CONTENTS

IITSBE, Vol. 2(11) 2011

Section: Business process modeling

3-8 INVESTIGATION OF FINANCIAL MARKET PREDICTION BY RECURRENT NEURAL NETWORK
Nijolė Maknickienė, Aleksandras Vytautas Rutkauskas, Algirdas Maknickas

9-15 MODELING OF BUSINESS PROCESSES. 1. OVERVIEW OF MODELS AND METAMODELS
Sigitas Vilkelis

16-21 MODELING OF BUSINESS PROCESSES. 2. PROTOTYPES OF TRANSFORMATION
Sigitas Vilkelis

Section: IT - applied solutions

22-26 CROWN: APPLIED TOOL FOR CARS-TO-RAMAN SPECTRUM DECOMPOSITION
Ildar Galiko, Alytis Grudis
Investigation of financial market prediction by recurrent neural network

Nijolė Maknickienė1, a, Aleksandras Vytautas Rutkauskas1, Algirdas Maknickas2
1 Faculty of Business Management, Vilnius Gediminas Technical University, Sauletekio al. 11, Vilnius, Lithuania
2 Faculty of Fundamental Sciences, Vilnius Gediminas Technical University, Saulėtekio al. 11, Vilnius, Lithuania

Received 17 August 2011, accepted 17 October 2011

Abstract. Recurrent neural networks as fundamentally different neural network from feed-forward architectures was investigated for modelling of non linear behaviour of financial markets. Recurrent neural networks could be configured with the correct choice of parameters such as the number of neurons, the number of epochs, the amount of data and their relationship with the training data for predictions of financial markets. By exploring of learning and forecasting of the recurrent neural networks is observed the same effect: better learning, which often is described by the root mean square error does not guarantee a better prediction. There are such a recurrent neural networks settings where the best results of non linear time series forecasting could be obtained. New method of orthogonal input data was proposed, which improve process of EVOLINO RNN learning and forecasting.


Keywords: Financial Forecasting and Simulation; Time-Series Models; Dynamic Quantile Regressions; Dynamic Treatment Models; Forecasting Models; Simulation Methods; Neural Networks and Related Topics; Recurrent Neural Networks; EVOLINO learning algorithm; Non linear time series; Orthogonal inputs; Prediction of financial markets.

JEL: G17; C32; C53; C45.

Short title: Investigation of financial market prediction by RNN.

Introduction

Modeling of non linear processes is actual in two aspects. Realistic models of their history helps to understand the inner structure of nonlinearity. On the other hand, correct understanding of nonlinearity improves the prediction of these processes. For this purpose researchers use the most common Mackey-Glass system [1] as standards tester of non linear processes. The best recognized tools for the finance currency markets is in the last decade neural networks [2-4] or by Reinforcement-Learning Agents [5-6]. Intensive researches of financial market data by neural networks shows that the best learning stage of neural networks does not always lead to correct forecasting.

Financial markets could be explained by using profitability, reliability or risk investment models and analysis methods. Opposite to statistical analysis there could be sophisticated reinforcement learning agents [6] or neural networks [2-4]. The best suited neural networks for the recursive nonlinearity are Recurrent Neural Networks (RNN). Behavior of time series in financial, stock or currency markets are influenced by psychology of trades and are strongly non linear and hardly predictable. Using the RNN in modeling of financial time series is based on founding an acceptable learning model for RNN’s. RNN’s are fundamentally different from feedforward architectures in the sense that they not only operate on an input space but also on an internal state space [7-8]. For the better improvement of RNN learning the EVOLINO algorithm [1] could be selected because it very clearly shows training and validation of the recurrent neural network for non linear data inputs.

The goal of present works is to understand how RNN works for modeling and prediction of the financial markets, their behavioral analysis and paying attention to the acceptance of the chosen method for the anticipation. Author of Ref. [9] was exploring the human mind distinguished reproductive thinking, which only echoes the familiar issues, and productive thinking, that creates something new. In order to solve a specific task certain knowledge is needed.

Meanwhile, not everybody, who has the knowledge required for the task, is able to use it productively. There is no direct link between the past experience and new (productive) thinking. By observing the RNN learning and forecasting the same effect is observed - better learning, which is described by the Root Mean Square Error (RMSE), does not guarantee a better prediction.

The aims of this paper are to find the best conditions, where EVOLINO RNN becomes a good instrument of financial markets prediction. It will be investigated collection of RNN parameters like number of epochs, number of neurons to achieve strong learning of RNN and good prediction of financial markets. The input orthogonalization method is proposed attaining this goal.

aCorresponding author, cell: +370 (687) 25206; email: nijole.maknickiene@vgtu.lt
1. EVOLINO Learning Algorithm. Description

Neural networks aid to monitor of the non linear processes in the learning activity. The comparison of various methods evaluates neural networks learning algorithms of non linear processes and increase their prediction accuracy. Schmidhuber et al. [10] introduced a general framework of sequence learning algorithm Evolution of recurrent systems with Linear Near Outputs (EVOLINO) [1]. EVOLINO uses evolution to discover good RNN hidden node weights, while using the methods such as linear regression or quadratic programming to compute optimal linear mappings from the hidden state to the output.

When quadratic programming is used to maximize the margin, it is impossible to obtain the first evolutionary recurrent support vector machines. EVOLINO-based Long Short-Term Memory (LSTM) can solve tasks that Echo State nets cannot [1]. There was introduced a new class of recurrent, truly sequential SVM-like devices with internal adaptive states, trained by a novel method called Evolution of systems with Kernel-based outputs (EVOKE), an instance of the recent EVOLINO class of methods.

EVOKE evokes recurrent neural networks to detect and represent temporal dependencies while using quadratic programming/support vector regression and pseudo-inverse regression. EVOKE is the first SVM-based mechanism which knows how to classify a context-sensitive language. It also outperforms recent state-of-the-art gradient-based Recurrent Neural Networks (RNNs) on various time series prediction tasks. RNN learning is used for context-sensitive languages recognition and is a difficult and often increasing problem for standard RNNs, because it requires unlimited memory resources.

For these array of problems investigated by authors of Ref. [1], [10-12], EVOLINO based LSTM learns in approximately 3 min on average and it is able to generalize substantially better that gradient-based LSTM. With EVOLINO it makes impossible to learn functions composed of multiple superimposed oscillators such as double sine and triple sine. Investigated network reached good learning and still makes very accurate predictions [1], [10-12]. The Mackey-Glass system is a standard benchmark for non linear time series prediction. Authors of Ref. [1] show deviation between the curves of EVOLINO generated and Mackey-glass system. EVOLINO is capable of making precise (0.0019) prediction in tasks like the Mackey-Glass benchmark.

The block diagram of EVOLINO recurrent neural network is shown in Fig 1. EVOLINO RNN forms LSTM network with $N = 4n$ memory cells, where $N$ is total amount of neurons and $n$ is amount of memory cells. The genetic evolution algorithm is applied to each quartet of memory cells separately. The cell has an internal state $S$ together with a forget gate ($G_F$) that determines how much the state is attenuated at each time step. The input gate ($G_I$) controls access to the cell by the external inputs that are summed into the $\Sigma$ unit, and the output gate ($G_O$) controls when and how much the cell fires.

Dark blue nodes represent the multiplication function and the linear regression Moore-Penrose pseudo-inverse method used to compute the output (light blue circle) [13]. The detail description of EVOLINO RNN algorithm could be found in Ref. [1], [10].

