MRMSolve: A Tool for Transient Analysis of Large Markov Reward Models*

S. Rácz†, B. P. Tóth‡, M. Telek‡

† Dept. of Telecommunications and Telematics
Technical University of Budapest, 1521 Budapest, Hungary

† Department of Telecommunications
Technical University of Budapest, 1521 Budapest, Hungary

† raczs@ttt-atm.ttt.bme.hu; ‡ {ptot,telek}@indus.hit.bme.hu

Abstract. MRMSolve is a new analysis tool developed for the evaluation of large Markov Reward Models (MRM) that provides the moments of the accumulated reward and the completion time. MRMSolve is based on the Java technology, hence it allows to access the tool from any node connected with the Internet as long as it possesses a Java-enabled Web browser.

Introduction Markov reward models have been effectively used for the performance and preformability analysis of computer and communications systems. The evaluation of the distribution of reward measures (i.e., the accumulated reward and the completion time) is a computationally hard procedure, hence it can not be performed for MRMs with a large state space (10^5-10^6 states). A recently introduced numerical technique allows to evaluate the moments of reward measures of MRMs with large state space [2]. The MRMSolve tool is based on this numerical technique.

MRMSolve is composed by an analysis engine and the graphical user interface (GUI). The analysis engine is a C++ implementation of the numerical method introduced in [2]. The GUI provides a world wide access to the analysis engine using the JAVA technology.

In this summary we focus on the structure and the GUI of MRMSolve and we do not enter the details of modeling and analysis of real systems with MRMs.

The structure of MRMSolve MRMSolve makes the use of the client server architecture, as it is depicted on Figure 2. The server runs on a powerful remote machine, while the client program can be downloaded to any JAVA enabled computer that is connected with the Internet. The MRM to be analyzed and the required performance measures are defined on the GUI. When the model definition is complete and correct the model is uploaded to the server machine, which performs the (usually time consuming) computation and downloads the result to the client.

* This work was supported by OTKA T-30685 and F-23971.

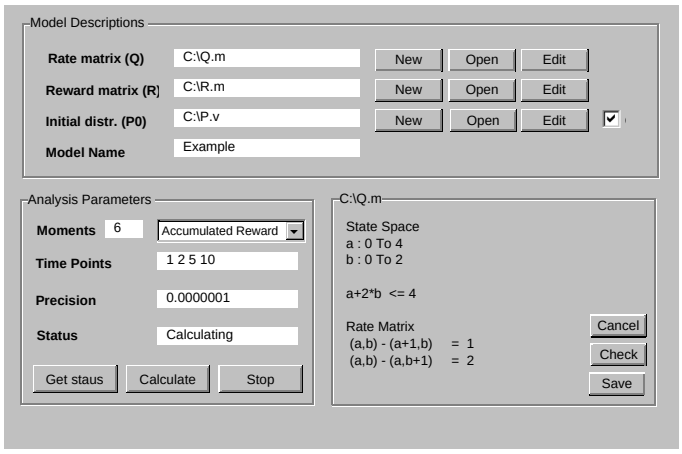

Fig. 1. The input screen of MRMSolve

In the design of MRMSolve it is considered that the client machine is not necessarily powerful and the connection between the client and the server can be slow depending on the condition of Internet. Based on these considerations the computing effort required at the client side is minimized and an effective data representation is used to describe and upload MRMs with large state space as it is discussed below. The data size of the results are usually negligible.

Graphical user interface MRMSolve can be started with appletviewer or with any Java enabled web browser from the following address:
`http://indus.hit.bme.hu/~ MRMSolve` After starting the Java applet the Input screen of MRMSolve appears (Figure 1). On this screen the name of the model, and the model data (generator matrix, reward rate vector and the initial distribution vector) has to be defined. There are two ways to define input matrices and vectors. Existing model description files can be selected and opened using the "Open" button, or new model descriptions can be created using the built-in editor by clicking on the "New" button. Any opened model description files can be modified using the same editor.

The consistency of the opened description file can be checked using the "Check" button. The required results and the parameters of the analysis can be defined in the lower left block of the input screen, i.e., the number of the moments, time (reward) points, required precision. The calculation can be started by hitting the "Calculate" button and can be stopped with "Stop" button. After starting calculation the client applet uploads all model parameters to the server where a C++ programs evaluates the model. The data communication with the client is implemented in Java in the server side as well. During the computation the actual status of the evaluation (e.g. uploading model, processing rate matrix description, calculating, processing result file,...) is indicated on the input screen in the "Status" field. The text file of the obtained results are shown on the Results screen.

