Rule based System for Product Lifecycle Management

Anuradha V. Yenkikar
Pune institute of Computer Technology
Pune University, Pune

Sachin S. Pande
Pune institute of Computer Technology
Pune University, Pune

ABSTRACT

Product lifecycle management helps to plan and manage the process of product development and create the best possible products. Innovations in new product introduction are need in most organizations today. Every product development process has different phases. Every role in the new product development process has expert knowledge in respective phase. Software optimizer can be developed based on the knowledge base of the project. This paper presents modeling and development of rule based expert system prototype for product lifecycle management with special reference to the new product development. It often called as knowledge based system. Proposed system contains a knowledge base of accumulated. Knowledge is in the form of rules. It evaluates decisions based on rules established.

Keywords

Product Lifecycle Management, Rule Based System, Expert System, New Product Development.

1. INTRODUCTION

In artificial intelligence, an expert system (ES) is a program that bears decision-making ability of human expert and solves problems like an expert. Removing troubles in this way helps save the time spent through the manual in search for an answer. For enhancing work efficiency and system stability, it is quite valuable. It also helps to reduce of missing data, better collection of data and no data replication etc [1]. Literature survey reveals that many expert systems reported in various fields [3] [4] [6] [7].

Typically, an expert system has a knowledge base of accumulated experience in the form of a set of rules. This system evaluates decisions based on rules established. Such system called as rule based system (RBS) or knowledge-based system. This paper extends application of rule based expert system technology for product lifecycle management called as 'PLM-RBS'.

2. IDENTIFICATION OF PRODUCT LIFECYCLE MANAGEMENT PROBLEM

It is necessary to understand about PLM domain for identification of their problems. PLM is to manage a product's definition from concept to retirement. The product lifecycle goes through multiple phases, involves many professional disciplines, and requires many skills and processes. It improves product's quality, reduces prototyping cost, reduces time to market and reuses original data. Their processes usually automated with PLM systems include management, sales and marketing, quality assurance, manufacturing and customer service support [12]. In industry, it is considered that new product development (NPD) is first and important stage in generating and commercializing new product with process of PLM. It is process of bringing new product into market. Their phases can iterate as needed. These phases briefly described as below.

Stage 1: Idea generation: New products ideas have to come from some sources include market research, employees, consultants, customers etc.

Stage 2: Requirement gathering and analysis: Requirements needed to implement idea as per demand gathered and check requirements are feasible and workable to develop.

Stage 3: Concept testing: The idea taken to the target audience is not a working prototype at this stage. It is just a concept.

Stage 4: Product designing: Physically design and manufacture the product, estimate based upon competition and customer feedback.

Stage 5: Product development: At this stage, the product prototype developed. The prototype tested through all the desired expectations and presented to the target audience for reviews about the product to see if changes need to be made.

Stage 6: Commercialization: If the marketing stage has been successful for prototype, the product will go for deeper launch. There are certain factors that need to take into account before a product launched. These include how the product launched and where the product launched.

Product development process problems often related to bad linkages across functional structures, unclear business strategy, inadequate marketing information, unclear for decision making, incorrect assumptions, not integrated data for decision making, poor management of evolving expectations, miscalculated costs and knee-jerk problem solving etc [2].

3. RULE BASED SYSTEM MAJOR COMPONENTS

The main components of proposed rule based expert system are shown in Fig. 1. Three major components are knowledge base (KB), inference engine (IE) and user interface (UI).

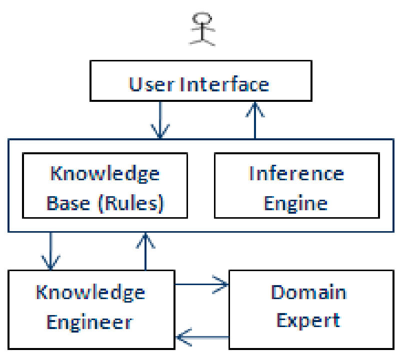

Fig. 1: Components of Rule Based System for PLM

There is interaction between the RBS and different individuals. The major roles of these individuals are domain expert, knowledge engineer and user as shown in Fig. 2. Domain expert provides with his accumulated knowledge and experience to put it in KB as shown in Fig. 1. Knowledge engineer plays the role of system designer who built the user interface, design the declarative format of the KB and encodes knowledge in the form of rules and implements the IE. User is the individual who will be consulting with the system to get solution to the problem.

