UDC 331.108.3+00.9=111

EXPERT SYSTEMS DEVELOPMENT APPROACH IN MODERN INDUSTRIAL APPLICATIONS

VITAL DRAZDOU, ARKADZI OSKIN
Polotsk State University, Belarus

The article represents the overview of expert system approach in modern industrial applications. Shows base advantages and disadvantages of using this approach, performance comparison. The inference contains issues, which need to solutions.

As networking and mobile technologies develop, modern software solves a wide range of tasks.

As a result, software products include components realizing its complicated business logic. Software specifications are constantly changing, importance of new solutions and techniques of new, more effective products design appear.

Methods of qualitative software writing have been known for 30 years already. In spite of considerable achievements, the software field does not use its whole potential for different reasons [2].

The issues of implementation application logic with the approach used in expert systems on the stack of Java virtual machine technologies are analysed in the article.

Java-code writing is the most widespread way of business logic components, i.e. J2EE and J2SE, design. The drawback of the code is its complexity and entanglement, which cause a problem to experienced developers as well. Moreover, any changes of logic, even the simplest ones, require recompilation and unwinding of the application.

Nowadays, there are a few software processors with commercial and open source codes. Commercial processors allow define the regulations in special languages similar to English. In other cases, the regulations are defined by means of scripting languages, such as Groovy or Python [3].

Drools is a rules processor with an open source code, Java written and implementing the rules according to the Rete algorithm. Business-rules application can be characterized in declarative form due to Drools, using a language simple for learning and understanding and not connected with XML. Patterns of Java code can be implemented in file rules, thus Drools becomes more convenient and favourable for use [4, 5].

Among Drools advantage the following can be named.

– Public support;
– Easy utilization;
– High speed of rules obeying;
– Correspondence to API specifications for Java (JavaRuleEngine API – JSR 94);

The main purpose of Drools rules use is processes description and solution making. The example of a discount making and potential clients searching is analysed in two different variant: with and without of Drools rules.

Rules of the discount calculation:

– Payment in cash does not mean the discount;
– Debit card payment means a 5 per cent discount of the total amount paid;
– Credit card payment means a 10 per cent discount of the total amount paid.

These are the rules of potential clients searching:

– Possibility for a potential client to get a loan of 80 per cent of the purchase total cost;
– To send the loan proposal by e-mail if it does not exceed 500 units;
– To call a potential client if the loan exceeds 500 units.

The application is represented as web service to make test process more convenient.

Four rules are used to characterize the implementation of the Drools rules. Each of the rules defines a discount detection and search for a potential client.

Following applications were developed to realize an abovenamed logic:

– Rule for finding loan amount to the debit card owners;
– Rule for finding loan amount to the credit card owners;
– Rule of the e-mail to the client with a loan proposal;
– Rule for a loan proposal call to the client.

The example of using the discount rule is shown in Fig. 1.

```
Package com.my.rules;
import          com.my.domain.Client
rule            "Creditcarddiscount"
when $client : Client(paymentType.equals("CREDIT_CARD") )
then            Floatdiscount = $client.getSpentedSum() * 0.1f;
                $client.setSpentedSum($client.getSpentedSum() - discount);
end
```

Fig. 1. Example of drools rule for the client's discount

From example we can see that the rule works only in case of the debit card payment, and a new sum is credited adjusted for a 10 per cent discount.

Behavioral pattern strategy was used in classical realization for discounts and potential clients search. This approach makes the process ultimately flexible.In the Fig.2. we can see the architecture of the classical implementation.

