ANALYSIS AND EVALUATION OF COGNITIVE BEHAVIOR IN SOFTWARE INTERFACES USING AN EXPERT SYSTEM

Saad Masood Butt & Wan Fatimah Wan Ahmad
Computer and Information Sciences Department, Universiti Teknologi PETRONAS,
Tronoh, Perak, Malaysia

ABSTRACT

In most of the situations, usability evaluations of software interfaces are done by usability experts. Using such professionals needs a certain dimension in business. So, in a lot of small and medium scaled company's, software developers are compelled to learn to manage usability factors. This is not much simpler than training usability engineers on how to create a software application. As a remedy, an expert system CASI for software developers has been designed. In this paper, the expert system of Cognitive Analysis of Software Interfaces (CASI) is outlined to integrate cognitive modelling concepts and is considered as a crucial process for the development of interactive software interfaces. The recommended expert system is entirely dependent on the complete analysis of the user actions and specifications that display the psychological strategy of particular users. Moreover, this system helps designers and software developers to evaluate software prototypes in an intelligent way based on user perception and evaluation views. The paper presents a case study on the development of a rehabilitation database for a person with physical limitations. The results mentioned in this paper show that with the help of the expert system CASI more usability problems in the software interfaces can be detected. Hence, enhancing the usability of software interfaces by an automated CASI system is feasible.

KEYWORDS: *Software Engineering (SE), Human Computer Interaction (HCI), Cognitive Science, Software Interface, Artificial Intelligence (AI), Expert System, Usability Evaluation, Usability Engineering (UE), User Interface, Cognitive Analysis of Software Interface (CASI).*

I. INTRODUCTION

In the designing of the software interface, experts of the SE and HCI need to understand the user's behavior, user's familiarity with different features of a software interface and user's expertise while working with other software interfaces. The HCI deals with social, cognitive and interaction phenomena. Where the social layer is focused on how people interact with each other as well as with technology based on the surroundings.

A Software Interface is an effective source to transfer information and provide communication between a user and a computer. Designing a software interface that is easy to use, easy to learn, and easy to memorize are the attributes of the software usability evaluation [1]. The software usability evaluation is an important concept in the discipline of the HCI.

In the HCI, Usability Engineering plays an important role to achieve users' goals in an effective, efficient and satisfied way. It's a discipline that helps to achieve usability during the design of software interfaces. Usability engineering itself is a vast topic but usability evaluation is the part that contains the various techniques like the heuristic evaluation, guideline reviews and cognitive walk-through [2].

In this paper, an expert system CASI has been developed in order to produce a highly interactive software interface to achieve the user's goals. The paper is divided into five sections. Section 1 is the

Introduction; section 2 is the literature review; section 3 focuses on the expert system CASI; section 4 discusses the case study of the expert system CASI. In the end, section 5 shows the results.

II. LITERATURE REVIEW

The paper [3] describes a design process that helps to link both the SE and HCI processes. The scenarios presented in this paper serve as a link between the two disciplines. In the end, a tool was discussed, Scenic Vista, that works as a prototype to link the design artifacts of the SE and HCI.

The methodology mentioned in [4] discusses the integration of the modern system's development life cycle (SDLC) with the human-computer interaction (HCI) in an information system (IS). As in the traditional development lifecycles of the IS, the role of the HCI is too low only at the design phase or at a later stage that affects the overall development. Thus, there is a gap found between the HCI and SE, and in order to remove this gap a human-centered IS development approach is introduced.

According to [5], a software development team needs to focus on the functionality of the system as well as increase the Usability of the software during the SDLC. One of the methods used in Usability Testing is the Heuristic Evaluation (HE). The HE is a good method to find major and minor problems in the software interface. The HE's main goal is to find Usability problems in the software interface so that they can be attended to as part of the software design process.

As mentioned in [6], Nielsen developed 10 heuristics but later 12 heuristics were developed against the original 10 heuristics. Research shows that the modified heuristics are more efficient and capture more of the defects that were missed by the old heuristics. Despite these benefits, some research shows the pitfalls of the HE. It shows that the HE does not find as many defects as some other Usability Engineering methods. A single evaluator may be able to find a small percentage of defects, so it is useful to involve more than one evaluator and later their results can be aggregated [7].

