

ExSIDE: Component Based Object Oriented Expert System's Integrated Development Environment

Mohamad Hanif Md Saad[#], Rabiah Adawiyah Shahad^{*}, Kong Win^{*}, Aini Hussain^{*}

[#] Department of Mechanical & Material Engineering, Universiti Kebangsaan Malaysia, Bangi, Selangor, 43600, Malaysia

^{*}Department of Electrical, Electronic & System Engineering, Universiti Kebangsaan Malaysia, Bangi, Selangor, 43600, Malaysia

E-mail: hanifsaad@ukm.edu.my, rabiahshahad@siswa.ukm.edu.my, kwinn.ukm@gmail.com, draini@ukm.edu.my

Abstract— This paper describes the design and development of a component-based object oriented Expert System's Integrated Development Environment (ExSIDE). It is integrated with (i) a user-friendly manual and automated knowledge acquisition and management tool (ExSIDE_KAMT);(ii) an independent and customizable runtime module (ExSIDE_RTM); (iii) an object-oriented in-process Component Object Model (COM)-based inference engine (ExSIDE_IE); (iv) an object-oriented out-of-process COM-based inference engine (ExSIDE_IESvr); (v) and a PHP based inference engine (ExSIDE_PHP). ExSIDE_RTM can function independently as an Expert System Shell (ESS) and helps user to develop Expert Systems rapidly. ExSIDE_IE and ExSIDE_IES can be integrated with COM-supporting general purpose and scientific application development tools such as variants of C/C++/C#, BASIC (Visual BASIC®, REALbasic®, Java, MATLAB®, LabVIEW®, and Mathematica®) to develop more advanced Expert Systems. Finally, ExSIDE_IE and ExSIDE_PHP can be used with Active Server Pages (ASP) and PHP technologies to generate web based Expert Systems. The unique framework of the ExSIDE enables rapid development of Expert Systems' on PC and web for technical and non-technical users. The overall system was developed successfully, and its usability was demonstrated via five unique Expert Systems case studies discussed in this paper.

Keywords— Expert System; Decision Tree; Object Oriented; COM Components.

I. INTRODUCTION

Since their inception more than 40 years ago, Expert Systems have been developed for identification, classification, and decision support in engineering, medicine, agriculture, business, and education. Liao [1] listed more than 100 expert systems developed for application in various fields between 1995-2004. TABLE 1 presents a list of recently developed Expert Systems.

TABLE 1. RECENTLY DEVELOPED EXPERT SYSTEMS

#	Authors	Expert System Applications
1	Zhou <i>et al.</i> (2004) [2]	An intelligent support system for air pollution control at coal-fired power plant
2	Shet <i>et al.</i> (2005) [3]	Video monitoring of human activity.
3	Helman <i>et al.</i> (2005) [4]	Extract and interpret information from video surveillance images.
4	Eldrandaly, K. A. <i>et al.</i> (2005) [5]	A COM-based expert system for selecting the suitable map projection in ArcGIS.
5	Ismail, A. <i>et al.</i> (2007) [6]	Train managers at all levels of the construction industry using state-of-the-art tools, techniques and

6	Krausz, B. & Herpers, R. (2008) [7]	methodologies of project management. Automatically detect and analyze events from a video surveillance system.
7	Qiana <i>et al.</i> (2008) [8]	Helps plant operators in monitoring and diagnosing of abnormal situations in refining process of lubricating oil.
8	Abu-Naser <i>et al.</i> (2008) [9]	Determine plant diseases and provide methods for treatment and protection.
9	H. Heydari Main & M Saadi Mesgari (2009) [10]	Assist urban planners in assessing the suitability of different usages for a piece of land.
10	Abu-Naser <i>et al.</i> (2010) [11]	Endocrine related disease and Diabetes Mellitus diagnosis system in Gaza, Palestin.
11	T. F. Blessia <i>et al.</i> (2011) [12]	Assist medical professionals in diagnosis and treatment of osteoarthritis.
12	Resdiansyah Mansyur <i>et al.</i> (2011). [13]	Select and determine a suitable Transport Demand Management scheme for congestion mitigation.
13	Chena <i>et al.</i> (2012) [14]	A web-based expert system for nutrition diagnosis according to Nutritional Care Process and Model (NCPM) defined by American Dietetic Association (ADA).

A. Available Expert System Development Tools

Several expert system development tools are available, including Kappa-PC [15]), CLIPS, Jess [16], VisiRule, Drools, and d3web. Among them, Kappa PC and CLIPS are the most frequently used. Most of these tools are expert system shells, which are empty expert systems that the user can customize according to specific needs. Newer expert system shells, such as JavaDON [17] and JessGUI (a graphical user interface or GUI for the Jess shell by [18]), are also included with rich GUI for expert systems development. Some expert systems were developed using symbolic programming languages such as Prolog, e.g., VidMAP [3], and Lisp [19], e.g., TMYCIN [20] and VACE [21]. However, learning symbolic programming languages can be challenging to newcomers and users who are not technology savvy. This can hamper the development of expert systems for non-technical field applications. Several attempts were made to bring expert system development process into the general public domain, e.g [22] developed an easy to use web based expert system development tool which they targeted towards non-Artificial Intelligent (AI) experts. JessGUI [18] was an excellent effort in making expert system development easier. However, since it is built on top of Jess, the system development language is primarily restricted to Just Another Vulnerability Announcement (JAVA).

