [12] and later became known as a Justification-based TMS (JTMS), although Doyle himself now prefers the term Reasoning Maintenance System. The JTMS works by maintaining what is known as a dependency network. Briefly each fact or sentence in a knowledge base is represented by a sentence node. All of the sentence nodes receive arcs from justification nodes.

Each sentence node also has an associated label, IN or OUT, to represent whether the sentence is currently believed or not as well as a list of justification nodes which support the sentence and a list of justification nodes which are supported by the sentence. The justification nodes themselves only possess a label, IN or OUT representing the current belief state of the node. This structure puts in place the necessary infrastructure to provide justifications for conclusions drawn: the TMS can simply follow back the list of justifications supporting a given sentence. It also supports dependency-driven backtracking, that is, ”the justification of a sentence, as maintained by a TMS, provides a natural indication of what assumptions need to be changed if we want to invalidate that sentence” [21]. As further advantages, JTMSs also support default reasoning (assuming default facts in the absence of firmer justification) and help to recognise inconsistencies in the knowledge base.

To illustrate consider the diagram in figure 2.2; here there are four sentence nodes, two of which are assumed to be true (humid and windy), one which is assumed false (thunder) and one whose truth is initially unknown. There are rules for labelling the nodes of the network. Firstly a justification node is IN if all of the nodes in its IN-LIST (its input nodes) are IN. In this case the justification node Humid ˆ Windy => Rain is IN since both Humid and Windy are IN.

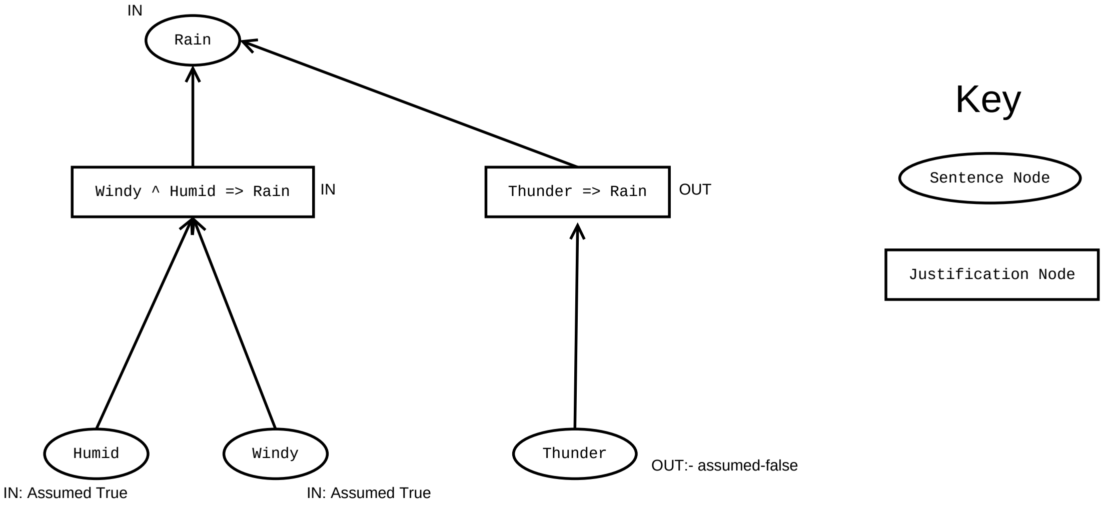


Figure 2.2: An example JTMS network

Next, a sentence node is IN if either it is assumed IN or else it has an input from at least one justification node which is labelled IN. In this case the fact Rain is IN because the justification node Humid ˆ Windy => Rain is IN. It is simple to output the justification for any derived fact by outputting its justification nodes. For example in this case the system might return something like:

```
"It is going to rain because it is humid and windy."
```

If both of the justification nodes had been IN the system might have returned output like:

```
"It is going to rain because it is humid and windy and there is thunder."
```

The network also prevents unnecessary calculation. For example consider the case where Rain is justified by both of the justification nodes and there are several other inferences which depend on Rain. If Thunder is retracted the fact Rain is still IN because it is justified by Humid ˆ Windy => Rain. This means that none of the facts deeper in the inference chain need to be recalculated when Thunder is retracted

2.4.2 Assumption-based Truth Maintenance System

Johan de Kleer expanded on the JTMS with an Assumption based TMS (ATMS) [11]. The ATMS system was developed to address the fact that when sentences are retracted in a JTMS the dependency network needs to be relabelled, which depending on the depth of the sentence being retracted could amount in considerable computation. The idea behind the ATMS is that rather than maintaining one belief context, all belief contexts are held at once along with information about which sets environments (sets of assumptions) must hold in order for the context to be valid. Thus instead of labelling each node with a simple IN or OUT, it is labelled with the set consisting of all environments which justify the node. The advantage of this is that ”since all environments are simultaneously available, it is not necessary to enable or retract sentences, and thus re-label the network” [21]. Instead all that needs to be done is to change which environment is currently being examined.

2.4.3 Logic-based Truth Maintenance System

One problem with both of the ATMS and the JTMS is that the system has no context of the meaning of any sentences. Thus it is perfectly possible for either system to hold p as a justification for p; ideally the inference engine should stop this happening however with very complex knowledge-bases where some facts may have many justifications this situation can arise. The Logical-based Truth Maintenance System can recognise some of the propositional semantics of sentences in order to address this problem.

2.5 Summary

This chapter has given an overview of expert systems technology including describing several available expert system shells. The history and sginificant developments of the ERA system have been discussed and finally a variety of Truth Maintenance Systems have been presented.

Chapter 3

Specification

This chapter first discusses the functional specification of ERA version 4 by specifying all of the required capabilities and justifying their inclusion in the new version of the program. The development of a Prolog prototype JTMS and the influence this had on the design specifications is then discussed. To conclude, a final low-level specification is detailed along with justifications of design choices and why different alternatives were not explored.

3.1 Functional Specification

ERA version 4 needs to allow a user to review a paper by interacting with a series of web pages. These web pages contain standard HTML form components which collect the user's answers to various challenges generated by the system. Through this interaction ERA should be able to draw inferences about the paper being reviewed and based on these inferences guide the user through a series of questions which comprehensively analyses the paper under review. In particular the prompts ERA presents to the user should be dynamically generated based upon the inferences which the system has already drawn from previous parts of the dialogue. As in previous versions the chief mechanism for achieving this is a forward-chaining expert system.

ERA version 4 is required to address one of the observed shortcomings in previous ERA systems by allowing users to backtrack through previously completed sections of the dialogue in order to change their responses. This should increase usability greatly and effectively allows a user to tackle the sections of the dialogue in an arbitrary order. In order to preserve consistency in the knowledge base this functionality is provided by a JTMS module built into ERA version 4. Since jumping around in the dialogue will inevitably result in some of ERA's previous inferences being retracted it is clear that the inferences made by a forward-chaining expert system alone would soon become inconsistent. The JTMS also provides the capacity for more sophisticated reasoning by increasing the potential for the system to give justifications for the inferences it has drawn.

Another key drawback of the previous versions of ERA is that the inference performed is relatively shallow, with inference chains typically never extending through more than 2 or 3 derivation steps, where a derivation step is defined as an application of a rule. The result of this is that the inferences performed often seem relatively obvious to the user and don't really change the way the user would approach the review. It is hoped that deepening the inference capabilities of ERA will lead to non-obvious inferences being drawn which can both improve the quality of reviews produced and possibly even yield some insights into how experts themselves perform the reviewal process.

3.2 Prolog Prototype

The prototype JTMS was built around a forward-chaining mechanism in Prolog. This required very little effort and involved only minor alterations to Prolog's existing backtracking mechanism. The JTMS itself was implemented as described in section 2.4.1. The resulting system is designed to run interactively within the Prolog command-line interface. The user commands available are:

make_horn(+Horns, +Head)

This predicate creates a horn clause in the knowledgebase. Horns can be a list of predicates with an arbitrary number of arguments or atomic facts. The head can be a predicate or atomic fact. The semantic meaning of the created horn clause is:

```
Horn1 ˆ  ˆ HornN => Head
```

This can be used to either assert rules by including un-instantiated variables in the Horns and Head or can be used to assert facts by using a list containing the atom true as the Horns. So for example the Horn clause 'rule' which states every significant and original paper should be published; "forall(X).significant(X) ˆ original(X) => accept(X)" would be entered as;

```
make_horn([significant(X), original(X)], accept(X)).
```

Whereas the fact that significant(myPaper) is true would be entered as:

```
make_horn([true], significant(myPaper)).
```

This command also invokes the JTMS to add appropriate sentence nodes to the knowledgebase. Facts which are asserted in this way are given a justification of true which denotes that they are assumptions which should be believed in the absence of any other justification.

inference

This command invokes the forward-chaining mechanism and results in any conclusions which can be drawn from the currently believed facts and rules being added to the knowledgebase. This command also makes calls to the JTMS to ensure that the appropriate justifications are attached to any facts which are added as a result of the forward chaining.

retract_horn(+Hornclause)

This predicate can be used to retract rules or facts from the knowledgebase. The argument is given in the form horn(Horns, Head) and as with the command make_horn/2 it can be used to retract either facts (by setting the Horns argument to [true]) or rules. This predicate also invokes the JTMS to retract any facts which become unjustified after the retraction and to update the justification sets of the JTMS network appropriately.

write_nodes

This displays the current nodes in the JTMS network

load_test_kb

This command loads the test knowledgebase which was used during development of the Prolog prototype. This contains a simple toy example which was used for debugging purposes.

The following shows an example dialogue from the prototype. This demonstrates the inference procedure and what happens when a fact is retracted. Note that when tweets(tweety) is retracted the JTMS maintains the consistency of the knowledgebase by retracting all of the facts which are solely justified by Horns containing tweets(tweety). Note that in Prolog's internal representation the variables are represented by an underscore and index such as _196. In the actual file these were given the human readable label X but prolog translates this for its internal use.

```
| ?- [’c:/jtms/one.pl’].
```

```
{consulting c:/jtms/one.pl...}
{loading d:/data/prolog/library/lists.ql...}
{loaded d:/data/prolog/library/lists.ql in module lists, 0 msec 27688 bytes}
{consulting c:/jtms/inference.pl...}
{c:/jtms/inference.pl consulted, 0 msec 4488 bytes}
{consulting c:/jtms/jtms.pl...}
{c:/jtms/jtms.pl consulted, 0 msec 4616 bytes}
{consulting c:/jtms/misc.pl...}
{c:/jtms/misc.pl consulted, 0 msec 1720 bytes}
{consulting c:/jtms/data.pl...}
{c:/jtms/data.pl consulted, 16 msec 816 bytes}
{c:/jtms/one.pl consulted, 31 msec 40664 bytes}

yes
| ?- load_test_kb.
adding [bird(_196),alive(_196)]=>flies(_196)
adding [bird(_196),alive(_196),penguin(_196)]=>swims(_196)
adding [tweets(_196)]=>alive(_196)
adding [true]=>bird(tweety)
adding [true]=>tweets(tweety)
adding [true]=>penguin(pingu)
adding [true]=>alive(pingu)
adding [true]=>bird(pingu)

yes
| ?- inference.
adding [true]=>flies(pingu)
adding [true]=>swims(pingu)
adding [true]=>alive(tweety)
adding [true]=>flies(tweety)

yes
| ?- retract_horn(horn([true], tweets(tweety))).
The JTMS found the following nodes which may be justified by the retracted
sentence
```

```
node(alive(tweety),[[tweets(tweety)]],true)
The JTMS found the following nodes which may be justified by the retracted
sentence
node(flies(tweety),[[bird(tweety),alive(tweety)]],true)
The JTMS found the following no nodes which may be justified by the retracted
sentence
Removing horn([true],flies(tweety)) from the knowledge base
Removing horn([true],alive(tweety)) from the knowledge base

yes
| ?-
```

While this prototype exploited Prolog's built-in type matching and search facilities (and as a consequence was developed very rapidly), it did however give a good idea of the types of internal objects which would be required to build a JTMS in another language as well as the types of functions that would be needed. Indeed, the majority of the non-API functions were kept in the final version of the JTMS, although, since they were implemented in a very different language, their contents do not appear similar. The notion of operating on objects which essentially have two properties, a list of horns and a head was fundamental in the architecture of both the final inference engine and the final JTMS.

The facilities which Prolog provided would also have to be re-implemented in the final solution. In particular, providing pattern-matching and forward-chaining via modus ponens would be key first stages in implementing the inference engine without which the JTMS wouldn't be viable. The result was that a loose ordering of implementation tasks was established based on the prototype: first the basic rule and fact objects would have to be implemented, followed by a unification algorithm to match terms which contained variables to instantiated versions, which would then allow the implementation of forward-chaining modus ponens. Only once all of these prerequisites were completed could the JTMS itself be implemented.

3.3 Implementation Options for ERA 4

There were essentially three options for building a JTMS into the ERA system, each having distinct advantages and disadvantages. Two of these options reused the existing ERA code-base (which is described below) and one option involved a complete re-implementation of the system. The design considerations involved with each architecture are described in this section, as well as the reasoning behind the final decision to completely re-implement the system.

3.3.1 The architecture of ERA Version 3

ERA is built using the webCLIPS (see section N.N) expert system shell, which is a descendent of the original CLIPS expert system shell. Information is presented to the user in a series of web-browser ’screens’, the content and order of which is a direct result of an inference. Hence, the data for each screens, including the HTML mark-up, is included within the conclusions part of the system’s inference rules. Further state information is carried from screen to screen (and hence, from inference to inference) using hidden HTML form elements in the mark-up. A typical rule from the previous ERA code-base might be:

```
(defrule MAIN::summary2c_nat_system
    (declare (salience 50))
(file ?file)
?del1 <- (summary2_nat start)
?nat_system <- (nat_system)
?del2 <- (nat_name)
=>
(retract ?del1)
(retract ?del2)
(printout t "
  <h2 align=center>Natural System</h2>
    <hr>
    <p>
      <form name=MyForm action=http://www.inf.ed.ac.uk/cgi-bin/courses/irm
      /era/webclips.exe method=post>
<input type=hidden name=fact value=\"(ScreenName (ScrnName ERA))\">
<input type=hidden name=fact value=\"(factgroup " ?file ")\">
<input type=hidden name=fact value=\"(summary3 start)\">
<br>
<b>You said that the technique models a natural system</b>
<br>
What natural system does the technique model?
<br>
<input type=hidden name=factname1 value=nat_name>
<textarea name=factvalue1 cols=60 rows=4 wrap=hard onKeyPress=
\"return letternumber(event)\"></textarea>
<br><br>
<input type=submit value=Continue>
       </form>
      </p>
")
(save-facts ?file)
)
```

This rule doesn't provide any inference about the paper beyond a simple if-then decision which causes the rule to fire if the paper describes a natural system. The data model, presentation layer and actual inference are hopelessly entangled within rules such as these. In fact, examination of the previous ERA code-base reveals that the purpose of the vast majority of the rules is to control program flow - they do not provide any inference about the paper under review at all. Indeed, the only inferences which are not controlling program flow are those in the various calculator methods which are invoked if a user clicks the help button.

For instance one of the prompts in the previous version of ERA is ”How original is this paper?” The user can then either select from 4 options to rate the originality ranging from ”Trailblazing” to ”It’s all been said many times before” (the selection of which effectively assigns a ’score’ to the paper’s originality) or else click the help button. Clicking the help button results in a screen which essentially asks how many similar papers to the one under review the user is aware of, paraphrased according to what type of paper it is. The calculator rules are used to increment a running total based on the user’s answers to the questions asked in the ”help” section and then apply simple cut-offs to determine the recommended originality rating (and hence, score).

These fundamental architectural problems mean that there is a very serious trade-off involved if the code-base is to be reused. The use of the same inference system to control the program flow, presentation and actual inference about the paper has led to an incredibly intricately tangled code-base which means that trying to build on the code may take more time than re-implementing the code from scratch particularly since the number of rules which actually perform inference about the paper is very small (a cursory examination of the code-base suggests in the order of 10

3.3.2 The Different Options

Bearing the existing architecture of ERA in mind, we now consider the three options for implementing version 4 of the system:

3.3.2.1 Three Layered Architecture (webCLIPS, PHP/Perl, Prolog)

In this architecture, the existing ERA system, running on webCLIPS, would be modified to run in ’cookies’ mode. This causes ERA to store all current rules/facts in an external file. This file could then be parsed using a scripting language such as PHP or Perl and fed to a web-enabled Prolog JTMS (based on the prototype) which would overwrite the file written by webCLIPS to reflect the results of running the truth-maintenance procedures over the current knowledge base. The PHP/Perl layer would also be responsible for controlling program flow.

