Toward Uncertain Business Intelligence: the Case of Key Indicators*

Carlos Rodríguez, Florian Daniel, Fabio Casati
University of Trento, Italy
{crodriguez, daniel, casati}@disi.unitn.it

Cinzia Cappiello
Politecnico di Milano, Italy
cappiell@elet.polimi.it

Abstract. Decision support systems and, in particular, business intelligence techniques are widely used by enterprises for monitoring and analyzing operations to understand in which aspects the business is not performing well and even how to improve it. These tools provide valuable results in the context of single departments and business processes, while they are often not suitable in scenarios driven by web-enabled intercompany cooperation and IT outsourcing. In such contexts, the adoption of service-oriented company IT architectures and the use of external web services may prevent the comprehensive view over a distributed business process and raise doubts about the reliability of computed outputs. We analyze how these scenarios impact on information quality in business intelligence applications and lead to non-trivial research challenges. We propose the notions of uncertain events and uncertain key indicators, a model to express and store uncertainty, and a tool to compute with and visualize uncertainty.

Keywords. Uncertain Business Intelligence, Uncertain Key Indicators, Cooperative Processes, Data Quality, Possible Worlds.

Introduction

The increased usage of IT to support business operations and the advances in business intelligence (BI) techniques create the opportunity for monitoring and analyzing operations to understand in which aspects a business is not performing well and even how to improve it. This has been happening for a while in the context of single departments and business processes, but now it is extending to BI applications that integrate data from multiple departments and even multiple companies. Common examples are the now omnipresent Enterprise Data Warehouse [1], which aggregates process data across departments and geographies; business process outsourcing scenarios, in which the execution of a process is delegated to other companies; or inter-company cooperation, where data and processes are shared across multiple companies.

While BI applications are often complex and comprise multiple kinds of analyses, one of the most widely used metaphors is that of *Key Indicators* (KI) [2], a set of values that summarize the performance of critical business operations. KIs are used to detect problems and trigger business decisions.

Despite the importance of KIs to business, little attention has been devoted to the expressiveness of KIs if they are computed out of low-quality data and to how possible uncertainties can be communicated to the BI analysts. Even in closed scenarios there are many possible sources of uncertainty in BI applications [4], and the problem is magnified when data comes from multiple sources and is collected with different methods and frequency by different departments, institutions, and geographies. In some cases, uncertainty can easily be predicted or detected (e.g., a partner does not send data on time or a source has an inherently unreliable data collection method), while in others the problems are occasional and harder to recognize. The goal of this article is to understand how to deal with the lack of a comprehensive knowledge about organizational business processes and how to compute meaningful indicators, despite uncertainty in the underlying data.

Motivation: Key Assurance Indicators in Healthcare

In the context of the EU project MASTER[1] (Managing Assurance, Security and Trust for sERvices; 9.3M€ of funding) we are developing diagnostic algorithms to assess and report on compliance, even in presence of uncertain data. So-called *Key Assurance Indicators* (KAIs) are used to measure performance against compliance requirements, e.g., deriving from a privacy law. Algorithms are being tested in collaboration with Hospital San Raffaele (Milano, Italy), which provides the necessary, distributed business context: their outpatient drug dispensation process. We summarize the process in Figure 1.