2. Inputs of Recurrent Neural Network

Authors of Ref. [14] analyzed what inputs should be chosen to receive the best models and got the inputs allowing a better prediction. They have found, that it is common to use the orthogonal inputs, where orthogonality of inputs is equivalent to orthogonality of $n$ dimension vectors. The orthogonality of vectors is result of following inner product for two vectors $f$ and $g$:

$$\langle f, g \rangle_w = \sum_n f(n) \ast g(n) \ast w(n)$$

where $w(n)$ is a non-negative weight vector of the definition of inner product. These vectors are orthogonal if above described inner product is zero:

$$\sum_n f(n) \ast g(n) \ast w(n) = 0$$

This method was selected for finding the best inputs in EVOLINO learning process too. The most tools of financial prediction are used for searching dependences between time series of financial indicators or similar patterns in these time series. For this purpose the time series orthogonalization were exploit as follows

$$| \sum_n f(n) \ast g(n) | = \varepsilon.$$  

where absolute value of scalar multiplication of vectors $\varepsilon$ describes degree of orthogonality, because true orthogonality (2) could not be reached for time series of financial markets and non-negative weight vector is $w(n) = 1$. Prediction of one time series output were obtained by the two most orthogonal time series inputs. The influence of data orthogonality were investigated in the range of $\varepsilon \in [0.00001 \div 0.001]$.
3. Reproductive and productive learning

Behaviour of human brains could be divided into a productive and reproductive thinking. Reproductive thinking only echoes acquired knowledge and productive thinking creates something new. These brain processes are not straightforward. RNN algorithm ability to learn non linear process is measured and evaluated by RMSE aid. By studying non linear processes, such as a stock or currency markets or fluctuations in solar activity, or others, the RNN prediction is very important research in today areas. It could be distinguished into two major aspects of forecasting: i) how many correct data points of the process could be predicted; ii) how many correct directions of the process could be predicted. If we set itself the objective of accurately predicting of values of the non linear process, we will be facing the problem, as the average value of deviation is acceptable. For such processes, as the shares of a stock or currency market, the prediction of the direction is sufficient for the making the reasonable decision of the future investment. This work will attempt to prove or reject that. The proving or rejection will be confirmed in the investigation of RNN EVOLINO algorithm with finding of parameters where the prediction is the best.

4. Criteria learning and predictions

For investigation of Root Mean Square Error (RMSE) learning and prediction processes we used program framework [15] adopted for multiple inputs. We selected two parameters - RMSE and correlation for comparison of learned and predicted time series. RMSE is often used in RNN as a learning criteria. Learning of RNN means tests of trained neural network, where data for tests were used after wash out of 1/3 start data of training data. Observation of good RMSE result for learning some times do not imply good forecasting. For this purpose we use correlation coefficient too. As it was mentioned above, moving direction of stock shares or currency ratios is more important than the prediction of their exact values.

The correlation coefficient is located in range \((-1) \div (+1)\). A value of (+1) implies that a linear equation describes the relationship between data ranges and predictive ranges perfectly, with all data points lying on a line in which Y increases as X increases. A value of (-1) implies that all data points lie on a line in which Y decreases as X increases.

5. Results and discussion

5.1. Selection of right numbers of epochs

Investigating some certain phenomena of artificial neural networks to work as memory structure or as a predictor is very important to clarify the behavior of the RNN.

It is very important to have not only a well-selected input data and training ranges, but also a good selection of epochs and the number of neural network. Epochs number describes the number of times when NN data are processed, and may wrongly appear that the higher the number of epochs leads to the better learning and prediction. Finally, it was studied the RMSE dependence on the number of epochs, taking the familiar orthogonal data ranges of currency market USD/JPY with using of XAU/USD as a additional input for improving of convergence.

The obtained results of learning RMSE are taken in Table 1. Results shows that a small number of epochs does not provide RNN learning. Only 76 epochs starts to learn RNN. RMSE dependence on epochs shows that after reached of 164 epochs learning is stabilizing and further increasing of the epochs does not make sense.

5.2. The importance of the number of neurons

The variance of the number of neurons is very important neural networks parameter in the RNN learning. At first glance it may seem that the more neurons are used the better the prediction result will be. But a large amount of neurons took more calculation time and it is important too. Therefore, it is necessary to find optimal number of neurons, which are able to learn and predict data of time series. The results of dependence of RMSE and correlation coefficient from the number of neurons were obtained with of data 85 points of USD/JPY with additional input of XAU/USD.

The studying the dependences of the number of neurons on learning RMSE could be seen that learning slowly increases, when the number of neurons increases from 16 to 64 neurons. Starting from 68 neurons learning suddenly increases 26 times and the behavior of RNN learning becomes excellent. Investigation of the dependences the number of neurons on the RMSE and correlation coefficients of prediction could be shown in Table 2.

The three areas of distinct neural networks amount have been found.

---

Table 1. Dependence of test RMSE of learning from the number of epochs with data of USD/JPY and Gold

<table>
<thead>
<tr>
<th>Number of epochs</th>
<th>RMSE of learned RNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>-0.008298</td>
</tr>
<tr>
<td>32</td>
<td>-0.007691</td>
</tr>
<tr>
<td>64</td>
<td>-0.007232</td>
</tr>
<tr>
<td>72</td>
<td>-0.006012</td>
</tr>
<tr>
<td>76</td>
<td>-0.002237</td>
</tr>
<tr>
<td>80</td>
<td>-0.002664</td>
</tr>
<tr>
<td>120</td>
<td>-0.002270</td>
</tr>
<tr>
<td>164</td>
<td>-0.001667</td>
</tr>
<tr>
<td>172</td>
<td>-0.001652</td>
</tr>
<tr>
<td>188</td>
<td>-0.001523</td>
</tr>
<tr>
<td>200</td>
<td>-0.001824</td>
</tr>
<tr>
<td>220</td>
<td>-0.001671</td>
</tr>
</tbody>
</table>
Table 2. Dependence of RMSE and correlation coefficient from the number of neurons.

<table>
<thead>
<tr>
<th>Number of neurons</th>
<th>RMSE of learning</th>
<th>RMSE of prediction</th>
<th>Correlation of prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>-0.009911</td>
<td>1.207037</td>
<td>-0.275067</td>
</tr>
<tr>
<td>20</td>
<td>-0.010139</td>
<td>0.269538</td>
<td>-0.334100</td>
</tr>
<tr>
<td>24</td>
<td>-0.006452</td>
<td>0.132038</td>
<td>0.000400</td>
</tr>
<tr>
<td>28</td>
<td>-0.005201</td>
<td>0.229054</td>
<td>0.053900</td>
</tr>
<tr>
<td>32</td>
<td>-0.003299</td>
<td>0.478200</td>
<td>0.011100</td>
</tr>
<tr>
<td>36</td>
<td>-0.002659</td>
<td>0.190520</td>
<td>0.035680</td>
</tr>
<tr>
<td>40</td>
<td>-0.002075</td>
<td>0.403500</td>
<td>0.184900</td>
</tr>
<tr>
<td>44</td>
<td>-0.001483</td>
<td>0.419292</td>
<td>0.12925</td>
</tr>
<tr>
<td>48</td>
<td>-0.000941</td>
<td>0.258178</td>
<td>0.399926</td>
</tr>
<tr>
<td>52</td>
<td>-0.000354</td>
<td>0.363497</td>
<td>0.157300</td>
</tr>
<tr>
<td>56</td>
<td>-0.000110</td>
<td>0.560873</td>
<td>0.042367</td>
</tr>
<tr>
<td>60</td>
<td>-0.005167</td>
<td>0.945073</td>
<td>0.015375</td>
</tr>
<tr>
<td>64</td>
<td>-1.600e-29</td>
<td>0.627907</td>
<td>0.068900</td>
</tr>
<tr>
<td>68</td>
<td>-1.350e-29</td>
<td>0.533008</td>
<td>0.383733</td>
</tr>
<tr>
<td>72</td>
<td>-5.587e-30</td>
<td>0.312393</td>
<td>0.224267</td>
</tr>
<tr>
<td>76</td>
<td>-7.252e-30</td>
<td>0.665393</td>
<td>0.272200</td>
</tr>
<tr>
<td>80</td>
<td>-1.450e-30</td>
<td>0.481330</td>
<td>0.227167</td>
</tr>
<tr>
<td>84</td>
<td>-2.300e-31</td>
<td>0.882620</td>
<td>0.292600</td>
</tr>
</tbody>
</table>

The first area is for numbers of neurons from 16 to 40 where averages of correlation coefficients are in interval $[-1 \div 0.1]$. This proves that there are not enough neurons in RNN to learn and to predict. The second area for number of neurons is from 52 to 56 where averages of correlation coefficients are in interval $[0.1 \div 1]$, this proves that RNN try to learn and predict the data. All values of correlation coefficient in this area are in interval $[0 \div 1]$, this proves that RNN predict directions of USD/JPY very well.