Turban [23] indicated that many expert system development tools have to be supplemented by capabilities provided by other system development tools. Most expert system development tools, especially expert system shells, lack high-quality graphical and numerical computational elements as well as database manipulation facilities and direct hardware interfacing capabilities. Modern scientific application prototyping and development tools, such as MATLAB® and Mathematica®, are good at numerical calculations and visualizations. General-purpose programming languages and Integrated Development Environments (IDE), such as Microsoft Visual Studio .Net® and Real Studio®, are also used to develop scientific applications. These tools offer flexibility in handling databases, excellent GUI, fast execution speed, and a user-friendly environment for application development. Expert systems can be developed using these tools directly. However, the developer must program the inference engine and develop the knowledge acquisition and management tools.

B. Objectives

The objective of this study is to develop a user-friendly and integrated object-oriented expert system development tool inclusive of an Expert System shell, knowledge acquisition and development tools, and a component-based inference engine that can be used to develop Expert Systems for both desktop and web based applications. The developed system will integrate and unify Expert System development process and enables users to develop Expert Systems using the shell approach and also via direct integration with general-purpose programming IDE and scientific application development tools. The user-friendly User Interface (UI) and knowledge representation will allow technical and non-

technical users to develop their own Expert Systems with ease.

C. Theory

1) Main Components of Expert Systems

An expert system consists primarily of an inference engine, a knowledge base, a working memory, and a user interface. The knowledge base consists of a set of explicit rules expressed in the form: if <condition> then <consequence>. The inference engine evaluates the rules, whilst the working memory is used to store current values of parameters, usually obtained via the user interface, to be evaluated against the preset conditions.

2) Decision Tree (DT)

A Decision Tree (DT) is a popular classification algorithm used to derive conclusions. It consists of leaves and nodes. A leaf is the end of a path in the decision tree. Each leaf depicts a possible conclusion. At every node, a decision is made. The outcome of the decision determines the next node or leaf in the path. For classification purposes, the leaves represent the classes that the tree can classify. In a DT, inferences can be made by using the depth first or breadth first approach. A DT can be used as the knowledge base representation and inference mechanism for developing an Expert System [24].

II. MATERIAL & METHOD

The overall system described in this paper is called ExSIDE. TABLE 2 summarizes the implementation strategies for ExSIDE. TABLE 3 shows ExSIDE's component. Fig. 1 presents the overall system framework. ExSIDE_KAMT can be used to create and manage the Knowledge Base (KB). An Expert System can be easily developed via the shell approach (left path in Fig. 1). The Expert System developed can be run using ExSIDE_RTM as standalone application. Advanced developer can integrate ExSIDE's inference engines (ExSIDE_IE and ExSIDE_IES) with other programming IDE to develop complex standalone

TABLE 1. SUMMARY OF EXSIDE'S IMPLEMENTATION STRATEGIES

#	Component	Notes
1	ExSIDE	Main integrated development environment (IDE) For Developing Expert Systems
2	ExSIDE_KAMT	Knowledge acquisition and management tool. Can be used to manually developed the KB (using domain experts) or automatically generate the KB from training data.
3	ExSIDE_RTM	ExSIDE's Desktop Runtime Module. The Expert System shell in ExSIDE infrastructure.
4	ExSIDE_IE	COM In-Process DT Forward Chaining Inference Engine. was developed as an ActiveX-DLL (ActiveX Dynamic Link Library)
5	ExSIDE_IES	COM Out-of- Process DT Forward Chaining Inference Engine. It was developed as an out-of-process ActiveX Exe Server, which can be controlled by client applications.

6	ExSIDE_PHP	PHP based DT Forward Chaining module to cater for web based Expert System development
---	------------	---

Expert Systems. Using the same KB, a web application can be generated from ExSIDE in the form of ASP and PHP codes. The Microsoft® Active Server Pages (ASP) codes uses ExSIDE_IE whereas the Personal Home Page (PHP) codes uses ExSIDE_PHP for inferencing purposes.

Interaction between the user and the system is executed in two ways: (i) via question and answer session with an interacting user and (ii) via direct update to the working memory which stores the value for the variables used for comparison at every node of the DT. Mode (i) is designated as Interactive Operation Mode whereas Mode (ii) is designated as Silent Mode Operation Mode. The Interactive Operation Mode can be used when the Expert Systems to be developed is intended to interact lively with a user (e.g.: medical diagnosis application) whereas the Silent Operation Mode is more useful when the inferencing process is to be integrated with data acquired in real-time from data acquisition system (e.g.: online machine diagnosis system).

