

RULE ENGINE

First Review Report

Date:

Registration Number : 2011242019
Name of the Candidate : T. RINOLD SIMON
Name of the Company : UDProducts

1. Description of the problem :

This project is a study of the development of the Rule Engine, which is a validation system for quality assurance of product used. The system must include a validation page where the users of the system must be able to build rules and validate products by using the rules they have built, without making a new version of the system. The system must use AJAX to give the user an immediate feedback on the validation.

2. Background Study :

The main quality attribute focus is robustness and user-friendliness. They are best used in applications where the business logic is too dynamic to be managed at the source code level -- that is, where a change in a business policy needs to be immediately reflected. The goal was to execute the rules by making comparison, validation, decalring different formats to validate.

3. Objectives and Scope of the Project :

A rule engine is a tool for efficient decision making because it can make decisions based on thousands of facts quickly, reliably, and repeatedly. Unless like other Rule Engine's available in market, in which, each time when a new rule is made, someone have to code and make a new version of the system before the rule can be used. This is not efficient and

takes a lot of time. The main object of this project is to make a system where the rules are not hard coded into the code, but are inserted by an administrator/user. The rules must be saved so they are easy to access and edit.

4. Literature Survey :

The rule engine determines when to evaluate each rule based on the input required for the rule as well as the results obtained from the evaluation of previous rules. You do not need to specify the order or the dependencies of the rules. Create multiple rules to cover each situation, rather than creating one complex rule with many possible matching scenarios. Rules are typically structured as a well-defined pair of condition and action statements.

5. Methodology :

i. Assumptions, Constraints and Limitations :

Has the system achieved good robustness and user-friendliness? They could not save a rule that is not valid. When making a rule they got no feedback when it was saved. The system has to cover more so the administrator/user can do more, like removing and editing rules. In general, it is a best practice to write rule conditions as a set in which all conditions must be met for the rule to fire.

ii. Proposed Methods/algorithm used :

TheSolution is Pattern matching algorithm for implementing rule systems. Rete_algorithm replaces if ... Then. Builds a network of nodes from the rules, each node corresponds to a condition of a rule Path from root to leaf defines a complete rule. When a fact or combination facts match all patterns for a rule a leaf node is reached and the rule's action is fired

iii. Platforms/Development Tools :

Language : Ruby

Framework :

Backend : Rails for serverside scripting

Ember js for client side scripting(AJAX)

Frontend : Foundation

Database software platform : MongoDB

Signature of Student

**Signature of Mentor
from the Organization**

**Signature of
Internal Guide**