The advantages of this method are that it reuses code from both the Prolog prototype and the previous version of ERA. However it requires working on 3 disjoint code-bases and would lead to highly un-maintainable code since program flow would be split between all of these code-bases.

3.3.2.2 Interface PHP JTMS with the existing ERA system

In this architecture a separate JTMS will be built in PHP based on the Prolog prototype. webCLIPS will be run in cookies mode so that all visible facts/rules are written to a file in-between conscutive HTML pages that are presented to the user. This will allow the facts and rules to be parsed to build the JTMS network. The JTMS would then rewrite the file based on the results of JTMS analysis before the next webCLIPS screen is evoked.

In this case the webCLIPS code would need to be embedded within PHP pages which control program flow to ensure that webCLIPS doesn’t try to read in the facts/rules until after the JTMS has updated them.

This has the advantage of reusing the previous ERA code and allowing the JTMS to be implemented separately from the rest of the code-base. However it would also require reverse engineering the format of the ERA cookies file and it would be very difficult to guarantee the integrity of this file in all possible program states. The result would be a relatively fragile system.

3.3.2.3 Re-implement system using modular and object-oriented PHP.

In this architecture a simple forward-chaining inference engine would be built as a separate module capable of handling at least the sub-set of logic used by the existing ERA system. Current CPU and memory resources, and the relatively small knowledge-base in this project, mean that performance shouldn't be an issue but if so the Rete algorithm could be implemented.

The JTMS would be implemented as a separate module, again creating a reusable component. Both the inference engine and the JTMS would have a set of well-defined APIs which would be used to build general expert systems.

In this case a central control-flow module would interface between the JTMS and inference engine modules to re-implement the logical rules found in ERA version 3. The actual input and output would be handled by standard HTML forms and could be styled using CSS to create a polished web application.

A further advantage of this approach is that state information can be stored either in a database (such as mySQL or, more simply, as flat files) or in PHP session variables which would allow the addition of a more sophisticated user interface. For example all screens could have a side-bar with a list of previously made decisions. The user could backtrack on one of the decisions in the sidebar simply by clicking on it.

This architecture would mean that the code would be much cleaner and highly maintainable. The application could look slicker since the program logic can easily be separated from styling information (via CSS) and only one language is used in the code-base. The main disadvantage of this method is obviously that it is a lot more work since it doesn't reuse any of the existing ERA knowledge-/code-base except for the few inference rules which are actually relevant to the paper itself.

3.3.3 The Decision to Reimplement

Based mostly on the problems with the existing ERA code-base it was considered that a more modern approach to this application would be beneficial and while it might be slightly more work, the poor design of any of the other solutions could lead to problems which would be so difficult to trace that they would quickly eat away any time benefit gained by reusing the existing ERA code-base. The highly reusable and clean design of a properly implemented object-oriented system also contributed the decision to take the third option, namely to re-implement ERA from scratch in PHP.

The ease of propagating state information provided by PHP’s built-in session support also means that it would no longer be necessary to pass state information through hidden HTML form elements, making the system much easier to interact with from a programming perspective. A PHP session is essentially a number which is given to each user; this number is propagated either through cookies, if the user’s system allows it, or else by appending the session name and id to the URL of the page. This number points to a set of variables on the server. Since the session ’follows’ the user on each page they visit, this results in a set of variables which are always available to the programmer throughout the user’s visit. These variables can then be used to store state information between pages.

Chapter 4

System Architecture and Implementation

The ERA v4 system is implemented from the ground-up in the PHP server-side scripting language. The system itself is split into two main layers. The first, the foundation layer, handles the inference, reasoning and truth maintenance operations. Above this, the application layer provides the program flow and user interaction logic which completes the system. Each of these layers will be described in this chapter.

4.1 The Foundation Layer

The foundation layer comprises several separate components, as shown in figure 4.1. The knowledgebase is initialised by passing it a set of facts and rules. This can either be done programmatically through the defined constructors or using the separate parser which allows facts and rules to be entered in a more human readable format. The knowledgebase is a stand-alone module which can perform forward chaining inference but can also control the separate JTMS module if it is present. This section describes each of these modules in turn in the order encountered during typical execution.

4.1.1 The Parser

The parser is a simple module which takes a sequence of entries in the mini-language described below and converts these entries to the knowledgebase's internal rule and fact objects. The parser itself uses a tokenizer to split the input into individual lines delimited with a semi-colon, and then uses regular expressions to search for patterns

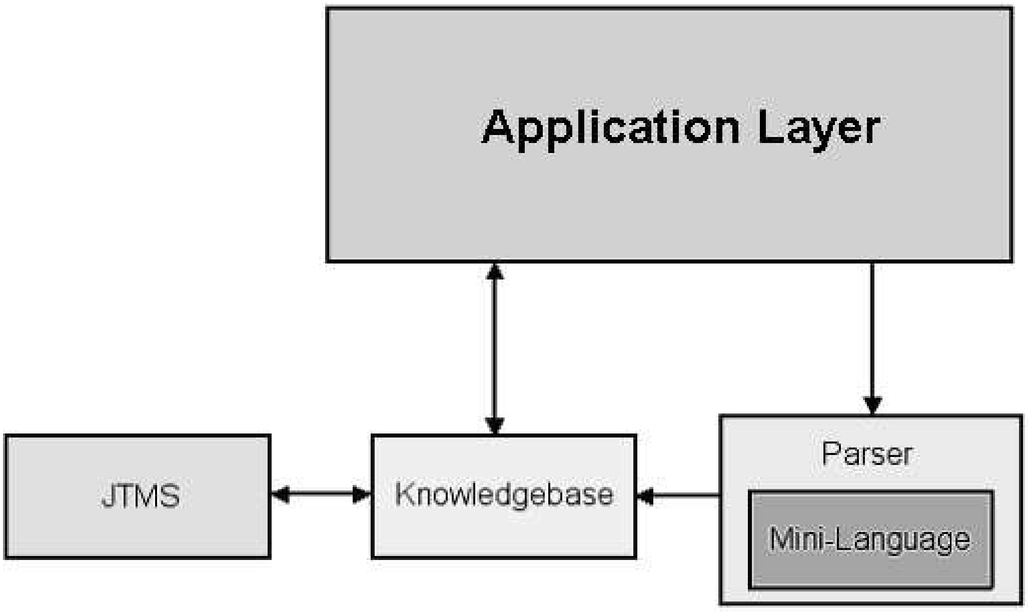


Figure 4.1: The foundation layer architecture

which correspond to rule or fact entries. Any lines which do not fit one of these patterns are simply disregarded (thus allowing for the insertion of comments into the file). While it would be advantageous to have a more sophisticated parser, which could raise warnings when encountering malformed facts or rules, it was considered that these extra diagnostic facilities would not justify the extra time spent on this relatively minor component.

4.1.1.1 Mini Language

The mini-language recognised by the parser consists of a single text entry (that is a string containing alphanumeric characters and punctuation) with each line of input terminated with a semi-colon character. Each line may contain either a fact or a rule.

Literals, used to describe static entities in the domain in question are recognised as a string of alphanumeric characters. So, for example, the literal ready is most sensibly represented by the string ”ready”.

Variables are represented by a string of alphanumeric characters delimited by asterisk ”*” characters. Thus a variable A would be best represented as the string ”*A*”.

Terms are represented by a finite string of alphanumeric characters followed by a set of arguments delimited by braces ”(” and ”)”. Arguments are a finite comma-separated string of literals or variables. A line consisting of a term all of whose arguments are literals is interpreted as asserting the corresponding fact.

Rules, which must correspond to horn clauses, are represented by a finite sequence of terms followed by the string ”=>”, which represents the implication operator, followed by exactly one term. Thus a hypothetical rule which asserts that if a paper is outstanding and is about web programming then it should be accepted for publication would be represented by the composite string ”subject(A, webprogramming) ˆ outstanding(A) =¿ publish(A);” (with the universal quantification over all variables being implicit).

White-space of any kind is accepted in the input and is stripped before parsing. This allows the lines to be entered in a formatted manner (using tabs or carriage returns for example), which greatly increases code readability and ease of use.

The table below shows a general schema for the mini-language:

Feature	Schema	Example
Line	Fact/Rule;	Subject(myPaper, bioinformatics);
Term	Name(arg1,,argN)	Subject(myPaper, bioinformatics)
Argument	Literal / Variable	*Paper*
Rule	Term1 ˆ ˆ TermN > TermN+1	Subject(*paper*, *subject*) ˆ subject(*Journal*, *Subject*) => relevant(*paper*, *Journal)
Literal	name	myPaper
Variable	*name*	*Subject*

4.1.2 The KnowledgeBase

The knowledgebase module is responsible for both storing facts and rules as would be expected but also performs the inference which ERA relies upon. The module has been designed so that it can be used as a stand-alone component as well as in tandem with the JTMS module or indeed as part of the ERA system.

The knowledgebase performs forward chaining via modus ponens on a subset of first-order predicate calculus corresponding to Horn clauses. Examination of previous versions of ERA have not revealed any uses of non-Horn clauses that can’t be trivially converted to Horn clauses. A common example would be rules of the form:

```
term1 ˆ  ˆ termN => assertion1 ˆˆ assertionM
```

These cases can simply be rewritten as a set of M horn clauses as follows;

```
term1 ˆ  ˆ termN => assertion1
term1 ˆ  ˆ termN => assertion2

 term1 ˆ  ˆ termN => assertionM
```

Based on these observations it was decided that restricting the inference engine to handling only Horn clauses would still be adequately expressive for the ERA system.

The forward-chaining mechanism built into the inference engine performs modus ponens using a simple breadth-first search. Since ERA does not have a very large knowledge base it was considered unnecessary to implement a more efficient form of search. The forward-chaining system will also unify variables using the unification algorithm described by Maxim [24].

When the knowledgebase module is used in conjunction with the JTMS module there needs to be a way for the knowledgebase to perform operations on the JTMS such as calling for the JTMS to create a new node or informing the JTMS that a node has been retracted. In order to do this the programmer registers an instantiation of the JTMS module with the knowledgebase module. This allows the knowledge base to invoke methods of the JTMS but also means that any other TMS module with the same API could be used in place of the JTMS module. Furthermore, if no TMS module is registered at all then the knowledgebase will operate in a stand-alone fashion.

4.1.3 The JTMS

The JTMS module is also designed to be used as a stand-alone component. Since a JTMS requires a knowledgebase and inference engine to operate on, it can’t be used in isolation; however, different inference engines can be registered with the JTMS when it is initialised.

The JTMS module implements the algorithm described in section 2.4.1 to ensure that unnecessary backtracking does not occur when facts are retracted from the knowledgebase. The JTMS module will use the knowledgebase module’s API to retract any inconsistent (no longer justified) facts which result from the change to the knowledgebase. Each of these retractions in turn will be applied to the JTMS network until all inconsistent sentences have been removed.

By default, since the data must be passed through the web, nodes which are considered OUT are removed from the network and reconstructed if needed, rather than their label being set to OUT as the algorithm requires. This results in slightly more CPU load since object construction and destruction occurs more frequently but the difference is negligible. This also has the convenient side effect of producing a ’bijection’ between facts in the knowledge base and nodes in the JTMS. Those using the JTMS module in stand alone mode and wishing to keep nodes which have been made OUT and simply re-label them can do so by commenting one line in the code.

In order to demonstrate the JTMS in action consider the following example. Let us assume that through the user's actions the system has entered the following facts which describe the paper being reviewed:

```
Facts before modus ponens:
original(hypothesis1, myPaper)
nontrivial(hypothesis1, myPaper)
original(hypothesis2, myPaper)
nontrivial(hypothesis2, myPaper)
relevant(myPaper, strongly)
review(myPaper)
comprehensive(myPaper)
```

These might have been asserted from other longer chains of inference as a result of the users decisions, and they represent the belief that the paper has two original and non-trivial hypotheses, that it is strongly relevant to the journal for which it is being considered and that it is also a review which comprehensively covers the subject area. Amongst the rules in the system are the following three rules which we will consider:

```
Rules before modus ponens:
original(*hypothesis*, *paper*) ˆ nontrivial(*hypothesis*, *paper*) => significant(
significant(*paper*) ˆ relevant(*paper*, strongly) => accept(*paper*)
review(*paper*) ˆ comprehensive(*paper*) => accept(*paper*)
```

These rules assert respectively that 1. a paper which has an original and non-trivial hypothesis is significant, 2. a significant and strongly relevant paper should be accepted and 3. that a paper which comprehensively reviews the subject should also be accepted. As seen from the output of the KB and JTMS systems this leads the system to conclude the additional facts significant(myPaper) and accept(myPaper):

```
Facts after modus ponens:
original(hypothesis1, myPaper)
nontrivial(hypothesis1, myPaper)
original(hypothesis2, myPaper)
nontrivial(hypothesis2, myPaper)
relevant(myPaper, strongly)
```

```
review(myPaper)
comprehensive(myPaper)
significant(myPaper)
accept(myPaper)
```

The first network in figure 4.2 shows the current state of the JTMS network. Notice that justification nodes correspond to instantiated horns in this implementation. This sacrifices some storage space but has the benefit of yielding a less connected network and also requires less search when checking if a node is IN or OUT. It is also important to note that all of the nodes are considered IN at this point. The sentence nodes which were entered initially have been treated as assumptions, that is, they have been assigned a justification of true. The node true is considered to always be IN. These extra justification nodes have been omitted for clarity.

Imagine that the user then backtracks and as a result of further research finds an earlier paper which describes hypothesis1. The implications of this for the network are represented in the second network in figure 4.2. This leads to the retraction of the fact original(hypothesis1, myPaper). This means that the justification node to which this fact connects becomes OUT. Notice however that the fact significant(myPaper) is still IN since it was also justified by claims about its second hypothesis, and a node only requires one of its justifications to be IN in order to be considered IN itself. We can confirm that the system believes significant(myPaper) in this case by retracting the fact orginal(hypothesis1, myPaper) and outputting the currently believed facts:

```
Retracting original(hypothesis1, myPaper)

Facts after retracting original(hypothesis1, myPaper):
nontrivial(hypothesis1, myPaper)
original(hypothesis2, myPaper)
nontrivial(hypothesis2, myPaper)
relevant(myPaper, strongly)
review(myPaper)
comprehensive(myPaper)
significant(myPaper)
accept(myPaper)
```

Now consider what happens if the user discovers some further papers which show that the second hypothesis is not original either. In this case we would expect the

