Towards the verification of business process simulation in a JADE framework

D. Vymětal, R. Šperka, K. Slaninová, M. Spišák

Abstract—Globalization and economic recession in part of the world challenge the enterprise managements with a necessity to improve flexibility and new market approaches. For this, effective management decision support is needed. One of the most interesting decision support tools is simulation of business processes and behavioral patterns. The motivation for research field presented in this paper is to investigate some kinds of behavioral patterns from the agent-based simulation of business process outputs point of view.

Multi-agent system was implemented which is able to deal with unpredictable phenomena surrounding every company nowadays complemented by investigations of agent behavior. It is difficult to simply observe outputs of agent behavior, using large amount of agents representing real situations on the market. For better visualization, behavioral patterns taking the similarity of agents’ behavior into consideration were identified. The patterns were extracted by means of sequential pattern mining focused to events combined with traditional methods and time warping. Obtained results are presented by complex network and social network analysis tools. For better presentation of large networks, methods of spectral graph partitioning are used. Visualization of behavior patterns is suitable for better understanding of outputs in business process simulation using multi-agent systems. Moreover, the method allows for transparent search upon the network based on the queries meeting specific needs of potential observer. Proposed method is used for the analysis of agent-based system outputs to find the influence while testing various input parameters during the business process modeling and simulation.

Keywords—software agents; process mining; log files; business process simulation.

I. INTRODUCTION

The managements of companies cope with rapid changes on the markets caused by globalization and recession in some parts of the world. The management tasks concentrate to a great deal on the flexibility, customer satisfaction, optimal production, social planning and other strategic tasks aiming to fulfill difficult success criteria [Suchanek and Bucki (2011)], [Bucki and Chramcov (2011)] and others. In this sense, the information systems meet with challenge requiring operational processes support containing more intelligence features. Common approaches to this task include data warehouse and data mining methods [see e.g. Lacko (2003)], [Wolf (2006)]. Nevertheless, when using large data sets needed by business intelligence methods, companies often meet difficulties. Thus, many researchers concentrate on process analysis rather than on Key Performance Indicators (KPI) analysis in order to enhance organizations ability to survive in present market environments.

Process models are usually used for: “Insight, Discussion, Documentation, Verification, Performance analysis, Animation, Specification and Configuration” [(Aalst, 2011, p. 6)]. There are two classes of the process models – informal models specifying higher levels of a process structure and formal models resulting in an operational code in the end. The formal approaches to the process modeling use mostly mathematical tools. [see e.g. Liu and Trivedi (2011)], [Gries et al. (2011)]. There has been the extensive discussion on alignment of both types of method types lately. However, this discussion is beyond the scope of this paper. One of the process model shortcomings is a lack of value flow orientation shown in the process and workflow modeling generally. The shortage of value flow orientation can be overcome by value flow models, typically represented by the REA ontology. [Vymětal (2009)], [Huňka et al. (2011)], [Hruby (2006)], [McCarthy (1982)] It has been shown lately, that also the value flow models can be at least partly formalized [Ito and Vymětal (2012)]. However, independently of the modeling type, a lack of connection between both model types described by whatever means may be used and real running operational processes in the company can hardly be denied. This is why a new paradigm, namely process mining aimed to discover actually running processes using process logs has been met with large interest by process analysts lately [(Aalst, 2011)]. The methods used in process mining can be classified as a formal approach to process modeling.
In our research, we take even more factors in consideration. Extensive studies have shown that a business company must be looked upon as a system with social functions and responsibilities, where individuals besides the company KPIs also follow their personal aims and preferences [see e.g. the paper from Sharma, Sharma and Devi (2009) summarizing the Corporate Social Responsibility research of many other authors]. Similar situation can be observed on the markets. The customers and suppliers are acting there following their own goals and targets. We have to take other fluctuations like government decisions, global market fluctuations and other disturbances into consideration. In the end, we have rather stochastic system to work with. This is why our approach is based on two key features:

1) Simulation as a tool for decision support.
2) Software agent usage.

The generic business company used for our simulations is based on the control loop paradigm [Wolf (2006)], [Vymetal and Šperka (2011)], [Barnett (2003)].

