

What-If Analyzer for DMN-based Decision Models

Jacob Feldman

OpenRules, Inc., 53 Riviera Dr.,
Monroe, NJ 08831, USA
jacobfeldman@openrules.com

Abstract. This paper describes a new web-based graphical tool “What-If Analyzer for Decision Modeling” that supports what-analysis of different business decision models created in accordance with the DMN standard. It allows a user to modify the decision model by activating/deactivating business rules that represent its decision logic, and immediately see changes in the decision variables. With a simple click it may produce and navigate through multiple decisions that satisfy all active rules within the same decision model. If the decision model specifies a business objective that depends on other decision variables, then What-If Analyzer may find optimal decisions that minimize or maximize this objective. What-If Analyzer is implemented as an advanced component of the popular open source business decision management system “OpenRules”.

Keywords: Decision Modeling, Decision Optimization, DMN, Multiple and Optimal Decisions.

1 Introduction

Five years ago we published a paper [1] that described how constraint solvers may serve as rule engines in the context of modern business rules and decision management systems. At that time we considered business decision models created based on the methodological approach described in the book “The Decision Model” [3]. Since then OMG has approved a new standard “Decision Model and Notation” (DMN) [2] that quickly gains popularity among decision management vendors, experts, and practitioners. The DMN standard covers more powerful and complex decision models to compare with [3]. In this paper we are applying the results of [1] to DMN-based decision models by presenting them as constraint satisfaction and optimization problems. We also describe a generic graphical interface that allows a user to do what-if analysis of such decision models finding multiple feasible decisions and optimal decisions which minimize/maximize certain business objectives.

In the section 2 we provide a formal definition of the DMN-based decision model as a constraint satisfaction and optimization problem. The section 3 describes a new OpenRules tool called “What-If Analyzer for Decision Modeling” [10] that utilizes constraint solvers to do real-time what-if analysis of decision models. This section demonstrates the major functionality of What-If Analyzer using decision models for two problems: a simple arithmetic problem and a loan origination example. The

section 4 describes how What-If Analyzer has been implemented including the execution engines, graphical interface, and implementation restrictions. And finally, in the section 5 we describe the related work and future development.

2 Decision Model as Constraint Satisfaction Problem

Formally, a **decision model** is defined by a set of **variables**, $X = \{ X_1, X_2, \dots, X_n \}$, and a set of rules, $R = \{ R_1, R_2, \dots, R_m \}$. Each variable X_i has a nonempty **domain** D_i of possible **values**. Each rule R_i defines relationships between different variables from a subset of X and specifies the allowable combinations of values for that subset. A **decision** is an assignment of values to all mentioned variables, $\{ X_i=v_i, X_j=v_j, \dots \}$, that satisfies all the rules.

The decision model can produce multiple decisions. Some decision models require a decision that minimizes/maximizes an **objective function**, which usually depends on some variables.

This representation demonstrates that the Decision Model is quite similar the classical definition of CSP – see for example Russel and Norvig [4] where decision variables X_i correspond to constrained variables and where rules R_i correspond to constraints.

In [1] we had shown how a decision model created in accordance the methodological approach [3] can be reformulated is a special case of CSP with using mainly conditional constraints. The DMN standard [2] extends decision modeling functionality beyond [3] by allowing various multi-hit decision tables, various aggregation functions, and a special expression language “FEEL” that include loops, indexes, and invocation of external functions. These new constructs may lead us beyond conditional constraints and we will need to show precisely how each of them can be implemented within a constraint satisfaction environment. However, the majority of the current DMN implementations [5] do not cover (at least yet) these new DMN concepts and in this paper we will consider DMN-based decision models without multi-hit tables and advanced FEEL features.

3 What-If Analyzer for Decision Modeling

“What-If Analyzer for Decision Modeling” [10] is a stand-alone web application built on top of OpenRules [6]. Its main purpose is to support what-if analysis of DMN-based decision models. What-if analysis is the process of changing business rules that represent business logic to see how these changes affect the outcome of the decision model. What-If Analyzer functionality includes:

- Downloading a DMN-based decision model
- Show it graphically by presenting the current state of all decision variables and all business rules

- Activation/Deactivation of different rules and with an immediate propagation of these actions by displaying changes in the domains of affected decision variables.
- Find a feasible solution that satisfies all currently active rules
- Find and navigate through multiple feasible solutions
- Find optimal solutions for the defined business objectives.

