Evaluation Process for GraPE (A Web-Based Expert System for Reviewing Grant Proposal)

Zeratul Izzah Mohd Yusoh, Fakulti Teknologi Maklumat dan Komunikasi, Kolej Universiti Teknikal Kebangsaan Malaysia, 75450 Ayer Keroh, Melaka, Malaysia
Tel: +6-06-2333039, Fax: +6-06-2332508, Email: zeratul@kutkm.edu.my

Abdul Samad Shibghatullah, Fakulti Teknologi Maklumat dan Komunikasi, Kolej Universiti Teknikal Kebangsaan Malaysia, 75450 Ayer Keroh, Melaka.
Tel: 06-2332510, Fax: 06-2332508, E-mail: samad@kutkm.edu.my

Prof Alan Bundy, Centre of Intelligent Systems and their Applications, School of Informatics, University of Edinburgh, Edinburgh EH8 9LE Scotland, United Kingdom.
Tel: +44-131-650-2716, Fax: +44-131-650-6513, Email: a.bundy@ed.ac.uk

Dr Stephen Potter, Centre of Intelligent Systems and their Applications, School of Informatics, University of Edinburgh, Edinburgh EH8 9LE Scotland, United Kingdom
Tel: +44-131-650-2753, Fax: +44-131-650-6513, Email: stephenp@inf.ed.ac.uk

ABSTRACT

A reviewing process of grant proposals is of paramount importance and tiresome for a referee. It excessively consumes time in reading and analysing the content. However, there is no exact methodology to help and assist referees in the evaluation process. The objective of this project is to build a web-based expert system to assist beginner referees in this process. The system is based on ERA (Electronic Referee Assistant), a knowledge-based advisor for Informatics research papers, and named GraPE (Grant Proposal ERA). GraPE aims to provide guidance to beginner referees to help them make a better review and become better reviewers. This paper will describe GraPE and the system evaluation that has been carried out in order to confirm the hypothesis proposed in this project

Keywords: Expert System, Electronic Reviewing, Grant Proposal, System design.

INTRODUCTION

The objective of this project is to build an expert system that will act as a referee advisor, assisting referees of Informatics grant proposals. GraPE is based on a hierarchical structure of questions that a referee must answer when reviewing a grant proposal. There are several sections, each corresponding to a key aspect of a proposal, and having the goal of assessing the quality of that aspect of the proposal under review. If referee is initially unable to make this assessment, then it is decomposed into assessments of a number of more basic aspects and so on, a level is reached where the referee can respond. In this way, GraPE will ensure that the referee covers all the important issues in reviewing a grant proposal. Then GraPE will provide a recommended assessment based on the answers given by referee to all the questions.

The system is based on ERA (Electronic Referee Assistant), a knowledge-based advisor for referees of Informatics research papers developed by [1]. The principle source of expertise for ERA is from a paper written by [2]. Most of the modules in ERA are based on guidelines.
provided in the ‘Note to CADE-12 Referees’ [3]. While ERA has been successful in providing assistance in refereeing papers for workshops, conferences and journals, it cannot be applied to the process of refereeing Informatics grant proposals. This assistance is to be provided by the system developed during this project. GraPE will act as a referee advisor, assisting referees of Informatics grant proposals.

Expert system technology is the appropriate choice for this kind of system [4]. An expert system is designed to be a ‘specialist’, focusing on solving problems in a specific domain. Through expert system technology, the decision-making ability of a human expert can be emulated, and in this project, it is hoped that the system developed can emulate the methodology (and, to a certain extent, decision-making) of experts in the task of refereeing grant proposals. The study and development of expert systems, which is an area in Artificial Intelligence, was first introduced in the late 60s [5]. Expert systems have been implemented to solve a wide range of problems such as planning, diagnosis, monitoring, design and many more [4][6]. Among the domains that use expert system to solve problems are medicine, engineering, mathematics, business, computer science and education [5][6][7][8].

