TECHNOLOGICAL AND ECONOMIC DEVELOPMENT OF ECONOMY
Baltic Journal on Sustainability
2010
16(3): 455–470

AGENT-BASED PRODUCTION PLANNING SUPPORT SYSTEM

Serge Parshutin[1], **Arkady Borisov**[2]

Institute of Information Technology, Riga Technical University, 1 Kalku Str., LV-1658 Riga, Latvia
E-mail: [1]*serge.parshutin@rtu.lv;* [2]*arkadijs.borisovs@cs.rtu.lv*

Received 12 February 2010; accepted 5 August 2010

Abstract. Production planning is the main aspect for a manufacturer affecting the income of a company. Correct production planning policy chosen for the right product at the right moment in the product life cycle (PLC) reduces production, storing and other related costs. This leads to such problems to be solved as defining the present PLC phase of a product and also determining the transition point - a moment of time when the PLC phase changes. The paper presents the Agent-Based Production Planning Support system meant for supporting a production manager in his/her production planning decisions. The developed system is based on the analysis of historical demand for products and on the information about transitions between phases in life cycles of those products. The system creates a model and uses it to forecast a transition point for a product that only begins its life on the market. The architecture of the developed system is presented and an analysis of testing on real-world data results is given.

Keywords: Software Agents, Data Mining, Decision Support, Production Planning, Product Life Cycle, Forecasting Transition Points.

Reference to this paper should be made as follows: Parshutin, S.; Borisov, A. 2010. Agent-based production planning support system, *Technological and Economic Development of Economy* 16(3): 455–470.

1. Introduction

Steadily evolving computer technologies are increasingly becoming an inherent part of successful enterprises management and keeping its activity at a high level. Different institutions are trying to reduce their costs by fully automating certain stages of manufacturing process as well as introducing various techniques intended for forecasting certain market indicators that affect general manufacturing process. Different statistical methods are employed as well, though an increasing interest in computational intelligence technologies and their practical application can be observed ever more.

ISSN 2029-4913 print/ISSN 2029-4921 online doi: 10.3846/tede.2010.28
http://www.tede.vgtu.lt

We are focusing our research on studying the ability to combine the Agent Technologies and Knowledge Discovery Technologies in order to develop the Software Agent-Based system that will apply Data Mining and Decision Support technologies to a real-world problem autonomously or in a semi-supervised manner. Software agent definitions and features are provided in sources (Ferber 1999; Wooldridge 2005). Application examples of software agents can be found in various fields of science, industry, environment monitoring etc. (Athanasiadis and Mitkas 2004; Symeonidis *et al. 2003*).

Product life cycle phase transition point forecasting can serve as an example of a problem where both Data Mining and Decision Support technologies should be applied. From the viewpoint of the management it is important to know, in which particular phase the product is, as that type of knowledge can be applied for selection of the production planning policy for a product in a particular phase (Merkuryev *et al. 2007*). For example, for the maturity phase in case of determined demand changing boundaries it is possible to apply cyclic planning (Campbell and Mabert 1991), whereas for the introduction and decline phase an individual planning is usually employed. The additional knowledge of what planning strategy should be chosen can help in gaining more precise order forecasting results in Order-Simulation based systems (Akanle and Zhang 2008) as also in Game Theory based strategies (Reaidy *et al. 2006).* As the technologies are evolving, the variability of products grows, making manual monitoring of PLCs a difficult and costly task for companies. Thus having an autonomous Agent-Based system that monitors market data, creates and automatically updates lists of products for what is reasonable to consider a production planning policy update or replacement, is a valuable alternative. This paper proposes a model of Agent-Based system that ensures the solving of the aforementioned task as well as provides an analysis of system testing results.

2. Problem statement

Any created product has a certain life cycle. The term "life cycle" is used to describe a period of product life from its introduction on the market to its withdrawal from the market. Life cycle can be described by different phases: traditional division assumes such phases like introduction, growth, maturity and decline (Kotler and Armstrong 2006). For products with a relatively long life cycle it is possible to see those phases by analyzing the demand time series. In most of the cases it is possible to make some simplification in PLC, merging introduction and growth phases into one phase -introduction.

An assumption that three different phases, namely, introduction, maturity and end-of-life are possible in the product life cycle, gives us two possible transitions. The first transition is between introduction and maturity phases and the second – between maturity and the product's end-of-life.

In general the main task can be defined as follows: *By having a demand data for different products and also having a transition points defined, create a model that can be used to forecast a PLC phase change moment for a new product that only begins its life on the market.* Formally the task is stated below.

As to data mining *(*Dunham 2003; Han and Kamber 2006; Hand *et al. 2001*), information about the demand for a particular product is a discrete time series, in which demand value is, as a rule, represented by the month. The task of forecasting transition points between life cycle phases may be formulated as follows. Let us assume that $D = \{d_1,...,d_i,...,d_n\}$ is a dataset and $d = \{a_1,...,a_j,...,a_l\}$ is a discrete time series whose duration equals *l* periods, where $l \in L = \{l_1,...,l_h,...,l_s\}$ and varies from record to record in the dataset *D*. For simplification, the index of d is omitted. Time series d represents a particular phase of a product life cycle, say introduction. Let us assume that for a particular transition, like introduction to maturity, a set of possible transition points $P = \{p_1,...,p_k,...,p_m\}$ is available. Having such assumptions the forecasting of a transition point for a new product represented by a time series $d' \notin D$, will start with finding an implication between historical datasets *D* and *P*, $f : D \to P$; followed by application of the model found to new data.

3. Structure of the proposed system

The main idea was to create a simple system that can be applied to the stated problem, as also to any other task that encapsulates the clustering of time series with special markers and forecasting a marker value for a new object, represented only by a time series.

The system developed contains three main elements – Data Management Agent, Data Mining Agent and Decision Analysis Agent shown in Figure 1.

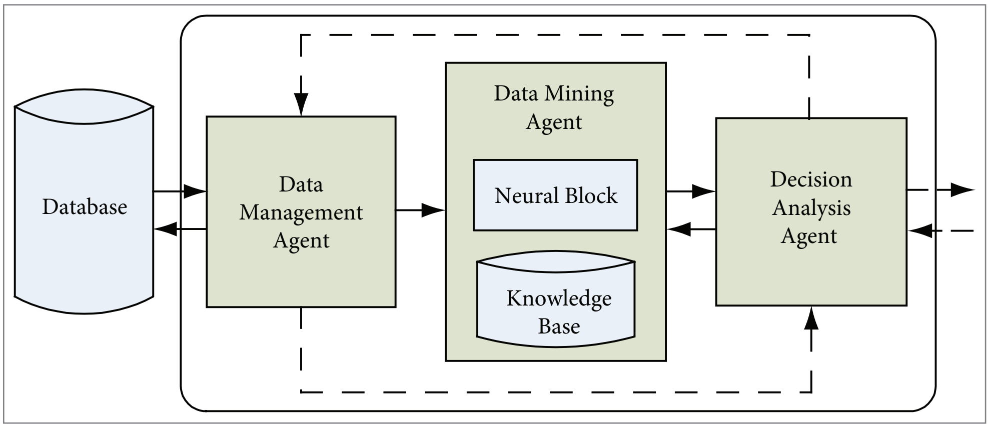

Fig. 1. Structure of the system

3.1. Data Management Agent

The Data Management Agent performs several tasks of managing data. It has a link to the database that contains the product demand data and is regularly updated. The Data Management Agent handles the preprocessing of the data – data normalization, exclusion of obvious outliers and data transformation to a defined format before sending it to the Data Mining Agent. Such preprocessing allows us to lessen the impact of noisiness and dominance of data (Chakrabarti *et al. 2009*; Tan *et al. 2006*). The transformed data record displays the demand for a product collected within a known period of time the length of which is set by the system – day, week, month, etc. Each data record is marked with one or both markers – transition indicators; namely, M1 indicates the period when product switched from Introduction phase to Maturity phase; and M2 indicates the period when product switched from Maturity phase to End-of-Life phase. Marks on transitions can guarantee that a model will be built; if we have patterns of transitions in historical data, then, theoretically, in the presence of a model for generalisation, we are able to recognise those patterns in new data. Assigning markers to the demand time series is the only process that is currently done manually by a human expert.

As the database is regularly updated, the Data Management Agent monitors the new data and at a defined moment of time forwards the subset of new data to the Data Mining Agent for updating the knowledge base.

3.2. Data Mining Agent

The actions, performed by the Data Mining Agent, cover such processes as initialization of a training process, performing system training and testing processes, imitation of On-line data flowing during system training and testing, Knowledge base creation and maintaining it up-to-date.

The knowledge base of the Data Mining Agent is based on the modular self-organising maps placed in the Neural Block. Each neural net can either handle records, with equal duration *l* or it can proceed with time series with different durations. Which option will be selected depends on the total load distribution policy, currently defined by the user. Let us illustrate the load distribution.

Let us assume that the duration of discrete time series in dataset *D* varies from 10 to 16 periods, thus $l \in L = \{10,11,...,16\}$. In this case, the total system load will consist of seven values that time series duration can take. Let us assume that the load has to be distributed uniformly over modules on condition that an individual load on separate module should not exceed four values, that is $q = 4$. Under such conditions, two neural networks will be created by the moment of system initialization. The first neural network will process time series of duration $l \in \{10,11,12,13\}$; the remaining time series with duration of 14, 15 and 16 periods will be sent to the second neural net.

The option of having the On-line data flowing procedure becomes necessary due to the specifics of a chosen system application environment. In the real-world situation, the information about the demand for a new product is becoming available gradually: after a regular period finishes, new demand data appear. Due to that, the system must be trained to recognize transition points that will occur in the future having only several first periods of the demand time series available.

The algorithm employed is executed taking into account the following details. Preprocessed by the Data Management Agent time series *d* with duration *l*, containing demand data within introduction or maturity phase and the appropriate marker (M1 or M2 respectively) with value *p,* is sent to the Data Mining Agent.

Bearing in mind that the minimal duration of a time series should be $l_{\min}$ and greater, the algorithm of On-line data flowing procedure will include these steps:

1. Define $l^* = l_{\min}$;
2. Process the first l^* periods of a record *d* with a marker value *p;*
3. If $l^* < l$, then increase the value of l^* by one period and return to step 2; else proceed to step 4;
4. End processing of record *d.*

According to the chosen policy of system load distribution, the fraction of a time series with marker is directed to the neural net responsible for processing the time series of specific duration.

Each self-organising map is based on the modified Kohonen map. The number of synaptic weights of a neuron in the classical Kohonen map equals the length of records - the duration of a time series, in the input dataset (Dreyfus 2005; Kohonen 2001; Obermayer and Sejnowski 2001). Due to such limitations, it is only possible to use the classical Kohonen map in the developed system when $q = 1$ for each neural network. To be able to maintain the system functionality while $q > 1$, it is necessary to apply some modifications to the classical Kohonen map.

A heuristic based on substituting the distance measure is considered in this work. We propose a substitution of a Euclidean distance measure with a measure based on Dynamic Time Warping (DTW). Using DTW allows us to process a time series with different duration by one neural net. The classical Dynamic Time Warping algorithm that is applied in the system is fully described in (Keogh and Pazzani 2001).

At the training stage, the neural networks are organised using the data prepared by the Data Management Agent. That is followed by forming clusters found by the neural networks which will form the main part of the knowledge base.

3.3. Decision Analysis Agent

The Decision Analysis Agent is the element that uses the knowledge base of the Data Mining Agent to forecast transition points for new products and also to analyse the alternatives of using cyclic or non-cyclic planning policies for particular products. The Decision Analysis Agent follows certain algorithms whose examples are provided in the next subsection.

3.4. System functioning algorithm

The system functioning algorithm consists of two main stages – System Learning and System Application. The System Learning stage contains two inner steps – System Training followed by System Testing and Validation. The System Application stage is the stage when the Decision Analysis Agent receives requests from the user and following certain algorithms described in current subsection, provides the user with certain information. The system functioning algorithm can be observed in Figure 2.

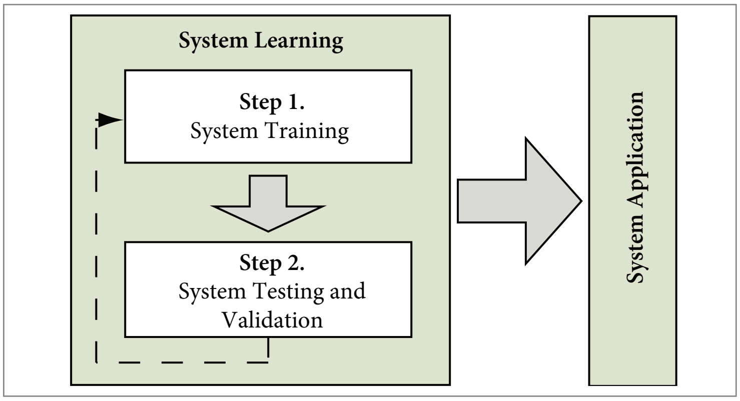

Fig. 2. System functioning algorithm

3.4.1. System learning

The system learning process is fired when the Data Management Agent sends some prepared data to the Data Mining Agent with "Start initial learning" or "Update" command. The "Start initial learning" command is sent when the system is launched for the first time. This will start the process of determination and setting of the basic system's parameters: the number of neural networks in the Neural Block, dimensionality and topology of neural networks and the number of synapses each neuron will have in each network. The number of networks in the Neural Block, *n*, is calculated empirically. Given a policy assuming uniform distribution of general load among the networks, formula (1) can be used to calculate the number of networks in the Neural Block:

$$n = \left\lceil \frac{|L|}{q} \right\rceil, \tag{1}$$

where q – each network's individual load; $\lceil\ \rceil$ – symbol of rounding up.

After the n is calculated, for each neural network m; an interval of time series durations $\left[l_{i,\min}; l_{i,\max}\right]$ is set. The records with duration $l \in \left[l_{i,\min}; l_{i,\max}\right]$ will be processed by module m_i. Given a uniform load distribution, equation (2) can be used:

$$\begin{cases} i = 1, & l_{i,\min} = l_{\min}, \\ i > 1, & l_{i,\min} = l_{i-1,\max} + 1; \end{cases}$$
$$l_{i,\max} = l_{i,\min} + q - 1. \tag{2}$$

The number of neurons of each network is determined empirically depending on the task stated. The number of synapses of a neuron in each network can be calculated using formula (3) below:

$$b_{j,i} = \left\lceil \frac{l_{i,\min} + l_{i,\max}}{2} \right\rceil, \tag{3}$$

where $b_{j,i}$ is the number of synapses of neuron j in network m_i.

As an alternative, median calculation can be used for determining the number of synapses of a neuron. Assuming that network m_i can process discrete time series of duration $l \in \left[l_1, l_2, \ldots, l_i, \ldots, l_k\right]$, let us denote for each value *l* the number of records in the training set (a data set forwarded by the Data Management Agent to the Data Mining Agent with "Start initial learning" command) having duration equal to *l, as* $f \in \left[f_1, f_2, \ldots, f_i, \ldots, f_k\right]$. Using that assumption, a median of time series durations a network m_i can process, may be calculated with formula (4):

$$Median = \frac{\sum_{i=1}^{k}\left(l_i \cdot f_i\right)}{\sum_{i=1}^{k} f_i}. \qquad (4)$$

The calculated median must be rounded to the integer thus obtaining the number of synaptic connections of a neuron. Whether the median will be rounded to a smaller or a greater value, to a large extent depends on the task to be solved.

As the main parameters are set, the initialisation of the system begins. Synaptic weights of each neuron in each network are assigned initial values – usually small values produced by the random number generator. At this moment networks in the Neural Block have no organization at all. Then the following main processes of the training step are launched: Competition, Cooperation and Synaptic Adaptation.

The Data Mining Agent processes each of the received records imitating the On-line data flowing. Fractions of time series are forwarded to the corresponding neural network m_i defined by the $\left[l_{i,\min}; l_{i,\max}\right]$ interval calculated with equation (2). Then the process of neuron competition for the right to become the winner (or best-matching neuron) for the processed fraction of a record begins. Discriminant function – the distance between the vector of the synaptic weights and discrete time series – is calculated using the DTW algorithm. Thus the neuron with the least total distance becomes the winner.

The winner neuron is located in the centre of the topological neighbourhood of cooperating neurons. Let us define lateral distance between the winner neuron (*i*) and re-excited neuron (*j*) as $ld_{j,i}$. Topological neighbourhood $h_{j,i}$ is symmetric with regard to the point of maximum defined at $ld_{j,i} = 0$. The amplitude of the topological neighbourhood $h_{j,i}$ decreases monotonically with the increase in lateral distance $ld_{j,i}$, which is the necessary condition of neural network convergence (Dreyfus 2005). Usually a Gaussian function is used for $h_{j,i}$ calculation (formula 5):

$$h_{j,i} = \exp\left(-\frac{ld_{j,i}^2}{2 \cdot \sigma^2(n)}\right). \qquad (5)$$

A decrease in the topological neighbourhood is gained at the expense of subsequent lessening the width of σ function of the topological neighbourhood $h_{j,i}$. One of possible kinds of σ value dependence on discrete time *n* is an exponential decline (formula 6):

$$\sigma(n) = \sigma_0 \cdot \exp\left(-\frac{n}{\tau_1}\right) \quad n = 0,1,2,\ldots, \qquad (6)$$

where σ_0 is the beginning value of σ and τ_1 - some time constant, such as the number of learning cycles.

To ensure the process of self-organisation, the synaptic weights of a neuron have to change in accordance with the input data, i.e. adapt to the input space. Let us assume that $w_j(n)$ is the vector of synaptic weights of neuron *j* at time moment (iteration, cycle) *n*. In this case, at time instant $n + 1$ the renewed vector $w_j(n+1)$ is calculated by formula (7):

$$w_j(n+1) = w_j(n) + \eta(n) \cdot h_{j,i(d)}(n) \cdot \left(d - w_j(n)\right), \tag{7}$$

where η – learning rate parameter; *d* – discrete time series from learning dataset.

Note how the difference between discrete time series and the vector of synaptic weights is calculated in expression (7). When the load is $q = 1$, that is when each neural network is processing discrete time series with a certain fixed duration, and DTW is not used, the difference between *d* and $w_j(n)$ is calculated as the difference between vectors of equal length. In other cases when DTW is employed, the fact of time warping has to be taken into account. For this purpose, during the organization process the Data Mining Agent fixes in memory a warping path on whose basis the distance between the vector of synaptic weights of the winner neuron and a discrete time series was calculated. Thus the following information is becoming available: the value of the discrete time series according to which the corresponding synaptic weight of the neuron has to be adjusted.

The network organization contains two main processes – initial organization and a convergence process. Initial organisation takes about 1000 iterations (Dreyfus 2005). During that process each network gets an initial organization, the learning rate parameter decreases from 0.1 but remains above 0.01.

The number of iterations the convergence process takes is at least 500 times larger than the number of neurons in the network (Dreyfus 2005). The main difference is that during the convergence process the topological neighbourhood of a winner neuron contains only the closest neighbours or just the winner neuron.

When the networks in the Neural Block are organized, the Data Mining Agent launches the cluster formation process. At the end of this process the knowledge base will be created. During the cluster formation process each record *d* of the dataset that was used for training is passed to the Neural Block with imitation of On-line data flowing. Algorithms for network m_i and winner neuron n_j^* determination fully coincide with those used in network organization.

In parallel, for each neuron n_j^* these statistics are kept: with which value of the key parameter *p* (transition point) records have got to neuron n_j^* and how many times. Cluster c_i is a neuron n_j that at least once became the winner neuron during cluster formation. The knowledge base will contain the clusters in each neural network, as well as the gathered statistics for the key parameter *p* in each cluster.

When the Data Management Agent changes the command to "Update", the records in the forwarded dataset will be used to update synaptic weights of neurons in the neural networks. If all records have durations that can be processed with neural networks in the Neural Block, then simply the training process with new records is launched. If Neural Block contains no neural network that can process durations that one or more new records have, then such neural network is created; and the training process with new records is launched. In both cases the neural networks organization and then the knowledge base are updated.

3.4.2. System evaluation

To evaluate the precision of transition point forecasts made by the system, two criteria are employed: Mean Absolute Error – MAE, to evaluate the accuracy of the system and Logical Error to evaluate whether decisions made by the system are logically correct.
The Mean Absolute Error (MAE) is calculated using formula (8):

$$MAE = \frac{\sum_{i=1}^{k} \left| p_i - r \right|}{k} \quad i = \left[1, 2, \ldots, k\right], \tag{8}$$

where k – the number of records used for testing; p_i – real value of the key parameter for record d_i; r – the value of the key parameter forecasted by the system.

Logical error provides information about the logical potential of the system. To calculate a logical error, it is necessary to define logically correct and logically incorrect decisions. As applied to the task of forecasting product life cycle phase transition period, logically correct and logically incorrect decisions could be described as follows:

1. Assume that discrete time series d has a duration equal to l_d, but the value of the key parameter – the period of product life cycle phase transition – is $p = p_d$, where $p_d > l_d$. This statement means that the real time of transition between the phases of the product life cycle has not come yet. Accordingly, logically correct decision is to forecast transition period r_d, where $r_d > l_d$. Logically incorrect decision in this case will be if $r_d \le l_d$.
2. Assume that discrete time series d has a duration equal to l_d, but the value of the key parameter – the period of product life cycle phase transition – is $p = p_d$, where $p_d \le l_d$. This statement gives evidence that the real transition moment has already come. Accordingly, logically correct decision could be forecasting transition period r_d, where $r_d \le l_d$. In its turn, logically incorrect decision will take place if $r_d > l_d$.

The statement that at $r_d = l_d$ transition has occurred can be considered correct as the availability of data about some period in record d shows that the period is logically finished and, consequently, the transition – if any was assumed in this period – has occurred.

3.4.3. Functional aspects of Decision Analysis Agent

The Decision Analysis Agent can perform its actions either by receiving a request from a user, or in autonomous mode, with defined interval of time (at the end of each period) reporting the decision analysis results. Products that are analysed by Decision Analysis Agent are products that are still evolving on the market. The list of such products is monitored by the Data Management Agent.

Either in autonomous mode or by request from a user the Decision Analysis Agent starts the process displayed in Figure 3. The depicted process includes three main steps – Determination of the Best Matching Cluster (BMC) for each of the evolving products; Formation of the List of Interest (LOI), the list of products for which it would be reasonable to reconsider the planning policy; Evaluation of a cyclic and non-cyclic planning policy for each product on the List of Interest. And then the fourth step comes – Reporting the results of the evaluation to a user.

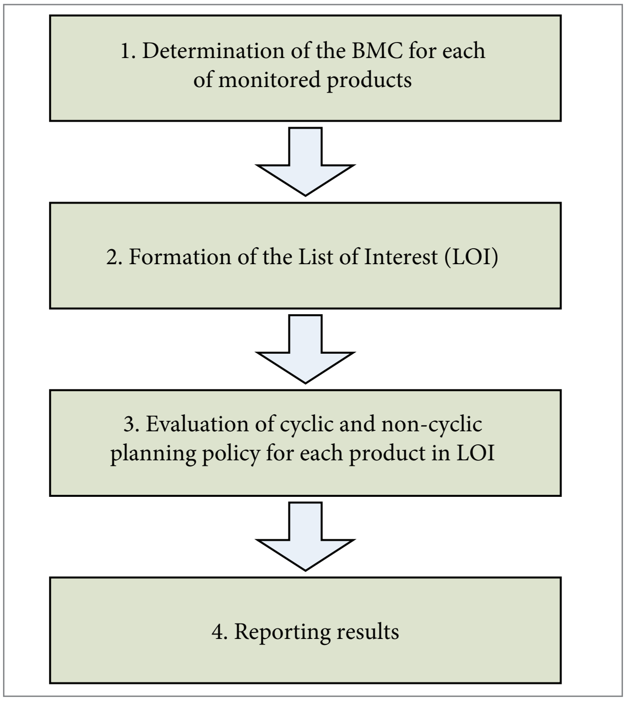
1. Determination of the BMC for each of monitored products
2. Formation of the List of Interest (LOI)
3. Evaluation of cyclic and non-cyclic planning policy for each product in LOI
4. Reporting results

Fig. 3. Decision Analysis Agent functioning diagram

Let us describe the processes hidden behind each of the main steps:

Step 1: Determination of the BMC for each of the evolving products. The Decision Analysis Agent sends a request to the Data Management Agent and receives a dataset containing evolving products. Each record is preprocessed and formatted by the Data Management Agent. Once the dataset is received it is sent to the Data Mining Agent with command "Find Best Matching Cluster".

The Data Mining Agent searches the knowledge base for the Best Matching Cluster for each of the demand time series and returns a list of found clusters to the Decision Analysis Agent. The information from the BMC contains a list of possible transition points for each analysed product – for the products in the introduction phase the M1 transition point is supplied and M2 transition point – for products in the maturity phase. This is where the Formation of the LOI begins.

Step 2: Formation of the List of Interest. The Best Matching Cluster may contain different information for products and several cases are possible:

1. The simplest case ($C1$) when the Best Matching Cluster contains only one possible transition point. In this case the Decision Analysis Agent assumes this transition point as preferable and follows a solution ($S1$) containing three major rules:
    a) If $l < p$ and $p - l > \theta$. Then: the product remains monitored and is not included in the List of Interest;

b) If $l < p$ and $p - l < \theta$. Then: the product is included in the List of Interest;

c) If $l > p$. Then: the product is included in the List of Interest.

Where l is the duration of the demand time series in periods; p - a forecasted transition point; and variable θ stores the minimal threshold of interest for either including the product in the List of Interest or not.

2. The case (C2) is when the BMC contains more than one possible transition point for a demand time series, but one of transition points has an expressed appearance frequency f. An appearance frequency may be stated as expressed if it exceeds some threshold, like 50%. In such case the solution (S2) will be that the Decision Analysis Agent accepts the transition point with an expressed f as the preferable one and follows rules from the S1 solution.
3. The third possible case (C3) is when the BMC contains several possible transition points, but there is not one with an expressed appearance frequency present. In this case several solutions are possible:
   a) Solution S3. The Decision Analysis Agent asks the Data Mining Agent if the neural network n_i where the BMC for a current product was found is the only one network in the Neural Block or if it is the network that processes time series with maximal possible duration. If so, then the Decision Analysis Agent follows the next two rules:
      i. If only one transition point has the highest (not expressed) f, then select it as a preferable one and follow rules from solution S1;
      ii. If several transition points have the highest f, then select a transition point with a minimal value as preferable and follow rules from solution S1. Thus if transition points with the highest f are 4th, 5th and 8th period, then the Decision Analysis Agent will choose the 4th period.

   b) Solution S4. If the solution S3 was not triggered, then the Decision Analysis Agent follows the next strategy:
      i. Decision Analysis Agent gives the Data Mining Agent a command to find the Best Matching Clusters in all networks n_j for a current product, where $j > i$ and i is the index of the network with current BMC. That means to search for a BMC in all other networks that process time series with duration larger than the duration of a current product, excluding the current network n_i;
      ii. The Data Mining Agent searches for BMCs in the knowledge base using only the first W' synaptic weights, where $w' = w_i$ and w_i is the number of synaptic weights in the current network n_i;
      iii. The list of found BMCs is returned to the Decision Analysis Agent;
      iv. The Decision Analysis Agent adds the BMC from the current neural network n_i to the list, selects the most matching cluster and checks which of three cases – C1, C2 or C3, is triggered. If case C1 or C2 is triggered, then the Decision Analysis Agent just follows the rules from solutions for those

cases (solutions S1 and S2 respectively). In case when the C3 is triggered again, the Decision Analysis Agent follows rules from the S3 solution.

An example of such situation may be displayed as follows. Let us assume that the Neural Block contains two neural networks: first network n_1 processes time series with duration 3, 4 and 5 periods and each neuron in n_1 has 4 synaptic weights; second network n_2 is able to process time series with 6, 7 and 8 and each neuron in n_2 has 7 synaptic weights. The current product falls into the n_1 network and Decision Analysis Agent chooses to check the next networks (to use solution S4). In this case the BMC will be searched in the network n_2, but only first 4 of 7 synaptic weights will be used to determine the best matching cluster in the n_2.

If at the end of formation of the List of Interest the list is empty, then the Decision Analysis Agent bypasses the third step and reports that products for which it would be reasonable to reconsider the planning policy were not found. In case LOI contains at least one product, the Decision Analysis Agent starts processes in the third step.

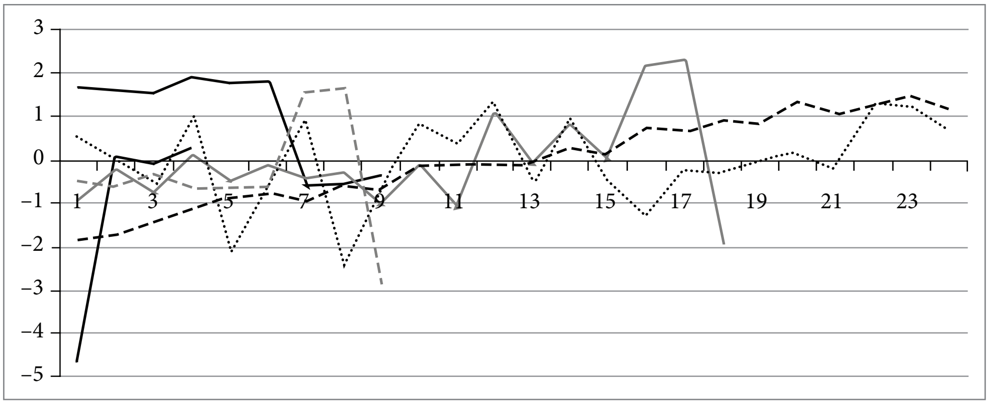

Fig. 4. Example of normalized data

Step 3: Evaluation of cyclic and non-cyclic planning policy for each product in LOI. At this step the Decision Analysis Agent measures the expenses of using cyclic or non-cyclic planning policy for each product on the List of Interest. As stated in (Campbell and Mabert 1991), the measure of Additional Cost of a Cyclic Schedule (ACCS) may be used for those purposes. The ACCS measures the gap between cyclic and non-cyclic planning policies, and it is calculated by formula (9):

$$ACCS = \frac{CPPC - NCPPC}{NCPPC}, \tag{9}$$

where CPPC is the Cyclic Planning Policy Cost and NCPPC – the Non-Cyclic Planning Policy Cost.

As the third step is finished, the user receives the results of analysis of the products from the List of Interest.

4. Basic research results

The fact that the data describes real life process and marks of transitions were put by experts implies that some noisiness in the data is present.

The obtained dataset contains 199 real product demand time series whose minimal duration equals 4 and maximal – 24 periods. Each time series contains the demand during the introduction phase of a specific product plus one period of the maturity phase, and is marked with M1 marker. To normalize the data, the Z-score with standard deviation normalization method was applied. As the true bounds of the demand data in the dataset are unknown and the difference between values of various time series is big, the chosen normalization method is one of the most suitable ones.

Figure 4 displays an example of time series data used in experiments. As can be seen, the time series differs not only in duration, but also in amplitude and its pattern.

The main target of the performed experiments was to comparatively analyse the precision of forecasting transition points with square neural network topology with 8 neighbours applied while using different network load *q*. The more precise the forecasted transition point, the more precise the result returned by the Decision Analysis Agent.

Network load *q* was changing incrementally from one to five. To calculate the system errors - Mean Absolute Error (MAE) and Logical Error (LE), a 10-fold cross validation method was applied, totally giving 50 system runs.

Table 1 contains the size of the network, the total number of neurons, and also supplies the number of iterations for initial organization and convergence process.

Table 1. Network and learning parameters

Size	**Neurons**	**Initial organization**	**Convergence process**
5 x 5	25	1000 iter.	12500 iter.

The learning parameters used for network organisation in each run are given in Table 2. For each learning parameter the starting value and the minimal (last) value are supplied, as well as the type of function used for managing the parameter decline process.

Table 2. Learning coefficients

Parameter	**Starts with**	**Ends with**	**Function**
Learning coefficient - η	0.9	0.01	Exp.
σ for Gaussian neighbourhood	0.5	0.01	Exp.

Table 3 gives measurements of Mean Absolute Error and Logical Error (see paragraph 3.4.2) gathered during experiments with the system while changing the individual network load – *q*.

Table 3. On-line Mean Absolute Error – MAE and LE

Err.	Topology	q = 1	q = 2	q = 3	q = 4	q = 5
MAE	SQR-8	2.0	2.1	2.4	2.2	2.4
LE (%)	SQR-8	13.5	11.7	14.5	12.3	17.0

The results obtained show that the created agent-based system is able to predict transition points for new products with certain precision using a model built on historical demand data. The system was able to make logically correct (refer to paragraph 3.4.1) decisions in at least 83.0% and at most in 87.7% of times. Thus the Mean Absolute Error lies close to 2 periods. Together with the specificity of the dataset obtained and the size of the network, it is possible to conclude that the created agent-based system can be used as a data mining tool to gain additional knowledge for solving a planning policy management task. It can also be applied to doing other tasks connected with forecasting the value of a target parameter for a time dependent variable, followed by a Decision Analysis process.

5. Conclusions and future work

For practitioners of management of the product life cycle the knowledge of the phase the product is currently in and when the transition between phases will occur, is topical. Such knowledge, in particular, helps to select either the cyclic or non-cyclic policy of planning a supply chain operation.

In this paper, the task of forecasting the transition points between different phases of the product life cycle is stated, and the structure of Agent-Based Production Planning Support system, which helps to solve this task, is shown. The functional aspects of the Decision Analysis Agent are described. Experimentally gathered results show that the created Agent-Based Production Planning Support system has some potential and can process real demand data, create a model on the basis of historical data, forecast possible transition points and theoretically analyse expenses for cyclic and non-cyclic planning policies.

For future research it is necessary to examine the developed system on data from different production fields and, which is also important, to get a response from practitioners of supply chain management who will use these systems. The examination of other technologies for creating the knowledge base in the Data Mining Agent will be performed.

Another important point is that the modest data volume that was used for practical experiments is related to the fact that it is necessary to have transition marks in historical data from experts and practitioners. The more products, the more complicated for human is to make all these marks – in practice the amount of marked data will always be restricted. As a result, another possible direction of future research is treatment of forecasting the transition points in the context of a semi-supervised learning (Zhu 2008). In this case, there is a small set with marked transitions and also a large dataset in which transitions are not marked. In that kind of situation it is necessary to create a model which will be able to apply the knowledge gathered on a small set of marked data to new (test) data.

References

Akanle, O. M.; Zhang, D. Z. 2008. Agent-based model for optimising supply-chain configurations, *International Journal of Production Economics* 115(2): 444–460. doi:10.1016/j.ijpe.2008.02.019

Athanasiadis, I. and Mitkas, P. 2004. An agent-based intelligent environmental monitoring system, *Management of Environmental Quality: An International Journal* 15(3): 238–249.doi:10.1108/14777830410531216

Campbell, G. M. and Mabert, V. A. 1991. Cyclical schedules for capacitated lot sizing with dynamic demands, *Management Science* 37(4): 409–427. doi:10.1287/mnsc.37.4.409

Chakrabarti, S.; Cox, T.; Frank, H.; Güting, R. H.; Han, J.; Jiang, X.; Kamber, M.; Lightstone, S. S.; Nadeau, T. P.; Neapolitan, R. E.; Pyle, D.; Refaat, W.; Schneider, M.; Teorey, T. J.; Witten, I. H. 2009. *Data mining: know it all*. Morgan Kaufmann Publishers, an imprint of Elsevier.

Dreyfus, G. 2005. *Neural Networks. Methodology and Applications*. Springer Verlag, Berlin Heidelberg.

Dunham, M. 2003. *Data Mining Introductory and Advanced Topics*. Prentice Hall.

Ferber, J. 1999. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Pearson Education.

Han, J. and Kamber, M. 2006. *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufman, 2nd edition.

Hand, D. J.; Mannila, H. and Smyth, P. 2001. *Principles of Data Mining*. The MIT Press.

Keogh, E. and Pazzani, M. 2001. Derivative dynamic time warping, in *Proc. of the First SIAM International Conference on Data Mining*, Chicago, USA.

Kohonen, T. 2001. *Self-Organizing Maps*. Springer, 3rd edition.

Kotler, P. and Armstrong, G. 2006. *Principles of Marketing*. Prentice Hall, 11th edition.

Merkuryev, Y.; Merkuryeva, G.; Desmet, B. and Jacquet-Lagreze, E. 2007. Integrating analytical and simulation techniques in multi-echelon cyclic planning, in *Proc. of the First Asia International Conference on Modelling and Simulation*, 460–464. IEEE Computer Society.

Obermayer, K. and Sejnowski, T. (Eds.). 2001. *Self-Organising Map Formation*. The MIT Press.

Reaidy, J.; Massotte, P.; Diep, D. 2006. Comparison of negotiation protocols in dynamic agent-based manufacturing systems, *International Journal of Production Economics* 99(1–2): 117–130. doi:10.1016/j.ijpe.2004.12.011

Symeonidis, A.; Kehagias, D. and Mitkas, P. 2003. Intelligent policy recommendations on enterprise resource planning by the use of agent technology and data mining techniques, *Expert Systems with Applications* 25(4): 589–602. doi:10.1016/S0957-4174(03)00099-X

Tan, P.-N.; Steinbach, M. and Kumar, V. 2006. *Introduction to Data Mining*. Pearson Education.

Wooldridge, M. 2005. *An Introduction to Multiagent Systems*. John Wiley & Sons, Ltd.

Zhu, X. 2008. *Semi-supervised learning literature survey*. Technical Report 1530, Department of Computer Sciences, University of Wisconsin.

AGENTINĖ GAMYBOS PLANAVIMO PARAMOS SISTEMA

S. Parshutin, A. Borisov

Santrauka. Gamybos planavimas – pagrindinis aspektas, nuo kurio priklauso bendrovės pajamos. Teisingos gamybos planavimo politikos pasirinkimas konkretaus gaminio gyvavimo ciklo (GGC) metu mažina gamybos, sandėliavimo ir kitas išlaidas. Reikia nustatyti esamą gaminio gyvavimo ciklo etapą ir perėjimo tašką – laiko momentą, kai pasikeičia gyvavimo ciklo etapas. Pateikiama agentinė gamybos planavimo paramos sistema, skirta gamybos vadovo planavimo sprendimams pagrįsti. Sukurtoji sistema pagrįsta retrospektyvine produktų paklausos analize ir informacija apie tų produktų gyvavimo ciklo fazių pokyčius. Sistema kuria modelį ir taiko jį prognozuojant produkto, kuris tik pradeda savo gyvavimą rinkoje, perėjimo tašką. Pateikiama sukurtos sistemos struktūra ir bandymų su realiais duomenimis analizės rezultatai.

Reikšminiai žodžiai: agentai, duomenų gavimas, sprendimų parama, gamybos planavimas, produkto gyvavimo ciklas, perėjimo taškų prognozavimas.

Serge PARSHUTIN is Ph.D. student at the Faculty of Computer Science and Information Technology of Riga Technical University (Latvia) and a Lecturer with the Department of Modelling and Simulation at the Riga Technical University. He received his MSc degree in Information Technology from Riga Technical University in 2006. He participated in the European research project ECLIPS (2006-2009, www.eclipsproject.com). His research interests include data mining and knowledge extraction, intelligent information systems, evolutionary computing, decision support, soft computing and agent technologies.

Arkady BORISOV is Professor of Computer Science in the Faculty of Computer Science and Information Technology at Riga Technical University (Latvia). He holds a Doctor of Technical Sciences degree in Control in Technical Systems and the Dr.habil.sci.comp. degree. He has supervised a number of national research grants and participated in the European research project ECLIPS. His research interests include fuzzy sets, fuzzy logic and computational intelligence. He has 205 publications in the area.