As mentioned in [8] "Automation is the use of control systems and information technologies to reduce the need for human manpower in the production of goods and services". Today automation is required to perform daily routines and repetitive work. It is also important to automate those software processes that take a considerable amount of time and contain a cycle between various processes. As discussed in [9], the HE evaluators feel that it is difficult to a make report on paper, which is time-consuming and cumbersome. So there is a need to have some type AI based interface evaluator system, which is discussed in section 4.

The HCI strategy concentrates on the human-machine relationships and users. It describes what a program should do from a user's viewpoint. It views users' restrictions like physical, intellectual, successful and behavioural. The HCI growth distinguishes between the users' obligations and the systems' obligations during users' interactions with the systems and how users can socialize with the systems.

Zhang et al. [10] have recommended a strategy that views the HCI concerns and has particular cases of assess items. Table 1 presents the information of the HCI concerns which are composed of four significant places, namely the actual physical, intellectual, affective and behavioral along with their example evaluated items. These HCI concerns highlight on the non-functionality specification research of the software development.

As defined by Lawson [12], the disappointment of the user in software is "the occurrence of an obstacle that prevented the satisfaction of a need". The latest reports on users' disappointment features the problems that took place behind the screen level [11] and the issues of using business sites [14]. These problems took place once the software was developed and sent to the customers [13].

Another research into users' disappointment by Besserie et al. [14] has outlined the disappointment of the computer-based performance knowledge by the users during their everyday performance. The outcome of their research reveals that one-third to one-half of the time is invested before using the system due to the problems in using the application which causes the disappointment. Frustration considerably impacts the level of job fulfillment, office efficiency and public well-being.

III. EXPERT SYSTEM CASI

The expert system evaluates the interface per prototype and works on the concept of inference [15]. In this expert system there are some Facts and Rules which have been defined. The Facts are like

inferences and on the basis of these Facts some Rules have been defined by the users and are stored in the Inference Engine. The Rules are either self-defined or system defined. The self-defined Rules are based on the user's interest whereas the system defined Rules contain a combination of Heuristic and Cognitive walk through. These Rules help to evaluate the user prototypes and architectural prototypes. In this paper, the author has discussed a case study of the development system and has focused on user defined Rules. The expert system CASI contains three phases.

a. Facts and Rules
b. Decision Tree
c. Results

a. Facts and Rules

For this system, five Rules are defined:
Rule A: Go back to the previous Process, i.e., IUP
Symbol: R_A
Rule 1: Easy to use
This means that the prototype makes the task easy to use.
Symbol: R_1
Rule 2: Easy to learn
The task is easy to learn and the next time the user performs the same task easily without thinking much.
Symbol: R_2
Rule 3: User perception
The interface was designed according to the user's perception.
Symbol: R_3
Rule 4: Easy Mastery
The interface provides enough information so that the user doesn't need to study the Help file.
Symbol: R_4
Rule 5: Provided Functionality
All the functionalities are available that the user stated during the requirement gathering phase.
Symbol: R_5