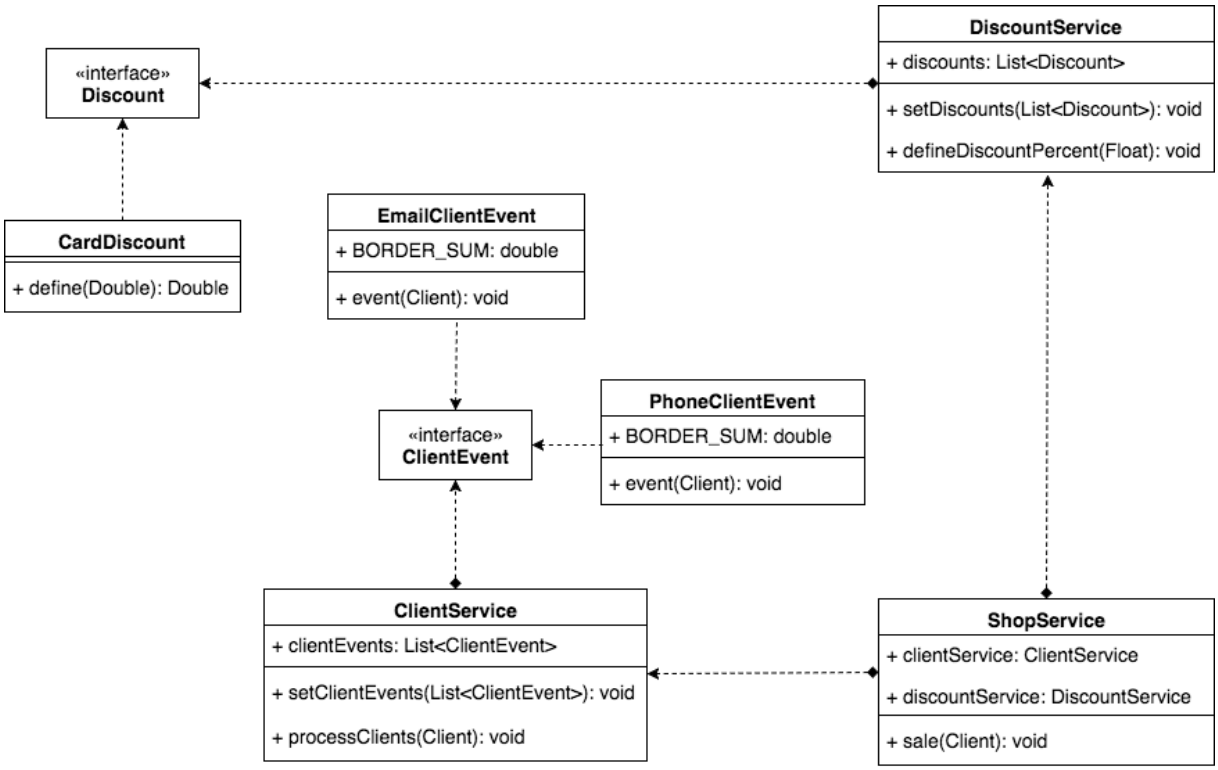

Fig. 2. Class diagram for the application.

The aspects of the program work, in the context of the research, are analyzed further.

Efficiency and SFT are key specification at the stage of its running.

System performance is a key criterion of the running and usually it crucial for the choice of this or that technique [2]. Due to Rete Algorithm Drools rules have a rather effective performance, thus becoming competitive in the market [6].

Rete algorithm provides it with higher efficiency. Using Rete, the expert system draws a special graph or a prefix tree, which nods are parts of the rules. The way from the root to the leaf makes the whole condition to certain products. Each nod contains a list of factors meeting the rule. Adding or changing of a factor causes its running over the net; the nods, containing corresponding rules, are marked. The rule is met, when rule's conditions are satisfied, and the system reaches the graph leaf [7].

The application imitates a web site. So two web sites, having equal input characteristics and result, but different in the mechanism of a business component operation, are tested. To test application is used Apache JMeter tool [8].

Characteristics tested:

– REQs: 10000.
– Flows referring to the service simultaneously: 100.

The tables below provide the test result.

Aggregated results of tests for the Drools implementation

Total REQs	Average time (ms)	Median (ms)	REQ maximum time (ms)	Errors (%)
10000	31	27	880	0

Aggregated results of tests for the classical implementation

Total REQs	Average time (ms)	Median (ms)	REQ maximum time (ms)	Errors (%)
10000	12	8	885	0

The data above prove a greater effectiveness of Java in implementation, which, consecutively, limits a rules approach in the system, where high efficiency is required. It is important to mention high SFT in both cases.

Thus, the rules approach is suitable in cases not requiring an immediate reply and delays are unimportant. On the other hand, systems, based on the rule processor, are complicated and need an expert for a knowledge base update.

Having analysed the data, the ways for further study can be pointed: alternative approaches to decision making and business logic description, optimization of the actual approach to the rules and the search for algorithms of its automatic training.

REFERENCES

1. Аппаратные и программные средства информационных технологий и систем // Информационные технологии в бизнесе / под ред. Милана Желена. – СПб, 2002.
2. Steve McConnell. Professional Software Development // Publisher Addison-Wesley, 2007.
3. Flow-managed persistence in Spring Web Flow 2 [Electronic resource] // IBM developerworks journal, 2010. – Mode of access: http://www.ibm.com/developerworks. – Date of access: 31.10.2016.
4. Lucas Amador. Drools Developer's Cookbook // Packt Publishing, 2012.
5. Use Drools and JPA for continuous, real-time data profiling [Electronic resource] // IBM developerworks journal, 2012. – Mode of http://www.ibm.com/developerworks. – Date of access: 20.10.2016.
6. World's fastest rules engine [Electronic resource] // Infoworld, 2010. – Mode of access: http://www.infoworld.com. – Date of access: 20.10.2016.
7. Robert, B. Doorenbos Production Matching for Large Learning Systems / B. Robert ; Computer Science Department. Carnegie Mellon University. – Pittsburgh, PA, 1995.
8. Sandeep Bhatti, Raj Kumari. Comparative Study of Load Testing Tools / Sandeep Bhatti, Raj Kumari // International Journal of Innovative Research in Computerand Communication Engineering. – 2015.