Expert System for Car Failure Diagnostics

RONNY MORÁN, GUSTAVO LONDA, RENATO ILLESCAS

GROUP NUMBER #2

Introduction

After the increase in the fleet of cars, many workshops do not supply the attention they have to offer their customers and if they do, this is a much longer time than usual, either for a diagnosis and repair or just one Routine maintenance.

For this reason, the need to automate a process of mechanical failure diagnosis, in which the expert mechanic transfers his knowledge to a computerized system that simulates his knowledge, to be used by other branches or work equipment, that way can attack the problem directly and in a relatively short time in the diagnosis of the fault presented in the automotive.

Replicating the expert knowledge in the area of automotive mechanics led us to create the CarSE expert system, in which the mechanic or his work team after a questionnaire of questions to the client, selects in the expert system the presented failures and the system shows the diagnosis found in real time.

The requirement analysis and problem conceptualization

Our expert system is a desktop application with friendly user interface of easy handling and free access, users will be able to select different problems / faults of their cars, to obtain the diagnostic / recommendation of the car.

The expert system for the diagnosis of car failure called CARSE, analyzes the faults/diagnosis mentioned by the workshop and manuals consultanted and divided into these sections: engine, transmission, steering, suspension and brakes. From them we took the most common cases of failures/diagnostic and could build a knowledge base.

During the process of raising requirements and the knowledge base, the following people were consulted: Engineer Cristian Arias (professor of FIMCP), Engineer Sixto Alvarado (Supervisor of truck maintenance in Holcim), Tnlgo. Douglas Jiménez (Mechanical workshop).

The problem will be solved using the rules methodology (expert systems). he components of the expert systems are:

- **Knowledge Base:** Has the knowledge to formulate, understand and solve specific problems. It is composed of two basic elements: special heuristics and rules that govern the use of knowledge to solve specific problems of a particular domain.

- **Facts Base: (Working Memory):** Base where the facts are on problem that has been subject a long analysis period and grows as new facts are discovered in cycles of inference.

- **Inference Engine:** Is the control structure of an expert system, the Inference Engine is an automatic mechanism that allows to answer queries from the information that is in the Knowledge Base based on rules.
  The inference engine conclude with the knowledge of the problem and its solution determines how the rules are applied.

  Expert Systems Types
  They use two elements: facts and rules which intertwine to form the premises
  Facts: Known for particular situations, they are dynamic, ie they can be changed and stored in the working memory
  Rules: are general relations between set of objects do not change unless the expert system incorporates learning elements, is stored in the knowledge base

- **User interfaces:** Is interaction between the Expert System and the end user, and is done through natural language.

Analysis of the selected development tool

SHELLS for JAVA

- CLIPS (C Language Integrated Production System)
- JESS (Java Expert System Shell)
- PROLOG (Programming in Logic)
- DROOLS (Business Rules Management System)

JESS
Using JESS, we can build expert systems in Java as it has the ability to reason, using knowledge provided by the expert through declarative rules. Jess uses an improved version of the Rete algorithm to process the rules and is a very efficient mechanism to solve the difficult problem of the match between the network of rete in several to several relationships (all against all nodes)

Each JESS rules engine contains a knowledge base called facts, this collection is working memory and is important because the rules can only react to additions, deletions and changes in memory. You can not write a rule of Jess that reacts to something else.
Jess support algorithm Forward and backward chaining.

We choose Procedural representation that is represented as IF - THEN Rules.

Why Procedural representation?
The knowledge is represented as a set of instructions IF - THEN type of rules, this represent a very human friendly knowledge representation. Facts are connected through rules, and can have more than one condition.
Procedural representation (programs) express explicitly the interrelations between all fragments of knowledge but are difficult to modify, these systems have ease of use of Meta-Knowledge, which allows explicitly decompose the problem.

Why JESS?
In JESS the knowledge representation is write how IF - THEN rules (Easy, friendly representation) are in the rules.clp file in the project with the inference engine that "simulates" backward chaining and have ease of integration with Java Language (Because java is simple, object-oriented, cross-platform (Windows, Mac or Linux), there is a lot of documentation).
In Jess to use backward chaining you can use the "do-backward-chaining" function after the facts deftemplate is defined.