TABLE 2. MAIN COMPONENTS OF EXSIDE

#	Feature	Implementation strategy
1	Knowledge representation	Decision Tree
2	Inferencing Strategy	Decision Tree Forward chaining; depth first search
3	Inference Engine Implementation	Desktop Application: COM ActiveX DLL and COM ActiveX exe; Web Application (ASP) COM ActiveX DLL; Web Application (PHP): PHP web module
4	User Interaction / Working Data Acquisition	(i) Question-and-answer based interaction with human (Interactive Mode) and (ii) Direct update of working variable by human user or interaction with hardware (Silent Mode)
1	Knowledge representation	Decision Tree
2	Inferencing Strategy	Decision Tree Forward chaining; depth first search
3	Inference Engine Implementation	Desktop Application: COM ActiveX DLL and COM ActiveX exe; Web Application (ASP) COM ActiveX DLL; Web Application (PHP): PHP web module
4	User Interaction / Working Data Acquisition	(i) Question-and-answer based interaction with human (Interactive Mode) and (ii) Direct update of working variable by human user or interaction with hardware (Silent Mode)

III. RESULTS AND DISCUSSION

ExSIDE was gradually developed since 2003 [25] using Microsoft Visual BASIC® 6.0 programming environment. ExSIDE and its components were developed for execution on Microsoft Windows-based operating systems (e.g., Windows XP 2000, Vista, 7). During the earlier

development stage, ExSIDE was known as the Simple Expert Systems Development Tool (SESST). The system was renamed as ExSIDE to accommodate latest enhancement and upgrades. To date, ExSIDE (and previously SESST) has been used to develop several Expert Systems [25], [26].

The KB used in ExSIDE is represented in the form of a DT. It can be constructed manually by a domain expert. The

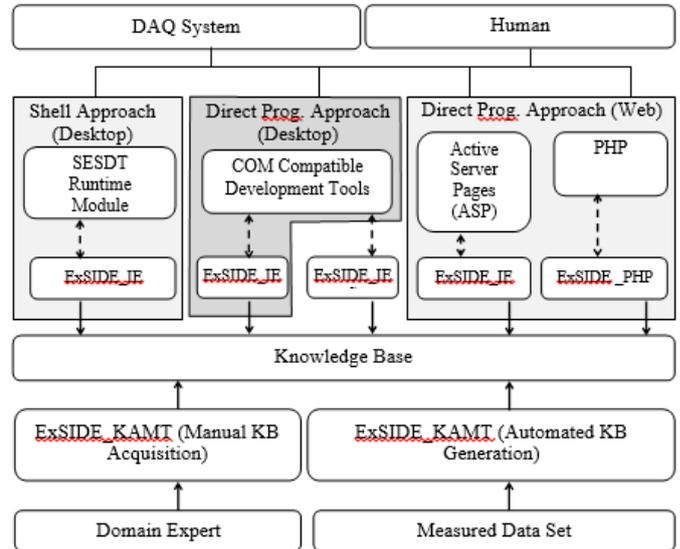


Fig. 1 ExSIDE Overall Framework

KB can also be generated automatically by ExSIDE. Given a set of training data (X1, X2, X3, .. Xn, Y), where Xn and Y is a vector of m-elements, ExSIDE_KAMT will generate a vector P with q (q<=m) elements representing the identification rules. The vector Xn represents the values for each input variables and the vector Y denotes the corresponding identified class.

For shell based Expert System, ExSIDE will generate an independent executable file runnable on Windows based operating system. This executable file is integrated with ExSIDE.RTM. The user interface for the shell based Expert System can be designed using ExSIDE form designer. ExSIDE_RTM also supports scripting, via Microsoft's VBScript technology, to further customize the user interface.

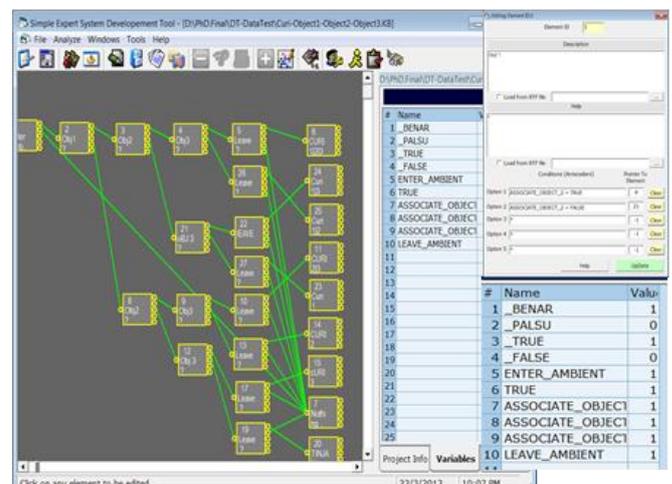


Fig. 2. ExSIDE main user interface (left); Details of node info (top right), Attribute/Variables and Facts/Working Memory (bottom right)

CODE LIST 1. EXAMPLE OF AN EXSIDE.IE APPLICATION IN VISUAL BASIC 6.0

For Expert Systems developed using ExSIDE's inference engine, the tool that is used to develop the Expert System (e.g: Microsoft Visual C#) will generate the necessary executable file (or script, if it is used with script based tool e.g.: Matlab) as the Expert System.

