Knowledge Driven Behavioural Analysis in Process Intelligence

Antonia Azzini, Paolo Ceravolo, Ernesto Damiani, and Francesco Zavatarelli

Computer Science Department, Università degli Studi di Milano
via Bramante, 65 - 26013 - Crema, Italy
email{*name*}.{*surname*}@unimi.it

Abstract. In this paper we illustrate how the knowledge driven Behaviour Analysis, which has been used in the KITE.it process management framework, can support the evolution of analytics from descriptive to predictive. We describe how the methodology uses an iterative three-step process: first the descriptive knowledge is collected, querying the knowledge base, then the prescriptive and predictive knowledge phases allow us to evaluate business rules and objectives, extract unexpected business patterns, and screen exceptions. The procedure is iterative since this novel knowledge drives the definition of new descriptive analytics that can be combined with business rules and objectives to increase our level of knowledge on the combination between process behaviour and contextual information.

1 Introduction

Process Intelligence (PI), i.e. the convergence of operational business intelligence [1] and real-time application integration, has gain a lot of attention in the last years, especially around applications involving sensor networks [2]. The final aim is to provide more accurate and fast decisions on the strategic and operational management levels. Most of the current studies on PI focus on the analysis of the process behavior and support performance improvement limited to this aspects [3]. But, descriptive analysis is contextual in nature [4], its value is clarified by the knowledge you have on a process, for instance in terms of business rules that apply and constrain a process [5]. In particular our claim is that, to insert PI into a consistent knowledge acquisition process [6], the level of its maturity and practical implementation has to evolve in the following directions:

- Not limit their analysis to process behavior but enlarge the scope to any other auxiliary data that is connected to process execution.
- Not limit to descriptive analysis but exploit the acquired knowledge for predictive analysis.

For this purpose, we introduced the KITE Knowledge Acquisition Process [6], a methodology dealing with the evolution of analytics from descriptive to prescriptive, to predictive intention. In KITE an initial set of metrics offer the initial *descriptive knowledge*. Then our analytics support the evaluation of process instances based on their consistency with policies, business rules and KPI, defined at the strategic level.

These constrains are refereed in general as *prescriptions*. Process instances violating prescriptions offer a crucial source of knowledge acquisition as *predictive analytics* can evaluate the incidence of specific variables on violations, to then derive predictive knowledge. Indeed, predictive analytics involves searching for meaningful relationships among variables and representing those relationships in models. There are response variables - things we are trying to predict, in our case violations to prescriptions. There are explanatory variables or predictors - things we observe. To generalise, as much as possible our predictive power, predictors in our case are any data related to resource auxiliary to process execution. Actually, in our approach, metrics measure process behaviour in an extended sense, as the information retrieved is not limited to the workflow, but include data related to any resource auxiliary to the process execution, as already discussed in [7]. Our approach differs from traditional predictive analytics because it is centred on the knowledge provided by the organization via Business Rules and other documentation. This approach was framed by KITE in the firm belief that it can put in contact PI and predictive analytics with Knowledge Management.

The paper is organized as follow. Section 2 starts the discussion with the related work. Sections 3 and 4 describe the KITE framework. Section 5 describes how KITE knowledge acquisition process works. Section 6 deals with behavioural and predictive analysis. Section 7 illustrate our ideas through an example. Section 8 proposes some conclusions.

2 Related Work

Predictive analytics applied to process monitoring is often limited, or strongly depended, to temporal analysis. For instance in [8] temporal logic rules are adopted to define business constraints. The approach is then focused on the evaluation of these constraints at execution time, to generate alerts that can prevent the development of violations. In [9], the authors present a set of approaches based on annotated transition systems containing time information extracted from event logs. The aim is again to check time conformance at execution time, as executions not aligned with annotated transitions predict the remaining processing time, and recommend countermeasures to the end users. An approach for prediction of abnormal termination of business processes has been presented in [10]. Here, a fault detection algorithm (local outlier factor) is used to estimate the probability of abnormal termination. Alarms are provided to early notify probable abnormal terminations to prevent risks rather than merely reactive correction of risk eventualities. Other approaches go beyond temporal analysis extending predictive analytics to include ad-hoc contextual information. In [11], a clustering approach on SLA properties is coupled with behavioral analysis to discovered and model performance predictors. In [12], the authors propose an approach running statistical analysis on process-related data, notably the activities performed, their sequence, resource availability, capabilities and interaction patterns. In [13], the authors propose an approach for Root Cause Analysis based on classification algorithms. After enriching a log with information like workload, occurrence of delay and involvement of resources, they use decision trees to identify the causes of overtime faults. In such an analysis, the availability of attributes/features that may explain the root cause of some phenomena is crucial.