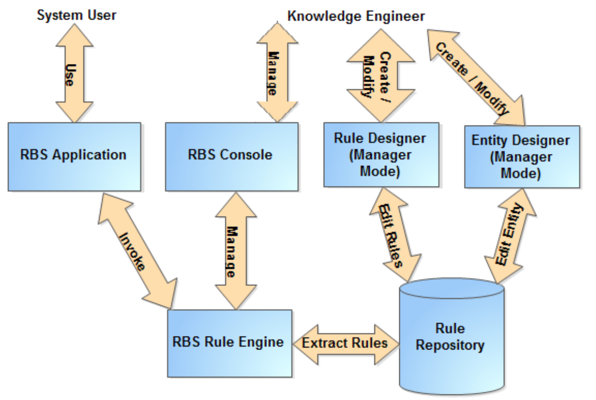

Fig. 2: Roles and their responsibilities in Rule Based System for PLM

3.1 Knowledge Base

KB is a representation of different stages of PLM domain knowledge, encoded in business rules, normally provided by human expert from their years of experience and used to solve real world problems. Knowledge base, also called as rule base. A business rule defines constraints the business. It can be a policy which is a definition relating to criteria organizations do with the product, customers or employees. A rule is a statement that applies in decision-making or computation to produce new information. All business rules expressed by using IF-THEN format.

3.1.1 Design of KB

Rule from KB has condition and action part. Condition part of rule divided to two sections: entity and value for entity. The entity represents the attribute which are combined as rules for the RBS to infer. Rule syntax is given as follows.

Rule #: If {(<entity_name1> operator <entity_value1>) or (< entity_name2> operator < entity_value2>)}

then {< entity_name3> = < entity_default_value3>}

3.1.2 Sample Business rules from PLM Domain:

rule #1: If {requirement is related to reliability, cost} then {Requirement category = "Marketing requirement"}
rule #2: If {BOM Structure is being released} then **{**Cost, weight and positional attributes should be populated, Trigger rule #3, Estimated cost < x% of the target cost, Put hold on the assembly release, Send notification to stakeholders}
rule #3: if {requirement state = "Released"} then { the requirement should have at least one related item for fulfillment, the requirement should have an associated rating (priority)}
rule #4: if {(any part is being released) and (the part has one or more associated issues)} then { Alert message ‘the part should be released only when all the associated issues are in <resolved> state’}
rule #5: if {requirements size > 1} then {requirements category =”Safety”}
rule #6: If{((system estimated cost – system target cost) * 100 / system target cost) > 25} then {send notification to finance team}

Table 1: Sample business rules from PLM domain.

Rules in KB represented by using RuleML. RuleML is a markup language for sharing rule bases on the World Wide Web. It is spanning across all industrially relevant web rules and it is about rule interoperation between industry standards such as JSR-94 and OCL. Therefore, in proposed system PLM-RBS, RuleML preferred to document complex business rules as well [10] [11]. One Sample business rule from PLM represented in RuleML format given as follows.

Rules file in ruleml

```
<Implies>
  <If >
   <And>
    <Atom>
      <Rel>=</Rel>
      <Var>REQUIREMENT</Var>
      <Ind>RELIABILITY</Ind>
    </Atom>
    <Atom>
      <Rel>=</Rel>
      <Var>REQUIREMENT</Var>
      <Ind>COST</Ind>
    </Atom>
  </And>
</If>
<Then>
  <And>
    <Atom>
      <Rel>=</Rel>
      <Var>CATEGORY</Var>
      <Ind>MARKETING</Ind>
    </Atom>
  </Then>
</Implies>
```

XML facilitates interoperability between different PLM applications and products means it uses for data exchange as standard format. XML tags can fully describe principal attributes of physical parts, documents, and engineering changes. Currently, XML preferred for interchange of data over web. Many Rules from same category can form one rule set [9]. Following XML file represents business rule from PLM domain.

