Knowledge-based System for diagnosing and troubleshooting common computer problems

Emil Darío Vega Gualán
María Fernanda Molina Ron

December 15, 2017

Abstract

Your abstract.

1 Introduction

Since the beginnings of the computer people have invested great efforts to improve the quality of life of human beings. For this, an attempt has been made to give the machines certain decision-making capacity, that is, a certain level of intelligence. This capacity can be used in any field of knowledge such as: medical, education, finance, transportation, computer science, telecommunication, etc.

Normally, computer systems can face hardware or software problems from time to time and endanger the functionality of the computer. These problems need a computer repair technician to be solves. The technicians follow traditional manuals of solving computer problems which may take to long time and high cost.
There are many diagnostic tools that are included as standard programs for Microsoft software systems. There are also many diagnostic tools available from third-party vendors that can assist in the troubleshooting process. Third-party vendor programs range from freeware and shareware to systems costing several thousands of dollars. But most problems can be diagnosed without expensive system diagnostics.[9]

The objective of the present study is to develop a computer troubleshooting Knowledge-based system that gives the correct diagnosis and recommendations to the most common and simple problems that a computer present without the need to call to a technician or spend large amounts of money for these services. Also counts with a friendly interface for the user, taking into account that not all users are familiar with computer topics.

1.1 Related Works

Hewlett-Packard Development Company,also known as HP Company, developed a PC Troubleshooting and Maintenance Guide.[2] The guide is developed to offer a list of solutions for software and hardware problems as well as offers a simple guide for the PC maintenance. A similar help is provide by the web page of Microsoft - Windows Help [8].

The book written by Bayo Akinnola in 2012 called Computer Troubleshooting, Using an Expert System: A Research Work [1] discusses the evolution of computer systems and designed an expert system to provide a easy access to some of the solution of the troubleshooting performed on the computer system.

1.2 Knowledge-based System

According to Dhananjay Kalbande on his slides *Knowledge based expert systems* [7] a knowledge-based System is computer program that uses artificial intelligence to solve problems within a specialized domain that ordinarily requires task-specific knowledge.

Knowledge-based system is a more general than the expert system. It becomes an expert system if it provides expert-level solutions or human expertise.

1.2.1 Components of a Knowledge-based System:

Knowledge base: Contains the system's knowledge organized in collection of facts about the system's domain. Knowledge is represented in the form of rules using IF-THEN-ELSE.

Inference Engine: This is the brain of the expert system that provides a methodology for reasoning about the information in the knowledge base, and derives an answer.

User Interface: Used by the user to communicate with the knowledge base.

2 Knowledge-based System for diagnosing and troubleshooting common computer problems.

2.1 Knowledge acquisition

"Knowledge acquisition (KA) is the process of acquiring relevant knowledge from domain experts and other sources of information such as books, databases, guidelines, manuals, journal articles and computer files." [5]. The knowledge acquisition of this project was based on manuals and articles of troubleshooting [3] founded on Internet.

2.2 Plan and Design

This Knowledge-based System wants to give a diagnosis of the possible failures that a computer can present and suggests a solution to these failures without having to appeal to a technician or spend large amounts of money for these services

The most difficult process lies in the diagnosis of the failure. That is why this system will only cover the simplest faults to detect. Generally these are usually those that are linked to audio, video, hard drive disk, monitor, mouse and keyboard.

2.2.1 Knowledge Representation and Conceptual Modeling:

In order to modeling the system we are going to use Decision trees and the knowledge representation in propositional Logic.

