

Complicated Algorithm - How to store rules separate from processing code?

Asked 15 years, 9 months ago Modified 10 years, 5 months ago

Viewed 384 times



0



I'm working on a project which will do some complicated analyzing on some user-supplied input. There will be 3 parts of the code:

1) Input supplied by user, such as keywords

2) Rules, such as if keyword 1 is repeated 3 times in keyword 5, do this, etc.

3) And the analyzing itself which executes the rules and processes the user input, and generates the output necessary based on the processing.

Naturally this will lead to a lot of spaghetti code and many, many if statements in the processing code. I want to avoid that, and keep the rules (i.e. the if statements) separately from the code which loops through the user input and generates the output.

How can I do that, i.e. what is the best way?

algorithm

Share

Improve this question

Follow

edited Jul 21, 2009 at 18:40



Rich Seller

84k ● 23 ● 175 ● 179

asked Mar 27, 2009 at 9:38



Ali

267k ● 267 ● 589 ● 782

7 Answers

Sorted by:

Highest score (default)



2

If you have enough rules that you want to externalize, you could try using a business rules engines, like [Drools](#) in Java.



A business rules engine is a software system that executes one or more business rules in a runtime production environment. The rules might come from legal regulation ("An employee can be fired for any reason or no reason but not for an illegal reason"), company policy ("All customers that spend more than \$100 at one time will receive a 10% discount"), or other sources. ([Wikipedia](#))

It could be a little bit overhead depending of what you're trying to do. In my company we're using such kind of tools

for our quality analysis tool.

Share Improve this answer
Follow

answered Mar 27, 2009 at 9:45



[Julien Hoarau](#)

50k ● 20 ● 132 ● 117



1



Store it in XML. Easy to parse and update.

I had designed a code generator, which can be controllable from a xml file. For each command I had a entry in the xml. I was processing the node to generate the opcode for that command. Node itself contains the actions I need to do for getting the opcode. For some commands I had to look into database, all those things I had put in this xml file.

Share Improve this answer
Follow

edited Mar 27, 2009 at 10:15

answered Mar 27, 2009 at 9:42



[Vinay](#)

4,773 ● 7 ● 35 ● 43



1



Well, i doubt that it is necessary to have hughe if statements if polymorphism is applied correctly.

Actually, you need a proper domain model for your rules. This goes somehow into the direction of the command pattern, depending on the complexitiy of your code maybe in combination with the state machine pattern.



Once you have your model, defining rules is instantiate them correctly.

This could be done by having an xml definition, which is parsed and transformed into your model. But the new modern and even more fancy way would be using DSLs. If you program in Java and have a certain freedom about your libraries, this would be a proper use case for [Embedded DSLs with Groovy](#). Basically you would need a Builder which constructs your model, that's all.

Share Improve this answer

answered Mar 27, 2009 at 10:43

Follow



Lars

509 ● 5 ● 12



0

You always can implement factory that will create certain strategies according to passed parameters. And then you will use those strategies in your code without any if.



Share Improve this answer

answered Mar 27, 2009 at 9:42

Follow



Mykola Golubyev

59.7k ● 15 ● 93 ● 102





0

If it's just detecting keywords, a finite state machine or similar. If it's doing more, then other pattern matching systems, such as rules engines.



Share Improve this answer

answered Mar 27, 2009 at 9:49

Follow



Pete Kirkham

49.3k ● 5 ● 94 ● 173



0

Adding an **embedded scripting language** to your application might help. The rules would then be expressed in scripts, executed by the applications on processing.



The idea is that scripts are easy to change and contain high level logic that will be executed by your application in details.



There are a lot of scripting languages available to do this : lua, Python, Falcon, squirrel, angelscript, etc.

Share Improve this answer

edited Mar 27, 2009 at 9:53

Follow

answered Mar 27, 2009 at 9:42



Klaim

69.5k ● 36 ● 135 ● 190

Have a look at [rule engines](#)!



The approach from Lars may also be arguable.

0

Share Improve this answer

answered Mar 27, 2009 at 10:47

Follow



[Martin K.](#)

4,701 ● 7 ● 36 ● 49