Third area for numbers of neurons is from 60 to 100 where averages of correlation coefficients fit interval $[-1 \div 0.1]$, this proves that increasing of the number of neurons improves learning and RMSE of learning, but suddenly RNN stop to predict. Correlation coefficients versus amount of neurons are presented in Fig. 2. Presented curves shows, that a zone of amount of neurons exists where maximum correlation could be achieved.

The similar results are obtained for five and for ten points prediction.

5.3. Variation of data amount

The last stage of investigation was the variance of the input data size. It was important to know how many days are sufficient to monitor the financial or foreign exchange market in order to obtain reliable forecasting using RNN. In this purpose dependence the number of data on RMSE of learning, RMSE of predicting and correlation of prediction were obtained. Study of dependences the number of data to the RMSE and correlation coefficients and finding of suitable, predictable RNN shows that the RNN behave in the same way as in previous investigations. There are three distinct neural networks areas in the number of data: under learned, best learned and over learned. Dependence the number of data on learning and prediction RMSE and correlation coefficients for USD/JPY and XAU/USD inputs and USD/JPY output is presented in Table 3. Three kinds of behavior of learning and prediction are given in Fig. 3.

The first area could be separated in which there is not enough data for RNN learning and prediction. The second area of numbers of input data is the best area for RNN learning and prediction of input data. The third area showed that increasing of the number of data improves learning and RMSE of learning, but RNN stop to predict and the further increase of number of neurons do not imply better prediction.

All three studies have shown that the RNN prediction could be obtained when the neural network parameters such as epochs, number of neurons and the number of data are in a certain range.

Table 3. Dependence on learning and prediction RMSE and correlation coefficients from the number of data

<table>
<thead>
<tr>
<th>Number of data</th>
<th>Number of neurons</th>
<th>RMSE of learning</th>
<th>RMSE of prediction</th>
<th>Correlation of prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>36</td>
<td>-0.000054</td>
<td>0.09935</td>
<td>0.4528</td>
</tr>
<tr>
<td>57</td>
<td>36</td>
<td>-0.000415</td>
<td>0.29459</td>
<td>0.5328</td>
</tr>
<tr>
<td>65</td>
<td>36</td>
<td>-0.001570</td>
<td>0.2077</td>
<td>0.7571</td>
</tr>
<tr>
<td>70</td>
<td>36</td>
<td>-0.002385</td>
<td>0.29763</td>
<td>0.9103</td>
</tr>
<tr>
<td>76</td>
<td>36</td>
<td>-0.002808</td>
<td>0.07895</td>
<td>0.1006</td>
</tr>
<tr>
<td>85</td>
<td>36</td>
<td>-0.001714</td>
<td>0.1286</td>
<td>0.8726</td>
</tr>
<tr>
<td>85</td>
<td>64</td>
<td>-0.007561</td>
<td>0.09726</td>
<td>0.3360</td>
</tr>
<tr>
<td>90</td>
<td>64</td>
<td>-0.001452</td>
<td>0.1564</td>
<td>-0.0286</td>
</tr>
<tr>
<td>95</td>
<td>64</td>
<td>-0.000714</td>
<td>0.18421</td>
<td>-0.7302</td>
</tr>
<tr>
<td>100</td>
<td>64</td>
<td>-0.000990</td>
<td>0.14608</td>
<td>-0.7320</td>
</tr>
<tr>
<td>105</td>
<td>64</td>
<td>-0.001245</td>
<td>0.3206</td>
<td>0.6198</td>
</tr>
<tr>
<td>110</td>
<td>64</td>
<td>-0.001777</td>
<td>0.23879</td>
<td>0.7114</td>
</tr>
<tr>
<td>115</td>
<td>64</td>
<td>-0.002061</td>
<td>0.2093</td>
<td>-0.2290</td>
</tr>
<tr>
<td>120</td>
<td>64</td>
<td>-0.004170</td>
<td>0.25555</td>
<td>-0.5788</td>
</tr>
<tr>
<td>125</td>
<td>64</td>
<td>-0.003538</td>
<td>0.16155</td>
<td>-0.7229</td>
</tr>
<tr>
<td>130</td>
<td>64</td>
<td>-0.003216</td>
<td>0.14268</td>
<td>-0.8211</td>
</tr>
</tbody>
</table>
Our studies have shown the most 164 epochs is enough. Number of neurons must be in interval $[52 \div 56]$ and number of data in intervals $[80 \div 85]$ or $[105 \div 110]$. The averaged correlation coefficient of forecasting for this range of initial parameters of RNN were reached value of 0.32 and, in separate studies, even 0.9579.

6. Conclusions

The aim of the presented work was finding of the best conditions where the RNN makes the best prediction of currency markets. It was investigated that the prediction of the evolution of the USD/JPY exchange daily rates for 15 March 2010 in 5 following days. Data were collected from 1 Januar 2009 till 15 March 2010. USD/JPY exchange rates trained for the same period of XAU/USD data inputs. The obtained results show that.

1. The lowest values of orthogonality degree description coefficient $\varepsilon$ improve stability of RNN learning and prediction of investigated non linear time series. The confirmation of quantitative dependences needs future investigation.

2. The learning and prediction of RNN, like the human brain productive and reproductive thinking, are independent and different. The better RMSE of learning do not guarantee the better achieving of prediction.

3. Combinations of parameters of RNN such as the number of epochs, data and neurons amount, determine different behavior of learning and prediction. weak learning without prediction; strong learning with prediction; excellent learning without prediction.

4. The investigation of financial data gives such group of parameters of RNN in EVOLINO algorithm, were RNN predict directions and values of a currency market. The group of RNN parameters for given data was found where the average of correlation coefficient of forecasting reaches maximum 0.938 and has value equal to 0.400.

References


Modeling of business processes.
1. Overview of models and metamodels

Sigitas Vilkelis
Department of Informatics, Faculty of Mathematics and Informatics
Vilnius University, Naugarduko 9, Vilnius, Lithuania

Received 28 June 2011, accepted 10 September 2011

Abstract. Different aspects of business process modeling are observed and table structure describing business processes was formalized. Methods and categories of modeling, as well as terminology, existing techniques and tools for Model Driven Architecture (MDA) are revised and possibilities of their application in transformation of business process models are evaluated.


Keywords: Model Driven Architecture; MDA.
JEL: C88; M15.
Short title: Overview of models - 1.

Introduction

Nowadays, under the influence of IT on business, it is difficult to imagine a successful but completely not computerized business. The influence of IT on computerization of specific business processes is essential and has many forms - from digital data storage to automatization of difficult multi-user business processes. Creation of business models is very important in keeping up with the advancing technologies. They are necessary for a change of system platform. In order to create a successful model, specific business terminology has to be known along with the software terminology. The resulting model is a combination of two different subjects: the business terminology and the abilities of business process modeling tools.

This is the reason why business models are usually created by two groups of people: the ones employed in a specific area who know the details well and programmers creating the software system to implementing the business model. Business analysts receive the requirements of initial business processes from customers and pass them to programmers.

Although the phase of requirement clarification employs several people, a fixed business process description structure is required. It is needed to be able to look for similar processes in the same structure when creating models for other customers as well as enabling formal documentation and creation of requirements that should be met.