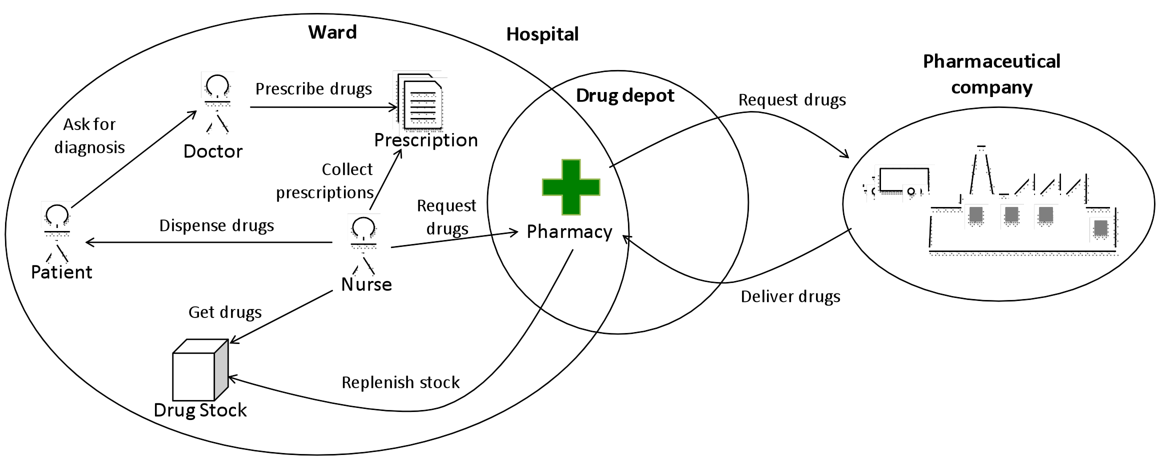


Figure 1 Outpatient drug dispensation in a hospital

The process starts with the patient's visit to the doctor in the hospital's ward. In the case any treatment is needed, the doctor sends an according prescription for drugs to the nurse, and the patient can ask the nurse for the dispensation of the drugs. The nurse collects all drug prescriptions and checks whether all necessary drug quantities are in stock. If yes, he/she can immediately dispense the drugs to the patient. If not, he/she must issue a drug request to the

[1] http://www.master-fp7.eu

Pharmacy of the hospital, which is then in charge of providing the requested drugs. If, in turn, the Pharmacy is running out of stock, the personnel in charge issues a request to the Pharmaceutical Company that provides drugs to the Pharmacy. By law, the hospital must guarantee that all patient data are anonymized throughout the process, and the hospital's internal policy states that drug replenishment by the Pharmacy must occur within maximum two business days. In order to control, for instance, this latter aspect, the hospital wants to compute a KAI called *Average Replenishment Duration* (ARD), which allows the hospital to monitor the time it takes to refill the *Ward's* drug stock.

From the IT point of view, the drug dispensation process is supported by several web service-based information systems that interact inside a service-oriented architecture (SOA). For instance, there are web services for issuing drug requests in the various dependencies of the institute, and the pharmaceutical companies the hospital cooperates with accept drug requests through web service interfaces. To retrieve the data requested by the hospital's BI application, during the execution of the process suitable events are generated, which can be logged and analyzed. In this article, we assume each arrow in Figure 1 corresponds to an event in form of a simple SOAP message.

Uncertain Events

The above process describes a BI scenario where data are sourced from multiple cooperating entities or companies. This kind of scenario is typically characterized by different levels of visibility into a partner's business activities and by different levels of trust in the visible data that can be obtained from each partner.

In the case of *cooperative processes* (processes that span across organizational domains [3], e.g., the *Ward*, the *Stock management*, and the *Pharmaceutical company*), we can distinguish three kinds of business events: (i) *Internal* events that stem from the activities that are under the control of the company (the *Ward*) and consequently are completely visible and trustable. (ii) *Shared* events that are originated in the activities that are shared with the integrated partner (the *Stock management*); depending on the technical solution adopted for the implementation of the cooperative part of the process, the visibility into its internals (the events) might be lower than in the case of own activities; similarly, trust into events might be lower. (iii) *External* events that are part of the partner's *internal processes*; these events are typically hidden to the company, and we cannot analyze them (e.g., we do not have access to the *Stock managements* internal processes). Similarly, we can associate visibility and trust levels to the case of *outsourced processes* (the production and shipment of drugs by the *Pharmaceutical company*). Yet, in this case both visibility and trust are typically lower than in the cooperative process scenario.

The visibility into shared or outsourced processes has typically structural or organizational roots (e.g., the use of incompatible IT systems or privacy restrictions) that do not frequently change over time. Trust in partners and the information they provide might instead vary with faster dynamics, e.g., based on trust assessment systems that automatically assess trust values for partners from past interactions (see sidebar 1 for more details).

Trust and Reputation in Web-based Collaboration

Trust and reputation are concepts studied in different fields, e.g., economics, sociology, computer science, and biology. Although there is a growing literature on theory and applications of trust and reputation systems, definitions are not always coherent[1]. However, the concept of trust is undoubtedly associated with the concept of reliability[2]: *trust* is the *subjective* probability by which a party expects that another party performs a given action on which its welfare or business depends[3,1]; *reputation* is the *general* opinion about a person, a company, or an object. Therefore, while trust derives from personal and subjective phenomena, reputation can be considered as a collective measure of trustworthiness based on the referrals or ratings from members in a community.

To computer scientists, trust and reputation are particularly significant to support decisions in Internet-based service provisioning. Especially, reputation is able to drive the relationships of individuals and firms in online marketplaces[4,5]. For instance, collaborative filtering systems are used to judge the behavior of a party and to assist other parties in deciding whether or not to start business with that party. A reputation system collects, distributes, and aggregates feedbacks about participants' past behavior and discourage unfair behavior[6]. The cross-analysis of different reputation systems enables the realization of mechanisms and methods for the online reputation monitoring and improvement[7].

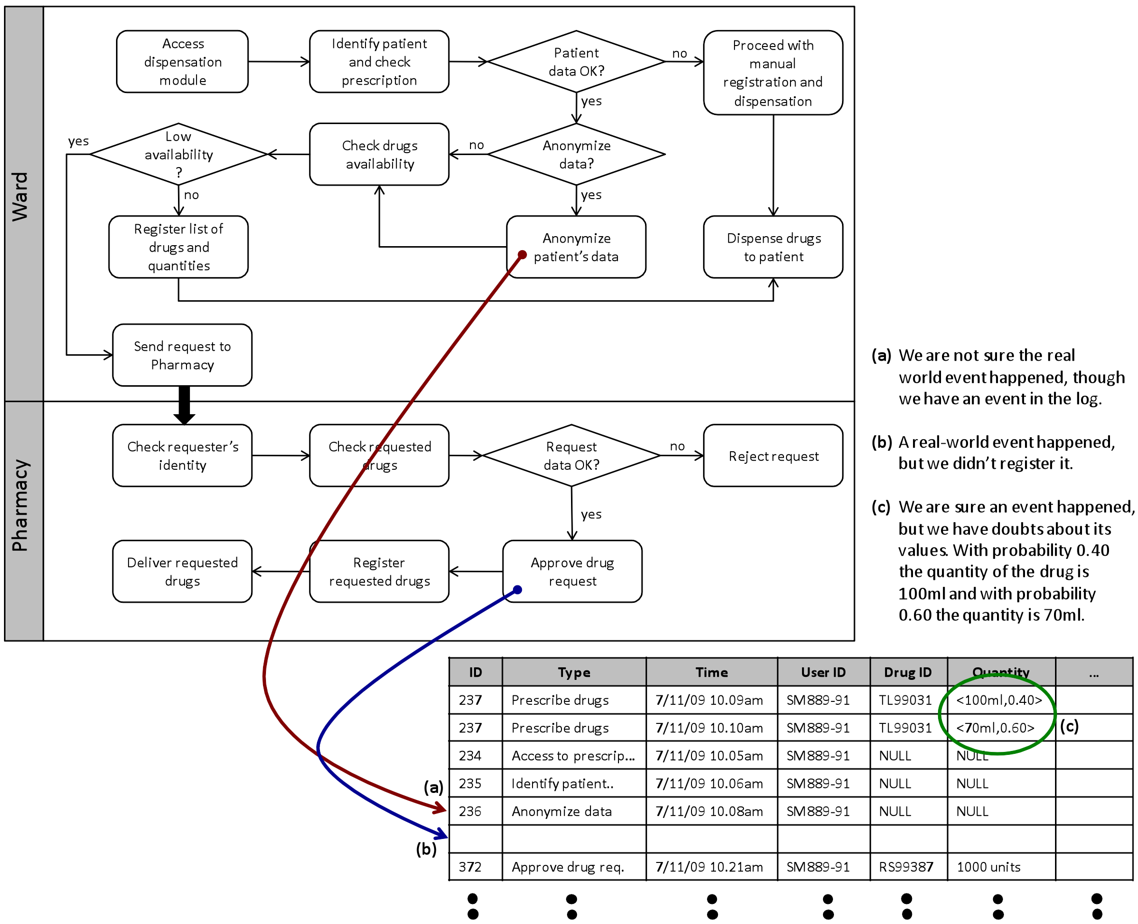
References

1. D.H. McKnight and N.L. Chervany. The Meanings of Trust. Technical Report MISRC Working Paper Series 96-04, University of Minnesota, Management Information Systems Reseach Center, 1996.
2. A. Jøsang R. Ismail C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems, 43(2), 2007, pp.618-644.
3. D. Gambetta (1988) Trust: Making and Breaking Cooperative Relations, Oxford: Basil Blackwell.
4. M. Fan, Y. Tan, A. B. Whinston. Evaluation and Design of Online Cooperative Feedback Mechanisms for Reputation Management. IEEE TKDE 17(2), 2005, pp. 244-254.
5. G. Zacharia, P. Maes. Collaborative Reputation Mechanisms in Electronic Marketplaces. Proc. 32nd Hawaii International Conf on System Sciences, 1999.
6. P. Resnick, K. Kuwabara, R. Zeckhauser, E. Friedman (2000a). Reputation Systems. Communications of the ACM, 43(12): 45-48.
7. C. Ziegler, M. Skubacz. Towards Automated Reputation and Brand Monitoring on the Web. In Proceedings of the 2006 IEEE/WIC/ACM international Conference on Web intelligence, 2006.

