

Decision Logic Tables as a DSL



[Jeffrey Cameron](#) 4 posts since

Apr 21, 2009

I just came across MPS from a Martin Fowler blog entry and thought to myself ... what an intriguing idea ... I wonder if I can apply it to a problem I have.

At my work I have to process Decision Logic Tables. These are currently written in a specific form with lines like so

% parameter : value

\$ commonAction

@ condition ;Y;

& action ;X;

where commonActions, conditions and actions are all mathematical expressions. For example a condition could be: $VAR1 > 12$

I wonder if there is any benefit in pursuing implementing these with MPS as a DSL or should I stick with my current method of parsing these things out with CoCo/R? Having little experience with MPS I just want an expert opinion before embarking on a learning project only to discover my problem isn't suited to the domain.

Thanks



[Konstantin Solomatov](#) 313 posts since

May 18, 2004 1. **Re: Decision Logic Tables as a DSL** Apr 22, 2009 12:06 PM

Hello Jeffrey,

By using MPS you will be able to spend less time on infrastructure code: grammar, editor, typesystem. You will also have a decent IDE for your code. If you implement the same stuff with the mainstream technologies, you will spend much more time for such infrastructure. If you use MPS, you will be able to spend this time on your domain.

BTW, we experimented with logical stuff in MPS. We've implemented PROLOG with interpreter in it and even tried to extend it with domain specific constructs. Unfortunately, the language isn't included in the MPS build

Regards,

Konstantin



[Tom Brus](#) 80 posts since

Jun 2, 2003 [2](#). **Re: Decision Logic Tables as a DSL** Apr 22, 2009 10:27 PM

Hi Konstantin,

This just sparked a question. (I don't know Jeffrey, nor his language)

Assume that Jeffrey implements his language in MPS.

Assume that Jeffrey has a load of ASCII files with definitions in his %\$@&-syntax.

What would be the quickest way to get these definitions into MPS?

-Tom



[Konstantin Solomatov](#) 313 posts since

May 18, 2004 3. **Re: Decision Logic Tables as a DSL** Apr 22, 2009 10:28 PM

Tom, it depends on a task which he wants to accomplish. If he wants to edit these tables a lot, I would recommend MPS. If all he wants to load these tables in his program and use them somehow, it's better to do so with text parser.



[Tom Brus](#) 80 posts since

Jun 2, 2003 4. **Re: Decision Logic Tables as a DSL** Apr 22, 2009 10:55 PM

Konstantin,

(quick answer! ;-)

The task I had in mind was replacing the NotePad/vi/Emacs he is probably using now to edit his definitions by the MPS editor/IDE and maintain his models from then on with a much friendlier tool 😊.

By "*I would recommend MPS*", do you mean entering them **by hand** in the MPS editor he created? That could mean days/weeks of tedious (and error prone) work, right?

By "*with a text parser*", do you mean: write a text parser in your favorite language and take care to write out XML in the correct way for MPS? (BTW he probably already has a parser and only needs to write out his AST as XML)

Does this imply that MPS has no tools to make this easier? I could imagine that internally MPS already has an AST to XML writer in some form. So MPS could make it easier if he could just write a %\$@&-AST to MPS-DecisionLogicTables-AST transformation. right? Is there a way to use this functionality? I could even envisage an ASCII-Import-language (defined in MPS) which lets you define a text based language and produces a parser that takes text and produces a model or XML for a specific MPS-language. Jeffrey could use this

Decision Logic Tables as a DSL

language to define how his %\$@&-language maps to his DecisionLogicTables-language and he would get an importer generated.

Sorry for brainstorming a bit here. These thoughts must have come up in your group in some way, just wondering what your opinion on these things is.

Thanks,

Tom



[Konstantin Solomatov](#) 313 posts since

May 18, 2004 5. **Re: Decision Logic Tables as a DSL** Apr 22, 2009 11:13 PM

By "*I would recommend MPS*", do you mean entering them **by hand** in the MPS editor he created? That could mean days/weeks of tedious (and error prone) work, right?

If he has a parser, he can use it order to convert code to MPS. He mustn't use our XML format, he can just use java classes from MPS stuff. Actually, when we impor java classes in MPS we do very similar thing.



[Jeffrey Cameron](#) 4 posts since

Apr 21, 2009 6. **Re: Decision Logic Tables as a DSL** Apr 23, 2009 12:06 AM

Hey all,

My program actually reads in the files and compiles them into something useful for later on.

I was using CoCo/R (in conjunction with Visual Studio) to do the parsing. I was thinking of re-writing it in ANTLR since the program is more robust but when I came across the article on MPS I thought to myself ... maybe this would do it in a much nicer fashion ...

Does that clarify the issue some?



[Konstantin Solomatov](#) 313 posts since

May 18, 2004 7. **Re: Decision Logic Tables as a DSL** Apr 23, 2009 12:29 AM

We don't have a parser generator in MPS but if you implement a parser with ANTLR, you can use it to convert files that you have into MPS. If you don't have too much code, it might be worth writing them in manually. After you converted them in MPS, you will have IDE, you will be create language extensions which will allow you to express better what you want.

BTW, what do you compile your rules to? C#? XML?



[Jeffrey Cameron](#) 4 posts since

Apr 21, 2009 8. **Re: Decision Logic Tables as a DSL** Apr 24, 2009 8:30 PM

Hi Konstantin,

I use Coco/R right now to convert the DLT into C# objects on the fly. It's a bit tedious however, which is why I thought MPS would be well suited to my needs.

That's an interesting idea to use an MPS in the intermediate, Perhaps I will give that a try.

Thanks for all your help!



[Jeffrey Cameron](#) 4 posts since

Apr 21, 2009 9. **Re: Decision Logic Tables as a DSL** Apr 23, 2009 12:09 AM

I use C#/.NET 3.5 😊