The technologies have been advancing rapidly, but customers still use simple tables, filling the forms by type or even by hand. Models created this way have much higher error probability and it takes additional time converting them to modern standards processed by a computer. Switching to structures defined by the modern standards would reduce the workload for everyone - from analysts, communicating with customers, to programmers, doing all the software package creation for the same customer. Due to the customer’s habit of using tables, there is a need for special instruments designed to convert these tables to processes understandable by computers.

This work is devoted to the literature review containing business process modeling and its application of architecture related to these models. Methods and categories of modeling as well as terminology, existing techniques and tools for Model Driven Architecture (MDA) are revised and possibilities of their application in transformation of business process models are evaluated.

1. General formulation of a task

According to the experience of the author, the following problems occur when using table modeling.
1. A process modeled using tables will not avoid making a lot of errors, until actual programming takes place.
2. Such process may not be added to running business management systems.
3. Logical business process errors are difficult to notice in a spreadsheet description.

The first problem may be solved in three steps. First of all,
the structure of a table should be defined. Secondly, rules should be set to define the formal table structure and finally, these rules should be written in formal form.

The solution to second problem may be expressed as a creation of a process to transform table-based business processes to ones using standard notation. This enables easier manipulation with these processes in management of software systems. The third problem may be solved by applying the same method as in the second problem. The transformed process may be represented graphically (as two-dimensional or three-dimensional distribution), which would help to avoid logical errors.

These are the general problems occurring in process modeling. By clarifying these problems, the following steps are pointed out for this work.

A. Define the business processes described in the tables.
B. Create a process of transformation of table-based business process to formal business process notation.
C. Create a structure of rules and define the rules that processes described in tables should follow.
D. Demonstrate the process of rules and transformation inspection by releasing a prototype of business process transformation and verification.

In order to achieve the A-task, partial sub-tasks were done.
A1. Analysis of existing business process notations was done and a common set of elements was taken from them.
A2. According to the resulting list, formal structure was made and fields of data as well as data types within fields were described.

The steps to complete the B-task are presented below.
B1. Architectural ideas based on models for transformation of table-based business processes to formal business process notation were revised and applied.
B2. Rules of transformation were defined.

C-task was completed by performing the following steps.
C1. A set of rules to be used by table-based business process was defined.
C2. This set of rules was applied to formal table-based business processes.

Actions taken to accomplish the D-task were described in three positions.
D1. According to the business modeling at a workplace of the author and analysis of References, entirety of technological tools for practical solution was made.
D2. A tool from a selected technological environment was chosen to ensure the transformation of table-based business process to formal business process notation.
D3. The selected tool was extended with a rule inspection. The specific steps and their execution are described in following chapters.

2. Methods and notations. Review

Business processes may be modeled using various methods, such as mathematical, chart plotting tools or languages defining business processes. Fig. 1 represents the classification scheme of a business process modeling.

As we can see in the Fig. 1, strict technique separation into three categories is absent, because, some of them have common features. All categories are reviewed below.

2.1. Categories of business process modeling

Mathematical Methods. Mathematical methods are based on strict formalism which results in models which are completely accurate and satisfying all requirements. These models have fewer errors and may be verified using formal methods. Nevertheless, it is inconvenient to create business models using mathematical methods, because advanced mathematical, logical and field-specific knowledge is required.

Chart-based models. The biggest advantage over mathematical methods to chart methods is that they are defined not in mathematical formula, but in graphical charts. This is very important feature, when there is a need to present it to a customer. The business processes are less formal this way, but much more popular.

Languages of business processes. The most popular models are based on languages of business processes. The popularity is based on the applicability of it - all languages can be represented graphically and have their unique document saving format. Graphical representation is important for understanding the process and the possibility to save them provides a possibility to exchange the documents between different tools.

2.2. Description of business process modeling

Modeling of business processes could be performed using specific tools (created for business process modeling only),
although there are some tools that are general modeling tools with extensions enabling business process modeling. Most of the methods exist as standard tools, such as Object Management Group (OMG), Unified Modeling Language (UML), Business Process Modeling Notation (BPMN), XML Process Definition Language (XPDL) [2].

BPEL4WS. Modeling standard BPEL4WS was created by unified forces of BEA, IBM, Microsoft and other companies. The purpose of it is describing of processes provided on internet and enables creation of difficult business processes, by combining several independent actions to one set of jobs. Such combination is suitable for business process modeling and it is based on Web Service Definition Language (WSDL), so different web services exchange XML documents. Service oriented (SO) business process modeling has its own advantages over methods using strict process arrangement. They are: flexibility and easiness of changes [3].

BPMN. BPMN is a standard created by Business Process Management Initiative (BPMI), describing Business Process Diagram (BPD) based on diagram creation technique used in graphical representation of business process models. This standard is very popular amongst analysts as well as programmers, because it puts the client needs in a simple way, but it has only the graphical notation, lacking formal description of business process. BPMN documentation [4] also lacks definition of how should the graphical data be stored in a way that would be understandable by computer. This creates an uncertainty of a format used, because every tool uses its own data format (incompatibility could occur). Although, there are some tools that use XPDL’s XML documents [5].

There are several important notation problems of BPMN explored in Ref. [6]. This notation has nothing in common with representation of user interface of business process in program system and there is no connection between the business process and the modeled field. The authors of Ref. [6] found a solution to both of these problems. Using rules of transformation, a problem of transforming BPMN diagrams into YAWL diagrams is analyzed in Refs. [7-8]. Working tool is able to perform such transformation, which also shows, that BPMN notation described formally may be transformed into any other formal structure.

UML. UML is one of the most popular program systems and business process modeling languages. It defines many different diagrams made for objective system modeling, and diagrams such as activity enable modeling of fully-fledged business process. Business processes may be defined as an oriented graph, made of peaks and bows using UML. Peaks represent the performed single or combined activity. Peaks may be used for execution control as well. Start and end peaks are assigned to execution control peaks, marking the beginning and the end of a process. Connection and branching in UML may be used to model separation or addition of several parallel processes [2].

UML sequence diagram can be used to show exchanging the information between people participating in business process. Extended activity diagram is suitable for creation of a model describing communication between people [9]. Modeling of variability using UML is presented in Ref. [10]. Common fields where changes are usually made are presented in this paper, these are the changes during the business process, data transmission sequence or activities of business process. It is said, that the description of business process should not go into details, but instead represents only the general sequence of actions making space for actions that could possibly change over time. Nevertheless, this method can only be theoretical, because it is essential to specify common input and output points, sequence of activities, even the exact processed data in practical business process.

The information provided by Ref. [11] describing the transformation of UML business process to XPDL business processes is very beneficial. The general idea is to fill in the missing data in UML diagrams and transform them by using XSLT to XPDL. In order to have XPDL documents fully structured, UML diagrams are filled with extra stereotypes.

XPDL. XPDL is a language of business process description introduced by Workflow Management Coalition (WfMC) to define a general data exchange form and supporting moving of different process description between different tools. The purpose of XPDL 2.0 is description of business processes presented in BPMN graphical notation.

Petri net. Petri net is a way of business process modeling using mathematical methods and graphical imaging. It consists of places, transitions and arcs. Places may be marked and moved to other places by following the rules. It is very convenient to describe and analyze parallel, asynchronous or distributed systems. As a graphical tool, Petri net can be used to represent graphical connections similar to sequence diagrams. Simulation of dynamic and parallel systems is also performed by using bookmarks in these Petri nets.

The example of business process modeling is presented in Ref. [12]: the basic of modeling is based on two components - activities and resources. Specific resources, such as human labor, specific data or even internet services, are required for these activities. The biggest benefit of using Petri net is the scalability of models: using several layer modeling, even the smallest processes may be modeled and combined to larger ones, thus creating a clear and detailed business model.