GraPE was built using the WebClips expert system shell. CLIPS (C Language Integrated Production System) is an expert system programming language that supports three common paradigms of expert systems, that is rule-based, object oriented, and procedural [3][4]. It was designed at the NASA/Johnson Space Center specifically to provide expert systems technology that can fit within NASA’s operational computing environment. All the features that CLIPS offers make it suitable for the system developed in this project. WebCLIPS is the version of CLIPS for web-based system development. GraPE is also developed as web-based system. This is for easier access to the system, and makes it available on the Internet, so that users will not have to install the system on their machine before using it. WebCLIPS has all CLIPS features and uses CLIPS syntax, and following CLIPS standard, except WebCLIPS has some embedded HTML command in some of the system’s screen. The HTML tag is for displaying the output, and for system’s interfaces.

The objective of this paper is to describe the system and the evaluation process of GraPE. The evaluation process consists of two parts, that is, evaluating the results of reviewing processes and evaluating feedback from users. This paper has been organised as follows. Section 2 presents an overview on GraPE. Section 3 describes the evaluation plan. Section 4 describes the reviewing process and section 5 presents the feedback from users. Finally, conclusions are presented in section 6.

**OVERVIEW OF GRAPE**

This section describes the design and development of the system, from the adaptation of ERA to the final version of the system. Two prototypes were built during the development of GraPE. The first prototype is the result of adapting ERA to GraPE. Using this prototype, the functionality of GraPE and the control of the system were tested. The second prototype is a more complete version of GraPE, after adding all aspects that were discussed in previous section. Aspects that were added will be described in detail in Section 2.2. In the development phase of both prototypes, there are two crucial stages, namely, the design of the modules of GraPE, and the design and development of GraPE’s control and inferencing processes. Both stages will be described in detail in the following sections.

**Adaptation of ERA modules**

Some modules in GraPE are taken from ERA. Most of the modules in ERA are based on guidelines provided in the ‘Note to CADE-12 Referees’ [2]. There are 8 modules in ERA, and all modules in ERA were adapted for GraPE, with some alterations in the ‘Summary’ and ‘Validity’ modules. This is to tailor the review process of grant proposal, instead of research papers. The ‘Summary’ module in ERA is intended to help referees construct summaries and identify flaws in the argument in the paper in order to get a clearer idea of the essential contents of the paper. Referees of grant proposals also need to do this, but it is not obligatory to construct a summary. So, the ‘Overview’ module is created, to replace the ‘Summary’ module. The ‘Overview’ module not asking referees to construct a summary of the proposal, but it collects information about the background and the basic idea of the proposal, and this information is saved for use in later modules.

The ‘Validity’ module in ERA is used to highlight errors in the paper, and depends on the nature of the paper itself. For example, any algorithm stated in the paper should be bug-free and expressed using consistent terminology, and an experimental paper should have some justification for its results. For grant proposals, since the result of the research is yet to be known, a ‘Validity’ module is not suitable for inclusion in the system. However, it is important for each proposal to propose a sound methodology that should lead to
valid results. So, the ‘Validity’ module was slightly altered and replaced by another module, called ‘Method/Technique Used’, intended to review the appropriateness of method/technique proposed.

GraPE has the rest of ERA’s modules, that is, ‘Relevance’, ‘Originality’, ‘Significance’, ‘Presentation’, ‘Overall’ and ‘Comment & Corrections’. The ‘Originality’ module and ‘Significance’ module were renamed to ‘Novelty’ and ‘Impact’ respectively. This is to standardize the terminology used in GraPE with the referee forms provided by funding agency, for example, EPSRC and NERC (Natural Environment Research Council). The basic functionality of each module adapted is still as same as it was in ERA, namely to rate the proposal in that aspect. However, the questions featured in the sections will be different. This is to tailor the process to reviewing grant proposals instead of research papers.

Additional Modules

There are also some additional modules that are important for reviewing grant proposals but not for reviewing research papers. The additional modules are created because they correspond to additional criteria that will be applied by the referee of any funding agency when they are asked to review proposals. The additional modules are: Proposer’s Background, Project Management, Resource Management, Collaboration, and Dissemination Strategy.

a) Proposer’s Background: This module asked users to rate the ability of the proposer to solve the problem proposed. The questions in this module cover the proposer’s track record, area of expertise, and contribution in related areas.

b) Project Management: The Project Management module assessed how appropriate the project management plan that has been specified in the proposal is. This should include realistic timescales and milestones, and distributions of tasks among group members.

c) Resource Management: This module is to check whether the resources requested by the proposer are admissible and relevant to the proposal. The objective of the proposal should be to justify the resources requested.

d) Collaboration: Some proposals might have identified collaborators. This module is to rate the strength of the collaborations, and the contribution of the collaborators to the proposal.

e) Dissemination Strategy: In this module, arrangements for dissemination will be rated, assessing whether the dissemination plans are likely to be effective or not.

