Detecting Diseases in Farm Animals With Embedded System

JINGYU ZHANG, RAFAEL CALVO, RICHARD SHEPHARD, CRAIG JIN
School of Electrical and Information Engineering, School of Electrical and Information Engineering,
Australian Biosecurity CRC, School of Electrical and Information Engineering
The University of Sydney NSW, 2006
AUSTRALIA

Abstract: - Veterinary epidemiology is a rapidly developing science. A key factor for success is the ease of data capture. However, real-time data collection is quite hard because in many such systems, users must return to the home or office to import the data to computer or fill a paper-based data entry form. This paper presents an embedded application system for animal disease report and Surveillance system in Veterinary Science which provides real-time Surveillance and allows instant analysis. Using the system, users can access user-friendly software to help diagnose diseases or problems in herds. It designed to deliver benefits to producers, veterinarians, animal health workers, stock agents, government departments and the beef industry.

Key-Words: - Computer Applications in Veterinary Science. Software Design

1. Introduction

Remote extensive cattle grazing regions of Australia are characterized by large herds, long distances between properties and communities, a single notifiable disease reporting system is not very effective [1, 2] . Therfore, We investigate the potential for syndromic disease information to be captured by observers creating the disease diagnostic embedded application software to analyzed, and report disease occurrence. Changes in relative frequencies, notifying local government veterinary authorities that a given disease syndrome has emerged or changed in frequency, potentially leading to targeted surveillance efforts being focused towards investigation of the syndrome.

Under reporting of disease events in farm animals has been identified in numerous studies and is a significant gap in Australia. National surveillance processes in that it becomes difficult to generate information to support claims of freedom from disease and reduces our capacity for early detection of emerging disease problems. The main sources of animal health surveillance information are veterinary laboratories, but these sources have been declining and represent only a small proportion of animal disease events and provide virtually no information on the health status of livestock in the remote pastoral regions of northern Australia which are the main supply areas of our beef exports. In one such Australian project, the Bovine Syndrome Surveillance System (BOSSS) is being developed[3, 4]. In this project, users (stakeholders in the farming industry) collect information about individual sick animals. This information is then used to diagnose according to a large database of symptoms and diseases and to possibly detect a new disease. Additionally, As often happens in many other areas where data collection is important, a key aspect for success is the way of data capture[5].

Electronic data capture can be achieved either using a web-based data submission system (providing real-time access to a centralized database and allowing instant analysis), or by the use of hand held computing devices. Web-based systems mean that useful data is not captured in a real-time, interactive manner; on the other hand, handset devices do not have the capability to deal with enormous amounts of data [6, 7]. Therefore a mass data acquisition system is required which provides the collector with guidance, which is adaptable to the given circumstances.

In the animal health care scenario, a mobile application is an obvious approach for data collection, but, as we will discuss later, this use-case scenario brings many additional challenges due to the environmental conditions in which the data are collected. These challenges motivate the need for designing an application framework [5-8] that improves software reusability, both at the code and the design level.

2. System Overview

The mobile data collection environment for the system that we are developing must have the following qualities to promote adequate data capture; firstly, be able to travel with observers (and therefore be present with the observer when handling stock; secondly, be able to work well in a difficult environment, whether it has network coverage or not, with bad lighting conditions, thirdly, be easy to use and learn, fourthly, be capable of providing feedback to the user at the time of data entry. In general, users will have the data collection device with them as they observe and interact with the diseased animal.

2.1 Requirements Model

We analyze those requirements in more detail. In Figure 1, we describe an overall description of what the software should do. Cattle subcomponent, Disease subcomponent and Sync Information subcomponent are contained in the surveillance system. First, cattle component covers the herd analysis and record keeping needs of cattle producers. Second, disease component contains a vast amount of information on animal disease which can be searched in different ways and used either for increasing your knowledge or clarifying your understanding about diseases. Each disease explanation is written in straightforward language and may have photos to view as well. Third, synchronization component is the process of making sure that handset devices and server contain the same up-to-date information.