Another similar sample of modeling is described in Ref. [13]. The main difference between the previous one is the usage of several Petri nets: ontological, based on abstract understanding of business process, concept, introducing the
business transaction term and system interfaces and functional net describing interfaces, services and data streams. This type of model enables the evaluation of a business process from three different angles, which proves useful when there is a need to confirm the logic of a process.

**IDEF0 and IDEF3.** The purpose of *Integration Definition for Functional Modeling* (IDEF0) is functional modeling based on usage of text and graphical markings on organized and systematic models. This increases the understanding of project and integration activities as well as defines the requirements.

Method called *Integrated DEFINition for Process Description Capture* (IDEF3) defines the collection and documentation of processes. It incorporates easy-to-understand priorities and connections between actions.

**VPML-S.** A new graphical business process modeling language *Service-Oriented Visual Business Process Modeling Language* (VPML-S) was created as a language based on UML, extended with stereotypes [14]. Its purpose is modeling of service-based business processes. The main goal was to create a language that would have a decent graphical notation and would not require specific IT knowledge in order to use it. A business process written in this language is fully compatible with BPEL language, which defines business process as a septum: activities, products, resources, connections, events, attributes, partners. Every part of the septum is strictly defined using mathematical methods. It may be stated that it can be used to model business processes because it supports internet services, although there are a couple of problems - there may be a lack of support for the language and poor availability of tools enabling modeling in this language, because it is fairly new, written in 2008 and designed for academic purposes.

**Existing Activity Diagram.** Modeling of a Business Process could be based on *Existing Activity Diagram*. If a business has documentation of its activities as activity diagrams, they may be transformed to business processes as if reusing them [15]. Such modeling of business processes saves time on analysis and documentation of existing business processes. It may also increase the quality of business processes and reduce the error probability. The information on usage of such diagrams should be retrieved at first, in order to know whether they are still valid: for example, if a company kept records of such diagrams for their first year, but discontinued afterwards, there is no use for these diagrams.

**JBPM and JPDL.** *JBoss Business Process Management* (JBPM) is a management system, filling the gap between analysts and programmers. It is flexible and provides a way of process modeling, suitable for both of these groups.

Input data for JBPM is presented as descriptions of graphical business processes. The process represents a sequence of actions that are defined as transitions from one activity to another. These graphical diagrams of business processes are the basic way of communication between analysts and programmers.

JBPM as JBPM Process Definition Language is based on *Process Virtual Machine*, which is able to support several languages devoted to the business process definition. JPDL is currently the basic language, created by a business itself. JPDL is a flexible language with extension possibilities, which, according to experience of author, enables easy implementation of JBPM JPDL processes to active systems [16].

### 3. Transformation technique

#### 3.1. Models and meta-models

The amount of tools and techniques for digitization of businesses has been growing constantly as well as the amount of digitized businesses themselves. One of the systems was based on creation of a model for every step describing it with a required level of detail. This method prevents creation of many documents and allows transformation of the model to a certain software. Such technique was named *Model Driven Software Development* (MDSD). Traditional cycle of software creation is shown in Fig. 2, and extended cycle of *Model Driven Architecture* (MDA) based software creation is shown in Fig. 3. Several software development areas are based on MDA. One of them is called *Model Driven Engineering* - MDE.
MDA is developed by OMG group and it specializes not only in software development but in separating logic of business and software from a specific technology in software. A digitized business solution defined by Platform Independent Model (PIM) based on UML and other OMG standards with MDA ideas may be implemented in any specific platform using web services, such as .NET, J2EE and others. PIM models define a specific software functionality required by business separately from a software based on specific technology. That way technological restrictions are avoided and moving to another technological environment freely is encouraged. Platform Specific Models (PSM) are derived from PIM models and then transformed to a software supported by specific PSM environment [18-21]. Relation between PIM and PSM is shown in Fig. 4.

One of the general properties of MDA is transformation of models. During one transformation, PIM is joined with additional information and PSM is generated. During another transformation, a realization of software is generated using mapping method. A realization of specific transformation depends on the software system. A kind of transformations exists when models written in PIM language are transformed to models written in PSM language. PIM and PSM meta-models and rules of transformation are defined to enable such process. This transformation is performed between two PIM and PSM models with specific values. Graphical representation of this process is shown in Fig. 5.

3.2. Types of model transformations

Model-to-Model (M2M) transformations have been presented previously [17, 19]. There is another way of transformations - the so-called Model-to-Text (M2T) transformation. It converts a model to any text: from software code to documents of any format. A lot of various tools can be either commercial or open-source. Both types will be reviewed.

Open-Source Model Transformation Tools. Kermeta package was created by INRIA Triskell [20]. It is based on Eclipse platform and the environment is of object-oriented type. The purpose of it is describing and transforming of models and meta-models, as well as their simulations. Kermeta is created as an extension to Eclipse Modeling Framework (EMF).

MOFScript is a tool for M2T transformations based on EMF as well. Its purpose is transformation of models and metamodels based on Meta-Object Facility (MOF).

The IBM Model Transformation Framework (MTF) - is a tool for describing relations between meta-models in QVT and it is based on EMF as well.

The ATL Engine - a language similar to QVT written by INRIA Atlas. It is one of the most important technologies in Eclipse M2M project, created as a bunch of add-ons and it works as a built-in programming language to perform, describe and trace transformations between models [22].

OpenArchitectureWare (oAW) - a flexible framework working along XMI and based on templates.

Generative Model Transformer (GMT) - an Eclipse project for a model transformation technology for Eclipse. Several current tools are a part of GMT: AMW (Model Weaving), Epsilon (Model Merging), MoDisco (Model Discovery), MOFScript (M2T), openArchitectureWare, UMLX (Graphical Transformation), VIATRA2 (Visual Automated Transformations).
**OpenMDX** - open-source MDA environment consisting of several tools based on XMI and supporting multi-platform (J2EE, .NET) code generation [21].

**Commercial Model Transformation Tools. **ArcStyler - a commercial MDA tool created by Interactive Objects. It is sold along with MagicDraw UML tool, but it supports other UML tools as well.

**Model Component Compiler** (MCC) - a commercial product of InferData supporting transformations of M2T to J2EE.

**Xactium XMF Mosaic** - a tool supporting M2M transformations.

**Model-in-Action** and MDA - a tool created by Mia software, that supports generation of software and M2M transformations on flexible framework.

**MetaEdit+** - built-in environment of modeling and metamodeling for creation of languages and source generation. It supports XML and SOAP/Webservice transformations for models and meta-models.

**MDWorkbench** - a tool supporting M2T and M2M transformations accepting any meta-model format as an input. It is based on Eclipse and EMF [22].

**Conclusions**

Business process modeling of author’s workplace was analyzed and table structure describing business processes was formalized. A process of transformation of a table-based business process to a standard description business process notation, based on MDA, was defined. Rules that have to be met by table-based business process were structured and described. Transformation and rule checking were implemented using Eclipse with standard and oAW plugins.

A successful prototype and process definition prove that objectives were completed successfully. The achieved results will improve and speed up work of several people: analysts will be able to check and see graphical business process while filling the business process description table. This will prevent logical errors that occur while creating a business process from separate tasks. Programmers will be able to retrieve the notation which are structured and depicted in specific form of business process. Transformations will be done directly from table-based business process. They will only need to fill some additional information and implement it to a running system. The transformation process using BPMN notation has obvious advantages - business processes described by this notation can be transformed to almost any other business process notation providing almost unlimited expandability for such process. This possibility enables company modeling their processes by tables to adjust their final result to used technologies or specific client requests without changing their initial table-based business process transformation to standard description business process.

**Acknowledgments**

This work was prepared as a part of the Master Thesis. Author would like to thank Prof. Rimantas Vaicekauskas (Vilnius university) for valuable comments and suggestions.

---

**References**

Modeling of business processes.
2. Prototypes of transformation

Sigitas Vilkelis a
Department of Informatics, Faculty of Mathematics and Informatics
Vilnius University, Naugarduko 9, Vilnius, Lithuania

Received 28 June 2011, accepted 10 September 2011

Abstract. Working method devoted to formal table-based business process transformation using Eclipse to formal business process notation is presented here. The resulting formal business process notation document may be used as a template in further programming jobs by implementing it to a working system.