On the side of knowledge acquisition procedures the literature presents several works specifically oriented to the area of business process management [14]. However only a few are really considering analytics as a key element of this process. For instance in [15] the authors exploit the notion of knowledge maintenance process. process mining is applied to analyze the knowledge maintenance logs to discover process and then construct a more appropriate knowledge maintenance process model. The proposed approach has been applied in the knowledge management system.

Our work is characterized by the introduction of an extended notion of process behavior that provide a generalized systematic approach to captures process features beyond workflow execution. This element is the exploited within a knowledge acquisition methodology that exploits prescriptive and predictive analytics to acquire novel and unexpected knowledge.

3 The KITE Methodology

KITE.it is a project co-funded by the Italian Ministry for Economic Development, within the "Industria 2015" Program, in the area of "New technologies for Made in Italy" [16]. The exit from the great global crisis towards a new cycle of development requires to move from organizational and inter-organizational models, based on a strict definition of roles and organizational boundaries. In this context, KITE.it is aimed at developing a business and social cooperation framework that enables interoperability among enterprises and other knowledge workers, making available a variety of tools and technologies developed to connect the processes of an organization to those of suppliers or to involve customers in planning and assessing activities. In fact, the KITE.it framework should be capable of supporting procedures such as *i)* creation, contextualization and execution of metrics, *ii)* connection between metrics and strategic level, and *iii)* inception and capitalization of the results. The final goal is driving the monitoring process to derive previously unknown and potentially unexpected knowledge.
To circumscribe our discussion, in this paper we examine a single aspect of the KITE.it Framework, focusing on how it was extended to cover data integration and interoperability, as discussed in Section 4. Moreover, we are considering how these characteristics was exploited in guiding the Knowledge Acquisition Process, as discussed in Sec 5.

4 The KITE Knowledge Base

The KITE Knowledge Base (KKB) has to integrate a variety of heterogeneous data from the different sources composing the KITE.it Framework.

This requirement is faced adopting a graph-based model to structure and link data according to the Web Standards, the so-called Resource Description Framework. Generally speaking, the Resource Description Framework (RDF) [17] provides a standard for defining vocabularies, which can be adopted to generate directed labeled graphs [18], in which entities edges and value are associated with terms of the vocabulary. For this reason, RDF is an extremely generic data representation model that can be extended easily with any domain-specific information. Moreover, RDF is a monotonic declarative language, i.e. the acquisition of new data cannot invalidate the information previously acquired.

The atomic elements of a RDF graph are triples[1]. Triples are composed by three elements: resources, relations between resources and attributes of resources. These elements are modeled within the labelled oriented graph, as the atomic structure $< s, p, o >$ where s is subject, p is predicate and o is object, combined as shown in Figure 1.

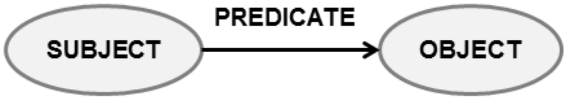

Fig. 1. RDF subject-object relation.

New information is inserted into an RDF graph by adding new triples to the data set. It is therefore easy to understand why such a representation can provide big benefits for real time business process analysis: data can be appended ‘on the fly’ to the existing one, and it will become part of the graph, available for any analytical application, without the need for reconfiguration or any other data preparation steps.

Assuming pairwise disjoint infinite sets I, B, L (*IRIs*[2], *Blank nodes*, *RDF Literals*).

Definition 1 *A tuple* $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$ *is called an RDF triple.*

An RDF graph G is a set of RDF triples. An interesting feature of RDF standards is that multiple graphs can be stored in a single RDF Dataset. As stated in the specifications “*An RDF Dataset comprises one graph, the default graph, which does not have a name, and zero or more named graphs, where each named graph is identified by an IRI*”.

RDF standard vocabularies allow external applications to query data through SPARQL query language [19]. SPARQL is a standard query language for RDF graphs based on conjunctive queries on triple patterns, identifying paths in the RDF graph. Thus, queries can be seen as graph views. SPARQL is supported by most of the triples stores available.

