Project Proposal

Stephen Carse

March 1, 2013

1 Principle Investigator

Stephen Carse

g10c0493@campus.ru.ac.za

Honours Student, Department of Computer Science, Rhodes University

Supervisor: Dr. Karen Bradshaw

2 Project Title

Proposed title: Veterinary Diagnosis Expert System for Remote Use

3 Statement of the Problem

In many remote areas, access to a veterinary professional is often restricted. Farmers and game reserve staff are often confronted with an animal displaying signs of illness, injury or distress and are forced to wait for the arrival of a vet. In some cases the person responsible may be unsure whether it is necessary to call for a vet, which can be a costly exercise. The delay in the arrival of medical attention can often be very distressing for both the animal and people involved. An expert system accessible in remote areas capable of diagnosing ailments affecting animals or advising on the necessity of calling a vet could greatly assist farmers and game reserve staff. The system could potentially provide a first line of diagnosis before a veterinary expert arrives and could also assist in verifying the diagnosis of a vet and provide assistance to the vet in diagnosing. Owing to the prospective users of the system not having easy access to a desktop PC during the normal course of their working day or while attending to a sick animal, the system should be accessible through a simple interface on a mobile phone. The field of the expert system will by necessity be limited to a subset of animals and/or conditions.

4 Objective of the Research

The objectives of this research are:

- Primary goals:
  - Develop an expert system for a specific veterinary domain
  - Provide an interface to interact with the expert system using a mobile phone
- Secondary goals:
  - Provide a simple interface for an expert to add new information and knowledge
  - Automatically translate new information and knowledge to rules and add these to the knowledge base
  - Allow users and experts to provide feedback on previous diagnosis of the system
  - Take feedback into account in future diagnoses

5 Background and History

Artificial intelligence (AI) is the discipline of enabling a computer to think intelligently [6]. Humans have long been fascinated with the idea of being able to grant intelligence to machines and it has been the focus of much study. One of the key questions faced was how to determine if a machine was indeed intelligent. Answering the question was made difficult by humans' limited understanding of the brain's thought processes and the difficulty in defining intelligence itself. Alan Turing proposed the following definition in the 1940s [4]: Questions should be asked of both a human and a computer. The questioner should be unable to distinguish between the human and computer based on the responses received in order for the computer to be deemed intelligent.

Expert systems have been developed as a branch of AI and contain specialised knowledge for a particular field that only an expert in that field would possess. The system can then be used to make informed decisions and conclusions on a particular situation in that field based on this knowledge as a real expert would [5]. Owing to the vast number of rules required for even a single domain, no expert system exists which can model the intelligence of an expert in all fields. These systems are rather focused on a particular field and are used to replace or assist an expert in making decisions or in verifying an expert's conclusions [5]. Examples of fields in which expert systems are applied include law, tax, accounting, automated call centres, medical and education [2, 7]. The United Kingdom's National Health Service has an online tool that allows people to enter their symptoms or ask various health related questions in order to receive advice and information and be directed to a doctor or other health practitioner, where appropriate [1].

An expert system consists of four main components [9]. The *knowledge base* stores the knowledge possessed by the system, which is used to mimic the decision-making ability of the expert, and includes both firm facts and heuristics. The *working memory* contains information provided by the user during this particular session and the conclusions reached based on this information. The *inference engine* uses knowledge stored in the knowledge base and the information currently in the working memory to make new inferences and decisions. It also produces explanations of these solutions aimed towards the user of the system [3]. Lastly, the *user interface* allows the user to interact with the system, both to allow the user to provide information about the problem and for the system to provide feedback to the user.

The explanation capability of an expert system allows it to explain the rationale followed in arriving at its conclusions [7]. As a human expert, such as a doctor, should be able to justify their conclusions, so should the system be able to explain how it reached its conclusions. This allows the verification of these conclusions by an expert and can also assist an expert in merely confirming a diagnosis without being required to redo the entire diagnosis process. An additional benefit is to allow the system to be debugged by inspecting its reasoning process [5].

6 Approach

Development of an expert system can broadly occur through two different means: programming from scratch or making use of shells that assist in the development process [8]. When developed from scratch, specialised programming languages developed specifically for the expression of logical statements are typically used. Expert system shells are expert systems that have had their knowledge removed [3]. They consist of an inference engine and an interface to assist in the construction of the knowledge base. The knowledge base, inference engine and user interface are provided, to varying degrees, when a shell is used [7].