Keywords: Model Driven Architecture; MDA; Eclipse.
JEL: C88; M15.
Short title: Prototypes of transformation - 2.

Introduction

Previous publication [1] is aimed to review the different aspects related to transformation between business processes. Methods and categories of modeling, as well as terminology, existing techniques and tools for Model Driven Architecture (MDA) were revised [2-4] and possibilities of their application in transformation of business process models are evaluated.

The basics of the business process transformation and requirements of expandability are defined, as well as the input and output data on every step of transformation. A choice of using indirect transformation, by adding a temporary business process notation, instead of direct transformation is explained.

A working method devoted to formal table-based business process transformation using Eclipse to formal business process notation is presented here. The resulting formal business process notation document may be used as a template in further programming jobs by implementing it to a working system.

1. MDA Tools for models

MDA is a new outlook to a software, implementing traditional models by using them as input or output data. The purpose of MDA is replacing traditional diagrams, such as UML [5] or ordinary text by models. This allows access to lower layer models by applying transformations, generating machine code at the lowest layer. All these transformations are defined by specific rules of transformation from one model to another [6].

MDA technique was used in this work to perform a transformation from a business process described in a table to formal business process notation.

According to author’s experience, analysts of business systems still use tables instead of standard approved techniques and methods when communicating with their customers to provide easier apprehension throughout computerization of a business. The objective of this work was created to make sure that transformation of structured tables to a digital format understood by computer is possible. This provides several ways to find the solution.

One of them is the transformation of documents described above using XSLT. It is not very convenient, because if the initial table structure changes, which may occur when analyst has to adjust it for different customers, the structure of XSLT transformation has to change too. This may be a problem while working with several customers at once as well, because different XSLT transformations are needed for every separate table.

Another way is to separate variable part of table-based business structure from standard description business process describing the resulting non-variable part. A transformation between table-based business structure and standard description process should be performed by defining meta-models describing it, because standard description process has to always remain unchanged, even if the table-based process changes. The most convenient representation of table-based process structure is a CSV document with a predefined row-column structure.

---

a Corresponding author, email: sigitassv@gmail.com
The standard description business process can be defined using BPMN and a transformation between CSV and BPMN meta-models may be performed [7].

The BPMN structure may not be applied to a system directly, therefore a transformation to a specific business process notation has to be performed. Fig. 1 represents the transformation process graphically.

A third way to solve the problem of transformations is saving a formal table-based business process as a CSV or any other document format capable of maintaining organized row-column structure. Every section of CSV document is separated by using a comma or semicolon. Several sections constitute a record which takes up one row. A table of business process description in such form may be analyzed and processed by using computer tools.

To avoid scenario presented in first solution, variable and non-variable parts have to be separated. The conversion to specific business process notation is performed after transformation of CSV document to standard business process notation, which is BPMN in this case. As the analysis of literature has shown, business processes in BPMN are much clearer and accessible with many software development tools. The notation itself has a wide choice of available elements ensuring proper conversion to a chosen specific business process notation.

The conversion between CSV and BPMN may be performed by using XSLT transformation as well, although a problem of supporting the platform appears. In order to avoid it, meta-models describing CSV and BPMN business processes should be defined. After that, transformation rules have to be introduced as well. Another part of the process is transformation of BPMN to a specific business process notation. They should be chosen according to the experience of maintenance personnel and current software running in a business they are going to be used on and to ease the implementation. The meta-model of specific business process notation, as well as the transformation rules have to be defined.

The process described above, enables the following:

i) the meta-models and transformation rules between BPMN and specific business process notations may remain constant when the business processes defined in CSV change;

ii) changing the transformation rules independently of CSV, BPMN or specific business process notation meta-models;

iii) transformation of business process defined in BPMN to any specific business process notation, so changing platforms may be performed keeping old and working business processes by transforming them to the new notation.

As we can see, the resulting specific business process notation is ready to implement along the modeling tools and technologies used in business, even if it is not completely full because of specifics of different notations. Still, the resulting notation may be a decent fundament for further implementations and it minimizes the amount of human labor for conversion of a table-based business process to a specific business process notation. A prototype resembling this transformation will be presented in next chapter.

2. A Prototype of Business Process Transformation

A prototype of transformation of formal table-based business process to a standard description business process was released by using a plentiful list of literature and the theoretical part revised above.

2.1. General Process of Transformation

A decision to perform an experiment of transformation of CSV document to a specific business process notation using a prototype was made. The specific business process notation was selected to be a notation based on JBPM JPDL standard used in author’s workplace. The graphic representation of the bonds between models and data are shown in Fig. 2.

It is based on M2M transformations which are described by standard and the oAW plugins of Eclipse [8-9]. The crucial transformation is between CSV Ecore model and the Ecore model of general BPMN elements used for other standards. When the requirements of the prototype were known, open-source Eclipse environment was chosen, supplemented with standard and oAW plugins:
i) EMF/Ecore was used for description and securing of meta-models;

ii) ATL - a plugin supporting transformation language enabling creation and execution of transformation rules [10];

iii) oAW Check - a tool for description of rules of meta-models defined by Ecore construction, enabling the confirmation whether a specific meta-model follows the rules.

The graphical environment of technology is shown in Fig. 3.

At first, the structure of CSV document was analyzed and Domain Specific Language (DSL) was defined using the Xtext plugin for Eclipse. The DSL was used to get an Ecore model, describing the structure of CSV document in terms of Eclipse EMF framework. Ecore is a meta-model of EMF framework, supporting saving of models in XMI format. XMI is a standard for XML Metadata Interchange between different systems created by OMG.

Because ATL supports M2M transformations between Ecore models, a requirement of having both models in Ecore format was made [11].

BPMN Ecore model was defined afterwards. In order to do that, a standard describing BPMN was analyzed and a set of usable elements was chosen. This set is a subset of all possible BPMN notation elements and an Ecore model was created for it.

Once two Ecore models were obtained, rules of transformation between CSV Ecore and BPMN Ecore models were defined using ATL tool. This was an important step of all conversion process confirming that formal table-based business process can be transformed to a BPMN. Afterwards, final step - transformation to a specific JBPM JPDL business process notation took place.

This transformation is required to make sure that the transformations are correct by comparing the result to other business processes used in author’s workplace. Such business process would be used in further programming and implementation work as well. Afterwards, analysis of the resulting process was performed and an Ecore model was created. Transformation rules had to be defined as well. The final result of this process is an XMI document describing the JBPM JPDL business process of author’s workplace. Graphical process of transformation is shown in Fig. 4.

A detailed overview on the process of transformation is shown next.

1. The sample business process is formed as a CSV document.
2. An ATL transformation is performed on it to get a document of XMI format corresponding to the CSV Ecore model.
3. Rules of checking CSV XMI document were formed by using oAW Check Constraints tool for Ecore models.
4. Using this tool, CSV XMI document is statically checked if the initial document has any errors.
5. Afterwards, the CSV XMI document is transformed to BPMN XMI document by using ATL transformation rules.

Fig. 2. Bonds between models and data used in transformation.

Fig. 3. Technological environment of prototype implementation.
Fig. 4. General transformation process realized in prototype.

The final transformation of BPMN XMI document to JBPM JPDL business process notation is performed, resulting in a XMI document of specific business process notation. A prototype based on Eclipse plugins was realized. The Ecore models and transformation rules used will be presented.

2.2. Ecore Model of CSV Document

A structure of text-based business processes was specified in order to describe a meta-model of CSV document. Formal analyzing of table structure must be done. After analysis of a table structure used for business process notations, a new, more simple and more convenient structure was created, which is shown in Fig. 5.

Fig. 5. Metamodel of table-based business process.

