Int J Adv Manuf Technol (2013) 66:1725–1743
DOI 10.1007/s00170-012-4453-0

ORIGINAL ARTICLE

Adaptive process control based on a self-learning mechanism in autonomous manufacturing systems

Seid Žapčević · Peter Butala

Received: 29 March 2012 / Accepted: 6 August 2012 / Published online: 29 August 2012

Abstract To survive in the highly competitive global economy, manufacturing systems must be able to adapt to new circumstances. An important prerequisite for adaptation is the ability to learn, a process based on knowledge discovery and growth. The aim of this research is to uncover knowledge by examining a large volume of real-time manufacturing data collected during manufacturing operations and to use the insights gained to support decision-making and adaptive process control. The paper presents the concept of a self-learning autonomous work system. This concept introduces a learning loop into a manufacturing system composed of data acquisition, data mining (DM), and knowledge-building models. Two methods for DM are applied. A descriptive DM method enables discovery of patterns in data that may contribute to a better understanding of the manufacturing processes. A predictive process provides knowledge in the form of rules, which can then be used for enhanced decision-making. To illustrate the utility of the knowledge models, the concept of adaptive process control is introduced and implemented in a high pressure die-casting domain. A case study based on industrial data collected during die-casting operations provides a demonstration of the concept.

S. Žapčević
Technical Faculty, University of Bihać,
Bihać, Bosnia and Herzegovina

P. Butala (✉)
Department of Control and Manufacturing Systems,
University of Ljubljana,
Ljubljana, Slovenia
e-mail: peter.butala@fs.uni-lj.si
URL: www.lakos.fs.uni-lj.si

P. Butala
Department of Mechanical Engineering Science,
University of Johannesburg,
Johannesburg, South Africa

Keywords Manufacturing system · Data mining · Knowledge discovery · Adaptive process control · High-pressure die casting

1 Introduction

Manufacturing enterprises are facing an ever more turbulent environment where customised and personalised products are in demand, new technologies are continuously emerging, markets are frequently shifting, national economies are affected by severe crises, and competition is globally increasing. The rapidly changing conditions, needs, and opportunities of the twenty-first-century global market are forcing manufacturing enterprises to steadily adapt themselves to new situations.

An important prerequisite for adaptation is the process of learning. Systems with learning capabilities are the only systems that are able to adapt themselves to emerging situations and changing conditions. A continuous system's adaptation results in its growth or evolution over time.

Of course, many different sources of knowledge and ways of learning exist that could be put into practice in a manufacturing system. These knowledge sources may include everything from conventional textbooks and handbooks to digital libraries and other electronic resources. Various other learning and training methods can be implemented, from classical education to learning-by-doing and modern e-learning techniques.

Knowledge discovery in databases (KDD) is an advanced learning technique. This technique facilitates learning by data mining (DM) of actual data collected during the operation of a business system and stored in large databases and/or data warehouses. KDD enables detection of patterns of behaviour and causalities hidden in the data and thus represents a valuable source of new specific knowledge, which may contribute to a better understanding of the observed system and to enhanced management and control.

In this paper, the process of learning from large databases of manufacturing operational data is described. In modern manufacturing systems, operational data are collected on-line and in real time by supervisory control and data acquisition (SCADA) and/or manufacturing execution systems (MES). The SCADA and MES databases usually comprise large volumes of historical and actual data on processes, operations, resources, workpieces, and environment. These data represent a valuable asset and a source of knowledge that must be extracted in a meaningful way and properly managed for utility in a system's operations.

This paper addresses the issue of knowledge discovery and management in manufacturing systems. The concept of a self-learning mechanism for autonomous work systems (AWS) is proposed, based on on-line acquisition of production data, warehousing of the data, DM, knowledge elicitation, and management. Due to the scope of the issue, this research is limited to learning from process data with the objectives of providing a better understanding of manufacturing processes and enhancing process control to improve process performance. Two different DM techniques are applied: (1) descriptive DM for understanding patterns in process data and (2) predictive DM for decision making in adaptive process control. Three levels of knowledge models are introduced: (1) a meta-meta-model on a generic process level, (2) a meta-model (which is an implementation of the meta-meta-model in a certain process domain), and (3) a knowledge model, which represents specialisation of the meta-model for a specific dataset. Additionally, the concept of adaptive process control is described in detail.

The approach is implemented in high-pressure die-casting processes, which are broadly used in modern industries such as the automotive and consumer electronic industries; the approach is verified on an industrial dataset from a die casting foundry.

2 Literature review

2.1 DM methods

With the development and implementation of computerised information systems, more and more data are collected during various business processes and stored in large databases. Traditional statistical methods and tools are not capable of analysing such large volumes of data. Hence, new methods for dealing with large volumes of data in databases are becoming extremely important with the objective of extracting useful information and knowledge for decision-making; these methods include knowledge discovery in databases and DM methodologies.

The KDD approach, as noted in Ref. [1], seeks new knowledge in a certain application domain. This approach is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. The process can be generalised to non-database sources of information, although it emphasises databases as the primary source. In general [2], there are two main objectives in KDD: (1) validation of a hypothesis and (2) discovery of new patterns. Discovery can be further divided into 'prediction', where the knowledge extracted enables better forecasting of the values of the entities represented in the dataset, and into 'description', where the extracted knowledge is aimed at improving comprehension of the patterns discovered.

KDD is composed of several steps implemented in a sequence. Each subsequent step is initiated upon successful completion of the previous step and requires the result generated by the previous step as its input. The KDD procedure can be defined by a model [3]. In the context of this research, a six-step KDD process model is adopted as defined by Cios and Kurgan [4]. The model is composed of the following steps: (1) understanding the problem domain, (2) understanding the data, (3) preparing the data, (4) mining of data, (5) evaluating the discovered knowledge, and (6) using the discovered knowledge. In KDD, DM can be seen as the key process that leads to knowledge discovery.

Fayyad et al. [3] define DM as a step in the KDD process that consists of applying data analysis and discovery algorithms that (under acceptable computational efficiency limitations) produces a particular enumeration of patterns (or models) over the data. DM denotes a combination of concepts and algorithms from machine learning, statistics, artificial intelligence, and data management [5]. DM has been successfully used in many domains, such as banking, finance, marketing, insurance, science, and engineering, for discovering hidden relations and causalities in data.

2.1.1 Descriptive DM methods

The descriptive approaches, as depicted in [6], fall into two categories: identifying interesting patterns in the data and clustering the data into meaningful groups. Algorithms for discovery of patterns in large datasets, such as sales data, are the key success stories of DM research [7]. Additionally, algorithms for pattern discovery in time series datasets, which are collected by process monitoring, have been developed [8, 9]. The patterns discovered from process data can provide insights into the relationship between different features and can also be used to discover association rules.

Another descriptive DM method is clustering, which is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). According to

Ref. [10], the existing clustering methods can be categorised according to several criteria. Based on the method used to identify the clusters, they can be classified into partitioned, hierarchical, density-based, and grid-based algorithms [2]. Clustering is an interesting, useful, and challenging problem. It has great potential in applications like object recognition, image segmentation, and information filtering and retrieval.

2.1.2 Predictive DM methods

Bharatheesh and Iyengar [11] define predictive DM as the process of automatically creating a classification model from a set of examples, called the training set, which belongs to a set of classes. Once a model is created, it can be used to automatically predict the classes of other unclassified examples. Predictive DM is applied to a range of techniques that find relationships between a specific variable (called the target variable) and the other variables in a data set. Predictive DM methods can be implemented for classification, value prediction, and association rules, among others. Commonly used predictive methods include statistical methods, decision trees, rule algorithms and their hybrids, artificial neural networks, and support vector machines.

The statistical methods used for classification and prediction include Bayesian methods and regression models [1]. Decision trees, rule algorithms, and their hybrids form another category of predictive DM and are described in [12]. Decision trees are constructed by analysing a set of training examples for which the class labels are known [13]. The decision trees are then applied to classify previously unseen examples. If trained on high-quality data, decision trees can make very accurate predictions. Examples of advanced decision tree algorithms are ID3, C4.5, ID5R, and 1RD. According to Ref. [14], the most well-known decision tree learner is C4.5 (C5.0 is its recent upgrade), which is widely used and has also been incorporated into other commercial DM tools (e.g., Clementine and Kepler). An artificial neural network (ANN), often referred to as a 'neural network' (NN), is a mathematical model or computational model based on biological neural networks. In other words, it is an emulation of biological neural systems [15]. According to their topology, ANNs can be classified as feed-forward neural networks and recurrent networks. Support vector machines (SVMs) are useful techniques for data classification. Boser et al. [16] define the support vector machine as a computer algorithm that learns by example how to assign labels to objects. SVM has been considered as one of the most effective supervised learning algorithm in many pattern recognition problems [17]. SVM provides better classification results than other methods such as neural networks or decision trees [18].

2.2 DM in manufacturing systems

Manufacturing is one of the modern-day domains that must deal with large databases [19]. Data collected in manufacturing are related to various manufacturing processes, such as product and process design, resource planning, operations management, material processing, assembly, material logistics, maintenance, etc.

Several cases illustrating the application of DM in industry can be found in the literature. According to [20], manufacturing enterprises collect and store large volumes of various data that are not sufficiently exploited for competitive improvement. DM, together with knowledge management and business intelligence, has the potential to change the stakes. These techniques are able to assist enterprises to collect, extract, create, and deliver manufacturing knowledge in a competitive environment.

A comprehensive review of the literature on DM applications in manufacturing, with a special emphasis on the type of knowledge mined, is provided [21]. The paper offers an overview of relationships among knowledge areas, mined knowledge and used DM techniques. Additionally, the integration of DM systems within manufacturing systems is recognised as one of the perspective areas for research. The power of mining process data for the purposes of revealing hidden correlations between process outcome and process parameters is examined in [22].

Implementation of DM in the design of products and manufacturing processes is presented in [23]. DM can be used to extract knowledge that can be used to explain the past, avoid past mistakes, and propose future improvements to past strategies to make the design more effective and efficient. Kusiak [24] outlines areas of product and manufacturing system design with a potential for DM applications. In another paper, Kusiak [25] proposes a framework for knowledge management and implementation for decision-making in manufacturing and service applications. The framework provides different decision support tools, such as decision tables and decision maps.

As clearly indicated in the literature review, KDD and DM provide effective and efficient methods and tools for learning and can be successfully implemented in manufacturing as well. However, there exists no approach that would capitalise on knowledge learned from data and use the knowledge for decision-making in a systemic way. Based on this recognition, a new concept of the manufacturing system is introduced in the next section, one which has the ability to learn from its history, i.e., from previously performed manufacturing operations.