Detection of Power problems:

Description:
p= The computer turns on
q= Hear a beep
r= Hear a continuous tone
s= unexpected restart
t= unexpected shutdown
v= Power Problems

- **Rule 1:** If the computer does not turn on and no beep is heard then there is a Power Problem .

$$(\sim p \wedge \sim q) \rightarrow v$$

- **Rule 2:** If the computer does not turn on and a continuous tone is heard then there is a Power Problem .

$$(\sim p \wedge r) \rightarrow v$$

- **Rule 3:** If the computer turns on and an unexpected restart happened then there is a Power Problem .

$$(p \wedge s) \rightarrow v$$

- **Rule 4:** If the computer turns on and an unexpected shutdown happened then there is a Power Problem .

$$(p \wedge t) \rightarrow v$$

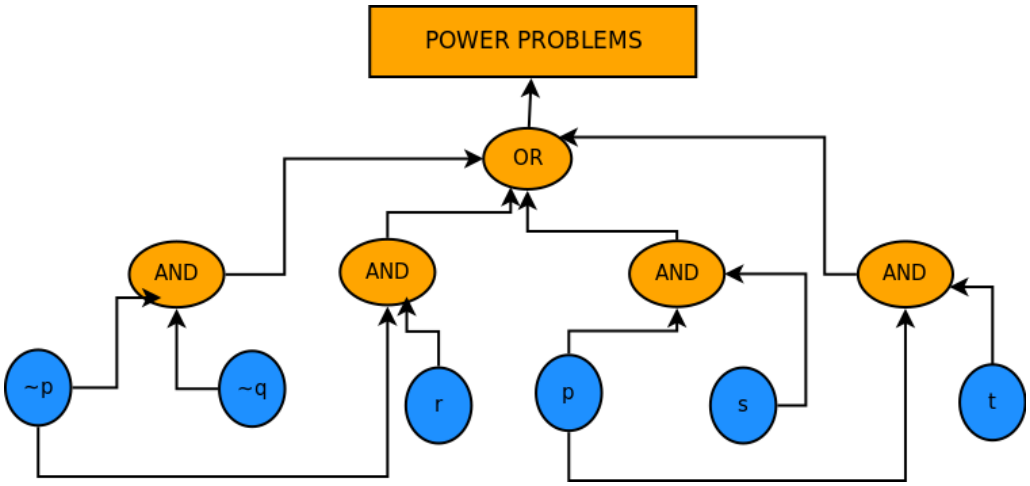

Figure 1: Decision tree for power problems

Detection of Monitor/Display problems:

Detection of keyboard problems:

Description:
p= The computer turns on
q= Hear six short beeps
r= Keyboard doesn't respond
s= Keyboard is not recognized by the PC
t= Keyboard Problems

- **Rule 1:** If the computer does not turn on and six short beeps are heard then there is a keyboard problem .

$$(\sim p \wedge q) \rightarrow t$$

- **Rule 2:** If the computer turns on and the keyboard doesn't respond then there is a keyboard problem.

$$(p \wedge r) \rightarrow t$$

- **Rule 3:** If the computer turns on and an the keyboard is not recognized by the PC then there is a power problem.

$$(p \wedge s) \rightarrow t$$

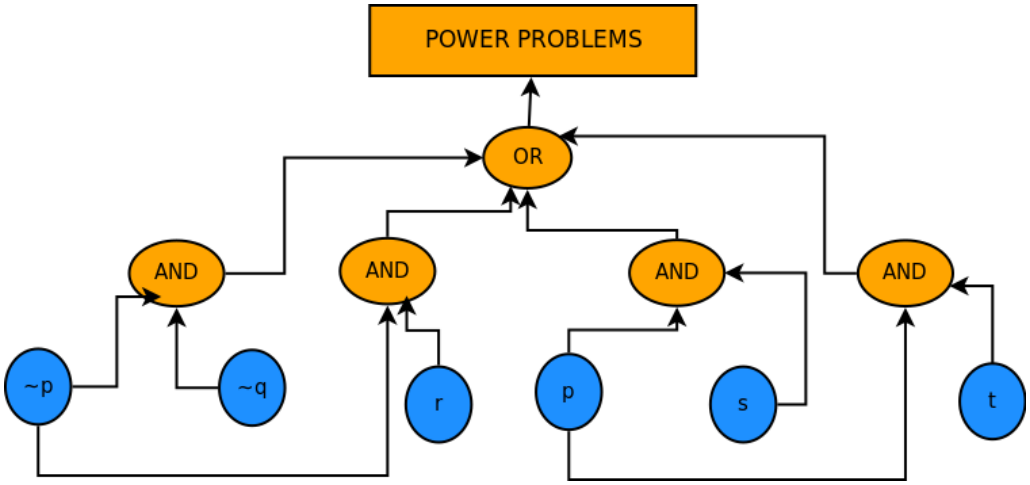

Figure 2: Decision tree for power problems

Detection of Mouse problems:

Description:
p= The computer turns on
q= The mouse turns on
r= The mouse does not respond
s= The computer recognizes the mouse
t= Mouse problem

- **Rule 1:** If the computer turns on and the mouse also turn on but mouse does not respond then there is a mouse problem.

$$(p \wedge q \wedge r) \rightarrow t$$

- **Rule 2:** If the computer turns on and it does not recognize the mouse then there is a mouse problem.

$$(p \wedge \sim q) \rightarrow t$$

- **Rule 3:** If the computer turns on and it recognizes the mouse but the mouse does not respond then there is a mouse problem.

$$(p \wedge s \wedge r) \rightarrow t$$

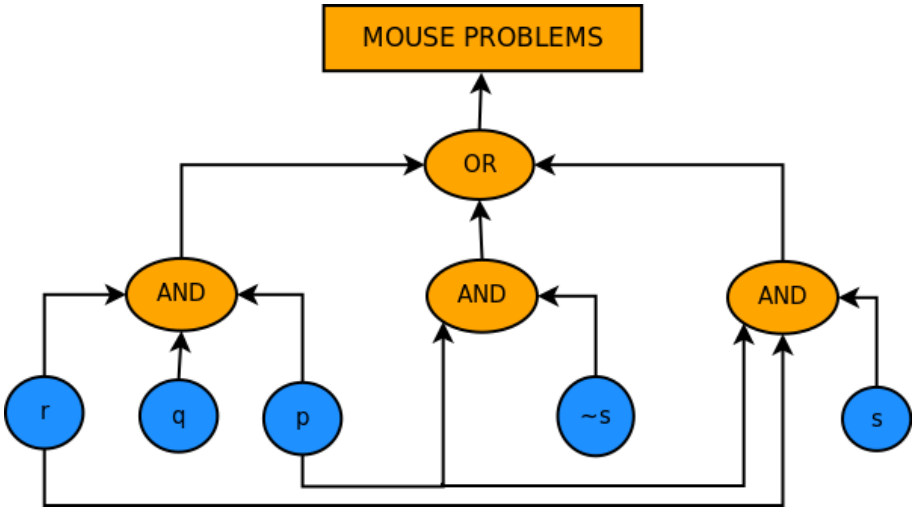

Figure 3: Decision tree for mouse problems

Detection of Audio problems:

Detection of Video problems:

Description:
p= Video does not play
q= A message appears
r= Video problem

- **Rule 1:** If the video does not play then there is a video problem.

$$p \rightarrow r$$

- **Rule 2:** If the video does not play and a message appears then there is a video problem.

$$(p \wedge q) \rightarrow r$$

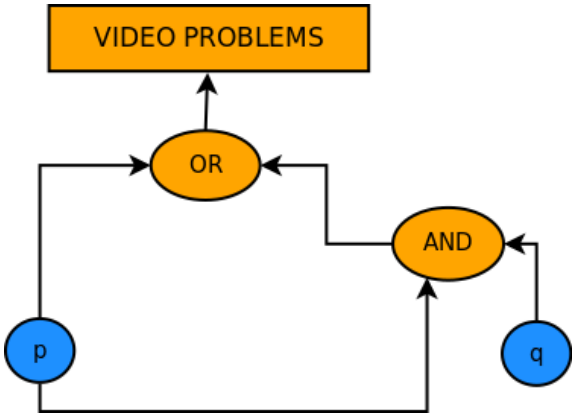

Figure 4: Decision tree for video problems

Detection of Hard Disk Drive problems:

Description:
p= The computer turns on
q= The operating system is slow
r= There are constant restarts
s= Blue screen-shots appears
t= There are errors when saving files
v= Metallic noise is heard
w= Hard disk drive problem

- **Rule 1:** If the computer turns on and the operating system is going slow and there are constant restarts then there is a HDD problem.

$$(p \wedge q \wedge r) \rightarrow w$$

- **Rule 2:** If the computer turns on and blue screen-shots appears and there are errors when saving files then there is a HDD problem.

$$(p \wedge s \wedge t) \rightarrow w$$

- **Rule 3:** If the computer turns on and blue screen-shots appears and metallic noise is heard then there is a HDD problem.

$$(p \wedge s \wedge v) \rightarrow w$$

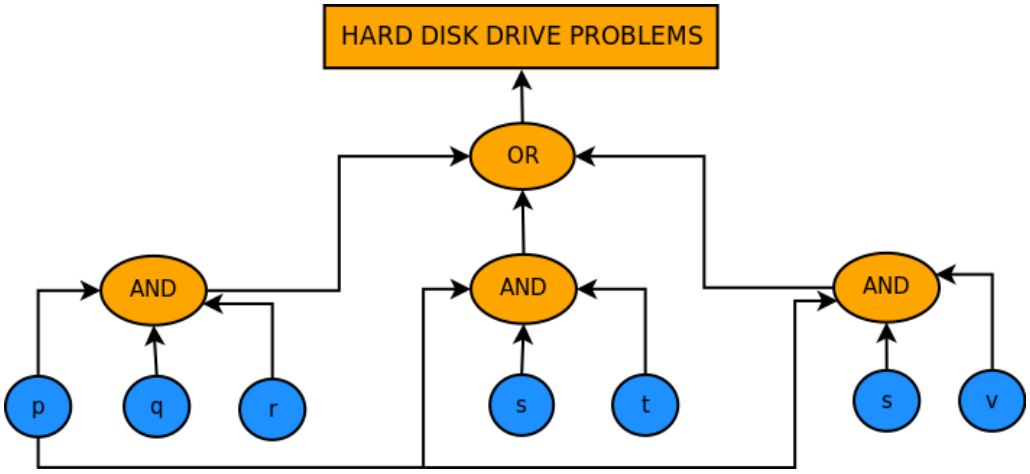

Figure 5: Decision tree for hard disk drive problems

Miscellaneous problems:

Description:
p= The computer turns on
q= Unexpected closures
r= The operating system freezes
s= Incorrect time and date
t= Drivers or spy-ware problem
u= Clock battery problem

- **Rule 1:** If the computer turns on and there are unexpected closures and the operating system freezes then there is a drivers or spy-ware problem.

$$(p \wedge q \wedge r) \rightarrow t$$

- **Rule 2:** If the computer turns on and the time and date area incorrect then there is a clock battery problem.