We will demonstrate various capabilities of the What-If Analyzer using two sample decision model:

- A simple arithmetic decision model
- A loan origination decision model.

3.1 Analyzing Simple Arithmetic Decision Model

First, let's consider a small decision model that deals with only 4 decision variables and various arithmetic constraints defined on them. Here it is a sample of its graphical view:

What-If Analyzer for Decision Modeling
 Activate/Deactivate Business Rules, Find, Optimize, and Analyze Different Decisions

Selected Decision Model: Equations and Inequalities Settings

This simple decision model allows you to activate/deactivate various arithmetic constraints on integer decision variables

Solution Next Prev Minimize Maximize Activate All Rules Deactivate All Rules

Decision Variables (Glossary)		Business Rules (Original Rules)			
Variable Name	Possible Values	Rule	Decision Table	Active	Conflict
X	2..7	X+Y=Z	XplusYeqZ	<input checked="" type="checkbox"/>	-
Y	3..8	Objective = X*Y-Z	DefineObjective	<input checked="" type="checkbox"/>	-
Z	6..10	X < Y	BinaryConstraints	<input checked="" type="checkbox"/>	-
Objective	5..20	Z > 5	BinaryConstraints	<input checked="" type="checkbox"/>	-
		Z < 5	BinaryConstraints	<input type="checkbox"/>	-
		Y < 6	BinaryConstraints	<input type="checkbox"/>	-
		Z < 8	BinaryConstraints	<input type="checkbox"/>	-

Rule 'Z > 5' activated OpenRules Release 6.3.4

This decision model was downloaded from Excel-files which presented it in accordance with the OpenRules [6] format. On the left you may see 4 decision variables X, Y, Z, and Objective that were initially defined in the following Excel-based glossary:

Glossary glossary				
Decision Variable	Business Concept	Attribute	Domain	Unknown
X	XYZ	x	0-10	TRUE
Y		y	0-10	TRUE
Z		z	0-10	TRUE
Objective		objective	5-20	TRUE

On the right you can see all rules (active and not) collected from decision tables which were initially defined in Excel-based as follows:

DecisionTable XplusYeqZ				
ActionXoperYcompareZ				
X <oper> Y <compare> Z				
X	+	Y	=	Z

DecisionTable DefineObjective			
ConclusionVarWithName			
Objective			
X*Y-Z	Is	:= \$V{X}.multiply(\$V{Y}).minus(\$V{Z})	

DecisionTable BinaryConstraints			
On/Off	Var	Oper	Var/Value
On	X	<	Y
Off	Z	>	5
Off	Z	<	5
Off	X	>	4
Off	Y	<	6
Off	Z	<	8
Off	Objective	<	10
Off	Objective	>=	12

When a user activates the rule “Z > 5” by checking the proper flag in the table “Business Rules”, the Analyzer immediately modifies the domain of all affected variables in the table “Decision Variables” on the left. If we deactivate this rule and try to activate the rule “Z < 5”, the analyzer immediately will point us to the conflict:

Solution Next Prev Minimize Maximize

Activate All Rules Deactivate All Rules

Decision Variables ([Glossary](#))

Variable Name	Possible Values
X	1..8
Y	2..9
Z	3..10
Objective	5..20

Business Rules ([Original Rules](#))

Rule	Decision Table	Active	Conflict
X+Y=Z	XplusYeqZ	<input checked="" type="checkbox"/>	-
Objective = X*Y-Z	DefineObjective	<input checked="" type="checkbox"/>	-
X < Y	BinaryConstraints	<input checked="" type="checkbox"/>	-
Z > 5	BinaryConstraints	<input type="checkbox"/>	-
Z < 5	BinaryConstraints	<input checked="" type="checkbox"/>	Yes
Y < 6	BinaryConstraints	<input type="checkbox"/>	-
Z < 8	BinaryConstraints	<input type="checkbox"/>	-

Rule 'Z < 5' activated (with errors)

This way the Analyzer constantly checks consistency of the entire decision model: it is trying to find conflict among all (!) active rules; not only within one decision table but across all decision tables – a functionality that were usually out of reach for the majority consistency validation rules.