The use of a shell provides the advantages of speeding up development of the expert system and the simpler maintenance of the knowledge base [8] It has drawbacks, however, in the inference engine, as its reasoning cannot be customised, user interface, as some can be cumbersome and lack user-friendliness, and explanation facilities, as they cannot be customised to the requirements of the specific system. Programming the system manually does result in a longer and more complex development period but means that all aspects of the system can be tailored to this particular system's requirements and can potentially result in a more efficient system [8].

Owing to the variety of tools available and the potential impact of the choice of these tools to use on the final system, the decision of which to use needs to be well researched and carefully considered. I will initially evaluate the different shells, commercial and open source, and the different programming languages typically used for expert system development to determine the best tool or combination of tools to use for the construction of the expert system. Examples of shells that are available are CLIPS, Jess, Drools and VP Expert. Programming languages include LISP and PROLOG.

Once I have determined the tools that I will use for the implementation of the expert system I will develop a small, prototype application to ensure that the tools will function effectively together and verify that I have made a sound decision in their selection.

Following the construction of the proof of concept of the expert system, I will move on to the extraction of the information required for the expert system from the expert(s). This information will be converted into rules and placed into the knowledge base. Development of the inference engine will occur after and in tandem with this process, depending on the specific tasks required to be performed.

The initial implementation of the expert system will not rely on a fluid mobile interface but will be a stripped-down desktop version, which can easily be used during development to verify the functioning of and test the expert system. Only once development of the knowledge base, working area and inference system is sufficiently underway will significant attention be paid to the mobile user interface aspect of the system. Due to the user only ever observing the interface, this should be as clear and user-friendly as possible and work in tandem with an efficient expert system backend to deliver clear, correct information to the user.

The expert system should function at the capability of the expert and the decisions it produces should be considered reliable; testing of the system is therefore an essential part of its development and will be undertaken throughout the development process. This will occur in consultation with the expert(s) as their schedules permit. Feedback on the performance of the system will be taken into account and any modifications made as required.

7 Requirements / Resources

- Technical requirements:
  - A web server for the hosting of the expert system
  - Expert system development tools, to be determined
  - Appropriate tools for the development of the user interface, to be determined. Should an application be developed, the applicable platform SDK and a device running that platform will be required.

- Personnel requirements:
  - An expert in the veterinary field to provide the expert knowledge

8 Progression Timeline

Deadline	Activity
01 March	Submit project proposal
08 March	Set up research website
12 March	Seminar 1
31 March	Explore development tools and techniques
30 April	Decide on development tools
27 May	Submit literature survey
30 May	Develop prototype application, begin proper application
30 July / 06 August	Seminar 2
15 August	Translation of knowledge to rules, construction of knowledge base
16 September	Short paper submitted
20 September	Complete core of expert system
30 September	Complete mobile interface
27 October	Complete testing and further modifications
28 / 29 October	Seminar 3
1 November	Project Handin
4 November	Website complete
19 / 20 November	Oral research exam

References

[1] NHS Direct Symptom Checker. http://www.nhsdirect.nhs.uk/CheckSymptoms.aspx.

[2] Ioannis M. Dokas & Alexandre Alapetite. A development process meta-model for Web based expert systems: the Web engineering point of view. http://www.risoe.dk/rispubl/SYS/syspdf/ris-r-1570.pdf, October 2006. Denmark.

[3] Marko Bohanec and Vladislav Rajkovic. DEX: An expert system shell for decision support. *Sistemica*, pages 145–157, 1990.

[4] Andries P. Engelbrecht. *Computational Intelliegence: An Introduction*. Wiley, 2002.

[5] Joseph Giarratano and Gary Riley. *Expert Systems: Principles and Programming*. Thomson, 4 edition, 2005.

[6] R.I. Levine, D.E. Drang, and B. Edelson. *A comprehensive guide to AI and expert systems*. Byte books. McGraw-Hill, 1986.

[7] M. Negnevitsky. *Artificial Intelligence: A Guide To Intelligent Systems.* Addison-Wesley, 2005.

[8] Motaz Saad. CLIPS: Expert System Shell. Lecture Slides from the Department of Computer Science, Islamic University of Gaza, September 2008.

[9] Dieter Vogts. Investigation of a Web-based expert system shell. *South African Journal of Information Management*, 3, 2001.