$$(p \wedge s) \rightarrow u$$

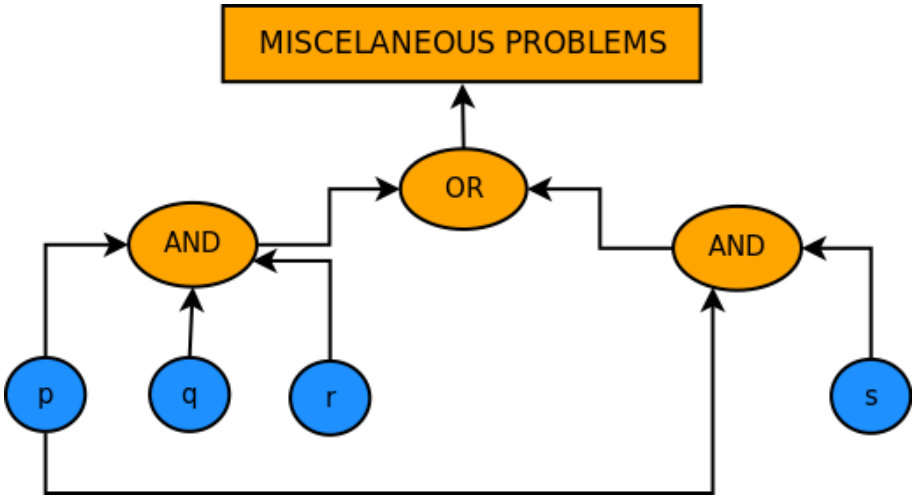

Figure 6: Decision tree miscellaneous problems

2.3 Implementation

The programming language that was used was Java. Java is a language oriented to objects. It means that Java is an practical and easy language to program [6]. The IDE used was Eclipse. Eclipse allows to correct the errors easier. Furthermore, Eclipse has WindowBuilder which is a very useful GUI designer. These were the main tools which were used to develop our expert system. Another resource was MiGlayout which is the most versatile and flexible Swing and SWT Layout Manager for Java [4].

In order to develop the application we crate five classes: Main.java, TurOn.java, Beep.Java, Failures.java, DiagnosisNRecomendation.java. These classes were created for a better organization and understanding of how the application works.

2.4 Testing and Evaluation

The application when the computer does not turn on, it ask if the user hears a beep as show the figure 7.

Figure 7: Question for beeps.

As fig 7 there are many windows with question to guide the user to an accurate solution. Furthermore, there are windows that show several options to decide like show the figure 8.

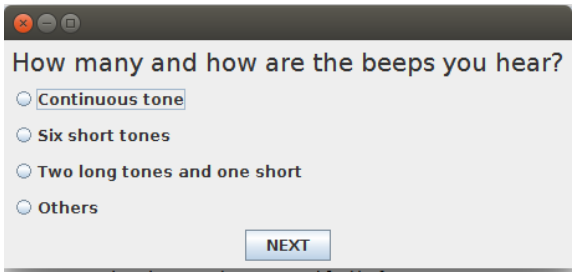

Figure 8: Options to decide

The fig 9 shows all the options of possible failures when the computer is on.

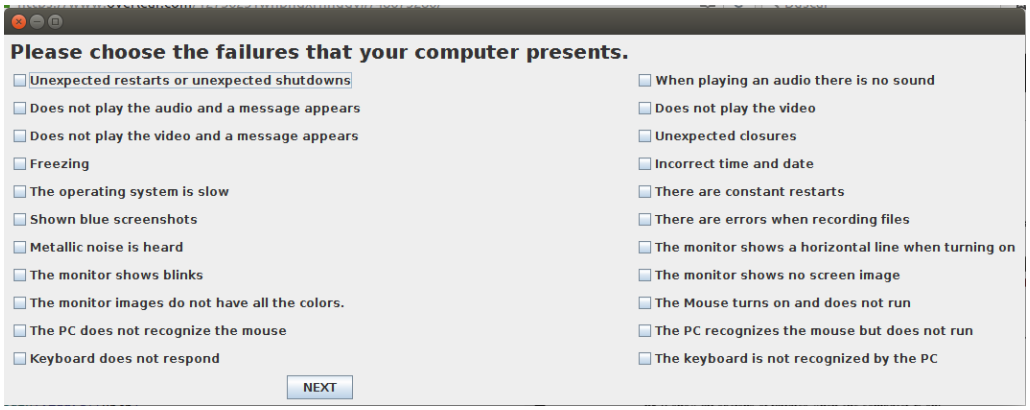

Figure 9: Menu of failures

The figure 10 shows the window of diagnosis and some recommendations depending on the faults that the computer has. It also shows some notes that the user should take into account.