ExSIDE can also be used to develop web based Expert Systems using Microsoft Active Server Pages (ASP) and the PHP (Personal Home Page / PHP:Hypertext Preprocessor) technology. ExSIDE's compiler will automatically generate the necessary ASP and PHP codes from the KB.

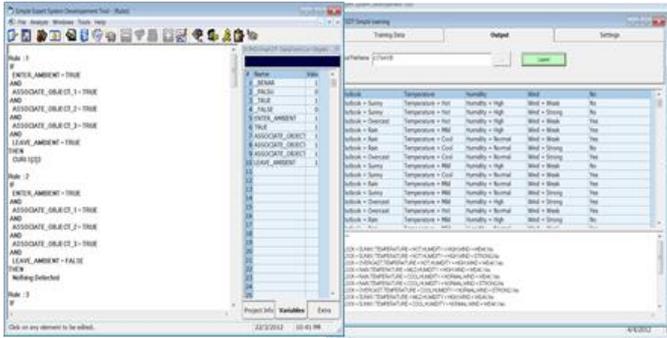


Fig. 3. Rules generated from KB manually developed using ExSIDE_KAMT (Left) and Automatic Generation of KB From Measured Training Data using ExSIDE_KAMT

```
Dim ExSIDE_IE as new ExSIDE.IE
Dim InParam(5) as string, OutParam(5) as string,
Result as long
InParam(0) = 'D:\RoomSurveillance.KB'
' Set the Knowledge base filename
ExSIDE_IE.SendMessage("KB_SET
_FILENAME",InParam,OutParam)
' Load the Knowledge base File
ExSIDE_IE.SendMessage
("KB_LOAD_FILE",InParam,OutParam)
' Set Variable Object1_Removed Value to 1
InParam(0) = "Object1_Removed" : InParam(1) = "1"
ExSIDE_IE.SendMessage("VARIABLE_UPDATE_VALUE
",InParam,OutParam)
' Get and Display The Result
ExSIDE_IE.SendMessage("SYS_EVALUATE_RULE
",InParam,OutParam)
if OutParam(0) = "" then MsgBox "The Result
Is:"+OutParam(1)
```

CODE LIST 2. EXSIDE: COMPONENT BASED OBJECT ORIENTED

```
// Create the type variable
ExSIDE_IES_Type =
System.Type.GetTypeFromProgID("ExSIDE_IES.App");
// Initialize the component
ExSIDE_IES =
System.Activator.CreateInstance(ExSIDE_IES_Type);
// Set Knowledge Base FileName
InParam[1] =
String.Copy("d:\\RoomSurveillance.KB");
param[0] = String.Copy("KB_SET_FILENAME");
param[1] = InParam; param[2] = OutParam;
ExSIDE_IES_Type.InvokeMember("SendMessage",
System.Reflection.BindingFlags.InvokeMethod,null,
ExSIDE_IES, param);
// Load Knowledge Base FileName
param[0] = String.Copy("KB_LOAD_FILE");
param[1] = InParam; param[2] = OutParam;
ExSIDE_IES_Type.InvokeMember("SendMessage",
System.Reflection.BindingFlags.InvokeMethod,null,
ExSIDE_IES, param);
// Display Main Window
param[0] = String.Copy("SYS_SHOW_WINDOW");
param[1] = InParam; param[2] = OutParam;
ExSIDE_IES_Type.InvokeMember("SendMessage",
System.Reflection.BindingFlags.InvokeMethod,null,
ExSIDE_IES, param);
null, ExSIDE_Svr, param);
```

CODE LIST 3. EXSIDE.IES APPLICATION IN MATLAB®

```
% Initiate the server
svrExSIDE = actxserver(' ExSIDE_IES.App');
% Set Input and output parameters
InParam = 'D:\RoomSurveillance.KB';OutParam = '';
% Set the KB filename
Result = svrExSIDE.SendMessageSingle(
'KB_SET_FILENAME',InParam,OutParam);
%Load the KB File
Result = svrExSIDE.SendMessageSingle('KB_LOAD
_FILE',InParam,OutParam);
% Show the main window
Result
=svrExSIDE.SendMessageSingle('SYS_SHOW_WINDOW',InParam,
OutParam);
```

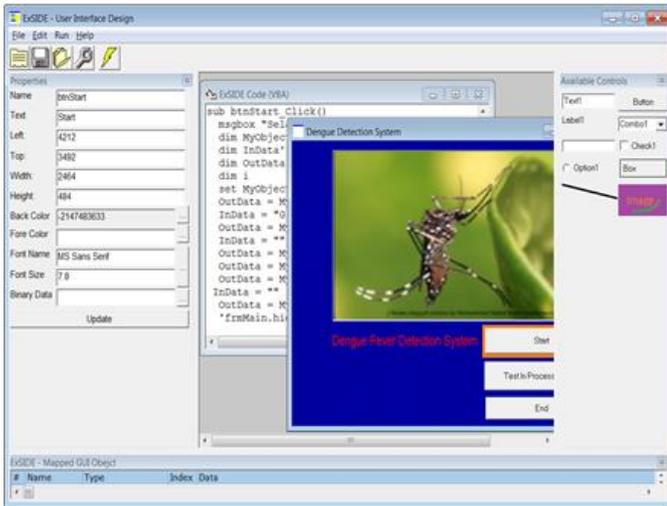


Fig. 4. User Interface Design for ExSIDE Shell Based Expert Systems

A. Integrating ExSIDE Inference Engine (ExSIDE_IE and ExSIDE_IES) With General Purpose and Scientific Application Development Tool