If we now introduce a novel infinite set V for variables, disjoint from I, B, and L we can define SPARQL patterns as in the following.

Definition 2 *A tuple* $t \in (I \cup L \cup V) \times (I \cup V) \times (I \cup L \cup V)$ *is called a SPARQL triple pattern. Where the blank nodes act as non-distinguished variables in graph patterns.*

Definition 3 *A finite set of SPARQL triple patterns can be constructed in a Graph Pattern (GP) using OPTIONAL, UNION, FILTER and JOIN. A Basic Graph Pattern is a set of triple patterns connect by the JOIN operator.*

The semantics of SPARQL is based on the notion of mapping, defined in [20] as a partial function $\mu : V \to (I \cup L \cup B)$. Where, if GP is a graph pattern and $var(GP)$ denotes

[1] An alternative terminology adopted in documentation is *statements* or eventually *tuples*.

[2] IRIs are the RDF URI references, IRIs allow all characters beyond the US-ASCII charset.

the set of variables occurring in GP; given a triple pattern t and a mapping μ such that $var(t) \subseteq dom(\mu)$, μ is the triple obtained by replacing the variable in t according to μ.

In [21], the authors present a framework based on RDF for business process monitoring and analysis. They define an RDF model to represent a generic business process that can be easily extended in order to describe any specific business process by only extending the RDF vocabulary and adding new triples to the triple store. The model is used as a reference by both monitoring applications (i.e., applications producing the data to be analyzed) and analyzing tools. On one side, a process monitor creates and maintains the extension of the generic business process vocabulary either at start time, if the process is known a priori, or at runtime while capturing process execution data, if the process is not known. Process execution data is then saved as triples with respect to the extended model. On the other side, the analyzing tools may send SPARQL queries to the continuously updated process execution RDF graph.

Figure 2 shows the schema of an RDF Dataset composed by the union of two graphs. The resources describing the generic model of a business process are tagged in blue. They can represent a sequence of different tasks, each having a start/end time and having zero or more sub-tasks. The resources tagged in yellow represent domain-specific concepts describing the repair and overhaul process in avionics. In this very simple extract we defined a process, in connection with its tasks, and the customer purchasing the overhaul operations.

Once this schema is defined any process execution is stored in the KKB in terms an RDF Dataset composed of triples conforming with the schema. For instance, in 1 a legal dataset is presented.

```
av:p1 rdf:type av:Overhaul
av:p1 bpm:hasTask av:t1
av:p1 bpm:hasTask av:t2
av:t1 bpm:followedBy av:t2
av:t1 bpm:startTime "2013-06-06 10:38:45"^^xsd:date
av:t1 bpm:endTime "2013-06-06 18:12:35 "^^xsd:date
av:t1 rdf:type av:Inspect
```

(1)

5 The KITE Knowledge Acquisition Process

The methodology considers the KITE Knowledge Acquisition Process (KKAP) as an investigation over the process executions, as registered in the KKB. In particular, this methodology is organised in iterations over three fundamental steps.

- *Descriptive Knowledge*: querying triples on the process execution, or any other auxiliary resource, you have a descriptive summary of the process in terms of frequency, dimension, and central tendency [22].
- *Prescriptive Knowledge*: evaluating the achievement of the business rules or the objectives associated to a process, as well as identifying unexpected patterns, you can screen of process executions isolating exceptions that are violating some prescription [23].