Rules file syntax in xml

```
<Rules>
  <Rule Id="RULE_1" Name="CATEGORISE_REQT">
    <Conditions>
      <Condition>
        <EntityName>REQUIREMENT</EntityName>
        <Operator>=</Operator>
        <Value>RELIBILITY</Value>
      </Condition>
      <Join>AND</Join>
      <Condition>
        <EntityName>REQUIREMENT</EntityName>
        <Operator>=</Operator>
        <Value>COST</Value>
     </Condition>
   <Conditions>
   <Action>
      <Assignment>
         <EntityName>CATEGORY</EntityName>
         <Operator>EQUAL</Operator>
         <Value>MARKETING</Value>
      </Assignment>
        <Trigger>
           <Id>6</Id>
        </Trigger>
        <Formula>
           <Formula>a=c</Formula>
        </Formula>
        <Notify>
           <String>Alert</String>
           <To>Stakeholder</To>
        </Notify>
      </Action>
    </Rule>
</Rules>
```

These kinds of XML rule sets are export by different PLM tools and applications. Rule set in XML converted into rule base of RuleML by using XSLT technique. Inference process will be from rule set in RuleML. Newly created rule or imported ruleset will get updated in rule set of XML as well as simultaneously in RuleML. This will achieve both advantages of XML and RuleML for proposed system [9].

As shown in Fig. 4, rules can be designed through knowledge acquisition process by knowledge engineer. When rules are going to be stored in rule set, it passes through constraint and validation framework which checks for whether rule exists.

3.2 Inference Engine

Inference engine also called as Rule Engine. It analyzes and processes rules to give conclusion as shown in Fig. 3. Proposed system's IE follows forward chaining reasoning method because it extracts data from user until goal reached. Data determines rule selections therefore it often called as data driven. Whenever there will be change in rules, object from working memory will get updated. Explanation component provided for system user is essential to understand how system works. It is used to choose selections between matched rules for expected outcome. Selected rule's object updated into working memory. Depending on selections made by user, action will be fired [5].

A fact is a statement that connects terms through prepositions and verb phrases into sensible business-relevant observations. Facts are real world situations and it can be true or false. It is input to working memory. It has entity name assigned to its default value or new value.

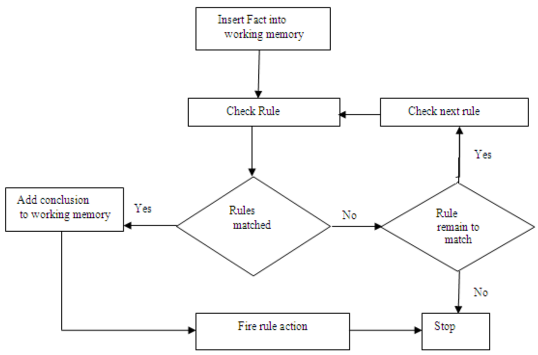

Fig. 3: Inference process of Rule Engine

Pattern matching component in rule engine matches entity name from fact with entity name used in rule's condition part.

3.3 User Interface

User Interface (UI) provided for system user to interact with PLM-RBS. Through UI, the facts inserted to get conclusion. Rules and entities created and updated through rule editor and entity editor respectively.

4. MODELING RBS-PLM

RBS is being developed to operate in dynamic environments where they must generate conclusions to take action. Models provided for the RBS-PLM mainly includes the knowledge-base and inference engine and the mechanism for interfacing to the user inputs. RBS's dynamic process is having 3 kinds of events. These events are given as follows.