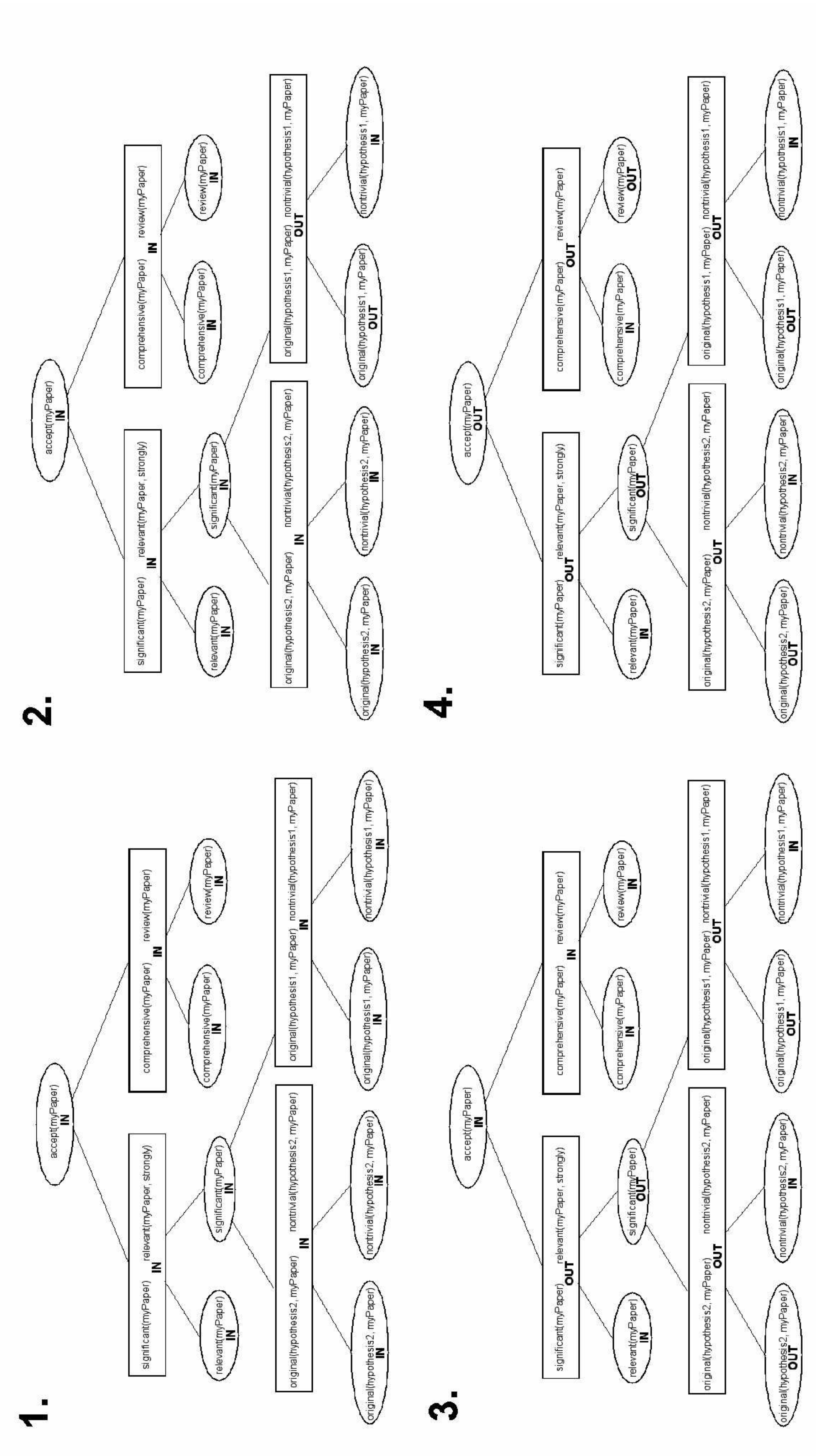

1.
accept(myPaper) IN
significant(myPaper) relevant(myPaper, strongly) IN
comprehensive(myPaper) review(myPaper) IN
relevant(myPaper) IN
significant(myPaper) IN
comprehensive(myPaper) IN
review(myPaper) IN
original(hypothesis2, myPaper) nontrivial(hypothesis2, myPaper) IN
original(hypothesis1, myPaper) nontrivial(hypothesis1, myPaper) IN
original(hypothesis2, myPaper) IN
nontrivial(hypothesis2, myPaper) IN
original(hypothesis1, myPaper) IN
nontrivial(hypothesis1, myPaper) IN
2.
accept(myPaper) IN
significant(myPaper) relevant(myPaper, strongly) IN
comprehensive(myPaper) review(myPaper) IN
relevant(myPaper) IN
significant(myPaper) IN
comprehensive(myPaper) IN
review(myPaper) IN
original(hypothesis2, myPaper) nontrivial(hypothesis2, myPaper) IN
original(hypothesis1, myPaper) nontrivial(hypothesis1, myPaper) OUT
original(hypothesis2, myPaper) IN
nontrivial(hypothesis2, myPaper) IN
original(hypothesis1, myPaper) OUT
nontrivial(hypothesis1, myPaper) IN
3.
accept(myPaper) IN
significant(myPaper) relevant(myPaper, strongly) OUT
comprehensive(myPaper) review(myPaper) IN
relevant(myPaper) IN
significant(myPaper) OUT
comprehensive(myPaper) IN
review(myPaper) IN
original(hypothesis2, myPaper) nontrivial(hypothesis2, myPaper) OUT
original(hypothesis1, myPaper) nontrivial(hypothesis1, myPaper) OUT
original(hypothesis2, myPaper) OUT
nontrivial(hypothesis2, myPaper) IN
original(hypothesis1, myPaper) OUT
nontrivial(hypothesis1, myPaper) IN
4.
accept(myPaper) OUT
significant(myPaper) relevant(myPaper, strongly) OUT
comprehensive(myPaper) review(myPaper) OUT
relevant(myPaper) IN
significant(myPaper) OUT
comprehensive(myPaper) IN
review(myPaper) OUT
original(hypothesis2, myPaper) nontrivial(hypothesis2, myPaper) OUT
original(hypothesis1, myPaper) nontrivial(hypothesis1, myPaper) OUT
original(hypothesis2, myPaper) OUT
nontrivial(hypothesis2, myPaper) IN
original(hypothesis1, myPaper) OUT
nontrivial(hypothesis1, myPaper) IN

Figure 4.2: A JTMS network at several states during exectuion

situation to resemble the third network in figure 2. Notice that the paper is no longer considered significant which means that one of the justifications for accepting the paper has been removed. The paper can still however be accepted on the basis that it is a comprehensive review of the subject. We can once again check the output of the program to confirm that it corresponds to the state of the JTMS network:

```
Retracting original(hypothesis2, myPaper)

Facts after retracting original(hypothesis2, myPaper):
nontrivial(hypothesis1, myPaper)
nontrivial(hypothesis2, myPaper)
relevant(myPaper, strongly)
review(myPaper)
comprehensive(myPaper)
accept(myPaper)
```

For completeness let us consider what would happen if the user backtracked further and decided that although the paper did provide some review of the subject it wasn't really the main focus of the paper. This could lead to the fact review(myPaper) being retracted. This finally means that all of the justifications for accept(myPaper) are OUT, and accordingly the paper is no longer accepted:

```
Retracting review(myPaper)

Facts after retracting review(myPaper):
nontrivial(hypothesis1, myPaper)
nontrivial(hypothesis2, myPaper)
relevant(myPaper, strongly)
comprehensive(myPaper)
```

4.2 The Application Layer

The application layer is the actual instance of an expert system which sits on top of the foundation layer. It is this layer that is different for each application which is built using the PHP inference engine and JTMS module. In the case of ERA version 4 the application layer is implemented primarily in server side PHP which alters the HTML which is delivered to the client. Client-side JavaScript is used to dynamically load data into pages and Cascading Style Sheets (CSS) are used to style the output. The architecture is presented visually in figure 4.3.

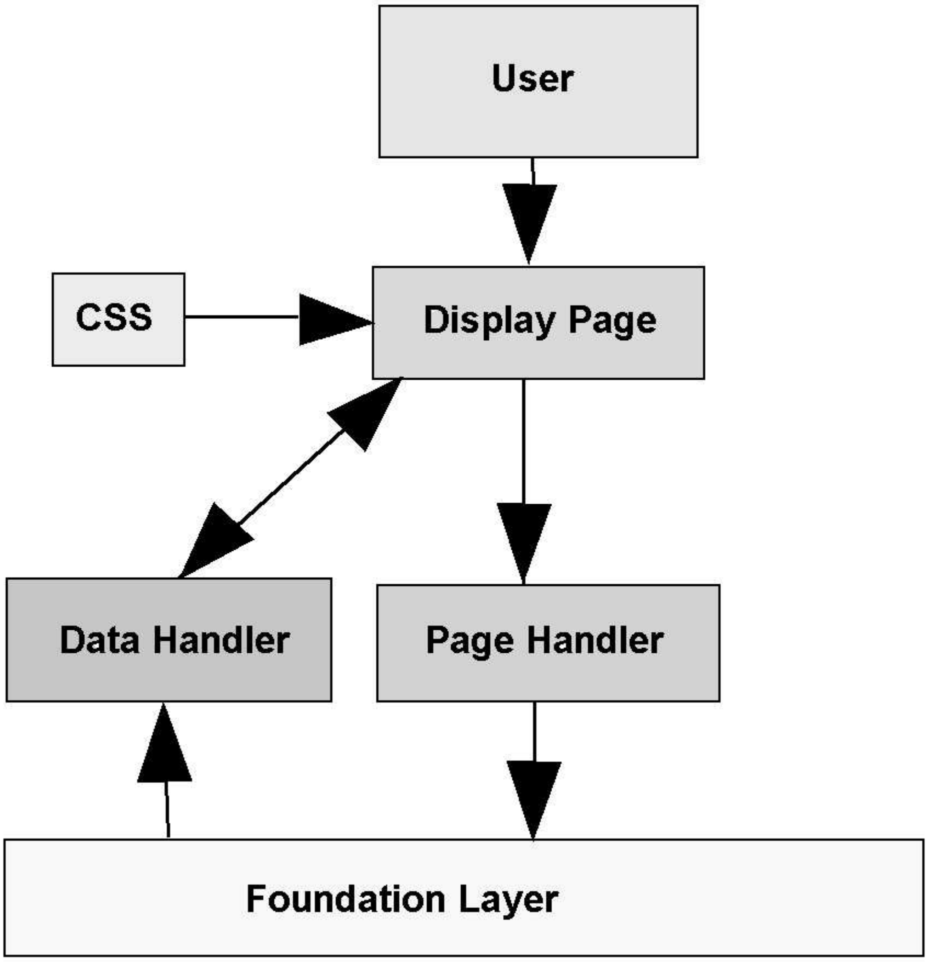


Figure 4.3: The application layer architecture

4.2.1 Application Layer architecture

The architecture of the application layer is loosely based on that of ERA version 3. The execution is divided between modules which are responsible for different criteria which should be judged in the final review. Each module has three component parts; the first is a PHP Display Page which displays the questions to the user. This page also monitors the user's response and loads subsequent questions into the page from a Data

Handler.

The second part of each module is a Data Handler; these PHP pages receive the input from the user and return HTML which can be injected into the page dynamically. This HTML contains further questions for the user which are dependent on the input. The Data Handler and Display Page interact through asynchronous calls to the server as described in section 4.2.2.

Some of these Data Handlers are split into smaller sub-handlers where there are a series of questions which are all closely related. For instance the Summary Data Handler has separate sub-handlers for handling questions about systems, questions about surveys and questions about the paper's hypotheses.

The third part of each module is a Page Handler which takes the data submitted by the user and asserts appropriate facts into the knowledgebase. Further, the Page Handler must remove any inappropriate facts if the page to which it belongs is revisited. The Page Handler is also responsible for passing execution to the next module as appropriate.

The list of modules is given below in the default order that a user would experience them (that is if the user did not use the backtracking functionality).

Setup

This module initialises the inference engine and JTMS in the foundation layer and sets up the data structures required by further modules; this includes setting the default order for the modules.

Initial Information

This module simply gathers the name of the paper being reviewed, the names of the paper's authors and the reviewer's name.

Summary

This module is the most complex in the system, it gathers information about the content of the paper by asking the reviewer a series of questions. The answers to these questions determine which questions are asked subsequently in this module. Typically these questions are aimed at categorizing the paper in some way or extracting verbose written information about some aspect of the paper. For instance several of the questions ask the user to describe parts of the paper.

Technique:

The Technique module was originally a sub module of the Summary module; however its complexity merited it being promoted to a full module with its own handler. This module asks the user to categorise any techniques described in the paper in much the same way as the Summary module asks the user to categorize systems or surveys. However ERA version 4 supports up to 10 independent techniques, so technique 1 could be categorised as being widely applicable whereas technique 2 could be categorised as being not widely applicable. This means that the user may need to visit the Technique module multiple times.

Multiple techniques are handled by using a HTTP GET variable embedded in the URL to the Technique module. This parameter controls which technique is currently in focus. For example if the Technique module was invoked with the URL "/technique.php?TechniqueNumber=1" then technique 1 would be in focus, whereas if the URL "/technique.php?TechniqueNumber=2" was used then technique 2 would be in focus.

This idea of having independent objects occurs frequently throughout ERA version 4; for instance there can be multiple hypotheses in a paper, or multiple tools used to create a composite system. However in other cases since the number of questions which depend on these objects is significantly smaller these are dealt with in their containing modules Page Handlers.

Relevance

This section asks the user to assess the relevance of the paper. If the user is unsure then more detailed questions are presented which should be easier to answer. The precise questions asked depend on the answers to the previous modules.

Originality

This section asks the user to assess the originality of the paper. As with the Relevance module the user may select the "I'm not sure" option which will then present a set of more detailed questions to help the user accurately gauge how original the paper is.

Validity

This module is similar to the previous two except that it contains questions which probe the correctness of the paper and urges the reviewer to look for flaws in informal or formal arguments and unjustified claims.

Presentation

The presentation module asks the user to assess how well the paper is presented. Here if the user selects the "I'm not sure" option the user is asked to assess the organisation of the paper, the standard of written English, the readability of the paper and the thoroughness of the bibliography. The first three of these sections can also be broken down into a series of simple yes / no questions.

Final Review

The last module collects all of the information entered by the user and constructs a report based on this information and on inferences drawn. Time constraints prevented adding a way to change any inferred sections of the report on this form but this would not in principle be a difficult thing to add to the module.

It is worth noting that although the final report has a section for significance there is no corresponding module. The reason for this is that the types of questions which determine significance are closely related to description and categorisation. For example the questions ”How widely applicable is this technique?” and ”Is this technique an improvement over the previous version?” are closely related to the description of the technique in question. For these reasons questions which help to determine significance have been amalgamated into the appropriate section of the Summary and Technique modules.

4.2.2 Providing a Dynamic Interface

One of the problems with using the web as an interface is that a web page is traditionally a static medium. This means that in a web application any server side logic can only be executed between page loads. A normal execution flow would be:

```
1. Server serves page to the client
2. Client sends data to the server by clicking a link causing a new page to load or
3. Server executes logic
4. Server serves the result as a new page
```

This results in many web applications either using a large number of page refreshes or a large number of small pages being used with very little functionality on each page. Either way, the result, is a far from fluid experience with a lot of waiting for the data to make the round-trip to and from the server.

ERA version 4 uses XMLHtppRequest Objects to make calls asynchronously to the server. This Java Script object is capable of querying the server without blocking execution of the page. Event handlers on this object can then inject the server’s response directly into the current page via the innerHTML property without the need for a refresh or new page load.

This approach has been dubbed AJAX (Asynchronous Java And XML) by adaptive path [19] and has been pushed into the spotlight by its aggressive use by companies like Google [17] [18] to create much richer web interfaces.

ERA version 4 uses this functionality to dynamically load questions into the page based on the user's responses. So for example consider the user responding to the question ”Does this paper describe a system?” First the user selects the Yes radio button on the form. This invokes a Java Script event handler which creates an XMLHtppRequest object which calls a server side Data Handler with ”system=yes” passed as a parameter. Execution is then passed back to the web browser GUI, which prevents server delays from locking the web page and preventing the user from taking any action. The Data Handler then returns a chunk of HTML containing questions about systems to the XMLHttpRequest object, which causes an event handler to execute which injects the HTML directly into page causing the question about systems to appear underneath the radio button which the user originally pressed.

4.2.3 Controlling Execution and Knowledge Base State over User Backtracks

The execution flow is controlled by a LIFO (Last In First Out) agenda data structure which contains a list of Display Pages which still need to be executed. When the Page Handler of any module is executed, the top object of the agenda is removed and execution flows to the next Display Page in the agenda. This behaviour means that the top item in the agenda will always contain the next unseen Display Page. The result is that no matter which module the user is currently viewing (either by using the ”back” link or jumping there directly) pressing the submit button will cause execution to flow to the next Display Page for which the user has not submitted data.

The back link is controlled by another LIFO data structure called the History. In this case each Display Page generates the back hyperlink by looking at the top item in the History. In addition the link adds a HTTP GET parameter of ”back=true” to the URL of link. So, for example, if the top entry of the History is ”technique.php” the ”back” link would in fact point to ”technique.php?param1=val1&&back=true. Pressing the submit button on a Display Page adds the Display Page's URL to the history. Finally if any display page detects a HTTP GET parameter of ”back=true” the top entry of the history is removed. This is required to ensure that multiple concurrent backtracks are correctly handled.