A prototype description

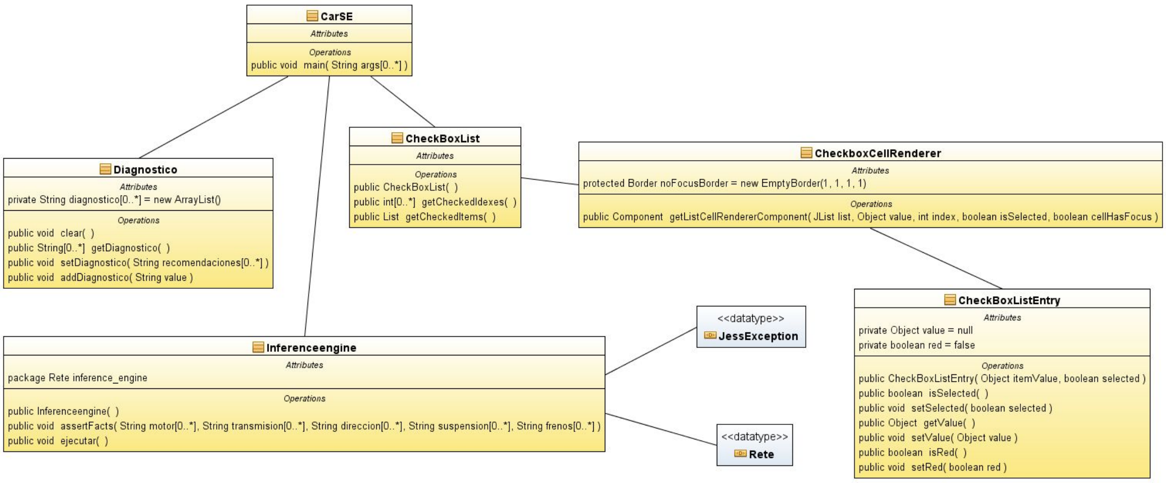
It's technically feasible to develop software using Procedural representation in JAVA environment along with JESS library (expert systems based on rules it has extensive documentation), knowledge base (we've experts in mechanics committed to contribute with knowledge) is wide and in fact is going to grow and this stored in a .clp file (based on CLIPS for easy access) so it will not allow before changes, also change the software code, that way it defragment the development components software for execution.

For the prototype the expert system as a desktop application, it used Netbeans IDE version 8.1 using the Java Programming Language, in which the Java expert system Shell library (JESS) was added where the class jess.Rete is the engine of Inference

The experts were identified by recommendation for the acquisition of knowledge, such as:
Engineer Cristian Arias (professor of FIMCP ESPOL), Engineer Sixto Alvarado (Supervisor of truck maintenance in HOLCIM), Tnlgo. Douglas Jiménez (Mechanical workshop).

The final product documentation using a formal language:

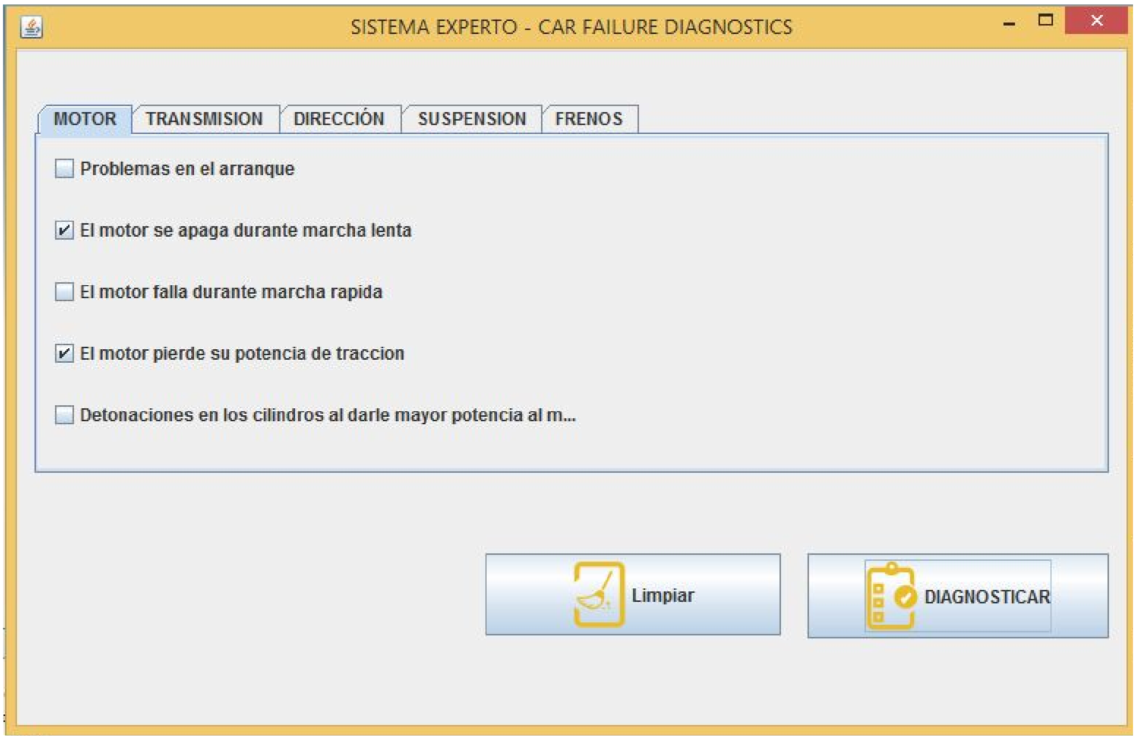
UML - CLASS DIAGRAM FOR EXPERT SYSTEM


CarSE
Attributes
Operations
public void main(String args[0..*])
Diagnostico
Attributes
private String diagnostico[0..*] = new ArrayList()
Operations
public void clear( )
public String[0..*] getDiagnostico( )
public void setDiagnostico(String recomendaciones[0..*])
public void addDiagnostico(String value)
CheckBoxList
Attributes
Operations
public CheckBoxList( )
public int[0..*] getCheckedIdexes( )
public List getCheckedItems( )
CheckboxCellRenderer
Attributes
protected Border noFocusBorder = new EmptyBorder(1, 1, 1, 1)
Operations
public Component getListCellRendererComponent(JList list, Object value, int index, boolean isSelected, boolean cellHasFocus)
Inferenceengine
Attributes
package Rete inference_engine
Operations
public Inferenceengine( )
public void assertFacts(String motor[0..*], String transmision[0..*], String direccion[0..*], String suspension[0..*], String frenos[0..*])
public void ejecutar( )
<<datatype>>
JessException
<<datatype>>
Rete
CheckBoxListEntry
Attributes
private Object value = null
private boolean red = false
Operations
public CheckBoxListEntry(Object itemValue, boolean selected)
public boolean isSelected( )
public void setSelected(boolean selected)
public Object getValue( )
public void setValue(Object value)
public boolean isRed( )
public void setRed(boolean red)

graphic 1

Test cases and the results:

The fault-capture screen accepts the user-selected faults (working memory) of the user for further evaluation and diagnosis.



