2007-01-3865

Decision Support Systems for Launch and Range Operations using Jess

Rajkumar Thirumalainambi
PSGS @ NASA Ames Research Center

ABSTRACT

The virtual test bed for launch and range operations developed at NASA Ames Research Center consists of various independent expert systems advising on weather effects, toxic gas dispersions and human health risk assessment during space-flight operations. An individual dedicated server supports each expert system and the master system gather information from the dedicated servers to support the launch decision-making process. Since the test bed is based on the web system, reducing network traffic and optimizing the knowledge base is critical to its success of real-time or near real-time operations. Jess, a fast rule engine and powerful scripting environment developed at Sandia National Laboratory has been adopted to build the expert systems providing robustness and scalability. Jess also supports XML representation of knowledge base with forward and backward chaining inference mechanism. Facts added - to working memory during run-time operations facilitates analyses of multiple scenarios. Knowledge base can be distributed with one inference engine performing the inference process. This paper discusses details of the knowledge base and inference engine using Jess for a launch and range virtual test bed.

INTRODUCTION

As NASA plans to launch humans for a Mars Mission with an advanced Crew Exploration Vehicle based on nuclear propulsion technology, the launch operations are rapidly becoming very complex in control and communications. To manage this complex launch operations, increasing number of experts are required for precise launch operations. The experts in various domains are distributed over the continental US and it is less likely that everyone can be present near the launch pad or operational centers. One feasible solution for distributed expert systems is the World Wide Web (WWW). NASA has developed over many years a set of contingency rules for launch commit criteria and expert knowledge to apply these rules at distributed locations. A distributed expert system for launch operations can operate concurrently to derive ultimate "Go/No-Go" decision within the scenario of a launch or landing of a space vehicle [3-11]. Here, NASA has implemented three different expert systems, which derive decisions based on launch contingency rules, the weather expert system, the toxic-gas exposure expert systems, and the human-health risk assessment expert system. A dedicated web server with data management system and physical models supports each expert system. Figure 1 shows some of the visualizations provided in the ILRO Weather Expert System.

Figure 1. ILRO Weather Expert System

Each expert system consists of an inference engine based on backward chaining process, a knowledge base on specific domains (for example, weather, toxic gas dispersion and human health risk assessment contingency rules for launch commit criteria) and a web based graphical user interface for the user interaction. Each expert system also has specific types of rules derived from the contingency rules. Interoperability among expert systems based on Internet protocols is very limited due to its natural independence. As an example, a rule is shown below which has been presently used in one of the expert systems [4].

Rule: Go_HCL_7000_15000
IF chemical = HCL AND launchtype = NORMAL AND concentration < 16 PPM AND 6999 M < range < 15001 M THEN launch = GO

The advantage of the above rule is human and machine readable, but it does not follow any present standards like XML. One of the major disadvantages of individual set of rules implemented on individual systems is the difficulty to certify the validation of the rules and the formulation of a central knowledge base. Jess the expert system developed by Sandia National Laboratory provides a comprehensive solution to adopt a centralized knowledge base and efficient validation of rules [2]. The following section provides details of Jess and its scalability and adoption to launch and range operations.

JESS

Jess is a rule engine and scripting language developed by Sandia National Laboratory in 1990. Jess is dynamic and java centric which provides access to java API. Jess is provided as a library, so that it can be embedded in other applications. Java beans play an important role in jess. A java bean property is most often a pair of methods named in a standard way. If the property is a string named label, the java methods look like this String getlabel(); and setLabel(). The get method is used to read the value of the property and set method to change it. The java.beans API includes a class named Introspector that can examine java beans and find properties defined according to this get/set naming system. Java bean should implement serializable to inform property of the state to the working memory. Jess working memory can contain ordered, unordered and shadow facts. Unordered facts are general-purpose facts, where as shadow facts are used to connect java bean that can interact with working memory in run time. Jess supports forward chaining and backward chaining rules and the rules have antecedents and consequents. Antecedents have to be satisfied to fire a consequent based on pattern matching techniques. The rules can be organized into modules using *defmodule*, which can act similar to templates. Jess provides user-function interfaces to implement custom functionalities in addition to the available functions. The user package can hold multiple user functions for easy access.