Recall that facts are only ever added to the knowledgebase by Page Handlers. Each Page Handler keeps track of all of the facts it has asserted in a session variable. The first action the Page Handler takes is to retract all of the facts it has stored in this session variable. New facts are then asserted based on the user input received. Since the user changing the user input to one of the sections necessarily results in another execution of the Page Handler for the module in question this ensures that the facts asserted as a direct result of user input are consistent. Any facts which may have been added due to inference are automatically dealt with since the retract method of the knowledgebase module in the foundation layer is directly linked to the JTMS.

In some cases changing input means the subsequent sections of the review need to be revisited. For example if in the Summary module the number of techniques is increased then the Technique module needs to revisited to collect responses about the added technique. In cases like this the appropriate Page Handler simply adds the appropriate modules to the agenda.

4.2.4 Supplementing Horn Clauses

On of the drawbacks of ERA version 4 is that it can only use horn clauses. This causes a problem when trying to implement rules which are essentially weighted averages like those of the calculator methods in ERA version 3. Instead the rules in the knowledgebase can be written so that the number of justifications for a given fact can be used as a threshold value. So for instance if the JTMS node associated with the fact ”recommendation (presentation)” has between 7 and 10 justifications the program would suggest a score of ”excellent” whereas if it had only between 1 and 3 justifications the system might recommend a score of ”poor”.

This lacks some of the expressiveness of the weighted averages which are used in ERA version 3 but does push the limits of what can be achieved through only using horn clauses for reasoning.

4.3 Summary

This section describes how ERA version 4 has been implemented using a two layer architecture to encourage reuse of the software components in the foundation layer. The implementation of the application layer is described in detail as well as how using asynchronous JavaScript can lead to a much richer user experience.

Chapter 5

Evaluation

This chapter describes the methodology used to evaluate ERA version 4, it goes on to present and discuss a summary of the results including an analysis of what the results reveal about the system. Full copies of the results are included in the appendices to this document.

5.1 Evaluation

The goal of this project is to improve the inference capabilities and usability of ERA, and hence also improve the quality of the reviews that it helps users to generate; the concrete approach adopted to do this has been to build a JTMS into ERA which would allow backtracking so that users can change their responses to parts of the dialogue at any time. That this JTMS works according to its specification can be demonstrated without the need for any kind of experimental design.

The first aspect of the evaluation deals with the question of whether using ERA version 4 produces a qualitatively better review than either using ERA version 3 or simply writing a review by hand.

This dissertation also makes the claim that the new version of ERA is more usable due in part to its backtracking capabilities, but also due to other new features such as an improved user interface. The first section of the evaluation is along this user satisfaction dimension and attempts to determine if ERA version 4 is a significant improvement over ERA version 3.

Since it is difficult to find enough volunteers to perform a number of reviews using various systems, where possible evaluation has been designed so that results from ERA version 3 [20] and ERA version 1 [7] could be used for comparison.

5.1.1 Methodology

Each of a selection of informatics students was asked to produce 1 review by hand using only the ”Notes to CADE-12 referees” [5] as a guideline, 1 review using ERA version 3 and 1 review using ERA version 4. In order to reduce any biasing caused by familiarity or gaining reviewing experience in the course of this evaluation process, the order in which the reviews were to be performed was randomised for each reviewer. All of the students in this experiment had at least 3 years university tuition in informatics or computer science. In addition one is employed in IT.

The papers for reviewal in each case were chosen randomly from a population of 4 real research papers which were part of the IRM 2004/2005 (ref on lit survey) course. These papers were selected for inclusion in the IRM coursework because each raised ’concerns’ - that is, it appeared to have significant flaws or controversial sections - which should be recognised in a good review of the paper.

Once completed, each of the review was analysed, and the number of concerns from Alan Bundy’s discussion of the results which were also remarked upon in the review was recorded. Cases where a review pointed out a concern which was not contained in Alan Bundy’s notes were dealt with on a case by case basis: those which were felt to be valid concerns were added to those on the list, with any others treated as false positives. This allows the calculation of mean precision and recall scores for each of the reviewal methods. In this context precision is defined as the proportion of issues raised by the reviewing technique which are actually real issues (as opposed to false positives) and recall is the proportion of issues which were spotted by a give reviewing method. It is hoped the ERA version 4 gives a higher precision and recall than does using ERA version 3 or writing a review by hand.

Finally the participants were asked to complete a short questionnaire. The first section of this questionnaire is based upon the one which appears in [20], with the questions were altered to refer specifically to ERA version 4. The final questions included were:

```
1. ERA version 4 helped me make a better review
2. ERA version 4 helped me to learn how to become a better reviewer
3. The relevant aspects of the papers were covered by ERA version 4
4. The interface of ERA version 4 is clear and easy to use
5. ERA version 4's reviewing process is well structured
6. ERA version 4 provided helpful answers to any questions I had
```

7. I would use ERA version 4 again

The first question, ”ERA version 4 helped me make a better review”, asks the user to qualitatively judge if she feels that ERA has improved the quality of her review. Positive responses can be seen partly as a measure of user satisfaction but also as indirect evidence suggesting the use of ERA does, in fact, improve reviews since review authors can be expected to be sensitive to the change in quality of their reviews.

The next question, ”ERA version 4 helped me to learn how to become a better reviewer”, also judges user satisfaction with the system but also assesses if ERA has managed to teach the user anything. This could be through either asking insightful questions or through explanations provided by the system.

Question number 3, ”The relevant aspects of the papers were covered by ERA version 4”, tries to assess whether the user feels that all pertinent questions were asked during the reviewal of the paper. Failure to ask a critical question could result in a substantially different review and could allow fatally flawed papers to be accepted. It is therefore important that the system is as complete as possible.

The fourth question, ”The interface of ERA version 4 is clear and easy to use”, is one which is particularly important for ERA version 4 since the user interface has several new additions. While the interface might be more powerful (in terms of the functionality it offers the user), it is important that it remains intuitive and usable.

Similarly the next question, ”ERA version 4's reviewing process is well structured”, is an important one, since the structure of ERA was altered slightly and it is important that the structure reflects a sensible and methodical approach to reviewing.

Question 6, ”ERA version 4 provided helpful answers to any questions I had”, works on two different axes: firstly it reflect how appropriate the inferences made by the system have been; and secondly it reflects how well-written the user information screens are. Negative responses represent fundamental flaws in either the inference rules used or in the way questions were presented to the user.

The final question, ”I would use ERA version 4 again”, simply gauges the user's overall satisfaction with the system. Positive responses to this question indicate that ERA is a useful and valuable tool.

Hutchinson [20] also included the questions ”The system's recommended ratings were satisfactory” and ”I have a great deal of experience reviewing papers”. The first was omitted since it is no longer relevant to ERA version 4 and the second was omitted since all of the reviewers were students and could be expected to have little experience of reviewing papers.

These questions were answered by the user by indicating a mark on a scale of 1 to 5, ranging from 1, representing strong agreement, to 5, representing strong disagreement. Hence, in general low scores represent responses that are more favourable to the system. This rating method also allows direct comparison of the data collected with those presented by both Caporale‘ and Hutchinson, who both used the same method.

In addition, several questions were added to the questionnaire to ask about the new features particular to ERA version 4. These questions are:

```
8. The sidebar panel in ERA version 4 was useful to me
9.  The "back" link in ERA version 4 was useful to me
10. The ability to return to previously completed sections of the review in ERA ver
```

As with the previous questions, the range of responses offered to the user were such that a low score would reflect positively on the system and a high score reflect negatively. These questions were designed specifically to decide if the backtracking facilities were felt by the user to be worthwhile additions to the system. Question 11 tries to determine if users find useful the ability to move freely between sections of the review, whereas the preceding two questions attempt to determine the usefulness of each of the two mechanisms offered for achieving this.

In addition, space for user comments was provided to allow users to give overall impressions of the system, comments about what was done well/badly and further suggestions for improving the system.

5.2 Results

This section discusses the results obtained through the experimentation. Unfortunately since this project was performed during the summer break for Undergraduate students finding a selection of volunteers was extremely difficult. Only three volunteers were available and as such most of the results are tentative at best. A more thorough evaluation could be completed if, like previous versions of ERA, this system could be used as part of the IRM module.

5.2.1 Analysing Results: Questionnaires

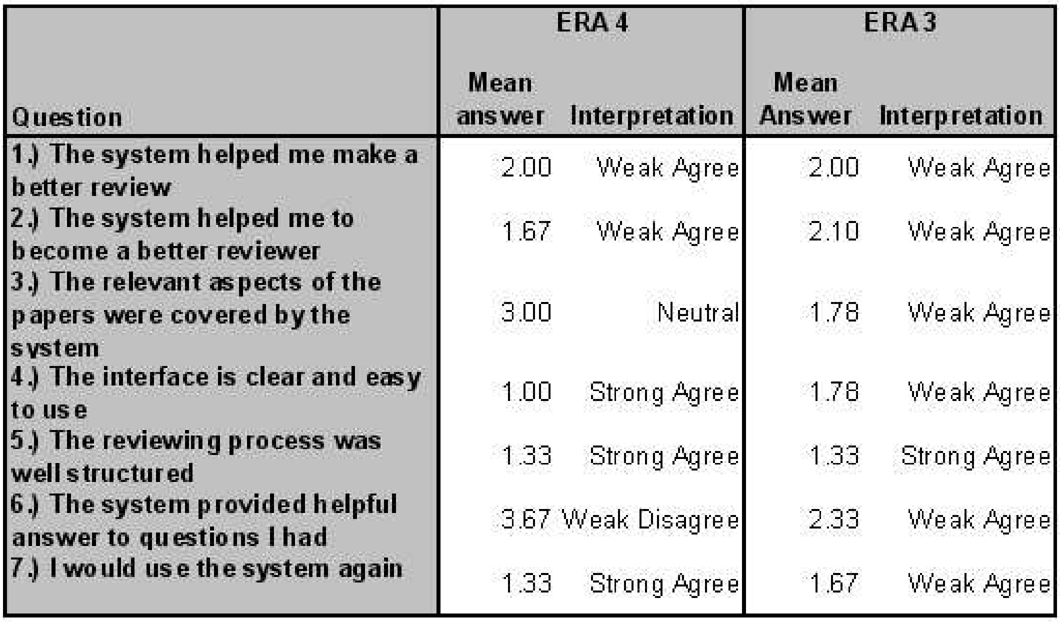
The methodology used to compare the results collected is essentially the same as that used in Hutchinson 2004. The results were compared to both ERA version 3 and ERA version 4. The tables below show the mean scores for each question along with a Students T-test probability. The T-test is a statistical test which, under certain assumptions about the problem, indicates the probability that the data came from the same population. It is general practice to consider a t-test probability of 0.05 significant; this corresponds to a 95% chance that the data comes from different distributions.

The data collected in this project is assumed to be taken from a source population which meets the criteria for the T-test. Namely that the underlying data is normally distributed with homogeneous variance between the two datasets and that the data is equi-distant interval data. This last assumption is slightly problematic since different people might not consider the difference between a response of 1 and 2 and the difference between a response of 2 and 3 to be equal. This is particularly the case for those people who ”never award a top mark” since they perceive the intervals at the extremes of the scales to be larger. In any case since each test candidate awarded at least one mark of 1 this was considered not to be the case.

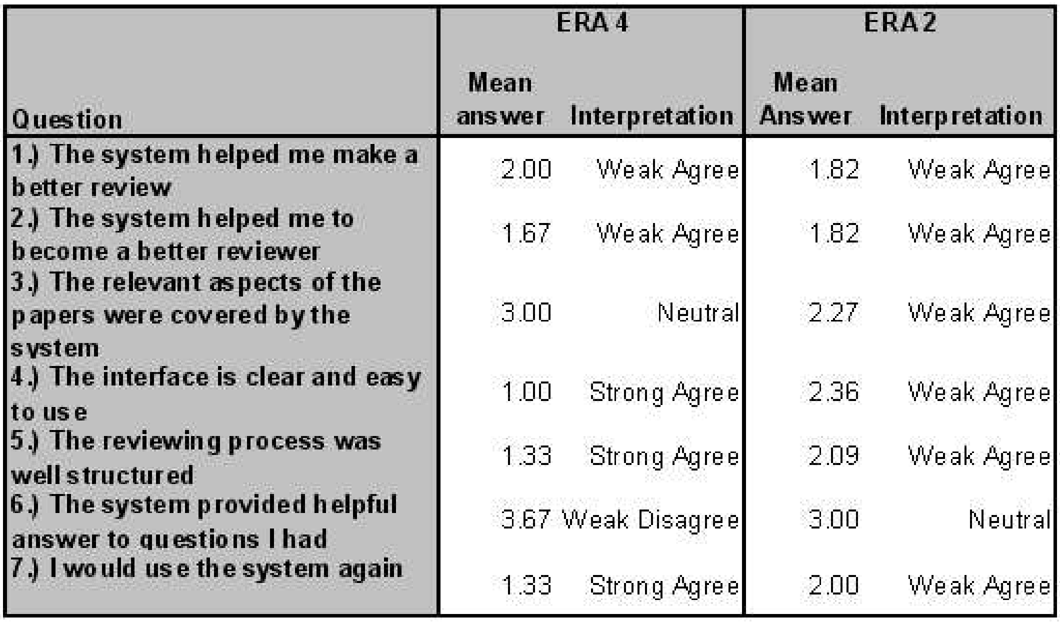
As can be seen for the majority of the comparisons there is no evidence that the results come from different distributions. This is not largely surprising since the data set for ERA 4 was only of size 3. Unfortunately the only significant result does not reflect well on ERA version 4. Question 3 shows a drop in mean response from Weak Agree to Neutral between ERA 3 and ERA 4 suggesting that ERA 4 did not cover as many relevant aspects of the paper as ERA 3 did. It is a possibility that this is a result of amalgamating questions of significance with the Summary and Technique modules. This results in one less verbose element of user feedback (that corresponding to motivating the choices in the now-absent Significance module) and possibly an impression that significance has not been properly assessed.

Question 6, the t-test probability for which approaches significance, saw an even more drastic drop in mean response from a Weak Agree to a Weak Disagree. This is somewhat expected since time constraints prevented the implementation of automated recommendations for any modules except for Presentation. The addition of these recommendations would hopefully raise the score significantly.

The results were also compared using the Mann-Whitney U test. As in [20] the total mark was calculated for each questionnaire and the results used as the population for the test. The Mann-Whitney U test is a hypothesis test of the equality of two populations and is an alternative to the t-test for ordinal data. The data in this case must be treated as ordinal rather than interval since the ”value” of each question is almost certainly not the same. For example one of the most important questions on the



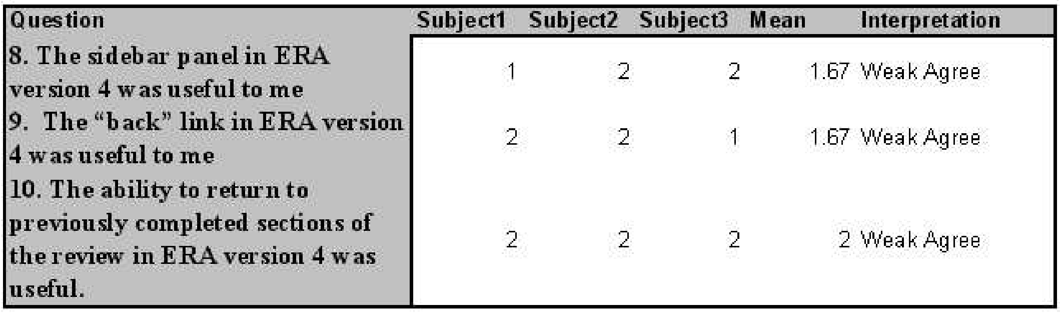
	ERA 4		ERA 3	
Question	Mean answer	Interpretation	Mean Answer	Interpretation
1.) The system helped me make a better review	2.00	Weak Agree	2.00	Weak Agree
2.) The system helped me to become a better reviewer	1.67	Weak Agree	2.10	Weak Agree
3.) The relevant aspects of the papers were covered by the system	3.00	Neutral	1.78	Weak Agree
4.) The interface is clear and easy to use	1.00	Strong Agree	1.78	Weak Agree
5.) The reviewing process was well structured	1.33	Strong Agree	1.33	Strong Agree
6.) The system provided helpful answer to questions I had	3.67	Weak Disagree	2.33	Weak Agree
7.) I would use the system again	1.33	Strong Agree	1.67	Weak Agree



	ERA 4		ERA 2	
Question	Mean answer	Interpretation	Mean Answer	Interpretation
1.) The system helped me make a better review	2.00	Weak Agree	1.82	Weak Agree
2.) The system helped me to become a better reviewer	1.67	Weak Agree	1.82	Weak Agree
3.) The relevant aspects of the papers were covered by the system	3.00	Neutral	2.27	Weak Agree
4.) The interface is clear and easy to use	1.00	Strong Agree	2.36	Weak Agree
5.) The reviewing process was well structured	1.33	Strong Agree	2.09	Weak Agree
6.) The system provided helpful answer to questions I had	3.67	Weak Disagree	3.00	Neutral
7.) I would use the system again	1.33	Strong Agree	2.00	Weak Agree

questionnaire is ”I would use this system again”, whereas the question ”The interface is clear and easy to use” could be considered of lesser importance. Thus the calculation of a total mark corresponds to performing a transformation on the data leading to the necessity of treating it as ordinal. The significance coefficients came out as p = 0.58 between ERA 3 and ERA 4 and p = 0.59 between ERA 2 and ERA 4. This is clearly under the p=0.05 confidence level and so there is no significant evidence to suggest that the data sets are different.