CODE LIST 1 shows the application of ExSIDE_IE in Microsoft Visual BASIC® 6.0 (Instantiation of ExSIDE_IE class, loading a KB, updating a variable, and evaluating the KB). CODE LIST 2 shows how the ExSIDE_IES can be used in the Microsoft Visual C# 2008 and later IDE (class instantiation, loading the KB file, and showing the KB in the server component). CODE LIST 3 shows how ExSIDE_IES can be used with MATLAB® (class instantiation, loading the KB file, and showing the KB in the server component).

B. Case Studies: Applications of ExSIDE in Developing Expert Systems and Scientific Application Development Tool

Five case studies of Expert Systems developed using ExSIDE are discussed as in TABLE 4.

TABLE 3. CASE STUDY OF EXPERT SYSTEM

#	Case Study	Expert System Type	Data Entry	Component / (External Software) Used
1	Power Quality Analysis System	Desktop Application, Direct Programming, Silent Operation Mode	Automatic Acquisition From Hardware	ExSIDE, ExSIDE_IE, (VB6, Matlab)
2	Intelligent Video Surveillance System	Desktop Application, Direct Programming, Silent Operation Mode	Automatic Acquisition From Hardware	ExSIDE, ExSIDE_IES, (Visual C# 2008 and above)
3	Dengue Fever Diagnosis	Desktop Application, Shell Based, Interactive Operation Mode	Human Input	ExSIDE, ExSIDE_RTM
4	Dengue Fever Diagnosis	ASP Web App, Direct Programming, Interactive Operation Mode	Human Input	ExSIDE, ExSIDE_IE
5	Dengue Fever Diagnosis	PHP Web App, Direct Programming, Interactive Operation Mode	Human Input	ExSIDE, ExSIDE_PHP

C. Case study 1: Power Quality Analysis

The Intelligent Power Quality Assessment Tool (IPQDA) [27] is an Expert System that can be used to evaluate the power quality of a recorded voltage signal [27].

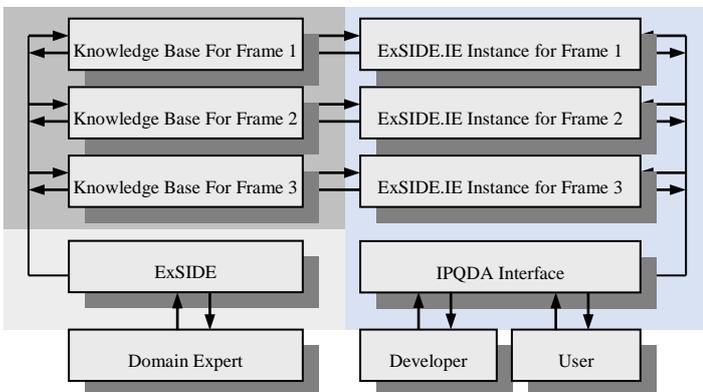


Fig. 5. IPQDA framework [27]

The signal was windowed into three overlapping frames. The data were filtered, and the fundamental frequency was removed. Finally, linear predictive coding (LPC) and the Fast Fourier Transform (FFT)-based coefficients were extracted and fed into the KB for evaluation. Fig. 5 presents the framework for the IPQDA.

D. Case study 1: Intelligent Video Surveillance System (InViSS™)

ExSIDE and ExSIDE_IES were used to develop an online Expert System-based video event detection, identification, and management system called InViSS™ (Intelligent Video Surveillance System). Fig. 6 shows InViSS™ automatically detecting a person picking up several items. InViSS™ acquires online video images from surveillance cameras, identifies the actors and objects, and recognizes their activities in real-time. ExSIDE_KAMT was used to develop the KB, while ExSIDE_IES evaluates the activities (the person took, moved, swapped, or inspected the objects). InViSS™ KB was automatically generated by ExSIDE_KAMT from 150 annotated video data of actor interaction with object in a monitored environment. Once an anomalous activity is detected, the management module of the InViSS™ takes action.