When a user pushes the button “**Solution**”, the analyzer will find and show the first feasible solution that satisfies all currently active rules. A user may push buttons “**Next**” and “**Prev**” to find and navigate through the various feasible solutions:

Solution Next Prev Minimize Maximize Activate All Rules Deactivate All Rules

Decision Variables ([Glossary](#))

Variable Name	Possible Values
X	3
Y	4
Z	7
Objective	5

Business Rules ([Original Rules](#))

Rule	Decision Table	Active	Conflict
$X+Y=Z$	XplusYeqZ	<input checked="" type="checkbox"/>	-
Objective = $X*Y-Z$	DefineObjective	<input checked="" type="checkbox"/>	-
$X < Y$	BinaryConstraints	<input checked="" type="checkbox"/>	-
$Z > 5$	BinaryConstraints	<input checked="" type="checkbox"/>	-
$Z < 5$	BinaryConstraints	<input type="checkbox"/>	-
$Y < 6$	BinaryConstraints	<input type="checkbox"/>	-
$Z < 8$	BinaryConstraints	<input type="checkbox"/>	-

Found solution #3 out of 8

This decision model specifies an objective using the rules (Objective = $X*Y-Z$). So, the Analyzer may find an optimal solution that minimizes or maximizes this objective:

Solution Next Prev Minimize Maximize Activate All Rules Deactivate All Rules

Decision Variables ([Glossary](#))

Variable Name	Possible Values
X	4
Y	6
Z	10
Objective	14

Business Rules ([Original Rules](#))

Rule	Decision Table	Active	Conflict
$X+Y=Z$	XplusYeqZ	<input checked="" type="checkbox"/>	-
Objective = $X*Y-Z$	DefineObjective	<input checked="" type="checkbox"/>	-
$X < Y$	BinaryConstraints	<input checked="" type="checkbox"/>	-
$Z > 5$	BinaryConstraints	<input checked="" type="checkbox"/>	-
$Z < 5$	BinaryConstraints	<input type="checkbox"/>	-
$Y < 6$	BinaryConstraints	<input type="checkbox"/>	-
$Z < 8$	BinaryConstraints	<input type="checkbox"/>	-

Maximized solution #7 found

A user may "Deactivate All Rules" and start activating various rules one-by-one and analyzing their contribution to the removal of certain values from the domains of decision variables. Actually a user of What-If Analyzer has an ability to emulate various rules execution modes usually available to only rule engine developers (such as backward-chaining when we activate constraints on the objective).

3.2 Analyzing Loan Origination Decision Model

Loan origination process is frequently used as a typical example of decision modeling. Initially we took a loan-prequalification decision model described at the DMCommunity [website](#). Being downloaded at the What-If Analyzer, it looks as follows:

OPEN RULES What-If Analyzer for Decision Modeling **OPEN RULES**
 Activate/Deactivate Business Rules, Find, Optimize, and Analyze Different Decisions

Selected Decision Model:
 Loan origination decision model with income validation and debt research rules - see [Description](#)