The final 3 questions on the review concerned the features added to the ERA version 4 interface, and consequently these were questions introduced for the evaluation of this version. The data set is small so it is presented here in its entirety along with the mean value.



Question	Subject1	Subject2	Subject3	Mean	Interpretation
8. The sidebar panel in ERA version 4 was useful to me	1	2	2	1.67	Weak Agree
9. The “back” link in ERA version 4 was useful to me	2	2	1	1.67	Weak Agree
10. The ability to return to previously completed sections of the review in ERA version 4 was useful.	2	2	2	2	Weak Agree

The data provides an encouraging mean response of Weak Agree for all of the questions which does indicate that the features were useful to the reviewers; however it should of course be noted that a sample size so small can’t really be considered representative.

5.2.2 Analysing Results: Reviews

When submitting their reviews two subjects reported problems working with ERA 3. Both reported the same error occurring which prevented them from completing their review. The error text reported was:

```
werror
[ARGACCES5] Function + expected argument #2 to be of type integer or float
[PRCCODE4] Execution halted during the actions of defrule overall.
```

I checked with the subjects and this error was repeated if they entered the same results again. It appears to be a bug which is caused by one of the select boxes which provides a score for one of the sections is not returning an integer. This is most likely to be an error in the HTML form, probably a missing missing VALUE = ”N” statement from one of the ¡OPTION¿ tags.

One of the reviewers enclosed what his responses would have been for the subsequent sections which could not be completed resulting in a complete review. The other review which was affected by this bug was left incomplete. It could be argued that this remaining incomplete review has a negative effect on the evaluation of ERA 3. However it should be noted that ERA 3 is a live deployed system and should be evaluated as such.

The data on the marked reviews is displayed below:

Subject1			
	Hand Review	ERA3	ERA4
Flaws identified	5	1	1
False Positives	0	0	1
Total Flaws	7	6	4
Precision	100%	100%	50%
Recall	71%	17%	25%

Subject1			
	Hand Review	ERA3	ERA4
Flaws identified	2	2	3
False Positives	1	0	1
Total Flaws	4	7	6
Precision	67%	100%	75%
Recall	50%	29%	50%

Subject1			
	Hand Review	ERA3	ERA4
Flaws identified	2	0	3
False Positives	0	0	0
Total Flaws	6	4	7
Precision	100%	100%	100%
Recall	33%	0%	43%

This leads to the average precision and recall scores given below:

	Hand review	ERA 3	ERA 4
Precision	100%	100%	75%
Recall	52%	15%	39%

As can be seen the hand review is substantially better than both of the expert systems. ERA 4 has a better recall than ERA 3 meaning that more flaws are correctly identified but it seems to hit more false positives. Examination of the reviews themselves suggests that ERA 4's ability to define multiple hypotheses allowed users to evaluate evidence as it related to each hypothesis separately which helped to reveal more of the flaws in the papers.

However it seems that neither method is yet as good as a hand-written review given the appropriate guidance. In this case the candidates were given access to the ”Notes to CADE-12 Referees” [5] document to aid them in performing their review. This implies that the expert knowledge encapsulated in this document has not yet been effectively exploited by either version of the ERA system.

5.2.3 Analysing Server Logs

Looking at the server logs for the ERA system allowed the tracking of how many times the ”Back” links and Sidebar links had been used. When a user follows a ”back” link the HTTP GET parameter ”back=true” is appended to the URL. Similarly when a sidebar link is followed the HTTP GET parameter ”sidebar=true” is appended to the URL. Hence, scanning the server logs for the occurrence of these two parameters reveals how often these capabilities were used. By grouping the appropriate entries by IP address we can tabulate this by user. (While these are listed as subject 1, subject 2 etc these enumerations are arbitrary and do not necessarily correspond to the subject enumerated in the previous section of this chapter.) The results are shown below:

	Subject1	Subject2	Subject 3	Total
Use of back link	1	2	1	4
Use of sidebar link	3	1	2	6

The functionality seems to have been well received with the sidebar links proving to be generally more popular. However, this could be due to their prominent positioning rather than a strong preference for this type of navigation system.

5.3 Summary

Despite only a small sample of volunteers the results collected show some interesting properties. The backtrackin functionality seems to be well received, and possibly due to its increased stability ERA version 4 seems to produce a better review than ERA version 3.

Chapter 6

Conclusion

This project has achieved the primary goal of adding backtracking support to the ERA system through the use of a JTMS. This allows users to use a ”back” link to traverse back through the built up ”history” of previously viewed pages in much the same way as users use the back button of their web browser. Alternatively users can select to jump back directly to any previously visited section of the report via a dynamically generated set of links in the right hand margin of the interface. The JTMS ensures that the knowledge base is kept in a consistent state throughout these operations. User evaluation on the completed system suggests that these features have made ERA version 4 more usable and this viewpoint is further supported by the fact that the lack of these features was the most common criticism of previous versions of ERA to date.

Through the fulfilment of this objective a set of reusable modules which together constitute a basic framework for logic programming on the web has been produced in the form of the foundation layer. These modules are written in PHP so that they can integrate directly with web based scripts through the provided APIs. These APIs also allow for a cleaner separation between the program control logic and the knowledge base.

The usability results comparing ERA version 4 with its predecessors were for the most part not significant due to the small sample size however the results do highlight a poor decision to absorb the questions about significance into the Summary and Technique modules. From a programming point of view it would not be challenging to reverse this decision and as such this is not seen as a fundamental problem with the system.

Analysis of the reviews produced suggest that there is still a long way to go to fully encapsulate the knowledge of even relatively inexperienced reviewers let alone domain experts however in general ERA version 4 seemed to produce better reviews than ERA 3.

The use of ERA 3 during evaluation also highlighted the stability of ERA 4 compared to ERA 3. Out of the three reviews written with ERA version 3 only one of them managed to complete without experiencing a fatal error. The apparent freedom from bugs in ERA 4 is attributed to a cleaner more modular code base that is much easier to examine. This make bugs less likely to occur in the first place but also makes them much easier to detect through unit testing and other similar techniques.

Unfortunately time constraints prevented significant increases in the amount of inference used by ERA. The inference performed is still relatively shallow and expanding the capabilities to produce more intelligent behaviour is definitely a non-trivial problem which would benefit from further work. The only significant increase in reasoning capacity is the ability to hold beliefs about an arbitrary number of similar but independent objects (such as multiple techniques). Although in this version of the ERA system the interface only allows the user to select up to 10 of these items this is a limitation only of the UI not the underlying logic.

6.1 Further Work

One obvious area for further work concerns the evaluation of ERA version 4. From the discussion in the previous chapter, it will be clear that much more evaluation is required to assess fully the alterations and additions that have been made to ERA in the course of this project.

While this project has attempted to make ERA significantly more sophisticated there are still plenty of avenues for further research and ways in which the program can be improved. Many of these suggestions easily have the scope to form projects in their own right.

6.1.1 Improving the Foundation Layer

Currently the inference engine implemented in the foundation layer is extremely limited. Being able to deal with only Horn clauses places some severe restrictions on the types of inference which are available to the system. For instance negation is not currently supported by the inference engine. While it is probably possible to artificially simulate the effects in the application layer this would be an inelegant solution at best.

The literature survey presented in chapter 2 identifies the need for a web-based inference engine since the current offerings are web ports of older expert system shells which can not be cleanly interfaced with a web application.

Producing a PHP (or other web-scripting language) inference engine which can cope with a richer subset of first order logic would be an ambitious project which would not only benefit future development of ERA, but would also benefit the wider AI community and beyond by allowing web-based expert systems to be developed much more easily. The prevalence and availability of the World Wide Web means that this would also have the effect of encouraging the dissemination of expertise to wider audiences; for example, one can imagine the benefits of delivering medical expertise over the web to physically remote communities.

The JTMS could also be extended into either an LTMS which is capable of recognising when two contradictory facts are held to be IN at once or an ATMS which holds the environments where each sentence becomes IN as arguments. Either of these variants on the JTMS could be easily integrated with the other elements in the foundation layer due to the modular design of the knowledge base and JTMS modules, and, again, would extend the possibilities for reasoning that are available over the web.

Improving these elements of the foundation layer could quickly lead to a framework for logic programming on the web. The requirements of web-based programming are significantly different to traditional programming due to the server-client nature of computing and the execution model; designing what is essentially a new expert system shell with these factors in mind would help both developers of expert systems and their potential users.

6.1.2 Examining the Reviewal Process

All of the knowledge acquisition work done for ERA to date has focused on the content of the ’static’ knowledge involved in assessing a paper - there has been relatively little work looking at the process by which a review is constructed (and it is the reviewing procedure that has been the focus of much of the work described here). Hence, it might be of benefit to study the methodologies applied by expert reviewers, with the intention of modelling these in ERA to encourage ’best practice’ approaches to reviewing. There are several ways which this could be done, each having their own advantages and disadvantages.

There are several well known methods of knowledge elicitation based upon interviewing domain experts ranging from informal question-and-answer sessions, via teach-back method, where an interviewer reformulates what an expert has said and tries to teach it back to him, through to more formal approaches using ideas like protocol analysis to capture process and laddered grids to try to identify decomposition hierarchies of the concepts that occur in a domain..

With suitable access to experts in the field a project which focuses on using some of these methods would be expected to lead to improvements in the procedural knowledge of ERA.

Another further possibility for improving the knowledge base of ERA is to use machine learning techniques to analyse the domain. The knowledge space of ERA has been calculated in the table below (by multiplying the different possible values of each possible asserted fact including its absence). As can be seen the space is huge but since we can expect many facts to be totally unrelated to each other, a machine learning approach might be able to evolve some rules which produce interesting results.

Summary	Technique	Relevance	Originality	Validity	Presentation	Total
1.97E+07	1.92E+09	1.50E+02	2.50E+03	1.57E+09	6.22E+07	1.39E+39

The appealing aspect of a machine learning approach is that we already have a system in place that is capable of capturing user data - namely ERA itself. There are several machine learning approaches which could be applicable here, but they would all involve a substantial number of expert reviews performed using ERA. One common method is to attempt to construct a decision tree based on the entropy of each argument.

6.1.3 Integrating ERA with conference management software

Integrating with a conference management system has been a goal for ERA for some time now and the potential advantages of doing so are easy to see. Reviews would be accessible instantly via a database and the submission process would be streamlined immensely. The implementation of ERA 4 in PHP has certainly brought this goal one step closer. PHP's excellent relational database support would make it trivial to serialise the existing data structures into something like mySQL or PostrgeSQL. With a suitable database schema and web front-end, reviews could be queried on any of their attributes allowing custom reports to be created quickly and easily.

For instance the final reviews could be queried across reviewers to find disagreements more easily, or all the reviews of papers by a particular author could be compared to find areas where the author is performing well, or poorly. These queries would be relatively easy to construct in SQL, although an even better long-term goal would be to make a front end for dynamically building such queries and creating custom reports.

6.2 Related Work

The ERA code base has also been used to produce an expert system known as GraPE (Grant Proposal Electronic Referee Assistant) [30] which is an expert system to help in the review of grant proposals. This is a problem which is closely related to that addressed by ERA and there is potential for the two project to benefit each other.

Like ERA GraPE suffers from having promising results but not managing to fully justify the hypothesis that expert systems can be useful advisors in the reviewal process which aid users in constructing better reviews. The problem in both cases seems to be one of knowledge elicitation and since both are reviewal tasks it seems logical to assume that some of the underlying principles might be the same between the two tasks.

6.3 Concluding Remarks

This project has achieved its main goal by providing ERA with a backtracking functionality and has significantly improved ERA in a number of ways however the problem of providing an intelligent useful tool has yet to be fully tackled. It is my opinion that this problem can’t be solved by adding more and more features to the code base.

The real gap in this work which has not been tackled is a thorough approach to Knowledge Elicitation, so far only static documents have been used in the Knowledge Acquisition process but these suffer from the problem that domain experts often don’t articulate their expertise in a way which is tractable for logical modelling. This is a difficult problem since having the level of access required to perform some of the structured interview techniques which are typically used in Knowledge Engineering is difficult and impractical at best.

The other alternative is to use machine learning techniques to train ERA into making more intelligent systems. The fact space covered by ERA is enormous and is likely to be extremely unpredictable since flaws in some sections may be fatal to a papers inclusion in a journal (such as relevance) whereas other may be more minor (such as presentation). It is my opinion that it is through pursuing this avenue that the greatest improvements for ERA now lie.

Bibliography

[1] Drools manual. 2004. http://drools.org/Rete Last accessed 30th November 2004.

[2] M Brent. Phlips homepage. 2004. http://phlips.sourceforge.net Last accessed November 30th 2004.

[3] B. Buchanan. Rule-based expert systems: The mycin experiments of the stanford heuristic programming project. 1984.

[4] Bruce H. Buchanan, B. and G. Reid. Fundamentals of expert systems. *Annual Review of Computer Science*, 3, 1988.

[5] A. Bundy. Notes to cade12 referees. 1997. http://homepages.inf.ed.ac.uk/bundy/how-tos/referee-notes.pdf Last accessed 25 August 2005.

[6] A. Bundy. Irm course web page. 2004. http://www.inf.ed.ac.uk/teaching/courses/irm/ Last accessed 30th November 2004.

[7] M. Caporale. *E.R.A. Electronic Referee Assistant*. PhD thesis, 2003.

[8] A. Crawley. Essence of artificial intelligence. 1997.

[9] Donnel Lopez et al Culbert, Riley. Clips basic programming guide v6.22. *CLIPS Reference Manual*, 2004.

[10] R. Davis and J. King. The origin of rule-based systems in ai. 1984.

[11] J. de Kleer. An assumption-based tms, artificial intelligence. *Artificial Intelligence*, 2(28):127–162, 1986.

[12] Jon Doyle. Truth maintenance systems for problem solving. *5th International Joint Conference on Artificial Intelligence*, 1977.

[13] Friedman Ernest and Hill. Jess homepage. 2004. http://herzberg.ca.sandia.gov/jess/index.shtml Last accessed November 30th 2004.

[14] C Forgy. Rete: A fast algorithm for the many pattern/ many object pattern matching problem. *Artificial Intelligence*, 19, 1982.

[15] Freeman and Hargis. *AI-depot*, 2004.

[16] Giordano. webclips homepage. 2000. http://www.monmout.com/ km2580/wchome.htm Last accessed 30th November 2004.

[17] Google. Gmail. 2004. http://www.gmail.com Last accessed 23 Aug 2005.

[18] Google. Google maps api documentation. 2005. http://www.google.com/apis/maps/documentation/ Last accessed 23 August 2005.

[19] J. Grant. 2005. http://www.adaptivepath.com/publications/essays/archives/000385.php Last accessed 23 August 2005.

[20] B. Hutchinson. *Adding Inference to ERA, the Electronic Referee Assistant, Msc thesis*. PhD thesis, 2004.

[21] Ingargolia. Cis587 course notes. 2004. http://www.cis.temple.edu/ ingargio/cis587/ Last accessed 24th August 2005.

[22] Tonberg J. *An Expert System for Project Supervision*. PhD thesis, 1995.

[23] J Lederberg. How dendral was conceived and born. *ACM Symposium on the History of Medical Informatics*, 1987.