□ Problemas en el arranque

☑ El motor se apaga durante marcha lenta

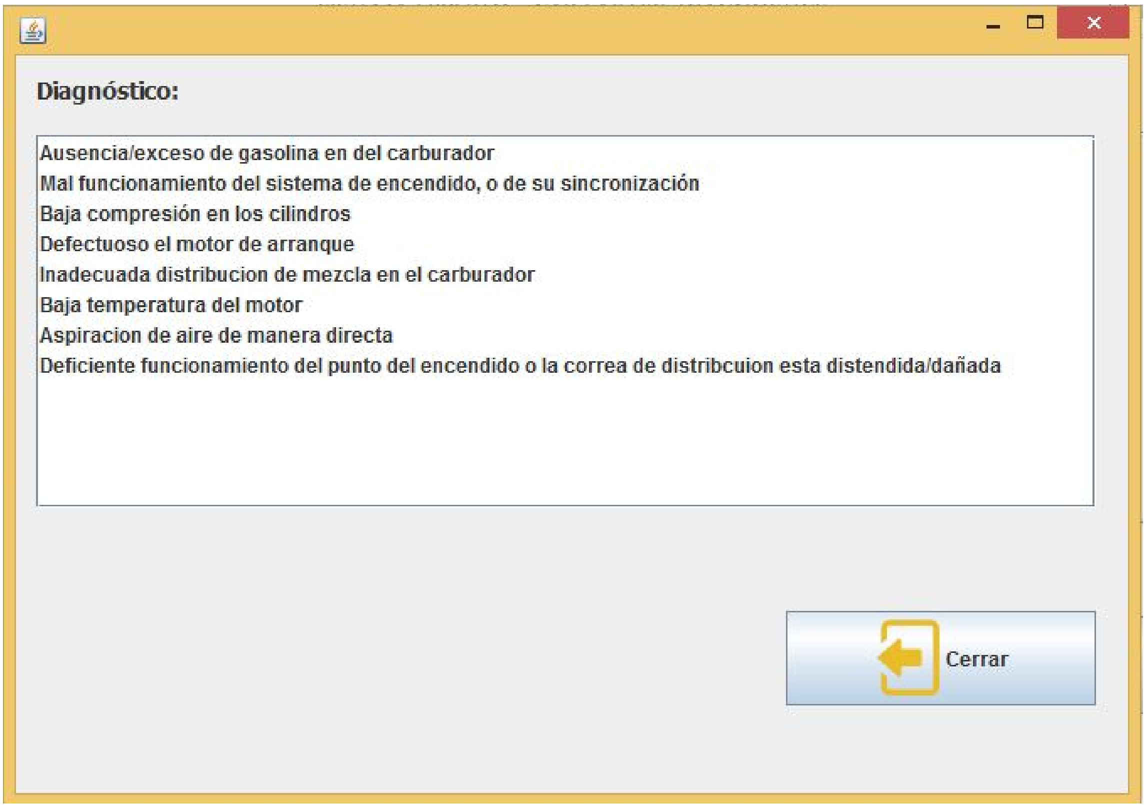
□ El motor falla durante marcha rapida

☑ El motor pierde su potencia de traccion

□ Detonaciones en los cilindros al darle mayor potencia al m...

graphic 2

Finally the Result window, shows the diagnosis found after the faults selected by the user:



Diagnóstico:

```
Ausencia/exceso de gasolina en del carburador
Mal funcionamiento del sistema de encendido, o de su sincronización
Baja compresión en los cilindros
Defectuoso el motor de arranque
Inadecuada distribucion de mezcla en el carburador
Baja temperatura del motor
Aspiracion de aire de manera directa
Deficiente funcionamiento del punto del encendido o la correa de distribcuion esta distendida/dañada
```

graphic 3

Conclusion and what have we learned:

We learned that the “Knowledge is power”.
The IF-THEN rules are a natural representation for expressing knowledge.
High computational cost because IF-THEN rules require pattern matching.
Maintenance in the knowledge base is complicated; Since when new knowledge (rules) is introduced to solve new problems we can include contradictions with the previously declared rules.
If we modify an existing rule, we must consider whether this modification affects all the rules associated with it.

References:

[1] "Expert System", 2017. [Online]. Available: http://www.emse.fr/~picard/cours/ai/chapter-jess.pdf. [Accessed: 07- Feb- 2017]
[2] "CLIPS User’s Guide", 2017. [Online]. Available: http://clipsrules.sourceforge.net/documentation/v630/ug.pdf. [Accessed: 07- Feb- 2017].
[3] "Jess, the Rule Engine for the Java Platform", *Jessrules.com*, 2017. [Online]. Available: http://www.jessrules.com/jess/. [Accessed: 07- Feb- 2017].
[4] "Prolog Programming Language", *Groups.engin.umd.umich.edu*, 2017. [Online]. Available: http://groups.engin.umd.umich.edu/CIS/course.des/cis400/prolog/prolog.html. [Accessed: 07- Feb- 2017].
[5] I. Bratko, *Prolog Programming for Artificial Intelligence*, 4th ed. Pearson Addison-Wesley, 2011, pp. 3-127.
[6] Haley, Paul. "Data-driven backward chaining." (1991).