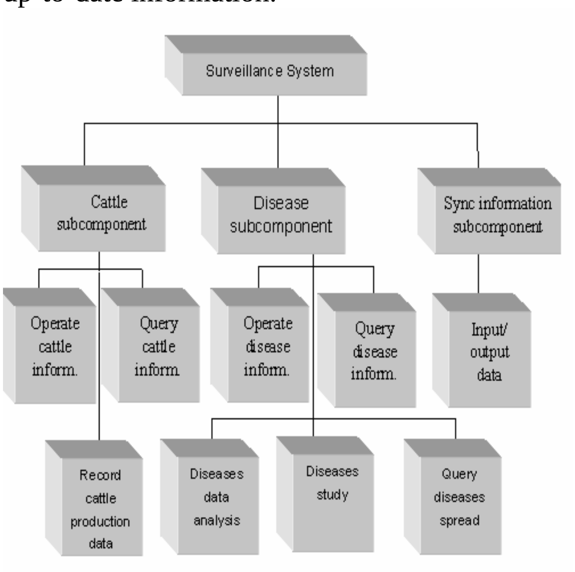

Figure- 1 The Components of the system

The Sequence diagram that corresponds to this model is shown in Figure 2. The sequence diagram is used to show the interactions between cattle producers and this embedded system in the sequential order that those interactions occur[9]. First of all, there are two scenarios in this diagram. The common scenario is user find out more about various symptoms which have seen in the cattle. Then enter information about symptoms using the surveillance interface. In another words, cattle producers use drop-down menus and graphical aids, the system provides people with a list of possible disease causes to consider and then the program ask you further questions about the affected animal/s as it attempts to identify the disease more accurately. Next, the information pass business controller to Data Access Interface in which both implementation and interface inheritance are supported. Then it switches information into two options. Firstly, DAI search the Indices aim to find out the objective data's location. The system will fetch data from DC if the data save in DC. After this, the response is able to recommend further investigations of the case by describing additional, specific observations, examinations, and sample collections or post mortem examinations that may be useful to more clearly identify the disease in the particular case. On the other hand, the program will obtain data from MCD if the indices show that the destination information is saved in MCD. Lastly, in the case of data can not be found in the handset device, our system provides the function of lookup diagnostic information from network through web services between embedded system and remote servers

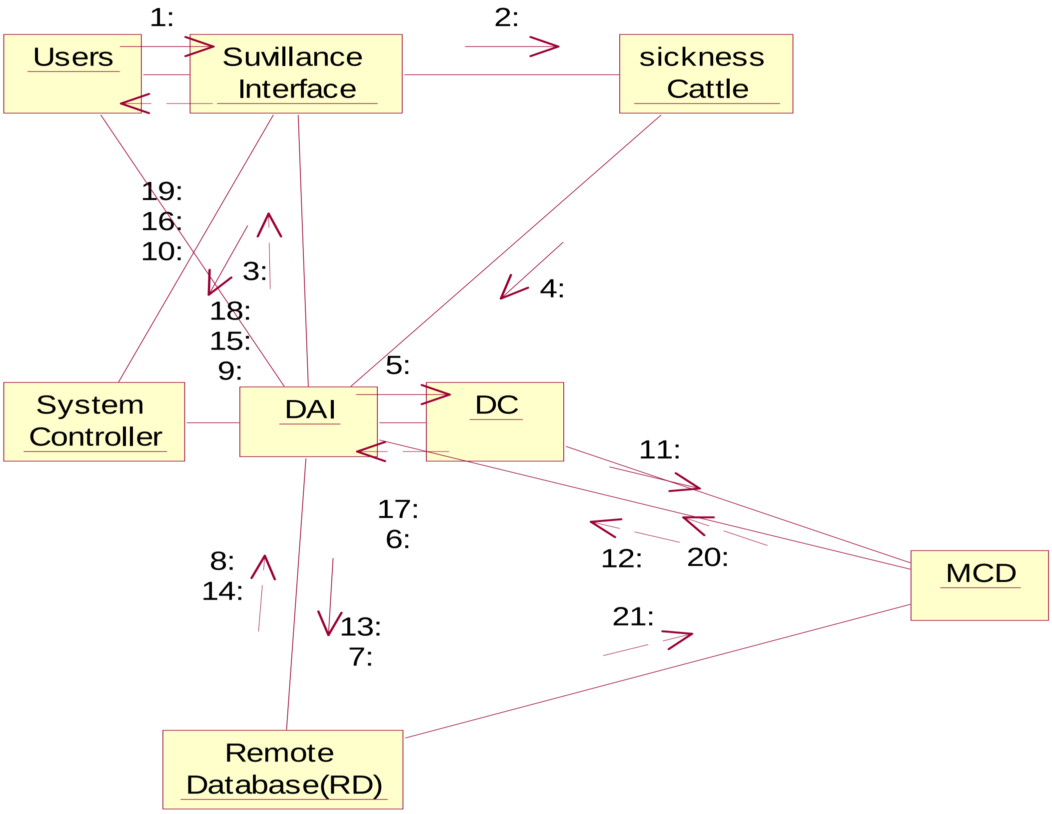

Figure- 2 Collaboration

because the server contains a vast amount of information on animal disease which can be searched in different ways and used either for increasing your knowledge or clarifying your understanding about diseases. Each disease explanation is written in straightforward language and may have photos to view as well. Finally, the diagnostic results are shown on the screen of handhold device.