3 Concept of a self-learning autonomous work system

3.1 Background and objectives

The objective of this research is to support decision-making in various levels of manufacturing systems with domain-specific knowledge gained from historical experiences and cognitions derived from operational data. It is expected that the new knowledge will contribute to improved management and control of manufacturing systems.

The concept originates from the work proposed by Butala and Sluga [26]. Within this work, an AWS is introduced and defined as a manufacturing system with rounded technological functionality and corresponding management functionality that is capable of and competent in performing particular manufacturing processes. As such, it is an autonomous and lean manufacturing structure and a suitable building block of intra- and/or inter-enterprise manufacturing networks.

The AWS is composed of the management entities, basic work entities (i.e., elementary work systems (EWS)), and monitoring entities. The entities are structured in two control loops to manage and control operations. The inner control loop enables real-time control of operations and is composed of planning and control by a controller along with feedback by sensor monitoring. The outer control loop enables performance-based control via management of resources and performance evaluation via feedback. The basic structure also includes the data and knowledge (D&K) base and the LAN communication infrastructure [26]. The control loops and the D&K base form an independent AWS information system, which corresponds to a manufacturing execution system (MES) and facilitates AWS autonomy as well as communication with other systems within a network.

Although the AWS concept ensures a high level of autonomy in decision-making and the communication needed for cooperation and collaboration with other elements of a network, an important capability is missing and that is the ability to learn. Only learning can facilitate adaptation and thus evolution of any systems. Therefore, the AWS concept is upgraded to the self-learning autonomous work system (SL.AWS) concept in this research.

3.2 Structure of a self-learning autonomous work system

The SL.AWS structure is revealed in Fig. 1. In addition to the basic AWS structure, which is structured in the real-time control loop and the performance management loop (as previously described), a learning loop is added on the top of the structure.

Figure 2 focuses solely on the data collection and learning loops. The learning loop is founded on the database where data collected in the real-time loop during manufacturing operations are stored. The database contains data on processes (process parameters), resources (human subjects, machines, tools, etc.), workpieces (quality parameters of input material and output components), operations (work orders, quantities, productivity, due dates, etc.), and the environment (air temperature, relative humidity and pressure, pollution, dust, noise, illumination, etc.). Data on process and environmental parameters, resource and workpiece states, and events related to production and operations are collected through three different channels: (1) on-line sampling of sensors' signals and data capturing for physical quantities, (2) periodical or event-driven sampling by communicating data from digital controllers of EWS (controller actions and on machine sensed parameters), and (3) event-driven sampling by interaction of human subjects (subjects states and operations events) with production data acquisition terminals as defined in Ref. [26].

This database represents the input for DM. The results of DM are used for knowledge elicitation. The newly discovered knowledge is then stored in the form of knowledge models in the knowledge base and managed for further use.

Knowledge models are structured on three levels. On the top level, meta-meta-models are introduced. A meta-meta-model is understood here as a reference description of meta-models of a kind and corresponds to a certain DM method, such as clustering. On the intermediate level, meta-models are grouped. A meta-model is understood as a domain-specific description of a knowledge model, e.g., for die casting, turning, grinding, etc. Each meta-model group corresponds to one meta-meta-model. On the lowest level, knowledge models are located. A knowledge model is an actual representation of knowledge based on a certain dataset and corresponds to one meta-model.

The meta-meta-models and meta-models facilitate controlled knowledge discovery, while the knowledge models represent new knowledge, which can be used for adaptive process control as well as for interactive decision support during process and operations planning, set-up procedures, quality management, forecasting of malfunctions, fault diagnosis, and maintenance planning, among others. It can also be used for discovering hidden relationships (for example between quality process parameters and the environment), which may then contribute to a better understanding of the process. The knowledge can also be accessed by the knowledge management function for other systems in the network. This newly introduced learning loop makes it possible for the SL.AWS to continuously learn from experiences regarding operations that are performed in the system and to constantly improve and evolve on this basis.

Due to the extensive scope of the issue, in this research, self-learning is limited to knowledge discovery in

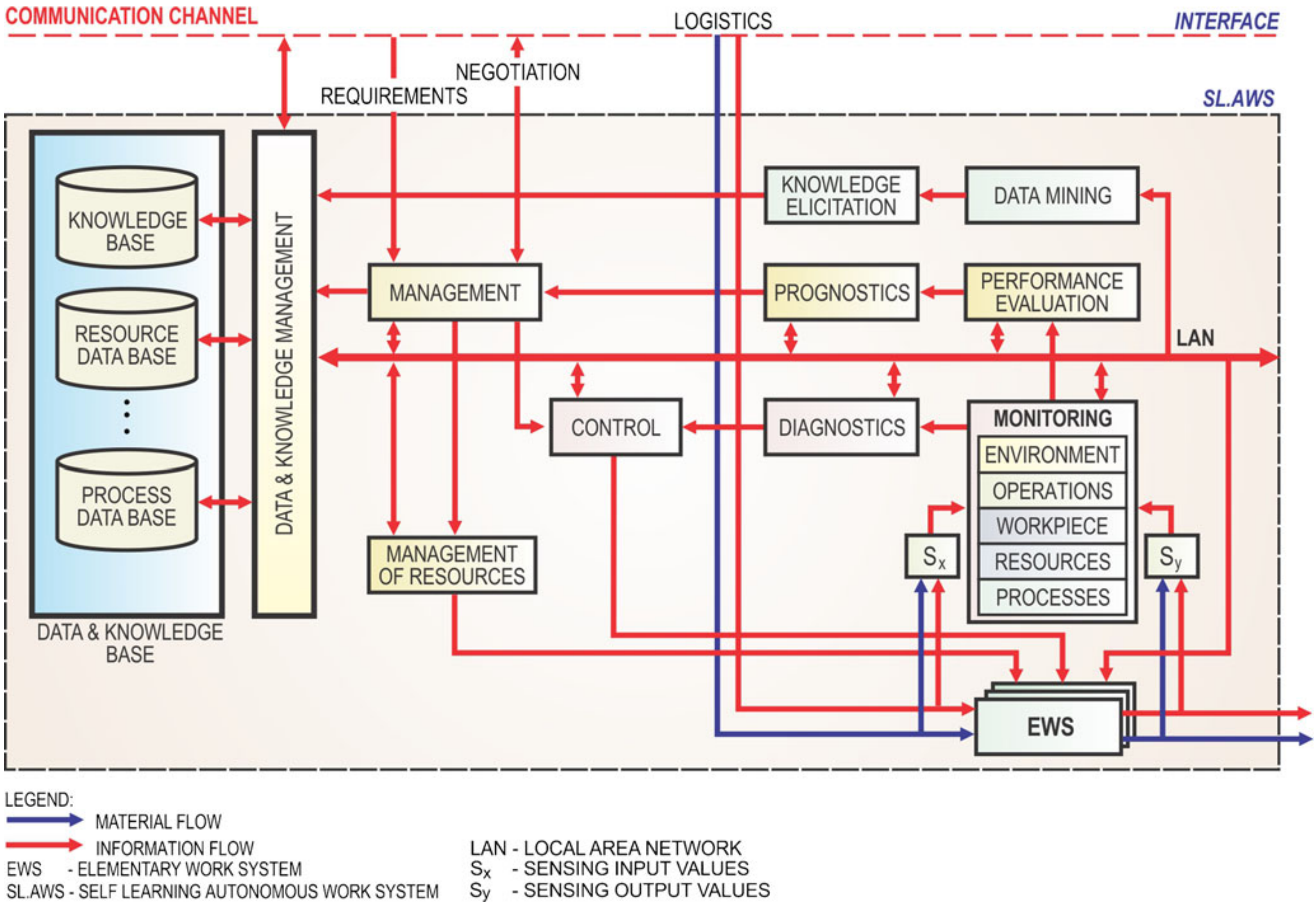

Fig. 1 Model of a self-learning autonomous work system (*SL.AWS*)

manufacturing process data. Possible implementation of new process knowledge in adaptive process control is discussed in the next section.

3.3 Building of knowledge meta-meta-models

Let us now explain how new process knowledge can be elicited in a corresponding knowledge model and made available for decision support. In SL.AWS, various manufacturing processes are carried out, such as turning, milling, welding, and high-pressure die casting (HPDC). During these operations, different process parameters are measured on-line, and the measured data are recorded in a process database. Thus, process data are collected over time during manufacturing process cycles and then stored. This dataset can be organised in a matrix **X** according to Eq. (1).

$$X = \begin{bmatrix} x_{11} & \cdot & \cdot & \cdot & x_{1p} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \;\cdot \\ \cdot & & & \cdot & \cdot \\ x_{n1} & \cdot & \cdot & \cdot & x_{np} \end{bmatrix} \qquad (1)$$

where, n is the number of instances and p is the number of attributes.

In practice, the number of instances is much greater than the number of attributes ($n>>p$), and the attributes can be of different types. In the case of process parameter measurements, the attributes are numerical values belonging to the set of real numbers ($x_{ij} \in R$).

The dataset **X** represents input for the process of knowledge discovery in databases. KDD is performed off-line in six steps [4]. In the first step, the data must be cleaned, and outliers that clearly do not correspond to the dataset have to be removed. Additionally, the attributes that are not relevant for knowledge discovery are excluded. The data cleaning process is executed by an expert from the particular domain with the aid of statistical tools. The cleaning process can be repeated several times in an arbitrary manner until the expert is satisfied with the remaining dataset.

The cleaned dataset is then ready for DM. To gain an overview of patterns in data, a descriptive DM methodology, such as partition-based clustering, is applied first. Partition-based clustering relies on a certain objective function whose minimisation is intended to lead to the

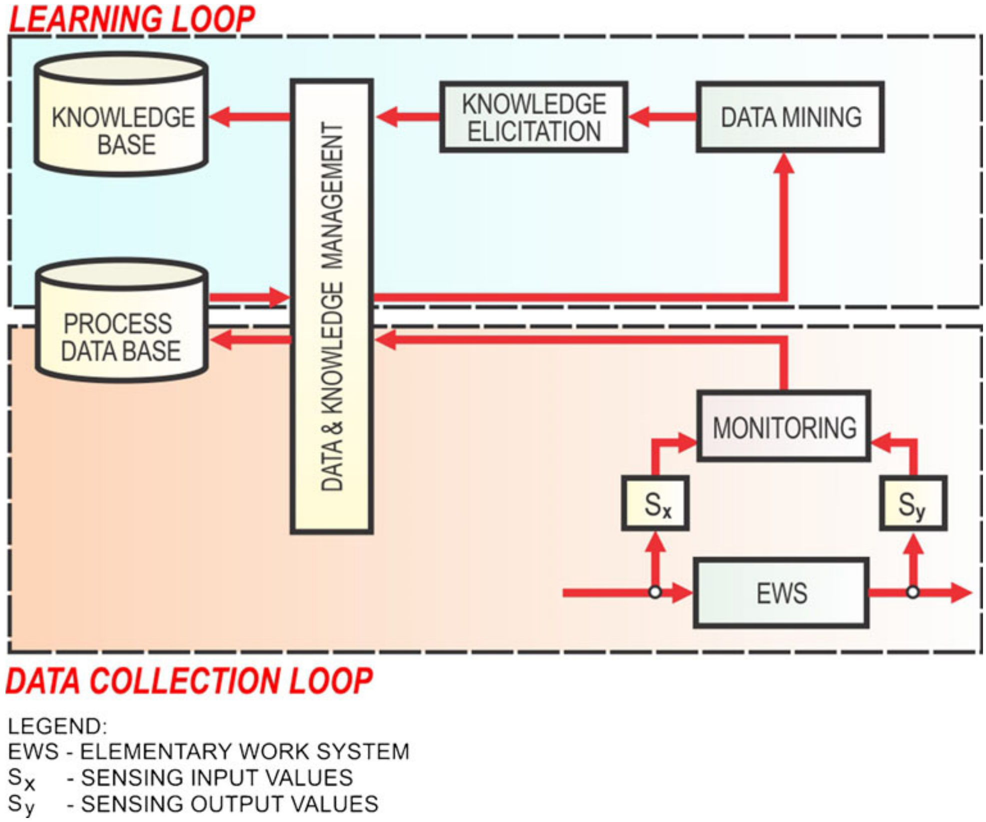

Fig. 2 Data collection and learning loops

'discovery' of the structure existing in the dataset. Clustering results in data clusters, i.e., groups of instances that evidently belong together (see Fig. 3). In the case of process datasets, centroid-based clustering generally provides useful information about patterns in data. Here, each cluster is defined by a central vector, or centroid, around which other instances of data belonging to a particular cluster are distributed. The centroids describing a certain dataset are defined in Eq. (2):

$$\left.\begin{array}{l} C_1 = (a_{11}, a_{12}, \ldots, a_{1p}) \\ C_2 = (a_{21}, a_{22}, \ldots, a_{2p}) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ C_i = (a_{i1}, a_{i2}, \ldots, a_{ij}) \end{array}\right\} \tag{2}$$

C_i ith centroid, $i=1,2, \ldots, k$

a_{ij} Value of jth attribute of the ith centroid, $i=1,2, \ldots, k$; $j=1, 2, \ldots, p$

k Is the number of centroids—clusters

p Is the number of attributes in the observed dataset **X**.

The centroids may not necessarily be members of the dataset.

Clustering is an arbitrary process guided by an expert and can be repeated until the expert is satisfied with the number of clusters k and distributions of data instances within them.

The centroids defined in Eq. (2) represent a descriptive knowledge meta-meta-model. By implementing it within a certain domain, a corresponding descriptive knowledge meta-model is obtained. In Section 4, the centroids meta-meta-model is applied to the HPDC domain by defining the concrete process attributes to be monitored and the corresponding data collected. When we implement the knowledge meta-model and corresponding DM algorithms to an actual dataset **X**, a new descriptive knowledge model is derived in terms of identified centroids that symbolise the new knowledge about the process articulated by the dataset **X**. Section 5 examines interpretation of knowledge derived from a dataset obtained during a series of die casting operations. This descriptive knowledge model can be stored in the knowledge base according to Fig. 2 and can be used for (1) definition of reference values for a machine controller and (2) selection of target values in adaptive process control.

In the next step, DM with predictive algorithms can be performed to obtain a predictive knowledge model. The objective is to discover a mathematical model that best fits the input data and can perform estimation of output process parameters during future process cycles and/or modification of input parameters to achieve target values of the output process parameters, thus improving process stability. One

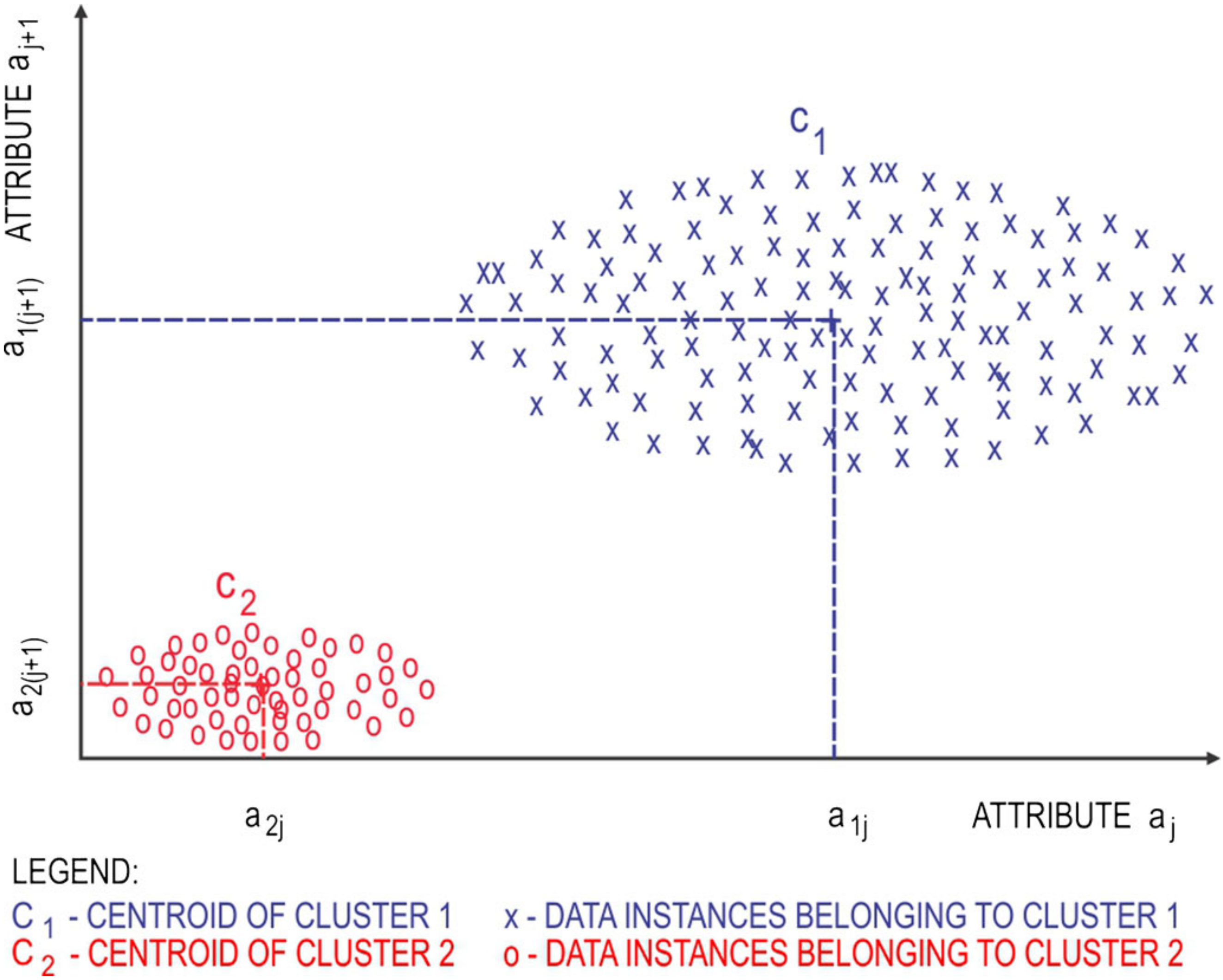

Fig. 3 Clustering of a dataset **X** in the a_j–a_{j+1} plane and the corresponding centroids C_1 and C_2

has to have in mind that the process data are numerical values (Eq. 1); therefore, a mathematical model is feasible. However, due to the large number of attributes, it is difficult to find a single model that fits the entire dataset.

Therefore, the dataset is divided into several smaller subsets for which local regression models are adjusted; these regression models fit the data well and describe a portion of the entire data space. For this purpose, rules with local regression models are generated, and their scope is then narrowed down to the condition portion of the rule (Eq. 3), that is,

$$\text{IF condition is } A_m \quad \text{THEN} \quad y = f_m(X, w_m) \tag{3}$$

The regression model (f_m) can be linear or nonlinear and applies only to the inputs x from the dataset **X** that belong to the information granule represented by A_m [1], where is the range of attribute A_m's value (range lower limit (l_m) and range upper limit (u_m)), w_m is the weight coefficient and y is the class type. The rule can be interpreted in the following manner: if an object has attribute values that fall in the ranges on the left hand side, then its class type is likely to be y (with some high probability).

For each class type y, a set of rules with corresponding local regression models is obtained that describes the given dataset **X**. In the case that the class type is numeric and all attributes are numeric as well, the linear regression models are suitable for the knowledge model. According to [12], the linear regression is an excellent, simple method for numeric prediction, and it has been widely used in statistical applications for decades. If the data exhibits a non-linear dependency, the best-fitting straight line will be found, where 'best' is interpreted as the least mean-squared difference.

Therefore, a set of rules with sets of local regression models are obtained; these sets form a predictive knowledge meta-meta-model in the following form:

$$\left.\begin{array}{l} y_1^{(1)} = w_{10}^{(1)}a_{10} + w_{12}^{(1)}a_{12} + w_{13}^{(1)}a_{13} + \ldots + w_{1p}^{(1)}a_{1p} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_1^{(q)} = w_{10}^{(q)}a_{10} + w_{12}^{(q)}a_{12} + w_{13}^{(q)}a_{13} + \ldots + w_{1p}^{(q)}a_{1p} \end{array}\right\}$$

$$\left.\begin{array}{l} y_2^{(1)} = w_{20}^{(1)}a_{20} + w_{21}^{(1)}a_{21} + w_{23}^{(1)}a_{23} + \ldots + w_{2p}^{(1)}a_{2p} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_2^{(q)} = w_{20}^{(q)}a_{20} + w_{21}^{(q)}a_{21} + w_{23}^{(q)}a_{23} + \ldots + w_{2p}^{(q)}a_{2p} \end{array}\right\}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\left.\begin{array}{l} y_j^{(1)} = w_{j0}^{(1)}a_{j0} + w_{j1}^{(1)}a_{j1} + w_{j2}^{(1)}a_{j2} + \ldots + w_{j(p-1)}^{(1)}a_{j(p-1)} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_j^{(q)} = w_{j0}^{(q)}a_{j0} + w_{j1}^{(q)}a_{j1} + w_{j2}^{(q)}a_{j2} + \ldots + w_{j(p-1)}^{(q)}a_{j(p-1)} \end{array}\right\} \quad (4)$$

where:

y_j Class type, $j=1,2, \ldots, p$
a_{jp} pth attribute in jth class type
$a_{j0}=1, j=1,2, \ldots, p$
w_{jp} pth weight coefficient in jth class type
p Is the number of attributes

(1), … , (q)—qth regression model corresponding to the rule number for the jth class type.

Equations (3) and (4) represent the predictive knowledge meta-meta-model. As in the case of descriptive knowledge meta-meta-model, by implementing this model on a certain domain, a corresponding predictive knowledge meta-model is obtained, as described in Section 4 for the die casting domain. In this model, the corresponding vector of attributes a, and class types y must be defined. For example, for the class types y, those input process parameters are selected that are controlled by a machine-tool and have influence on the values of the output process parameters. The knowledge meta-model and corresponding DM algorithms can now be implemented on an actual dataset **X** to obtain a new predictive knowledge model in terms of a matching set of rules. This specific knowledge model is stored in the knowledge base, and thus the off-line KDD procedure is completed. In the case study presented in Section 5, the predictive knowledge model in the form of rules is applied to adaptive process control in die casting.

3.4 Concept of adaptive process control

Next, an application of the descriptive and predictive process knowledge models in adaptive process control is conceptualised. The overall idea is depicted in Fig. 4 and is adapted from Ref. [27]. The core of Fig. 4 represents the so-called EWS composed of three vital elements: a process, a process implementation device, and a human subject. A more detailed description of EWS can be found in Ref. [27]. Within EWS, there is a PID control loop in which a logic controller controls the process implementation device (e.g., machine tool) over power elements (actuators). Another control loop can be implemented for adaptive process control as indicated in the upper part of Fig. 4. Here, the selected input and output process parameters must be monitored. The monitored data are used for (1) continual acquisition of data in the data collection loop (according to Fig. 2) and (2) identification of the process state in each process cycle as a basis for adaptive process control (as shown in Fig. 4).

We now explain how the adaptive process control loop acts. The objective is to accurately achieve the target values of the output process parameters and maintain a stable process, i.e., to minimise variance of the output process parameters.

At the beginning of the operation, when no particular knowledge model is available, the initial reference values (**R**) for the PID control loop, which are determined by a process planner or recorded as experienced settings from previous operations, are downloaded and set on the logic controller via the communication interface (see Fig. 4). Process cycles can now begin on this basis. Without a priori knowledge, the output process parameter data usually varies noticeably due to limited control over the process and numerous uncontrolled disturbances that influence process stability. However, in the case where the process has been already performed and the corresponding dataset has been analysed by centroid-based clustering, then the clusters and corresponding centroids are recognised and stored in the knowledge base. In this case, an expert can decide arbitrarily to which centroid he or she would like to adjust the output process parameters by defining the values of the attributes belonging to the selected centroid as the target values for process control.

In the next step, the appropriate rules that form the predictive part of the knowledge base must be selected. For this purpose, the data vector (**DATA**) from the last process cycle must be considered. These data are compared with the conditional parts of the rules (Eq. 3) to select the appropriate rule for each class type. By instantiating the target values into the selected rules (Eq. 4), one obtains the adapted reference values ($\mathbf{R_A}$), which are set on the logic controller as the new reference for the next process cycle (see Fig. 4). This procedure is repeated in each process cycle.

We now illustrate the adaptive control procedure in the example shown in Fig. 5, which shows one cluster of data instances and its centroid plotted in an a_j–a_{j+1} plane. Colour coding indicates different densities of data instances. Let us consider that the first process cycle, which is based on initial

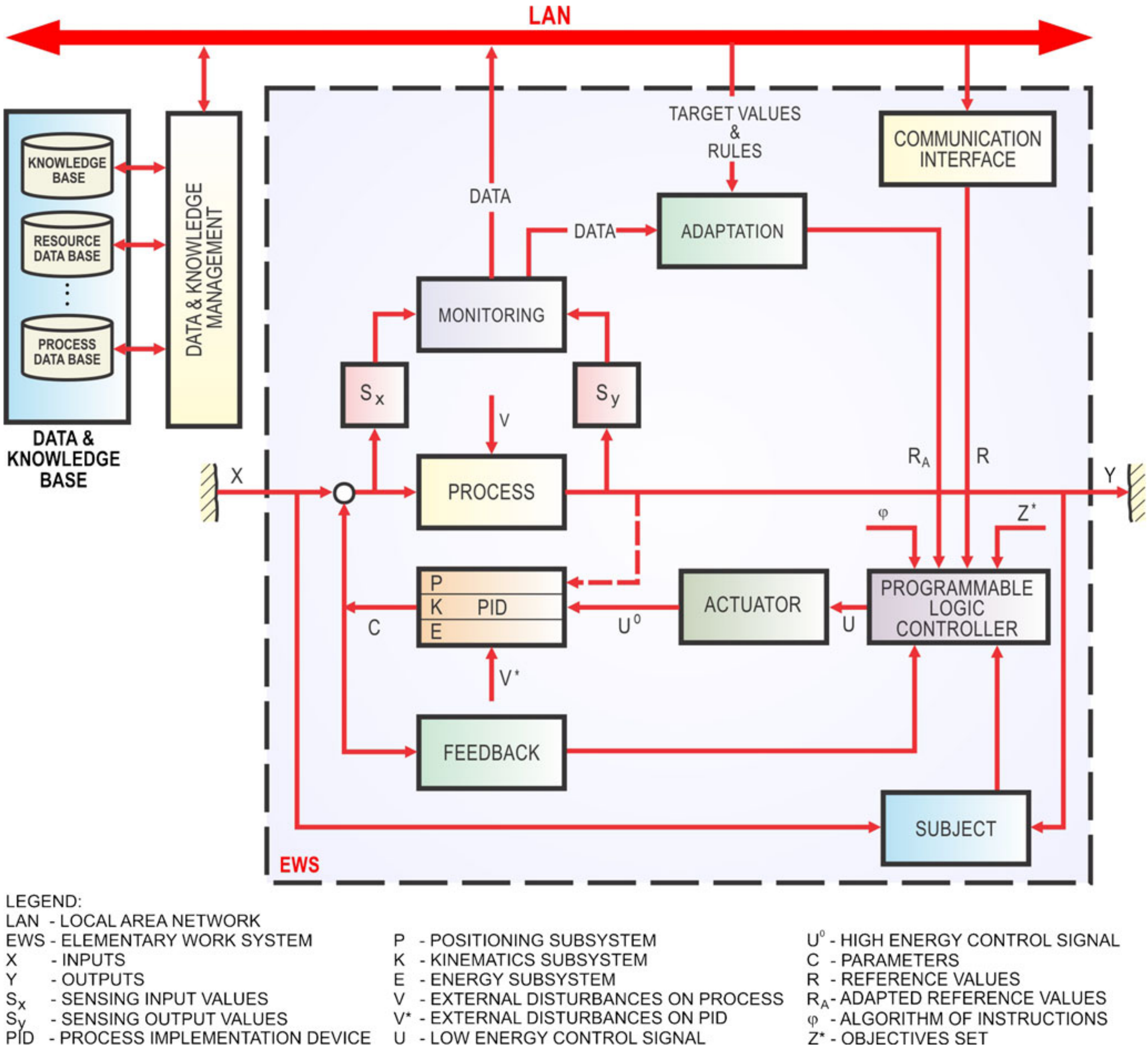

Fig. 4 Concept of adaptive process control

reference values, results in the point noted with 1. In the next cycle, the adapted reference values bring the output process parameters to the point marked with 2, and so on. As we can see, the adaptation of reference values causes convergence of the resulting output process parameters toward the target values, i.e., toward the selected centroid. Hence, it is expected that higher process stability and thus lower variance will be achieved, as indicated in Fig. 6.

4 DM and knowledge discovery in die casting cells

SL.AWS introduces the generic concept of self-learning in autonomous manufacturing systems. To develop a learning methodology and verify it in a real industrial case study, the domain of observation must be limited to a certain technology. In this research, HPDC is considered as the domain of observation.

4.1 HPDC process

HPDC is a manufacturing process for net shape production of precise metal pieces of complicated shape that are produced in large quantities. As stated in Ref. [28], the process (in comparison to alternative manufacturing processes) is very productive, but it requires a complicated and therefore expensive tool. As a consequence, die casting is the preferred technology for large lots,

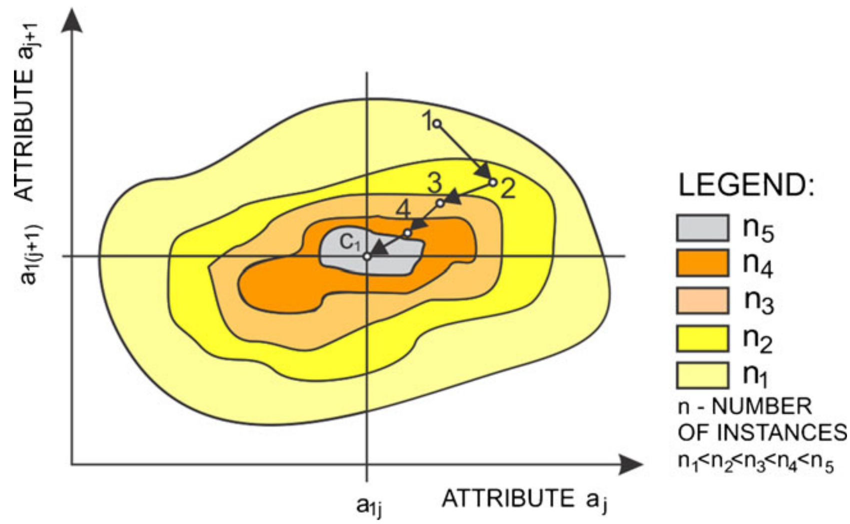

Fig. 5 Distribution of instances in cluster 1 defined with centroid C_1 and convergence of data instances toward C_1 during adaptive process control

generally of more than 50.000 pieces. HPDC is widely used in the automotive industry [29] and in manufacturing of telecommunication equipment, personal computers, and consumer electronic products [30]. High precision and quality of appearance of metal components are the most significant goals in the die casting process. The die-casting process is successfully used to produce components from aluminium and magnesium alloys due to advantages such as low cost, high productivity and stable quality. Chiang et al. [31] notes that the HPDC process technology is very complicated in practical applications and that it is crucial to properly set-up the parameters of the HPDC process to obtain good mechanical properties and the desired performance for the manufactured aluminium-alloy components.

According to Ref. [32], the quality of a die pressure casting is the result of a great number of parameters. Some of these parameters are controllable, and others are disturbance factors. The casting density is considered to be the most representative quality characteristic in the die casting process because it is related to many internal imperfections in a casting (porosity, shrinkage porosity, micro-voids, etc.).

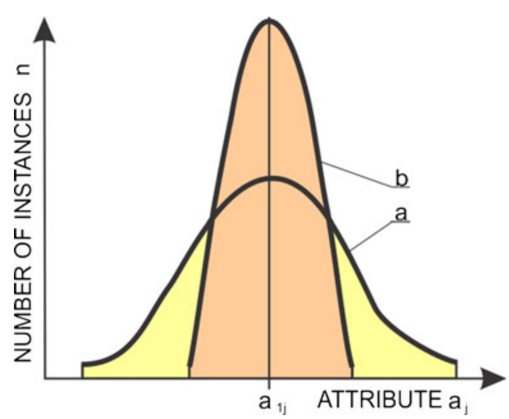

Fig. 6 Normal distribution of the attribute a_j around its centroid a_{1j} before (*a*) and after (*b*) adaptive process control

Rai et al. [33] states that the most important parameters influencing quality and productivity of the die casting process are melt temperature, mould temperature, and first- and second-phase plunger velocities. Improper selection of any of these parameters may cause defects (such as voids, sinks, distortions and cracks) in the castings and drive longer process cycle time. Similarly, Yarlagadda and Chiang [34] define four major parameters influencing the quality of die-casted parts: the melt temperature, the injection pressure, the injection time or flow rate, and the mould or cavity temperature.

In preparation of the HPDC process, the initial setting of the process operating parameters is established on an experimental basis. The setting procedure is time consuming and produces a lot of defect castings. Due to rapid expansion of the die casting process and the goal of producing better quality products in a short period of time, there is an ever-increasing demand to replace this expert-reliant traditional trial-and-error method with an advanced method based on knowledge. Therefore, knowledge of the correct injection speed, injection pressure, die temperature, and pouring temperature is very important [35].

The abovementioned issues clearly indicate that there is a need for developing a model for fast and accurate prediction of process parameters. Such a model must enable process stability, high part quality, and high productivity. As an answer to these needs, several solutions based on advanced numerical simulation techniques are emerging, such as the finite element method (FEM), finite difference method or boundary element method, and artificial intelligence. At present, various FEM-based commercial software packages are available for complex 3D-flow behaviour simulations of molten metal inside the die cavities, such as ProCast, MoldFlow™, C-Mold, etc. Excessive computational time and high software prices limit the use of this software, which is mainly applied in research by academics or by large manufacturing enterprises [33].

Artificial intelligence techniques like ANN or genetic algorithms (GAs), in combination with DoE (design of experiments) techniques like the Taguchi method, have been widely used for solving complicated and multivariable manufacturing problems [32, 34–37]. In all of these research contributions, one common characteristic is that they use experimental results or data from simulation software as well as from experts in the die casting industry for training. Despite these sophisticated models based on ANN and GAs, there is still a lack of comprehensive models that would take into account the complex relationships between the overall die casting process variables and would assist in achieving optimal results (i.e., better quality and higher productivity) from a given die casting machine by assigning appropriate values to process variables in an interactive environment [33].

4.2 DM and knowledge discovery in the CoCAST database

There were several reasons for selecting HPDC as the application area for self-learning and adaptive process control. First, HPDC is a modern and widely used technology in high-tech sectors, where customers demand high and robust quality and a reliable supply. Second, the AWS concept was studied and the adequate manufacturing execution system called CoCAST (Collaborative CASTing) was developed and implemented in a die casting foundry [38]. As a consequence, a large CoCAST database was available with process data that had been collected over several years. These data represented a real challenge for DM and knowledge discovery.

There are several highly automated die casting cells in operation in the observed HPDC foundry. Five of them are monitored by the MES systems. Among these five cells, four are identical and can produce the same castings. The identified basic process entity, which must be observed, is related to a particular casting and thus to the corresponding tool. Neither the die casting machine nor the operator has a noticeable influence on the process [39]. This cognition clearly indicates that, for learning, data related to operations with a particular tool must be taken into account.

During operations, each die casting process cycle is monitored and recorded on the cells. Each cycle record consists of values related to twelve attributes. The attributes are structured in three groups. The first three attributes are organisational (cell id, tool id, date-and-time stamp), the following five are input process parameters (time of phase 1, velocity of phase 2, pressure at phase 2, pressure at phase 3, and reaction time of phase 3) and the last four are output process parameters (pressure at return, pressure at closing, temperature of oil, and thickness of tablet). The scheme in Fig. 7 shows the measuring points in a die casting cell where the input and output process parameters are acquired.

The Weka DM suite [40] was selected for DM. Weka is an open source software program that includes numerous well-known DM algorithms and tools, and as such, offers plenty possibilities for research. The CoCAST database was examined with several Weka algorithms. The objective here was to obtain knowledge models, where knowledge is represented in the forms of the centroid meta-meta-model according to Eq. (2) and the rule meta-meta-model according to Eqs. (3) and (4), which would correspond well to the nature of the die casting process data and would enable prediction and adaptive control of the process. In particular, the statistical analysis of the CoCAST data indicated a high variance of process parameters despite computerised control of the die casting cells. Because of this variance, product quality and productivity were not as high as expected. Therefore, it was anticipated that adaptive process control based on knowledge discovered in the process datasets would decrease the process variance and increase its stability (see Fig. 6), which would then contribute to higher quality of castings and higher productivity of operations.

The process data stored in the CoCAST process database came from successfully performed process cycles that resulted in quality castings. To find possible hidden patterns in the dataset, an algorithm for descriptive DM can be selected from the Weka suite. The clusters.Simple.Kmeans algorithm turned out to be a suitable method for finding clusters in the CoCAST dataset.

To adopt the descriptive meta-meta-model defined in Eq. (2), a domain expert must select attributes from the dataset that are relevant for knowledge discovery. Three input process parameters (time of phase 1, velocity of phase 2, and pressure at phase 3) and four output process parameters (pressure at return, pressure at closing, temperature of oil, and thickness of tablet) were selected in the CoCAST database because they were recognised as the key parameters of the HPDC process. During clustering, the parameter k determining the number of clusters must be determined arbitrarily by the expert. Thus, k clusters are generated and are defined by their centroids according to the meta-model defined in Eq. (5).

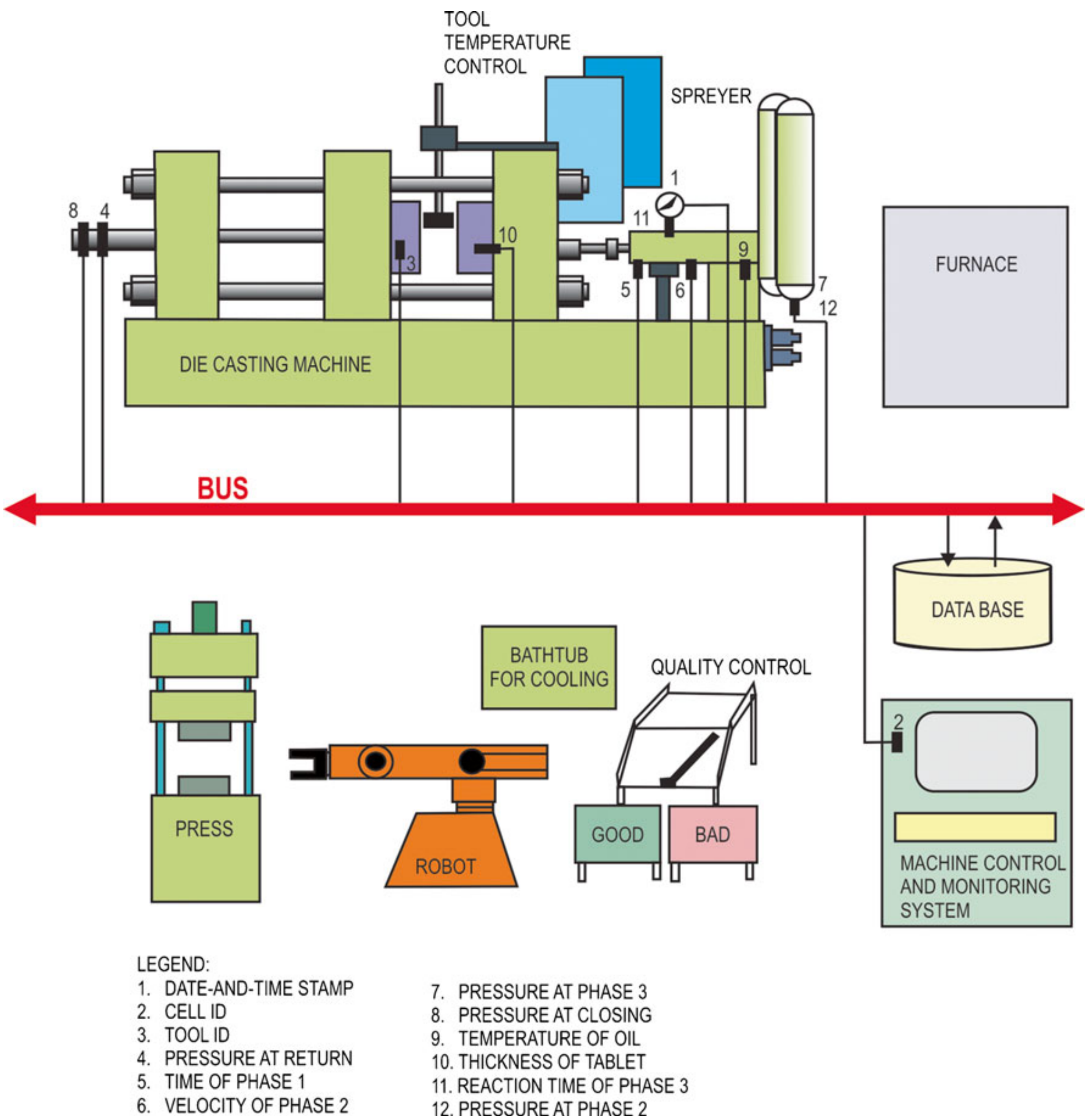

Fig. 7 Measuring points for input and output process parameter in a die casting cell

$$\left.\begin{array}{l}
C_1 = \left(\begin{array}{r}\text{TimeOfPhase1}_1, \text{VelocityOfPhase2}_1, \text{PressureAtPhase3}_1, \text{PressureAtReturn}_1, \\ \text{PressureAtClosing}_1, \text{TemperatureOfOil}_1, \text{ThicknessOfTablet}_1\end{array}\right) \\
C_2 = \left(\begin{array}{r}\text{TimeOfPhase1}_2, \text{VelocityOfPhase2}_2, \text{PressureAtPhase3}_2, \text{PressureAtReturn}_2, \\ \text{PressureAtClosing}_2, \text{TemperatureOfOil}_2, \text{ThicknessOfTablet}_2\end{array}\right) \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
C_k = \left(\begin{array}{r}\text{TimeOfPhase1}_k, \text{VelocityOfPhase2}_k, \text{PressureAtPhase3}_k, \text{PressureAtReturn}_k, \\ \text{PressureAtClosing}_k, \text{TemperatureOfOil}_k, \text{ThicknessOfTablet}_k\end{array}\right)
\end{array}\right\} \tag{5}$$

where:

$C_1, C_2, \ldots, C_k$	Is kth centroid, k=1,2, ...,k
k	Is number of centroids—clusters
TimeOfPhase1	Is the time of phase 1
VelocityOfPhase2	Is the velocity of phase 2

PressureAtPhase3	Is the pressure at phase 3
PressureAtReturn	Is the pressure at return
PressureAtClosing	Is the pressure at closing
TemperatureOfOil	Is the temperature of the oil
ThicknessOfTablet	Is the thickness of the tablet.

By analysing the DM report and visualising the results, the expert can decide which cluster should be selected for the reference. The corresponding centroid can then be taken as the vector of target values in adaptive control of the HPDC process. The target values are those values of the process parameters that are intended to be kept stable during adaptive control. Therefore, they represent new knowledge about the process, which is created from the observed dataset.

For learning and building of a predictive knowledge model, the weka.classifiers.rules.M5Rules algorithm is applied. This algorithm is selected because it generated knowledge models that best fit the learning datasets in our investigations while achieving the highest correlation with the datasets and the smallest fitting errors. The algorithm provides a set of IF…THEN rules that forms the knowledge model. In case of the die casting process, rules correlating the input process parameters with output process parameters have to be extracted. The generic form of the rules for calculation of the three input parameters (i.e., time of phase 1, velocity of phase 2, and pressure at phase 3) are given in Eqs. (6)–(8). This set of rules forms the knowledge meta-model and is universal for all die casting cells.

$$\left.\begin{array}{l}\text{TimeOfPhase1}^{(1)} = w_{11}^{(1)}\text{PressureAtReturn} + w_{12}^{(1)}\text{VelocityOfPhase2} + w_{13}^{(1)}\text{PressureAtPhase3} \\ \qquad +w_{14}^{(1)}\text{PressureAtClosing} + w_{15}^{(1)}\text{TemperatureOfOil} + w_{16}^{(1)}\text{ThicknessOfTablet} + w_{10}^{(1)} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \text{TimeOfPhase1}^{(q)} = w_{11}^{(q)}\text{PressureAtReturn} + w_{12}^{(q)}\text{VelocityOfPhase2} + w_{13}^{(q)}\text{PressureAtPhase3} \\ \qquad +w_{14}^{(q)}\text{PressureAtClosing} + w_{15}^{(q)}\text{TemperatureOfOil} + w_{16}^{(q)}\text{ThicknessOfTablet} + w_{10}^{(q)}\end{array}\right\} \tag{6}$$

$$\left.\begin{array}{l}\text{VelocityOfPhase2}^{(1)} = w_{21}^{(1)}\text{PressureAtReturn} + w_{22}^{(1)}\text{TimeOfPhase1} + w_{23}^{(1)}\text{PressureAtPhase3} \\ \qquad +w_{24}^{(1)}\text{PressureAtClosing} + w_{25}^{(1)}\text{TemperatureOfOil} + w_{26}^{(1)}\text{ThicknessOfTablet} + w_{20}^{(1)} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \text{VelocityOfPhase2}^{(q)} = w_{21}^{(q)}\text{PressureAtReturn} + w_{22}^{(q)}\text{TimeOfPhase1} + w_{23}^{(q)}\text{PressureAtPhase3} \\ \qquad +w_{24}^{(q)}\text{PressureAtClosing} + w_{25}^{(q)}\text{TemperatureOfOil} + w_{26}^{(q)}\text{ThicknessOfTablet} + w_{20}^{(q)}\end{array}\right\} \tag{7}$$

$$\left.\begin{array}{l}\text{PressureAtPhase3}^{(1)} = w_{31}^{(1)}\text{PressureAtReturn} + w_{32}^{(1)}\text{TimeOfPhase1} + w_{33}^{(1)}\text{VelocityOfPhase2} \\ \qquad +w_{34}^{(1)}\text{PressureAtClosing} + w_{35}^{(1)}\text{TemperatureOfOil} + w_{36}^{(1)}\text{ThicknessOfTablet} + w_{30}^{(1)} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \text{PressureAtPhase3}^{(q)} = w_{31}^{(q)}\text{PressureAtReturn} + w_{32}^{(q)}\text{TimeOfPhase1} + w_{33}^{(q)}\text{VelocityOfPhase2} \\ \qquad +w_{34}^{(q)}\text{PressureAtClosing} + w_{35}^{(q)}\text{TemperatureOfOil} + w_{36}^{(q)}\text{ThicknessOfTablet} + w_{30}^{(q)}\end{array}\right\} \tag{8}$$

where:

TimeOfPhase1	Is the time of phase 1
VelocityOfPhase2	Is the velocity of phase 2
PressureAtPhase3	Is the pressure at phase 3
PressureAtReturn	Is the pressure at return
PressureAtClosing	Is the pressure at closing
TemperatureOfOil	Is the temperature of the oil

ThicknessOfTablet	Is the thickness of the tablet
$w_{jp}^{(q)}$	Are weights, where j=1, 2, 3 (number of class type), p=0, 1, 2, … 6 (number of attributes), q=1, 2, 3, …, q (number of rule in ith class type).

The values of the weights are determined by the algorithm according to the actual values of the process parameters, thus fitting them to the learning dataset. In this way, a particular knowledge model is obtained. Hence, the particular knowledge model consists of a number of rules in the form of linear equations that provide sectional continuous description of the solution space. The quality of the knowledge is assessed by statistical measures such as the correlation coefficient, mean absolute error, root mean squared error, relative absolute error, and root relative squared error.

Let us briefly explain how the concept of adaptive control introduced in Section 3 can be applied in a die casting system. The main element of the system is the casting process that transforms the melted metal into a casting. Other process inputs are generated by the die casting machine (position, kinematic, and energy), which are actuated by a piston of a hydraulic motor. Interactions between the machine, melt, die (tool), and its cavity, constitute the die-casting process, which results in a casting (with a tablet), heat, and other effects as outputs. The machine is controlled by a programmable logic controller. The controller converts input reference values into control signals, which are conducted to the machine where they are actuated to the process inputs. According to this description, it is clear that the controller manipulates the machine and not the process itself. Therefore, due to various effects and disturbances that occur in the system, the process is not as stable as it should be. To obtain better control over the casting process, adaptive process control should be introduced, as shown in the upper part of Fig. 4. On-line measuring of certain input and output parameters and their association with the conditional part of a rule enables selection of the appropriate rule from the rule set for each process parameter according to the knowledge model represented in Eqs. (6)–(8). Then, the corresponding rule facilitates calculation of adapted input process parameters according to a specific objective, which can be determined form the target centroid. Thus, it is expected that the casting process will be better controlled and will result in smaller variance of the output process parameters.

It should be mentioned that the form of the knowledge model defined in Eqs. (6)–(8) is represented by the meta-model and can be implemented in any die casting cell. During learning on actual process data, the knowledge meta-model is then instantiated to the actual knowledge model, which is (as was already explained) attributed to a tool and thus to a product.

The presented approach is illustrated in an industrial case study portrayed in the next section.

5 Case study

For illustration of these concepts, a portion of data was extracted from the CoCAST database. The selected data were matched to die casting operations related to a particular batch, which corresponds to production of a certain casting with a particular tool on a particular die casting cell. Such a dataset represents a basic learning unit.

The investigated batch was produced over a period of two and half years in several lots with shorter or longer interruptions between them. The investigated dataset is composed of 56.225 instances, which were recorded between the following time intervals: 23 September–02 November 2007 (12.437 records), 28 April–03 May 2008 (2.859 records), 06–18 June 2008 (14.338 records) 16 October–20 November 2008 (23.996 records), 06–09 April 2009 (378 records), 29 May–09 June 2009 (6.637 records), 24 June–07 July 2009 (14.121 records), 24–24 September 2009 (14.154 records), and 17–25 March 2010 (5.638 records). It should be noted that the instances were recorded only if the entire die-casting cycle was successfully performed and resulted in a quality casting. In other cases, the appropriate alarm messages were recorded.

The selected dataset was processed within the Weka environment according to the KDD steps. First, the data were cleaned by a die casting expert with the help of statistic and visualisation tools. The objective of this step is to eliminate outlying data that do not correspond to the process from the dataset. In addition, attributes that are not relevant for the selected dataset and objectives of KDD are excluded.

After cleaning, the remaining dataset includes 56.047 instances and 7 attributes out of 12. The statistical characteristics of the attributes of the remaining dataset are given in Table 1.

To gain an overview of the data, the Weka clustering algorithm clusterer Simple.KMeans was applied first. The results are two clusters, i.e., cluster 1 and cluster 2, the centroids of which are given in Table 2.

Visualisation of the results is shown in Fig. 8, which clearly indicates how the instances are grouped in two distinctive clusters—clusters 1 and 2. The results are visualised in four different planes (Fig. 8a–d) showing the clusters in accordance with the attribute pairs. In Fig. 8a, one can see that all instances belonging to cluster 1 have the value of the attribute PressureAtPhase3 distributed around the value 355 bar while the instances belonging to cluster 2 are distributed around the value 370 bar. Similarly, as shown in Fig. 8b, the value of the attribute ThicknessOfTablet of the instances belonging to cluster 2 is equal to 0 mm, and

Table 1 Statistical values of the selected attributes

Attribute					Statistic value			
Name	Type	Missing (%)	Distinct	Unique (%)	Min	Max	Mean	SD
TimeOfPhase1 (ms)	Numeric	0 (0)	904	98 (0)	1.337	3.211	1.828,02	160,70
VelocityOfPhase2 (m/s)	Numeric	0 (0)	13.196	5.285 (9)	1,40	3,00	2,57	0,46
PressureAtPhase3 (bar)	Numeric	0 (0)	13	1 (0)	354,60	370,37	359,11	6,69
PressureAtReturn (bar)	Numeric	0 (0)	3.373	407 (1)	19,82	121,99	38,10	21,24
PressureAtClosing (bar)	Numeric	0 (0)	40.735	29.868 (53)	85,06	100,00	94,86	3,07
TemperatureOfOil (°C)	Numeric	0 (0)	1.396	109 (0)	24,68	48,06	38,09	3,62
ThicknessOfTablet (mm)	Numeric	0 (0)	221	25 (0)	0	33,90	15,35	9,65

this occurred at the beginning of the data recording for a certain period of time. A query in the database returned that this event occurred during the period from 23 September to 02 November 2007 (12.243 instances recorded) and in the period from 28 April to 03 May 2008 (2.813 instances recorded). Altogether, 15.036 instances in a row have the same value (0) for the attribute ThicknessOfTablet, and all of them belong to cluster 2. Fig. 8c confirms that the instances belonging to cluster 2 have the value of attribute ThicknessOfTablet equal to 0; additionally, the instances belonging to cluster 1 are distributed around the value 38,27 bar for the attribute PressureAtReturn and around the value 21,11 mm for the attribute ThicknessOfTablet. The thick black line in Fig. 8d indicates the cluster centroid (1.874,31 ms) around which most of the instances of cluster 1 (from the attribute TimeOfPhase1) are distributed.

As we can see, the use of DM and knowledge discovery software provides new insight into the die casting process and its characteristics.

As explained in Chapter 4, another type of knowledge is needed for adaptive control, which must be structured in a set of rules. For the die casting case, the weka.classifiers.rules.M5Rules algorithm turned out to be the most suitable. Therefore, the selected dataset was processed with the mentioned algorithm, and a set of rules explaining the relationship of the three input process parameters, namely, TimeOfPhase1, VelocityOfPhase2, and PressureAtPhase3, was generated. Thus, three subsets of rules form the knowledge model. Table 3 summarises the statistical features of the rule subsets. One can see that the number of rules is quite high, especially for the indirectly controlled input parameters TimeOfPhase1 (136 rules) and VelocityOfPhase2 (182 rules). This fact indicates that (1) the interrelations among process parameters are quite complex and (2) the model is fine-grained. The values of the correlation coefficient and errors (Table 3) confirm this last observation.

From the statistical characteristics of the selected attributes given in Table 1, one can notice high deviations of values except for the attribute PressureAtPhase3. This observation is because this attribute is directly controlled on the die casting machine, while the other two input attributes (i.e., TimeOfPhase1 and VelocityOfPhase2) are indirectly controlled. The standard deviations of these two parameters reflect the variance and dynamics of the resisting load, which influences the kinematics and dynamics of the plunger that pushes molten metal into a die.

Next, we examine how variances of process parameters can be reduced to achieve better process stability, as explained in Chapter 4. Consider that the objective of adaptive control would be more narrowly distributed functions dispersed around the centroid values. For the target values, one can take the centroid values of cluster 1 (Table 2), which is the most significant cluster. Based on this assumption, the target values for the process parameters would be:

Table 2 Cluster centroids

Attribute	Full data 56.047 (100 %)	Cluster 1 40.719 (73 %)	Cluster 2 15.328 (27 %)
TimeOfPhase1 (ms)	1.828,02	1.874,31	1.705,04
VelocityOfPhase2 (m/s)	2,57	2,84	1,86
PressureAtPhase3 (bar)	359,11	355,00	370,00
PressureAtReturn (bar)	38,10	38,27	37,64
PressureAtClosing (bar)	94,85	94,73	95,18
TemperatureOfOil (°C)	38,09	37,21	40,43
ThicknessOfTablet (mm)	15,35	21,11	0,07

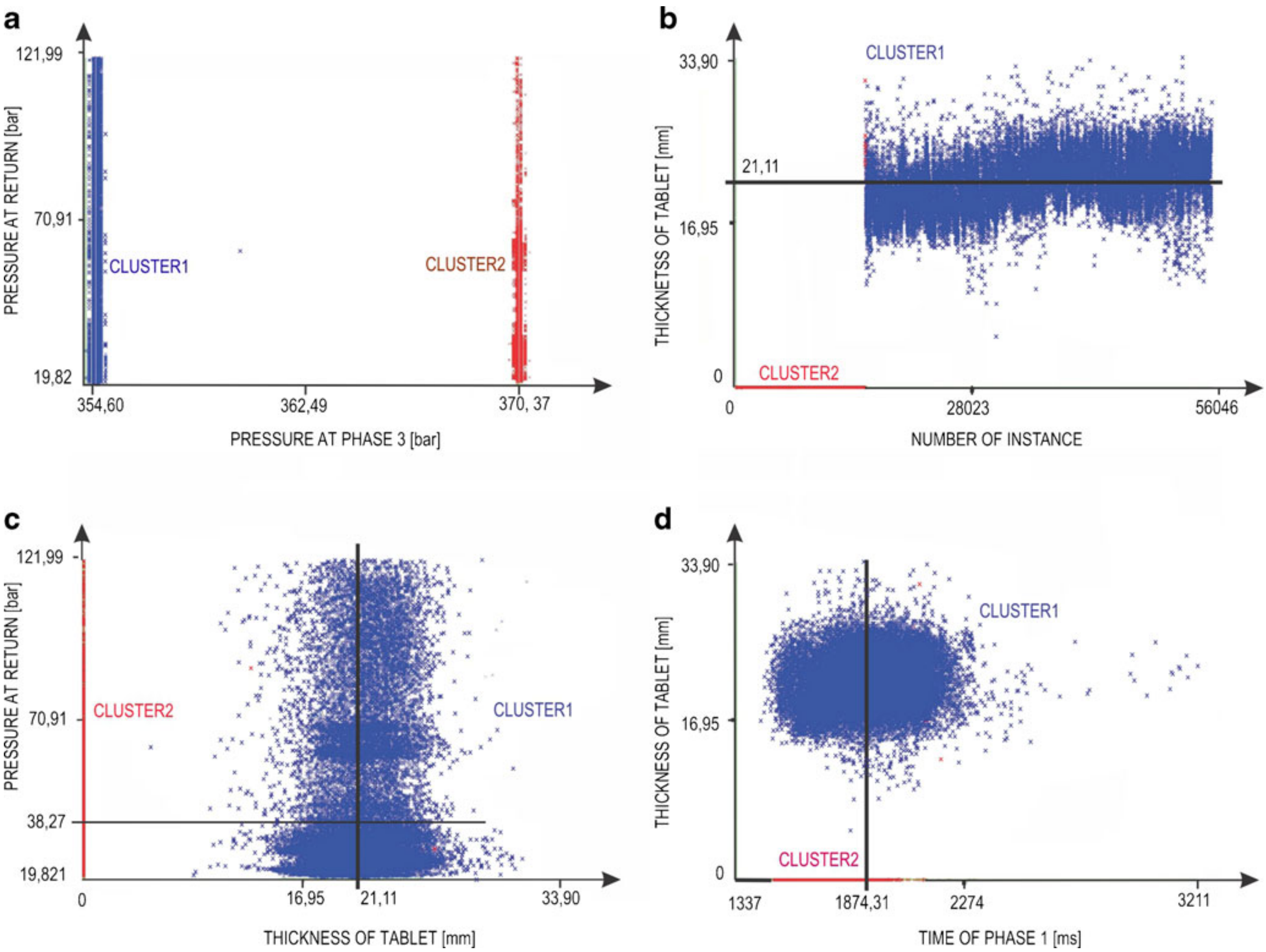

Fig. 8 Visualisation of the results of clustering

PressureAtReturn=38,27 bar, TimeOfPhase1=1.874,31 ms, VelocityOfPhase2=2,84 m/s, PressureAtPhase3=355 bar, PressureAtClosing=94,73 bar, TemperatureOfOil=37,21 °C, and ThicknessOfTablet=21,11 mm.

The rule set and centroids form the knowledge model for the particular case. Next, we consider how the knowledge model can be applied for adaptive process control. To calculate the new reference values for the input process parameters in the next process cycle, the values of the parameters of the last process cycle must be known. These values dictate selection of the appropriate rules, which are then used for calculation of the new reference values for TimeOfPhase1, VelocityOfPhase2, and PressureAtPhase3.

For example, let us assume that the measured values of the last process cycle are: PressureAtReturn=90,10 bar, TimeOfPhase1=1.905 ms, VelocityOfPhase2=2,88 m/s,

Table 3 Cross-validation of models

M5Rules	TimeOfPhase1	VelocityOfPhase2	PressureAtPhase3
Number of rules	136	182	9
Time to build the model (s)	3.237,28	5.700,42	8,05
Correlation coefficient	0,79	0,99	0,999
Mean absolute error	67,56	0,04	0,07
Root mean squared error	98,86	0,07	0,17
Relative absolute error (%)	50,70	9,51	1,26
Root relative squared error (%)	61,52	15,91	2,60
Total number of instances	56.047	56.047	56.047

PressureAtPhase3=355,04 bar, PressureAtClosing=96,67 bar, TemperatureOfOil=37,93 °C, and ThicknessOfTablet=20,60 mm. Next, one can select the appropriate rules from the rule set. Selection of an appropriate rule is performed by verification of the conditional part of a rule. For example, the conditional part of Rule 1 for TimeOfPhase1 states:

IF

$$\begin{aligned} \text{ThicknessOfTablet} &> 6{,}85 \\ \text{PressureAtReturn} &<= 32{,}19 \\ \text{PressureAtClosing} &<= 92{,}21 \\ \text{TemperatureOfOil} &> 39{,}85 \\ \text{VelocityOfPhase2} &<= 2{,}92 \end{aligned}$$

It can be observed that the conditional part of the rule is not satisfied because the following conditions are not satisfied:

PressureAtReturn	Should be smaller than or equal to 32,19, but in our case it is greater (90,10)
PressureAtClosing	Should be smaller than or equal to 92,21, but in our case it is greater (96,67)
TemperatureOfOil	Should be greater than 39,85, but in our case is smaller (37,93).

The first rule in the rule set, whose conditional part is completely satisfied, is Rule 25. The action part of the rule then gives the formula for calculation of the TimeOfPhase1:

$$\begin{aligned} &\text{TimeOfPhase1} = 0{,}5689 \cdot \text{PressureAtReturn} + 1.036{,}8386 \cdot \\ &\text{VelocityOfPhase2} + 67{,}4787 \cdot \text{PressureAtPhase3} - 0{,}0899 \cdot \\ &\text{PressureAtClosing} - 15{,}4466 \cdot \text{TemperatureOfOil} + 20{,}4164 \cdot \\ &\text{ThicknessOfTablet} - 24.902{,}1816 \end{aligned} \tag{9}$$

Now we can substitute the variables with the target values (from Table 2 for Cluster 1):

$$\begin{aligned} \text{TimeOfPhase1} &= 0{,}5689 \cdot 38{,}27 + 1.036{,}8366 \cdot 2{,}84 \\ &+ 67{,}4787 \cdot 355 - 0{,}0899 \cdot 94{,}73 - 15{,}4466 \cdot \\ &37{,}21 + 20{,}4164 \cdot 21{,}11 - 24.902{,}1816 \\ &= 1.866{,}81264 \end{aligned} \tag{10}$$

Thus, we obtain the final result for the process parameter TimeOfPhase1=1.866,81 ms, which can be considered for adaptive control.

Analogously, for calculation of VelocityOfPhase2, the first rule whose conditional part correspond to the actual values of process parameters is Rule 7:

IF

$$\begin{aligned} \text{PressureAtPhase3} &<= 362{,}49 \\ \text{TemperatureOfOil} &<= 38{,}56 \\ \text{TimeOfPhase1} &> 1.763{,}50 \\ \text{TimeOfPhase1} &<= 1.922{,}50 \\ \text{TemperatureOfOil} &> 36{,}06 \\ \text{PressureAtClosing} &> 94{,}10 \\ \text{PressureAtClosing} &<= 97{,}22 \\ \text{ThicknessOfTablet} &<= 22{,}25 \\ \text{PressureAtClosing} &> 94{,}98 \end{aligned}$$

THEN

$$\begin{aligned} \text{VelocityOfPhase2} = {} & 0 \cdot \text{PressureAtReturn} - 0 \cdot \text{TimeOfPhase1} \\ & + 0{,}0001 \cdot \text{PressureAtPhase3} + 0{,}0002 \cdot \\ & \text{PressureAtClosing} + 0{,}0003 \cdot \text{TemperatureOfOil} \\ & - 0{,}0004 \cdot \text{ThicknessOfTablet} + 2{,}9162 = 2{,}97 \end{aligned} \tag{11}$$

Hence, the calculated reference value for the next cycle for VelocityOfPhase2=2,97 m/s.

The last calculation is performed for PressureAtPhase3. Here, several rules in the rule set fit the actual values. Therefore, the first one that fits the condition is selected, which is Rule 1:

IF

$$\begin{aligned} \text{ThicknessOfTablet} &> 2{,}60 \\ \text{TimeOfPhase1} &<= 2.021{,}50 \end{aligned}$$

THEN

$$\begin{aligned} \text{PressureAtPhase3} = {} & 0 \cdot \text{PressureAtReturn} + 0 \cdot \text{TimeOfPhase1} \\ & - 0{,}0029 \cdot \text{VelocityOfPhase2} - 0{,}0002 \cdot \\ & \text{ThicknessOfTablet} + 355{,}02 = 355{,}08 \end{aligned} \tag{12}$$

Thus, the calculated reference value for PressureAtPhase3=355,08 bar.

The calculated new reference values from the vector of adapted reference values $\mathbf{R_A}$ can now be set on the logic controller for the next process cycle.

The presented case clearly indicates the usefulness of the particularised knowledge models.

6 Conclusions and discussion

In this paper, the concept of a SL.AWS is proposed. This system is based the concept of AWS and represents a potential building block for future intelligent and adaptable manufacturing networks. Learning in AWS is introduced through a learning loop that includes DM, knowledge discovery, and knowledge management components. The learning loop provides a systemic solution for knowledge discovery in databases.

The main advantage of the proposed self-learning concept is that the knowledge models rely on historical data collected during manufacturing operations on existent machinery, with real tools, in authentic industrial environments and under dynamic conditions. The historical data are stored in a database and used for processing with advance DM and knowledge discovery methods.

Newly discovered knowledge is stored in the form of knowledge models in the data-and-knowledge base. The generated knowledge models incorporate complex relations among a great number of process variables and represents adequate knowledge for supporting decision making and adaptive process control in manufacturing systems.

A three level knowledge model is developed, which is composed of the following: (1) a generic reference model (meta-meta-model), (2) a process specific reference model (meta-model), and (3) an actual model based on historical data. All models rely on mathematical formulations.

The models incorporate complex relationships among process variables and other relevant factors of the work system as well as the environment that influence process stability, product quality, and system productivity. Therefore, learning feedback is established, which enables the work system to learn continuously from experiences in its own operations.

The described knowledge meta-model is developed for HPDC. The model is generic in nature and can be used in any die casting cell that provides monitoring of specific process parameters. The meta-model is instantiated into a particular knowledge model by learning on a particular dataset.

The concept of adaptive process control in die casting is also described in the paper. The paper shows how new knowledge in the form of clusters and rules can be applied to decrease the process variance and thus increase the product quality and the productivity of operations.

The presented approach is demonstrated in an industrial case study carried out on a large set of production data, and this demonstration clearly indicates the feasibility of this approach.

Further research will be aimed at the implementation of adaptive control as well as issues of knowledge management in a manufacturing network.

Acknowledgement This work was partially supported by the Slovene Research Agency, grant no. L2-2001.

References

1. Cios KJ, Pedrycz W, Swiniarski RW, Kurgan LA (2007) Data mining. A knowledge discovery approach. Springer, New York
2. Symeonidis AL, Mitkas PA (2005) Agent intelligence through data mining. Springer, New York
3. Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery in databases. AI Mag 17:37–54
4. Cios KJ, Kurgan LA (2005) Trends in data mining and knowledge discovery. In: Pal NR, Jain LC (eds) Advanced technique in knowledge discovery and data mining. Springer, London, pp 1–26
5. Harding JA, Shahbaz M, Srinivas S, Kusiak A (2006) Data mining in manufacturing: a review. J Manuf Sci Eng 128:969–976
6. Charaniya S, Hu WS, Karypis G (2008) Mining bioprocess data: opportunities and challenges. Trends Biotechnol 26:690–699
7. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Bocca JB, Jarke M, Zaniolo C (eds) Proceedings of the 20th International Conference Very Large Data Bases. Morgan Kaufmann, San Francisco, pp. 487–499
8. Han J, Pei J, Yin Y, Mao R (2004) Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Min Knowl Disc 8:53–87
9. Seno M, Karypis G (2001) LPMiner: an algorithm for finding frequent itemsets using length-decreasing support constraint. In: Cercone N, Lin TY, Wu X (eds) Proceedings of the 2001 IEEE International Conference on Data Mining. IEEE Computer Society, Washington, pp. 505–512
10. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31:264–323
11. Bharatheesh TL, Iyengar SS (2004) Predictive data mining for delinquency modeling. In: Arabnia HR et al. (eds) Proceedings of the 2004 International Conference on Embedded Systems and Applications, Las Vegas, USA, CSREA Press, pp. 99–105
12. Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufman, San Francisco
13. Kingsford C, Salzberg SL (2008) What are decision trees? Nat Biotechnol 26:1011–1013
14. Lavrač N (1999) Selected techniques for data mining in medicine. Artif Intell Med 16:3–23
15. Singh Y, Chauhan AS (2009) Neural networks in data mining. J Theor Appl Inf Technol 5:37–42
16. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Haussler D (ed) Proceedings of the fifth annual workshop on Computational learning theory COLT 92, Pittsburgh. ACM, New York, pp. 144–152
17. Chen KY, Chen LS, Chen MC, Lee CL (2011) Using SVM based method for equipment fault detection in a thermal power plant. Comput Ind 62:42–50
18. Asl BM, Setarehdan SK, Mohebbi M (2008) Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal. Artif Intell Med 44:51–64
19. Silberglitt R, Anton P, Howell D, et al (2006) Technical Report. The Global Technology Revolution 2020, In-Depth Analyses, Bio/Nano/Materials/Information Trends, Drivers, Barriers, and Social Implications. RAND National Security Research Division, Santa Monica. Available from www.rand.org/pubs/technical_reports/2006/RAND_TR303.pdf. Accessed 5 March 2012.
20. Wang K (2007) Applying data mining to manufacturing: the nature and implications. J Intell Manuf 18:487–495

21. Choudhary AK, Harding JA, Tiwari MK (2009) Data mining in manufacturing: a review based on the kind of knowledge. J Intell Manuf 20:501–521
22. Charaniya S, Le H, Rangwala H, Mills K, Johnson K (2010) Mining manufacturing data for discovery of high productivity process characteristics. J Biotechnol 147:186–197
23. Reich Y (2005) Data mining of design products and processes. In: Maimon O, Rokach L (eds) Data mining and knowledge discovery handbook. Springer, New York, pp 1167–1187
24. Kusiak A (2006a) Data mining in design of products and production systems. In: Dolgui A, Morel G, Pereira CE (eds) Information Control Problems in Manufacturing—INCOM 2006. A Proceedings Volume from the 12th IFAC Conference. Elsevier, Oxford, pp. 49–53
25. Kusiak A (2006) Data mining: manufacturing and service applications. Int J Prod Res 44:4175–4191
26. Butala P, Sluga A (2006) Autonomous work systems in manufacturing networks. CIRP Ann Manuf Technol 55:521–524
27. Peklenik J (1988) Fertigungskybernetik, Eine neue wissenschaftliche Disziplin für die Produktionstechnik, Festvortrag analässlich der Verleihung des Georg-Schlesinger Preises 1988 des Landes Berlin, TU-Berlin
28. Faessler A, Loher M (1996) Quality control in die casting with neural networks. In: 1st International Symposium on Neuro-Fuzzy Systems. IEEE, New York, pp. 147–153
29. Cleary PW, Ha J, Prakash M, Nguyen T (2003) SPH: a new way of modelling high pressure die casting. Third International Conference on CFD in the Minerals and Process Industries, CSIRO, Available from: http://www.cfd.com.au/cfd_conf03/index.html. Accessed 23 October 2009
30. Chiang KT, Liu NM, Chou CC (2008) Machining parameters optimization on the die casting process of magnesium alloy using the grey-based fuzzy algorithm. Int J Adv Manuf Tech 38:229–237
31. Chiang KT, Liu NM, Tsai TC (2009) Modeling and analysis of the effects of processing parameters on the performance characteristics in the high pressure die casting process of Al-SI alloys. Int J Adv Manuf Tech 41:1076–1084
32. Syrcos GP (2003) Die casting process optimization using Taguchi methods. J Mater Process Technol 135:68–74
33. Rai JK, Lajimi AM, Xirouchakis P (2008) An intelligent system for predicting HPDC process variables in interactive environment. J Mater Process Technol 203:72–79
34. Yarlagadda PKDV, Chiang ECW (1999) A neural network system for the prediction of process parameters in pressure die casting. J Mater Process Technol 89–90:583–590
35. Yarlagadda PKDV (2000) Prediction of die casting process parameters by using an artificial neural network model for zinc alloys. Int J Prod Res 38:119–139
36. Yarlagadda PKDV (2002) Development of an integrated neural network system for prediction of process parameters in metal injection moulding. J Mater Process Technol 130–131: 315–320
37. Krimpenis A, Benardos PG, Vosniakos GC, Koukouvitaki A (2006) Simulation-based selection of optimum pressure die-casting process parameters using neural nets and genetic algorithms. Int J Adv Manuf Tech 27:509–517
38. Sluga A, Butala P, Peklenik J (2005) A conceptual framework for collaborative design and operations of manufacturing work systems. CIRP Ann Manuf Technol 54:437–440
39. Vrabič R, Butala P (2011) Computational mechanics approach to managing complexity in manufacturing systems. CIRP Ann Manuf Technol 60:503–506
40. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. SIGKDD Explor Newsl 11:10–18