J. Basic. Appl. Sci. Res., **2(10)10384-10395, 2012**

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Convergence of Data Mining and Process Management for Operational Inteligence

Ramzan Talib, Ghulam Rasool, Muhammad Umair

Department of Computer Science,
COMSATS Institute of Information Technology,
Lahore, Pakistan

ABSTRACT

Convergence of data mining and process management is ideal – but still limited. Data mining techniques helps in actionable knowledge discovery but lack for context awareness whereas process management systems support context awareness but lack for operational intelligence. To make process management systems operational intelligent, data mining techniques can be integrated within them in removing different inefficiencies. This paper presents an example of such a convergence in resolving one of the inefficiency relating to its resource management specifically to its static agent assignment strategies. To highlight the potentials of this convergence, an exemplary use case from textile industry is presented and discussed in depth along with experiments and experiences from textile industry.

KEYWORDS- Process Management; WfMS; data mining; performance evaluation; goal model; continuous resource management; context awareness

I. INTRODUCTION

Data mining techniques, algorithms and methods have long stood alone—they are consuming data stemming from various applications, analyze them, and provide rules and models supported by that data. But simply utilizing application data within data mining methods silos misses part of picture because they lack the context of business process in which data was created [1]. Whereas process management systems automate processes and manage the flow of work between workflow participants (resources i.e. human agents). A workflow (or process model) defines process steps, their order, under which conditions and when they will be carried out, by whom (resources i.e. human agents) within an organization, with which tools (i.e. applications), and define the flow of data within these process steps [2].

In most circumstances, business process determines the context for data eventually used in data mining. Context is very important in business decision making and when data is taken out of context, the result are, at least, limited, if not downright misleading [1]. Therefore, by combining data mining and process management technologies, organization can leverage from context relevant information in mining methods for producing more concise knowledge. This knowledge can be of utmost importance—not only for decision makers in order to increase business benefit—but also for automated systems like workflow management system (WfMS) to become operational intelligent.

Operational intelligence is a form of real-time dynamic, business analytics that delivers visibility, insight into business process and delvers actionable information. The purpose of operational intelligence is to monitor business process and activities to detect situations relating to inefficiencies [3]. For instance, data mining techniques can be integrated within workflow management system to remove inefficiency related to workflow resource management.

This paper particularly focuses on how these two technologies collaborate in producing more concise and actionable knowledge especially when decisions are to be made by process execution engine on its own to achieve operational intelligence. Specifically it demonstrates how process management technology provides context awareness which is useful for data mining and how data mining techniques employ context awareness from process management to produce more concise and actionable knowledge again useful for process management by focusing one of the crucial functions of WfMS concerning to resource management.

One of the crucial functions of a WfMS is assigning tasks (processes) to users (workflow resources: employee or human agents) in order to execute them. A well identified open issue in workflow resource management literature [4][5][6], is "how to allocate agents to processes on the basis of history of workflow executions": [**Pattern 9: R-HBA]**. As a matter of fact, history of workflow execution is neither analytically evaluated nor incorporated into future assignment strategies. As a result, agents simply continue with their processes even if some of them have an awful "success history" [7][8]. This situation is unsatisfactory as it has a negative impact on the overall business performance [9]. It also contradicts the philosophy of continuous resource management [10] – only successful agents remain enacting processes.

***Corresponding Author**: Muhammad Umair, Department of Computer Science, COMSATS Institute of Information Technology, Lahore, Pakistan. mumair@ciitlahore.edu.pk

Principally, WfMS defines *assignment policies* to allocate eligible agents to processes, mostly expressed in terms of roles. Roles are defined during process design time on the basis of similar capabilities and skills of individuals within an organization. For example a **Cutting Process** of a garment industry is assigned to ***Cutters*** (role; a group of users capable to cut the fabric for specific garment). During the execution of a process all of the eligible agents are selected that fit into the role assigned to the process step in execution and are informed about the task to perform. Any eligible agent, not only the efficient one, can select the process and execute it.

Currently in almost all WfMS those agents that are found to be eligible to enact a certain process are thought of to be "successful". However, at present WfMS do not have any mechanism to determine "*how successful*" certain processes are actually being performed by these agents. As a consequence, agents go on and on with enacting processes even if they have an awful success history and therefore drag the overall performance of the whole process down [10]. Also, due to this consequence, business process management becomes ineffective. It would be great if a WfMS could distinguish "successful" from "unsuccessful" agents utilizing the workflow history and then leverage this knowledge for future assignments – an ultimate demand of competitive business environments.

In competitive business environment (e.g. garment production process), not only agents are assigned to tasks but also clear descriptions of what goals they should achieve. A goal is a measurable target that an organization sets up to be achieved by authorized agents [11]. It defines "*what is success*" and "*how to measure it*". For example, consider a goal like "*cutting fault should be less than 2%*" that defines a "*success criteria*" for **Cutters** against which they are evaluated. But, in contrast to competitive business environments, WfMS traditionally lack to model measurable goals in business process models and because of that do not support evaluating the performance of certain agents [12].

In this paper we will demonstrate when goals are define within the process model they provide context awareness for data mining techniques. We will also overview Agent Performance Evaluation (APE) Framework to demonstrates how data mining techniques employ context awareness from goal model to analytically analyze "who is performing a certain process how well and under what certain conditions" towards achieving operational intelligence within workflow management system.

This paper is divided into seven sections. After this introduction, Section 2 provides a general overview of a WfMS and its different mechanisms for agent assignment strategies. In Section 3 we discuss related work. Section 4 explains how process model incorporates context awareness for actionable knowledge discovery. In Section 5, we present the architecture and execution semantics of APE Framework. Details of our experimental results and experiences are given in Section 6. Finally, Section 7 concludes this paper and provides an outlook on future work.

II. WORKFLOW MANAGEMENT

A. Overview

The WfMS automates processes according to a *process model* which, according to [2], consists of five major perspectives. The **Functional** perspective defines the skeleton of the process. It identifies process steps and their purpose. A process step can either be atomic or it can be composite and serves as a container to constitute a process hierarchy. The **Data** perspective defines input and output data of a process. It defines flow of data between processes and relates it to external data models. The **Operational** perspective describes tools, (programs, systems, etc.) that are required for the execution of a process. The **Behavioral** perspective determines the control flow, i.e. the order in which the single steps of a process are being scheduled for execution by a WfMS. Last but not least the **Organizational** perspective determines agents who are eligible to perform a certain process in order to achieve the business **goal**.

Figure 1 shows the process model of a garment production process modeled in i>PM process modeling environment [13] where process steps (e.g. Market Making, Cutting, etc.) are depicted as a rectangle; the **Functional** perspective is represented within that rectangle in a text box. Small text at the lower left corner of the process step represents the **Organizational** perspective; here the role e.g. "Cutters" was assigned to the process step **Cutting**. The **Goal** construct is described by the small text at the lower right corner ("Cutters Goal"). **Data and Data Flow** are described by small boxes (data) that are placed on the black arrows (data flow) which connect two steps of a process; a data flow arrow always starts at the producer side of a data item and ends at the consumer side. The execution order of a process is, when this is not specified by data dependencies, defined with the help of the **Behavioral** perspective represented by grey arrows. A text just above the upper left corner of a process step denotes information about the **Operational** perspective.

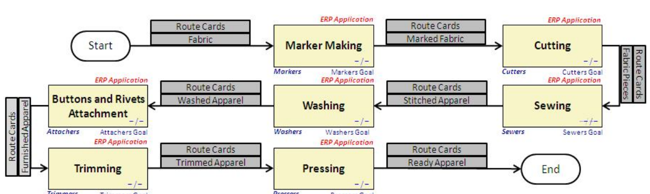

Figure 1: Production process of an apparel division

Process models are executed in a process execution infrastructure called ProcessNavigator [14]. Its ultimate objective is to deliver the right data at the right time with the right tool to the right people (the people, meeting organizational goals). ProcessNavigator also maintains a process log (in a relational database). This process log – when integrated with the process application data – provides an 'information source' for APE Framework.

B. Overview of Agent Assignment Strategies

Real world enterprises demand different ways or manners for the assignment of their processes to employees. A process can be assigned to an agent directly (e.g. Amir) or on the basis of a role (e.g. Cutters), a group (e.g. agents working on same project) or an organizational relation (e.g. *ManagerOf(Ali)* can ***Sign*** Ali's leave). Additional constraints like *delegation* (e.g. if a user is not available, process should be allocated to an alternate user to avoid excessive delay), *binding of duties* (e.g. customer complaint should be handled by a person who sold the product) and *separation of duties* (e.g. user should not approve his own bill) are also required.

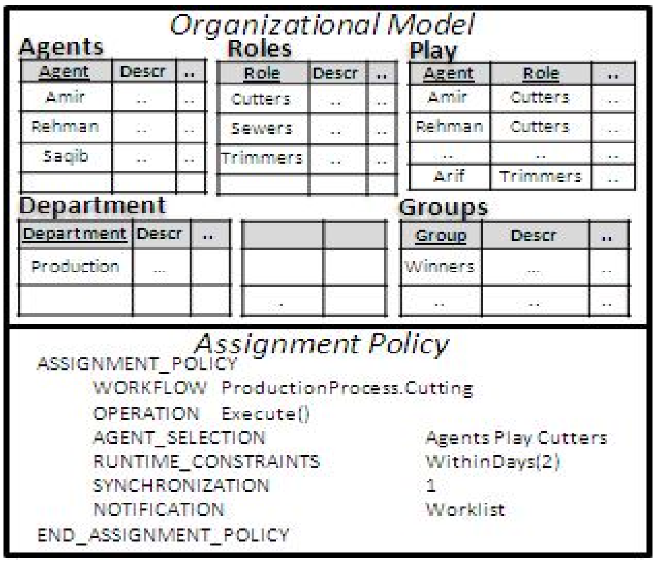

Figure 2: Organizational Model and Assignment Policy

WfMS defines and utilizes *organizational model* and *assignment policies* in order to support different ways of agent assignments. An organizational model (cf. upper part of Figure 2) describes the *organizational structure* and its *population*. The organizational structure defines all objects (e.g. *agent, role, group, department*, etc.) and relationships types (e.g. *plays*) that are required to model the organization whereas the organizational population is a concrete instantiation of individual entities and relationships (e.g. 'Cutters' and 'Sewers' are instances of a 'role'). The assignment policies (cf. lower part of Figure 2) are defined to specify all eligible agents by giving *criteria* that a user must meet. The ProcessNavigator determines a specific agent for the execution of a process basing on the organizational model and the policies which are automatically evaluated against the model.

III. RELATED WORK

This section details the state of the art of WfMS resource management in terms of organizational models and assignment strategies, they offer for eligible agent selection.

In [10], [15], [16] and [17] organizational models are proposed that define organizational structures as entities with relationships among them and organizational population as instantiation of the organizational structure in terms of users and the roles they play or the groups they belong to. In [15], Bussler developed an abstract meta-model for specifying an organizational structure. It is more general than [10], [16] or [17] since it does impose any restriction on the organization being modeled ([10], [16] and [17] presume limited organizational entities and relationships).

WfMS traditionally use roles as means for abstracting from the assignment of a concrete agent and his or her skills; but, as [15], [16], [17] and [18] point out, the role concept alone is not sufficient to cope with the requirements of task assignment of real world workflow applications. Some examples, where this can be easily seen, were presented by Bussler in [15]: assignment of a task to an agent that is somehow related to another agent (e.g. ManagerOf(clerk)), the same agent as the previous task, etc. As a solution, Bussler present a Policy Language and Policy Resolution Framework for specification and execution of generic task assignments in WfMS.

The work of Bussler was expanded by HP-Lab [16] and Cao et al. [17]. HP-Lab proposed an SQL-like policy language called Resource Query Language (RQL) [16] that is able to specify three types of policies: *requirement*, *qualification* and *substitution* policies. These policies deal with organizational constraints like *role delegation* (e.g. if a user is not available, WfMS should be able to locate alternate users to avoid excessive delay), *binding of roles* (e.g. customer complaint should be handled by a person who sold the product) and *separation of duties* (e.g. user should not approve his own bill). This work is further extended by Cao et al. [17] in which they define the Task Assignment Policy Language (TAPL) for handling similar constraints (*role delegation*, *role binding* and *separation of duties*). TAPL adds some new features with respect to RQL such as WHERE, WHEN and WITH clauses in *require*, *substitute* and *reject* policies to represent complicate resource allocation conditions.

Moreover WfMSs like Staffware [19], WebSphere [20], FLOWer [21] or COSA [22], offer mechanisms to specify additional constraints like *delegation*, *binding of duties* and *separation of duties*. The downside of all these approaches is a missing support for enforcing the assignment of "only successful" agents to processes.

Also, Kumar et al. pointed out that role based assignments are critical when suitable workers are not available [18]. It becomes crucial especially when a process needs to be executed before reaching a certain deadline. Processes then have to be assigned to a lesser qualified staff. They presented an assignment methodology along with a simulation model that dynamically creates a balance between a qualified agent and accomplishment of work. In comparison to our work their methodology do not incorporate the agent success history because they used static metrics that are defined only once during design time.

State of the art workflow resource allocation patterns are presented in [4], [18]. These patterns capture different ways in which resources are utilized to perform a task. It is pointed out that existing modeling notations such as Business Process Management Notations (BPMN) and Unified Modeling Language (UML) 2.0 Activity Diagrams do not support Pattern 9: R-HBA [5]. Also, existing WfMSs such as Staffware Process Suite version 9 [19] (TIBCO), WebSphere MQ Workflow 3.4 [20] (IBM), FLOWer 3 [21] (Pallas Athena) and COSA 4.2 [22] (TRANSFLOW) do not directly support history based resource allocation ([4], [5] and [6]) i.e. neither the history of 'business success' is evaluated nor it is used for future task assignments. For example, it would be very beneficial to allocate a ***heart bypass task*** to the surgeon who has 'successfully' completed such tasks over the past three months.

Only Oracle BPEL Process Manager V.10.1.2 [23] and iPlanet 6.0 [24] (SUN) partially support Pattern 9: R-HBA. They evaluate history just on the basis of 'more executions **-** more experience' and/or 'quick execution time' without focusing on success of business process performance. For example, the ***heart bypass task*** should not need to be assigned to a surgeon who is able to complete this surgery fastest – instead, it should be assigned to a surgeon who will most likely complete the surgery successfully.

IV. PROCESS MODEL INCORPORATES CONTEXT AWARENESS

Process management technology defines the "context awareness" of business processes within the process models. Since, it precisely defines syntax and semantics for different perspectives of processes like which data being utilized, what are the eligible agents who execute them, when and in which order processes are executed, and applications required to execute them etc. Therefore, this context awareness explicitly defined for each process can be utilized to support domain relevant information in data mining methods for delivering promising knowledge or actions.

According to [12] goal definition consists of two parts: measurement definition and context definition. Measurement definition includes goal name, goal description, goal metric (data used to measure the goal), data source (e.g. FaultTable), data type, and a goal query which is needed for retrieving required data. It also includes the specification of different groups. Each group is specified in terms of group name, start value, end value, and priority as shown in Figure 3 (lower part). These groups are used to describe:

- What are different success levels?
- How to identify a certain success level using goal metric values?
- How to rank the superiority of different success levels i.e. which group is better than others?

Based on the current goal metric value, a performance evaluation mechanism determines a particular group. For example, when the goal metric value (FabricFaultPercentage) is less than 2 (%) this corresponds with the group "Good". Next, for this identified group (i.e. "Good"), a performance evaluation mechanism determines its "priority" utilizing its concern group specification. The priority of that particular group determine the rank of its 'success' among different success levels – the higher the value of the group priority, the better the level of its business success.

These priorities are essential because performance evaluation mechanism cannot rank different levels of goal achievement simply from interpreting the words like "Good", "Average" or "Poor". Also, these priorities guide performance evaluation mechanisms to perform certain actions that are required for continuous resource management. For example, it could revoke the authorization of a person who achieved a goal having lowest priority (e.g. -100 for 'Poor' group). On the whole these priorities are used by performance evaluation mechanism to determine the superiority of certain 'success level' so that it can perform corresponding action that is ultimately required for continuous resource management.

Measurement definition not only helps to measure goal metric but also supports context awareness for data mining techniques. For example, it describes many data items that help in performing the following preprocessing and post-processing tasks normally used within the knowledge discovery phases:

- Data Extraction: Measurement definition describes data items namely “Data Source” and “Goal Query” that in fact help for data extraction because this SQL statement is used to extract the required dataset used within mining methods.
- Feature Selection: The “Goal Metric” defined in the measurement definition helps to identify data element used for goal computation. Data elements that are specified in the SELECT Clause of a goal query actually helps in retrieving only those data elements which are very important for goal computation rather than selecting all data of a process even some of the data elements are not useful for goal computation.
- Data Discretization: Each instance of a workflow execution history extracted using the goal query has numeric value for the data element “Goal Metric” (e.g. FabricFaltPercentage). Before applying data mining algorithm for analysis, it needs to be discretized into nominal values of different success levels (i.e. Good, Average and Poor). Group formulation within the Measurement Definition helps to discretize “Goal Metric” values into nominal groups belonging to different success levels. Depending upon the goal metric value, a specific success level is selected which lies within the boundaries of Start Value and End Value of a particular group definition. For example, when the goal metric value (e.g. FabricFaultPercentage) is less than 2 (%) this corresponds with the cutters’ goal group “Good”.
- Ranking: Data element “Priority” defined within the Group Formulation part of goal measurement definition basically helps in determining the superiority of a specific success level among different success levels. It is hard to decide which success level is superior to another: neither on the basis of their nominal values nor on the smaller/greater value of a Goal Metric. Thus, the higher the value of the success level priority, the better the level of its superiority within the application domain.
- Grouping: On the basis of a specific value of a goal metric, Group Formulation criteria defined within goal definition helps to identify names of different success levels.
- Refinement/Tuning of Mining Algorithm: On the basis of a number of instances extracted through the Goal Query along with the statistics of individual data elements of a goal query help for further refinement and adjustment of the parameters (i.e. setting the support and confidence values) of mining algorithms.

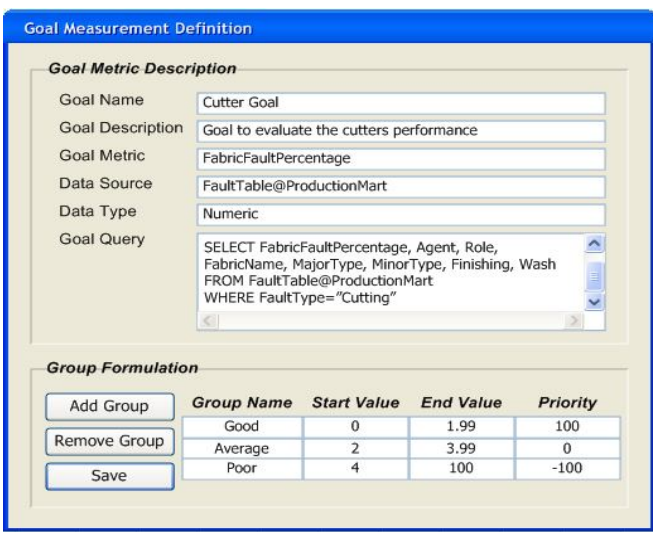

Figure 3: Goal Modeling.

A goal context definition describes the influencing factors of a process in terms of Organizational Definition and Data Definition. Within this scenario, context definition supports domain knowledge for two purposes: feedback and attribute removal.

Organizational Element within the Organizational Definition specifies a particular table of the organizational database where agent competency profiles are defined. For example, Organizational Element **Play**, defined in the context definition, helps to identify the particular table of the organizational database. Also, Organizational Attribute defines the list of those attributes that comprises the primary key of the Organizational Element (i.e. **Play**). These attributes help in uniquely determining particular instances in the table and are ultimately required in locating and updating agent competency profiles.

Similarly, Data Definition is concerned with specifying those data elements that may influence the achievement of a specific goal and are included in the select clause of a goal query e.g. FabricName, MajorType, MinorType, Fining, Wash. This definition helps data mining techniques to perform some preprocessing for the removal of ineffective attributes. Thus, it helps in removing those attributes that are even selected within a goal query but actually are not relevant for the mining algorithm e.g. FabricName is an attribute that has high cardinality and therefore can be removed within the preprocessing stage.

```
ContextDefinition
    OrganizationalDefinition
        OrganizationalElement  Play;
        OrganizationalAttribute Agent;
        OrganizationalAttribute Role;
    EndOrganizationalDefinition
    DataDefinition
        FabricName                              String;
        MajorType                               Nominal;
        MinorType                               Nominal;
        Finishing                               Nominal;
        Wash                   Nominal;
    EndDataDefinition
EndContextDefinition
```

Figure 4: Context Definition

V. AGENT PERFORMANCE EVALUATION FRAMEWORK

Agent Performance Evaluation (APE) Framework [25] aims at enabling WfMSs operationally intelligent so that WfMS can distinguish successful from unsuccessful agents using the execution history of processes. As a consequence, WfMSs can update assignment rules to reflect experiences made in the past by assigning "successful" agents before assigning less successful staff. It applies data mining on information stored in organizational and operational databases to determine successful agents in terms of *agent competency profiles* i.e. quantitative measure of performance that specify who is performing how well (cf. Figure 5, upper part). These *agent competency profiles* are then updated (semi)automatically [25].

The architecture of the APE Framework along its associated components is shown in shown in Figure 5 (lower part). It has five major conceptual architectural abstractions: namely Goal Retrieval, Data Extraction, Analysis, Feedback and Visualization.

Within the **Goal Retrieval** phase, the complete definition of a goal is loaded from a process model utilizing a repository that holds all process models of an organization. Then, data needed for the computation of a goal metric is extracted from the operational database within the **Data Extraction** phase. This phase executes the query contained in the goal definition for retrieving the proper data. Then data are then analyzed within the **Analysis** phase for computing agent competency profiles using the extracted data. During the **Feedback** phase, these competency profiles are updated and written back to the organizational database. Finally, the **Visualization** phase presents the competency profiles and updates to process controllers for double-checking decisions taken by the automated system; also a process controller can influence the parameters of the evaluation algorithms to meet changing requirements (this is not shown in Figure 4).

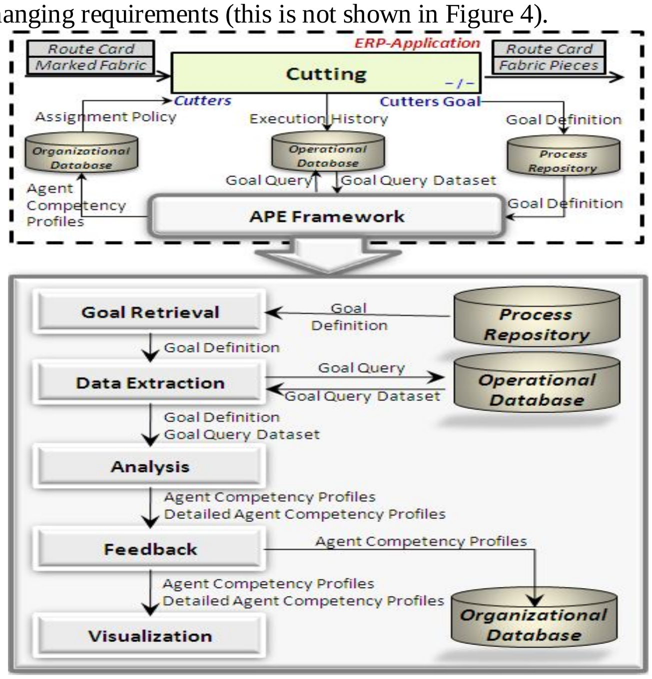

Figure 5 : Agent Performance Evaluation (APE) Framework

Please note that changes to the organizational database (i.e. changes to competency profiles) do not overwrite any data within the database. Thus allows for retrieving a history from the organizational database which can be leveraged in evaluating the success of certain training methods and courses (e.g. did the performance increase after an agent visited a special training or not). However, a WfMS uses usually the latest version of a competency profile only.

The whole analysis process can be either started automatically (e.g. each time a process instance ends or after a certain number of instances finished) or upon a manual request. Latter possibility is favorable since all evaluations performed are highly sensitive with respect to the relationship between employer and employees. Therefore it is pointed out that the whole analysis and evaluation process together with updating the assignment rules within the organizational database to be a process which has to be supervised by one or more human agents (e.g. process controllers) within the organization.

The following paragraphs will now detail the Analysis, Feedback and Visualization phases.

In order to compute the competency of an agent, the **Analysis** phase makes use of pre-tuned mining methods like PAPE [7], CAPE [8] and OAPE. Which method is chosen, again, depends on the interpretation of the context of a process (as given in the goal definition). Context of a process is classified on the basis of the number of consumed data elements into three categories. If a process involves many data elements, the PAPE method is used; if a process involves one data element only, CAPE is applied and if a process does not use any (input) data, OAPE is the method of choice.

However, before a particular algorithm can be started, some preprocessing tasks (e.g. data discretization, filtering, etc.) can become eminent and are therefore executed beforehand. These preprocessing tasks are also driven by context awareness supported within the process model, e.g. discretization of success groups is guided by the goal measurement definition and filtering of relevant data is performed by the goal query that contains the collaborative and agreed-upon-consensus of relevant domain experts. Nevertheless, context awareness also helps in adjusting the parameters of mining algorithm [25].

As stated above, PAPE is selected in case a process consumes many data elements such as the Cutting process. PAPE then consists of two steps, namely Classification and Post-Processing. During the first step, classification techniques (j48 decision tree, cf. [26] from the weka library [27]) are applied in order to arrange data according to the groups defined within the goal definition. The output is a decision tree that tells under what circumstances each group was reached by each agent (cf. upper part of Figure 5).

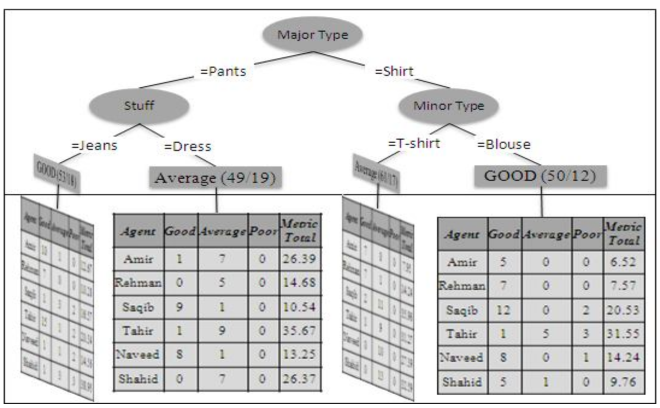

Figure 6: Decision Tree Integrated with Agent Intelligence Matrices

Although the main aim in generating such a decision tree is to find out what factors within the data influence the performance of the process related to the goal that should be reached, these factors are not sufficient. Additionally, this information is linked with the organizational structure in order to find out who performs how well under what conditions. This link is established by the second step of the PAPE method, the post-processing. Basically, application data elements, goal groups, the goal metric and agent data are used in traversing the decision tree from the root down to its leaves where the data elements are the dominant information in traversing the tree. For each data element, the tree is traversed and whenever a leaf is reached, the agent intelligence matrix (cf. lower part of Figure 6) is being updated – the number of times each group has been reached is incremented and the figure MetricTotal is being updated for the particular agent involved in the execution.

The agent intelligence matrix is then used to answer the aforementioned question “who is performing how well under what conditions”. This information is summed up in the agent competency profile that first contains information about the “conditions” and second performance information for each agent ever involved in a process running under these conditions (please note that learned “conditions” is referred as “pattern”). Such a competency profile for a specific condition/pattern is given in the following:

Pattern: (MajorType=Shirt AND MinorType=Blouse)

Agent Competency Profiles:
<Agent, Amir><Goal Metric, 1.30><Good, 5><Average, 0><Poor, 0>
<Agent, Rehman><Goal Metric, 1.1>< Good, 7><Average, 0><Poor, 0>
<Agent, Saqib><Goal Metric, 1.47><Good, 12><Average, 0><Poor, 2>
<Agent, Tahir><Goal Metric, 3.51><Good, 1><Average, 5><Poor, 3>
<Agent, Naveed><Goal Metric, 1.58><Good, 8><Average, 0><Poor, 1>
<Agent, Shahid><Goal Metric, 1.63><Good, 5><Average, 1><Poor, 0>

It can be easily seen from above profile that the agent called Amir is performing well while Tahir's performance is only average if a blouse is being produced. For each pattern from the decision tree, such a profile is computed providing information about who performs best under what conditions.

From the decision tree as depicted in Figure 6 (upper part), it is obvious why there are other methods to determine the performance, for example, when learned tree is empty to emit patterns, there is no or only one application data elements available. In case tree is empty (single node) or there is no data element available for computing different scenarios of agent performance, the OAPE method is used. OAPE learns the agent competency profile by directly computing the goal metric and determining the result by looking up in which group (defined in the goal definition) the value of the metric resides. Again, the competency profile is computed in a similar style as it was done by PAPE – but without any patterns. The following depicts such a competency profiles learned by OAPE method:

<Agent, Amir><Goal Metric, 1.73><Good, 23><Average, 8><Poor, 0>
<Agent, Rehman><Goal Metric, 1.69>< Good, 21><Average, 6><Poor, 0>
<Agent, Saqib><Goal Metric, 1.94><Good, 24><Average, 15><Poor, 4>
<Agent, Tahir><Goal Metric, 2.93><Good, 18><Average, 24><Poor, 5>

Last but not least, the CAPE method is used for computing agent competency profiles in case there is only one application data item defined in the goal context definition. Here, objective is to determine whether agents are performing differently for each context value or not? It is necessary because, each different value of application data element forms different process contexts and as a result agents may also have dissimilar competencies specific to them. Therefore CAPE method learns agent competency profiles following two different notions: first, profiles in-general likewise OAPE method; second, context-specific profiles, agent competency profiles are learned separately, specific to each context value, where context value specify a condition (likewise the condition of pattern as in PAPE method) as following:

In-General Profiles:
<Agent, Amir><Goal Metric, 1.88><Good, 37><Average, 15><Poor, 5>
<Agent, Rehman><Goal Metric, 0.69>< Good, 24><Average, 3><Poor, 0>
<Agent, Saqib><Goal Metric, 3.4><Good, 12><Average, 41><Poor, 3>
<Agent, Tahir><Goal Metric, 2.35><Good, 11><Average, 28><Poor, 5>

Context-Specific Profile: for Context (FiberType = Tencel)
<Agent, Amir><Goal Metric, 1.39><Good, 12><Average, 10><Poor, 2>
<Agent, Rehman><Goal Metric, 1.83>< Good, 7><Average, 1><Poor, 0>
<Agent, Saqib><Goal Metric, 2.94><Good, 3><Average, 19><Poor, 1>
<Agent, Tahir><Goal Metric, 1.54><Good, 7><Average, 2><Poor, 1>

Please note that similar context specific profiles are learned for other context values like Naylon, Dainer, and Romex etc, for example, for Weaving process.

To determine "what are the contexts" where agents are performing dissimilarly i.e. have predominant agent competency profiles, CAPE method uses Euclidean Distance [28] function. It computes intra-agents-competencies distances from different sets of agent profiles, i.e. in-general profiles and all of context-specific profiles (e.g. specific to Tencel, Naylon, Dainer and Romex etc.) separately. Those contexts, whose intra-agents-competencies distance is greater than generic intra-agent-competencies distance, are determined. These are the contexts where agents have predominant expertise, therefore, demands for the adoption of context specific competency profiles into organizational database; otherwise generic agent competency profiles are optimally adopted. For further detail we refer to [8].

During the **Feedback** phase, the agent competency profiles contained in the organizational database are updated. Thus, the WfMS can take decisions basing on the "current" competency of each agent and not on that one which was set up "once" and maybe a long time ago. Of course, assignment policies and organizational database structures need to be adapted in order to be able to apply our methodology; however, we do not consider this as a major obstacle.

The organizational database, besides the normal organizational structure, includes information about the competency of each agent for certain scenarios. For instance, in case there are many factors that influence the performance of a certain agent, these factors need to be shown in the database. In our example, the organizational database contains a table with agents and a table with roles. An additional link table named "Play" identifies, which agent plays what role. In addition to this quite usual structure, our example database was extended to include competency values for each agent depending on the role he or she plays (cf. upper part of Figure 7).

Also, assignment policies need to be changed such that they also include the newly created attributes of the database. This is shown in Figure 7 (lower part) too; the assignment policy depicted in the Figure selects cutters that have a competency lower than 2.0. If this policy is used during the assignment of agents within a WfMS, only those agents are selected to be candidates for executing a process which were most successful in the past (under the same conditions).

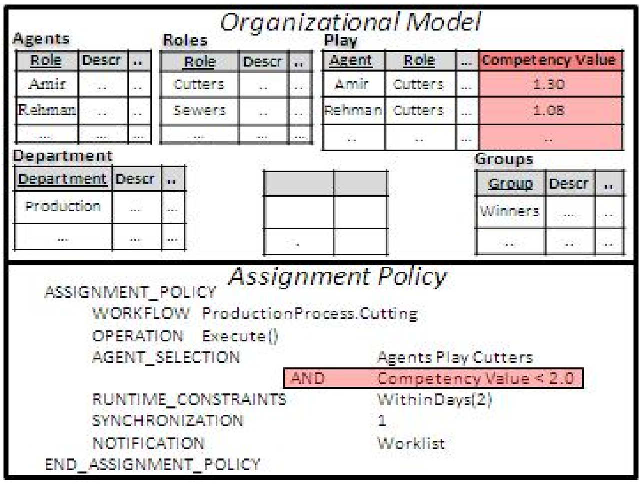

Figure 7: Competency Driven organizational Model and Assignment Policy

The update of the organizational database and the assignment policies for each process can be performed either automatically or semi-automatically.

Automatic updates are possible if the competency profiles are simple; such simple profiles are constructed by the OAPE method. Since there are no data items that influence the competency analysis, only the competency values need to be updated. The assignment policy refers to a specific competency value only and does not need to reflect various factors. An update of the organizational database can then be performed by executing a single SQL statement.

Semi-automatic updates should be performed in case PAPE or CAPE was applied as performance evaluation mechanism. One reason is that updating the assignment policy is not so easy for a system (it might lead to long and hard to comprehend assignment policies). Besides that, also the number of influencing factors is crucial for the size of resulting database; if there are many different factors and if a competency value is to be stored for each factor, the number of columns within exemplary "Play" table explodes.

```
Assignment Policy
ASSIGNMENT_POLICY
    WORKFLOW ProductionProcess.Cutting
    OPERATION Execute()
    AGENT_SELECTION
      IF (MajorType='Shirt' AND MinorType='Blouse')
        THEN {Agent Play BlouseCutters AND Competency Value < 2.0}
      IF (MajorType='Shirt' AND MinorType='T-shirt')
        THEN {Agent Play T-shirtCutters AND Competency Value < 2.0}
      IF (MajorType='Pants' AND Stuf='Dress')
        THEN {Agent Play DressPantsCutters AND Competency Value < 2.0}
      IF (MajorType='Pants' AND MinorType='Jeans')
        THEN Agent Play JeansPantsCutters AND Competency Value < 2.0}
      ELSE {Agent Play Cutters AND Competency Value < 2.0}
    RUNTIME_CONSTRAINTS          WithinDays(2)
    SYNCHRONIZATION              1
    NOTIFICATION                 Worklist
END_ASSIGNMENT_POLICY
```

Figure 8: Assignment Policy Learned by PAPE Method

Therefore, it is more feasible to use the computed competency values for splitting up the "Cutter" role depending on the different factors, e.g. into BlouseCutters, DressPantsCutters, T-shirtCutters and JeansPantsCutters. Then, agents which are "Cutters" but are not good a cutting fabric for blouses, are not selected for such jobs since they are not playing this specific role. Since introducing new hierarchies for existing roles needs major updates which most likely will also have side-effects (e.g. the process models could be updated as well), this kind of change should not be performed automatically – at least not without any supervision. Figure 8 shows an example of an assignment policy which was updated in order to reflect the new hierarchy in the "Cutter" role.

Within the **Visualization** phase, agent competency profiles are presented in different representations such as pie graphs, tables and 3D scattered graphs. Figure 9 shows the overall performance of the cutting process (upper part) and detailed agent competency profiles (lower part) learned using OAPE. Similarly multiple graphs (along with their pattern/context conditions) are displayed for competency profiles learned by using PAPE or CAPE.

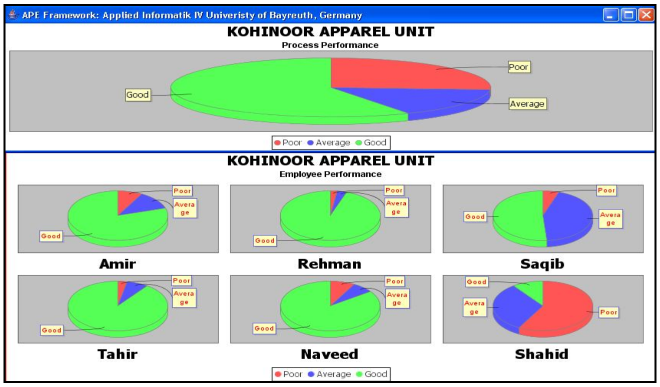

Figure 9: Performance Graph

VI. EXPERIMENTAL RESULTS AND EXPERIENCES

Over the last few years, we were involved with textile industry [29] for its process improvement effort particularly for supporting operational intelligence within WfMS in terms of improved agent assignment strategies. Their three production units namely Apparel, Dyeing and Weaving were analyzed. Each production unit has a list of processes that are performed to complete its production. Within a period of probably 39 months (from Sep 2008 to Jan 2012), we analyzed their 21 processes (Table 1). Processes within these production units were performed by more than 730 employees with the total of 39900 process instances completed, averaging 91 completed activities per day.

Statistic of the industry indicates, quick and correct decision "who should execute a certain process" is crucial. It became obvious that a manual assignment of agents is not feasible since (1) it is very time-consuming to determine successful agents more or less manually" and (2) the quality of such a manual assignment heavily depends on the experience and analytical skills of the person that performs it since a supervisor usually cannot perform a deep analysis as opposed to mining algorithms.

In order to reduce the burden of manual analysis by the process controllers and to allocate only successful employees to their business processes we deployed there our methodology. It automatically allocates resources to their processes on the basis of history of achieved business success. An example implementation in garment production processes is verifying its feasibility, since the end of 2011 they are testing it successfully. Although this field test is yet not perfect and yet complete since some implementations are still on their way but still it has convincingly demonstrated that competitive business environments can assign their employees to their processes successfully based on the history of achieved business success.

Division Name	Processes	Employees Involved	Process Instances	Time Period	
				From	To
Apparel Unit	Marker Making	20-28	2376	Mar 2008	Jan 2012
	Cutting	20-29	2079	Jun 2008	Jan 2012
	Sewing	82-07	2073	Jun 2008	Jan 2012
	Washing	25-35	1188	Dec 2008	Sep 2011
	Attachment	25-40	1188	Dec 2008	Sep 2011
	Pressing	40-50	1395	Dec 2008	Sep 2011
Dyeing Unit	Washing	10-16	1395	Sep 2008	Sep 2011
	Singing	10-16	1395	Sep 2008	Jan 2012
	Rotation	10-16	1395	Sep 2008	Jan 2012
	Desizing	10-16	1395	Sep 2008	Jan 2012
	Beaching	10-16	1395	Sep 2008	Jan 2012
	Mercerizing	10-16	1395	Sep 2008	Jan 2012
	Peaching	10-16	1395	Sep 2008	Jan 2012
	Dyeing	10-16	1395	Sep 2008	Jan 2012
	Thermosol	10-16	1395	Sep 2008	Jan 2012
	Stantoring	10-16	1395	Sep 2008	Jan 2012
	Sanforizing	10-16	1395	Sep 2008	Jan 2012
Weaving Unit	Waraping	80-120	4752	Dec 2008	Jan 2012
	Sizing	80-120	4752	Dec 2008	Jan 2012
	Weaving	80-120	4752	Dec 2008	Jan 2012

Table 1: Statistics of Production Units

We believe that this approach shows some promises for improving current state of WfMS' task managements. Our future plans include for the investigation of other inefficiencies of WfMS for further convergences of process management to make it more operational intelligent.

VII. CONCLUSION

The aim of this paper was to introduce the operational intelligence within WfMS through the use APE Framework that supports WfMS in selection and allocation of successful agents to its forthcoming processes. It also demonstrates that goal modeling concept supports context awareness for data mining techniques towards learning actions for WfMS. Also, the suitability of an exemplary implementation of APE Framework integrated with a WfMS (ProcessNavigator [14]) is presented along with our learning and experiences.

Our industrial experience indicates that the WfMS – through the use of data mining – can determine "who" is performing "how well" and under "what certain conditions" processes are being performed by "whom" and up to "what certain success level". In this way it helps WfMS to allocate agents according to the expertise of their employee – thus puts employee talents to the best use; instead of simple layoff or downsizing. Moreover, we observed that, when performance is evaluated while fostering more on employee development (i.e. training) and motivations (e.g. awards), results are more effective. Also, the convergence of data mining and process management has proven to be promising – validated from the suitability of goal concept within the process model and APE Framework in an industrial setting.

REFERENCES

[1] Gile, K., Russom, P., Moore, C. and Teubner, C. (2004), The Emergence of Process-Centric BI, Forrester Research, Cambridge, MA

[2] Jablonski, S., Bussler, C.: Workflow management. Modeling concepts, architecture and implementation, Thomson, London, UK, 1996

[3] Marjanovic, O.: The next stage of operational business intelligence – creating new challenges for business process management", in Sprague, R.H. Jr (Ed.), Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07), 2007, IEEE Computer Society, Los Alamitos.

[4] N. Russell, A. ter Hofstede,D. Edmond, W. van der Aalst, "Workflow resource patterns", Technical Report WP 127, Eindhoven Univ. of Technology (2004).

[5] N. Russell, W. von der Aalst, "Evaluation of the BPEL$People and WS-Human task extension to WS-BPEL 2.0 using the workflow resource patterns", BPMN Center Report BPMN-07-10, 2007

[6] P. Wohed, W. von der Aalst, M. DoDumas, ter Hofstede, N. Russell, "On the suitability of BPMN for business process modeling", In Proceeding of the 4th International Conference on Business Process Management (BPM) 2006

[7] S. Jablonski, R. Talib, "Agent assignment for process management: pattern based agent performance evaluation", ADMI, LNCS5680, pp. 155-169, 2009. Springer-Verlag, Berlin Heidelberg 2009.

[8] S. Jablonski, R. Talib, "Agent assignment for process management: agent performance evaluation", International Coneference on Frontier of Information Technology (FIT 2009), ACM digital library (ISBN 978-1-60558-642-7). Abbottabad, Pakistan 2009.

[9] C. Moore, "Common mistakes in workflow implementations", Giga Information Group, Cambridge M.A. (2002).

[10] M. zur Muehlen, "Organizational management in workflow applications", Informational Technology and Management Journal. Kluwer Academic Publisher, 5 (2004) 3, pp.271-291

[11] A. Shaheen, A. Mahbob, "Prioritization of key performance indcators: an integration of analytical hierarchical process and goal setting", In International Journal of Productivity and Performance Management Vol 56 No. 3. 2007 pp 226-240. DOI 10.1108/17410400731437.

[12] R. Talib, B. Volz, S. Jablonski, "Agent assignment for process management: Goal modeling for continuous resource management", submitted in "The 6th International Workshop on Business Process Design (BPD'10)"in conjunction with 8th International Conference on Business Process Management 2010.

[13] ProDatO Integration Technology GmbH: Handbuch iPM Integrated Process Manager. Softwaredocumentation (in German), Erlangen, Germany, 2005, www.prodato.de

[14] M. Faerber, S. Meerkamm, S. Jablonski, “The ProcessNavigator – flexible process execution for product development projects”, International Conference on Engineering Design (ICED _09), August 24-27, 2009, Standford (CA), USA.

[15] C. Bussler, S. Jablonski, Policy resolution for workflow management systems, in Proceeding of the 28th Hawai International Conference on System Sciences, page 831. IEEE Computer Society, 1995.

[16] H. Yan-Nong, S. Ning-Chien, Policies in resource manager of workflow systems: modeling, enforcement and management HPL-98-156, <http:/www.hpl.hp.com/techreports/98/HPL-98-156.pdf>, 1998.

[17] J. Cao, J. Chen, H. Zhao, M. Li, “A policy-based authorization model for workflow-enabled dynamic process management” In Journal of Network and Computer Applications 32, page 412-422, 2009.

[18] A. Kumar, W. von der Aalst, E. Verbeek, “Dynamic work distribution in workflow management systems: how to balance quality and performance”, Journal of Management Information Systems, 18(3):157-193, 2002.

[19] Staffware Process Suite – Defining Staffware Procedures – Issue 2. Staffware plc, Maidenhead, 2002.

[20] IBM Websphere MQ Workflow – Programming Guide – Version 3.4. IBM Corp., 2003.

[21] Wave-Front. FLOWer 3 Designers Guide. Wave-Front BV, 2004.

[22] TRANSFLOW. COSA 4 Business-Process Designer’s Guide. TRANSFLOW AG, Pullheim, 2003.

[23] N.A. Mulyar. Pattern-based Evaluation of Oracle-BPEL (v.10.1.2). Technical report, Center Report BPM-05-24, BPMcenter.org, 2005.

[24] Sun Microsystems. Sun ONE Integration Server EAI, iPlanet Version 3.1, Documentation. Palo Alto, California, USA, 2003.

[25] R. Talib, B. Volz, S. Jablonski, “Agent assignment for process management: Agent performance evaluation framework”. The International Workshop on Domain Driven Data Mining, Sydney, Australia, December 14-17, 2010.

[26] Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993).

[27] Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd Ed Morgan Kaufmann San Francisco 2005.

[28] Elena Deza & Michel Marie Deza (2009) *Encyclopedia of Distances*, page 94, Springer.

[29] Kohinoor Weaving Mills Limited. URL http://www.kohinoorweaving