1. Input event $e_u \in E_u$

Where, $E_u = \{e_{u1}, e_{u2}... e_{un}\}$: Set of input events and fact is a input.

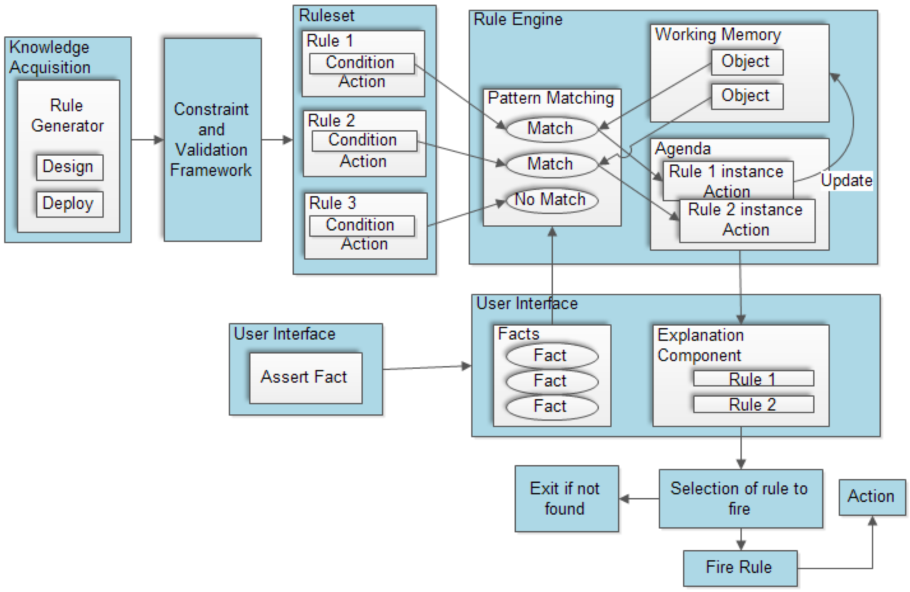

Fig 3: Rule Based Expert System for PLM architecture

2. Selection event $e_s \in E_s$

Where, $E_s = \{e_{s1}\ e_{s2}... e_{sn}\}$: set of selection events i.e. user's selection of rule from explanation component.

3. Output event $e_o \in E_o$

Where, $E_o = \{e_{o1}\ e_{o2}\ldots\ e_{on}\}$: set of process output events i.e. conclusion generated by PLM-RBS.

$E = E_u \cup E_s \cup E_o$, where, E is the set of all events.

$e_k = e_{uk} \cup e_{sk} \cup e_{ok}$ at event time k.

Complete PLM-RBS represented as follows:

$C = \{E, R, I_v\}$

Where, C= Rule Based System, E= set of all events, R= set of rules and I_v = set of all valid inference loop events.

Fact $f \in F$ where, $F = \{f_1, f_2... f_n\}$: set of facts

$F \rightarrow \{0, 1\}$: facts that are true or false

Rule $r \in R$ where, $R = \{r_1, r_2\ldots\ r_n\}$: set of rules

r = IF Φ THEN Ψ where, Φ = condition in rule, Ψ = action in rule, it makes 'fl' true

IF {condition Φ evaluates to true at time k for the given state e_{sk} and the input event e_{ik}}

Then {Action Ψ can be taken to get the output event e_{ok}}

$e_{ok} = \{r, e_{sk}\} \subset e_{ik}$

Inference loop $i_v \in I_v$ where, $I_v = \{i_{v1}, i_{v2}, i_{vn}\}$: set of valid inference loop events [4].

Select phase picks one rule from Γ_k to fire given as follows.