Architecture

The architecture of current system for managing Veterinary and animal data has led us to develop a multilevel architecture for design data management (see Figure 3). Five components consist of the application. The system's core is the database component. Its responsibility is to provide an efficient interface to stored data, and to isolate the higher layers from considerations of reliability, physical storage structure, data sharing, access control, and integrity control. It supports unstructured data and record clustering, while providing a simple and flexible interface upon which the rest of the system is built.

A Data Access Interface component is built on top of the database component and wrapper layer that makes data in relational database appear as objects to the business logic[10, 11]. It is responsible for managing the connection of databases, and organizing the alternative databases and multiple versions within the database supported by the database component. It provides an interface for extracting and replacing subcomponents of a design from the design hierarchy. The final layer consists of programs that interpret the data stored about a design[12]. Design interpreters provide an interface to design tools that is appropriate for manipulating the data at hand, e.g., layout geometries, transistors, etc. In the remainder of this section, we describe each of the levels in more detail.

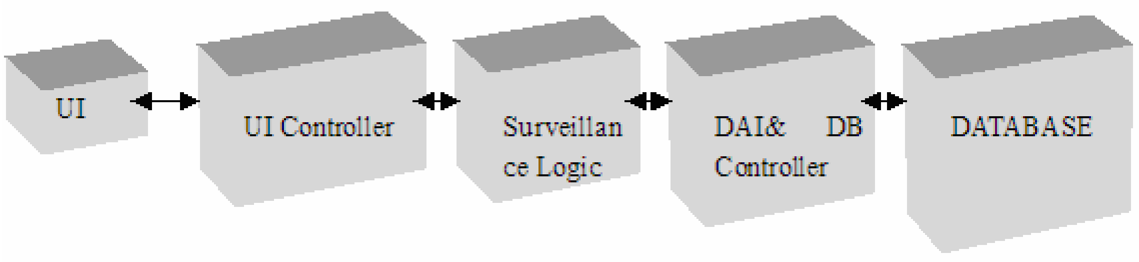

Figure 3. The components of the system

For a mobile application framework, the architecture and partitioning of the system operation are important and essential to good system design and directly influences the software's availability, usability, reusability, performance and maintainability. Mobile application systems are quite different to distribution systems and web base systems because they have to take into account the limited operating system, limited screen space and cost limitations. Nonetheless, mobile applications are 'smart client' applications that have increasing complex roles given improvements in

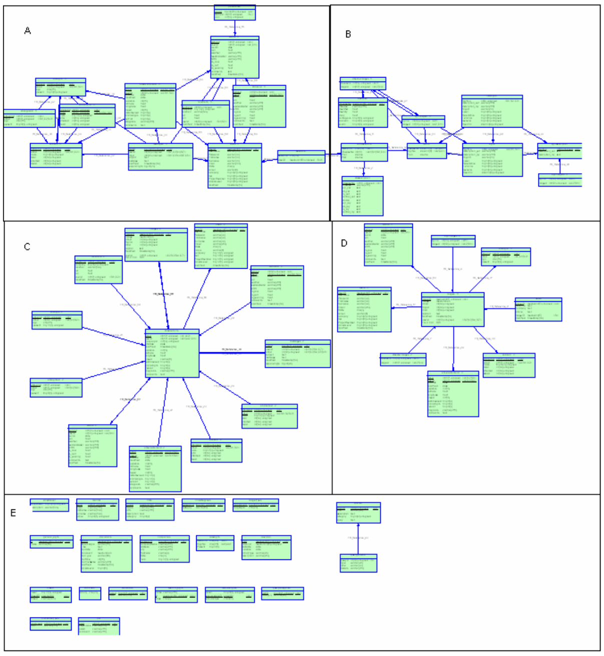

Figure 4. Tables

hardware. More server-side and pc-side applications are moving to the mobile platform presenting a challenging research problem for mobile application framework development.