b. Decision Tree of CASI

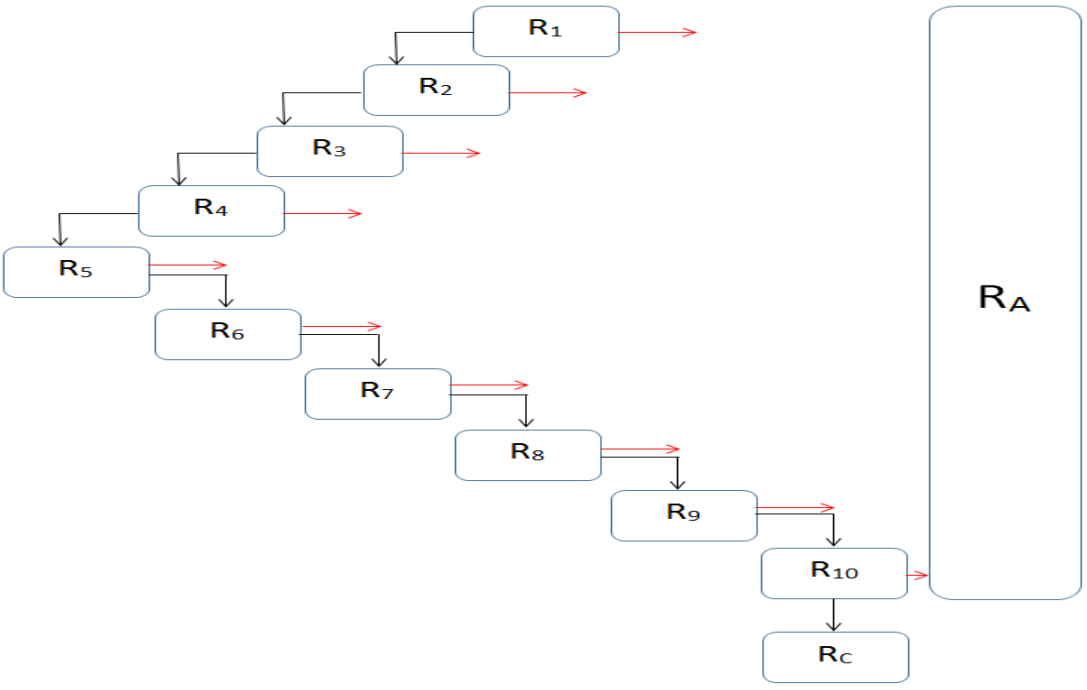

Figure 1: Decision tree of CASI

Rule R1, R2, R3 and R4 are stored in the Inference Engine. The expert system evaluates the output (that comes from the IUP phase) by R1. If R1 proves to be correct, then the prototype will move to R2 for evaluation. If it fails at any Rule, then the flow will move towards RA. RA is a state to improve the prototype according to the self-defined or system defined Rules.

c. CASI Process

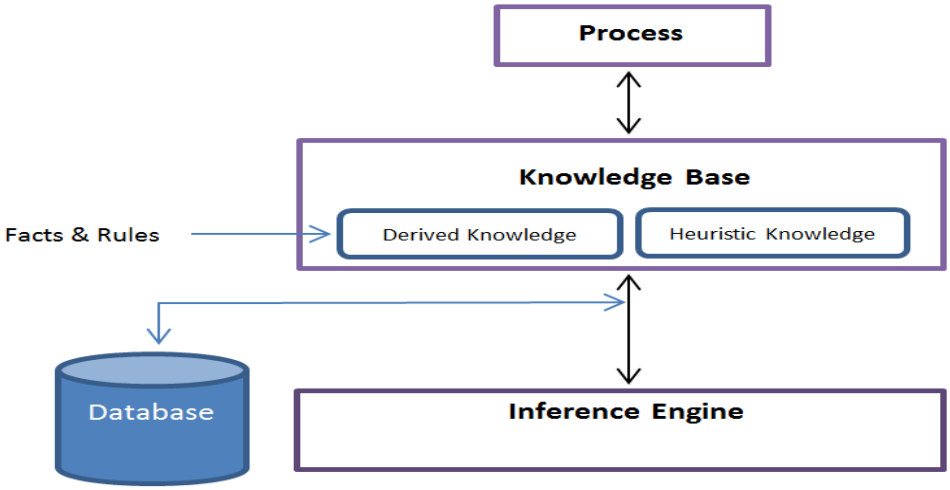

Figure 2: CASI Process

CASI contains four elements named: Process, Knowledge Base, Inference Engine and Database. Figure 2 depicts the clear understanding of the flow of the process between these elements.

IV. EXPERIMENTAL MODEL

In this section, the author discusses the case study which is the development of the university online classroom booking system that was built on the UZAB Model. Each prototype was tested by the expert system CASI. Further improvement was noted where the expert system could not evaluate according to the user's perception.