The description of fields of formal structure is shown in Table 1.

Rules of business process description table. The data presented in a table of the last chapter have to meet these conditions.

1. Task number field has to be unique for every process.
2. Task name has to be unique in a single process.
3. Task priority has to be between 1 and 5.
4. Every task except last has to have their initial or creation conditions.
5. Task end condition of the last task should not have any other tasks.

These are the basic and initial formal table-based business process rules. The advanced rules that need to be met by business process descriptions are presented below.

1. If a task is not the first one, its initial conditions have to be the same as the end condition of a previous task.
2. If there are several beginning tasks in a process, their initial conditions cannot overlap which means that several initial conditions cannot be met by the same set of data.
3. A process must have at least one first and last task.
4. The same task may not be used in several processes.
5. Initial (beginning) and final conditions may not overlap with one task.

As we can see, every rule is for initial or end conditions because the most of errors are found in these parts. With the help of successful implementation of verification of these parts, the time it takes to model these processes may be reduced significantly.

Table 1. Description of fields of the formal structure.

<table>
<thead>
<tr>
<th>Table field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task number</td>
<td>Unique task number for every process.</td>
</tr>
<tr>
<td>Name or number of a process</td>
<td>Grouping of tasks to a process.</td>
</tr>
<tr>
<td>Task name</td>
<td>Unique task name in a process.</td>
</tr>
<tr>
<td>Priority</td>
<td>Task execution priority, a number between 1 and 5.</td>
</tr>
<tr>
<td>Initial/task creation conditions</td>
<td>Conditions describing the creation of a task. Comparison of initial attributes with fixed values in a condition.</td>
</tr>
<tr>
<td>Task completion conditions</td>
<td>Conditions upon which the current task should be finished and another task created. If there are no more tasks, the process itself is finished.</td>
</tr>
<tr>
<td>Is there another task in queue? (Y/N)</td>
<td>Marks whether this is the last task of a process.</td>
</tr>
<tr>
<td>Is this task the first one? (Y/N)</td>
<td>Marks the first task of a process.</td>
</tr>
<tr>
<td>Expression of task asignment to a user</td>
<td>An existing username used to login to system is set as an expression value.</td>
</tr>
</tbody>
</table>
Vilkėlis. Prototypes of transformation - 2. 20

grammar org.xtext.example.\cvd\CsvDsl with org.eclipse.xtext.common.\Model {\string+=string} {\number_of_Task = ID';'} {\processName = STRING';'} {\taskName = STRING';'} {\contitionOfCreation = STRING';'} {\ExpressionOfDefinitionTo = STRING';'} {\runTime = STRING';'} {\priority = INT';'} {\FinalyngConditions = STRING';'} {\isTheLastTaskInProcess = ('Yes'|'No')';'} {\isTheFirstTaskInProcess = ('Yes'|'No')';'}

Fig. 6. Example of the textual business process DSL.

Meta-model of CSV Document. DSL of a business process described using Xtext plugin is shown in Fig. 6. Even complex business processes may be described in such simple structure. Fig. 7 represents an Ecore model created from this DSL using Eclipse plugin. This CSV document model will be used in further M2M transformation process, where this model will be input (source), and standard description business process notation will be transformation output (objective) model.

3. Ecore Model of BPMN Elements Subset

Fig. 8 represents a set of elements and its Ecore model resulted after performing BPMN analysis. BPMN Ecore model was defined using BPMN standard [12]. This BPMN Ecore model is used for semantic BPMN. It is obvious that there should be another model involving depicting of elements (fonts, font sizes, element colours etc.) in business process diagrams, because BPMN is a graphical business process notation. A BPMN plugin using Eclipse platform, creates two documents at once, when creating BPMN business process diagram. One of them is semantic model, while the other is a document containing graphical element parameters. Since graphical positioning of elements is not important, it will be left aside and only semantic model and its structure will be analyzed.

3.1. JBPM JPDL Business Process Notation Ecore Model

The final model transformation result is a JBPM JPDL business process notation document, recognizable by Eclipse plugins. Fig. 9 represents Ecore model describing the document structure for this transformation.

By performing M2M transformation on BPMN Ecore model, a resulting JBPM JPDL Ecore model with XMI document format is received and used in further programming stage for final implementation to current system. Rules of transformation predicting details of the process have to be defined. These details consist of representation of an element from one model to another.

Fig. 9. Ecore model of JBPM JPDL notation elements.
3.2. Transformation Rules

The Ecore models described earlier do not perform any functions themselves, so transformation rules for transformation between models have to be defined. The following M2M transformations were described by using ATL tool and its Eclipse plugin:

i) from CSV Ecore to BPMN Ecore models;
ii) from BPMN Ecore to JBPM JPDL notation Ecore models;

ATL tool enables definition of transformation rules between a source Ecore model and a objective Ecore model bonding elements between them. An example transformation rule for converting a business process name from CSV Ecore to BPMN Ecore model is shown in Fig. 10.

``` ATL
rule ProcessName {
from\s : CsvDsl!Line\nto\t : bpmn!BpmnDiagram {
\title <- s.processName
```

Fig. 10. An example of a transformation rule.

The transformation rules were defined and a JBPM JPDL business process notation XMI document was received. This experiment shows that a business process described by a formal structure table may be converted to a business process notation of standard description type.

Results and Conclusions

Successful achievement of goals and tests of prototype leads to following statements:

i) formal structure tables may be used instead of standard business process notations to describe business processes;
ii) formal structure tables can be transformed to standard description business process notations;
iii) created prototype enables automation of conversion between formal table business process and a standard description business process.

The created solutions and achieved results may be applied in any business where business processes are not modeled using standard notations. By applying presented ideas and structuring initial business process tables to a structured row-column structure, transformations to a standard business process notation may be performed.

The described solution may be improved by expanding business process table structure, as well as increasing amount of fields moved from it to a middle BPMN notation, as well as defining transformation rules for several business process notations instead of one. That way, business processes could be transformed to more business process notations, as well as the sketches received after transformations would be fuller.

Acknowledgments

This work was prepared as a part of the Master Thesis. Author would like to thank Prof. Rimantas Vaicekauskas (Vilnius university) for valuable comments and suggestions.

References

CROWN: Applied tool for CARS-to-Raman spectrum decomposition

Ildar Galikov a, Alytis Gruodis

1 Faculty of Physics, Vilnius University, Saulėtekio 9, korp. 3, Vilnius, Lithuania

Received 2 Februar 2011, accepted 10 May 2011

Abstract. Novel advanced tool CROWN was created for CARS spectrum decomposition purposes: in order to extract Gauss- or Lorentz-shaped components containing Raman spectrum. Win32 type application was created in object-oriented programming (OOP) manner using Visual Studio 8.0 package. C++ language was used for programming GDI interface using standard windows.h. library.


Keywords: CROWN; CARS; CARS-to-Raman; Spectrum decomposition.

PACS: 42.65.Dr

Short title: CROWN: CARS-to-Raman.

Introduction

Coherent anti-Stokes (AS) Raman scattering (CARS) technique [1] is popular as a unique macroscopic as well as microscopic [2] tool in material sciences [3], biology [4], biophysics [5], medicinal physics [6], etc. The benefit of CARS technique could be described as sensitivity to intramolecular changes and versatility due to χ(3) behaviour [7]. CARS signal according to the experimental schema is blue-shifted from laser excitation frequencies – it means that CARS signal could be easily detected in the presence of strong luminescence [8].

Visualization of the digital data containing spectral dependencies belongs to the one of the most important computer tasks in computational physics. Information graphics software allows to users the fast manipulation in order to create the suitable visualization form from equipment-provided data-set. Computer-algebra systems (such as Maple [9], MathCad [10] and Mathematica[11]), numerical-software systems (LabView [12], MatLab [13], Scilab [14]) are quite useful for creating the two-dimensional or three-dimensional space projection containing object-oriented dependencies.