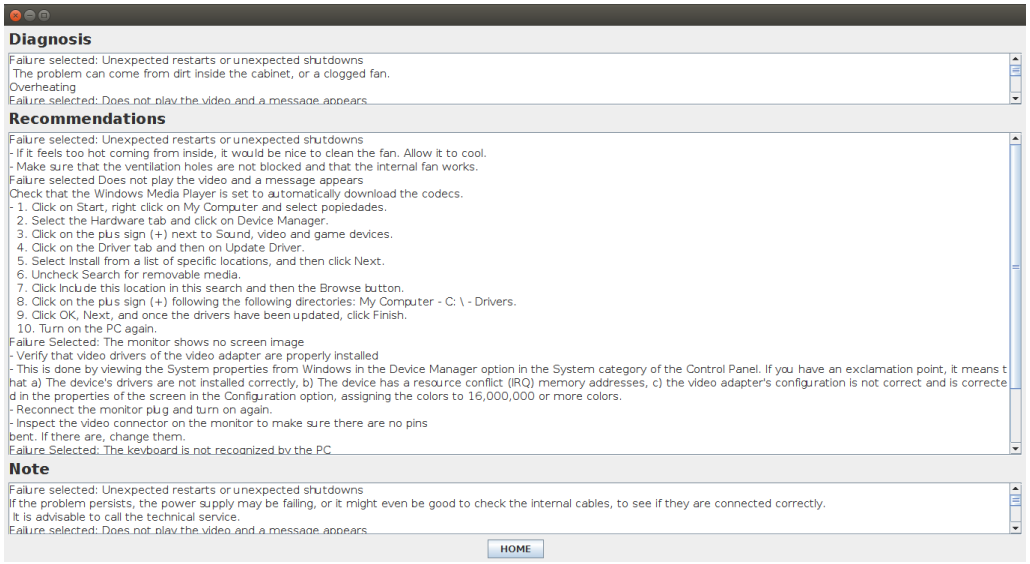

Figure 10: Diagnosis and recommendations

2.5 Recommendations

- The application could be extended to a larger field, including more failures.
- The application could be modified to give more explicit solutions with examples and some figures.
- If you want to have more detailed solutions it is recommended to have an expert engineer in technical failures.
- To make the application more accessible, it may be good to implement a version for smartphones.
- In order to have a better application experience, you should perhaps classify the users according to their level of computer knowledge.

References

[1] B. Akinnola. *Computer Troubleshooting, Using an Expert System: A Research Work.* LAP Lambert Academic Publishing, 2012. Retrieved from `https://dl.acm.org/citation.cfm?id=2222699`.

[2] H. P. Company. *PC Troubleshooting and Maintenance Guide.* Hewlett-Packard Development Company, 2005-2006. Retrieved from `http://h10032.www1.hp.com/ctg/Manual/c00757358.pdf`.

[3] E. Cueva and Y. Gonzales. Desarrollo de un sistema experto para el diagn[ostico de fallas en computadoras y sus posibles soluciones. 2015. Retrieved from `https://es.scribd.com/doc/234632213/SISTEMA-EXPERTO-PARA-EL-DIAGNOSTICO-DE-FALLAS-DE-COMPUTADOR`

[4] Eclipse, 2017, Retrieved form `https://www.eclipse.org/`.

[5] A. A. Ergardo. Self learning computer troubleshooting expert system. 2016. Retrieved from `http://aircconline.com/ijaia/V7N1/7116ijaia05.pdf`.

[6] GoJava. Java powers our digital world, 2017, Retrieved form `https://go.java/index.html?intcmp=gojava-banner-java-com`.

[7] D. Kalbande. Knowledge based expert systems, 2016, Retrieved from http://scope.spit.ac.in/files/2017/07/Knowledge-Based-Systems.pdf.

[8] Microsoft. Windows support - windows 10 help, 2017, Retrieved from https://support.microsoft.com/en-us/products/windows?os=windows-10.

[9] R. M. Roberts. *Computer Service and Repair: A Guide to Upgrading, Configuring, Troubleshooting, and Networking Personal Computers*. The Goodheart-Willcox Company, 2005. Retrieved from https://www.g-wonlinetextbooks.com/computer-service-repair-2008/2.