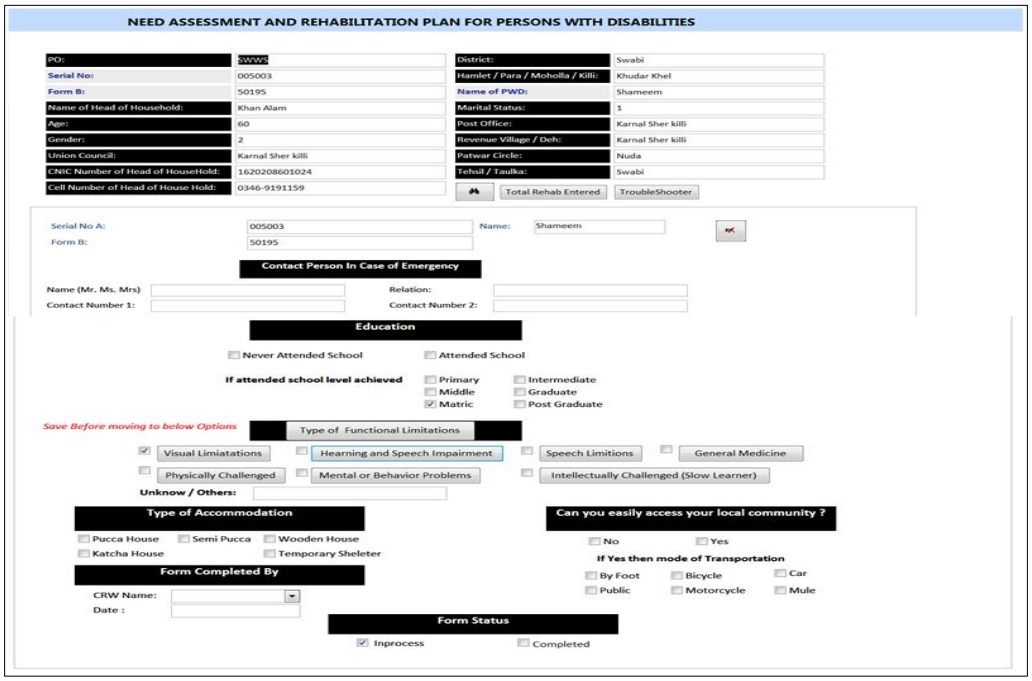

Figure 3: Main Screen

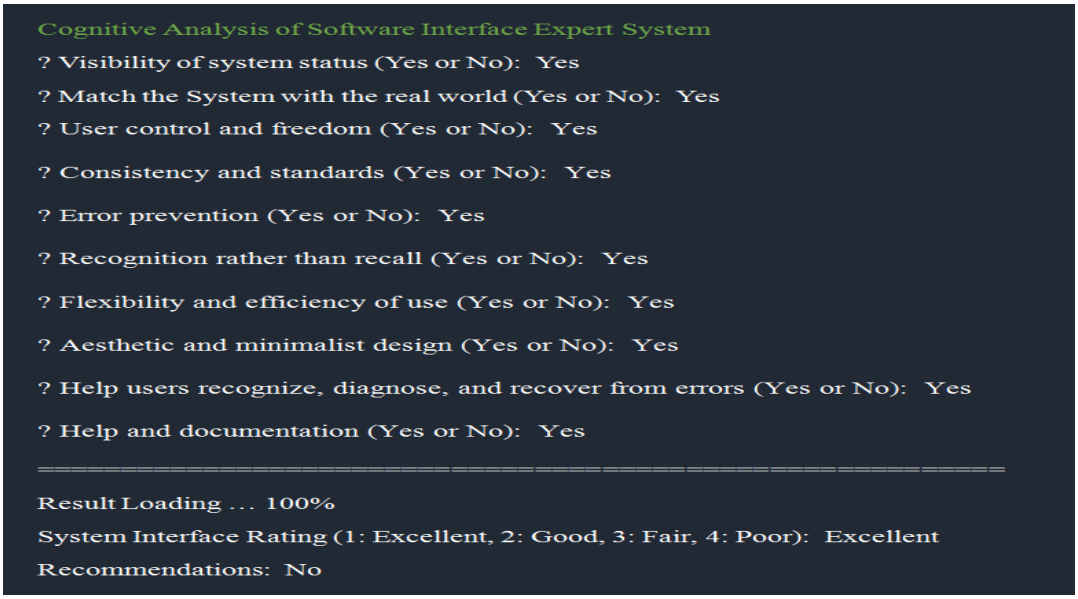

Figure 4: Expert system CASI Evaluates Main Screen

Figure 4 shows the results of the expert system CASI while evaluating the Main Screen. Termination occurs where any Rule fails to achieve the user's goal. Similarly, Figure 6 shows the result of the visual limitation screen.