Decision Variables (Global)		Business Rules (Global)	
Variable Name	Possible Values	Rule	Active
Monthly Income	6400	IF Total Income > Doubled Total Debt THEN Income Validation Result = SUFFICIENT	<input checked="" type="checkbox"/>
Monthly Debt	3000	IF Total Income <= Doubled Total Debt THEN Income Validation Result = UNSUFFICIENT	<input checked="" type="checkbox"/>
Mortgage Holder	Yes	IF Mortgage Holder = Yes THEN Debt Research Result = High	<input checked="" type="checkbox"/>
Outside Credit Score	735	IF Mortgage Holder = No AND Outside Credit Score > 100 AND Outside Credit Score <= 550 THEN Debt Research Result = High	<input checked="" type="checkbox"/>
Loan Holder	Yes	IF Mortgage Holder = No AND Outside Credit Score > 550 AND Outside Credit Score <= 900 AND Loan Holder = Yes AND Credit Card Balance <= 0 THEN Debt Research Result = Mid	<input checked="" type="checkbox"/>
Credit Card Balance	1200	IF Mortgage Holder = No AND Outside Credit Score > 550 AND Outside Credit Score <= 900 AND Loan Holder = Yes AND Credit Card Balance > 0 AND Education Loan Balance > 0 THEN Debt Research Result = High	<input checked="" type="checkbox"/>
Education Loan Balance	0	IF Mortgage Holder = No AND Outside Credit Score > 550 AND Outside Credit Score <= 900 AND Loan Holder = Yes AND Credit Card Balance <= 0 AND Education Loan Balance <= 0 AND Internal Credit Rating ISONEOF A, B, C THEN Debt Research Result = High	<input checked="" type="checkbox"/>
Internal Credit Rating	C	IF Mortgage Holder = No AND Outside Credit Score > 550 AND Outside Credit Score <= 900 AND Loan Holder = Yes AND Credit Card Balance > 0 AND Education Loan Balance <= 0 AND Internal Credit Rating ISONEOF D, F THEN Debt Research Result = High	<input checked="" type="checkbox"/>
Internal Analyst Opinion	Mid	IF Mortgage Holder = No AND Outside Credit Score > 550 AND Outside Credit Score <= 900 AND Loan Holder = Yes AND Credit Card Balance <= 0 AND Education Loan Balance <= 0 AND Internal Credit Rating ISONEOF D, F THEN Debt Research Result = Mid	<input checked="" type="checkbox"/>
Loan Term	72		
Total Income	460800	IF Mortgage Holder = No AND Outside Credit Score > 550 AND Outside Credit Score <= 900 AND Loan Holder = No AND Credit Card Balance > 0 THEN Debt Research Result = Low	<input checked="" type="checkbox"/>
Total Debt	216000	IF Mortgage Holder = No AND Outside Credit Score > 550 AND Outside Credit Score <= 900 AND Loan Holder = No AND Credit Card Balance <= 0 AND Education Loan Balance > 0 AND Internal Credit Rating ISONEOF D, F THEN Debt Research Result = Low	<input checked="" type="checkbox"/>
Doubled Total Debt	432000	IF Mortgage Holder = No AND Outside Credit Score > 550 AND Outside Credit Score <= 900 AND Loan Holder = No AND Credit Card Balance <= 0 AND Education Loan Balance <= 0 AND Internal Credit Rating ISONEOF A, B, C THEN Debt Research Result = Low	<input checked="" type="checkbox"/>
Income Validation Result	SUFFICIENT	IF Mortgage Holder = No AND Outside Credit Score > 550 AND Outside Credit Score <= 900 AND Loan Holder = No AND Credit Card Balance <= 0 AND Education Loan Balance > 0 AND Internal Credit Rating ISONEOF D, F THEN Debt Research Result = High	<input checked="" type="checkbox"/>
Debt Research Result	High	IF Mortgage Holder = No AND Outside Credit Score > 550 AND Outside Credit Score <= 900 AND Loan Holder = No AND Credit Card Balance <= 0 AND Education Loan Balance > 0 AND Internal Credit Rating ISONEOF A, B, C THEN Debt Research Result = Low	<input checked="" type="checkbox"/>
Loan Qualification Result	QUALIFIED,NOT QUALIFIED,?	IF Mortgage Holder = No AND Internal Analyst Opinion = High THEN Debt Research Result = High	<input checked="" type="checkbox"/>
		IF Mortgage Holder = No AND Internal Analyst Opinion = Mid THEN Debt Research Result = Mid	<input checked="" type="checkbox"/>
		IF Mortgage Holder = No AND Internal Analyst Opinion = Low THEN Debt Research Result = Low	<input checked="" type="checkbox"/>
		IF Internal Analyst Opinion = Low THEN Debt Research Result = Low	<input checked="" type="checkbox"/>
		IF Income Validation Result = UNSUFFICIENT THEN Loan Qualification Result = NOT QUALIFIED	<input checked="" type="checkbox"/>
		IF Income Validation Result = SUFFICIENT AND Debt Research Result = Low THEN Loan Qualification Result = NOT QUALIFIED	<input checked="" type="checkbox"/>
		IF Income Validation Result = SUFFICIENT AND Debt Research Result ISONEOF Mid, High THEN Loan Qualification Result = QUALIFIED	<input checked="" type="checkbox"/>

Rule 'IF Income Validation Result = SUFFICIENT AND Debt Research Result ISONEOF Mid, High THEN Loan Qualification Result = QUALIFIED' deactivated

OpenRules Release 3.1.4

In this example a borrower provides the requested loan amount, loan term and her personal info financial information as an input, and the decision model produces one of two possible decision outputs:

- the borrower is qualified for the loan
- the borrower is not qualified for the loan with an explanation.

While this model will be successfully executed by the Analyzer, it does not provide a flexibility to keep the borrower interested when an initial loan request is rejected. It would be much more beneficial if the decision model can also recommend how to modify the loan request (e.g. decrease loan amount or increase loan term) to still make the borrower qualified.

Below we will consider a modified decision model that allows a loan officer to consider different borrowing options using the What-If Analyzer. To do that, we will consider Loan Amount and Loan Term as unknown decision variables which should belong to the output of our decision model. Then Total Income and Total Debt also

become unknown, e.g. Total Income = Monthly Income * Loan Term. For simplicity we will use the following income validation rules:

DecisionTable DetermineIncomeValidationResult				
ConditionXoperY			Conclusion	
X <oper> Y			Income Validation Result	
Total Income	>	Accumulated Debt	Is	SUFFICIENT
Total Income	<=	Accumulated Debt	Is	UNSUFFICIENT

Here the decision variable “Accumulated Debt” is equal to the sum of the total loan amount and the total debt over the selected loan term. We also added several rules that will allow us to define different constraints on loan amount and loan term. The proper decision model inside the What-If Analyzer will look as follows:

Decision Variables (Glossary)		Business Rules (Original Rules)		
Variable Name	Possible Values	Rule	Active	Conflict
Monthly Income	4100	Total Debt = expression	<input checked="" type="checkbox"/>	-
Monthly Debt	2600	Total Income = expression	<input checked="" type="checkbox"/>	-
Loan Amount	50000	Accumulated Debt = expression	<input checked="" type="checkbox"/>	-
Loan Term	24	Loan Amount >= 35000	<input checked="" type="checkbox"/>	-
Total Income	98400	Loan Amount = 50000	<input checked="" type="checkbox"/>	-
Total Debt	62400	Loan Amount <= 75000	<input checked="" type="checkbox"/>	-
Accumulated Debt	112400	Loan Amount >= 40000	<input type="checkbox"/>	-
Income Validation Result	UNSUFFICIENT	Loan Amount >= 50000	<input type="checkbox"/>	-
		Loan Amount >= 60000	<input type="checkbox"/>	-
		Loan Term = 24	<input checked="" type="checkbox"/>	-
		Loan Term = 36	<input type="checkbox"/>	-
		Loan Term = 72	<input type="checkbox"/>	-
		Loan Term <= 36	<input type="checkbox"/>	-
		IF Total Income > Accumulated Debt THEN Income Validation Result = SUFFICIENT	<input checked="" type="checkbox"/>	-
		IF Total Income <= Accumulated Debt THEN Income Validation Result = UNSUFFICIENT	<input checked="" type="checkbox"/>	-
		Income Validation Result = SUFFICIENT	<input checked="" type="checkbox"/>	Yes

As you can see, the loan request here is converted to two active rules:

- Loan Amount = 50000
- Loan Term = 24 (months)

Please note that we added a rule (the last one) that enforces Income Validation Result to be SUFFICIENT. The What-If Analyzer immediately produces a conflict as the last rule contradicts to income validation rules. So, we may deactivate the rule “Loan Amount = 50000” leaving Loan Amount being between 35000 and 75000. Here is the resulting view:

Decision Variables (Glossary)		Business Rules (Original Rules)		
Variable Name	Possible Values	Rule	Active	Conflict
Monthly Income	4100	Total Debt = expression	<input checked="" type="checkbox"/>	-
Monthly Debt	2600	Total Income = expression	<input checked="" type="checkbox"/>	-
Loan Amount	35000..35999	Accumulated Debt = expression	<input checked="" type="checkbox"/>	-
Loan Term	24	Loan Amount >= 35000	<input checked="" type="checkbox"/>	-
Total Income	98400	Loan Amount = 50000	<input type="checkbox"/>	-
Total Debt	62400	Loan Amount <= 75000	<input checked="" type="checkbox"/>	-
Accumulated Debt	97400..98399	Loan Amount >= 40000	<input type="checkbox"/>	-
Income Validation Result	SUFFICIENT	Loan Amount >= 50000	<input type="checkbox"/>	-
		Loan Amount >= 60000	<input type="checkbox"/>	-
		Loan Term = 24	<input checked="" type="checkbox"/>	-
		Loan Term = 36	<input type="checkbox"/>	-
		Loan Term = 72	<input type="checkbox"/>	-
		Loan Term <= 36	<input type="checkbox"/>	-
		IF Total Income > Accumulated Debt THEN Income Validation Result = SUFFICIENT	<input checked="" type="checkbox"/>	-
		IF Total Income <= Accumulated Debt THEN Income Validation Result = UNSUFFICIENT	<input checked="" type="checkbox"/>	-
		Income Validation Result = SUFFICIENT	<input checked="" type="checkbox"/>	-

Rule 'Loan Amount = 50000' deactivated

Now the conflict is resolved, and the decision variable “Loan Amount” is bounded to 35000..35999 as defined a pure propagation of the rule deactivation event. It is interesting to see that the maximal loan amount is now 35999 that can be explained by the automatically propagated rule “Loan Term = 24”.

To increase the maximally allowed Loan Amount we may try to increase Loan Term. Let’s deactivate the rule “Loan Term = 24” but activate the rule “Loan Term <= 36”. When we click on the button “Solution”, the Analyzer will quickly find the first feasible solution, and then we may click to the buttons “Next” and “Prev” to navigate through multiple solutions.

For example, after 28 clicks on the button “Next” the Analyzers will show a solution # 28 out of 500. Actually the Analyzer may find much more feasible solutions if we change the limit “500” using the Analyzer’s button “Settings”. Here is the solution#28:

Decision Variables (Glossary)		Business Rules (Original Rules)		
Variable Name	Possible Values	Rule	Active	Conflict
Monthly Income	4100	Total Debt = expression	<input checked="" type="checkbox"/>	-
Monthly Debt	2600	Total Income = expression	<input checked="" type="checkbox"/>	-
Loan Amount	35013	Accumulated Debt = expression	<input checked="" type="checkbox"/>	-
Loan Term	36	Loan Amount >= 35000	<input checked="" type="checkbox"/>	-
Total Income	147600	Loan Amount = 50000	<input type="checkbox"/>	-
Total Debt	93600	Loan Amount <= 75000	<input checked="" type="checkbox"/>	-
Accumulated Debt	128613	Loan Amount >= 40000	<input type="checkbox"/>	-
Income Validation Result	SUFFICIENT	Loan Amount >= 50000	<input type="checkbox"/>	-
		Loan Amount >= 60000	<input type="checkbox"/>	-
		Loan Term = 24	<input type="checkbox"/>	-
		Loan Term = 36	<input type="checkbox"/>	-
		Loan Term = 72	<input type="checkbox"/>	-
		Loan Term <= 36	<input checked="" type="checkbox"/>	-
		IF Total Income > Accumulated Debt THEN Income Validation Result = SUFFICIENT	<input checked="" type="checkbox"/>	-
		IF Total Income <= Accumulated Debt THEN Income Validation Result = UNSUFFICIENT	<input checked="" type="checkbox"/>	-
		Income Validation Result = SUFFICIENT	<input checked="" type="checkbox"/>	-

Found solution #28 out of 500

When the total number of solutions is relatively small it may be useful for a user to manually navigate through different solutions. However, it would be too tedious or even impossible to do that. Instead, we may simply click on the button “Maximize” and the Analyzer (after a small delay) will find the maximal Loan Amount. Here is a list of all decision variables with values which correspond to the optimal decision:

Decision Variables (Glossary)	
Variable Name	Possible Values
Monthly Income	4100
Monthly Debt	2600
Loan Amount	53999
Loan Term	36
Total Income	147600
Total Debt	93600
Accumulated Debt	147599
Income Validation Result	SUFFICIENT

Maximized solution #2 found

So, the maximal Loan Amount is \$53,999 while selected the Loan Term is 36. Similarly, we may continue to deactivate and activate other limits to find various optimal combinations of Loan Amount and Loan Term while satisfying other active loan origination rules.

4 Implementation

Execution Engines. What-If Analyzer has been built on top of OpenRules Engine known as “Rule Solver” [7]. Rule Solver is capable to download OpenRules decision model directly from Excel files and automatically generate the corresponding constraint satisfaction and optimization problem. Rule Solver is based on the standard Java Constraint Programming API defined by the Java Specification Request (JSR) 331 [8]. The use of the JSR 331 allows a user not to commit to a particular CP vendor and to try different compliant solvers before choosing the most suitable one based on its technical and business applicability. With this approach, we may switch between different underlying CP solvers compliant with the JSR 331 without any changes in the code.