The control of GraPE

GraPE will begin with a start page where the introduction to and instructions for GraPE will be displayed. The reviewing process will start with the Overview module, and be followed by the other modules. When the process is complete, GraPE will display the report of the review, and users can start another review, or exit the system. Figure 1 shows the flow of GraPE. The input and information from users will be stored in machine’s memory for later use in the inference process. When the review process is completed, all the facts stored will be retrieved in order to display the report of the review.
EVALUATION PLAN

The evaluation plan consists of two parts, that is, evaluating the results of reviewing processes and evaluating feedback from users. For both parts, four different people have been recruited to be the ‘referees’ of grant proposals, and they have to fill in the feedback form after completing the reviewing process. All the testers are postgraduate students, and none of them had prior experience in refereeing a grant proposal, nor in writing a proposal. The testers had evaluated two proposals, one using the EPSRC form, and the other using GraPE. The proposals used in this evaluation are taken from the Informatics Research Methodology (IRM) 2004 coursework. None of the testers has attended this module previously. By using these proposals, the testers had quite a wide range of selection of grant proposals, since they are from different areas of informatics. All grant proposals gathered were listed and the testers chose two proposals to be reviewed in this evaluation process. The result of the evaluation then compared with expert’s evaluation. The expert is a member of EPSRC Peer Review College.

The testers will be divided into two groups, with two persons in each group. The first group will evaluate the first proposal with GraPE, and the second proposal with the EPSRC form. And the second group will use the form to evaluate the first proposal, and use GraPE for the second proposal. This is to try to check the difference of method/style of reviewing between the two groups if any exists, and to analyse whether the style of refereeing in one mechanism has influence in another.

An analysis form was prepared to analyse all the reviews that were submitted by the testers. The analysis form describes all aspects that a referee should take into consideration when reviewing a grant proposal. It is created based on knowledge gathered in the Knowledge Acquisition phase, and based on the guidelines available for proposers in writing a proposal. The form was verified by the expert. It is possible to argue that the analysis form can be biased, more to GraPE than the form. But, the knowledge acquisition activities that have been done in this project include knowledge sources from experienced referees, and various guidelines and article related in this matter.

The results of the analysis of the reviews, and the users’ feedback are described in the following sections.

REVIEWING PROCESS

Each of the testers had evaluated two different proposals, which they have chosen from the list of grant proposals, taken from IRM coursework. They have chosen proposals that are related to their area, or close. They completed a review for each proposal, either using GraPE or the paper form. The time permitted to the testers to complete their evaluation of each proposal was about five days. Altogether, eight reviews were collected, four of them being GraPE’s summary review reports, and the other four completed EPSRC referee forms.

As stated previously, the reports of the reviewing process were to be analysed using the form given, and the results of the analysis will be described in three aspects, that is, the difference between reviews made using GraPE and the reviews made using the form, the comparison between the two groups of testers, and comparison of the marks given the papers (final outcome) by GraPE and marks given the expert. The final outcome from completed EPSRC review form was not included in this comparison process because the final outcome from the form is not in numerical value.

GraPE reviews vs. form reviews

The analysis form contains all the sections that referees should consider during the reviewing process. Based on the analysis, all reviews that have been done using the paper forms have neglected to comment two aspects, namely ‘Proposer’s Background’, and ‘Presentation’. The obvious reason for this is that there are no sections covering those aspects in the form. It can be argued that the two sections are trivial in the reviewing process, but from the Knowledge Acquisition that has been done, ‘Proposer’s background’ has been ranked by experts as among the top five sections that are important in reviewing a grant proposal, while the ‘Presentation’ aspect can affect a referee’s impressions of the proposal. In addition, various articles that are related in this matter have addressed that presentation of a proposal is among the criteria that are important when writing a grant proposal. Among the articles are ‘Writing a good grant proposal’ by Jones and Bundy [2], and an article published at University of Manchester website (www.hop.man.ac.uk) entitled ‘How do you propose?’.