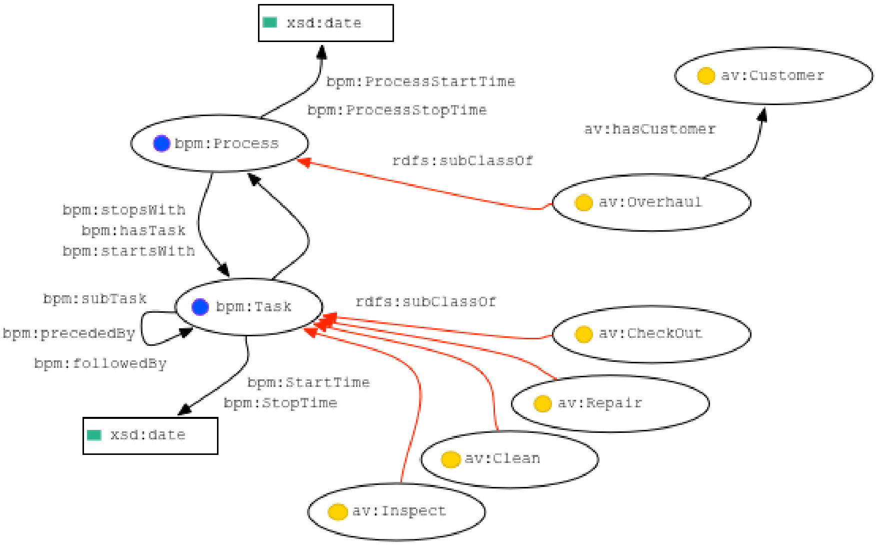

Fig. 2. RDF Representation of a generic business process.

- *Predictive Knowledge*: process executions screened by prescriptions can be further investigated evaluating the incidence of specific properties on specific partitions of the KKB. This allows to acquire novel knowledge on the process that eventually can result in new descriptive or prescriptive knowledge.

Before providing further definitions let us clarify our purpose by a simple example of two iterations.

5.1 First iteration

The engine maintenance is a very complex process performed by the aerospace industry. Generally speaking, maintenance operations are needed on a regular time basis (*Inspect Only*, *Minor Revision* or *General Revision*, according to the number of flown hours) or when a part has failed (*Out of Order*), as shown in table 1. The activities vary accordingly.

	Inspect (I)	**Disassembly** (DA)	**Inspect Mod.** (IM)	**Repair** (R)	**Clean** (C)	**Assembly** (A)	**Bench Test** (BT)	**Checkout** (CO)
General Rev.		Yes	Yes		Yes	Yes		Yes
Minor Rev.	Yes				Yes			Yes
Inspect Only	Yes							Yes
Out of Order	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 1. Representation of the maintenance processes in the aerospace industry (simplified).

Suppose to focus on minor and general revision processes, and collect the duration in days of all the process executions involving the activities *Inspect*, *Clean* and *CheckOut* ($I \gg C \gg CO$ in short) in case of minor revision, or *Disassembly*, *Inspect Module*, *Clean*, *Assembly* and *CheckOut* ($DA \gg IM \gg C \gg A \gg CO$ in short) when general revision is performed. Results can be summarised as illustrated in table 2. In this way you have Descriptive Knowledge about the processes.

ProcessID	**Task Sequence**	**Duration (days)**
p12	Minor Revision	3
p31	Minor Revision	4
p33	Minor Revision	5
p39	Minor Revision	3
p11	Minor Revision	8
p05	Minor Revision	5
p101	General Revision	12
p102	General Revision	11
p103	General Revision	13
p104	General Revision	11
...	...	...

Table 2. Duration in days of process executions involving Minor and General Revision.

To acquire Prescriptive Knowledge you have to compare your data with some prescriptions. By this term here we refer to any constraint or property the business processes execution should satisfy. In the Business Process Management literature, this function is typically associated with Business Rules [24], even if their scope is not limited at assessing the business behavior but involves the business structure as well (for instance defining the corporate governance). Business Rules can derive from internal objectives and strategies or from external factors such as contractual constrains or legal requirements. However, Business Rules can also be discovered by data mining [25] or process mining [26], for instance by identifying recurrent behavior.

Once a prescription is defined you are able to partition the dataset based on the violations of this prescription. If the violation can be associated to an intensity the partitions depend on a degree, otherwise the partition is binary. For instance Business Rules could prescribe the expected duration of process executions: $Duration \leq 7\ days$ if Minor Revision and $Duration \leq 11\ days$ if General Revision. Table 3 shows the result of this operation. The prescription that have been learned from a dataset d can be applied to other datasets D, under the assumption that d is a representative sample of D.

The notion of violation is crucial in the KITE methodology as it identify an observation that is not consistent with our expectations and we would like to avoid for future executions. Investigating the incidence of specific resources on the sub set of the violations we can induce additional knowledge to support explanation or resolution of process executions violating our prescription. To draw conclusions of our example let us introduce an additional resource in our view of the dataset, as illustrated in Table 4. If we can observe a significant incidence of this resource to the subset of the violations we

ProcessID	Task Sequence	Duration	Violation
p12	Minor Revision	3	NO
p39	Minor Revision	3	NO
p31	Minor Revision	4	NO
p33	Minor Revision	5	NO
p05	Minor Revision	5	NO
p102	General Revision	11	NO
p104	General Revision	11	NO
...	...	...	...
p11	Minor Revision	8	YES
p101	General Revision	12	YES
p103	General Revision	13	YES
...	...	...	...