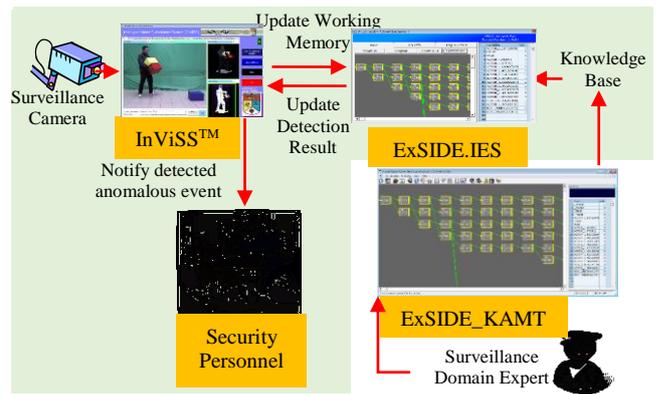


Fig.6. InViSS™ Overall Infrastructure

E. Case study 3: Dengue Fever Diagnosis Support System (DFDSS)

Dengue fever (DF) is a life-threatening disease caused by the mosquito *Aedes aegypti*, which is common in Asia [28]. DF is an acute febrile viral disease frequently presented with headache, bone or joint and muscular pains, and rash as symptoms [29]. In a small percentage of dengue infections, a more severe form of the disease, which is the dengue haemorrhagic fever (DHF), occurs. DHF is characterized by acute fever associated with a haemorrhagic diathesis and the tendency to develop shock, which can lead to mortality. Cases of DHF are classified according to WHO specifications into four grades [30]. In Grade I, the patient has a fever accompanied by non-specific constitutional symptoms, and the only haemorrhagic manifestation is a positive tourniquet test that results in petechial rash. In Grade II, the patient experiences spontaneous bleeding from any part of the body through the skin. In Grade III, the patient experiences circulatory failure manifested by a rapid and weak pulse, the narrowing of pulse pressure or

hypotension, cold and clammy skin, and restlessness. In Grade IV, the patient experiences profound shock with undetectable blood and pulse [31].

This case study (Dengue Fever Diagnosis Support System, DFDSS) was implemented using slight variations of the work by [31]. They developed a DT that could be used to classify DF and DHF Levels I and II based on clinical symptoms, such as fever, headache, taste aberration, arthralgia, weakness of lower limbs, myalgia, vomiting, retro-orbital pain, abdominal pain, maculopapular rash, petechial rash, bleeding gums, blood-stained saliva, and ecchymosis [31]. Fig. 7(left) shows the DT for detecting DF and DHF Levels I and II. DFDSS was designed as shell based Expert System. The system requires continuous user interaction and provides an explanation facility to the user as illustrated in Fig. 7 (right).

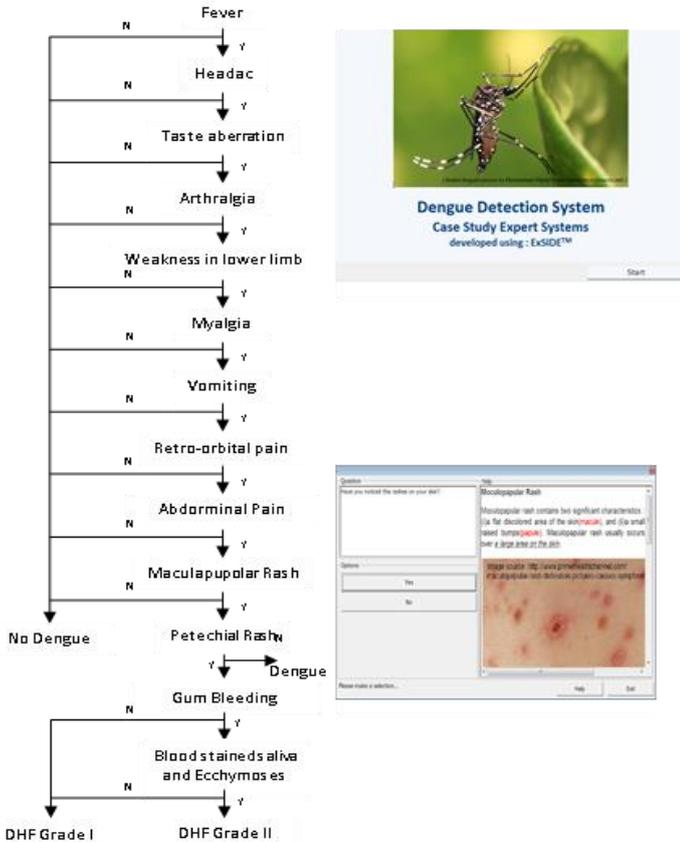


Fig. 7. DF and DF Levels I and II Dengue Fever Diagnostic DT (Left), DF Diagnostic Expert Systems: Introductory page (top right) & diagnostic session (bottom right)

F. Case Study 4: DF Diagnosis support system over the web (Using ASP Technology)

Case Study 4 is a web version of Case Study 3. It was developed using Active Server Pages (ASP) and VB Script. The desktop based expert system in Case Study 3 requires continuous user interaction whereas Case Study 4 only requires user interaction in the beginning part where user inputs the necessary data. The inference component used was ExSIDE_IE. The resulting web page is shown in Fig. 8.