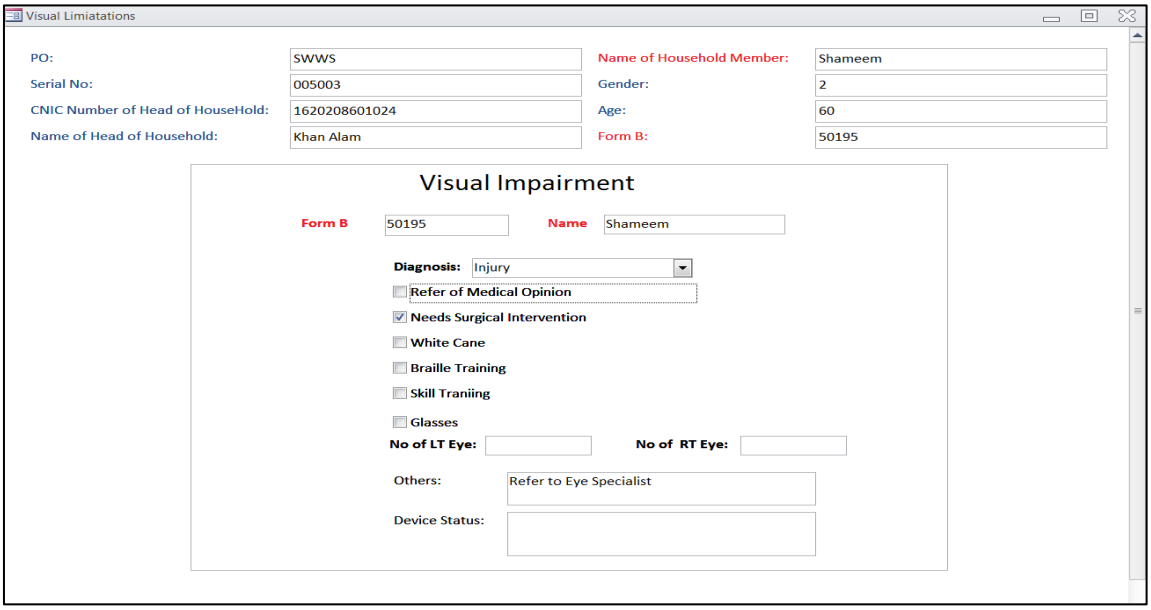

Figure 5: Visual Limitation Screen

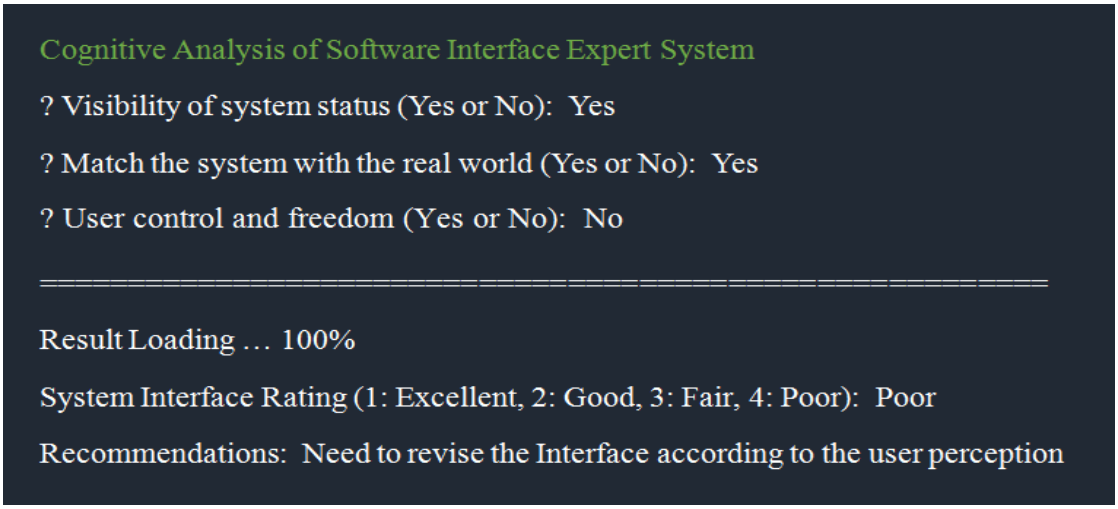

Figure 6: Expert system CASI Evaluates Visual Limitation Screen

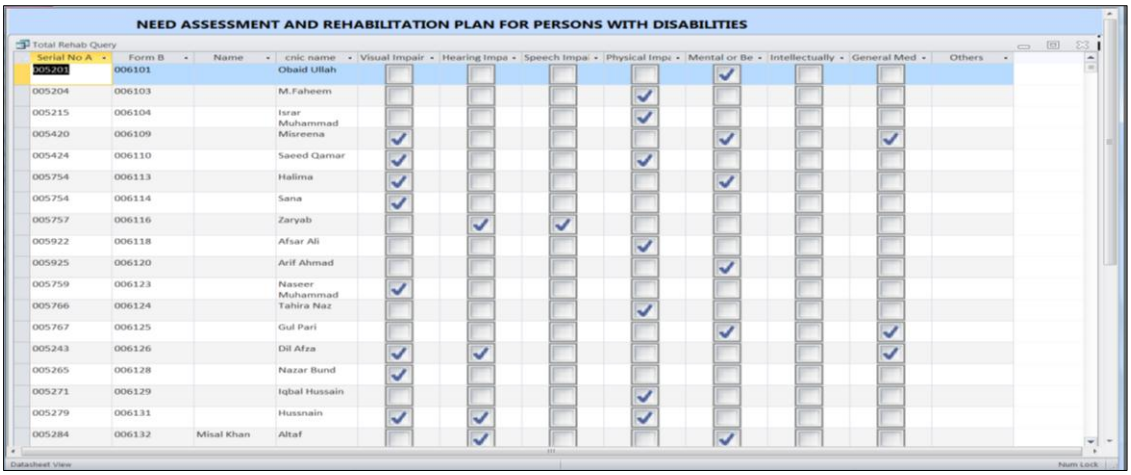

Figure 7: Datasheet View

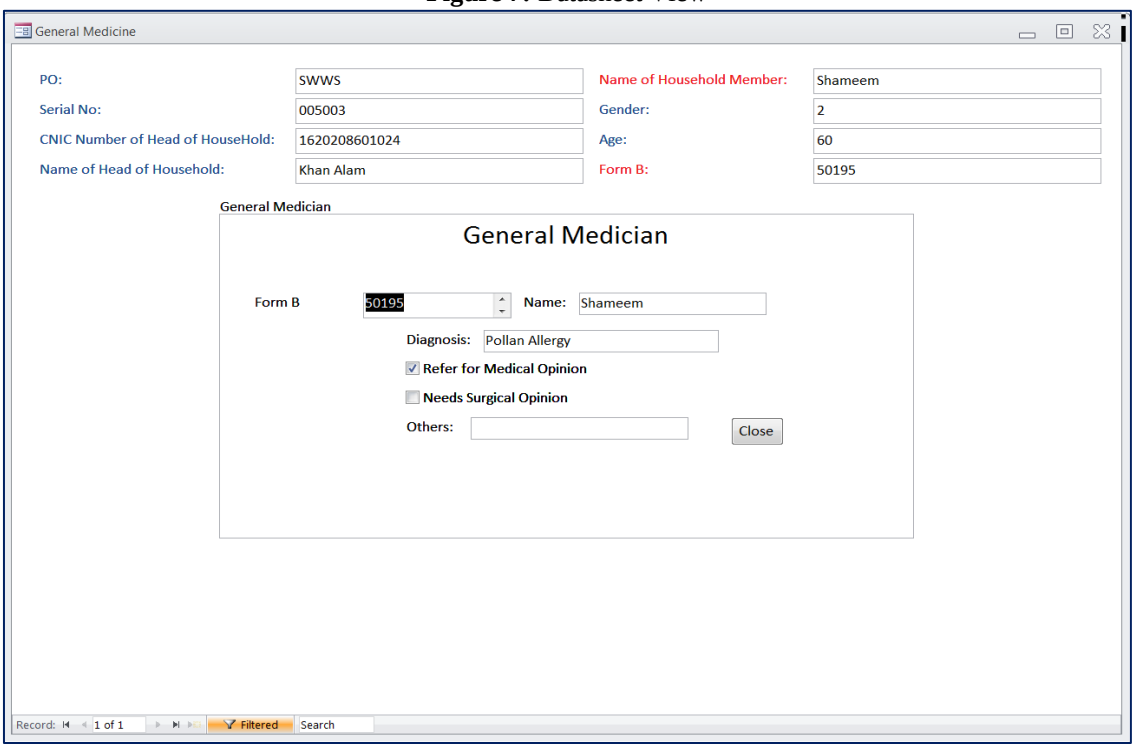

Figure 8: General Mediciation Screen

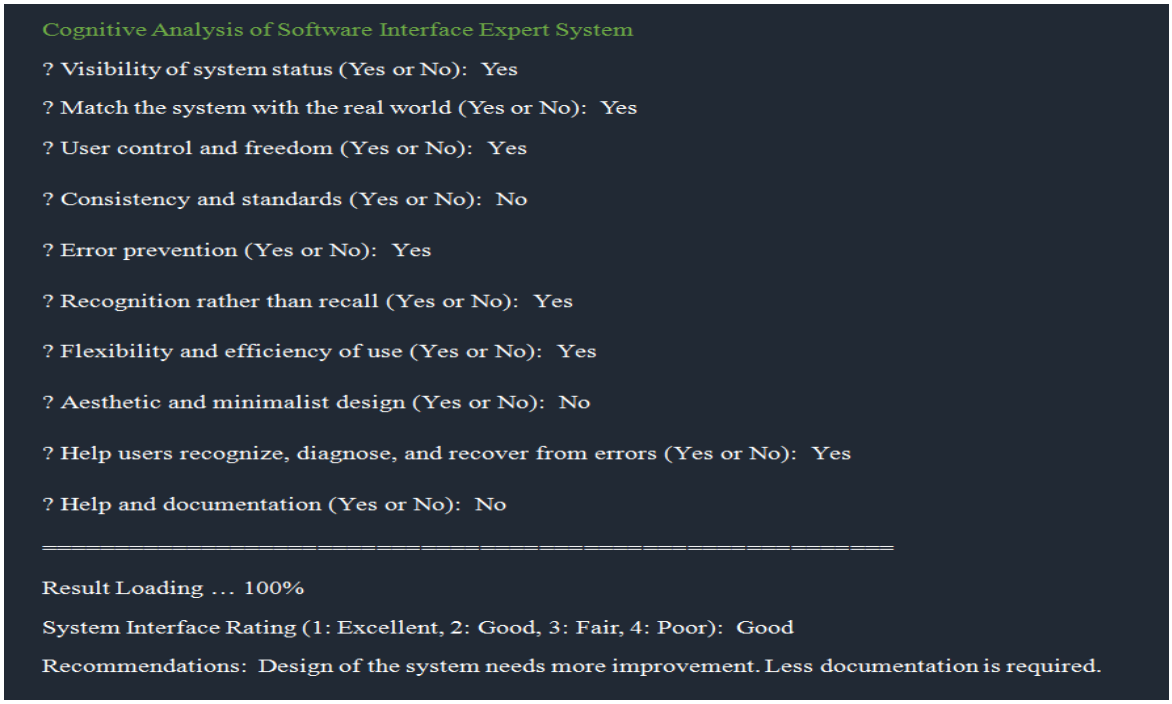

Figure 9: Evaluation of General Mediciation Interface

V. ANALYSIS AND RESULTS

The paper has brought a solution to the field of interface evaluation for the SE and HCI experts. On the one hand, it demonstrates how the expert system CASI is usable in the HCI design. On the other hand, it shows the benefits of using the expert system CASI based on the case study. The feedback obtained from the evaluators was generally positive towards the acceptance of the expert system CASI. All of the evaluators liked the new method of evaluating the software interface but provided recommendations for future improvement in the expert system CASI.