[24] B. Maxim. Resolution and unification. *CIS 479/579 Lecture Notes*, 2004. http://www.engin.umd.umich.edu/CIS/course.des/cis579/ppt/lec9.ppt Last accessed 20 July 2005.

[25] M Menken. Jclips homepage. 2004. http://www.cs.vu.nl/ mrmenken/jclips/ Last accessed November 30th 2004.

[26] S. Russel and P. Norvig. Artificial intelligence: A modern approach. 1995.

[27] V Trakkidou. *An Expert System to Advise on Doing Projects*. PhD thesis, 2001.

[28] Stanford University. Historical projects. Unknown. http://smi-web.stanford.edu/projects/history.htmlDENDRAL Last accessed 25 August 2005.

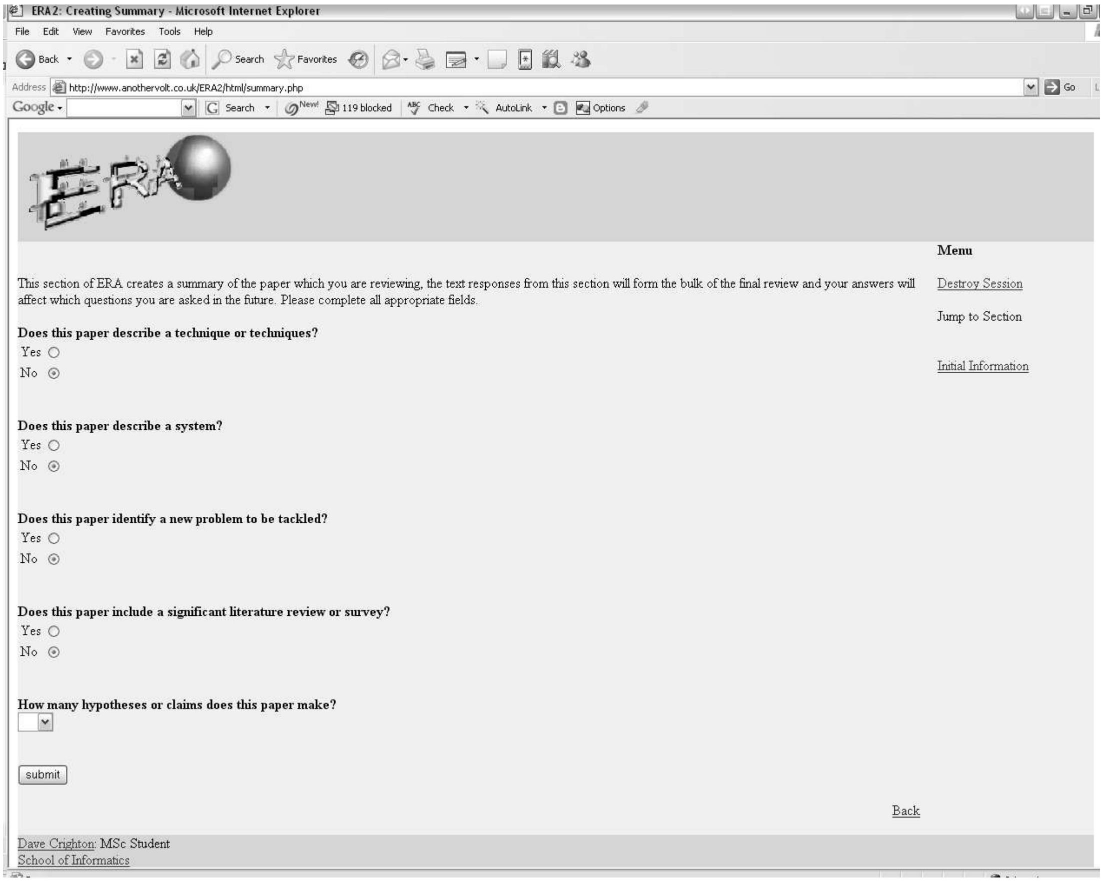
[29] F. et al Yu. An evaluation of mycin's advice. *Journal of the Americal Medical Association*, 242, 1984.

[30] Z. Yusoh. *GraPE: Grant Proposal Electronic Referee Assistant*. PhD thesis, 2004.

Appendix A

Example Screens From ERA 4

These screens show a typical run through the system.


ERA2: Creating Summary - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://www.anothervolt.co.uk/ERA2/html/summary.php
Go
Google
Search
New!
119 blocked
Check
AutoLink
Options
ERA

Menu

Destroy Session

Jump to Section

Initial Information

This section of ERA creates a summary of the paper which you are reviewing, the text responses from this section will form the bulk of the final review and your answers will affect which questions you are asked in the future. Please complete all appropriate fields.

Does this paper describe a technique or techniques?
Yes ○
No ●

Does this paper describe a system?
Yes ○
No ●

Does this paper identify a new problem to be tackled?
Yes ○
No ●

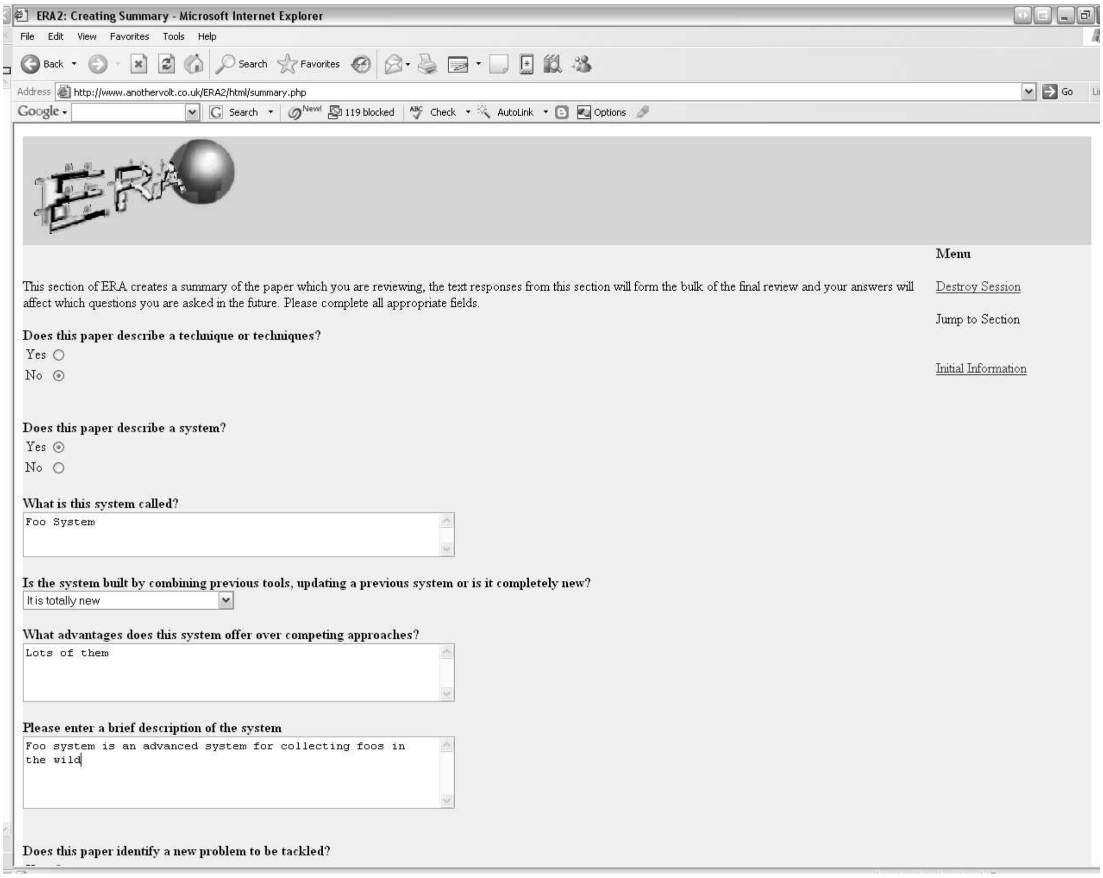
Does this paper include a significant literature review or survey?
Yes ○
No ●

How many hypotheses or claims does this paper make?

submit

Back

Dave Crighton: MSc Student
School of Informatics


ERA2: Creating Summary - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.anothervolt.co.uk/ERA2/html/summary.php
Go
Google Search 119 blocked Check AutoLink Options
ERA

Menu

Destroy Session

Jump to Section

Initial Information

This section of ERA creates a summary of the paper which you are reviewing, the text responses from this section will form the bulk of the final review and your answers will affect which questions you are asked in the future. Please complete all appropriate fields.

Does this paper describe a technique or techniques?
Yes ○
No ●

Does this paper describe a system?
Yes ●
No ○

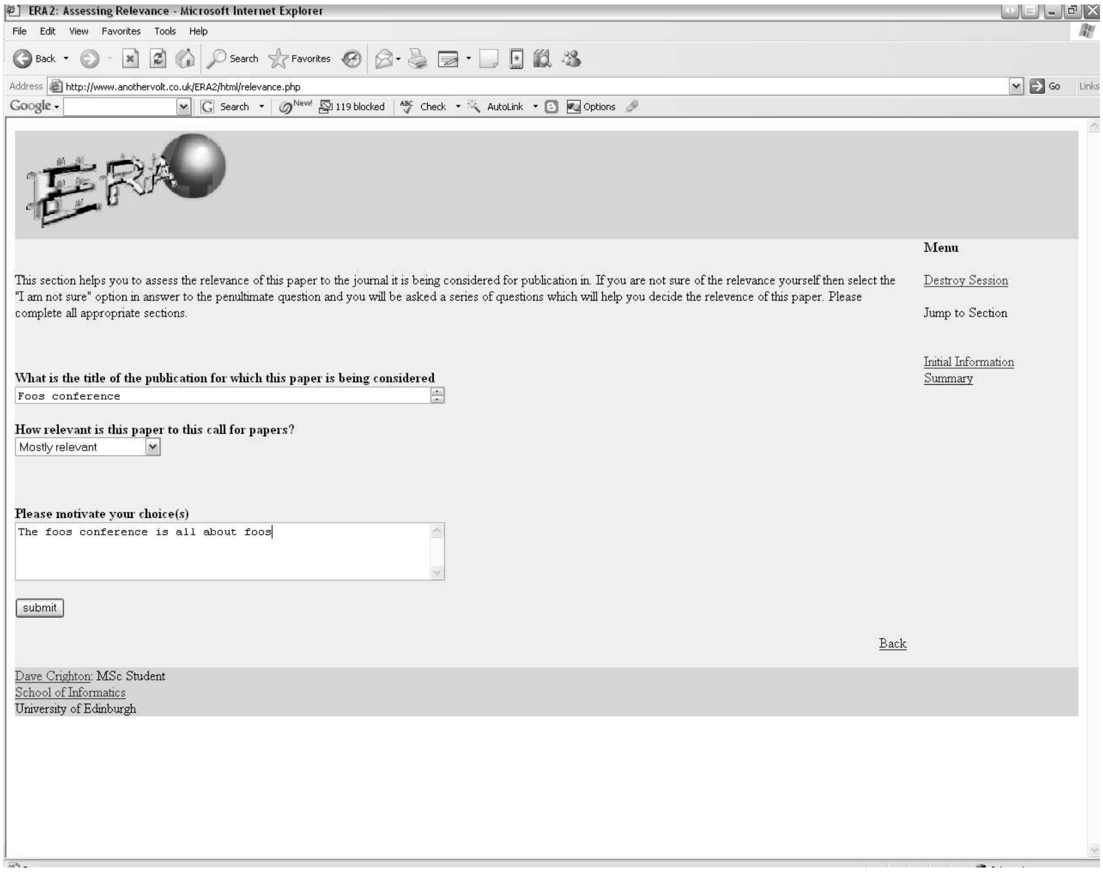
What is this system called?
Foo System

Is the system built by combining previous tools, updating a previous system or is it completely new?
It is totally new

What advantages does this system offer over competing approaches?
Lots of them

Please enter a brief description of the system
Foo system is an advanced system for collecting foos in the wild

Does this paper identify a new problem to be tackled?


ERA2: Assessing Relevance - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.anothervolt.co.uk/ERA2/html/relevance.php
Go Links
Google Search 119 blocked Check AutoLink Options
ERA

Menu

Destroy Session

Jump to Section

Initial Information
Summary

This section helps you to assess the relevance of this paper to the journal it is being considered for publication in. If you are not sure of the relevance yourself then select the "I am not sure" option in answer to the penultimate question and you will be asked a series of questions which will help you decide the relevence of this paper. Please complete all appropriate sections.

What is the title of the publication for which this paper is being considered
Foos conference

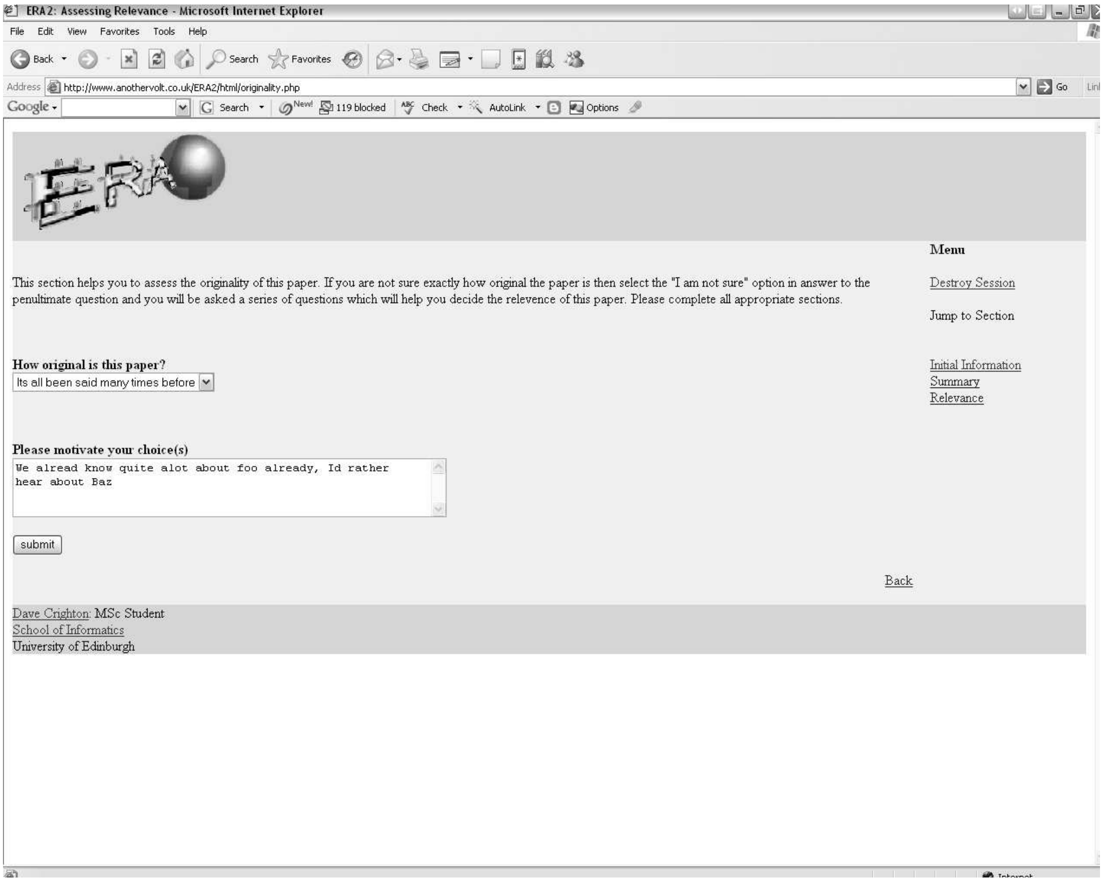
How relevant is this paper to this call for papers?
Mostly relevant

Please motivate your choice(s)
The foos conference is all about foos

submit

Back

Dave Crighton: MSc Student
School of Informatics
University of Edinburgh


ERA2: Assessing Relevance - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Go
Google Search 119 blocked Check AutoLink Options
ERA
Internet

Address http://www.anothervolt.co.uk/ERA2/html/originality.php

Menu

This section helps you to assess the originality of this paper. If you are not sure exactly how original the paper is then select the "I am not sure" option in answer to the penultimate question and you will be asked a series of questions which will help you decide the relevence of this paper. Please complete all appropriate sections.