Although not elaborate, one of the features in GraPE that might save some effort is the fixed structure in reviewing process. In GraPE, the users will be guided from one section to the next, and help is given in each section whenever
Comparison of marks

Besides the method of reviewing a grant proposal, the system build in this project is also attempts to infer the overall quality of the proposal, that is, whether it should be funded or not. It would have been interesting to compare GraPE's result with the real result from the funding agencies. However, since all the proposals used in this evaluation process are taken from IRM student coursework, the result of the evaluation will be compared with the final mark of the coursework assessment. GraPE’s final outcome is based on the score calculated for each section. While for the coursework’s assessment, a final mark is given as its final score/outcome based on the assessment in three sections namely ‘Quality of Proposal’, ‘Management of Project’, and ‘Presentation of Proposal’. All the testers have used GraPE at least once, and from the eight reviews that were collected, four of them are GraPE’s review reports. The final score (in percentage value) of GraPE for each GraPE’s review report is shown below, together with the marks from the coursework assessment:

<table>
<thead>
<tr>
<th>Grant Proposal</th>
<th>GraPE’s final score</th>
<th>Coursework’s final score</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>70</td>
<td>71</td>
</tr>
<tr>
<td>P2</td>
<td>76</td>
<td>68</td>
</tr>
<tr>
<td>P3</td>
<td>71</td>
<td>63</td>
</tr>
<tr>
<td>P4</td>
<td>68</td>
<td>49</td>
</tr>
</tbody>
</table>

From the figure we can see that for P1, P2, and P3, the marks given by GraPE are within 10 percent of the mark given in the actual assessment. P1 has a lower mark than the actual assessment, while P2 and P3 each has a higher mark. But for P4, the difference between the two marks appears more significant. If we consider the difference in marks as the measurement of this comparison, the GraPE results look promising with three out of the four results giving a relatively small difference compared to the actual marks. In the other hand, the large difference for P4 cannot simply be ignored.

For P4, based on the final score, GraPE has recommended ‘Fundable (Minor Correction)’ as the final outcome. The tester has rejected the recommendation, and select ‘Fundable (Major Correction)’ as the final outcome. The range of percentage mark for ‘Fundable (Major
Correction) is 30-49%, which is in the range of the coursework’s final score. This shows that the tester and the expert think that the proposal is not very good, but GraPE has inferred the opposite. Based on this result, it is possible to argue that the final outcome for a proposal cannot be determined based solely on GraPE’s final marks. There are some other criteria that should be taken into consideration in deciding whether a proposal should be funded or not, which are not covered by GraPE. P4 is among the examples. The proposal might have score in each section in GraPE, but it might have others criterion that referees think it is not worth to be funded. In the other hands, a grant proposal might have low marks from GraPE’s evaluation but it may have a high commercial value, or the area of the proposal might have little existing funding, hence the proposal should be funded (almost) regardless of GraPE’s result. However, that can be considered as a special case, which might be handled by enhancing GraPE’s functionality in a later version.

USER’S FEEDBACK

Each of the testers filled in a feedback questionnaire about GraPE. The questionnaire consists of 13 questions, which cover the aspects of GraPE’s interfaces, GraPE’s reviewing process, its performance, and the user’s appreciation of GraPE. The analysis of all the feedback collected was done based on the aspects mentioned above. For each question, a statement was made, and testers were to indicate their level of agreement with the statement. All the questions (except Question 8) have five levels of agreement (Strongly agree, Agree, Neutral, Disagree, Strongly disagree), which have been numbered from 1 to 5 respectively. This number will be considered as the score for that question for analysis purposes. Each statement is positive about GraPE, so the greater the degree of agreement with the statement (and the lower the scores), the greater the tester’s satisfaction with the system. Question 8 asks about the ‘Help’ function in the system. The question has two possible answers for testers to choose, that is, ‘Yes’ and ‘No’.

Question 1-2

The first two questions are about GraPE’s interface. The questions, summarized answers, and an analysis of the answers are as follows:

1) The system is easy to use

Most of the testers agree with this statement. However, one of them commented on the lack of explanation of how to use the system. The comment also includes suggestions that GraPE should provide an explanation of the purpose and philosophy underlying the system. This is a good suggestion as there is just a short explanation of the GraPE functionality at the start of the system. First time users might want to know what the aim and objective of GraPE is, and some explanation about its expert system foundations also might be useful. Even though this seems to be trivial and does not affect GraPE’s main functionality, it is still important for the users of the system to know what are the main objectives of the system, so that they can get clearer idea of the system and can make full use of the features offered.

2) The interaction with the system is clear and understandable.

The main issue raised for this statement is about the labelling of the ‘Help!!!’ button. Three out of four testers had commented that they thought the button is just a general help function about how to use the system, instead of helping them to evaluate the section of proposal in question. It is agreed that the label is quite misleading, and the user cannot gain full access to GraPE’s functionality if the label is misunderstood. The ‘Help’ function is actually where much of the inferencing capability of this system lies, and by not using the function, GraPE will not be fully involved in assisting users through the evaluation process. Based on these comments, I have changed the caption for ‘Help’ button to, ‘I’m not sure, please help!’

Question 3-8

The second part of the questionnaire consists of questions about the evaluation process itself. Questions 3.4.5.6.7 have five levels of agreement as described previously, while Question 8 has only two options, that is ‘Yes’ and ‘No’. The question, comments, and analysis of the responses are summarized as follows:

3) The question and information provided are easy to understand.

All testers seem to agree with this statement. However, a tester suggests that there should be more explanation in some sections, but does not mention which sections.

4) The questions asked were generally in sensible order.

The response for this statement was generally good. One of the testers commented that the order of section in GraPE
should be tallied with the order of section in a proposal. Even though this is possible, it’s quite hard to implement, since there is no standard structure for a grant proposal. GraPE’s structure is based on the importance of the sections, where more important sections are put earlier in the process.

5) The possible answers for each question are sufficient.

The main issue raised here is that the possible answers provided in certain sections are not the referee’s desirable answers, but they are forced to select one of those available. Sections that are problematic in this respect are ‘Proposer’s Background’ and ‘Resource Management’. For ‘Proposer’s Background’, there are some proposals that do not include this section at all, but the GraPE structured approach doesn’t allow a referee to skip any sections. This is because, if the user skips any section, there will be missing answers (facts) and this will affect the inference process in later sections. For the ‘Resource Management’ section, one of the testers suggested that time allocation should be included in the sub-question of the section, and GraPE’s possible answers should cater for the case of too many resources requested by a proposal. When developing GraPE, it was thought that all the sections in the system are compulsory for and common to all proposals, but testing revealed that this is not the case. The solution to this issue is to add another possible answer for the question, which is ‘Not applicable’. The question of time allocation has now been included in GraPE, but in ‘Project Management’ section, where the timescale and the milestones of the project are considered (by EPSRC) among the factors when appraising the management arranged for the proposal. And the possible answers provided for that question now cover the case where too much time is allocated.

6) The system covered all the relevant aspects of the reviewing process.

All testers agree with the statement. It is possible to argue that these responses are from inexperienced referees, and experienced referees might have other opinions. However, the knowledge used to develop the system is gathered from experienced referees and Professor Bundy later verified it after the KA activity. So, we can say that GraPE covered relevant, and common aspects of reviewing grant proposals.

7) The system helps me to address important aspects of reviewing a grant proposal.

Most users agree with the statement. Since all of them are inexperienced referee, we can say that the users feel that GraPE is offering them useful assistance.

8) Did you use any of ‘HELP’ option?

Three out of the four testers has used this function. It is possible to infer that the tester who didn’t use the function has misunderstood the ‘Help!!!’ caption. The responses collected from users that had used the function were generally good. They found the function to be helpful, and they can sense the ‘intelligence’ of GraPE when making the recommendations. However, one of the testers gave this comment. ‘Sometimes, pressing ‘Help’ button did help. Nevertheless, the eventual recommendation provided via going through the options provided in ‘Help’ can be considered rigid and does not portray what [was] wanted to be expressed by the reviewer.’