$\Gamma_k = \{r: \{r, e_{ok} = \{r, e_{sk}\} \subset e_{ik}\}\}$

Here, Action Ψ can be taken when the Φ of rule $r \in R$ evaluates to true for e_{sk},

5. IMPLEMENTATION

For proposed system, author can use existing inference engine shell which support forward chaining. Knowledge engineer uses chosen shell to develop knowledge base in respective domain. A program developed in Java for implementation of user interface, explanation component, working memory and inference engine shell. Rule engine of PLM-RBS is scalable. Using Java's synchronization capability, allows applications to share objects and rules. Rules pre-parsed when the rule engine created. It avoids ruleset parsing time when the engine evaluates business rules. The unique key features taken into consideration for implementation of proposed system are collections of expanded rules, expressiveness and the decidability of the resulting queries, ability to easily write rules in a human understandable format, validation on rules, extendibility to re-design the language to add new functionality, interoperability with a variety of proof engines and tools and re-use of knowledge

6. CONCLUSION AND FUTURE WORK

This paper presented modeling and development of rule based expert system for product lifecycle management. The proposed system is useful for product developers or NPD users. It helps to offer best solution for particular problem in the PLM domain and generates a conclusion that meets intelligence at the same time. Depending on generated conclusion, suitable action will be taken and notified to product developers and stakeholders. This expert system checked by industry expert as well as NPD users. These users have given positive initial feedback. The systems future scope can be widened by adding more selection criteria on matching rules to get exact solution.

7. REFERENCES

[1] Segev Wasserkrug, Avigdor Gal, Opher Etzion, and Yulia Turchin, "Efficient Processing of Uncertain Events in Rule-Based Systems", IEEE Trans. VOL. 24, NO. 1, JANUARY 2012

[2] Ioannis Hatzilygeroud, and Jim Prentzas, "Integrated Rule-Based Learning and Inference", IEEE Trans. VOL. 22, NO. 11, NOVEMBER 2010

[3] Christina G. Siontorou, Fragiskos A. Batzias, and Victoria Tsakiri, "A Knowledge-Based Approach to Online Fault Diagnosis of FET Biosensors", IEEE Trans. VOL. 59, NO. 9, SEPTEMBER 2010

[4] Jinyuan Li, Shuming Tang, Xiqin Wang, Wei Duan, and Fei-Yue Wang, "Growing Artificial Transportation Systems: A Rule-Based Iterative Design Process", IEEE Trans. VOL. 12, NO. 2, JUNE 2011

[5] Rainer Knauf, Avelino J. Gonzalez and Thomas Abel, "A Framework for Validation of Rule-Based Systems", IEEE Trans. VOL. 32, NO. 3, JUNE 2002

[6] M. S. Kandil, S. M. El-Debeiky, and N. E. Hasanien, "Long-Term Load Forecasting for Fast Developing Utility Using a Knowledge-Based Expert System", IEEE Trans. VOL. 17, NO. 2, MAY 2002

[7] Rita Cucchiara, Massimo Piccardi, and Paola Mello, "Image Analysis and Rule-Based Reasoning for a Traffic Monitoring System", IEEE Trans. Veh. Technol., VOL. 1, NO. 2, JUNE 2000

[8] Nick Bassiliades, Ioannis Vlahavas and Ahmed K. Elmagarmid, "E-DEVICE: An Extensible Active Knowledge Base System with Multiple Rule Type Support", IEEE Trans. VOL. 12, NO. 5, SEPTEMBER/OCTOBER 2000

[9] Xinwei Wang,"Mining Association Rules from Complex and Irregular XML Documents Using XSLT and Xquery", ALPIT Conf., Page(s): 314 - 319, July 2008

[10] Thanawut Auechaikul and Wiwat Vatanawood, "A Development of Business Rules with Decision Tables for Business Processes", IEEE Tranc.1-4244-1272-2/07, January 2007

[11] Wijesekera, D. , Michael, J.B., "Using RuleML to specify cross-domain information flow control policies ", IEEE Conference, JUNE 2009

[12] Valentina Gecevska, Paolo Chiabert, Zoran Anisic, Franco Lombardi, Franc Cus, "Product lifecycle management through innovative and competitive business environment", JIEM, OCTOBER 2010