Destroy Session

Jump to Section

Initial Information
Summary
Relevance

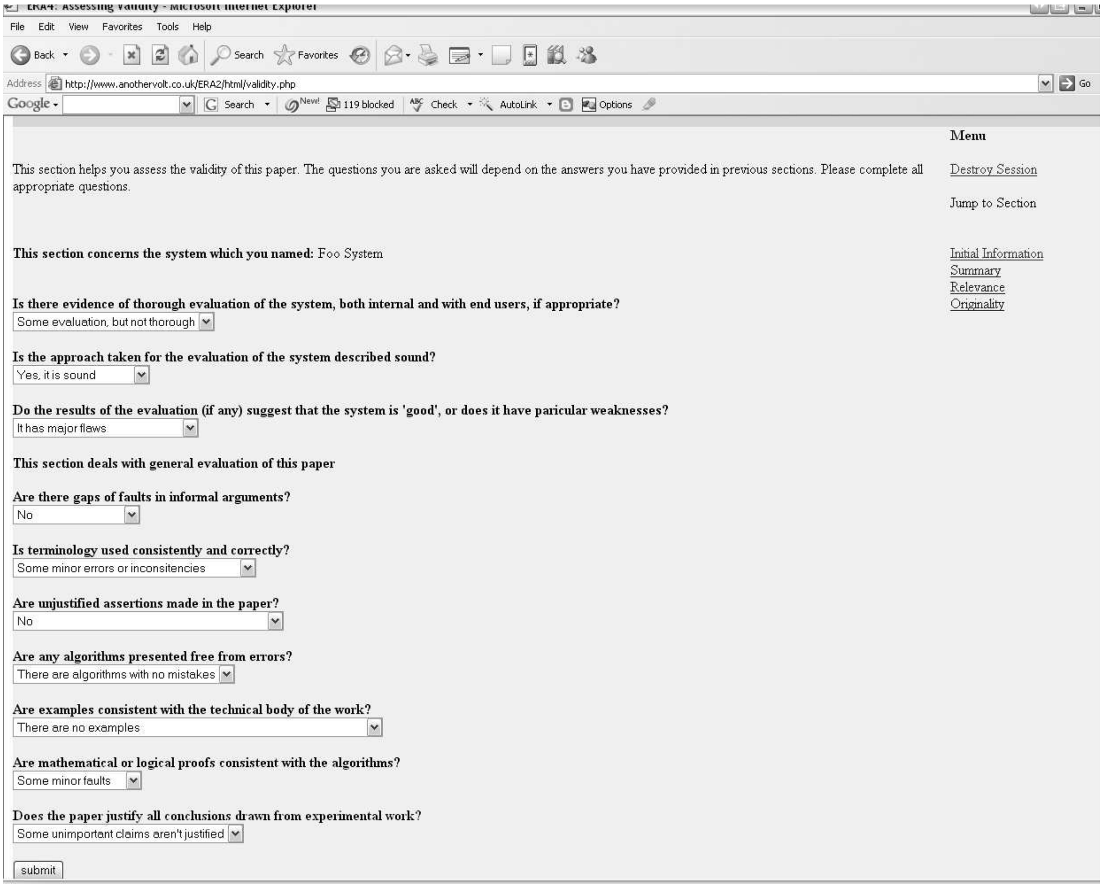
How original is this paper?
Its all been said many times before

Please motivate your choice(s)
```
We alread know quite alot about foo already, Id rather
hear about Baz
```

submit

Back

Dave Crighton: MSc Student
School of Informatics
University of Edinburgh


ERA4: Assessing Validity - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Go
Google Search 119 blocked Check AutoLink Options

Address http://www.anothervolt.co.uk/ERA2/html/validity.php

Menu

This section helps you assess the validity of this paper. The questions you are asked will depend on the answers you have provided in previous sections. Please complete all appropriate questions.

Destroy Session

Jump to Section

Initial Information
Summary
Relevance
Originality

This section concerns the system which you named: Foo System

Is there evidence of thorough evaluation of the system, both internal and with end users, if appropriate?
Some evaluation, but not thorough

Is the approach taken for the evaluation of the system described sound?
Yes, it is sound

Do the results of the evaluation (if any) suggest that the system is 'good', or does it have paricular weaknesses?
It has major flaws

This section deals with general evaluation of this paper

Are there gaps of faults in informal arguments?
No

Is terminology used consistently and correctly?
Some minor errors or inconsitencies

Are unjustified assertions made in the paper?
No

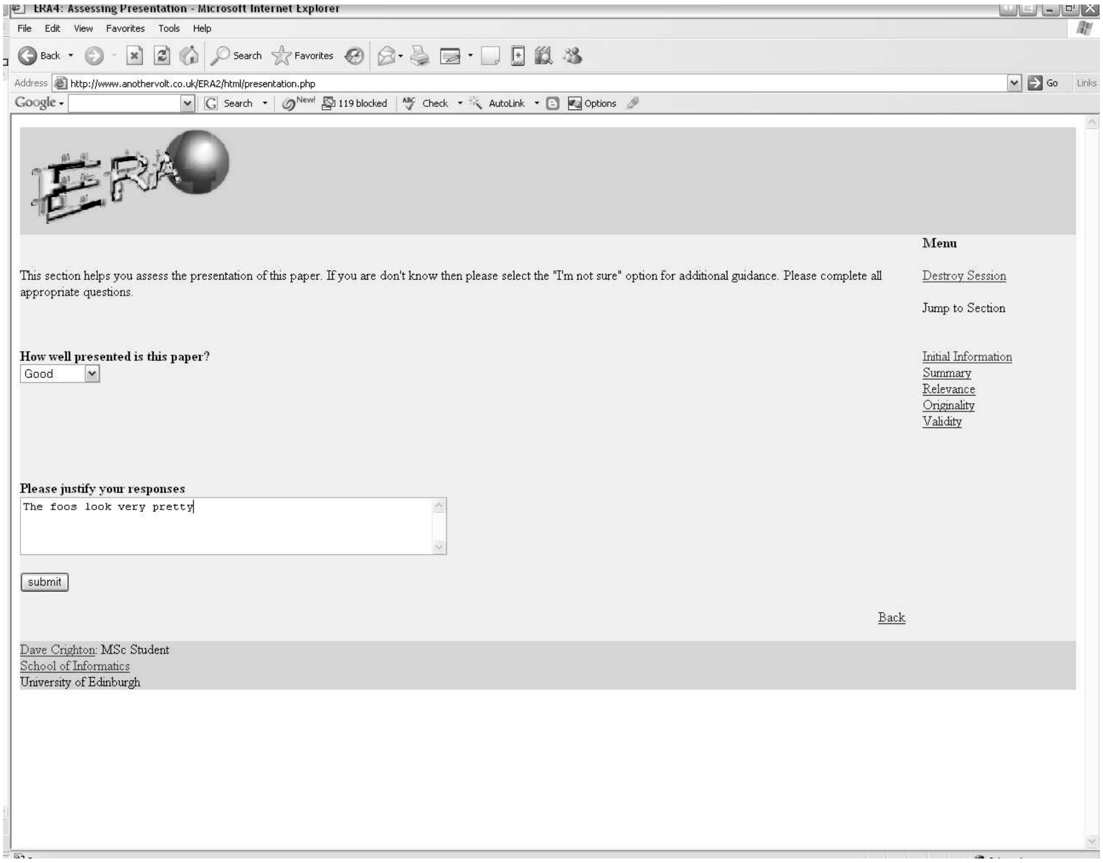
Are any algorithms presented free from errors?
There are algorithms with no mistakes

Are examples consistent with the technical body of the work?
There are no examples

Are mathematical or logical proofs consistent with the algorithms?
Some minor faults

Does the paper justify all conclusions drawn from experimental work?
Some unimportant claims aren't justified

submit


ERA4: Assessing Presentation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.anothervolt.co.uk/ERA2/html/presentation.php
Go Links
Google Search New! 119 blocked Check AutoLink Options
ERA

Menu

This section helps you assess the presentation of this paper. If you are don't know then please select the "I'm not sure" option for additional guidance. Please complete all appropriate questions.

Destroy Session

Jump to Section

How well presented is this paper?
Good

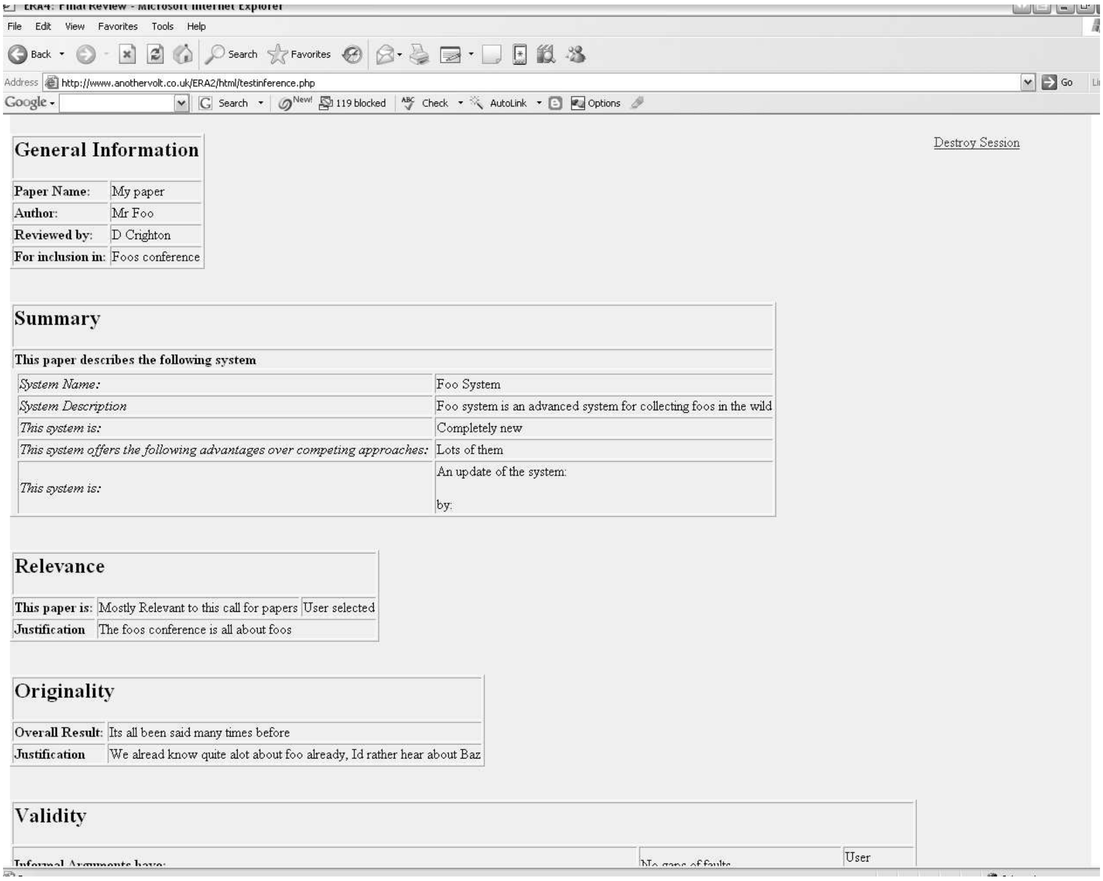
Initial Information
Summary
Relevance
Originality
Validity

Please justify your responses
The foos look very pretty

submit

Back

Dave Crighton: MSc Student
School of Informatics
University of Edinburgh


ERA4: Final Review - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.anothervolt.co.uk/ERA2/html/testinference.php
Go
Google Search New! 119 blocked Check AutoLink Options

Destroy Session

General Information

Paper Name:	My paper
Author:	Mr Foo
Reviewed by:	D Crighton
For inclusion in:	Foos conference

Summary

This paper describes the following system

System Name:	Foo System
System Description	Foo system is an advanced system for collecting foos in the wild
This system is:	Completely new
This system offers the following advantages over competing approaches:	Lots of them
This system is:	An update of the system: by:

Relevance

This paper is:	Mostly Relevant to this call for papers	User selected
Justification	The foos conference is all about foos	

Originality

Overall Result:	Its all been said many times before
Justification	We alread know quite alot about foo already, Id rather hear about Baz

Validity

Informal Arguments have:	No gaps of faults	User

Appendix B

ERA Version 4 reviews

B.1 1

General Information Paper Name: A framework for data-parallel knowledge discovery in databases Author: Alex A Freitas and Simaon H Lavington Reviewed by: Marc Roberts For inclusion in: IEE Colloquium on Knowledge Discovery and Data Mining

Summary This paper describes the following system

System Name: Parralel KDD Framework System Description This seems to be a framework where database queries are mapped onto set oriented ̈primitiveöbjects which allow operations to be executed in parralel on a parralel data server. This should allow an increase in efficiency as different parts of the query can be run in parralel This system is: Completely new This system offers the following advantages over competing approaches: The parralel KDD algorithms acheived a linear speed up over their sequential counterparts. This system is: An update of the system:

by:

This paper makes the following hypotheses or claims Hypothesis Name Performance hypothesis Hypothesis Description The parralel version give a roughly linear speed increase This hypothesis is supported by the following evidence The experimental evidence consists of running the two primitives (whatever they are) doing two different data mining operations (both versions are variants of TDIDT) over large databases Hypothesis Name generality Hypothesis Description The primitives can calculate any candidate rule measures, the examples given are Information Gain, Information Gain Ratio, Reduction of Gini Diversity Index, J-measure, Chi-squared and Cramers V coefficient This hypothesis is supported by the following evidence There seems to be no evidence to support this claim

Relevance This paper is: Definately Relevant to this call for papers User selected Justification This paper is explicitly about data mining and knowledge discovery. Its even in the title

Originality Has this paper been published before? Not to my knowledge User Selected Has similar work been published before? Yes User Selected How many similar systems exist? Many User Selected Justification A google search has turned upbooks on amazon on the same subject as well as numerous papers

Validity Informal Arguments have: Many or major gaps or faults User selected Terminology is used: Poorly with major inconsistencies or inaccuracies User selected Are there unjustified assertions present?: Major assumptions are not justified User selected Are any algorithms presented free from errors?: There are no algorithms User selected Are examples consistent with the technical body of the work?: There are no examples User selected Mathmatical or logical proofs are: User selected Does this paper justify all conclusion drawn from experimental work?: Major claims are unjustified User selected System Is there evidence of thorough evaluation of the system(, both internal and with end users, if appropriate?: Some evaluation but not thorough User selected Is the approach taken in evaluation the system (Parralel KDD Framework sound?: No, it is unsound User selected Do the results of the evaluation (if any) suggest that the system (Parralel KDD Framework is ’good’, or does it have paricular weaknesses?: It has major flaws User selected

Significance Survey System This system is: Totally new User selection

Presentation Overall Result: Poor User selected Justification The presentation is apalling with random phrases turned into acronyms for no apparent reason. The results are presented in the middle of the main text and dont appear to support the conclusion. Huge chunks of explanation are omitted and the reader is left guessing as to what most of the terms used are.