Again, the issue raised here is the rigidity of GraPE. Even though it is agreed that different proposals might need different styles of reviewing, GraPE has been developed based on a widely accepted proposal’s structure along with the results of the Knowledge Acquisition activity. In addition, while the ‘Help’ functions are to guide referees in the process, the recommendations proposed can be rejected, and referees can express their own opinions and give their own results to the system.

**Question 9 - 11**

Question 9 to 11 are about GraPE’s performance. The questions, comments from users and analyses are as below:

9) The system has all the function I expect it to have for a reviewing process.

The responses for this statement were generally good.

10) The system responded to inputs in short amount of time.

The comments for this statement are varied. Since GraPE is a web-based system, the time taken is not only based on the system, but based on the user’s connection to the Internet and the user’s machine as well. That may explain the varied responses from users. However, the third comment raised another thing of concern, that is, GraPE will crash when the ‘back’ button is pressed. This is actually a known limitation of the system. GraPE does not unassert information when it backtracks. Hence, backtracking and
overwriting previous sections can cause problems. Allowing
users to answer sections according to their preference
instead of fixed order by the system maybe can solve the
problem. The expert has proposed a project proposal to
enhanced ERA in this aspect, which is by using ATMS as
the mechanism for inferences. It was not implemented in
GraPE, as it will require the system to track the rules used
and mark the sections done, and a lack of time prevents the
implementation of this, but it would represent a good feature
to enhance the system for future use.

11) The system is consistent in terminology used, its
interface, and the process.

All users agreed with this statement.

**Question 12 - 13**

The two last questions are about the user’s appreciation
of GraPE, and further comments from users. The questions,
comments from users and analyses are as below:

12) I will use the system again when refereeing another
grant proposal.

While most of users agree with the statement, there is a
comment indicating that it would depend on the proposal.
This shows that the user thinks GraPE is not widely
applicable or useful across the full range of proposals. While
this may be true, providing a flexible structure for reviewing
process to suit for every kind proposal is quite a hard task.
All sections in GraPE are created based on the most
common aspects to be considered in a grant proposal as
identified in the Knowledge Acquisition phase. However,
the system can be enhanced in the future to ensure that it can
be used across a wider range of proposals.

13) I would recommend the system to others.

All users agreed with the statement, and a tester had
commented that GraPE is a good foundation for helping
beginner referees to evaluate grant proposals.

**CONCLUSION**

The main objective of this project is to build an expert
system that can provide help and guidance for inexperienced
referees in reviewing grant proposals. Expert system
technology was chosen for this project because it provides a
well-understood approach for delivering expert knowledge.
GraPE adapted some functions and features of ERA, and
added some new functions to tailor the process of reviewing
grant proposals. The objective of this paper is to describe the
system and the evaluation process of GraPE. The evaluation
process consists of two parts, that is, evaluating the results
of reviewing processes and evaluating feedback from users.
All the testers are postgraduate students, and none of them
had prior experience in refereeing a grant proposal, nor in
writing a proposal. The testers had evaluated two proposals,
one using the EPSRC form, and the other using GraPE. The
proposals used in this evaluation are taken from the
Informatics Research Methodology (IRM) 2004
coursework. The evaluation results suggest that GraPE can
provide some useful help for inexperienced referees.
However, the evaluation results are not enough to provide
conclusive evidence in support of the hypothesis of the
project. This is because of several limitations of the
evaluation phase, including limited time, and the small
number of testers involved. Among problems that were
identified based on the evaluation results are the structured
approach of GraPE that makes the system inflexible and
unable to be used for a wide range of proposals. This
inflexibility includes the reviewing style in GraPE, where
users have to answer each section in a fixed order. In
addition, some of the questions do not cover the complete
range of possibilities. It is hoped that these features can be
enhanced in future development of the system.

**REFERENCES**

year Project Report: Division of Informatics,

Proposal, URL: http://homepages.inf.ed.ac.uk/bundy/
how-tos/rsg-how-to-get-funding.html.

http://homepages.inf.ed.ac.uk/bundy/how-tos/referee-

(Structures and Strategies for Complex Problem

Modern Approach (International Edition), Prentice


Programming 4th Edition Book, Course Technology
Inc. 2005.