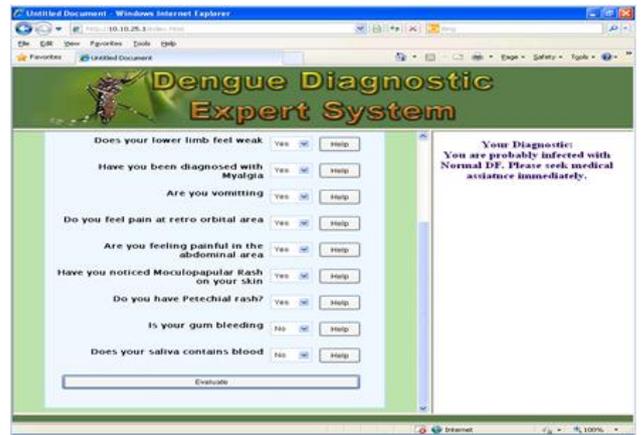


Fig. 8. Web based Dengue fever diagnostic expert systems migrated from case study

G. Case Study 5: DF Diagnosis support system over the web (Using PHP)

ASP is a proprietary Microsoft technology. Mostly, the web servers must run Microsoft Windows based operating system and some compatibility issues exist when the webpage is browsed using web browsers other than Microsoft's own Internet Explorer. Another server scripting technology which is more robust and widely supported is PHP. In Case Study 5, a PHP based web Expert System version of Case Study 3 was generated from ExSIDE. In this case study, inferencing process is done using ExSIDE_PHP. The web page is shown in Fig. 9.



Fig. 9. Web Based Dengue Fever Diagnostic Expert Systems, PHP Version (Case Study 5)

IV. CONCLUSION & FUTURE WORK

The desired Expert System IDE was successfully designed, developed, and tested. To summarize, ExSIDE:

- (i) allows technical and non-technical users to rapidly develop Expert Systems with ease via the graphical KB development and independent runtime module (Case Study 3);

- (ii) allows competent technical users to create Expert Systems via the integration of ExSIDE's inference engine (ExSIDE_IE and ExSIDE_IES) with general-purpose programming languages (C#/Visual C#, Visual BASIC, C/C++/Visual C++, plus other COM-supporting languages), and scientific & engineering development tools (MATLAB, LabVIEW, plus other tools supporting COM-integration) (Case Study 1 & 2);
- (iii) allows users to automatically generate a KB from training data (Case Study 2);
- (iv) can be used to develop desktop based Expert Systems application (Case Studies 1, 2, 3) and web-based Expert Systems (Case Study 4 & 5); and
- (v) can be used to develop Expert Systems which operates without user interaction (Case Studies 1 & 2), and Expert Systems which requires user interaction (Case Studies 3, 4 & 5).

Based on the list of achieved capabilities, we conclude that the developed system (ExSIDE) has achieved its objectives.

The ExSIDE development team plans to improve the system's learning capability and to upgrade the shell-based development tools. These tasks will involve enriching the shell-based development tools with a powerful GUI interface and embedded resources (pictures and videos) into the runtime data file. Case study 2, InViSSTM, is an ongoing project that will also be improved. ExSIDE will also be enhanced in terms of inference execution speed and KB analysis.

The development team also endeavors to ensure that ExSIDE remains freely available for the academic, scientific, and education communities.

ACKNOWLEDGMENT

The authors acknowledge the assistance of Ms. Norizan bt Baharuddin and Mr. Liang Xian Loong in preparing this manuscript and testing the expert systems. The authors also thank the government of Malaysia for sponsoring this research under the DIP-2012-03 Research Grant. The authors would also like to express their attitude to Mr Muhammad Mahdi Karim (<http://www.micro2macro.net/>) for allowing us to use the Aedes Aegypti picture in Case Studies 3, 4 & 5. The pictures of medical rash shown in Case Studies 3, 4 & 5 belong to the respective websites (indicated in the picture). They are not integrated with ExSIDE.

