Integration and Reuse of Data Mining in Business Processes - A Pattern-based Approach

Dennis Wegener*
Department Knowledge Discovery,
Fraunhofer IAIS,
Schloss Birlinghoven,
53754 Sankt Augustin
Germany
E-mail: dennis.wegener@iais.fraunhofer.de
*Corresponding author

Stefan Rüping
Department Knowledge Discovery,
Fraunhofer IAIS,
Schloss Birlinghoven,
53754 Sankt Augustin
Germany
E-mail: stefan.rueping@iais.fraunhofer.de

Abstract: Today’s business applications demand high flexibility in processing information and extracting knowledge from data. Thus, data mining becomes more and more an integral part of operating a business. However, the integration of data mining into business processes still requires a lot of coordination and manual adjustment. This paper aims at reducing this effort by reusing successful data mining solutions. We describe a novel approach on facilitating the integration based on process patterns for data mining and demonstrate that these patterns allow for easy reuse and can significantly speed up the process of integration. We empirically evaluate our approach in a case study of fraud detection in the health care domain.

Keywords: Data Mining Patterns; Business Processes; Reuse; Integration; BPM; CRISP.

Biographical notes: D. Wegener received his Diploma in Computer Science from the University of Bonn in 2006. In 2006 he joined the Knowledge Discovery group at the Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, where he works as key person in several industrial and EU-funded projects. Currently he works as project leader at the Integrated Data Mining group and he is doing his PhD in Computer Science at the University of Bonn. His research interests include distributed data mining and the integration of data mining into business processes.

S. Rüping studied Computer Science at the University of Dortmund and worked as research assistant at the artificial intelligence chair at the University of Dortmund, where he finished his PhD thesis on Learning Interpretable Models in 2006. Since October 2005 he is a member of the Knowledge Discovery group at the Fraunhofer Institute IAIS and is currently the leader of the Integrated Data Mining group. His research interests are on the combination of statistical and machine learning algorithms, novel learning tasks, and on the interpretability and adaptability of machine learning models.
1 Introduction

Businesses need to be more and more flexible in order to be competitive in today’s economy. Two big forces that drive flexibility are data mining, which helps enterprises to understand their customers, processes, and themselves better, and service oriented architectures (SOA), which help to be faster in implementing new business strategies and products. As data mining becomes more and more an integral part of executing a business, data mining functionality needs to be integrated into SOA in the context of existing applications (Hornick et al., 2006).

In previous work we presented an initial discussion on how to integrate data mining into business processes (Wegener and Rüping, 2010a). Here, we focus on how to enable the reuse of existing solutions that have been proven to be successful. Currently, there are two approaches of achieving this: Passing through a new CRISP process (Shearer, 2000) while being inspired by existing solutions (e.g. by personal experience or reading respective documentation and scientific papers), and reuse at implementation level (e.g. by copy-and-paste of existing code and workflows). In this paper we argue that there is a need for an approach in between.

The first approach for reuse by following the CRISP process, which describes the general procedure of performing data mining projects, does not suffice, as this approach is often too general. E.g. it has been identified that CRISP-DM lacks in the deployment phase (Rupnik and Jaklič, 2009), in guidance towards implementing particular tasks of data mining methodologies (Sharma and Osei-Bryson, 2009), and in the definition of phases important for engineering projects (Marbán et al., 2009). In addition, we detected that many redundancies and inefficiencies exist when following the CRISP standard data mining process model in parallel to standard BPM approaches (Wegener and Rüping, 2010a).

The second approach is to utilize reuse at implementation level. Modern SOA-based Business Process Management (BPM) environments, e.g. based on standards like BPEL (Jordan and Evdemon, 2007) and BPMN (White and Miers, 2008), provide flexible and user friendly environments and tools for designing, deploying and managing business applications. BPM principles, methods and tools support the creation and management of business processes by graphical modeling, (automatic) transformation into executable processes, easy deployment and easy inclusion of external services. Data mining solutions can be reused in such environments by making use of available data mining workflows and services. However, this approach is often too specific for being reusable efficiently. For example in an analysis of a large set of real-world data mining workflows (Bremer, 2010) we found out that the changes of a workflow during the lifetime of a data mining project are made to the same extent at the preprocessing and at the modeling part, which implies that understanding and representing the semantics of the data is a very important step. The changes for the preprocessing part consisted to 50% and for the modelling part to 75% of manual parameter optimizations. The challenge is to reuse such kinds of manual fine-tuning. Also, there are requirements specific to data mining. For instance, the choice of data mining algorithm has impact on the data understanding and the business understanding phase. Thus, data mining is a complex process that requires a lot of manual optimizations and is not always transferable due to the dependency to the data. A copy-and-paste approach, by taking over the data mining part from another business process, will only work if the business process into which the data mining is integrated has exactly the same properties as the original one. What is needed is a way to specify the correct level of abstraction and generalization for enabling reuse.

In this work, we aim at a formal representation of data mining processes to facilitate their integration and reuse in business processes as well as to support a faster identification and handling of problems due to changes in the business process or the data. As visualized in Fig. 1, the approach should support the description of the process at different levels of abstraction between the CRISP model as most general representation and executable workflows and code as most concrete representation. This requires to:

1. support the modeling of the data mining process as a whole, including its unique requirements, according to the CRISP tasks and phases
2. allow for the modeling and specification of tasks at implementation level in order to make use of existing code and services
3. simplify the integration into business processes in the context of modern BPM systems, which requires compatibility to relevant standards and tools.

This work contributes a new approach for the reuse and integration of existing data mining solutions in business processes in the context of modern BPM environments. This includes, based on CRISP, the definition of data mining process patterns, which encode reusable data mining solutions at the appropriate level of generality. They will be defined specific to the data mining problems to be addressed and will be based on a set of underlying data mining services. We will also define a process for applying these data mining patterns to business processes.

This paper extends the work published in (Wegener and Rüping, 2010b). The remainder of the paper is as follows: Sec. 2 introduces the field of BPM and data mining and gives an overview over the state-of-the-art and related work. In Sec. 3 we introduce our approach for a new process model for the reuse and integration of data mining into business processes. In Sec. 4 we give a case study on the integration including an example of a data mining process pattern and it’s integration. Sec. 5 concludes.
2 State-of-the-art and Related Work

A business process is a series of steps designed to produce a product or a service which includes all the activities that deliver particular results for a given customer (White, 2004). Business process management (BPM) is a discipline combining software capabilities and business expertise to accelerate business process improvement and to facilitate business innovation. In the context of BPM there exists a variety of standards for process modeling, visualization and execution as well as tools supporting features like pre-deployment validation of processes and the automation of process deployment, hiding the complexity of the distribution middleware. BPEL (Jordan and Evdemon, 2007) and BPMN (White and Miers, 2008) turned out to be some of the de-facto standards for the executable language for specifying interactions with web services resp. the graphical representation of processes. In the following, we will use the BPMN notation for the definition and visualization of process models.

Data mining, often also called knowledge discovery in databases (KDD), is the process of extracting (unknown) patterns from data (Fayyad et al., 1996). The goal of a data mining process is defined by the intended use from the user perspective and can be classified into two types: verification, where the system is limited to verifying the user’s hypothesis, and discovery, where the system autonomously finds new patterns. The discovery goal can be further subdivided into prediction, where the system finds patterns for predicting the future behavior of some entities, and description, where the system finds patterns for presentation in a human-understandable form. There exists a variety of different data mining methods and algorithms (Witten and Frank, 2005), which commonly involve the following classes of tasks: inferring rudimentary rules, statistical modeling, constructing decision trees, constructing rules, mining association rules, linear models, instance-based learning, and clustering. CRISP-DM (Shearer, 2000) is a standard process model for data mining that describes the life cycle of data mining projects as iterative process. The CRISP model consists of the following 6 phases:

- Business Understanding - understanding the project objectives and requirements from a business perspective and defining the data mining problem
- Data Understanding - getting to know the data and its quality
- Data Preparation - construct the final dataset from the initial raw data as input for the modeling
- Modeling - various modeling techniques are selected and applied, including the calibration of their specific settings
- Evaluation - assess how well the built model achieves the project objectives
- Deployment - the results of the data mining are delivered to the user.

Typically, the phases from Data Understanding to Evaluation are mainly performed only by the data miner, while the phases Business Understanding and Deployment involve both the data miner and the business user. The latter phases need a lot of knowledge transfer, coordination and manual adjustment. The modeling phase includes steps for training a model and for testing the model in order to determine its quality.
Once a model is built and considered as good during the evaluation phase, it is deployed and, e.g., used for predicting or classifying new data samples. Typically, the task of creating a data mining model is much more resource and time consuming but is performed much less frequently than the step of applying the model to the new data.

In our work we refer to data mining as part of the application that is represented by a business process. Data Mining on event log data in order to construct processes (Process Mining) and data mining on business processes in order to improve the business processes as part of the field Business Intelligence are not in the scope of this paper. The question that we want to answer is how can we best reuse data mining as part of a bigger business processes.

In (Russell et al., 2006) a set of workflow patterns describing the control-flow perspective of workflow systems is defined. Such process patterns have plenty of advantages (Atwood, 2006): BPM processes serve as both the specification and the source code. The modeled processes become the solutions deployed and provide a simple communication tool between end-users, business analysts, developers and the management. Process patterns provide a proven and simple technique to shorten the learning curve and improve productivity and quality of the processes designed as they are simple to understand, learn and apply immediately. Thus, we propose to provide a similar approach for the integration and reuse of data mining - process patterns that represent templates for different data mining problems.

Java Data Mining (JDM) is a standard for developing data mining solutions (Hornick et al., 2006). JDM provides a web service interface which can be used to set up, integrate and manage data mining processes in the context of BPMN and BPEL based environments. BPEL processes are designed which include calls to the JDM web services. These services interact with the JDM API on top of a Data Mining Engine. The authors present a use case on how to integrate data mining based on JDM into a business process based on BPEL in a Bank scenario. However, they do not specify a general concept on how to redesign the business processes in detail and which requirements exist for the modeling of the business process when including data mining.

In (Tsai and Tsai, 2005) a dynamic data mining process system is introduced. The idea is to set up each data mining activity as web service, to model and execute the data mining process in a BPEL environment and to get a PMML compliant model as result. The authors only present a single application as example and do not give a concept on how to model the data mining processes in general.

In (Rupnik and Jaklič, 2009) a methodology for the implementation of data mining into operational business processes is proposed, consisting of the phases Exploratory Data Mining, Deployment of IT into the Business Process, and Operational Data Mining. The first phase consists of performing a CRISP-DM process and evaluating the readiness for an integration of the data mining solution. In the second phase the data mining solution is deployed into the business environment, including changes in design and implementation of the business process and associated applications. The third phase describes the stages of the operational solution, including the CRISP-DM phases except business understanding and an additional phase for model updating. However, the authors do not specify detailed concepts for the modeling of the data mining process as part of the business process. In our work, we focus on how to perform the process (re-)design from the modeling point of view and aim at removing complexity for modeling and integration.

In (Altintas et al., 2005) the authors propose a framework for the reuse of scientific workflows, which are also based on reusable patterns that include abstract tasks. However, their work focuses rather on grid-related tasks like copying, job execution and monitoring than on data mining specific tasks.

3 Process Patterns for Integration and Reuse

We aim at integrating and reusing existing data mining processes that have proven to be successful, and hence want to develop a formal and concrete definition of the steps that are involved in the data mining process and of the steps that are necessary to reuse it in new business processes. Thus, we focus on the reuse and integration rather than on the data mining problem itself and consider a solution for the data mining problem to be available. In the area of data mining there exist a lot of situations and scenarios where existing solutions are reusable, especially when no research on new algorithms is necessary. These solutions can have different levels of generality. Lots of examples and ready-to-use algorithms are available in toolboxes, which only have to be integrated. However, due to a lack of formal support in practice a reuse and integration of existing solutions is not often or only informally done, which leads to a lot of unnecessary repetitive work. By our approach we want to facilitate the integration and reuse of these data mining solutions.

In the following, we will describe our approach in 4 steps. First, we will show how to modify the CRISP model to focus on the special case of reuse of existing solutions. Second, we will define the concept of data mining patterns. These patterns are created after the initial data mining process ended and include the definition, description and requirements of the data mining process, but are independent of the application scenario. The goal of these data mining patterns is to provide a flexible representation for different levels of generality. Third, we will describe how these patterns are applied for reuse by a meta process. This process will be started if a new application scenario is available and describes the tasks that can only be performed with
knowledge on the application scenario. Finally, we will show how data mining patterns can be created from existing data mining solutions.

3.1 Reuse and the CRISP model

In the following, we will present how the CRISP phases and tasks from (Shearer, 2000) differ in the case of executing CRISP from scratch and reusing existing solutions. Fig. 2 visualizes our approach.

Business understanding phase. The task *Determine Business Objectives* is a general task that is independent of the data mining. We arrange this task at the start of the meta-process, as it provides the information needed for the choice of the pattern. The task *Assess Situation* involves the setup of an inventory of resources, a collection of requirements, assumptions etc. In our scenario, this task does not apply as the important information is already available through the business process. In addition, the data mining goal is already specified by the data mining pattern. Thus, we transform the *Determine Data Mining Goals* task into a task that checks if the data mining goal is still matching and arrange it at the beginning of the data mining pattern. In the deployed executable process, this task will be obsolete. From here, the data mining specific tasks start. In our scenario the *Produce Project Plan* task is outside of the scope, as the project plan consists of following the proposed approach for the integration.

Data understanding phase. The tasks *Collect Initial Data*, *Describe Data* and *Explore Data* are obsolete as we assume the data to be available through the modeled business process. The verification of the data quality is arranged as second task *Verify Data Quality* in the pattern. In the deployed executable process, this task will be obsolete as well.

Data Preparation phase. The *Select Data* task is a preprocessing task of the pattern. E.g., a *Collect Data* task and a *Collect Label* task (in case of supervised learning methods) or a *Data Available* gateway could be part of the BPMN pool that contains the tasks that have to be connected to the tasks of the business process. The further tasks *Clean Data*, *Construct Data*, *Integrate Data* and *Format Data* are also preprocessing task of the pattern.

Modeling phase. The tasks *Select Modeling Technique* and *Generate Test Design* do not apply, as this information is already contained in the pattern. The *Build Model* task represents the training of the data mining model and specifies the modeling technique and is part of the pattern. The *Assess Model* task is split into the task *Testing* and the gateway *Model Finalized* inside the pattern. They specify the test design and the decision on when to deploy a model.

Evaluation phase. The task *Evaluate Results* involves a matching with the business objectives. Thus, we arrange this task at the meta-process. With this task, the data mining specific part ends. In contrast to CRISP, the evaluation takes place after the deployment of the process. The task *Review Process* is implicit contained in loops of the meta-process (changing the specification of tasks of a data mining pattern or choosing another pattern). The task *Determine Next Steps* does not apply as the next steps are defined by the meta-process.

Deployment phase. The planning of the deployment by the task *Plan Deployment* does not apply, as the way how to deploy the model and the process as a whole are defined by the pattern approach. The deployment itself is two-fold. On the one hand, there is the deployment of the data mining pattern into the business process. This task is arranged in
the meta-process. On the other hand, there is the deployment of the data mining model, which is part of the data mining pattern. The task Plan Monitoring and Maintenance does not apply as well, as monitoring and maintenance is a general part of the BPM context. The tasks Produce Final Report and Review Project are outside of the scope, as we are not interested in such a kind of deployment and as a review of the steps of the overall process are considered to be already part of the BPM solution.

**Additional Tasks.** In addition to the tasks of the CRISP phases there are additional tasks necessary for the deployment, management and usage of the data mining model. Typically, business processes are not designed for ad-hoc but for long term usage. Thus, the data mining has to be able to adapt to changes in the data during time, e.g. by re-building and deploying the data mining model periodically. For re-building a model, a Re-build model gateway activity is arranged before the tasks for preprocessing and building the model. The task DeployModel is performed after the model evaluation task. In addition, a Model apply task for applying the model to new datasets and related additional preprocessing task are part of the data mining pattern.

### 3.2 Data Mining Patterns

In the following, we will present our approach for the specification of data mining patterns at different levels of generality. The CRISP-DM methodology includes a four-level breakdown, which describes the instantiation of the CRISP process model in order to get a CRISP process. The 6 CRISP phases consist of several **generic tasks** which cover all possible data mining applications. Out of these tasks **specialized tasks** are described which specify specific actions for a certain situation. Finally, the **process instances** represent a record of actions and decisions of an actual data mining engagement. This top-down approach is a manual process which is not automated. In our approach, the CRISP breakdown is partially pre-defined by the process patterns, where some of the tasks are already defined on a detailed level.

The data mining tasks, including requirements and preconditions, can be modeled in a uniform way in the context of BPM by representing each task as a task in the business process. Encoding the requirements and preconditions allows a faster identification of problems with respect to the data mining tasks, e.g. in case the business process or the data changes, and thus enables a faster adaption of the integrated solution. However, we need to take into account that reuse may in some cases only be possible at a general or conceptual level. In addition, the result of a CRISP process in form of a successful data mining solution needs to be abstracted in order to reuse it. Such an abstraction can be done at different levels. E.g., on the one hand tasks like checking pre-requirements could only be formalized as manual tasks, but on the other hand there could be a detailed data mining workflow available where only some parameters for the modelling need to be specified, which is already formalized and only needs to be a little adapted. Thus, our approach needs to include the description of the general CRISP model, of executable workflows, and of abstractions in-between. To do so, we model the tasks different levels of granularity, resulting in the following hierarchy of tasks:

- **conceptual**: manual tasks, only a textual description is available
- **structural**: manual tasks, partially formalized description that pre-structures the task, e.g. pseudo-code or sub-patterns
- **configurable**: semi-automatic tasks, code is available but parameters need to be specified
- **executable**: automatic tasks, code and parameters are specified

Process patterns are only valid for process execution if they are specified to the executable level. Some general CRISP tasks, e.g. Determine Business Objectives, cannot be specified as technical step. Thus, the executable specification for such tasks would be a check task or empty/obsolete task. In a meta-process, which describes how to apply a given pattern to a business process, these tasks are specified (with different choices) according to the levels of the hierarchy. E.g., a Clean Data task could be specified as human task, as code that deletes records with missing values or as separate DM process for the prediction of missing values. Details on the meta-process will be given later in Section 3.3.

The following list shows to which kinds of tasks the generic CRISP tasks are transformed:

- Check tasks in the pattern, e.g. checking if the data quality is acceptable.
- Configurable tasks in the pattern, e.g. setting a certain service parameter by hand.
- Executable tasks or gateways in the pattern which can be executed without further specification.
- Tasks in the meta process that are independent of a certain pattern, e.g. checking if the business objective of the original data mining process and the new process are identical.
- Empty/Null task as the task is obsolete due to the pattern approach, e.g. producing a final report.

**Definition:** We consider the phases and tasks of the modified CRISP process from 3.1, including additional tasks and information that are necessary for an execution of such a process, as the most general data mining pattern. Every concretion of this process for an application according to the presented hierarchy is also a data mining pattern.
3.3 The Meta-Process

The process of applying a pattern for the integration of data mining into a given business process can be modeled as a process in the BPM context. In our approach, this meta-process describes the steps needed to use a pattern for a given business process. Fig. 4 visualizes the meta-process and its steps. The first task is to define the business objectives of the application. After that, a data mining pattern is selected that matches with these business objectives. Then, the tasks of the selected pattern are specified to the executable level according to the hierarchy of our approach proposed or a given specification is chosen. If it is observed that the pattern cannot be specified as executable, the meta-process steps back to the task of choosing a new pattern. If the pattern is executable, it is deployed into the business process. After that, the integrated process is executed. If the result is satisfying, the meta-process is finished. If it is not satisfying, the meta-process steps back to the task of finding a new specification.

The process of applying a pattern has a set of data mining patterns and a business process including the information about the application as input and an integrated process as output. The steps of the process still include a lot of manual work. However, tool-based support for applying patterns would be possible and might be implemented in the future. Such tools would guide the user through the steps of selecting patterns and specifying their tasks, helping to handle the different abstraction levels of the task hierarchy and to interface with data mining tools. The hierarchy of tasks from Sec. 3.2 is defined with respect to the different tools that are necessary to perform the tasks, e.g., a process enactor for executable tasks, a user interface for configurable tasks, a process modeling tool for structural tasks and a text viewer for conceptual tasks.

3.4 From CRISP to patterns

In the following, we will show the general approach of how to create a pattern based on the CRISP process from a data mining paper. Such papers consist of a lot of information on requirements, approaches, related work, literature, examples, configurations, pseudo-code, results, summaries, etc. Within this, typically a lot of information is included which can be used for creating a data mining pattern. Information that is not useful for the creation of a pattern has to be ignored. This includes, e.g., information on related work, literature and examples, as this information is related to other patterns. In addition, summaries and other redundancies have to be left out, as the patterns follow a more formal, structured approach. Experiments and results are also not important for creating a pattern, as the pattern is only focused on the process of the data mining solution. The remaining parts of the paper, which hold the information useful for the pattern, can be transformed into tasks of the data mining pattern depending on their relation to the CRISP phases as well as how precise they are described with respect to the pattern. Hence, the structure of a data mining pattern is based on the
information contained in the paper and on the level of abstraction in which it is presented.

An example of the presented approach will be given in Sec. 4.1. The approach of creating a pattern out of information contained in a data mining paper can be summarized as follows:

- Remove information on related work, literature, examples, results, summaries and other redundancies
- Transform descriptions of tasks and requirements as regards content into generic tasks or gateways
- Transform detailed descriptions of tasks and requirements, instructions and configurations into configurable or executable tasks or gateways
- Transform code and pseudo code into specialized or executable tasks or gateways
- Use figures and use-case diagrams for the arrangement of tasks, e.g. lanes, pools and groups of tasks.

4 Case Study

In (iWebCare Project, 2006) a solution for fraud detection based on similarity learning (Rüping et al., 2008) is presented. We choose this scenario as case study, as it already includes well specified business processes as well as a data mining solution that allows for reuse. The approach for fraud detection from (Rüping et al., 2008) is based on learning similarity measures for data records and is transferable for a generic class of fraud opportunities. The application scenario is based on detecting procurement fraud, e.g. an employee of a company placing an order to another company which is owned by himself. This is done by computing a similarity between employees and company owners based on several features such as name, address or bank accounts. We do not go into more details of the data mining method, as this is not important for understanding our pattern approach. Basically what is needed to apply this data mining solution to a problem is to first check if the problem is a procurement fraud problem, second to specify which features to be used for the similarity, and third to connect the inputs and outputs of the data mining process. For all other steps ready-to-use code is already available.

4.1 Creating a pattern

To demonstrate the approach of creating a pattern from Sec. 3.4, we take the paper (Rüping et al., 2008) as example. Leaving out information like related work, literature, examples, results, summaries and other redundancies, the paper includes most of the information needed for the creation of the pattern, e.g. several pre-requisites, data requirements, process steps, configuration information and pseudo-code. The second and third paragraph of Section 1 of the paper covers the pre-requisites that the problem addressed is procurement fraud and that it is assumed that there is no large training dataset available but user feedback is used for constituting a relevant similarity of employees and creditors. Section 2.1 describes requirements on the data, e.g. the attributes of the payroll and the creditor data. In Section 2.2, the steps of the process that involve the user feedback are described. Section 3 addresses a configuration issue on which of several similarity measures to involve in the computation of the similarity function. Section 4 presents pseudo-code on the algorithm that optimizes the overall distance measure. In addition, instructions on how to transfer a model to another dataset are given. Finally, Section 7 contains information on how to extend the framework by more similarity measures.

All this information can be used for the creation of the data mining pattern. Fig. 5 gives an example on how some parts of the paper can be mapped to the generic CRISP pattern from Fig. 3. The first paragraph can be mapped to the pre-requisite tasks of the pattern (1), which are evaluated first when integrating the pattern into a business process. These tasks cannot be specified more detailed as it is done in the paper. Thus, the tasks become tasks at the generic conceptual level in the pattern. The second paragraph does not hold any useful information for the pattern. The third paragraph describes a requirement to the data and thus can be mapped to the generic task of verifying the data quality (2). The figure presents the interaction with the user in terms of manually analyzing similarities and thus can be mapped to task of sending labeled data to the model building part of the pattern (3). The paper also contains pseudo-code. As experiments are provided as well, it can be assumed that there is existing runnable code available. Thus, in case the code is accessible, this kind of information can be transformed into a tasks at the executable level in the pattern.
Figure 5  An example on how to map the information of a paper to the generic CRISP pattern.

Figure 6  An example of a data mining pattern for procurement fraud.
Fig. 6 shows an example of a pattern for the presented approach to procurement fraud detection. In the top pool Requirements the pre-requirements for applying this pattern are modeled. This includes checking the data mining goal (procurement fraud detection) and the data format as well as sending data, receiving the result and sending labeled data. The other pools Data Mining Classification, Data Mining Model Building and Data Mining contain the (partially already specified executable) tasks of the data mining process and the respective services. It can be seen that the process is a data mining pattern according to our definition from Sec. 3, which contains tasks of the different levels of the task hierarchy. E.g., the task Check DM goal is supervised Fraud Detection is a specialized task of Check DM goal. It is a conceptual task which describes in textual form that the goal of the integration of the data mining solution has to be supervised fraud detection in order to match with the pattern. During the integration, the user manually inspects this task and specializes it to the empty task. The task Check if each attribute is based on a known data type represents the requirement for a specific data format. The data mining solution is based on combining different similarity measures for attributes of different data types. For each of those types, a measure has to exist. The data is sent to the data mining part of the process via the task Send supplier and payroll data and the labels via Send labeled Data. The assignment of the similarity measures is performed at the task Assign Similarity Measure. It is a configurable specialization of the task Train Model that can be further specialized by the user, e.g. by a configuration file. All executable tasks, e.g. the task Train Similarity, are connected to an underlying service.

4.2 Integration and Reuse

In (iWebCare Project, 2006) (Section 6) a set of business processes from the health care domain is presented which contains candidates for the integration of the fraud detection solution. We will focus on the business process Purchase Order Inspection from the RBH scenario, which consists of a random checking of several rules for a request for an order followed by the decision of placing or not placing this order. According to the business objectives of the business process, the data mining pattern for procurement fraud detection presented in Fig. 6 is a candidate for the integration. Following the Meta-Process defined in Section 3.3, the tasks of the data mining pattern have to be specified to the executable level and the pattern has to be deployed into the business process. Fig. 7 shows the result of the integration of the presented data mining pattern into the business process. Compared to the pattern, the task Check DM goal disappeared because the user took the decision that the data mining goal indeed matched and hence specialized it to an empty task. The tasks for sending data and receiving the results are connected with the tasks of the business process. The tasks Check if each attribute is based on a known data type and Assign Similarity Measure disappeared as well, as the user manually specified the checking and the assignment of the similarities according to his knowledge of the data (e.g. by a parameter file). During the integration, the pool Requirements and its tasks disappeared at all due to the specification and deployment tasks of the meta-process.

By this somewhat simplified example we have shown how to apply a data mining pattern to a business process in order to get an integrated process and how manual steps can be specified to executable tasks.

In addition, the reusability can be demonstrated by our example. In the Find new supplier process in (iWebCare Project, 2006) (Section 6), in which new suppliers are to be found for certain purchase order requests, a similar problem has to be solved. As the data mining solution is generic, it can be reused and integrated in both process examples easily (see Fig. 8).

5 Conclusion and Future Work

This paper is a first step towards support for an automation of a reuse of successful data mining processes. We have presented a new approach for easy reuse and integration of data mining in different business processes. Our approach is based on CRISP and includes the definition of data mining patterns, a definition of a hierarchy of tasks to guide the specialization of abstract patterns to concrete processes, and a meta-process for applying patterns to business processes. These data mining patterns allow for representing the reusable parts of a data mining process at different levels of generalization and provide a simple formal description for the reuse and integration of data mining. We evaluated our approach in a fraud detection case study in the health care domain.

Our approach is focused on reusable cases for data mining problems. Thus, in contrast to the more general CRISP model, the data mining patterns are not applicable in all cases, as several parts of the process are already pre-defined in the pattern. However, by the pre-definition a lot of unnecessary repetitive work is avoided. E.g., if data mining is to be integrated in the context of a very similar problem, the pattern can be just used again instead of applying the full CRISP-DM process twice. CRISP is iterative and does not describe the reuse of existing solutions at all. In our approach, we describe the reusable part by data mining patterns. The iterations are modeled in an outside meta-process.

Details on the definition, architecture and implementation of the underlying services remain to future work. Furthermore, we aim at a formalization that allows for tool support for the design and specification of high-level requirements as well as for the parameterization. In addition, the following questions need to be addressed in the future: How to model the data within the process? Key challenges are to define the
Figure 7  Integration of the procurement fraud pattern into the business process *place purchase order*.

Figure 8  Integration of the procurement fraud pattern into the business process *find new supplier*.
data-related interfaces between the business processes and the data mining patterns as well as to specify how to get the data out of the business process. Is the hierarchy adequate or is a more detailed level needed? E.g., more structure could be provided by the use of an ontology (Hilario et al., 2009). What is a good pattern, how can the quality of a pattern be determined and how to select a pattern from a pattern database?

References