The results obtained from the expert system CASI is based on Quality, Time, and Error detection, and it is found that the expert system CASI helps to improve the quality of the software interface and can detect more errors in software interfaces in less time.

a. Quality Improvement by CASI

The term quality in the field of interface evaluation means having zero defects and achieving the maximum interface usability. CASI proved this quality definition. CASI helps the SE and HCI experts to detect defects in order to achieve interface usability.

b. Time Saving

CASI provides rapid results in less time as compared to the traditional software interface evaluation techniques.

c. Error Detection

CASI is designed on those FACTS and RULES that help the SE and HCI experts to detect errors in software interfaces and help them to fix the error as soon as they are detected.

VI. CONCLUSIONS

With the rapid increase in the field of Cognitive Science and the growth of the interactive technology innovation, the computer is widely used in our daily life. The described expert system CASI is a helpful and effective approach to evaluate the software interfaces during their development phase. The expert system CASI will be challenging in the beginning when they are provided with the FACTS and RULES to evaluate every interface of the software. Though it is a good approach to produce a usable system that can fulfill a user's requirements and work up to the use's perception. Successful testing of an expert system will contribute to the evaluation of software interfaces according to the user's cognitive in a true manner. It is not the least point to evaluate software and increase usability.

Furthermore, new ideas and techniques must be considered to enhance the features of the expert system CASI.