For our research work, a multi-agent system was implemented which is able to deal with unpredictable phenomena surrounding every company nowadays. We use various types of agent behavior. It is difficult to simply observe outputs of agent behavior, when we use large amount of agents for the simulations of real situations on the market. Moreover, to check the models thoroughly, real business data need to be acquired what is a difficult task indeed. Instead of real data we use randomly generated data from a virtual high tech business company. Out motivation for further research was to check, if the simulated output patterns generated from randomly distributed input data and parameters are adequate to the model structure used for the simulation. For this analysis, a simple sales process was simulated using customer agents and sales representative agents negotiating the sales. The negotiations were governed by targets and aims of the agents on both sides and logged into history files. The outputs of the simulation were subject to reverse process by means of log mining and presentation using the ideas introduced by [van der Aalst (2011)]. The structure of the paper is as follows. In the second section the business process simulation and obtained outputs are described. In the third section the process mining methodology used for reverse process is described which is then presented in section 4. The last two sections describe the process mining results followed by conclusions.

II. BUSINESS PROCESS SIMULATION DESCRIPTION

Business Process Simulation Model (BPM) described in this section is based on a control loop paradigm. Market conditions as well as customers behavior are seen as an external part of the modeled system while the internal company behavior is subject to the simulation. We simulate core business processes of a business company like selling the goods to the customers as a part of the whole control loop (Fig. 1). Multi-agent system is implemented in order to serve as a BPM simulation framework. The subject of the simulation presented in this paper consists of the seller and customer agent types, the informative agent, and the manager agent. It represents the sales controlled component of the generic model. Seller agents interact with customer agents according to the multi-agent approach. The interaction is standardized and based on the FIPA contract-net protocol (Fig. 2) ([FIPA, 2002]).

This simplified system was extended by disturbances influencing the agents’ behavior. The disturbances occurrence is random and the number of customer agents is significantly higher than the number of seller agents. Under these circumstances, the whole system can be described as a stochastic system.

The behavior of agents in the simulation framework is influenced by more randomly generated parameters using normal distribution. The influence of randomly generated parameters on the simulation outputs while using different kinds of distributions is presented in our previous works e.g. [Vymetel, Spisák and Šperka (2012)]. The normal distribution seems to be optimal for modeling real business processes. The overall workflow of the system proposed can be described as follows. The customer agents randomly generate the requests
to buy some random pieces of goods. Seller agents react to
these requests according to their own internal decision
functions and follow the contracting. The purpose of the
manager agent is to manage the requests exchange. The
contracting results in the sales events to the customers. More
attributes of sales like costs, pieces sold and gross profit are
analyzed. These KPI attributes results could be used for
further analysis. Especially in situation, when real business
data are not available.

A. Mathematical Model and Implementation Details

The simplified model used to illustrate our assertions takes
only one kind of stock item in trade consideration depicted by
simulation experiments. The amount of stock items is not
limited. As many pieces the customer wants to buy, so many
he gets. A seller to customers’ ratio was chosen as 1:10 - one
seller serves for 10 customers. The customers were joined into
groups. Each group is being served by a certain seller. This
relationship is given. None of the agents can change its
counterpart. In each period turn (here we assume a week) the
customer agent decides whether to buy something or not. His
decision is defined randomly. If the customer decides not to
buy anything, his turn is over. Otherwise he creates a sales
request and sends it to his seller. The seller agent answers with
the proposal message (a quote starting with his maximal price
using limit price parameter such as – limit price * 1.25). This
quote can be accepted by the customer or not. An acceptance
is decided due to the valuation of a customer productivity
function, which can be formulated like in [Vymětal, Spišák and Šperka (2012)] as follows:

\[ c_n^m = \frac{\tau_n \cdot T_n \cdot \gamma \cdot \rho_m}{Z \cdot M \cdot y_{mi}^n} \]  

(1)

Where:
- \( c_n^m \) - the price of the \( n \)-th product quoted by \( m \)-th sales representative,
- \( \tau_n \) - the company market share for the \( n \)-th product \( 0 < \tau_n \leq 1 \),
- \( T_n \) - the market volume for the \( n \)-th product in local currency,
- \( \gamma \) - the competition coefficient lowering the sales success
  \( 0 < \gamma \leq 1 \),
- \( \rho_m \) - the quality parameter of the \( m \)-th sales representative
  \( 0,5 < \rho_m \leq 2 \),
- \( M \) - the number of customers,
- \( Z \) - the number of sales representatives in the company,
- \( y_{mi}^n \) - the requested number of the \( n \)-th product by the \( i \)-th
customer at \( m \)-th sales representative.

The proposed price must be less or equal the calculated
price (on behalf of the customer production function). If the
price is acceptable, the contract is awarded, otherwise not. The
negotiation between the seller agent and the customer agent
uses the contract-net protocol [FIPA, 2002]). If the price or
the quantity is not accepted by the customer, a rejection
message is send to the seller. In such case, the seller decreases
the price to the average of the limit price and the current price
(in every iteration is getting effectively closer and closer to the
limit price) and resends the quote back to the customer. The
message exchange repeats until there is an agreement or a
reserved time passes.

The seller is responsible to the manager agent. The manager
agent gathers data from all sellers in each turn and evaluates
the state of the company situation. These data are the result of
the simulation experiment. The BPM simulation outputs serve
to understand the company behavior in time. Different
simulation outputs depending on the agents’ decisions,
parameters, and behavior are obtained. The customer agents
need to know some information about the market (e.g.
company’s market share). The informative agent provides this
information. This agent is also responsible for the turn
management. The informative agent is representing outside or
controllable phenomena from the agents’ perspective.

The agent platform JADE [Bellifemine, Caire and Trucco
(2010)] was chosen for the implementation. It is a real tool for
rapid agent development with the communication language
feature presented. JADE is the complex platform for agents’
design, creation, and deployment. It provides the runtime
environment as a virtual place the agents exist in. There are
libraries of agent features to use and also graphical tools to
administrate them and monitor their state. [Wooldridge
(2009)] mentioned that JADE is the best-known and most
widely used agent-based development package.

When simulating the unpredictable phenomena, the multi-
agent system framework uses randomly (or pseudo randomly)
generated data from the normal distribution. They provide the
critical aspect of the uncertainty in a deterministic world. We
have chosen two important agents attributes to be generated by
the pseudorandom generator. These are sellers’ agent ability
(to sell) and customers’ agent decided quantity for purchase.
We triggered more simulation experiments with different
agents’ parameterizations. The obtained results are presented
in the next section.

B. Simulation Results

One year of trading processes (52 weeks) was simulated in
each simulation experiment. Different parameterizations for
each experiment were used. The KPI values obtained by BPM

Table 1 Aggregated KPI values in 52 weeks. [] Vymětal, Spišák and Šperka (2012)

<table>
<thead>
<tr>
<th></th>
<th>Pieces sold</th>
<th>Incomes (CZK)</th>
<th>Costs (CZK)</th>
<th>Gross Profit (CZK)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUM</td>
<td>1969</td>
<td>12306,25</td>
<td>7876</td>
<td>4430,25</td>
</tr>
<tr>
<td>AVG</td>
<td>37,87</td>
<td>236,66</td>
<td>151,46</td>
<td>85,2</td>
</tr>
<tr>
<td>St.dev.</td>
<td>14,68</td>
<td>91,73</td>
<td>58,7</td>
<td>33,02</td>
</tr>
</tbody>
</table>

simulation were different from case to case. The aggregated
results are presented in Tab. 1.

Three types of the final results can be seen in corresponding
three rows of this table. At the end of the year (52 weeks), four
types of KPIs were computed (Pieces sold, Incomes, Costs and Gross Profit values). In the first row of Table 1, the counted up KPI values are recorded. The average and the standard deviation values are listed in the following rows of Tab. 1 correspondingly. Similar KPI values achieve real companies on the real markets.

More important than absolute numbers is the course of KPI function. One year curves show similar trends. Typical KPI functions are presented in Fig. 3. Sharp fluctuations are typical for the current situation on the markets. The companies have to deal with these disturbances in order to survive. Therefore the agent-based BPM simulation shows fluctuating trend in KPI functions.

For this kind of simulation, a verification of the model used is of importance. To enable this verification, a log file is created during each running experiment. All the actions like request exchange, prices, contracting, dates, amounts etc. are recorded in the log file. In the next section the structure of the log file is described.

C. Log File Description

A log file recorded by the model is a typical text file, where each row represents one event performed by one agent during one instance of the negotiations – the case. The model records information like date and time, local name of the agent instance, agent class name, activity and activity information. Activities in the log file correspond to the communication between the agents. A sequence of actions performed by agent during one case is called trace in the presented model. In the model, there are five types of agents, as mentioned in Section 2 namely: CustomerAgent, SellerAgent, InformativeAgent, ManagerAgent and DisturbanceAgent. Two actions are common for all agents: IS_SETUP (agent is setup and ready to work) and TERMINATING (agent is taking down, his instance is being freed). Other actions depend on an agent class name.

As shown in Tab. 2, for example CustomerAgent provides the actions like sending result of decision, sending messages to SellerAgent, sending information about the amount of items for negotiation, sending reply to seller (limit price, offered price, acceptation, quantity), while SellerAgent provides actions like generation of ability

![Fig. 3 KPI values. One year of trading is presented (source: own).](image)

### Table 2 Types of actions in relation to agent type (source: own).

<table>
<thead>
<tr>
<th>Agent Class Name: CustomerAgent</th>
<th>Action Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECIDED_BUY</td>
<td>Result of decision</td>
<td></td>
</tr>
<tr>
<td>STARTING_WORK</td>
<td>Start of the agent work</td>
<td></td>
</tr>
<tr>
<td>HAS_NO_SELLER</td>
<td>No seller has been assigned to the agent</td>
<td></td>
</tr>
<tr>
<td>FINISHED_TURN</td>
<td>Turn maker decides that turn has been finished, or agent does not want to buy anything, or negotiation has been finished</td>
<td></td>
</tr>
<tr>
<td>GOT_INFO</td>
<td>Returned information from InformativeAgent</td>
<td></td>
</tr>
<tr>
<td>QUANTITY_GENERATED</td>
<td>The amount of stock items for negotiation was generated</td>
<td></td>
</tr>
<tr>
<td>SENT_CPF_MESSAGE</td>
<td>Message was sent to SellerAgent</td>
<td></td>
</tr>
<tr>
<td>RECEIVED_CPF_MESSAGE</td>
<td>Received proposal from SellerAgent</td>
<td></td>
</tr>
<tr>
<td>SENT_REPLY</td>
<td>Sent reply to SellerAgent</td>
<td></td>
</tr>
<tr>
<td>ITEM_PURCHASED</td>
<td>Item was successfully purchased</td>
<td></td>
</tr>
<tr>
<td>Agent Class Name: SellerAgent</td>
<td>ABILITY_GENERATED</td>
<td>Parameter of agent ability was generated</td>
</tr>
<tr>
<td>SENT_PROPOSAL</td>
<td>Proposal was sent to Customeragent</td>
<td></td>
</tr>
<tr>
<td>SOLD</td>
<td>Item was sold</td>
<td></td>
</tr>
<tr>
<td>Agent Class Name: ManagerAgent</td>
<td>TURN_RESULT</td>
<td>Result of the turn</td>
</tr>
<tr>
<td>Agent Class Name: InformativeAgent</td>
<td>OPENING_NEW_TURN</td>
<td>New turn is being opened</td>
</tr>
<tr>
<td>CLOSING_TURN</td>
<td>Turn is being closed (all customers finished)</td>
<td></td>
</tr>
<tr>
<td>FOUND_CUSTOMERS</td>
<td>After initialization, the agent have found all CustomerAgents to communicate with</td>
<td></td>
</tr>
<tr>
<td>Agent Class Name: DisturbanceAgent</td>
<td>DISTURBANCE_VALUE</td>
<td>Value of disturbance influence</td>
</tr>
</tbody>
</table>
parameter, sending a proposal to CustomerAgent, sending information of sold item etc. Example of analyzed log can be seen in Fig. 4.

![Figure 4 Example of log file (source: own)](image)

III. PROCESS MINING METHODOLOGY

The motivation of this paper is the analysis of BPM model outputs leading to the agent structure check. As the model is designed as a multi-agent system, this analysis is done through agent behavior analysis. The analysis of agent behavior can be divided into two main parts: (1) finding behavioral patterns, (2) finding groups of agents with similar behavior. The agent behavior is recorded in the form of traces and stored in the log file correspondingly. Thus, the agent behavior can be defined with the terms of process mining which are used commonly in the business sphere. Aalst defines an event log as follows [Aalst et al. (2011)], [Aalst et al. (2005)]:

**Definition 1.** Let \( A \) be a set of activities (also referred as tasks) and \( U \) as set of performers (resources, persons). \( E = A \times U \) is set of (possible) events (combinations of an activity and performer). For a given set \( A, A' \) is set of all finite sequences over \( A \). A finite sequence over \( A \) of length \( n \) is mapping \( \sigma = \langle a_1, a_2, ..., a_n \rangle \), where \( a_i = \sigma(i) \) for \( 1 \leq i \leq n \).

\( C = E \) set of possible event sequences.

**Definition 2.** A simple trace \( \sigma \) is a sequence of activities, i.e., \( \sigma \in A' \). A simple event log \( L \) is a multi-set of traces over \( A \), i.e., \( L \in B(A') \).

The topic of multiset is out of scope this paper, but can be thoroughly followed in Aalst et al. (2011). Then, the agent behavior can be described by a set of event sequences. The first step of the agent behavioral analysis is the finding of behavioral patterns. The behavioral patterns are discovered using similarity of extracted sequences. A sequence is an ordered list of elements, denoted \( \langle e_1, e_2, ..., e_l \rangle \). Given two sequences \( \alpha = \langle a_1, a_2, ..., a_m \rangle \) and \( \beta = \langle b_1, b_2, ..., b_n \rangle \). \( \alpha \) is called a subsequence of \( \beta \), denoted as \( \alpha \subseteq \beta \), if there exist integers \( 1 \leq j_1 < j_2 < ... < j_n \leq m \) such that \( a_i = b_{j_1}, a_2 = b_{j_2}, ..., a_n = b_{j_n} \). \( \beta \) is than a super sequence of \( \alpha \).

In the problem of finding a similar behavior, there are traditionally used methods of sequential pattern mining where usually frequently repeated patterns are extracted. In the case of finding behavioral patterns with relation to its performer (agent), there is a need to use other methods for the sequence comparison, see below.

Several algorithms for the comparison of two or more categorical sequences are generally known. Some of them deal with the fact, whether the sequences consist of ordered or unordered elements. Another algorithms focus on the comparison of the sequences with the different lengths and with the possible error or distortion.

The basic approach to the comparison of two sequences, where the order of elements is important, is the longest common substring (LCS) method (Gusfield 1997). As obvious from the name of the method, the main principle here is to find the length of the common longest substring. LCS method respects the order of elements in the sequence. However, its main disadvantage is that it can find only identical subsequences, where no extra element is presented in the sequence. This fact can be seen as too strict limitation in some domains, where exists the large amount of different sequences.

As a solution of this problem, the longest common substring method (LCSS) described for example in (Hirschberg, 1977) can be considered. Contrary to the longest common substring, this method allows for (or ignores) the inserted extra elements, and therefore, it is immune to slight distortions. Whether we define the similarity of compared sequences as a function using a length of common subsequence, we can find one characteristic of this method. The length of the common subsequence is not immune to recurrence of identical elements, which can occur only in one of the compared sequences. We can find such situations, for example due to inappropriate sampling or due to any kind of distortion.

In some applications, it is suitable (or sometimes even required) to eliminate such type of distortions and to work with them like with equivalent elements. This solution results in another method, the time-warped longest common subsequence (T-WLCS) ([Guo, 2004]), combining the advantages of LCSS method with dynamic time warping [Müller, 2007]; [Kocyan, 2012]. Dynamic time warping is used for finding the optimal visualization of elements in two sequences to match them as much as possible. This method is immune to minor distortions and to time nonlinearity. Moreover, the method emphasizes recurrence of elements in one of the compared sequences.

IV. EXTRACTION OF AGENT BEHAVIOR

The log file of BPM model contains the records of all events (activities) performed by the agents, such as communication between the seller and customer agents; see Section 2.3. Performers of the activities are the agents of following types: CustomerAgent, SellerAgent, ManagerAgent, DisturbanceAgent and InformatieAgent. Each activity can have additional information like market share of the product, total volume of the product on the market, price and quantity of proposed product, limit price, offered price and quantity in
the reply of CustomerAgent, and other. The process of agent behavior extraction can be seen in Fig. 5.

Using the principles of process mining methodology described in section 3, a set of all sequences S was extracted from the log file. Sequences represent the agents’ behavior during one turn (week) in the model. Due to the large amount of similar sequences, the next step for finding of behavioral patterns was performed. Behavioral patterns were extracted via LCSS and T-WLCS methods for the similarity measurement of sequences, described in Section 3. Both methods LCSS and T-WLCS find the longest common subsequence \( \alpha \) of compared sequences \( \beta_x \) and \( \beta_y \), where \( \alpha \subseteq \beta_x \wedge \alpha \subseteq \beta_y \), with relation to both methods. Similarity was counted by the (2).

\[
sim(\beta_x, \beta_y) = \frac{(l(\alpha)h)^2}{l(\beta_x)l(\beta_y)}
\]  