However, sometimes all functional possibilities of mentioned packages are not partially or fully required due to specificity of the task. Reordering and pre-manipulation (including sorting, fitting etc.) – such non-trivial operations are possible using Origin [15] package which, however, requires many non-automated operations. Using of Origin tools - worksheets and data fitting wizards – complicates the fast search of solution, but is very useful for slowly routine operations only. This work is devoted to create the novel fast tool for express estimation of CARS and Raman spectrum content without necessity to use routine, time-wasting fitting procedure in the some wizards.

1. Physical behaviour of CARS spectrum

Project CROWN was created as original graphical user-friendly tool for fast Gauss/Lorenz-shaped decomposition of CARS spectra S(ω) into Raman components (CARS2Raman). Adapted formula from Ref. [16] was used as follows:

\[ S(\omega_{AS}) = |I_{BCG}(\omega_{AS}) + I_R(\omega_{AS}) \cdot \cos \phi|^2 \] (4)

I_R and I_{BCG} represent the terms of Raman bands and backgrounds, respectively. Due to experimental conditions, such two terms represent different origin (changes of induced dipole moment for I_R (pure nuclear movement) and changes of nuclear dipole moment in the fast electrostatic field for I_{BCG} (nuclear movement in the electron environment, respectively)). Factor \cos(\phi), when \phi=0 deg or \phi=180 deg, represents the increasing or decreasing of intensity S. Angle \phi represents the difference of phases between Raman term and background term.

Single Raman bands I_R could be described using classical Gauss \( I^G \), Lorenz \( I^L \) or Voight \( I^V \) dependencies according to physical model of the task:

\[ I^V = x \cdot I^G + (1 - x) \cdot I^L, x \in [0 \div 1] \] (5)

\[ I^G = \frac{A_0}{\sigma} \sqrt{\frac{2}{\pi}} \exp\left(-\frac{2(\omega - \omega_0)^2}{\sigma^2}\right) \] (6)

\[ I^L = \frac{2A_0}{\pi} \frac{\sigma}{4(\omega - \omega_0)^2 + \sigma^2} \] (7)

A_0 represents the altitude – intensity at \omega_0, and \sigma represents the halbwidth of band.

Corresponding author, email: Ildar.Galikov@gmail.com
2. Structuric scheme of CROWN

CROWN is written using object-oriented programming (OOP) language well known JAVA [17]. Java programming language was selected, due to it’s portability between different operation systems. Different classes of program are responsible for data storage, calculating, visualization and Graphical User Interface (GUI). Therefore it creates the possibility for rapid updating of program.

Fig. 1 shows the principal scheme of the program. Firstly, input data as two-dimensional distribution is read and visualized. After that, user sets the background values. Next, user changes half-width or intensity of bands, in order to fit experimental and theoretically calculated from bands and half-width CARS spectra. When the purpose is achieved, bands parameters are writing to file. Used libraries and their description are shown in Table 1. The most important classes and sub-classes are shown and described in Table 2.

2. Description of graphical User Interface

Program Crown is written to give a possibility to do easy, comfortable and fast CARS spectra decomposition. CARS spectrum decomposition proceeds from opening data file with experimentally calculated values, presented as X, Y matrix.

Fig. 2 shows file choosing dialog. The same result could be achieved by pressing “alt+F3” keyboard keys combination. After reading file, user must select background values, by pressing mouse at two points. The background is a linear function that goes through selected two points.

Table 2. The most important classes and sub-classes.

<table>
<thead>
<tr>
<th>Class name</th>
<th>Sub-class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WindowClass</td>
<td>MouseMoved</td>
<td>GUI objects creation, response to user interactions</td>
</tr>
<tr>
<td></td>
<td>MouseClicked</td>
<td>Mouse motion events</td>
</tr>
<tr>
<td></td>
<td>JInfoPanel</td>
<td>Mouse pressed events</td>
</tr>
<tr>
<td></td>
<td>PaintPanel</td>
<td>Output of important information to the screen( the right part of the workspace)</td>
</tr>
<tr>
<td>Bands</td>
<td>SaveAs</td>
<td>Spectrum data visualization, alongside other painting work.</td>
</tr>
<tr>
<td>Spectrum</td>
<td>MenuLoadXY</td>
<td>Reading data from file.</td>
</tr>
<tr>
<td>ChooseDialog</td>
<td></td>
<td>Writing data to file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data of a list of bands, for Raman and CARS spectrums calculation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spectra (CARS, RAMAN, calculated CARS) data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dialog window, for changing bands data(x,y, phase).</td>
</tr>
</tbody>
</table>

Fig. 3-10 illustrates visualization and fitting processes. Fig. 3, 4, 5 show the whole background defining process. Green line shows the place, where the band would be added, if the left mouse button be pressed. Fig. 6 shows experimental CARS spectra data, selected background and 3 bands. Bands can be deleted by pressing right mouse button. The closest to the mouse band (by x coordinate) will be deleted.
Fig. 3. Selecting of the first point of the background. Red curve corresponds the input file values.

Fig. 4. Selecting the second point of background.

Fig. 5. After selecting both background points. At the right side of the workspace is written background equation. The black line corresponds to the background.

Fig. 6. Blue lines corresponds to the selected bands. Their values are shown in the information part of the workspace. Pink curve corresponds to theoretically calculated CARS spectrum. Black curve corresponds to the Raman spectrum.

Fig. 7 shows the result after pressing the right mouse button on the right side of all bands, presented Fig. 6. There is a possibility to change x, y and phase of any bond. To do that, user could either press middle mouse button (this selects the nearest band) or press the left button directly on the band (blue line). 8 picture shows the dialog window, of changing values of the middle band.

Band values could be fitted by pressing “fast fit”, “accurate fit” buttons or changing halfwidth values. 9 picture shows the results, after setting half-width = 35 and pressing “fast fit” two times. In purpose to get less difference between experimentally and theoretically calculated CARS spectrums, use “accurate fit” button. 10 picture shows the better fitting, after adding two more bands and using “accurate fit” algorithm.
Raman spectrum information could be exported by pressing "alt+F2" key combination, or by selecting the corresponding menu. 11 picture shows exported data. The first column shows X values, the second – Raman spectrum values. Other columns have values of Raman spectrum components each component place an accentual row in the Raman spectrum.

3. Fitting algorithm

It is important to understand, that CARS spectra changes, whenever band’s phase or intensity are changed. The fitting is done, by changing Y values of bands, depending on difference between experimental and theoretically calculated CARS spectro values. It would take a huge amount of computer resources to recalculate CARS and Raman spectrums after changing any bands value. To solve that problem we’re created two fitting algorithms – fast fitting and accurate fitting. Algorithms start to work after pressing „do fit“ or „accurate fit“ buttons.

Fast fitting algorithm. This algorithm is dedicated to fit approximately values. By running this algorithm every bands value changing by 5%, depending on difference of theoretically calculated value and value from inputted file at the bands X value. Theoretical CARS and RAMAN spectrums recalculated, after changing every bands value. This process is repeated thirty times.

Accurate fitting algorithm. This algorithm is dedicated to make theoretically calculated CARS spectrum as much closer to experimentally obtained spectrum value. It is obvious that bands with higher value have stronger influence on whole spectrum, so all bands are sorted by their value. CARS spectra recalculated right after changing any bands Y value. Therefore each other consequent calculations include corrective from all others. That is why this algorithm is much more accurate. Beginning from band with highest Y value and finishing with the lowest, band’s value is changing by 1% and Raman and CARS spectrums are recalculated. This process is repeated ten times.

Conclusions

Novel tool was created for express estimation of CARS and Raman spectrum content without necessity to use routine, time-wasting fitting procedure in the some wizards.

Acknowledgements

This work was partly supported by the Lithuanian State Science and Studies Foundation (project B-07013 “KARS-KOPAS”).

References


17. Java SE Development Kit 6 (JDK 6) 25 update.

