



Deliverable 4. Prototype Description.

Design and implementation of a predictive module prototype.

Grupo de Sistemas Inteligentes

Departamento de Ingeniería de Sistemas Telemáticos

Universidad Politécnica de Madrid.

Project Report

Madrid, February 2013

Authors:

Adrián Pérez Orozco

Álvaro Carrera Barroso

Carlos A. Iglesias Fernández

Executive Summary

This document describes the design and implementation of a prototype predictive module for Thales' railway maintenance network. This implementation relies on already existing predictive rules which have been previously obtained from data mining procedures.

The prototype provides therefore a way to apply said obtained knowledge to actual situations, evaluating the rules and determining an output for each situation. It works as a rule-engine which takes the current situation as an input and outputs a list of predicted events along with an associated confidence for each of them. The system has been implemented in the form of a Java module, and can therefore be used as a standalone system or be integrated onto larger systems at convenience. It relies on the JBoss Drools Expert library, which provides an efficient and reliable rule-engine environment.

In this document, the architecture of the implemented prototype will be explained on detail. Furthermore, performance specifications are described as a result of several testing procedures.

Contents

Executive Summary	i
Contents	ii
List of Figures	iii
List of tables	iv
1 Introduction	1

List of Figures

List of Tables

1 Module description

In this section we will provide a general description of the implemented prototype. This prototype allows the usage of already existing association rules, which have been already obtained as the result of a *Data Mining* procedure. These rules do not offer any functionality by themselves, as a system is needed to check whether their conditions are fulfilled and therefore a prediction can be made.

Rules are simply textual information in the form of "When A and B happen together, C has 80% chances of happening". This information would be useful for an operator who might be manually checking events and would be able to expect C after seeing A and B. However, real systems usually have a much larger set of possible events, and many more events happening during each observation, and therefore an automated system is needed to perform these operations.

We call such a system a *Rule engine*[?]. A rule engine is simply a system which evaluates input conditions and fires the rules which comply with these conditions, outputting the result of said rules. In our case