Graphical Interface. The graphical interface for the Analyzer has been built using OpenRules Dialog [9] that provides a simple rules-based environment for building web applications. We recognize that one universal graphical interface cannot satisfy all decision models that in real-world may require a lot of application-specific features. So, the choice of OpenRules Dialog allows a user to essentially customize the graphical interface of What-If Analyzer without major changes in the underlying logic.

Implementation Restrictions. The current implementation of What-If Analyzer supports decision models created in accordance the methodological approach described in [1]. It currently does not support multi-hit decision tables and aggregation functions allowed by the DMN standard [2]. It also supports only basic constructs of the DMN FEEL language and provides its own facilities for various constrained expressions implemented within Rule Solver.

While the latest version 1.1 of the DMN specifies interchange format, there are no practical tools yet that supports DMN XML. So, currently What-If Analyzer works only with OpenRules-based decision models.

In spite of these implementation restriction, What-If Analyzer covers a wide class of practical decision models, and empowers users with features that were out of reach for the majority of business rules and decision management systems.

5 Related Work and Future Development

What-If Analyzer for Decision Modeling provides unique features which so far were out of reach for the majority of business rules and decision management systems. However, a similar functionality was available for constraint-based environments oriented to more technical users. In particular, in [11] we described practical design patterns for constraint programming including the “Consistent Constrained Core” pattern that is very similar to how What-If Analyzer execution engine has been implemented. And in [12] we demonstrated how this pattern was applied to scheduling and resource allocation problems. Of course, these systems dealt directly with constraint satisfaction problems and did not include the proper transformation of business oriented decision models.

In the future, we plan to remove the implementation restrictions described above. It will require finding a new constraint-based approach to implementation of multi-hit decision tables with various aggregation functions. It will probably lead us far beyond of conditional constraints described in [1]. For example, even simple rule overrides already require a new approach as we cannot simply assign a new value to already instantiated decision variables within a constraint satisfaction environment as it will lead to a failure. Even more complicated issues to consider are the situations when a decision table is accumulating different values inside one decision variable using compound assignment operators such as “+=”. In such situations, we would probably need to accumulate all conditional assignments in an intermediate array before applying them to the resulting decision variable. We would also need to consider a converter from DMN FEEL expressions to constrained expressions using JSR-331. All these consideration considerations will be a subject for additional R&D.

When the DMN interchange format becomes commonly accepted, we will add a module that will download decision models presented in the DMN XML. It will allow us to extend the use of What-If Analyzer to decision models created not within the OpenRules framework but also by other decision modeling products.

References

1. J.Feldman: Representing and Solving Rule-Based Decision Models with Constraint Solvers, in Proc. of RuleML 2011 - America, LNCS 7018, pp. 208-221, 2011, Springer-Verlag Berlin Heidelberg 2011, <http://www.springerlink.com/content/e66077u456548231>
2. Decision Model and Notation™ (DMN™), Object Management Group (OMG), <http://www.omg.org/spec/DMN/Current>
3. B. von Halle, L.Goldberg: The Decision Model: A Business Logic Framework Linking Business and Technology. Auerbach Publications/Taylor & Francis Group, LLC (2009)
4. S.Russell, P.Norvig: Artificial intelligence: a modern approach, Third Edition, <http://aima.cs.berkeley.edu/2nd-ed/newchap05.pdf>
5. DMN Supporting Tools, Decision Management Community, <http://DMCommunity.org>, <http://openjvm.jvmhost.net/DMNtools>

6. OpenRules, Open Source Business Rules and Decision Management System, <http://openrules.com>
7. OpenRules Rule Solver, <http://openrules.com/rulesolver.htm>
8. Java Request Specification (JSR) 331: Constraint Programming API. Java Community Process, <http://www.jcp.org/en/jsr/detail?id=331>
9. OpenRules Dialog – a Questionnaire Builder, <http://openrules.com/ord.htm>
10. What-If Analyzer for Decision Modeling, <http://openrules.com/WhatIfAnalyzer.htm>
11. J. Feldman, D. Vergamini: Practical Patterns for Constraint Programming, PACP-98, 1998, London, <http://openrules.com/presentations/pact98.pdf>
12. J. Feldman: Interactive Constraint Propagation, INFORMS, November-1999, Philadelphia, <http://openrules.com/presentations/INFORMS99.ppt>