State Space
#Section 1
$< var_1 >$: $< initial\_value_1 >$ **To** $< end\_value_1 >$
$< var_2 >$: $< initial\_value_2 >$ **To** $< end\_value_2 >$
#Section 2
$< boolean\_expr >$
Matrix
$(var_1, var_2) - (< expr >, < expr >) = < expr >$
$(< expr >, < expr >) - (var_1, var_2) = < expr >$
$< boolean\_expr >$: $(var_1, var_2) - (< expr >, < expr >) = < expr >$
$< boolean\_expr >$: $(< expr >, < expr >) - (var_1, var_2) = < expr >$

Table 1. The syntax of matrix description

State Space	**State Space**	**State Space**
a : 0 **To** 4	a : 0 **To** 4	a : 0 **To** 4
b : 0 **To** 2	b : 0 **To** 2	b : 0 **To** 2
$a + 2 * b <= 4$	$a + 2 * b <= 4$	$a + 2 * b <= 4$
Rate Matrix	**Reward Vector**	**Initial Vector**
$(a, b) - (a + 1, b) = 1$	$(a, b) = a$	$(a, b) = 1$
$(a, b) - (a, b + 1) = 2$		
$(a, b) - (a - 1, b) = 5 * a$		
$(a, b) - (a, b - 1) = 4 * b$		
$(a, b) - (a - 1, b + 1) = 3 * a$		

Table 2. The rule-based description of the example

Data representation The applied client-server architecture requires an effective description of large MRM models at the client side, because it would not be possible to upload some Mbyte large generator matrix of a MRM of some hundred thousand states. To reduce the computational requirements at the client side and the amount of uploaded data a rule-based matrix (vector) description is applied. Basically a set of rules has to be provided at the client side that is used to build the generator matrix at the server side. This way only the set of rules are uploaded ($\sim KByte$). This rule based model description is simple and effective if the MRM posses a nice structure. In case of rather complicated MRMs a local installation of the analysis engine might help.

The applied rule based model description (Table 1.) are composed by general expression ($< expr >$) that can contain variables ($< var > \in \mathbf{Z}$) defined in #Section 1, arithmetic operators (+ - * /) and several functions (**min**(.,.), **max**(.,.), **sqrt**(.), **abs**(.), **power**(.,.), **log**(.), **ln**(.), ...). A boolean expression can contain variables, arithmetic operators, several functions and logical operators (AND, OR, NOT). Comments are denoted by #.

Application example Consider a transmission link of capacity C = 4 Mbps, which is offered calls according to a Poisson process belonging two different service classes. The calls of the first service class arrive at rate 1, depart at rate 5 and require 1 Mbps bandwidth. The calls of the second service class arrive at

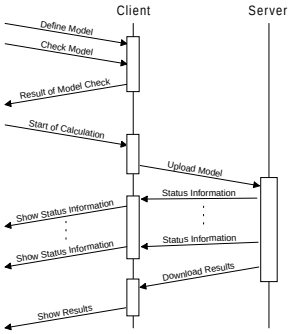

Fig. 2. Client server functions of MRMSolve.

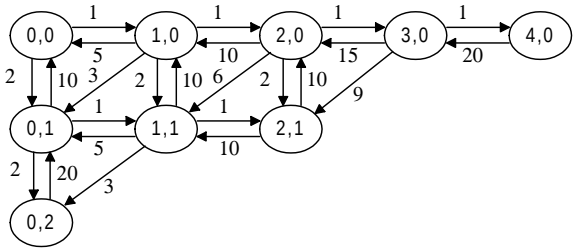

Fig. 3. The Markov chain of generated based on the model description

	$\mathbf{E}(\mathcal{B}(\mathbf{t}))$	$\mathbf{E}(\mathcal{B}(\mathbf{t})^2)$	$\mathbf{E}(\mathcal{B}(\mathbf{t})^3)$	$\mathbf{E}(\mathcal{B}(\mathbf{t})^4)$	$\mathbf{E}(\mathcal{B}(\mathbf{t})^5)$	$\mathbf{E}(\mathcal{B}(\mathbf{t})^6)$
$t = 1$	0.304	0.180	0.144	0.144	0.170	0.232
$t = 2$	0.427	0.302	0.281	0.319	0.425	0.649
$t = 5$	0.798	0.852	1.111	1.692	2.935	5.688
$t = 10$	1.416	2.381	4.581	9.874	23.48	60.92

Table 3. Moments of accumulated reward

rate 2, depart at rate 10 and require 2 Mbps. If there is enough free capacity the calls of the first service class expand their bandwidth to 2 Mbps at rate 3 and behave as second class calls from that time on. The system starts from a uniform initial distribution, which is defined as $(a, b) = 1$, because the state probabilities are automatically normalized by the program. The total amount of data transmitted at 1 Mbps capacity over the $(0, t)$ interval is evaluated by MRMSolve. The structure of underlying CTMC is shown in Figure 3, while the rule based description of the example is summarized in Table 3. The first 6 moments of the data transmitted at 1 Mbps is shown in Table 3.

Future plans MRMSolve is under development. The next version of the tool is going to include the GUI for the analysis of MRMs with rate and impulse rewards and the analysis method provided in [1].

References

1. S. Rácz and M. Telek. Performability analysis of Markov reward models with rate and impulse reward. In M. Silva B. Plateau, W. Stewart, editor, *Int. Conf. on Numerical solution of Markov chains*, pages 169–187, Zaragoza, Spain, 1999.
2. M. Telek and S. Rácz. Numerical analysis of large Markovian reward models. *Performance Evaluation*, 36&37:95–114, Aug 1999.