REFERENCES

1. S.-H. Liao, "Expert system methodologies and applications a decade review from 1995 to 2004," *Expert Syst. Appl.*, vol. 28, no. 1, pp. 93–103, 2005.
2. Q. Zhou, G. H. Huang, and C. W. Chan, "Development of an intelligent decision support system for air pollution control at coal-fired power plants," *Expert Syst. Appl.*, vol. 26, no. 3, pp. 335–356, 2004.
3. V. D. Shet, D. Harwood, and L. S. Davis, "Vidmap: video monitoring of activity with prolog," in *Advanced Video and Signal Based Surveillance*, 2005. AVSS 2005. IEEE Conference on, 2005, pp. 224–229.
4. H. Stern, U. Kartoun, and A. Shmilovici, "An expert system for surveillance picture understanding," *NATO Sci. Ser. SUB Ser. III Comput. Syst. Sci.*, vol. 198, p. 542, 2005.
5. K. A. Eldrandaly, "A COM-based expert system for selecting the suitable map projection in ArcGIS," *Expert Syst. Appl.*, vol. 31, no. 1, pp. 94–100, 2006.
6. A. Ismail, A. M. Abd, and Z. Bin Chik, "Modeling of risk assessment for integrated project management system in construction," 2006.
7. B. Krausz and R. Herpers, "Event detection for video surveillance using an expert system," in *Proceedings of the 1st ACM workshop on Analysis and retrieval of events/actions and workflows in video streams*, 2008, pp. 49–56.
8. Y. Qian, L. Xu, X. Li, L. Lin, and A. Kraslawski, "LUBRES: An expert system development and implementation for real-time fault diagnosis of a lubricating oil refining process," *Expert Syst. Appl.*, vol. 35, no. 3, pp. 1252–1266, 2008.
9. S. S. Abu-Naser, K. A. Kashkash, and M. Fayyad, "Developing an expert system for plant disease diagnosis," *J. Artif. Intell.*, vol. 1, pp. 78–85, 2008.
10. H. Heydari Main and M. Saadi Mesgari, "Developing a knowledge-based spatial decision support system for urban landuse allocation," *J. Appl. Sci.*, vol. 9, no. 9, pp. 1758–1763, 2009.
11. S. S. Abu-Naser, H. El-Hissi, M. Abu-Rass, and N. El-Khozondar, "An expert system for endocrine diagnosis and treatments using JESS," *J. Artif. Intell.*, vol. 3, no. 4, pp. 239–251, 2010.
12. T. F. Blessia, S. Singh, A. Kumar, and J. J. Vennila, "Application of knowledge based system for diagnosis of osteoarthritis," *J. Artif. Intell.*, vol. 4, pp. 269–278, 2011.
13. R. Mansyur, R. A. O. K. Rahmat, A. Ismail, and M. R. Kabit, "Decision support system for transport demand management: Object oriented approach using kappa PC 2.4 expert system shell," *ARPN J. Eng. Appl. Sci.*, vol. 6, no. 2, pp. 73–81, 2011.
14. Y. Chen, C.-Y. Hsu, L. Liu, and S. Yang, "Constructing a nutrition diagnosis expert system," *Expert Syst. Appl.*, vol. 39, no. 2, pp. 2132–2156, 2012.
15. T. Lydiard, "Kappa-PC," *IEEE Expert*, vol. 5, no. 5, pp. 71–77, 1990.
16. E. Friedman-Hill and L. Sandia, "Jess, the rule engine for the Java platform (2008)," Sandia Natl. Lab., 2003.
17. B. Tomić, J. Jovanović, and V. Devedžić, "JavaDON: an open-source expert system shell," *Expert Syst. Appl.*, vol. 31, no. 3, pp. 595–606, 2006.
18. J. Jovanović, D. Gašević, and V. Devedžić, "A GUI for jess," *Expert Syst. Appl.*, vol. 26, no. 4, pp. 625–637, 2004.
19. D. Verna, "How to make Lisp go faster than C," in *Lecture Notes in Engineering and Computer Science*, 2006, pp. 815–820.
20. G. Novak, *TMYCIN Expert System Tool*. Computer Science Department, University of Texas at Austin, 1987.
21. S. Guler, W. H. Liang, and I. A. Pushee, "A video event detection and mining framework," in *Computer Vision and Pattern Recognition Workshop*, 2003. CVPRW'03. Conference on, 2003, vol. 4, p. 42.
22. B. Ruiz-Mezcua, A. Garcia-Crespo, J. L. Lopez-Cuadrado, and I. Gonzalez-Carrasco, "An expert system development tool for non AI experts," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 597–609, 2011.
23. E. Turban, "Review of expert systems technology," *Eng. Manag. IEEE Trans.*, vol. 35, no. 2, pp. 71–81, 1988.
24. U. H. Rao and S. Mohapatra, "Deploying network management solutions in enterprises," in *Networked Computing (INC)*, 2010 6th International Conference on, 2010, pp. 1–6.
25. R. N. Mohd Jailaini Mohd Noor, Mohamad Hanif Md Saad, "Initial Design and Development of An Expert System Shell Framework For Engineering and Scientific Application," in *2nd Technical Postgraduate (TechPos) Symposium*, 2003, pp. 1–6.
26. A. Hussain, A. Mohamed, M. H. M. Saad, M. H. Shukairi, and N. S. Sayuti, "IPQDA: A software tool for Intelligent Analysis of Power Quality Disturbances," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3809 LNAI, pp. 1315–1318, 2005.
27. N. S. Hussain, A. Mohamed, A. Saad, Mohamad Hanif M., Sukairi, M.H., and Sayuti, "IPQDA: A Software Tool For Intelligent Analysis of Power Quality Disturbances," *Ina. 2005 Adv. Artif. Intell.*, vol. Vol. 3809, pp. pp. 1315–1318, 2005.
28. K. Chang, P.-L. Lu, W.-C. Ko, J.-J. Tsai, W.-H. Tsai, C.-D. Chen, Y.-H. Chen, T.-C. Chen, H.-C. Hsieh, C.-Y. Pan, and others, "Dengue fever scoring system: new strategy for the early

- detection of acute dengue virus infection in Taiwan,” *J. Formos. Med. Assoc.*, vol. 108, no. 11, pp. 879–885, 2009.
29. W. H. Organization, “Dengue Hemorrhagic Fever, Diagnosis, treatment, prevention and control, 2nd Edition, Geneva,” 1997.
 30. W. H. Organization, “WHO Report: Technical guides for diagnosis treatment, surveillance, prevention and control of Dengue Hemorrhagic Fever.,” 1975.
 31. F. Ibrahim, M. N. Taib, S. Sulaiman, and W. A. B. W. Abas, “Dengue fever (DF) and dengue haemorrhagic fever (DHF) symptoms analysis from an expert system perspective,” in *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International, 2001*, pp. 212–215.