

Process algebras for quantitative analysis

University of Pisa PhD School Examination

Jane Hillston and Stephen Gilmore
School of Informatics
The University of Edinburgh

February 21, 2010

This is the assessed examination for the lecture course given in February 2010. This is an individual examination: whilst general discussion with others is fine, the solutions that you submit must be your own work.

*Choose **either** question 1 (on PEPA) **or** question 2 (on Bio-PEPA). Submit your solution by email to Stephen.Gilmore@ed.ac.uk*

1 Modelling with PEPA

If you choose to do this question you will be working with the Eclipse PEPA Plug-in Project (available from <http://www.dcs.ed.ac.uk/pepa/tools/plugin/>). Download and install this on your own computer by following the instructions on that Web site.

1.1 Overview

A credit portal is a business-to-business (B2B) application where a bank wishes to offer its loan services to business customers. It is important that loan applications are processed efficiently (so that the bank does not lose customers to a rival bank). A high degree of automation is provided to meet this demand for efficient processing. Where human intervention is needed in the decision-making process this too must be driven by a deep sense of urgency in processing the request in a timely manner.

Some businesses which use the services of the bank are long-established customers with a lengthy and well-documented history of financial solidity and probity, together with substantial credit reserves and securities. Modest loan applications from these customers can be directly approved by a *pre-decision* process which validates the content of the loan request and the securities. In such a case it is possible that the decision to lend can be taken entirely by the application of the predefined rules stored on the server and executed by the local rule engine. This is the fastest route to approval of a loan request because it is one which completes entirely without human intervention.

Not all loan requests can be so rapidly approved. Many will require more lengthy scrutiny, evaluation and checking. In these cases a bank clerk will process the credit request. There are several possible next outcomes here. First, the clerk may approve the request but must then forward the credit request to a supervisor who must also approve the request. Second, the clerk may decline the request. Third, the clerk may enter into a negotiation with the customer with the intention of updating the request to reduce the capital requested, or change the terms of

repayment. This then initiates another request from the customer which is to be processed in the manner just described.

1.2 Description

We begin with a description of the bank’s customers and the process which they follow in order to secure a loan from the bank. This presents the customer’s view of the process in terms of activities and choices along the way.

The customer’s first action is to initiate a loan (*request*). To carry this through they must enter their balance data and securities (*enterData*) and send this to the credit portal with an XBRL upload (*uploadData*). (XBRL is the eXtensible Business Reporting Language.) The customer then waits to see if the request will be approved or declined (the *approve* and *decline* activities respectively, undertaken at the same rate, r_{inform}). If the request is approved the customer has no further business and the next waiting customer can be considered. If the request is declined the customer can try again (*reapply*) and will do so with probability t_0 . If they do not wish to reapply they can yield to the next customer.

The service is a reactive system. It takes no action until the upload of XBRL data is complete. At this point it validates the data by using a validation web service which determines whether or not the balance data is valid (*validateData*). We are not interested in the processing of invalid data in this scenario and so the next behaviour which we model is passing the valid data to the bank (*sendBank*). The service is then ready to receive the next request.

The relevant business functions of the bank are expressed in the *Bank* component. This documents the “predecision” phase which can have three possible outcomes. Some applications can be immediately approved (with probability p_0), and others immediately declined (with probability p_1). Some proportion need to be processed by a bank employee (with probability p_2) who will either decline the loan (with probability q_1), or approve it (with probability q_0). Approval requires confirmation, which may be forthcoming (with probability s_0) or not.

Values for rates and probabilities can be found in Tables 1 and 2.

Customer rates		Service rates		Bank rates	
$r_{request}$	= 3.11944	$r_{validate}$	= 1.43141	$r_{predecide}$	= 5.15757
$r_{enterData}$	= 0.04667	$r_{sendBank}$	= 1.53785	r_{decide}	= 0.01221
r_{upload}	= 0.88424			r_{inform}	= 0.45729
$r_{reapply}$	= 0.02036				

Table 1: Table of rate values used in the model. All rates are expressed at the granularity of minutes. The reciprocal of the rate gives the mean or expected value of the duration of the activity. Thus, the average time for one bank employee to decide on a loan is about 82 minutes ($1/r_{decide} = 81.90008$).

1.3 Assessment

1. Model the above system using PEPA and the PEPA Eclipse Plugin.
2. Use the analysis capabilities of the PEPA Eclipse Plugin to perform a variety of analyses on your model. Save these analyses in the form of graphs stored in the PNG image file format supported by the tool.
3. Submit the model and the graphs by email to Stephen.Gilmore@ed.ac.uk.