Back Menu

Destroy Session Dave Crighton: MSc Student School of Informatics University of Edinburgh

B.2 2

General Information Paper Name: The Personal Effect: Affective Impact of Animate Pedagogical Agents Author: Lester, Converse, Kahler, Barlow, Stone and Bhogal Reviewed by: John Henry For inclusion in: Proceedings of the SIGCHI conference on

Human factors in computing systems

Summary This paper describes the following techniques Technique 1 Name Animated Pedagogical agent for educational software Is a solution to the problem: It doesnt really solve a problem rather it aims to enhance multimedia learning environments Is a solution to the problem: To a certain extent this technique models a personal tutor This technique is: An an application of an existing technique to the new problem of: Animated agents have been used in other environments, for example Im sure everyone is familiar with Clippy! The application of a avatar to explain and provide advice in eductional is probably not new but it may not have been studied thoroughly in an academic setting before. This paper makes the following hypotheses or claims Hypothesis Name Improves Learning Hypothesis Description Using animated pedagogical agents increases the ability for children to learn in virtual microworlds. This hypothesis is supported by the following evidence The children were asked to complete a test showing comprehension and the ability to solve problems in the domain both before and after using the system. However this is fundamentally flawed by the failure to provide a control set in which there is no animated agent. Coouldnt the fact that the children just spent an hour experimenting with the domain be responsible for the increases shown in learning? Also the use of 100 subjects is quite small for a study of this sort. Can significant statistical conclusion really be drawn from this study? Hypothesis Name Motivation hypothesis Hypothesis Description Using animated pedagogical agents motivates children to learn. They find the agent useful and entertaining. This hypothesis is supported by the following evidence A questionaire was administered to the test subjects after completing their task on the system. This questionaiire guaged how useful and entertaining the children found the agent. Strangely even the agent which provided no advice was considered helpful and useful by the children

Relevance This paper is: Definately Relevant to this call for papers User selected Justification The paper is clearly relevant

Originality Overall Result: One step ahead of the pack Justification There has been lots of papers about animated agents in software but I havent́ read one with an extensive study into software aimed at primary school children

Validity Informal Arguments have: Many or major gaps or faults User selected Terminology is used: Consistently and correctly User selected Are there unjustified assertions present?: No User selected Are any algorithms presented free from errors?: There are no algorithms User selected Are examples consistent with the technical body of the work?: There are examples which are consistent with the technical body of the work User selected Mathmatical or logical proofs are: User selected Does this paper justify all conclusion drawn from experimental work?: Yes User selected Techniques

Significance Technique 1 This technique is: Quite widely applicable User selection Survey

Presentation Overall Result: Good User selected Justification The paper is easy to read and easy to understand

Back Menu

Destroy Session Dave Crighton: MSc Student School of Informatics University of Edinburgh

B.3 3

General Information Paper Name: M̈y hairiest bugẅar stories Author: Marc Eisenstadt Reviewed by: Jonathan Betts For inclusion in: n/a

Summary This paper makes the following hypotheses or claims Hypothesis Name Dimensions of Interest Hypothesis Description The classification of bugs can be divided into three main areas: a)why the bugs were difficult to find, b) how the bugs were found, c) root causes of bugs This hypothesis is supported by the following evidence The author performed and online survey asking for respondants to give ẅar storiesöf their worst bugs. The author then inferred these improtation d̈imensions of interestf̈or himself. Hypothesis Name Domination of certain Dimensions Hypothesis Description The dimension of why a bug is difficult is dominated by chasms in cause and effect adn the fact that some bugs preclude the use of debuggin tools. The dimension of how a bug was fixed was dominated by inserted print statements, data gathering and hand simulation. This hypothesis is supported by the following evidence The results of the authors classification of a collection of anecedotes provided by respondants on online bulletin boards and discussion groups.

Relevance This paper is: Definately Relevant to this call for papers User selected Justification n/a

Originality Overall Result: Yet another paper about... Justification The author of the paper seems to conclude in many places that his work merely confirms that done in more specific papers in the past.

Validity Informal Arguments have: No gaps of faults User selected Terminology is used: Consistently and correctly User selected Are there unjustified assertions present?: Major assumptions are not justified User selected Are any algorithms presented free from errors?: There are no algorithms User selected Are examples consistent with the technical body of the work?: There are no examples User selected Mathmatical or logical proofs are: Correct and consistent with any algorithms presented User selected Does this paper justify all conclusion drawn from experimental work?: Major claims are unjustified User selected

Significance Survey

Presentation Overall Result: Average User selected Justification The paper has excessive use of author created terminology when no such terminology was neccesary. The paper frequently refers to sources by reference number or author only, eg Ï undertook in [3]ör d̈escribed in Fryś paper.̈ Figures named as tables were not presented in a tabular form, just rows. In D̈imension 2: How foundẗhe author asserts there were four major bug-catching techniques and then goes on to list 10. Sizing of titles and subtitles is erratic.

Back Menu

Destroy Session

Dave Crighton: MSc Student School of Informatics University of Edinburgh

Appendix C

Hand Reviews

C.1 1

Report Upon A Framework for Data-Parallel Knowledge Discovery in Databases.

Validity The experimentation took the form of performing a particular knowledge extraction method on a small range of occasionally synthetic databases and real world datasets. Conclusions were drawn (it is implied but not stated) by comparing the performance of the knowledge extraction method performed on a multiple processor database server against a single processor server. However it is only stated that each of the multiple processors have ”roughly the same MIP rate” as the single processor system. It is inclear that any performance gains are the result of the parallelism introduced by the team. Furthermore no comparisons are drawn between the proposed parallelisation framework and existing solutions.

Significance of the Work The paper describes a generalized framework for performing knowledge extraction methods in parallel upon a data base. This framework is based upon primitives which the paper outlines should satisfy certain principles. Other than this the paper gives no specifics on the implementation of the primitives or the ways in which they should be utilised. As such the proposal seems to amount to little more than the suggestion that parallelism will be enhanced by the provision of subtasks which can be executed in tandem on the server side.

Originality For the reasons outlined above the paper brings little that is original.

Quality of Presentation

Organisation The paper lacks a separate abstract detailing the achievements of the work. There are instances of terminology and acronyms used but never defined. The bibliography and attribution are adequate however whilst references are made to the area of Knowledge Discovery algorithms, no references are made to other works or frameworks relating to the main subject of the paper. In individual segments the general dicussion, experimentation methodology and results are all presented in one block of text. No real conclusions seem to have been reached.

Readability Readability is poor due to inadequate separation of subsections as described above. All results are provided as inline text rather than tables making readability low. Generally there seems to be a lack of conclusion, conviction and direction to the paper.

English There did not appear to be any spelling or grammatical errors.

Overall Evaluation of the Paper The paper presents little content, conclusions or original elements. On top of this the only conclusions regarding supposed speed increases seem to be based (it is not explicitly stated) on runs of the same method on two entirely different systems. There is no comparison to other parallelisation techniques. As some of the problems presented related to inherently parallel tasks to perform upon the database, it is unclear what advantage the proposed system actually brings over simply performing normal serial tasks simultaneously.

I would not recommend this paper for inclusion.

Referee's Confidence in the Paper's Subject 2

Jonathan Betts

C.2 2

Review of: The Persona Effect: Affective Impact of Animated Pedagogical Agents, Lester et al.

Marc Roberts

Validity

The experimental procedure was in general very well thought out and analysis was thorough but there was no control group offered which had no pedagogical agent. This means that the hypothesis that using pedagogical agents in educational software improves the learning experience can not be fully tested. Rather this paper is comparing the effectiveness of various types of agent.

Another possible problem is that children answered that Herman the Bug was useful and helpful even when the muted agent was running. This is probably because the agent was still allowed to introduce the problem and describe how to use the software etc. While this might well be helpful to the children it does not really address the hypothesis which is being tested.

Significance

Despite some of the limitations of the method I think using pedagogical agents will see an increase for exactly the reasons described in the paper. Computers are now powerful and ubiquitous enough for this kind of agent to become widespread. This paper attempts to analyse the beneficial effects of using such agents and as such is a useful work for the field.

Originality

It seems logical that similar studies would exist on the effects of pedagogical agents but this one is highly specific to children and educational software. This is not a revolutionary, groundbreaking paper but it still makes an original and useful contribution to the field.

Appropriateness for: Proceedings of the SIGHI conference on Human Factors in computing systems.

The described Persona effect is definitely a Human Factor in a Computer System so this paper is wholly appropriate for inclusion.

Presentation

The organisation is clear with well defined abstract, introduction and conclusion sections. Data is presented in an intuitive way through the use of tables.

The paper is very readable with easily understood arguments and no cryptic or obtuse sentences

The written English is good with no noticeable grammatical or spelling errors

Confidence

This paper is not within my field of research but I feel qualified to give an opinion so the score is 2.

Overall

I feel that this paper should be accepted for reasons outlined above.

C.3 3

Paper Review: ”My hairiest bug” war stories Marc Eisenstadt

Reviewed by: John Henry

This paper is quite an informal exploratory account of bugs in the field but its validity does have some concerns. There was no discussion of why the bugs which were considered trivial were rejected for inclusion and their triviality could be highly subjective. There is also no effort to verify that the bugs have actually occurred rather anecdotal evidence from complete strangers on the internet is used as data. Some of the participants are identified as prominent members of different groups but this still doesn't preclude that the data isn't really that reliable.

The work doesn't appear to be very significant. Its main hypothesis seems to be that better debugging tools would be better which is pretty trivial. Similarly it is unclear how the suggested bug repository could be generalised enough to provide any help.

This work is original in that an analysis of anecdotal bug evidence has not been attempted before, however there are plenty of lists of anecdotal evidence available on the web. Also the paper itself identifies cases where bugs have been analysed in a more formal and theoretical way.

The paper seems relevant for inclusion in the Communications from ACM publications. This publication deals with computing from a distinctly software engineering angle and so this article would fit in well.

The material is quite well organised although its publication on the web may have distorted its original appearance somewhat. There are some inconsistencies in placement and sizes of headings for example but despite this the organisation is perfectly adequate. The written English is clear and concise, the only slightly obtuse section is the bit which describes the 2D grid used to draw conclusions about data-gathering cases where there is a large gulf between cause and effect.

This paper overlaps my area of expertise so I would give myself a confidence score of 4.

Accepting this paper would be purely dependant on the strength of other entries. It is adequate for inclusion however it should not hold a spot which may be used to publish a stronger contender.

Appendix D

ERA Version 3 Reports

D.1 1

ERA

IRM PAPER REVIEW 2004-05

Referee ID John Henry Paper ID A framework for data-parallel knowledge discovery in databases Author ID A A Freitas and S H Lavington

it is unclear whether the paper set out to test an existing hypothesis, or the hypothesis was formed during the investigation. Claims Two parralel data mining primitives perform better when implemented on parralel hardware than on sequential harware The paper supported these claims with experimental evidence Evidence The experimental data is runtime based on sample tasks computing different measures. The tasks were performed over some large training databases Summary This paper describes two data-mining primitives which are part of a framework for Knowledge Discovery. It doesnt discuss further what these primitives are

This paper describes a technique called Rule Induction Primitive and Instance Based Learning Primitive . This technique tackles a problem called data-parralel knowledge discovery . Some rival techniques for the same problem are , and .

(Marks on a scale 1-4, with 1 being the best)

Section Mark Motivation Relevance (1) Definitely relevant The publication is a conference on Knowledge discovery so the paper is quite obviously relevant.

The paper is relevant to the following keywords:

Computer Science

Originality (1) Trailblazing I am unaware of any other work in this field. Significance (1) Highly significant The problem tackled is only moderately difficult Only some people are likely to be interested in this work The work will probably not have much of an impact No actual work has been carried out yet The results are unnecessarily complicated, inefficient, wrong, impractical or limited in application Validity (1) Impeccable The biggest error is that the two machines compared are asserted as being equivelent with no source or justification Presentation (4) Poor The presentation is terrible, the over use of abbreviations like Sp for speed factor make it very difficult to read this paper Overall (4) Clear reject Its two hard to make sense of. Possibly a weak accept subject to being rewritten for clarity

---

Corrections These problems with the originality of the paper were identified:

These problems with the Significance of the paper were identified:

The problem tackled is only moderately difficult Only some people are likely to be interested in this work The work will probably not have much of an impact No actual work has been carried out yet The results are unnecessarily complicated, inefficient, wrong, impractical or limited in application

These problems with the Validity of the paper were identified:

The evaluation of the technique is insufficiently thorough The technique's evaluation is entirely unsound There are some minor gaps in informal arguments There are some minor unjustified assertions Some minor concludsions drawn from experimental work aren't justified fully

Referee Confidence (3) Somewhat uncertain Comments

Save Review - Start Over

Please take a moment to fill in this questionnaire. v3.0 developed by Brian Hutchison for the University of Edinburgh. Based on V 2.0 by Massimo Caporale.

D.2 2

IRM PAPER REVIEW 2004-05

Referee ID Jonathan Betts Paper ID The Persona Effect: Affective Impact of Animated Pedagogical Agents Author ID Jace C. Lester, Sharolyn A. Converse, Susan E. Kahler, S. Todd Barlow, Brian A. Stone, Ravinder S. Bhogal

---

The paper set out to test an existing hypothesis. Claims The inclusion of a pedagogical agent in learning environment systems greatly increases the sutdents learning and enjoyment of that environment. The paper supported these claims with theoretical evidence The paper supported these claims with experimental evidence Evidence The main thrust comes from a study of 100 students reactions to different variants of the agent and from tests of the students performance before and after experiencing the environment. Some conclusions are drawn from the writers own theories such as 'there may be a motivational effect' and others drawn from other reading. Summary The paper was concerned with the effect upon the learning experiences which can be enhanced by the presence of a lifelike animated character. The paper further goes on to claim that the experience is indeed greatly enhanced by the presence of such a character even if the character provides little or no feed back.

This paper describes a technique called The inclusion of an animated agent into an interactive learning enviroment . This technique tackles a problem called Enhancing enjoyment and learning in the enviroment. . Some rival techniques for the same problem are I'm sure there are many but I don't know them. , and . This technique is a combination of Animated Agents and Iteractive Learning Enviroments . This technique models The agent attempts to represent a believable life-like character. In as far as this it is attempting to mimic the appearance and communication between people. .

(Marks on a scale 1-4, with 1 being the best)

This paper describes a system called DESIGN-A-PLANT . Section Mark Motivation Relevance (1) Definitely relevant n/a

---

The paper is relevant to the following keywords:

Cognitive Science Artificial Intelligence Computer Science

Originality (h6) 0 The paper is a thourough discussion of a topic presenting interesting conclusions. Originality (h5) 0 The paper is a thourough discussion of a topic presenting interesting conclusions. Originality (h4) 0 The paper is a thourough discussion of a topic presenting interesting conclusions. Originality (2) One step ahead of the pack The paper is a thourough discussion of a topic presenting interesting conclusions.

OVERALL

The following are the marks you allocated in each of the previous 5 sections :

(NOTE: The marks are on a scale from 1 to 4, with 1 being the best)

Section Mark Relevance (1) Definitely relevant Originality (h6) 0 Significance (2) Significant Validity (2) Contains minor errors Presentation (1) Excellent

Total points : ——————————————————

werror

[ARGACCES5] Function + expected argument 2 to be of type integer or float [PRCCODE4] Execution halted during the actions of defrule overall.

No doubt you will be achingly pleased to know that after getting to the validity section of teh ERA system it spat out some error. I have included the report it forwarded me onto and the sections that it didn't complete in the report which I had added to the web forms. Not sure what you want to do with this.

I would have recommended the paper, I would also have put my ability to review it as low but not minimal.

Significance

2) I am unable to compare with rival techniques however as far as the area of reserach which this paper is concerned with ,Assisted Learning , this paper could be influential.

Validity 2) Contains Minor Errors

Spelling error in Question 7 Figure 3 'werre'.

The paper would have been enhanced by the inclusion of a baseline system which contained no agent in order to fully substantiate the claims that improvements of the learning performance were truely effected by the presence of the agent.

D.3 3

IRM PAPER REVIEW 2004-05

Referee ID Marc Roberts Paper ID My hairiest bug war stories Author ID Marc Eisenstadt

---

The paper formed a hypothesis through an exploratory investigation. Claims An online database of bugs would be a sueful tool and might be easy to assemble as programmers are found to be forthcoming with stroies about bugs. A large percentage of difficult bugs occur when there is a large gap between the cause of the bug and the effect of the bug. The paper supported these claims with experimental evidence Evidence The evidence is anecdotal but is most analagous to experimental analysis. Summary This was clearly an exploratory paper

This paper describes a new problem called Building an online database of bugs or developing tools which can use data about difficult bugs to produce better debugging tools . This paper is a survey. It covers Debugging anecdotes .

(Marks on a scale 1-4, with 1 being the best)

Section Mark Motivation Relevance (2) Mostly relevant Assuming that the publication in question is for the Association for Computing Machinary it is difficult to assess the relevance since the Comm publication no longer seems to be offered and ACM publish a long list of different jjournals. In the absence of further data I have classified this as Mostly relevant to give the paper the benefit of the doubt.

The paper is relevant to the following keywords:

Artificial Intelligence Computer Science

Originality (h3) 0 How is this really any different from the bug tracker databases which already exist for many software products? Originality (4) It's all been said many times before How is this really any different from the bug tracker databases which already exist for many software products?

Close this window These problems with the Significance of the paper were identified:

These problems with the Validity of the paper were identified: