RIGA TECHNICAL UNIVERSITY

Serge PARSHUTIN

DEVELOPMENT OF THE PRODUCT LIFECYCLE MANAGEMENT SUPPORT SYSTEM ON THE BASIS OF INTELLIGENT AGENT TECHNOLOGY AND DATA MINING METHODS

Ph.D. Thesis Summary

Riga 2011

RIGA TECHNICAL UNIVERSITY
Faculty of Computer Science and Information Technology
Information Technology Institute

Serge PARSHUTIN
Ph.D. student of the Management information technology programme

DEVELOPMENT OF THE PRODUCT LIFECYCLE MANAGEMENT SUPPORT SYSTEM ON THE BASIS OF INTELLIGENT AGENT TECHNOLOGY AND DATA MINING METHODS

Ph.D. Thesis Summary

Scientific advisor
Dr.habil.sc.comp., professor
A. BORISOVS

Riga 2011

UDK 004.89(043.2)
Pa 607 i

Parshutin S. Development of the product lifecycle management support system on the basis of intelligent agent technology and data mining methods. Ph.D. Thesis Summary.-Riga: RTU, 2011. -36 p.

Printed according to the decision Nr. 11-02 of RTU Promotion Council P-07 meeting, May 26, 2011

ISBN: 978-9934-10-168-7

This Ph.D. Thesis has been supported by the European Social Fund within the project "Support for the implementation of doctoral studies at Riga Technical University".

PH.D. THESIS

IS NOMINATED IN RIGA TECHNICAL UNIVERSITY FOR TAKING A DOCTOR'S DEGREE IN ENGINEERING SCIENCE

Defence of the Ph.D. Thesis for obtaining a doctor's degree in engineering science will take place on September 5, 2011 at Riga Technical University, Faculty of Computer science and Information Technology, Mezha street 1/3, room 202.

OFFICIAL OPPONENTS

Professor, Dr.habil.sc.ing. Jānis Grundspeņķis
Riga Technical University, Latvia

Professor, Dr.sc.ing. Oļegs Užga-Rebrovs
Rezekne Higher Education Institution, Latvia

Professor, DSc Mikhail Kovalyov
United Institute of Informatics Problems, Belarus

DECLARATION

I hereby declare that I have written this Thesis, which is submitted for reviewing in Riga Technical University for taking a doctor's degree in engineering science. This thesis has not submitted to any other university for taking a scientific degree.

Serge Parshutin …………………………….(signature)

Date: ………………………

Ph.D. thesis is written in Latvian, it contains an introduction, 5 chapters, conclusions, a list of references, 5 appendixes, 47 figures, 131 pages in total. There are 61 references.

GENERAL DESCRIPTION OF THE THESIS

Topicality of problem

The technological progress is constantly evolving; due to that modern information technologies bring new and different management features. Companies tend to implement different management systems in order to increase efficiency of planning and manufacturing processes, reducing the expenses at the same time. One of the topical tasks for the companies that manufacture and trade different products is the precise planning, which is an integral part of the product lifecycle management. This problem remains topical as the product lifecycle affects the manufacturing and planning strategies applied and the related expenses.

Product lifecycle management is a strategic approach to solve and/or overcome problems connected with different processes of a product lifecycle [41, 44]. Product lifecycle represents the product demand curve in the time period from the introduction of the product till its withdrawal from the market. Product lifecycle includes different phases that differ in demand level and its trend, and in expenses level. Most of the methods that are applied to determine the product lifecycle phase transition points are based on forecasting demand and applying an expert experience to define transition points.

The PLC phase transition point forecasting task is not new, though it still remains topical. Various scientists propose different solutions to this problem. The majority of solutions support answers to questions like “What the product demand will be?” and “Has the product passed to the next phase?”, but not to the question: “After which period the product will change the phase to the next one?”. The data mining methods support all the functionality to mine the knowledge base from statistical data, which will represent the relationships between product demand curve and a transition point. Intelligent agent technology (IAT) is fully applicable to designing and producing the complex systems, and in combination with data mining methods can be applied to the development of a multiagent PLC management support system, which will automatically mine and update the knowledge base, tracks the state of multiple products and timely provides managers with information on transition points, including cases when only the several first points of the demand curve are accessible.

Goals of research

The aim of the Ph.D. Thesis is to study the short time series clustering methods and to develop a model of the multiagent product lifecycle phase transition point forecasting system on the basis of the intelligent agent technology and data mining methods, capable of processing the successive data. More specifically, the following subtasks should be solved:

1. To study the concepts of the intelligent agent technology and define specific to the object of the research definitions of an agent and multiagent system.
2. To develop a model of the multiagent system capable of autonomous mining of the relations between product demand curve and the product lifecycle phase transition point, and forecasting the transition point to the new products.
3. To develop and describe the functioning algorithms of agents in the multiagent system with the ability to interact.
4. To study the short time series clustering methods and to develop the modifications in order to support the clustering of short time series of different length.
5. To develop a specialized software tool which would implement the proposed modified clustering methods, and apply it to measuring the effectiveness of the proposed multiagent PLC transition point forecasting system. To make a comparative analysis of the practical results obtained.

6. The data pre-processing methods should be studied and the data preparation should be performed.

Object of the research

The object of the research is the combination and modification strategies of the intelligent agent technology and the data mining methods, aimed at solving the tasks where application of the individual method or technology would be inefficient.

During the research the following hypotheses were made

1. Intelligent agent technology offers wide possibilities for formalization of system structure, processes and information flows. Application of the IAT contributes to the complex management systems design process.
2. Using the data mining methods it is possible to mine (produce) a model of the product lifecycle that can be applied to the new product lifecycle analysis.

Methods of the research

Intelligent agent technology, data mining and soft computing methods and their modifications and theory of information are used in the Ph.D. Thesis. Besides, self-organizing maps, hierarchical gravitational clustering algorithm, data pre-processing methods as well as distance measures: dynamic time warping (*DTW*) and the one proposed – *MEuclidean*, are employed.

Scientific novelty

1. The product lifecycle phase transition point forecasting method is proposed on the basis of the intelligent agent technology and the data mining methods. It was implemented by the proposed multiagent system.
2. The modification of the self-organizing maps algorithm is developed. The new algorithm supports the clustering of the short time series of different length.
3. The modification of the hierarchical gravitational clustering algorithm is developed. The new algorithm supports the clustering of the short time series of different length.
4. The modification of the *Euclidean* distance measure is proposed, which supports the distance calculation between time series of different length.

Practical use of the thesis and approbation

The ideas and concepts of the intelligent agent technology are studied within this doctoral thesis and in combination with the proposed modifications of clustering methods are applied to the development of the multiagent PLC management support system. The developed system was applied to forecasting the product lifecycle phase transition points using the data from the international chemical production company *HUNTSMAN* that were obtained in the scope of the international project of the European Community 6th Framework Programme – *ECLIPS* project. The efficiency of the proposed system was tested applying each of the developed clustering algorithm modifications.

The result of the system work is the automatically generated knowledge base, which contains a model of the relations between the product demand curve and the transition point. The knowledge base was used to forecast the transition points to the new products. The best results were obtained using the multiagent system with modified gravitational clustering algorithm. A specialized software tool was developed using the *MS Visual Basic'10* in order to perform practical experiments aimed at measuring the efficiency of the proposed system.

The results of the Ph.D. Thesis were presented at the following scientific conferences:

1. *Parshutin S. Managing product life cycle with multiagent data mining system // Industrial Conference on Data Mining, ICDM'10 Germany, Berlin, July 12-14, 2010.*
2. *Parshutin S., Borisov A. Data Mining Driven Decision Support // 16th International Multi-Conference ADVANCED COMPUTER SYSTEMS, ACS'2009, Poland, Miedzyzdroje, October 14-16, 2009.*
3. *Parshutin S., Aleksejeva L., Borisov A. Forecasting Product Life Cycle Phase Transition Points with Modular Neural Networks Based System // Industrial Conference on Data Mining, ICDM'09, Germany, Leipzig, July 20-22, 2009.*
4. *Parshutin S., Aleksejeva L., Borisov A. Time Series Analysis with Modular Neural Nerworks // RTU 49th International Scientific Conference, Latvia, Riga, October 13, 2008.*
5. *Parshutin S., Kuleshova G. Time Warping Techniques in Clustering Time Series // 14th International Conference on Soft Computing, MENDEL 2008, Czech Republic, Brno, June 18-20, 2008.*
6. *Parshutin S. Clustering Time Series of Different Length Using Self-Organising Maps // RTU 48th International Scientific Conference, Latvia, Riga, October, 2007.*
7. *Parshutin S.V. Clustering time series with Self-Organizing maps // International conference on Soft Computing and Integrated models in Artificial Intelligence (IMSCAI'2007) – May 28-30, Kolomna, Russia, 2007. In Russian.*

Main results of the Ph.D. Thesis

The intelligent agent technology and data mining methods are studied and analysed, the product lifecycle management support system is developed and evaluated. The main results of the Ph.D. Thesis are the following:

1. A multiagent product lifecycle management support system is developed. The system supports a manager with additional information about the market state of a product and progress directions, what further well-reasoned decisions to be taken while choosing planning, manufacturing and advertising strategies. It is concluded that by combining the intelligent agent technology with data mining methods it is possible to develop complex management and management support systems.
2. An analysis of the concepts of the intelligent agent technology is done. The obtained results point out that the application of the IAT contributes to the complex management systems design process.
3. Based on the main tasks of the thesis and defined specifications of the multiagent system the functioning algorithms for data management agent, data mining agent and decision support agent are developed. The obtained results show that the application of the intelligent agent technology contributes to the formalization of the structure and internal processes of the system.
4. An analysis of popular clustering method for suitability to cluster the product demand data is performed. The concepts of suitable self-organising maps and the gravitational clustering algorithm are analysed and described. An analysis of the algorithms for suitability to cluster the short time series of different length is performed. The obtained results point out that the classical self-organising maps and the gravitational clustering algorithm do not support an option to cluster the short time series of different length.

5. Modifications to the self-organising maps and the hierarchical gravitational clustering algorithm are developed. The modifications support an option of clustering time series of different length; the structural and procedural modifications of the algorithms were made.
6. The *MEuclidean* distance measure is proposed. This measure supports the calculation of a distance between time series of different length.
7. A specialized software tool is developed that realizes the conceptual and procedural structure of the proposed multiagent system and implements the developed modifications of the self-organising maps and gravitational clustering algorithm.
8. Using the developed software tool the approbation of the proposed system is performed and a comparative analysis of the proposed modified clustering algorithms and distance measures is done. The analysis of the system testing results confirms that the proposed multiagent system is capable of mining the knowledge from data, creating a knowledge base and applying it for analysis of new data in the autonomous mode with comparatively high accuracy.
9. An analysis of the obtained system testing results is performed. The results demonstrate that the data mining methods can be used for building the model of the product lifecycle that can be applied for analysis of PLC of the new products.

Research results of the Ph.D. Thesis were applied in the following projects:

1. Latvia – Belarus Co-operation Programme in Science and Engineering, Scientific Cooperation Project No. L7631, "Development of a complex of intelligent methods and medical and biological data processing algorithms for oncology disease diagnostics improvement" (Years 2010-2011).
2. Latvian Council of Science research grant No. 09.1564, "Simulation and computational intelligence methods for logistics and e-business optimization", Latvian Council of Science (Years 2010 – 2012).
3. Latvian Council of Science research grant No. 09.1201, "Simulation-based optimisation using computational intelligence", (Year 2009).
4. RTU scientific project No. ZP-2008/7 „Computational Intelligence Methods in Classification Tasks" (Years 2008-2009).
5. RTU scientific project No. ZP-2007/05 „Computational Intelligence Methods in Data Mining Tasks", (Years 2007-2008).
6. *European Project ECLIPS* (Extended Collaborative Integrated Life Cycle Supply Chain Planning System) of the European Community Sixth Framework Programme (Years 2006 - 2008).
7. Latvian Council of Science research grant No. 05.1639, "Intelligent computer technologies for ill-formalised decision-making tasks" (Years 2005 – 2008).
8. IZM - RTU scientific project R 7085. „Artificial intelligence in tasks of forecasting and control" (Year 2006).

Publications

The results of the Ph.D. Thesis have been published in 11 scientific papers. The list of publications can be found at the end of this summary. The research process has been supported by the European Social Fund within the project "Support for the implementation of doctoral studies at Riga Technical University".

Structure of the Ph.D. Thesis

The thesis contains an introduction, 5 chapters, conclusions, a list of references, 5 appendixes, 47 figures, 131 pages in total. There are 61 references.

The structure of the thesis is the following:

INTRODUCTION – presents the global information on the object of the research and defines the goals of the research.

CHAPTER 1 brings an introduction to the field of research; the main problem that is intended to be solved is defined and the concepts of the intelligent agent technology are studied and described. The chapter gives an insight to the features and architectures of an agent. The multiagent systems are compared to such fields like "Distributed/concurrent systems", "Artificial intelligence", "Economics/Game theory" and „Social science". The formalized problem statement concludes the chapter.

CHAPTER 2. In this chapter the motivation to application of IAT for solving the tasks, defined in the Ph.D. Thesis, is given. The model of the multiagent PLC management support system is proposed, its structure and features are described. The concepts and functional algorithms of each agent in the proposed system are described.

CHAPTER 3. The applied clustering algorithms are described and analysed in this chapter – self-organizing maps and gravitational clustering algorithm. The description to the proposed modifications of clustering algorithms that extend features of the classic algorithms with ability of clustering short time series of different length is given. The description of proposed and applied distance measures and also of the data pre-processing techniques are given in this chapter.

CHAPTER 4 describes the specialized software tool that was developed for experimental evaluation of efficiency of the proposed system.

CHAPTER 5 is devoted to the experimental approbation of the developed system. The chapter starts with the description of the data used. It is followed by the evaluation of the system learning error, results of which are applied to definition of the reliable strategy for system testing. The analysis of the obtained testing results and of the system efficiency concludes this chapter.

SUMMARY OF THESIS CHAPTERS

Chapter 1

1.1. Introduction to the field of research

Product lifecycle management is a strategic approach to solve and/or overcome problems connected with different processes of a product lifecycle [11, 25, 41, 44]. It is a complex process with multiple tasks, the list of which is highly dependent on individual needs of a company, which is using or is planning to implement the product lifecycle management system [41, 44]. The aim of the multiagent system, proposed in the thesis, is to support forecasting and planning process, the main objective of which is to provide the information about the product status on the market and possible trends of the current state. Availability of such information promotes production of more precise strategies for manufacturing, management, product promotion and others, which lessens expenses and improves total efficiency of a company.

Product lifecycle (Figure 1.1) represents the product demand curve from the introduction of the product till its withdrawal from the market. Product lifecycle consists of the known phases that represent the demand in a specific time interval. Most of the methods that are applied to determine the product lifecycle phase transition points use different dynamic linear models [53, 54] or the Bass diffusion model or modifications [9, 10, 46, 52, 53]. The main idea of those methods is to forecast a product demand curve and to apply a manager experience to define the bounds of the PLC phases. The efficiency of such methods may fall while the number of managed products arises. The main objective of such trend is that a continuous support of an expert team may result in expenses much higher than those associated with implementing an automated product lifecycle phase transition point forecasting system.

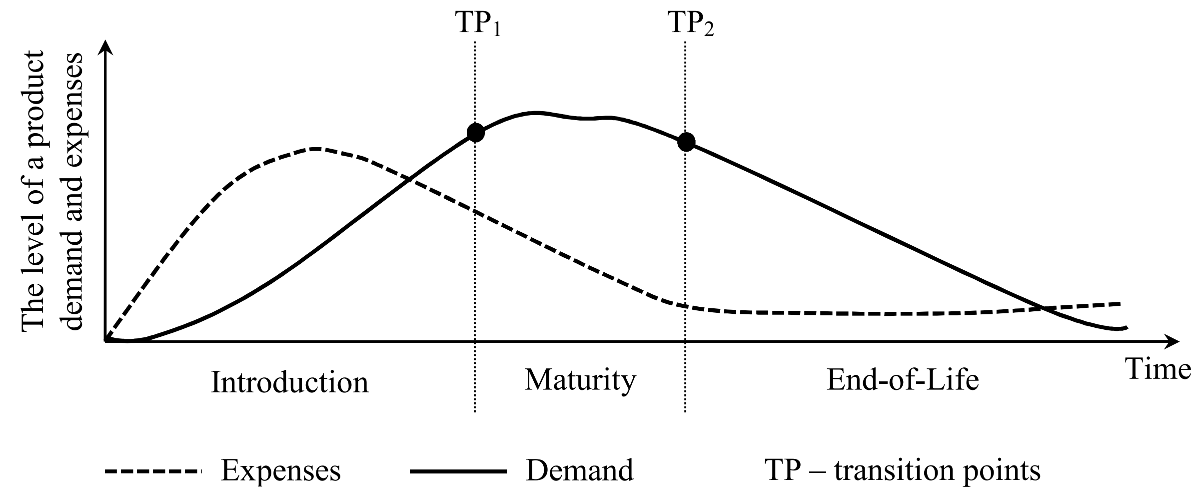


Figure 1.1. Phases of the product life cycle and a level of expenses

The mentioned product lifecycle phase transition point determination methods are capable to forecast a product demand with high accuracy; nevertheless, those methods can only answer the question: “What will be the product demand”, not the “After which period the product will change the phase to the next one?”. Availability of information about the transition points at the beginning of a product lifecycle supports managers with valuable data on product growth directions, thus supporting the development implementation of a motivated product and company growth strategy.

1.2. Concepts of an intelligent agent technology

The agent is a hardware or (more usually) software-based computer system that has the following characteristics [8, 50]:

1) **autonomy** – agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;
2) **social ability** – agents interact with other agents (and possibly humans) via some kind of agent-communication language;
3) **reactivity** – agents perceive their environment, (which may be the physical world, user, a collection of other agents, the Internet, or perhaps all of this combined), and respond in a timely fashion to changes that occur in it;
4) **pro-activeness** – agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative.

Agent exists in some kind of environment. An environment affects an agent through changing its state, and by performing the defined actions an agent affects the environment.

There are clear similarities between objects and agents, but obvious differences also exist. The first one is the degree to which agents and objects are autonomous. Regarding the object case, the control over performing some action (method) remains in the scope of an entity that is called the specific method. In the agent case the control over the decision whether to perform the action or not remains in scope of an agent. In general, it can be described with citation from the book [50] – „*Objects do it for free; agents do it because they want to*". On the other hand, it is possible to develop an agent with an object-oriented tool, implementing a procedure that will support the decision on running one of the defined methods, but this type of autonomy is not a part of the classical object model [8, 50]. The second important distinction between object and agent systems is with respect to the notion of flexible (reactive, proactive, social) autonomous behaviour. The standard object model has nothing whatsoever to say about how to build systems that integrate these types of behaviour [50]. The third important distinction between the standard object model and the agent systems is that agents are each considered to have their own thread of control – in the standard object model, there is a single thread of control in the system. There are many programming languages available that support the concurrent programming, but such languages do not capture the idea of agents as autonomous entities [50]. M.Wooldridge [50] communicates on the results of a comparison of multiagent systems to such field as "Distributed/concurrent systems", "Artificial intelligence", "Economics/Game theory" and „Social science". Similarities and sufficient differences are presented by Wooldridge.

1.3. Formalized problem statement

Analysing the processes of the product lifecycle management and the PLC transition point forecasting task it becomes possible to formalize the problem statement of the Ph.D. Thesis. The strategy of using the knowledge, mined from the statistical data, for analysis of a new data can be applied to automate the process of forecasting the PLC phase transition point. To fulfil this task it is necessary to build an autonomous system that selects the data from the repository, pre-processes it, then autonomously builds and updates the knowledge base and applies it to forecast the transition point for a new product. It is important to support the forecasting of a transition point in the case when only several first points of a demand curve are available.

The structure of the transition point forecasting system should be easy-to-understand; this option contributes to the design of the system at a more detailed level. This can be fulfilled by application of the intelligent agent technology (IAT), which offers sufficient functionality for designing the complex systems and distributing the process flow among agents in the multiagent system. The application of IAT provides an option for creating an autonomous agent for each of processes of data management, data mining and decision support. Each agent has its own objectives, but the information flow is present among agents - there is an interaction among agents. It is necessary to study and analyse the concepts of the intelligent agent technology that can be implemented in the multiagent system.

The data mining methods have all the necessary functionality for mining the knowledge base from the statistical data. The demand data are sequential data that are represented as short time series. It is necessary to perform a comparative analysis of the time series clustering methods and select the suitable ones. In most cases the data mining methods partly support the requested functionality, due to that it is necessary to develop a modifications of clustering methods that will support clustering of time series of different length and forecasting the transition point while only several first points of the demand curve are available.

In order to evaluate the efficiency of the proposed multiagent system it is necessary to develop a specialized software tool, which will implement the structure and functionality of the proposed system and developed modifications of the clustering methods (see chapter 4). It is necessary to present a list of experiments for trustworthy evaluation of the system. The experiments must be made, the results obtained must be analysed and conclusions on the efficiency of the proposed system must be made.

Chapter 2

Nowadays the agent technology is present in many industries including concurrent engineering [55], collaborative engineering design, manufacturing enterprise integration, supply chain management [5, 17, 38], manufacturing planning, scheduling and control, material handling, and holonic manufacturing systems [8].

N. Jennings [20] indicated the flexible, high-level interactions of agents that make the engineering of complex systems easier. Multiagent systems provide designers with means for it through the decentralisation of control. Multiagent decentralisation provides an ability to control each of iterations in the system by letting each agent continuously coordinate its actions with other agents. There is no universally accepted definition on what an agent is, thus this definitions are used in the Thesis:

- **An agent** is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.
- **An intelligent agent** is an agent with ability to learn. It is capable of mining, processing, storing and applying the knowledge about the environment it is situated in.
- **Multiagent system** contains agents and intelligent agents that can communicate, exchanging with different information.
- **Agent community** is an indivisible entity that contains agents with common attributes and behaviour. Comparing with multiagent systems, agents in the agent community are not communicating with one another.

2.1. The structure and concepts of the proposed multiagent system

Figure 2.1 presents a model of the proposed multiagent system. There are three tasks that were pointed out for the system:

1. The multiagent system must provide an access to the data and perform all the necessary data pre-processing steps.
2. By performing the intelligent data analysis the system must create a knowledge base containing a model of relations between the demand curve profile and the value of a PLC phase transition point.
3. While forecasting the transition point for a new product, the knowledge base must be used.

Those three tasks are not overlapping with each other, thus are interconnected aimed at the main target of forecasting the transition point for new products. This provides an option of assigning each task an individual agent and link agents in the multiagent system. It can be stated that such an option is effective, as the task control is decentralised and the information flow is provided through the agent communication.

The data management agent provides the access to the database and performs all the necessary data pre-processing steps. With respect to the agent architectures, provided in the book [40], the data management agent is closer to the target-based agent than to the others. This is correct as after obtaining data from the database, the agent analyses the data and defines the pre-processing steps that are necessary to perform for a particular sample of an unprocessed data.

The data mining agent uses data mining methods to mine the knowledge from the statistical data and creates the knowledge base. The data mining agent is a learning agent, as it has abilities to check the precision and the effectiveness of the knowledge base and to perform changes in the learning parameters in order to improve the performance of the system. During the data mining agent learning process the knowledge base is continuously adapted to the environment of the data mining agent.

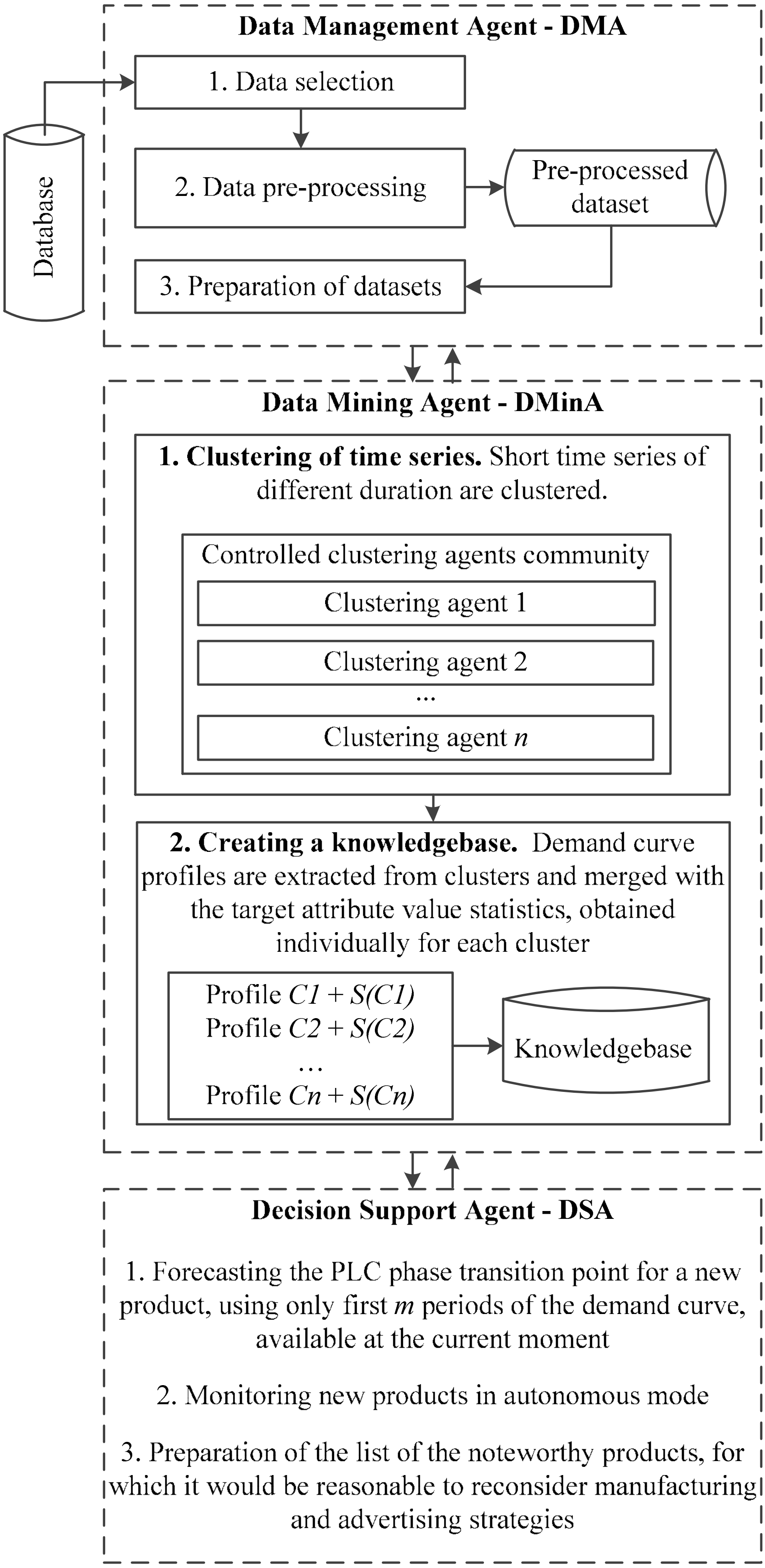


Figure 2.1. Structure of the proposed multiagent system

The main task of the decision support agent is the transition point forecasting for new products including cases when only several first points of the demand curve are available. The decision support agent is the utility-based agent. It uses the forecasting strategy to calculate the utility of the target attribute values and to select among them the suitable one.

The three mentioned agents are interconnected and form the proposed product lifecycle management support multiagent system. The analysis of an application of the proposed system to solving the mentioned task is presented in publications [27, 29, 30, 31, 33, 35].

2.1.1. Data management agent - DMA

The main task of the data management agent is the pre-processing of the data and providing access to the pre-processed data for other agents. The data management agent is the target-based agent without learning abilities. It uses the defined logic to choose the necessary pre-processing steps for a particular sample of an unprocessed data. The environment of a data management agent is the unprocessed data in the external database. As the new data appears, the environment forces the agent to perform all the necessary pre-processing. Thus an agent affects the environment by pre-processing the data. Figure 2.2 shows the internal processes of the data management agent. The processes are divided into two sectors:

- Sector *A* includes processes that can be performed autonomously. The result of performing those processes is the pre-processed dataset.
- Sector *R* joins the processes that are initiated with an external request. DMA runs these processes by responding to the requests from a data mining agent.

2.1.2. Data mining agent

The data mining agent (DMinA) has two main objectives – to create a model of the relations between the demand curve of a product and a transition point and make this model accessible for the decision support agent. The data mining agent is an intelligent agent. The environment it is situated in, is the data pre-processed by the data management agent that may contain learning data, testing data and new data. Changing itself, the environment forces the data mining agent to perform the learning process and build the knowledge base, and also the processes that should be made before the decision support agent forecasts the transition points for appropriate product. While performing the above mentioned actions the data mining agent changes the environment. Figure 2.3 shows the internal processes of the data mining agent.

The data mining agent can influence the environment directly or indirectly. The indirect case can be described as follows: the data mining agent sends a data request to the data management agent; as a result of this request, some changes are made to the environment, which starts one of direct influence processes. Several direct influence processes are possible including system learning process and system application process.

While the system is in learning state, the data mining agent is in learning state and the knowledge base mining process is in progress. The system learning can be initiated by the user or by the data mining agent, taking into account how old the present knowledge base is and how much changes to the environment (learning dataset) were made since the present knowledge base was built. From the side of data mining, the PLC phase transition point forecasting task is a classification task, as the target attribute has a finite set of values and descriptive attributes are also available – product demand in a defined time interval.

Formally, the knowledge base building process can be stated as follows. Let us accept that the dataset $D=\{d_1,d_2,...,d_i,...,d_n\}$ is a dataset with demand data, where each record $d=(x_1,x_2,...,x_j,...,x_l)$ is a discrete time series of length *l*. The record d_i represents the demand curve of the *i*-th product in the defined phase of a product lifecycle.

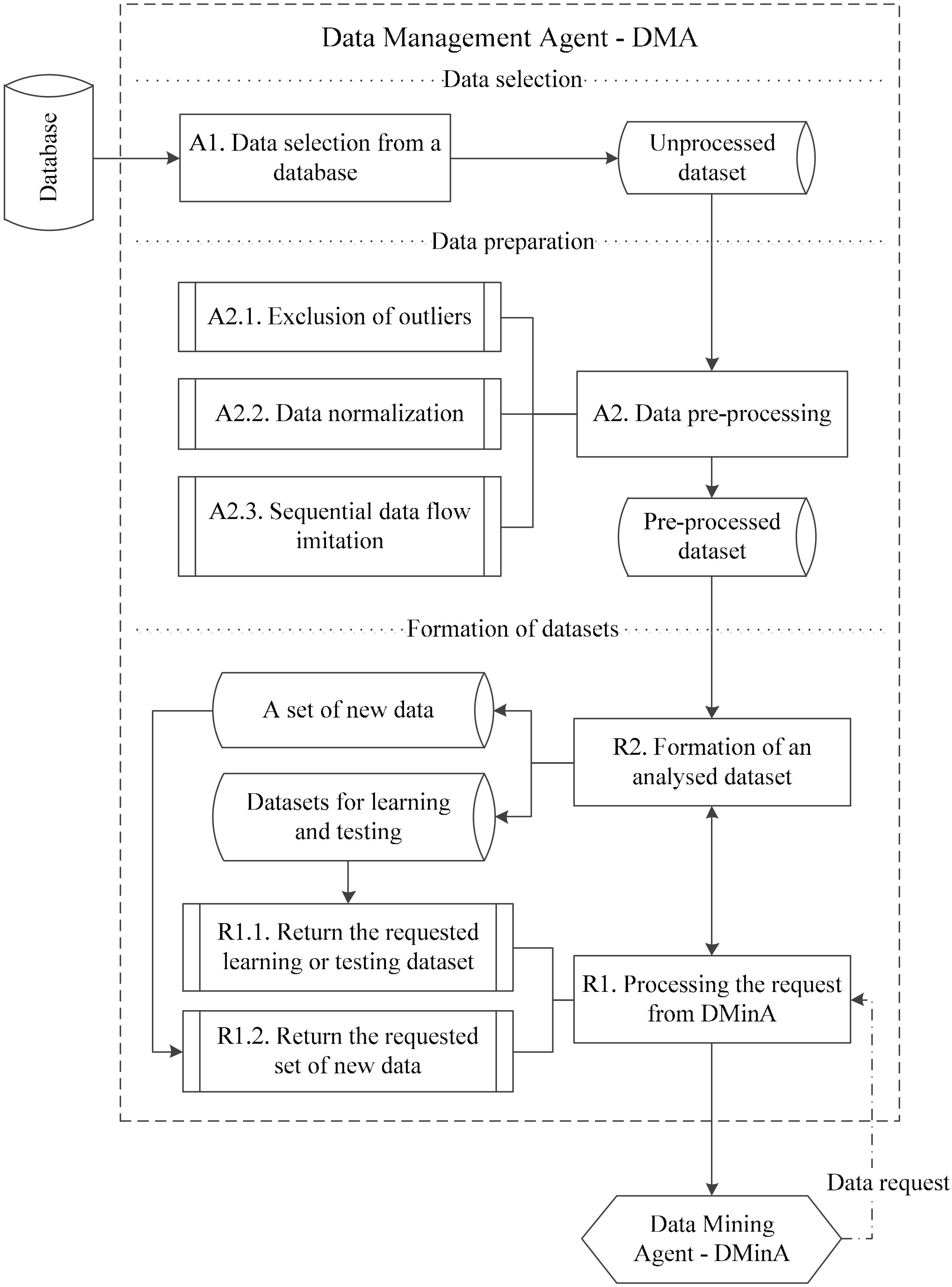


Figure 2.2. Internal processes of the data management agent

The length of a time series l is not a constant, it varies from record to record and can take values from the set L, $l \in L = \{l_1, l_2, ..., l_h, ..., l_s\}$. Each record $d_i \in D$ has a marker p, pointing at the last period of the product lifecycle phase, represented by the record d_i. The marker p has a finite set of values, $p \in P = \{p_1, p_2, ..., p_k, ..., p_m\}$. With this acceptation, the knowledge base can be represented as an implication model $f : D \to P$, which can be applied to forecast a value of a marker p for a new record $d' \notin D$. The learning process of a data mining agent (see Figure 2.3) contains the following five steps:

1.1. *Obtaining the parameters for learning process.* The values of parameters are obtained by sending an appropriate request. The first request is always sent to the user to obtain the initial values. The next requests may be sent to the user and also sent as internal requests; upon receiving them the data mining agent will use the results of the previous learning cycles to define the present parameters for learning.

1.2. *Obtaining the data for learning*. During this step the data mining agent creates the learning data request and sends it to the data management agent. Depending on the system testing strategy, defined by the learning parameters, the data management agent creates an appropriate learning set.

1.3. *Initialisation of a clustering agents community*. Each clustering agent implements the defined clustering algorithm (see Subsection 3.2); the parameters for each clustering agent are set during this step. The number of a clustering agent in the community depends on the load of clustering agent, defined by the parameter Q. Let us accept that the dataset D contains short time series of length l, $l \in L = \{l_1, l_2, ..., l_h, ..., l_s\}$, then the load of a clustering agent will define the number of individual values of l that can be processed by one clustering agent CA. The number of clustering agents n_{CA} is calculated using Equation (2.1):

$$n_{CA} = Roundup\left(\frac{|L|}{Q}\right), \tag{2.1}$$

1.4. *Data clustering*. The proposed multiagent system applies the modified algorithms of self-organising maps and hierarchical gravitational clustering (see Subsection 3.2) to cluster the data.

1.5. *Building the knowledge base*. The knowledge base has several levels, where the number of levels is equal to the number of clustering agents in the clustering agents community and each level stores the list of clusters obtained from the appropriate clustering agent.

The system application state merges two possible application cases – system testing and new data processing. The following steps are included in the system application process:

2.1. *Obtaining an appropriate data*. Depending on the system application case – test the system or forecast a transition point for a new product, the request for an appropriate data is sent to the data management agent. When it is necessary to forecast the transition point for a new product, a single record can be requested from the data management agent.

2.2. *Finding the closest cluster*. As the appropriate dataset is obtained, the closest cluster in the knowledge base is found for each record in the dataset. The level of a knowledge base, where the closest cluster should be found, is chosen depending on the length of a time series.

2.3. *Forwarding results to the decision support agent*. During this step the information from the closest cluster found is forwarded to the decision support agent, which makes the forecast.

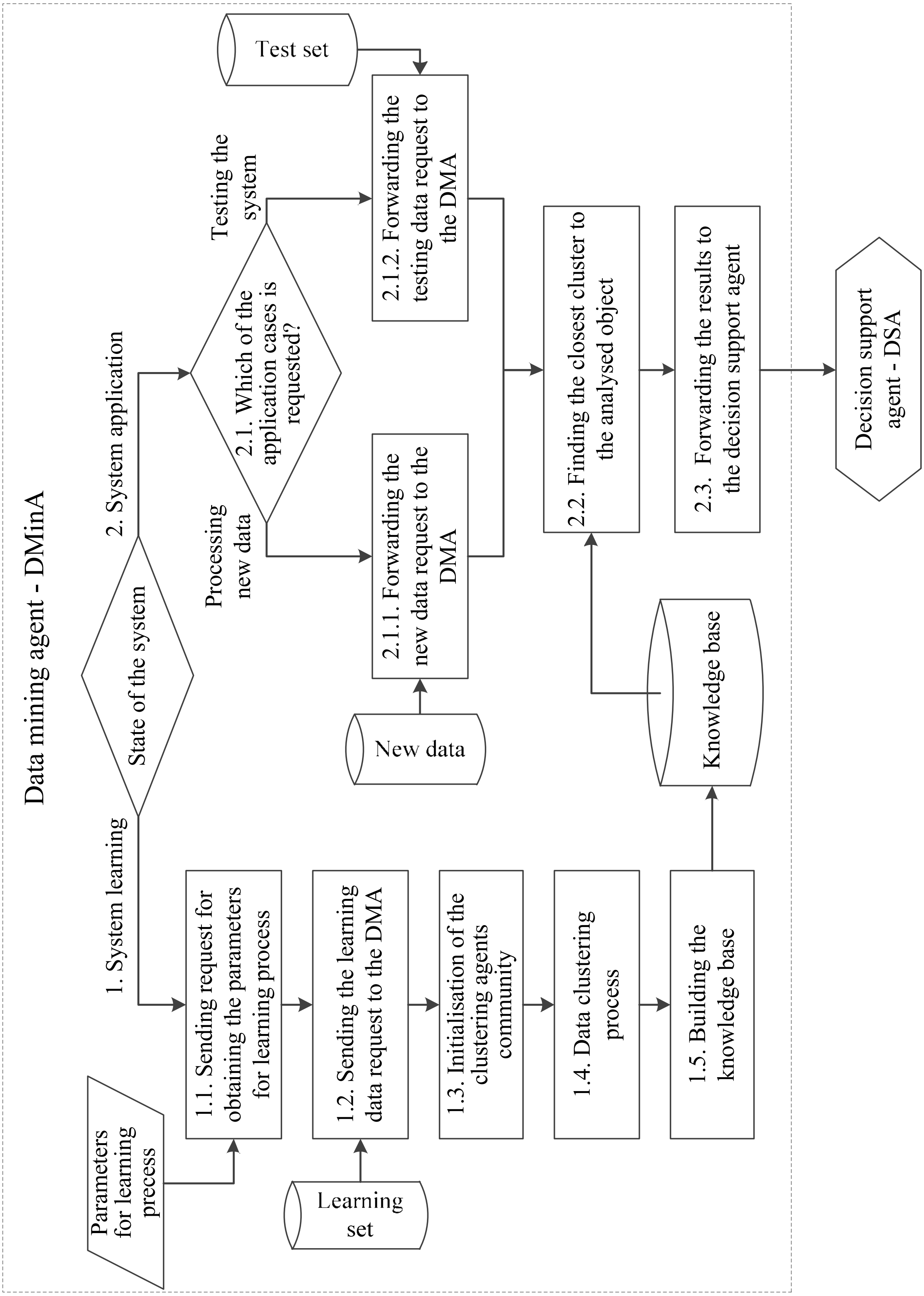


Figure 2.3. Internal processes of the data mining agent

The learning and testing cycle repeats until such values for learning parameters will be found that return an acceptable result. When the final values of the learning parameters are set, the knowledge base is rebuilt using both learning and testing datasets.

2.1.3. Decision support agent

The decision support agent is the one that forecasts the product lifecycle phase transition point for new products and monitors the selected and new products as well as prepares the list of the noteworthy products, for what it would be reasonable to reconsider manufacturing planning and advertising strategies.

The environment of the decision support agent is the new data and the dataset for testing. As the new record appears in the system, the decision support agent sends the request for information from the knowledge base and forecasts the transition point, thus changing the environment. Figure 2.4 shows the internal processes of the decision support agent. The decision support agent has two process flows separately for testing the system and for processing the new data. Both flows can be run autonomously or by request from the user. The process flow for testing the systems contains the following four steps:

1.1. *Analysis of testing data.* The data mining agent is being sent a request to find the closest clusters for records from the testing dataset. This request activates the following sequence of steps in the DMinA – step 2.1.2, step 2.2 and step 2.3.

1.2. *Forecasting procedure*. The proposed multiagent system uses the strategy described in Subsection 3.3 for choosing the value to forecast.

1.3. *Evaluation of the forecasting accuracy*. The strategy for evaluating the accuracy of the proposed system is described in Subsection 3.3.

1.4. *Reporting the results*. During this step the obtained results are forwarded to the user or to the data mining agent, which will use them to perform corrections in the learning parameters in order to improve the accuracy of the system.

The process flow for processing the new data contains the following four steps:

2.1. *Analysis of new data*. The processes of this step match the processes in the Step 1.1 of the system testing.

2.2. *Forecasting procedure*. The processes of this step match the processes in the Step 1.2 of system testing with the difference that the data that are processed are new data without the real value of the target attribute.

2.3. *Creating the list of the noteworthy products*. For selecting the noteworthy products the strategy, chosen by the user, is used. The proposed multiagent system uses the following strategy:

   a) If $l < p$ and $p - l > \theta$, then the product is not included in the list and remains monitored;
   b) If $l < p$ and $p - l \le \theta$, then the product is included in the list;
   c) If $l \ge p$, then the product is included in the list,

   where l is the length (number of periods) of the analysed time series; p is the forecasted value; θ is an importance threshold that is set by a user.

2.4. *Reporting the results*. The list of the noteworthy products is being forwarded to the user that takes the final decision for the status of each product on the list.

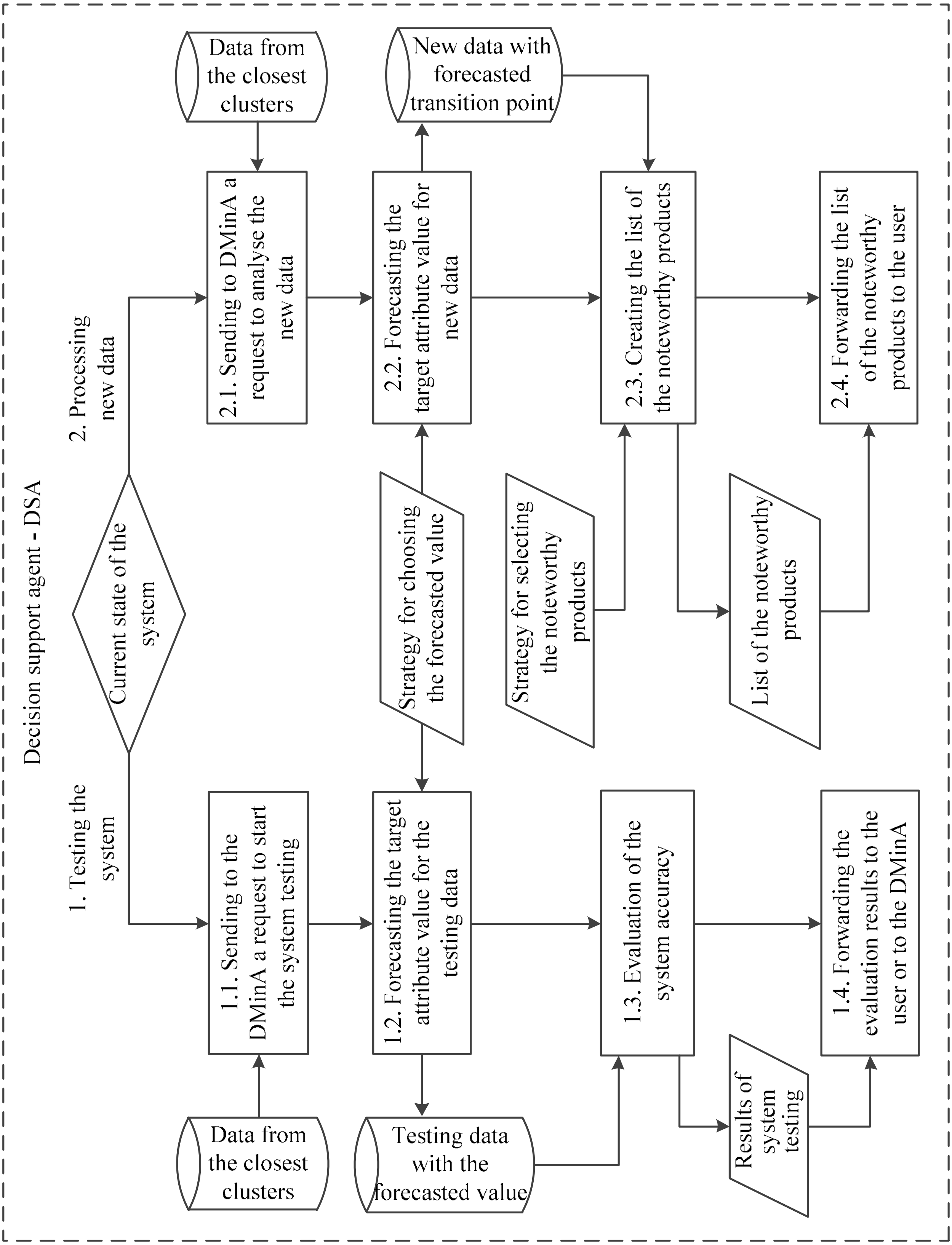


Figure 2.4. Internal processes of the decision support agent

Summary of Chapter 2

1. The analysis of shortage and advances of the multiagent systems gave the necessary information to motivate the application of the intelligent agent technology for solving the tasks of the Ph.D. Thesis.

2. Aiming at the system requirements defined in the thesis, a model of the multiagent system is developed, whose main objective is to forecast the product lifecycle phase transition point.
3. Applying the process distribution options of the multiagent systems, the tasks are defined for each agent. The functioning algorithms are developed and described for the data management, data mining and decision support agents.

From the summary of Chapter 2 it can be concluded that the first hypothesis is confirmed – the application of an intelligent agent technology contributes to the design of complex management and management support system by offering wide options for formalising the structure of the system and the processes and information flows. The application of intelligent agent technology contributed to the process of simplification of the complex product lifecycle management support system by offering options for distributing the processes among the interacting agents in the multiagent system.

Chapter 3

This chapter summarises the analysis and description of the data mining methods that are used by the proposed system. The importance of the data pre-processing is stated and the corresponding techniques are described in Subsection 3.1. Classical clustering algorithms and the developed modifications are given in Subsection 3.2. Subsection 3.3 provides a description of the procedure for forecasting value of a target attribute, as well as summarises the techniques for evaluating the accuracy of the system.

3.1. Data pre-processing

The amount of data, used in the real world while solving the data mining tasks, is large and grows with time and progress in the data technologies. In the most cases the data are gathered from different sources and have outliers, missing or contradictory values. The data may be improperly gathered due to the lack in technologies or to a human factor; there may be mismatches in titles or codes, or types of data; the chance remains for important attributes to remain unavailable due to the dysfunction in software or hardware. Unprepared and incorrect data lead to the incorrect results. That is the main objective for performing the data pre-processing before building the model of relations in data or system training or before any other knowledge discovery process. In total, the data pre-processing may take up to 80% from all data mining process. A number of data pre-processing techniques exist [2, 18, 37, 45]. These techniques are not exclusive and can be applied simultaneously:

- *Data cleaning* is used to lessen the noisiness in data, to take care of the missing values and to exclude the contradictory data.
- *Data transformation* includes such technologies as smoothing, aggregation, generalization, attribute construction and normalization that can improve the efficiency of the data mining methods.
- *Data integration* merges multiple data sources into one data repository.
- *Data reduction*.

3.2. Clustering data

Clustering or cluster analysis is the process of grouping objects into clusters, so that the similarity between objects in one cluster is more than the similarity between objects from the different clusters [2, 15, 18]. By result of data clustering it is possible to obtain the compact representation of data, such as the common models of a product lifecycle profiles

The cluster analysis can also be used to describe data for further analysis. The clustering creates an abstraction of an individual objects, multiple records are represented by several common prototypes (profiles), what being used while analysing the new data [2, 45]. As an example of it the process of forecasting of the product demand, described and analysed in the source [22] can be taken.

Several clustering algorithms were analysed in order to define those that are suitable for application to solving the tasks of the Thesis. The following algorithms were analysed [2]: *k*-means and *k*-medoids partition clustering algorithms; agglomerative, divisive and gravitational hierarchical clustering algorithms; *DBSCAN* clustering algorithm and self-organising maps. As the result of the analysis the self-organising maps and the hierarchical gravitational clustering algorithm was chosen. The research results on the suitability of those methods for clustering time series are summarised in publications [27, 28, 32, 34, 35, 60].

3.2.1. Distance measures

To measure the similarity between time series the proposed clustering algorithms use the distance between those time series that is calculated by *DTW – Dynamic Time Warping* distance measure or by the proposed modification of the *Euclidean* distance named *MEuclidean*.

The dynamic time warping is a technique that searches for a warping position between two time series. The *DTW* technique can be described with the next three main steps – calculate the distance matrix; find a warping path; calculate the distance using the warping path. The warping path $W(X,Y)$ is a sequence of elements from the distance matrix. The warping path is used to produce the comparison map of two time series *X* and *Y*.

The modifivation of the *Euclidean* distance measure is proposed in the Thesis, applying which becomes possible to calculate distance between time series of different length. The main idea of the *MEuclidean* measure is that the distance between two time series is calculated using only dymencions that those time series share (see Figure 3.1).

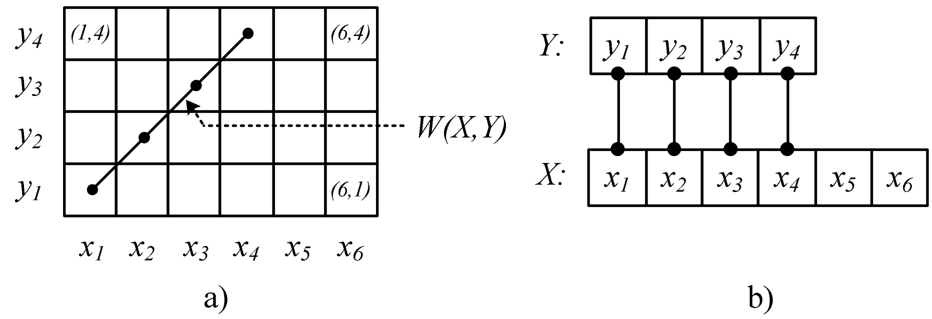


Figure 3.1. Example of a warping path and the comparison map for the *MEuclidean*

3.2.2. Self-organizing maps

The self-organising neural networks are artificial neural networks named *Self-Organising Maps* (*SOM*). These networks implement the unsupervised learning and competitive learning. Neurons of the network compete with one another for the right to be activated. The neuron that wins this right is called a winning neuron. The idea of the competitive learning is the application of the lateral connections between neurons [14, 61], proposed by *F.Rozenblatt* [39]. The three topologies of the self-organising maps were compared in the thesis while evaluating the proposed multiagent system – rectangular topology, cross-type topology and a linear topology. The classical algorithm of the self-organising maps has the next four main steps:

Step 1. *Initialization of a network*. The weight vector of each neuron is randomly assigned initial values. These values should be small in order to exclude any possible organisation between neurons at the beginning of the learning process [61].

Step 2. *Competition*. During this step for each input signal a winning neuron is found, which returns the best value of the fitness function, which in this thesis is calculated as a minimized distance between a time series *X* and a weight vector *W* of a neuron.

Step 3. *Cooperation*. During this step the winning neuron activates the nearest neighbours, what become the co-activated neurons. The topological neighbourhood of a winning neuron is defined as $h_{j,i(x)}$, where $i(x)$ is the index of a winning neuron for input signal *x* and *j* is the index of a co-activated neuron. The topological neighbourhood decreases with time, which is a unique option of self-organising maps.

Step 4. *Synoptic adaptation.* This process has two stages – the organisation of a neural network and the convergence of a neural network [23, 61]. During the organisation stage the topological ordering of the weight vectors of neurons is performed. During the convergence process the weight vectors are fitted to the input signals – to the training set. Equation (3.1) is used to perform the synoptic adaptation.

$$W_j(n+1) = W_j(n) + \eta(n) \cdot h_{j,i(x)}(n) \cdot \left(X - W_j(n)\right), \tag{3.1}$$

where $W_j(n)$ – weights vector of an *j*-th neuron in *n*-th iteration;

X – input signal, time series from the training dataset;

$\eta(n)$ – learning coefficient in *n*-th iteration;

$h_{j,i(x)}(n)$ – coefficient for topological neighbourhood in *n*-th iteration.

3.2.3. Modifications to the self-organizing maps algorithm

The classical self-organising maps do not have an option for processing input vectors of different length – all input vectors must have equal length. An *Euclidean* distance measure during the competition step of an algorithm was replaced with one of the distance measures – the proposed *MEuclidean* or *DTW*. While using each of mentioned measures, the comparison map *M* is built, which contains elements (x_i, w_j). Each element of a comparison map defines the value x_i of a time series *X*, which should be used to adopt the synoptic weight w_j. While using the *MEuclidean* distance measure the following cases are possible:

1. The length l_W of a weight vector *W* is equal to or smaller than the length l_X of a time series *X*. In this case all synoptic weights of a neuron are adapted to the input signal according to the comparison map *M*.
2. The length l_W of a weight vector *W* is larger than the length l_X of a time series *X*. In this case only first l_X synoptic weights of a neuron are adapted to the input signal according to the comparison map *M*. Other synoptic weights with index value larger than l_X, remain unchanged.

With respect to the defined cases the corresponding modifications are made to the self-organising maps algorithm. The synoptic adaptation process was changed to support the distance measures, described in Subsection 3.2.1. Using the *MEuclidean* distance measure the new Equation (3.2) is used for synoptic weights adaptation instead of Equation (3.1):

$$\forall w_k \in W_j, k \le l_{\min}: \quad w_k(n+1) = w_k(n) + \eta(n) \cdot h_{j,i(X)}(n) \cdot (x_k - w_k(n)), \tag{3.2}$$

where W_j – weights vector of an j-th neuron;

$l_{\min} = \min(l_X, l_W)$;

$\eta(n)$ – learning coefficient in n-th iteration;

$h_{j,i(X)}(n)$ – coefficient for topological neighbourhood in n-th iteration.

To support the process of building the knowledge base, a new step is added to the self-organising maps algorithm. During this step a list of clusters is created. The following strategy is proposed for extracting the clusters:

1. Each of the records in the training set is forwarded to the network input sensors;
2. The winning neuron for each record X is found. For each winning neuron a transition point statistics is S_j collected, containing a transition point value and frequency of it's appearing in this neuron, which is used as a rank of a transition point.
3. The cluster is a neuron that at least once became a winning neuron during the cluster extraction process. The weight vector of a neuron is taken as a centroid of a cluster.

The cluster $c_j \in C$ is an object that contains the weights vector W_j of the j-th neuron and the statistics S_j for the target attribute values $p_i \in P$ as the set of elements $(p_i, f)_j$ (see Figure 3.2).

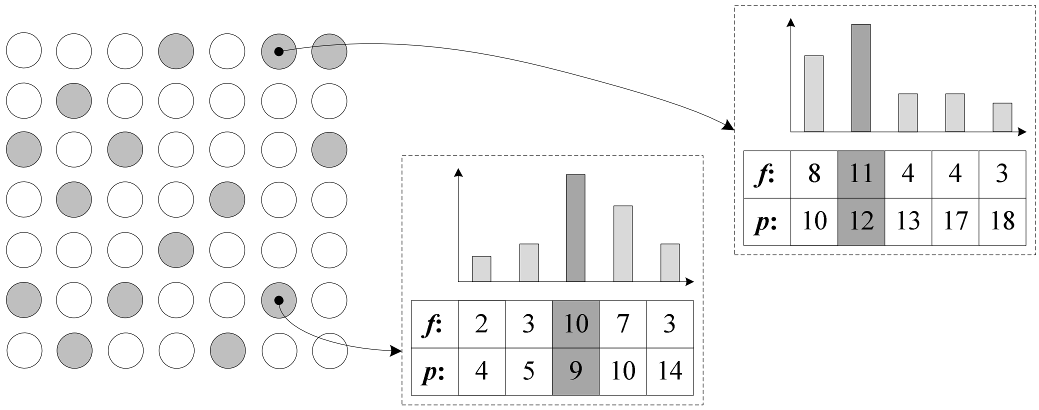


Figure 3.2. Example of the statistics S_j for self-organising maps

The element f of each tuple $(p_i, f)_j$ holds the frequency of appearing in the j-th neuron for the value p_i of a target attribute. The process of cluster extraction and building the list of clusters can be easily changed according to the tasks defined.

3.2.4. Gravitational clustering algorithm – *G-Algorithm*

The *G-Algorithm* [16] is an unsupervised robust clustering algorithm based on the *Gravitational law* and the *second Newton's motion law*. Each data record – a time series in the training set, is considered as an object in n-dimensional feature space and is moved by using the gravitational force and the second motion law. The gravitational clustering algorithm was developed by *W.E.Wright* [51]. This algorithm is a hierarchical agglomerative algorithm: the gravitational force is used as a mechanism for merging objects until only one object (cluster) remains in the feature space. The gravitational clustering algorithm has the following steps:

Step 1. First the clustering parameters are set – gravitational constant; whether the gravitational force decreases with time or not; minimal distance for merging objects and other parameters;

Step 2. The feature space is created, which is an environment where the objects are moved and merged into clusters;

Step 3. Each object is calculated a gravitational force exerted over it by other objects;

Step 4. According to the calculated gravitational force, each object is moved in the space. Equation (3.3) is used to calculate the changes to the object x position, forced by the gravitational force exerted by an object y:

$$x(t+1) = x(t) + \overrightarrow{d(t)} \cdot \frac{G \cdot m_y}{2 \cdot \left\| \overrightarrow{d(t)} \right\|^3}, \tag{3.3}$$

where $\overrightarrow{d(t)} = x(t) - y(t)$; G – gravitational constant; m_y – mass of an object y.

Step 5. It is checked whether some objects are close enough to be merged. If such objects exist, then these are merged. The algorithm returns to the step 2 and continues until the stopping condition becomes true.

Step 6. As the algorithm finishes the list of clusters is created. If a cumulative object has plenty enough objects merged, then it becomes a cluster and is included in the list. The centre (representative) of a cluster is calculated as a centroid of all objects merged in the cluster.

3.2.5. Modifications to the gravitational clustering algorithm

The classical gravitational clustering algorithm uses the *Euclidean* distance measure to evaluate the closeness of the objects. This brings limitations to the algorithm – all time series must have equal length.

In order to support an option to cluster short time series of different length the second, third and sixth steps of the gravitational clustering algorithm were modified. The following modifications were made:

Step 2. One of the feature space limitations that is defined like „*In the n-dimensional feature space only objects precisely with n dimensions can exist*" was changed to „*In the n-dimensional feature space can exist objects with m dimensions, were* $m \le n$". This means that in the 10-dimensional feature space a time series with four periods can exist, but it will be present only in the first four dimensions and will not be present in the dimensions with index larger than 4. This assumption allows to use new distance measures and to modify the gravitational force calculation strategy in the Step 3.

Step 3. An *Euclidean* distance was replaced by one of the distance measures, described in Subsection 3.2.1. Using the *MEuclidean* distance measure the distance between two objects X and Y is calculated using only shared dimensions of those objects. As a result, the following cases might occur:

1. The number of dimensions of an object X, over which the gravitational force is exerted, is less or equal to the number of dimensions of an object Y, which exerts the gravitational force over the object X. In this case the gravitational force is calculated for all dimensions of the object X. The

object is moved according to the gravitational force calculated for each dimension individually.

2. The number of dimensions of an object *X*, over which the gravitational force is exerted, is greater than the number of dimensions of an object *Y*, which exerts the gravitational force over the object *X*. In this case the gravitational force is calculated only for objects *X* and *Y* shared dimensions. The object *X* is moved only in those dimensions for which a gravitational force was calculated.

While using the *DTW* distance measure, the comparison map for compared objects is built. The map defines which dimension of an object *Y* exerts gravitational force over which dimension of an object *X*. Irrespective of the number of dimensions, always all dimensions of both objects are used to calculate the distance.

Step 6. As the centre of a cluster, the position of a cumulative object is taken, at the moment of cluster extraction. The statistics S_j is gathered for each target attribute value $p_i \in P$ that appears in the *j*-th cluster. The statistics contains tuples $(p_i, f)_j$, where f holds the frequency of appearing in the *j*-th neuron for the value p_i of a target attribute. The gathered statistics is used while forecasting the value of a target attribute for new objects.

3.3. Forecasting procedure and performance measures

The procedure for forecasting a value of a target attribute can be launched by a user request, while forecasting for a new product; or by an internal request from a decision support agent, while testing the system after the training process is finished or while the system is in the autonomous mode. In both cases the following algorithm, containing three main steps, is used to forecast a value of a target attribute:

Step 1. Define the level of the knowledge base where the closest cluster will be searched. Each level in the knowledge base contains a list of clusters from an appropriate clustering agent and has the same bound for the length of a time series;

Step 2. On the defined level of a knowledge base find the closest cluster c_j for an object that is currently analysed. The closeness between a cluster and an object is calculated using one of the distance measures, described in Subsection 3.2.1 – the smaller the distance is, the closer a cluster is to an object;

Step 3. The target attribute value statistics S_j in the cluster c_j is used to choose the forecasted value $p_i \in P$. In the proposed algorithm as a forecasted value the one with a higher frequency f is chosen.

At Step 3, a case when several values p_i have an equal frequency f is possible. In this case the smallest value p_i is taken. It is also possible to merge statistics from multiple clusters that are close to the object.

The accuracy of a forecast is the main index of system efficiency. The accuracy of the system is measured as a *Mean Absolute Error* of a forecast. As an additional index for system accuracy a *Logical Error* and a *Clustering error* can be used. The logical error holds the number of times the system has made logically incorrect decisions. To calculate the logical error first a logically correct and incorrect decision must be defined. With respect to the tasks defined in the thesis, logically correct and incorrect decision can be defined as follows:

1. Let us assume that the length of a time series is equal to g_l and the value of a target attribute is $p = p_l$, where $p_l > g_l$. This assumption points out that the product did not change the PLC phase yet. Due to that, logically correct would be to forecast a value $r_l > g_l$ and logically incorrect to forecast $r_l \leq g_l$;
2. Let us assume that the length of a time series is equal to g_l and the value of a target attribute is $p = p_l$, where $p_l \leq g_l$. This assumption points out that a product is in the next PLC phase. Due to that logically correct would be to forecast a value $r_l \leq g_l$ and logically incorrect to forecast $r_l > g_l$.

The statement that when $r_l = g_l$ the product is already in the next phase can be considered as correct, because if the demand data for some period l are available then this period is logically finished and all transitions, if any, have occurred. The logical error should not be used as a main index of the system accuracy, as it does not bring enough information of how precise the system is.

The clustering error shows how precise the training data were split into clusters. First the clustering error is calculated for each cluster and then summated, using the following algorithm:

Step 1. For each object $o_{i,j}$ in the cluster c_j a distance $d\left(o_{i,j}, c_j\right)$ is calculated.

Step 2. For each cluster c_j all calculated distances $d\left(o_{i,j}, c_j\right)$ are summated, thus a summated clustering error for each cluster is calculated;

Step 3. The total clustering error is calculated summating the errors of all clusters c_j.

Step 2 of the proposed algorithm can be changed in such way that distances $d\left(o_{i,j}, c_j\right)$ for each cluster c_j are not summated, but an average distance for each cluster is calculated. In this case a summated average clustering error is calculated.

Summary of Chapter 3

1. An analysis of the existing clustering methods was performed. As a result, suitable algorithms for clustering demand data were chosen for further application.
2. Performing the detailed analysis of the chosen clustering algorithms it is concluded that the classical algorithms are not capable of clustering short time series of different length.
3. The distance measures are proposed, using which it is possible to calculate distance between short time series of different length.
4. A modification to the self-organising maps is proposed that is capable of clustering short time series of different length.
5. A modification to the gravitational clustering algorithm is proposed that is capable of clustering short time series of different length.
6. The procedure for forecasting the target attribute value for new products is proposed.

Chapter 4

To evaluate an accuracy of the system, the specialised software tool was developed. It implements the structure of the proposed multiagent system and also the developed modifications of the clustering algorithms. The software tool gives an option to easily change all necessary parameters, to train and test the system and to save the results in a convenient format. The *Microsoft Visual Studio 2008* was used to develop the software tool, the last changes to the tool was made in the *Microsoft Visual Studio 2010*. The *Visual Basic* programming language was used. The chapter presents functional aspects of the software tool and of the user interface.

Chapter 5

5.1. The datasets

The data that was used in the experiments were obtained in the scope of the international project *ECLIPS* [56]. The dataset contained 1000 records representing the demand during the introduction phase for the products of an international chemical manufacturing company *HUNTSMAN* [58]. The length of time series was 23 periods and under, data were not normalised, contained missing values and outliers and were not aligned by time. Figure 5.1 shows an example of an unprocessed data.

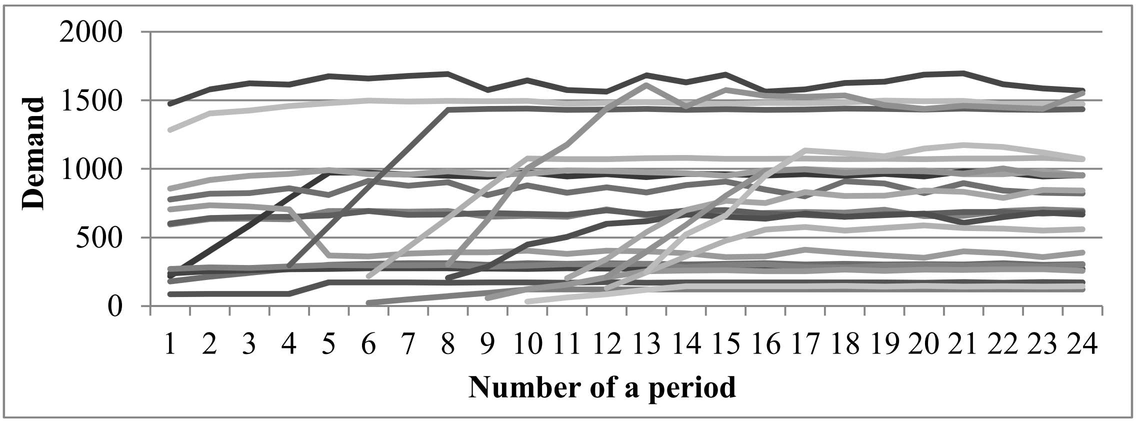


Figure 5.1. Example of the primary data

The majority of time series are similar in first periods and this can decrease the accuracy of the system. A constraint that the minimal length of a time series must be four periods was defined in order to decrease entropy in data. The time series were aligned by the first period. The records with missing values were excluded from a dataset, other records were normalised using the *Z-score with standard deviation normalisation method*.

After the data pre-processing step the dataset contained 312 normalised short time series with length from 4 till 23 periods, representing the product demand in the introduction phase. The number of the last period was used as a transition point to define the moment when product changed an introduction phase to maturity. The artificial dataset with manually decreased level of noise was used in parallel with a primary dataset. The artificial dataset was created using 60 profiles from the primary dataset. Equation (5.1) was used to generate 400 artificial time series with length from 4 till 23 periods:

$$x_i' = x_i\left(1-\eta+2\eta\cdot RND()\right), \tag{5.1}$$

where η - boundary coefficient, the value of which was set to 0.2 (20%) while generating the artificial dataset.

5.2. Practical experiments

The subsection presents the process of preparing a list of experiments to evaluate the accuracy of the proposed multiagent system. The accuracy evaluation process is described for the system with each of the proposed clustering methods (see Subsection 3.2), the obtained results are summarised and analysed. The parameters of the multiagent system have a finite set of values, which makes it possible to calculate the number of experiments for evaluating the accuracy of the system. Both clustering methods are tested with two dataset using the 10-fold cross-validation and two distance measures – *DTW* and proposed *MEuclidean*. During the testing process the load of the clustering agents took values in the interval from 1 to 20, $Q \in \{1, 2, ..., 20\}$. This gives 800 experiments in total for each clustering algorithm.

Three neural network topologies were tested for system with the *SOM* algorithm; this increases the number of experiments to 2400. For the system that uses the gravitational clustering algorithm only one parameter was changed, which defines whether the mass of a cumulative object will be calculated as the summated mass of all merged objects or will always remain equal to 1. This increases the number of experiments for the *GC* algorithm up to 1600. The 4000 experiments should be performed to evaluate the accuracy of the system in all combinations of the main parameters. The total number of experiments can be reduced by evaluating the learning error of the system.

5.2.1. Evaluation of the learning error and definition of the list of experiments

Using information from sources [16, 23, 61] and an empirically obtained experience, the values of several parameters for both clustering algorithms were assigned a constant values. These values were used while evaluating the learning error and were not changed during the system testing process when the testing error was calculated.

A list of 160 experiments was prepared for a multiagent system with gravitational clustering algorithm. Figures 5.2 and 5.3 show the learning error of a system with *GC*, evaluating on primary and artificial data, using the case when the mass of a cumulative object will always remain equal to 1. Similar results were obtained for the case when masses of the merged objects were summated – the smallest learning error was reached using the *MEuclidean* distance measure. This allows one to exclude the *DTW* measure from further experiments with the system that uses gravitational clustering algorithm.

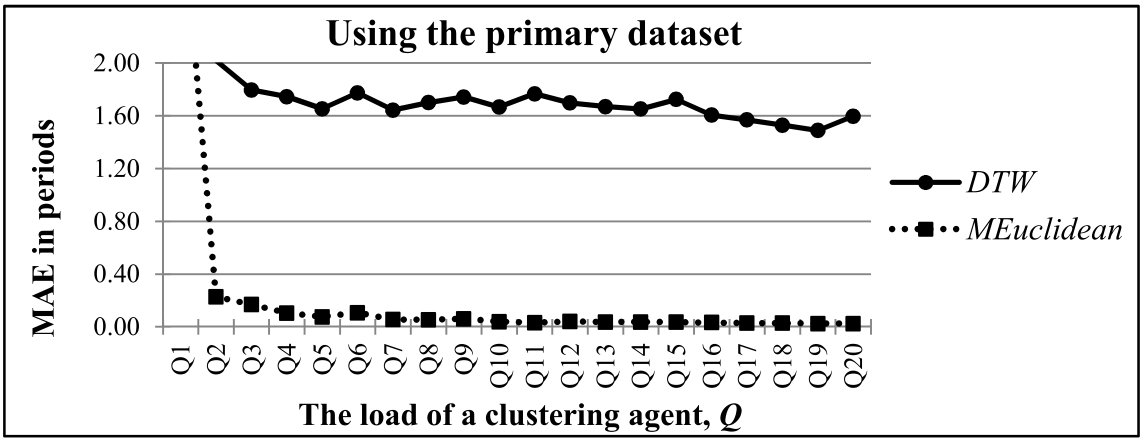


Figure 5.2. The learning error of a system with *GC* (without summating masses)

The case when masses are summated has a disadvantage. As objects are merging, the mass of a cumulative object increases, reaching the bound when the mass of an object has a larger impact on merging the objects as compared to the closeness of those objects. This allows one to evaluate the system only with the case when the mass of a cumulative object always remains equal to 1. The set of possible values of the clustering agents load was reduced to five values: for primary dataset - $Q \in \{16, 17, 18, 19, 20\}$; for artificial dataset - $Q \in \{15, 17, 18, 19, 20\}$.

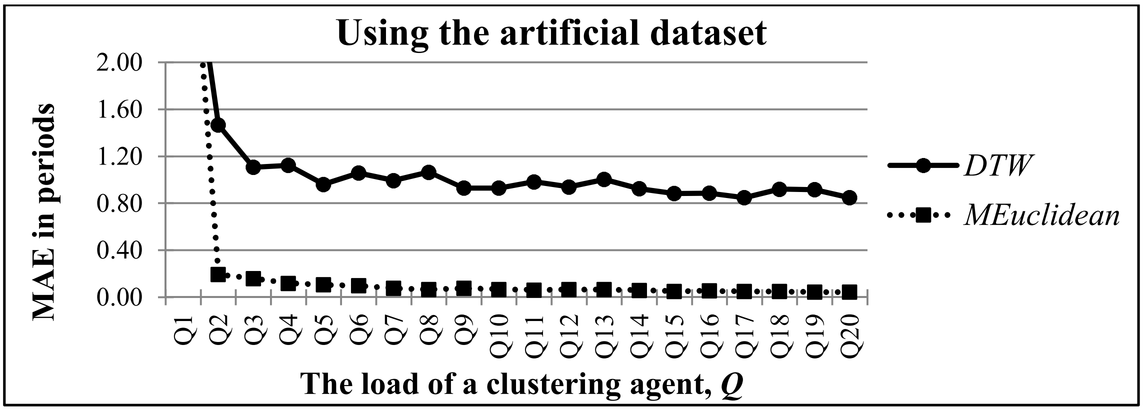


Figure 5.3. The learning error of a system with *GC* (without summating the masses)

The system that uses the *SOM* algorithm, was evaluated using both distance measures and three topologies – rectangular, cross-type and linear. Figure 5.4 depicts the average learning error for three topologies of the system with *SOM*, using the *MEuclidean* distance measure. It can be seen that the learning error is sufficiently higher than for the gravitational clustering algorithm, thus it can be concluded that the testing error will also be higher as compared to the *GC*. This allows one to reduce the set of the clustering agents load values to three values: for primary dataset - $Q \in \{5,6,7\}$; for artificial dataset - $Q \in \{7,8,9\}$.

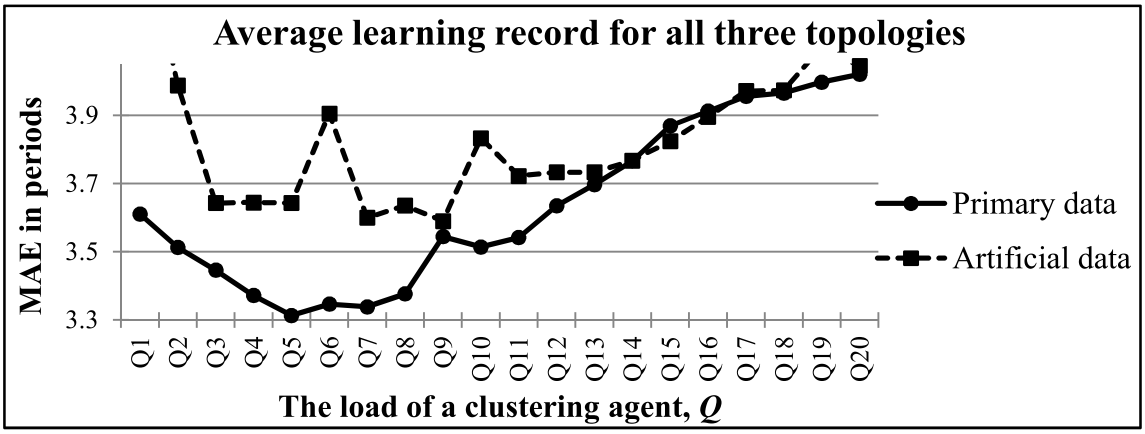
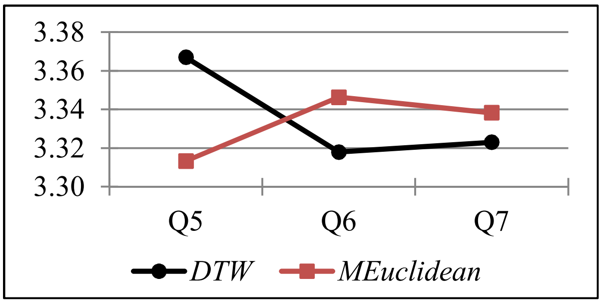
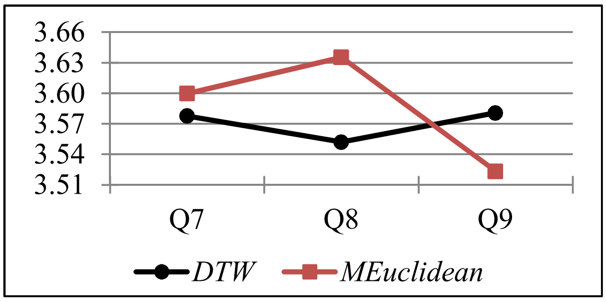


Figure 5.4. The average learning error of a system with *SOM*, using the *MEuclidean*

Figure 5.5 shows the learning error for system with *SOM*, comparing both distance measures. The results show that none of applied distance measures dominates other, thus the system with the *SOM* algorithm is tested using both distance measures.



a) Using primary dataset



b) Using artificial dataset

Figure 5.5. The average learning error of a system with *SOM*

5.2.2. Evaluating the accuracy of the multiagent system with the *SOM* algorithm

The system was tested using both datasets and applying the 10-fold cross-validation. Table 5.1 summarises the testing results, evaluating with the primary dataset. The best results for each topology are marked with bold for each distance measure separately.

Table 5.1

Testing error of a system with *SOM* evaluating on the primary dataset (in periods)

The load of a clust. agent	Linear topology		Cross-type topology		Rectangular top.	
	DTW	*MEuclidean*	*DTW*	*MEuclidean*	*DTW*	*MEuclidean*
Q = 5	3.475	3.447	3.362	3.437	3.434	3.578
Q = 6	**3.267**	**3.267**	**3.332**	**3.404**	**3.353**	**3.353**
Q = 7	3.344	3.411	3.372	3.513	3.371	3.509

Summarising the results obtained, it can be concluded that the smallest testing error for the system with *SOM* algorithm evaluating with the primary dataset is equal to 3.267 periods and was obtained using the linear topology, clustering agents load *Q* equal to 6 and independently of the distance measure. Table 5.2 summarises the testing results evaluating with the artificial dataset. The best results for each topology are marked with bold for each distance measure separately.

Table 5.2

Testing error of a system with *SOM* evaluating on the artificial dataset (in periods)

The load of a clust. agent	Linear topology		Cross-type topology		Rectangular top.	
	DTW	*MEuclidean*	*DTW*	*MEuclidean*	*DTW*	*MEuclidean*
Q = 7	3.521	**3.889**	3.553	**3.889**	3.657	**3.910**
Q = 8	**3.512**	3.925	**3.494**	3.925	**3.565**	4.000
Q = 9	3.575	4.017	3.553	4.090	3.653	4.135

The smallest testing error for the system with the *SOM* algorithm evaluating with the artificial dataset is equal to 3.494 periods and was obtained using the cross-type topology , the *DTW* distance measure and the clustering agent load *Q* equal to 8.

5.2.3. Evaluating the accuracy of the multiagent system with the *GC* algorithm

This subsection summarises the testing results of the system that uses gravitational clustering algorithm. The 10-fold cross-validation was used to test the system. Figure 5.6 shows the testing results of evaluation with the primary data. The testing error remains on the level of two periods. The smallest error for the system with *GC* was obtained using the clustering agents load *Q* equal to 17 and was equal to 1.995 periods.

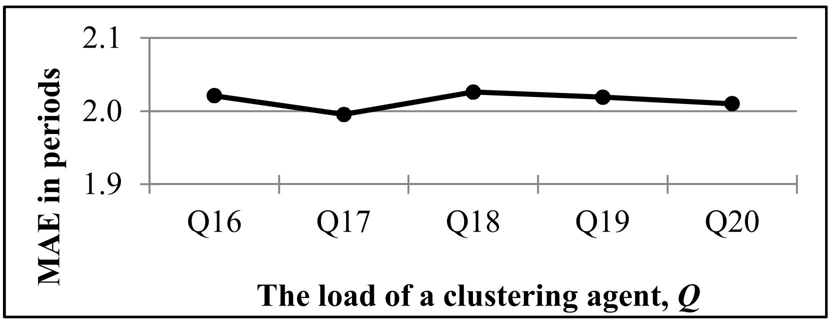


Figure 5.6. Testing error of the system with the *GC* algorithm (primary dataset)

Figure 5.7 illustrates testing results for the system that uses gravitational clustering evaluation with the artificial dataset. The testing error remains on the level of 0.2 periods. The smallest error for the system with *GC* was obtained using the clustering agents load *Q* equal to 20 and was equal to 0.2 periods.

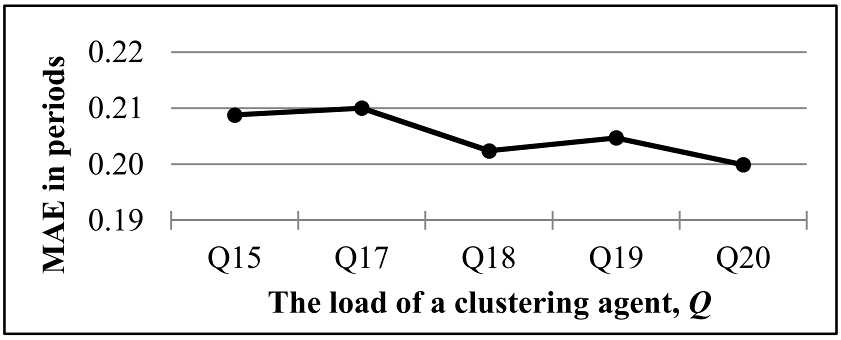


Figure 5.7. Testing error of the system with *GC* algorithm evaluation on artificial data

5.3. Analysis of obtained results

The best accuracy of the system that uses *SOM* evaluation with primary dataset was obtained using the linear topology and the clustering agent load *Q* equal to 6 and was equal to 3.267 periods. The distance measure did not affect the result – both distance measures returned testing error equal to 3.267 periods. The best result on average for three topologies was obtained using the *DTW* distance measure and the clustering agent load *Q* equal to 6 and was equal to 3.317 periods. At the same time the best result on average for three topologies that was obtained using the *MEuclidean* distance measure and the same clustering agent load was equal to 3.341 periods. Both the smallest and the average result can be considered as comparatively large error, as the one period time is considered to be one month. Evaluating the system that uses the gravitational clustering algorithm with the primary dataset the smallest testing error was equal to 1.995 periods. This result was obtained using the *MEuclidean* distance measure and a clustering agents load *Q* equal to 17. The testing error is smaller than the one returned by the system that uses the *SOM*, but still remains close to 2 periods. As a logical explanation for this the moment that data were obtained from a successive company. At the same moment values for the target attribute were defined by the expert group of a company, using individual experience and the market status. This increased the entropy in data. Taking into account the described situation the obtained testing error equal to 1.995 periods can be considered as a comparatively small one and it can be concluded that a system that uses the hierarchical gravitational clustering algorithm has a comparatively high accuracy.

Evaluating the system with artificial data that uses self-organising maps, the smallest testing error was equal to 3.494 periods and was obtained using the cross-type topology, *DTW* distance measure and the clustering agents load *Q* equal to 8. At the same time, while testing the system that uses *GC* with the same data, the minimal error was equal to 0.2 periods and was obtained using *MEuclidean* distance measure and clustering agents load *Q* equal to 20.

Analysing the testing results with both datasets of the system that uses the gravitational clustering algorithm the following relation was noticed. The best results were obtained using clustering agents load that was close to the maximal possible value of *Q*: *Q = 17* for primary dataset and *Q = 20* for artificial dataset. It was also noticed that the error decreases while increasing the clustering agents load. This can be explained by that when using the defined values for *Q*, the clustering agents society contained at most two clustering agents. As a result a larger amount of data becomes available for a clustering agent, which together with imitation of the flow of sequential data contributes to obtaining a more precise statistics for each cluster during the cluster extraction process. Practically, in the described case each cluster contains only one or two values of a target attribute, what contributes to accuracy of the system.

At the same moment analysing the results, obtained for the system that uses the self-organising maps, the tendency of testing error to decrease while enlarging the clustering agents load was not noticed. The smallest testing error testing with primary data was obtained using the clustering agents load equal to 6 and testing with artificial dataset – equal to 8. The differences in results for self-organising maps and for hierarchical gravitational clustering algorithm can be explained with conceptual distinctions of these two algorithms. During the clustering process the self-organising maps are focused on minimizing the clustering error, this concept is defined in the network adaptation process of *SOM.* As a result during the cluster extraction process a list of clusters with significantly small clustering error is created, but clusters not obligate contain precise statistics for the target attribute values, what was experimentally proven in Subsection 5.2.2. In comparison, the hierarchical clustering algorithm uses another strategy for building clusters. The algorithm calculates the distance between objects and sequentially merges closest objects into clusters. During this process clusters with both large and small clustering error are created. The sequential object merging joined with proposed *MEuclidean* distance measure contribute to obtaining the more precise statistics of a target attribute values for each cluster extracted, what was experimentally proven in Subsection 5.2.3.

Comparing the clustering algorithms by the learning error the gravitational clustering algorithm dominates the self-organising maps. The learning error of the system that uses the gravitational clustering algorithm is close to 0 and decreases while increasing the clustering agents load. Comparing the distance measures, the *MEuclidean* dominates the *DTW* for the system with gravitational clustering algorithm, what can be explained with that the flow of sequential data is being imitated, what contributes to obtaining more precise target attribute values statistics while using the *MEuclidean* distance measure.

Analysing the system testing results with artificial dataset, it becomes possible to conclude about the robustness of the clustering algorithms that were evaluated. As can be seen from the testing results of the system that uses the self-organising maps (see Subsection 5.2.2), the obtained testing error is comparatively large, though the error for the primary and for the artificial dataset remain on the same level. This allows to conclude that the self-organising maps are highly robust, since the artificial dataset has the noisiness level manually decreased comparatively to the primary dataset. At the same time the reaction to noise of the system that uses the gravitational clustering algorithm was more acute - comparatively to the testing results with artificial data, the evaluating the system with primary data returned 10 times worse results (see Subsection 5.2.3).

The performed analysis of the obtained results allow to conclude that the hierarchical clustering methods are less robust comparatively to the self-organising maps, nevertheless the hierarchical clustering algorithms more precisely split data into clusters thereby increasing the accuracy of forecasting the target attribute value. In conclusion of the chapter it should be stated that the obtained results and given conclusions about the accuracy of the system and clustering algorithms are pertinent to the tasks defined and data used in the Thesis.

CONCLUSIONS

The ideas and concepts of the intelligent agent technology that contribute to the design and modelling of the complex and distributed systems are analysed in the Ph.D. Thesis. The proposed multiagent system was developed for a specific task, more precisely – for product life cycle phase transition point forecasting. Using the intelligent agent technology the structure of the system was intentionally made simple, allowing easy adaptation of the system for other tasks, similar to those defined in the Thesis. The developed structure of the agents in the proposed system and the strategies chosen for process distribution allow agents to be easily changed without rebuilding the whole system. The results gathered during the development of the system prove the first of the hypotheses pointing that application of the IAT contributes to the complex management systems design process.

The combination of the data mining methods allowed developing of the distributed process for knowledge discovery from data and forecasting the target attribute value for new data, applying to the tasks defined in the Thesis. The flexibility of the data mining methods contributed to the development of modifications of the clustering algorithms with new features. The obtained testing results of the system confirm the second of hypotheses which stated that with data mining methods it is possible to mine (produce) the model of the product lifecycle that can be applied to the new product lifecycle analysis.

The scientific and practical results of the Ph.D. Thesis are the following:

1. The multiagent product lifecycle management support system is developed. The system supports a manager with additional information about the market state of a product and progress directions, which promotes making well-reasoned decisions while choosing planning, manufacturing and advertising strategies. It is concluded that by combining the intelligent agent technology with data mining methods it is possible to develop a complex management and management support systems.
2. An analysis of the concepts of the intelligent agent technology is done. The obtained results demonstrate that the application of the IAT contributes to the complex management systems design process.
3. On the basis of the main tasks of the thesis and defined specifications of the multiagent system the functioning algorithms for data management agent, data mining agent and decision support agent are developed. The obtained results point out that the application of the intelligent agent technology contributes to the formalization of the structure and internal processes of the system.
4. An analysis of popular clustering method for suitability to cluster the product demand data was performed. The concepts of suitable self-organising maps and the gravitational clustering algorithm are analysed and described. An analysis of the algorithms for suitability to cluster the short time series of different length was performed. The obtained results point out that the classical self-organising maps and the gravitational clustering algorithm do not support an option to cluster the short time series of different length.
5. The modifications of the self-organising maps and the hierarchical gravitational clustering algorithm are developed. The modifications support an option of clustering time series of different length; the structural and procedural modifications of the algorithms were made.
6. The *MEuclidean* distance measure proposed. This measure supports the calculation of the distance between time series of different length.
7. A specialised software tool is developed that realizes the conceptual and procedural structure of the proposed multiagent system and implements the developed modifications of the self-organising maps and gravitational clustering algorithm.
8. Using the developed software tool the approbation of the proposed system is performed and a comparative analysis of the proposed modified clustering algorithms and distance measures is done. The analysis of the system testing results point out that the proposed multiagent system is capable of mining the knowledge from data, create a knowledge base and apply it for analysis of new data in the autonomous mode with comparatively high accuracy.

9. An analysis of the obtained system testing results is performed. The results point out that the data mining methods can be used for building the model of the product lifecycle that can be applied for analysis of PLC of the new products.

LIST OF REFERENCES

1. *S.Paršutins, A.Sukovs, G.Kuļešova, A.Borisovs, L.Aleksejeva Datu ieguve: Programmlīdzekļi. – Rīga: Rīgas Tehniskā universitāte, SIA „Latgales Druka", 2008. – 111 lpp. ISBN: 978-9984-3798-68-4.*
2. *A.Sukovs, L.Aleksejeva, K.Makejeva, A.Borisovs. Datu ieguve: Pamati. – Rīga: Rīgas Tehniskā universitāte, SIA „Drukātava", 2006. – 130 lpp. ISBN: 978-9984-798-19-6.*
3. Adeyemi S.L. Just-in-Time Production Systems (JITPS) in Developing Countries: The Nigerian Experience // Journal of Social Sciences. – Vol.22, No.2 (2010), pp. 145-152.
4. Aitken J., Childerhouse P., Towill D. The impact of product life cycle on supply chain strategy // International Journal of Production Economics. – Vol.85, No.2 (2003), pp. 127-140.
5. Al-zu`bi H. Applying Electronic Supply Chain Management Using Multi-Agent System: A Managerial Perspective // International Arab Journal of e-Technology. – Vol.1, No.3 (2010), pp. 106-113.
6. Athanasiadis I.N., Mitkas P.A. Software Agents for Assessing Environmental Quality: Advantages and Limitations // Proceedings of 18th International Conference on Informatics for Environmental Protection: EnviroInfo 2004, Switzerland, Geneva, October 2004. – pp. 303-305.
7. Boukhelfa K., Boufaida M. A Generic Multi-Agent Architecture for the Virtual Enterprise // Lecture Notes in Informatics. – Vol.P-56, (2004), pp. 199-210. ISBN 3-88579-385-7.
8. *Chaib-Draa B., Muller J.P., (Eds.). Multiagent based Supply Chain Management. – Springer-Verlag Berlin Heidelberg, 2006. – 450 p. ISBN: 978-3-540-33875-8.*
9. Chang-Hong J., Xi-Hai D. Forecasting Demand of Short Life Cycle Products Based on Modified Bass Model // Science Technology and Engineering. – Vol.10, No.10 (2010), pp. 2577-2580. ISSN: 1671-1815.
10. Chien C.-F., Chen Y.-J., Peng J.-T. Manufacturing intelligence for semiconductor demand forecast based on technology diffusion and product life cycle // International Journal of Production Economics. – Vol.128, No.2 (2010), pp. 496-509.
11. CIMdata Inc. Product Lifecycle Management. CIMdata report. CIMdata Inc, 2002. – 9 p.
12. Cohen B. A New Approach to Strategic Forecasting // The Journal of Business Strategy. – Vol.9, No.5 (1988), pp. 38-43.
13. Cooper M.C., Maris F., Régnier P. Solving Temporally-Cyclic Planning Problems // Proceedings of 17th International Symposium on Temporal Representation and Reasoning, Paris, France, 6 – 8 Sept. 2010. – pp. 113-120. ISBN: 978-0-7695-4187-7.
14. *Dreyfus G. Neural Networks. Methodology and Applications. Second Edition. – Springer-Verlag Berlin Heidelberg, 2005. – 386 p. ISBN: 978-3-540-22980-3.*
15. *Dunham M.H. Data Mining. Introductory and Advanced Topics. – Prentice Hall, Pearson Education, 2003. – 315 p. ISBN: 0-13-088892-3.*

16. Gomez J., Dasgupta D., Nasraoui O. A New Gravitational Clustering Algorithm // Proceedings of the Third SIAM International Conference on Data Mining – May 1-3, San Francisco, USA, 2003. – pp. 83-94.

17. Grovez W., Collins J., Ketter W., Gini M. Analyzing Market Interactions in a Multi-Agent Supply Chain Environment // Lecture Notes in Business Information Processing. – Vol.52, (2010), pp. 44-58.

18. *Han J., Kamber M. Data Mining: Concepts and Techniques. Second Edition. – Morgan Kaufmann, Elsevier Inc., 2006. – 800 p. ISBN: 978-1-55860-901-3.*

19. *Japan Management Association. Kanban Just-in Time at Toyota: Management Begins at the Workplace. - Productivity Press, 1989. – 224 p. ISBN: 9780915299485.*

20. Jennings N. On agent-based software engineering // Artificial Intelligence. – Vol.117, (2000), pp. 277-296.

21. Keogh E.J., Pazzani M.J. Derivative Dynamic Time Warping // Proceedings of the First SIAM International Conference on Data Mining – April 5-7, Chicago, USA, 2001. – pp. 1-11.

22. Kirshners A., Parshutin S., Borisov A. Combining clustering and a decision tree classifier in a forecasting task // Automatic Control and Computer Science. - Vol.44, No.3 (2010), pp. 124-132. Cited: SpringerLink.

23. *Kohonen T. Self-Organizing Maps. Third Edition. – Springer-Verlag Berlin Heidelberg, 2001. – 501 p. ISBN: 3-540-67921-9.*

24. Lapide L. New Developments in Business Forecasting // The Journal of Business Forecasting Methods and Systems. – Vol.21, No.2 (2002), pp. 11-14.

25. Ming H.X.G., Lu W.F. Zhu C.F. Technology Challenges for Product Lifecycle Management / Technical report STR/04/058/SP. Singapore Institute of Manufacturing Technology, 2004. – pp. 348-353.

26. Nicholas R., Mangili G. Short Product Life Cycle Planning // IOM Control Magazine. – Vol.30, No.1 (2004), pp. 24-26.

27. Parshutin S., Aleksejeva L., Borisov A. Forecasting Product Life Cycle Phase Transition Points with Modular Neural Networks Based System // Lecture Notes in Artificial Intelligence 5633, Advances in Data Mining, Germany, Leipzig, July 20-22, 2009. – pp. 88-102. Cited: SpringerLink.

28. Parshutin S., Aleksejeva L., Borisov A. Time Series Analysis with Modular Neural Networks // Scientific Journal of Riga Technical University. Series 5, Computer Science. - Vol. 36, (2008), pp. 162-169. Cited: EBSCO.

29. Parshutin S., Borisov A. Agents Based Data Mining and Decision Support System // Lecture Notes in Artificial Intelligence 5680, Agents and Data Mining Interaction, Hungary, Budapest, 11-11. May, 2009. – pp. 36-49. Cited: SpringerLink.

30. Parshutin S., Borisov A. Agent-Based Production Planning Support System // Baltic Journal on Sustainability. Technological and Economic Development of Economy. – Vol.16, No.3 (2010), pp. 455-470. Cited: InformaWorld (ERA).

31. Parshutin S., Borisov A. Data Mining Driven Decision Support // Polish Journal of Environmental Studies. - Vol.18, No.4A (2009), pp. 8-11.

32. Parshutin S. Clustering Time Series of Different Length Using Self-Organising Maps // Scientific Journal of Riga Technical University. Series 5, Computer Science. – Vol. 31, (2007), pp. 104-110.

33. Parshutin S., Kirshners A. Intelligent Agent Technology in Modern Production and Trade Management // Efficient Decision Support Systems: Practice and Challenges – From Current to Future / Book Chapter. INTECH, 2011, In Press, 22 p.

34. Parshutin S., Kuleshova G. Time Warping Techniques in Clustering Time Series // Proceedings of 14th International Conference on Soft Computing, MENDEL 2008, Czech Republic, Brno, June 18-20, 2008. – pp. 175-180.

35. Parshutin S. Managing Product Life Cycle with MultiAgent Data Mining System. // Lecture Notes in Artificial Intelligence 6171, Advances in Data Mining, Germany, Berlin, 2010. – pp. 308-322. Cited: SpringerLink.

36. Pechoucek M., Vokrinek J., Becavar P. ExPlanTech: Multiagent Support for Manufacturing Decision Making // Intelligent Systems, IEEE. – Vol.20, No.1 (2005), pp. 67-74. ISSN: 1541-1672.

37. *Pyle D. Data Preparation for Data Mining. - Morgan Kaufmann, Academic Press, 1999. – 540 p. ISBN: 978-1558-605-29-9.*

38. Roy D., Anciaux D., Monteiro T., Ouzizi L. Multi-agent architecture for supply chain management // Journal of Manufacturing Technology Management. - Vol.15, No.8 (2004), pp. 745 – 755.

39. Rozenblatt F. The Perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review. – Vol.65. (1958), pp. 386-408.

40. *Russell S., Norvig P. Artificial Intelligence – A modern approach, 2nd edition. – Prentice Hall, 2003. – 1132 p. ISBN: 0137903952.*

41. *Saaksvuori A., Immonen A. Product Lifecycle Management, 3rd edition. – Springer, 2008. – 253 p. ISBN: 978-3-540-78173-8.*

42. Salvador S., Chan Ph. FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space // Intelligent Data Analysis. – Vol.11, No.5 (2007), pp. 561-580.

43. Schmidt E., Dada M., Ward J., Adams D. Using Cyclic Planning to Manage Capacity at ALCOA. // Journal of Interfaces. – Vol.31, No.3 (2001), pp. 16-27.

44. *Stark J. Product Lifecycle Management: 21st Century Paradigm for Product Realisation. – Springer-Verlag, 2005. – 441 p. ISBN: 978-1-85233-810-7.*

45. *Tan P.-N., Steinbach M., Kumar V. Introduction to Data Mining. – Addison Wesley, Pearson Education, 2006. – 769 p. ISBN: 0-321-32136-7.*

46. Van de Capelle J.-P. An examination of new product diffusion models. – RIT Printing Industry Center, 2004. – 52 p.

47. Venkatesan R., Kumar V. A genetic algorithms approach to growth phase forecasting of wireless subscribers // International Journal of Forecasting. – Vol.18, No.4 (2002), pp. 625-646.

48. *Weiss G. (Ed.). Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence. – The MIT Press, 1999. – 619 p. ISBN: 978-0262-73131-7.*

49. Wen Z.-H., Chen J.-L., Chang Q., Hu J.-L. State-action pairs reduction for strong cyclic planning via state reachability. Proceedings of International Conference on Fuzzy Systems and Knowledge Discovery, FSKD`10, Yantai, Shandong, 10-12 Aug. 2010. – pp. 1917-1921. ISBN: 978-1-4244-5931-5.

50. *Wooldridge M. An Introduction to MultiAgent Systems. – John Wiley & Sons, LTD, 2005, 3rd edition. – 348 p. ISBN: 978-0471-49691-5.*

51. Wright W.E. Gravitational clustering // Pattern Recognition Vol.9, (1977), pp. 151-166 Pergamon Press.

52. Xianhao X., Qizhi S. Forecasting for Products with Short Life Cycle Based on Improved Bass Model // Industrial Engineering and Management. – Vol.5, (2007), pp. 27-31. ISSN: 1007-5429.

53. Yelland P.M. A Model of the Product Lifecycle for Sales Forecasting. – Sun Microsystems, 2004. – 60 p.

54. Yelland P.M., Lee E. Forecasting Product Sales with Dynamic Linear Mixture Models. – Sun Microsystems, 2003. – 22 p.

55. Yongtong H., Ping L., Yuhong Y., Danian Z., Changchao M., Bode J., Shouju R. A Multiagent System for the Support of Concurrent Engineering. Proceedings of the IEEE International Conference on System, Man and Cybernetics, Beijing, China, 14-17 Oct., 1996. – pp. 959-964. ISBN: 0-7803-3280-6.

56. Business knowledge center NetMBA / Internet. - http://www.netmba.com/marketing/product/lifecycle/. Last viewed: 08.05.2011.

57. Official WEB-page of an international ECLIPS project. / Internet. – http://www.eclipsproject.com. Last viewed: 17.01.2011.

58. Official WEB-page of an international company HUNTSMAN / Internet. – http://www.huntsman.com. Last viewed: 17.01.2011.

59. *Гринглаз Л., Копытов Е. Математическая статистика с примерами решения на компьютере: Учеб. Пособие. – 2-е изд. – Рига: ВШЭК, 2002. - 326 с.*

60. Паршутин С.В. Кластеризация временных рядов с применением карт самоорганизации. // Сборник научных трудов международной конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (IMSCAI'2007) – 28-30 мая, Коломна, Россия, 2007. – 465-472 с.

61. *Хайкин С. Нейронные сети. Полный курс. Второе издание. Перевод с английского. – Вильямс, 2006. – 1104 с.*