Table 3. Dataset partitioned according to the prescription (business rule).

can attest the acquisition of novel knowledge that should be exploited for the definition of a second iteration of the KKAP.

ProcessID	Task Sequence	Duration	Violation	Customer Type
p12	Minor Revision	3	NO	Civil
p39	Minor Revision	3	NO	Civil
p31	Minor Revision	4	NO	Civil
p33	Minor Revision	5	NO	Military
p05	Minor Revision	5	NO	Civil
p102	General Revision	11	NO	Civil
p104	General Revision	11	NO	Civil
...	...	...	...	...
p11	Minor Revision	8	YES	Military
p101	General Revision	12	YES	Military
p103	General Revision	13	YES	Military
...	...	...	...	...

Table 4. Incidence of an additional resource (customer type) on the violations.

5.2 Second iteration

As previously stated, this is an iterative process, which takes the last data as a starting point for the second iteration, see Figure 3.

We start our second iteration as shown in table 5, where some other kind of processes (*Inspect Only* and *Out of Order*) are added to the dataset for a better understanding. We also introduce additional resources, in this case the notion whether the activities were performed by internal company staff or outsourced to somebody else.

Defining another prescription we are able again to partition the dataset based on the violations of the new business rule. Consider for example to define the following

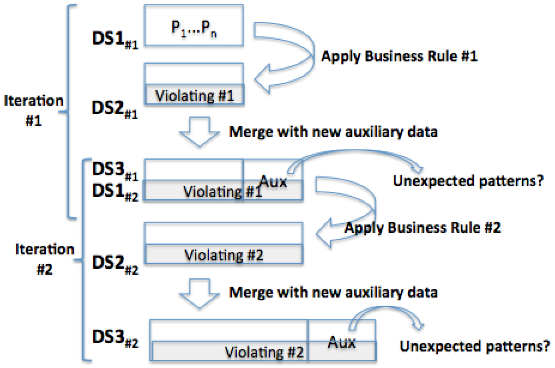

Fig. 3. iterations

prescription: operations must not be outsourced if the engine belongs to a military customer. Table 6 shows the result of this operation.

ProcessID	Task Sequence	Customer Type	Staff
p11	Minor Revision	Military	Outsourced
p101	General Revision	Military	Internal
p103	General Revision	Military	Internal
p202	Inspect Only	Military	Internal
p301	Out of Order	Military	Outsourced
...	...	...	...

Table 5. Responsibility of the activities, as performed by internal staff staff or outsourced.

ProcessID	Task Sequence	Customer Type	Staff	Violating
p101	General Revision	Military	Internal	NO
p103	General Revision	Military	Internal	NO
p202	Inspect Only	Military	Internal	NO
...	...	...	...	...
p11	Minor Revision	Military	Outsourced	YES
p301	Out of Order	Military	Outsourced	YES
...	...	...	...	...

Table 6. Violations of the second business rule.

Investigating again the incidence of a specific resource on the subset of the violation we can induce additional knowledge and support explanation. In our case, in order to draw conclusions we introduce an additional resource in our view of the dataset, as illustrated in Table 7. When we observe a significant incidence of this resource to the subset of the violations we have acquisition of novel knowledge.

ProcessID	Task Sequence	Customer Type	Staff	Violating	Certified
p101	General Revision	Military	Internal	NO	YES
p103	General Revision	Military	Internal	NO	YES
p202	Inspect Only	Military	Internal	NO	YES
...	...	...	...	...	...
p11	Minor Revision	Military	Outsourced	YES	NO
p301	Out of Order	Military	Outsourced	YES	NO
...	...	...	...	...	...

Table 7. An additional resource can lead to novel knowledge.

6 Predictive Analytics

As illustrated in [7] we extended the notion of Behavioral Analysis as a weaker form of classic behavior equivalence, where two compatible behaviors have to be equivalent with respect to activities they have in common [27]. To characterise a process execution log, for instance for detecting ordering relations among events, it is common to start by the definition of *process execution tracks*, *workflow trace* as defined in [28]. In our approach, we extended this definition by auxiliary resources, considering any data related to the events in a trace that are consistent with a graph pattern over the KKB.

As already mentioned the KKAP includes predictive analytics aimed at identifying the incidence of KKB's resources on process execution. In general, predictive analytics encompasses a variety of statistical techniques from modeling, machine learning, and data mining that uncovers relationships and patterns within large volumes of data that can be used to predict behavior and events [29]. Here we adopt the term to refer to this part of our methodology that is is forward-looking, i.e. uses past events to better understand the process. In particular our aim is to investigate data about resources auxiliary to process execution, searching for incidence with those process instances that are violating prescriptions. Now, our aim is to define how this incidence is evaluated.

The approach adopted in KITE is based on Bayesian statistics [30]. Bayesian statistics offers the theoretical framework for combining experimental and extra-experimental knowledge. In particular, Bayesian procedures, for evaluating the predictive power of a parameter in a statistical model, take into account both experimental data and information on the parameter incorporated in the so-called prior distribution[3]. This is an important point of distinction with frequentist approaches, most commonly used. The most practical consequence is that frequentist approaches impose assumptions on the distribution for both the random sample and the model tested. Different hypothesis tests have different model assumptions. For many tests, the model assumptions consist of several conditions. If any one of these conditions is not true, we do not know that the test is valid. But these assumptions cannot be easily verified on any kind of data sets, in particular when dealing with data flows acquired or consumed at low interval rates.

[3] It is however well know that the conflict between Bayesian and frequentist procedures tends to disappear as the sample size increases. Indeed, the discrepancies are limited when sampling information dominates the prior distribution or pre-experimental information may influence the estimates on prior distribution.

Following a Bayesian approach, we consider H an unknown hypothesis; $\mathbf{X} = \{X_1, ..., X_n\}$ is a set of independent and identically distributed observations. Let $x_n = (x_1, ..., x_n)$ be an observed sample; $\pi(H)$ is the prior probability of the hypothesis under test; $\pi(\mathbf{X}|H)$ the likelihood; and the posterior distribution is defined as in equation 2.

$$\pi(H|\mathbf{X}) = \frac{\pi(\mathbf{X}|H)\pi(H)}{\sum_n \pi(\mathbf{X}_n|H_n)\pi(H_n)} \tag{2}$$

Predictive modeling involves finding good subsets of predictors or explanatory variables. Models that fit the data well are better than models that fit the data poorly. Simple models are better than complex models. Working with a list of useful predictors, we can fit many models to the available data, then evaluate those models by their simplicity and by how well they fit the data.

7 A Preliminary Example

To illustrate the approach proposed in KITE.it, we now provide a running example. Let us start from a sample business rule stating that: "On an equipment fault, operators will visit customers premises within 12 hours from fault reporting". Our aim is to discover new knowledge from the information detected by monitoring the process in connection to this policy. We then formulate a *predictive analysis* considering the incidence of "previous visits to the same client by the same operator" to violations to these policies.

We start by a *descriptive metrics* that can be computed using a query listing the *excess_time*, expressed in hours and computed as the difference between *visit_time* and *fault_time*, for a set of tuples extracted from specific traces identified by *ProcessID*. Table 8 illustrate an sample of the results returned querying a data set.

ProcessID	FaultID	VisitID	OpId	ExcessTime
12	AF01	AEFF	1	20
31	AB00	AB07	3	3
33	A777	AA01	7	16
15	AB43	AA08	4	4
39	A605	AAB0	9	8
29	AK15	AA04	13	19
11	AG33	AA42	14	7
21	AB06	AB17	8	14
11	AG43	AA22	12	16
05	AB23	AA78	19	13

Table 8. Excess time from equipment fault to visit of an operator.

The *prescription* we want to apply to these traces imposes a constraint of form $excess\_time > 12$. Filtering traces by this constraint we obtain the set of violations $V: \{12, 33, 29, 21, 11, 05\}$. This set must be compared to the set of traces ordered by the

the number of previous visits by same operator to clients. Another *descriptive metrics* is then defined to extract these data, getting a distribution E. Table 9 illustrate an sample of the results returned.

ClientID	**VisitID**	**OpId**	**VisitPriorToFault**
C121	AEFF	1	3
C313	AB07	3	2
C236	AA01	7	6
C118	AA08	4	4
C259	AAB0	9	1
C329	AA04	13	1
C311	AA42	14	2
C111	AB17	8	4
C319	AA22	12	6
C209	AA78	19	3

Table 9. Visit prior to fault from the operators involved in visits listed in Table 8.

The *predictive analysis* is then executed by evaluating the incidence of different partition E on V. More specifically, referring to the equation 2, V is the hypotheses H we are testing and E is the observation $\mathbf{X}$. We are in other words evaluating how confident we are that observing a trace included in E this trace will also be in V.
If $\mathbf{X}$ is an ordinal variable we can test these incidence for each subset of the distribution by imposing a threshold α for defining membership of the subset under consideration.

$$\mathbf{X}_\alpha = \{x \geq \alpha, \forall x \in \mathbf{X}\} \tag{3}$$

So we can straightforwardly proceed to calculate the posterior probability $\pi(V|\mathbf{E}_\alpha)$. For instance taking $\alpha = 3$ we have six process instances in $\mathbf{E}_\alpha$, with five of them in V: $\pi(V|\mathbf{E}_\alpha) = \frac{5}{6} = 0.83$. Table 10 shows the results imposing a thresholds α for each value in E.

α	**Posterior Probability**	**Prior Probability**
1	0.6	0.6
2	0.625	0.6
3	0.83	0.6
4	0.75	0.6
6	1	0.6

Table 10. Incidence to violations of different α on values of “visit prior to fault”.

We find that process instances related to 3 or more “visit prior to fault time” present high probability to violate the business rules defining the expected execs time from fault to visit. In particular this is shaping a behavior that is potential dysfunctional, e.g. due to a “cry wolf” effect.

It is important to note that in this example we used “educated guesses” to decide the set of parameters to be used for the process behavior metrics. An exhaustive search of the right parameter set to identifying inductive metrics would be computationally very expensive. Clearly several not exhaustive approaches are possible. Ranging from considering the expert intuitions to game-theoretical algorithm aimed at identifying parameter sets based on their effectiveness in the winning strategy for an attacker wishing to fail the KPI without being caught, as explained in [31].

8 Conclusion

Most of the time, the literature has disregarded a notion of process behaviour that comprehensively includes alla data related to resources auxiliary to process execution. As a consequence, the method proposed for implementing Predictive Analytics usually are not fully integrated with a Knowledge Acquisition procedure, for instance, without providing concrete guidelines on how to move form one measurement step to another.

In this paper we put forward the idea that the full integration of PI capabilities requires to introduce a notion of Extended Behaviour, as the value of the information available in processes becomes one of the most important source for Predictive Analytics bringing to the acquisition of novel knowledge. .

Acknowledgment

This work was partly funded by the Italian Ministry of Economic Development under the “Industria 2015” contract - KITE.IT Project.

References

1. Surajit, C., Umeshwar, D., Vivek, N.: An overview of business intelligence technology. Commun. ACM **54**(8) (2011) 88–98
2. Yoo, Y.S., Yu, J., Lee, B.B., Bang, H.C.: A study on ubiquitous business process management for real-time open usn mash-up services. In: Information Technology Convergence, Secure and Trust Computing, and Data Management. Springer (2012) 21–27
3. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Process intelligence. In: Fundamentals of Business Process Management. Springer (2013) 353–383
4. Colombo, A., Damiani, E., Frati, F., Oltolina, S., Reed, K., Ruffatti, G.: The use of a meta-model to support multi-project process measurement. In: Proceedings of 15th Asia-Pacific software engineering conference (APSEC 2008), Beijing, China (2008) 503–510
5. Arigliano, F., Bianchini, D., Cappiello, C., Corallo, A., Ceravolo, P., Damiani, E., De Antonellis, V., Pernici, B., Plebani, P., Storelli, D., et al.: Monitoring business processes in the networked enterprise. In: Data-Driven Process Discovery and Analysis. Springer (2012) 21–38
6. Azzini, A., Ceravolo, P., Damiani, E., Zavatarelli, F., Vicari, C., Savarino, V.: Driving knowledge acquisition via metric life-cycle in process intelligence. In: Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business, ACM (2014) 26

7. Ceravolo, P., Zavatarelli, F.: Knowledge acquisition in process intelligence. In: Proceedings of the International Conference on Information and Communication Technology Research. (to be pblished)
8. Maggi, F.M., Francescomarino, C.D., Dumas, M., Ghidini, C.: Predictive monitoring of business processes. In: CAiSE. (2014) 457–472
9. van der Aalst, W., Schonenberg, M., Song, M.: Time prediction based on process mining. Information Systems **36**(2) (2011) 450 – 475 Special Issue: Semantic Integration of Data, Multimedia, and Services.
10. Kang, B., Kim, D., Kang, S.H.: Real-time business process monitoring method for prediction of abnormal termination using knni-based lof prediction. Expert Syst. Appl. **39**(5) (April 2012) 6061–6068
11. Folino, F., Guarascio, M., Pontieri, L.: Discovering context-aware models for predicting business process performances. In: On the Move to Meaningful Internet Systems: OTM 2012. Springer (2012) 287–304
12. Pika, A., van der Aalst, W.M., Fidge, C.J., ter Hofstede, A.H., Wynn, M.T.: Predicting deadline transgressions using event logs. In: Business Process Management Workshops, Springer (2013) 211–216
13. Suriadi, S., Ouyang, C., van der Aalst, W.M., ter Hofstede, A.H.: Root cause analysis with enriched process logs. In: Business Process Management Workshops, Springer (2013) 174–186
14. Papazoglou, M., Heuvel, W.V.D.: Business process development life cycle methodology. In: Communications of the ACM. (2007) 79–85
15. Li, M., Liu, L., Yin, L., Zhu, Y.: A process mining based approach to knowledge maintenance. Information Systems Frontiers **13**(3) (2011) 371–380
16. Arigliano, F., Azzini, A., Braghin, C., Caforio, A., Ceravolo, P., Damiani, E., Savarino, V., Vicari, C., Zavatarelli, F.: Knowledge and business intelligence technologies in cross-enterprise environments for italian advanced mechanical industry. In: Proceedings of the 3rd International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2013), Riva del Garda (TN), CEUR-WS.org (2013) 104–110
17. Hayes, P., McBride, B.: Resource description framework (rdf). `http://www.w3.org/standards/techs/rdf` Date: 2014.
18. Carroll, J., Bizer, C., Hayes, P., Stickler, P.: Named graphs. Journal of Web Semantics **3**(3) (2005)
19. Garlik, S.H., Seaborne, A.: Sparql 1.1 query language. `http://www.w3.org/TR/2013/REC\--sparql11\--query\--20130321/` Date: 2013.
20. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of sparql. ACM Transactions on Database Systems (TODS) **34**(3) (2009) 16
21. Leida, M., Majeed, B., Colombo, M., Chu, A.: Lightweight rdf data model for business processes analysis. Data-Driven Process Discovery and Analysis Series: Lecture Notes in Business Information Processing **116** (2012)
22. Holsapple, C.W.: The inseparability of modern knowledge management and computer-based technology. Journal of knowledge management **9**(1) (2005) 42–52
23. Arigliano, F., Bianchini, D., Cappiello, C., Corallo, A., Ceravolo, P., Damiani, E., Antonellis, V.D., Pernici, B., Plebani, P., Storelli, D., Vicari, C. Lecture notes in business information processing ; 116. In: Monitoring business processes in the networked enterprise. Springer, Berlin (2012)
24. Von Halle, B., Goldberg, L.: Business Rule Revolution (ebook): Running Business the Right Way. Happy About (2006)
25. Taylor, J.: Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics. Pearson Education (2011)

26. Bezerra, F., Wainer, J., van der Aalst, W.M.: Anomaly detection using process mining. In: Enterprise, Business-Process and Information Systems Modeling. Springer (2009) 149–161
27. Wombacher, A., Li, C.: Alternative approaches for workflow similarity. In: Services Computing (SCC), 2010 IEEE International Conference on, IEEE (2010) 337–345
28. Van Der Aalst, W., Van Hee, K.: Workflow management: models, methods, and systems. MIT press (2004)
29. Fanning, K., Centers, D.P.: Intelligent business process management: Hype or reality? Journal of Corporate Accounting & Finance **24**(5) (2013) 9–14
30. Lee, P.M.: Bayesian statistics: an introduction. John Wiley & Sons (2012)
31. Tomas, C.J.: Game theory methods of identifying potential key factors. https://people.stanford.edu/calebj/content/game-theory-methods-identifying-potential-key-factors-0 (2012)