Predecision	Employee decision	Supervisor decision	Reapplication
$p_0 = 0.20907$	$q_0 = 0.17441$	$s_0 = 0.56423$	$t_0 = 0.08970$
$p_1 = 0.32075$	$q_1 = 0.82559$	$s_1 = 0.43577$	$t_1 = 0.91030$
$p_2 = 0.47018$			

Table 2: Table of probability values used in the model. Each column sums to 1.

4. If needed, accompany your model with a brief description (1 page or less) describing any problems which you encountered in doing the analysis, or any additional assumptions which you needed to make.

2 Modelling with Bio-PEPA

If you choose to do this question you will be working with the Eclipse Bio-PEPA Plug-in Project (available from <http://biopepa.org>). Download and install this on your own computer by following the instructions on that Web site.

2.1 Overview

The BioModels website (biomodels.net) contains a database of biological models expressed in biological modelling languages such as SBML and CellML. The models are downloadable in XML format, or can be browsed as Web pages. (Figure 1 shows an example.) For this question you are required to choose a model from the BioModels database and to encode it in Bio-PEPA.

Models in the BioModels database vary in size and complexity. You are recommended to choose a model with a relatively small number of reactions and species and relatively simple kinetic functions.

2.2 Assessment

1. Encode your selected BioModels example in Bio-PEPA.
2. Analyse the model using the Bio-PEPA Eclipse Plug-in to produce a time series plot.
3. Use the analysis capabilities of the Bio-PEPA Eclipse Plugin to perform a variety of analyses on your model. Save these analyses in the form of graphs stored in the PNG image file format supported by the tool.
4. Generate an SBML model from the Bio-PEPA model using the SBML export feature of the Bio-PEPA Eclipse Plug-in.
5. Submit the Bio-PEPA and SBML models, and graphs to Stephen.Gilmore@ed.ac.uk.
6. If needed, accompany your model with a brief description (1 page or less) describing any problems which you encountered in doing the analysis, or any additional assumptions which you needed to make.

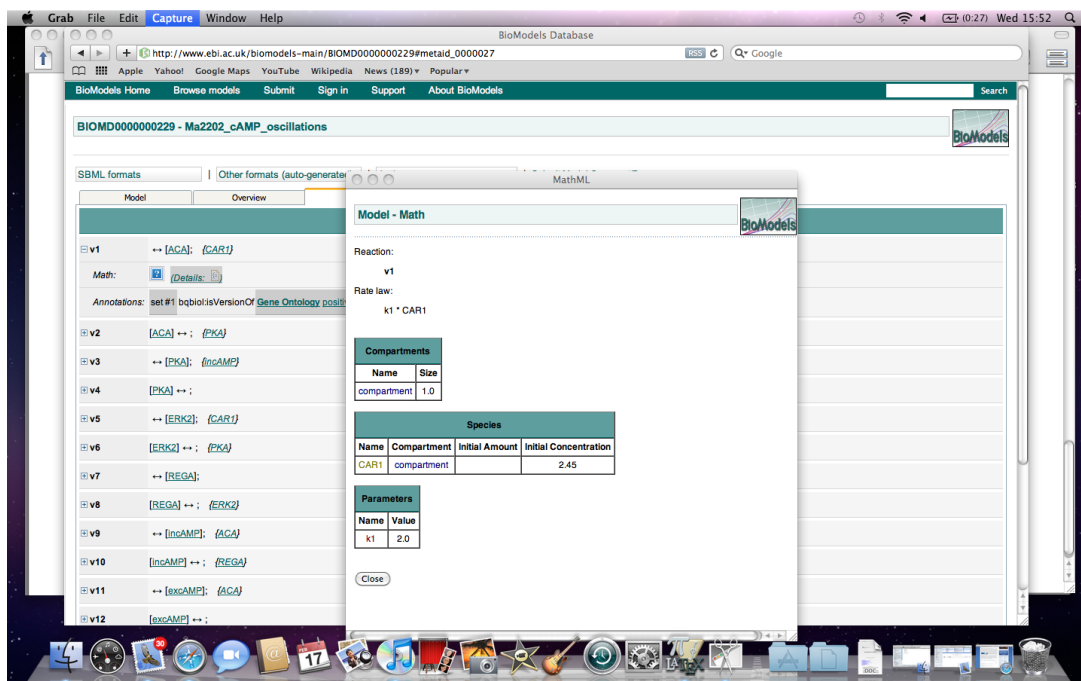


Figure 1: A screenshot of a model in the BioModels database