(2)

where \( l(\alpha) \) is a length of the longest common subsequence \( \alpha \) for sequences \( \beta_x \) and \( \beta_y \); \( l(\beta_x) \) and \( l(\beta_y) \) are lengths of compared sequences \( \beta_x \) and \( \beta_y \), and

\[
h = \frac{\min(l(\beta_x)l(\beta_y))}{\max(l(\beta_x)l(\beta_y))}
\]

(3)

On the basis of selected method for finding the similarity of sequences, the similarity matrix for sequences \( \left| S \right| \times \left| S \right| \) was constructed. Obtained matrix can be represented using tools of graph theory. For the visualization of relations from matrix weighted graph \( G(V, E) \) was used, where set \( V \) is represented by set of sequences \( S \) and edges \( E \) are assessed by weight \( w \), which is defined by the similarity of sequences, see Equation 2. Obtained graph consisted of large amount of similar sequences. Moreover, it was dense and very large for further processing. For better interpretation of results, spectral clustering method by Fiedler vector and algebraic connectivity (Fiedler, 1975) was used to divide graph into smaller components, which can be interpreted as behavioral patterns. Using behavioral patterns, a weighted graph with agents can be constructed, where each agent is described by its behavior, and edges in the graph represent similarity of agents on the basis of their behavior.

V. PROCESS MINING RESULTS

The analyzed log file consisted of 23556 records with the communication of 500 SellerAgents and 50 CustomerAgents. In the first phase of the analysis, 551 sequences were extracted, where 168 sequences were unique. Appearance of the sequences followed power law distribution. For the definition of sequence similarity LCSS and T-WLCS methods were used. Following tables and figures describe examples of results, which can be obtained after sequence analysis.

Tab. 3 shows the description of weighted graph of sequences obtained by LCSS method, for similarity level > 0.5.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>214</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isolated Nodes</td>
<td>2</td>
</tr>
<tr>
<td>Edges</td>
<td>12158</td>
</tr>
<tr>
<td>Connected Components</td>
<td>4 (166, 46, 1, 1)</td>
</tr>
<tr>
<td>Avg. Degree</td>
<td>56.813</td>
</tr>
<tr>
<td>Avg. Weighted Degree</td>
<td>37.436</td>
</tr>
</tbody>
</table>

From Tab. 3 can be seen that LCSS method divides sequences into 4 connected components of size 166, 46, 1 and 1 sequences (numbers in the brackets). Each component is created on the basis of sequence similarity. However, the first two components are too large for the definition of behavioral patterns.

The largest component (166 sequences) represents the customer agents’ sequences in this case. This component was subject of spectral clustering in the next step. Spectral clustering method divided the largest component into four, smaller ones with 60, 58, 32 and 16 sequences. Tools from graph theory provide support for better visualization of relations between sequences on the basis of their similarity. Fig. 6 shows weighted graph of sequences, where nodes represent unique sequences and edges.
The colors describe the components of similar sequences obtained by spectral clustering. Thus, each component can be interpreted as a behavioral pattern of one group of customer agents with the similar behavior. After the analysis, four groups of agents with the similar behavior within the customer agents’ population were found.

In Tab. 4 are presented the agents which can be described by the behavioral patterns from the largest component shown in Fig. 6.

After the application of spectral clustering method on the second largest component, 8 groups of agents with similar behavior were obtained. Another output example of the analysis is shown on Fig. 7, where weighted graph of sequences generated by T-WLCS method is presented, with the edge weight (similarity) being set up > 0.9. Presented network contains eight components for this level and method. Each component represents different type of agent behavior.

Thus, 8 groups of agents with similar behavior were identified using T-WLCS method as well, but with another similarity.

The purpose of stated examples was to demonstrate the usage of different process mining methods in the analysis of BPM simulation results. The analysis should be used to verify the BPM model implementation. The purpose of the verification is to confirm the correctness of proposed BPM model in order to use it in the future as a simulation framework. The process mining results in the graph forms were used to identify several groups of agents with the similar behavior. In the text above were identified more groups of agents within the customer agent population (LCSS method) and more groups of agents with similar behavior within the whole population of agents (T-WLCS method).