To implement the database, we use Data Centre (DC), and Memory Card Database (MCD) architecture. Implementing HB-Nary tree in the MCD tier, The HB-NAry tree is a multi-attribute search structure with behavior similar with the single-attribute N-Ary tree with holey brick function. Using holey brick (HB) deal with smaller bricks removed in k-dimensional space. To the objective of saving space, the empty nodes do not be contented in HB-Nary tree.

The entire information is stored in tables of rows and columns, which show in figure 4. Tables are converted to third normal form (3NF) [UML]. A block is signs information; B describes diseases information; C illustrates cattle information; D shows the tables which relation to Sync and E configuration & Indices. The relationship between block A and b is table matrix in which the primary key "sign ID" and "Disease ID" are contained.

The database of the animal Surveillance system is maintained and managed easily. Performance, especially complex queries demands, can slow to a halt and when your data is saved in various formats on different storage materials. DC and MCD database structure is a lot more powerful in all these areas, especially in terms of analysis. This database structure let we manipulate data in complex, interesting ways, allowing to retrieve all records that match the specific criteria, cross reference different tables.

3. Conclusion

This embedded syndrome surveillance system is a promising application for Veterinary science because the handset animal health information system provides data entry, data analysis and reporting for a syndrome surveillance system for use by producers in remote areas. Moreover, this system structure is a novel solution for component reuse and architectural design. It can support user and system initiative and separate the user interface from the rest of the application caused by using many design patterns in this architecture.

4. ACKNOWLEDGEMENTS

Australian Biosecurity CRC funds this project which roject number is 3.015Re.

References:

[1] P. V. Garnerin, A., "The French computer network for the surveillance of communicable diseases," in *Engineering in Medicine and Biology Society, 1988. Proceedings of the Annual International Conference of the IEEE,* 1988.

[2] J. O. W. Kephart, S.R.; Chess, D.M., "Computers and epidemiology," *Spectrum, IEEE,* vol. 30, pp. 20-26, May 1993.

[3] J. Zhang, Calvo, RA, Shephard, RW & Jin, C, "A Framework for Mobile Disease Report and Investigation. International Conference on Mobile Technology Applications and Systems," in *3rd International Conference on Mobile Technology Applications and Systems* Bangkok, 2006.

[4] A. C. R. Shephard, J. Toribio, P., "Development of the Bovine Syndromic Surveillance System (BOSSS)," in *the XIth International Symposium on Veterinary Epidemiology and Economics*, 2006.

[5] D. N. Taylor, R.N.G.; Amin, S.; James, A.; Boulton, S., "Combining dental epidemiological relational databases," in *24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society*, 2002.

[6] M. Q. L. Nancy L. Sprince, "Respiratory symptoms: Associations with pesticides, silos, and animal confinement in the Iowa Farm Family Health and Hazard Surveillance Project," *American Journal of Industrial Medicine,* vol. 38, pp. 455-462, 6 Sep 2000.

[7] B. V. Vourc'h G, Gibbens J, De Groot BD, McIntyre L, Poland R, et al, "Detecting Emerging Diseases in Farm Animals through Clinical Observations," *Emerg Infect Diseases,* vol. 12, February 23 2006.

[8] G. C. Ozalp Babaoglu, Andreas Deutsch, Gianni A. Di Caro, Frederick Ducatelle, Luca M. Gambardella, Niloy Ganguly, Márk Jelasity, Roberto Montemanni, Alberto Montresor, Tore Urnes "Design patterns from biology for distributed computing," *ACM Transactions on Autonomous and Adaptive Systems (TAAS),* vol. 1, pp. 26-66, September 2006.

[9] W. N. W. Robinson, H.G., "Finding reusable UML sequence diagrams automatically," *Software, IEEE,* vol. 21, pp. 60-67, Sep-Oct 2004.

[10] Y.-B. L. W.-R. L. J.-J. Chen, "Effects of cache mechanism on wireless data access," *Wireless Communications, IEEE Transactions on,* vol. 2, pp. 1247-1258, Nov 2003.

[11] A. K. Mario Antonioletti, Norman W. Paton, Andrew Eisenberg, Simon Laws, Susan Malaika, Jim Melton, Dave Pearson "Industry perspectives: The WS-DAI family of specifications for web service data access and integration " *ACM SIGMOD Record* vol. 35, pp. 48-55, March 2006.

[12] D. B. Matic, D.; Kegalj, H., "Data access architecture in object oriented applications using design patternsccess," in *Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean*, 2004.