ACKNOWLEDGMENT

The author of the paper would like to thank Universiti Teknologi PETRONAS, software evaluators and other staff members for their valuable feedback during the intermediate phase of the methodology presented in this paper.

REFERENCES

[1] Yonglei Tao, "Work in progress - introducing usability concepts in early phases of software development", 35th ASEE/IEEE Frontiers in Education Conference, Publication Year: 2009 , Page(s): 702 – 706.
[2] Ritter,F.,E., Baxter, G., D., Jones, G., and Young, R., M., 2000. "User interface evaluation: How cognitive models can help".
[3] G. Mori, F. Paterm and C. Santoro. Ctte Support for developing and analyzing task models for interactive system design. IEEE Trans. Software Eng., 28(8):797–813, 2002.
[4] A. Dix, J. E. Finlay, G. D. Abowd, and R. Beale. Human-Computer Interaction (3rd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.
[5] A. Monk, P. Wright, J. Haber, and L. Davenport. Improving Your Human-Computer Interface: A Practical Approach. Prentice Hall International, Hemel Hempstead, 1993.
[6] M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. ACM Comput. Surv., 33:470–516, December 2001.
[7] P. G. Polson, C. Lewis, J. Rieman, and C. Wharton. Cognitive walk- throughs: a method for theory-based evaluation of user interfaces. Int. J. Man-Mach. Stud., 36:741–773, May 1992.
[8] J. Nielsen and R. L.Mack. Usability inspection methods. Wiley, 1 edition, April 1994.
[9] http://en.wikipedia.org/wiki/Automation last accessed 2-6-2012.
[10] Law, E.L.-C., Hvannberg, E.T., 2004a. Analysis of strategies for improving and estimating the effectiveness of heuristic evaluation. In:NordiCHI 2004, Tampere, Finland, pp. 241–250.
[11] Ashok Sivaji, Azween Abdullah, Alan G. Downe, "Usability Testing Methodology: Effectiveness of Heuristic Evaluation in E-Government Website Development", ISBN 978-0-7695-4412-4, Proceedings of 2011 Fifth Asia Modelling, AMS 2011 Conference,pp.68-72.
[12] http://www.usabilitybok.org/methods/p275?section=basic-description last accessed 7-5-2011.
[13] R. Molich, A. D. Thomsen, B. Karyukina, L. Schmidt, M. Ede, W. van Oel, and M. Arcuri. Comparative evaluation of usability tests. In CHI '99 extended abstracts on Human factors in computing systems, CHI '99, pages 83–84, New York, NY, USA, 1999. ACM.
[14] Lawson,R. "Frustration: The development of a scientific concept". New York: MacMillan 1965.
[15] Patrick, J. R. "Future of the Internet. Keynote Speech". Americas Conference on Information Systems, 2003.
[16] Zhang, P., Carey, J., Te'eni, D., and Tremaine, M. "Integrating Human-Computer Interaction Development into the Systems Development Life Cycle: A Methodology". Communications of the Association for Information Systems, vol. 15, pp. 512-543, 2005.
[17] Bryant, M. "Introduction to user involvement", The Sainsbury Center for Mental Health, 2001.
[18] Rosen,J. "A Methodology of Evaluating Predictive Metrics", Software Metrics Symposium, IEEE Computer Society, 1998.
[19] Integrating Human-Computer Interaction Development into SDLC: A Methodology, Proceedings of the Americas Conference on Information Systems,New York, August 2004.
[20] http://www.useit.com/papers/heuristic/heuristic_list.html

Authors

Saad Masood Butt received his BS (Software Engineering) degree from Bahria University Islamabad, Pakistan in 2008. He completed his MS (Software Engineering) degree in 2010 from Bahria University Islamabad, Islamabad Pakistan. He is the recognized Engineer of Pakistan approved by Higher Education Commission and Pakistan Engineering Council (PEC). He has got more than 4 years' experience and was associated with various organizations in Pakistan. Currently, he is pursuing his PhD degree in the department of Computer and Information Sciences at Universiti Teknologi PETRONAS, Malaysia.

Wan Fatimah obtained her Ph.D from Universiti Kebangsaan Malaysia. She is currently an Associate Professor at Universiti Teknologi PETRONAS, Malaysia. Her research interests include topics on Multimedia, Human computer interaction, mathematics education, e-learning