JESS RULES AND XML

Jess rules can be transformed into XML using XSLT translators. XML editors can be used to edit and view as a tree. Accessing the jess rules in XML format using standard parsers makes a scalable and robust knowledgebase. The javax.rules package designed by JSR 94 provides various API to access the standard rule engine along with representing rules in XML format. The above rule in section 1 can be rewritten into this format.

```
<?xml version="1.0" encoding="UTF-8"?>
<rule-execution-set>
<name> Go_HCL_7000_15000</name>
<code>
(defclass java.lang.String java.lang.Integer)
(defrule Go_HCL_7000_15000
(java.lang.String (OBJECT ?chemical))
(java.lang.String (OBJECT ?launchtypel))
(java.lang.Integer (OBJECT ?concentration))
(java.lang.Integer (OBJECT ?range))
(test (lt 16(?concentration intValue)))
=&gt;
(printout t "Launch GO"crlf))
</code>
</rule-execution-set>
```

In the above rule, all the antecedent conditions are not checked, instead only the concentration is being validated. Javax.rules provides administrative and run time interactions with suitable API to access rules, runtime tasks into manipulating working memory and executing rules.

DECISION SUPPORT SYSTEMS FOR LAUNCH AND RANGE OPERATIONS

The extreme complexity of the launch and range operations require cooperation among the multiple advisory expert systems. The expert systems are supported by real-time and model data to make an optimized decision-making during launch operations. The uncertainty involved in decision-making process is a reflection of the physical models rather than expert system by itself. To minimize uncertainty and derive best-optimized decisions is always a challenge in a short time frame. The U.S.A. Air force authority depending on the criticality and implications of the decision can waive conflicting decisions of launch operations.

The Jess system provides an efficient way of managing multiple knowledge bases into a single knowledge base by defmodule. The defmodule can be used to categorize weather, toxic gas and human health risk assessment rules providing easy interoperability among these rules. Jess rules can extend any number of slots which makes launch and range operations rules sets to be minimal due to rule functionality. Jess does not support dynamic changes of rules during runtime. In launch and range operations, the contingency rules are well established for a given space vehicle. Jess depends on Rete's algorithm with a class named "Rete", which is the main class [1]. It can be instantiated any number of times in an application. The states of engine can be invoked at run time as well. The bench mark of jess against various commercial rule engines are far superior and can handle voluminous rule sets in the shortest time frame. The richness of the Jess API makes the system suitable for adoption in distributed environments as well.

CONCLUSION

Initially, ILRO test bed has sets of individual expert systems with rules expressed in natural language. Jess and its standardized rule format expression provide to the test bed an integrated knowledge base with automatic validation mechanisms. Jess allows inheriting most of the Java software of the test bed; however, it is not an object-oriented system.

REFERENCES

1. R. Forgy C. L. A fast algorithm for the many pattern/many object pattern match. *Artificial Intelligence*, 19(1982), 1991.
2. Jess, the rule engine for java platform. http://herzberg.ca.sandia.gov/jess/docs/70, version7.0a5, Draft February 2nd 2005.
3. Jorge Bardina and T. Rajkumar (2003) Intelligent Launch and Range virtual test bed (ILR-VTB), *Aerosense 2003*, Orlando, Florida, USA
4. T. Rajkumar and Jorge Bardina (2003) Web-based Weather Expert System for Space Shuttle Launch, *IEEE SMC 2003*, Washington DC, USA
5. T. Rajkumar and Jorge Bardina (2004) Human Health Risk Assessment Simulations in a Distributed Environment for Shuttle Launch, Defense *and Security Symposium 2004*, Orlando, Florida, USA
6. Jorge Bardina and T. Rajkumar (2004) Web-based virtual test bed for Shuttle Launch Operations, *Defense and Security Symposium 2004*, Orlando, Florida, USA
7. T. Rajkumar and Jorge Bardina (2004) Debris dispersion modeling, *ESM 2004, 18th European Simulation Multiconference,* Germany
8. Jorge Bardina and T. Rajkumar (2004) Modeling and simulation of Shuttle launch and range operations, *ESM 2004, 18th European Simulation Multiconference,* Germany.
9. T. Rajkumar and Jorge Bardina (2005) Non-intrusive Techniques of Inspections during Pre-launch Phase of Space Vehicle, *Defense Security Symposium 2005*, USA.
10. Jorge Bardina and T. Rajkumar (2005) Business Intelligence Modeling in Launch Operations, *Defense Security Symposium 2005*, USAT. Rajkumar and Jorge Bardina (2005) Decision Facilitator for Launch Operations using Intelligent Agents, *SAE, WAC 2005*, Dallas, USA

CONTACT

Rajkumar Thirumalainambi, Ph.D. is a Computer Scientist at Perot Systems Government Services, Inc (PSGS), working for NASA Ames Research Center, Moffett Field, California. His research interest areas are neural networks, fuzzy logic, genetic algorithms, modeling and simulation. He has participated in multiple NASA projects including the development and software of a virtual test bed for real time simulation of space vehicles using artificial intelligence techniques.

His contact address is:
NASA Ames Research Center
Mail Stop 269-2
Moffett Field, CA 94035
E-mail: rajkumar@mail.arc.nasa.gov