VI. CONCLUSION

The verification possibilities of Business Process Simulation model was presented in this paper. The simulation was based on the control loop methodology and multi-agent paradigm. Multi-agent simulation framework was developed to provide simulation environment for the virtual company. The implementation was done in the JADE development platform. For the simplification, only the sales part of virtual trading company was used. This part consists of the customer, seller, informative, and the manager agent types. Only one stock item during 52 weeks was traded. Current economic environment with sharp fluctuations and disturbances was involved into the system through the random number generation of some agents attributes. Simulation experiments running in the implemented multi-agent system produced quality Key Performance Indicator outputs. All trading actions of all actors were recorded to the log file during the simulation.

The verification process used the records stored in the log file. Using the methods for of process mining and sequence mining, the behavioral patterns of agents in the system were obtained. Moreover, the groups of agents described by behavioral patterns were obtained. The examples and outlines of the process mining analysis were demonstrated. Formal description and visualization possibilities of used methods were stated. Obtained results were presented by complex network and social network analysis tools using methods of spectral graph partitioning. Proposed method was used in the verification process, where the authors can investigate the

<table>
<thead>
<tr>
<th>Component</th>
<th>Size</th>
<th>Agents</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>60</td>
<td>ca0011, ca0379, ca0393, ca0250, ca0472, ca0245, ca0147, ca0197, ca0286, ca0338, ca0342, ca0224, ca0375, ca0350, ca0185, ca0381, ca0485, ca0044, ca0128, ca0470, ca0089, ca0173, ca0111, ca0408, ca0366, ca0211, ca0243, ca0383, ca0039, ca0489, ca0352, ca0043, ca0134, ca0013, ca0481, ca0190, ca0175, ca0374, ca0438, ca0329, ca0353, ca0311, ca0024, ca0256, ca0091, ca0475, ca0337, ca0260, ca0457, ca0163, ca0001, ca0041, ca0087, ca0436, ca0332, ca0143, ca0355, ca0447, ca0031, ca0253</td>
</tr>
<tr>
<td>C2</td>
<td>58</td>
<td>ca0274, ca0303, ca0273, ca0440, ca0345, ca0247, ca0418, ca0124, ca0398, ca0313, ca0261, ca0466, ca0033, ca0469, ca0056, ca0240, ca0090, ca0483, ca0187, ca0027, ca0298, ca0214, ca0129, ca0360, ca0036, ca0060, ca0486, ca0450, ca0002, ca0053, ca0140, ca0479, ca0216, ca0085, ca0411, ca0395, ca0198, ca0448, ca0348, ca0354, ca0047, ca0219, ca0125, ca0189, ca0343, ca0297, ca0463, ca0005, ca0172, ca0403, ca0478, ca0426, ca0328, ca0170, ca0015, ca0188, ca0416, ca0006</td>
</tr>
<tr>
<td>C3</td>
<td>32</td>
<td>ca0182, ca0055, ca0153, ca0169, ca0424, ca0233, ca0488, ca0014, ca0378, ca0437, ca0399, ca0195, ca0127, ca0490, ca0391, ca0131, ca0376, ca0071, ca0265, ca0037, ca0496, ca0318, ca0467, ca0347, ca0387, ca0494, ca0246, ca0054, ca0092, ca0372, ca0223, ca0213, ca0228, ca0362, ca0093, ca0109, ca0120, ca0344, ca0207, ca0052, ca0272, ca0004, ca0212, ca0201, ca0205, ca0050, ca0097, ca0331</td>
</tr>
<tr>
<td>C4</td>
<td>16</td>
<td>ca0379, ca0393, ca0250, ca0472, ca0245, ca0147, ca0197, ca0286, ca0338, ca0342, ca0224, ca0375, ca0350, ca0185, ca0381, ca0485, ca0044, ca0128, ca0470, ca0089, ca0173, ca0111, ca0408, ca0366, ca0211, ca0243, ca0383, ca0039, ca0489, ca0352, ca0043, ca0134, ca0013, ca0481, ca0190, ca0175, ca0374, ca0438, ca0329, ca0353, ca0311, ca0024, ca0256, ca0091, ca0475, ca0337, ca0260, ca0457, ca0163, ca0001, ca0041, ca0087, ca0436, ca0332, ca0143, ca0355, ca0447, ca0031, ca0253</td>
</tr>
</tbody>
</table>
influence of input settings of agents to their real behavior. The results obtained could be valuable in modeling and simulation process, where the setting of agent attributes and the impact of environment disturbances to the system are the topic of the analysis. The future research will be aimed to use the right process mining methods or their combination in order to verify the correctness of Business Process Simulation model.

REFERENCES