With the use of web services and the SOA, cooperative processes moved to the Web. The consequent reliability problems raise information quality issues in the collection of the events upon which BI algorithms can perform their analyses. In this context, we identify some issues that are strongly related with the way events are collected (the situation is graphically represented in Figure 2):

- *We registered an event in the log, yet we are not sure the corresponding real-world event really happened* (case (a)). For instance, it may happen that the system is not able to successfully anonymize a patient's data, e.g., due to a failure in the algorithm. If the failure is not registered properly, we register a wrong anonymization event.
- *A real-world event happened, but we couldn't register it in the log* (case (b)). In a running production system, large amounts of events may be published concurrently and, e.g., due to network overloads or system downtimes, events may get lost.
- *A real-world event happened, but we have conflicting alternatives for it* (case (c)). For instance, it may happen that a doctor prescribes a specific quantity of drug (e.g., 80 ml),

but there are only doses of 100 ml or 70 ml available. During data cleaning (before running the BI algorithms) the system may detect the mismatch and track it by keeping both options and associating probabilities to them, trying to reflect the doctor's actual intention (see sidebar 2 for details on uncertain data management).



ID	Type	Time	User ID	Drug ID	Quantity	...
237	Prescribe drugs	7/11/09 10.09am	SM889-91	TL99031	<100ml,0.40>	
237	Prescribe drugs	7/11/09 10.10am	SM889-91	TL99031	<70ml,0.60>	
234	Access to prescrip...	7/11/09 10.05am	SM889-91	NULL	NULL	
235	Identify patient..	7/11/09 10.06am	SM889-91	NULL	NULL	
236	Anonymize data	7/11/09 10.08am	SM889-91	NULL	NULL	
372	Approve drug req.	7/11/09 10.21am	SM889-91	RS99387	1000 units	

Figure 2 Typical data quality problems in web-based BI

Uncertain/Probabilistic Data Management

In traditional data management, such as in relational databases, data items either exist or not in the database and data that exist are assumed true (they reflect reality) and correct (there are no errors). On the contrary, in *Probabilistic/Uncertain Data Management* (UDM) this is not taken for granted anymore, and the existence and values of data items are considered probabilistic events. As a consequence, also answering a query over these data becomes probabilistic.

UDM is motivated, among others, by the large number of applications that naturally need to take into consideration uncertainties emerging from the particular domain (e.g., sensor networks and risk analysis) and by the ever increasing speed at which data are automatically generated (e.g., in social networks and real-time systems). In this latter case, noise and incompleteness are ubiquitous because performing cleaning procedures at the same pace at which data is generated is simply impractical. Therefore, the need to manage and process uncertain data is real.

Research on UDM can be grouped into two big areas: *uncertain data modeling*[1] and *query processing on uncertain data*[2]. In the former area, the focus is on the modeling of uncertain data in such a way that data can be kept rich and useful for the applications that use them, while keeping the efficiency in terms of physical data management. The latter area addresses the problem of efficiently querying uncertain data, while providing rich semantics to both the definition of queries and the results coming from the query evaluation. Several tools for uncertain data management have been proposed, for instance, Mystic[3], Trio[4], Orion[5], and MayBMS[6].

References

1. T. Green, V. Tannen. Models for Incomplete and Probabilistic Information. Data Eng. Bull., vol. 29(1), 2006.
2. R. Cheng, D. Kalashnikov, S. Prabhakar. Querying Imprecise Data in Moving Object Environments. IEEE Transactions Knowledge and Data Engineering, vol. 16(9), 2004.
3. C. Re, D. Suciu. Managing Probabilistic Data with MystiQ: The Can-Do, the Could-Do, and the Can't-Do. Springer Verlag/Heidelberg, vol. 5291, 2008, pp. 5-18.
4. O. Benjelloun, A.D. Sarma, C. Hayworth, J. Widom. An Introduction to ULDBs and the Trio System. IEEE Data Eng. Bull., 29(1), 2006.
5. S. Singh, C. Mayfield, S. Mittal, S. Prabhakar, S. Hambrusch, R. Shah. Orion 2.0: native support for uncertain data. ACM SIGMOD'08, Vancouver, Canada, 2008, pp. 1239-1242.
6. L. Antova, T. Jansen, C. Koch, D. Olteanu. Fast and Simple Relational Processing of Uncertain Data. ICDE'08, Cancun, Mexico, 2008, pp. 983-992.

Dealing with Uncertainty in Event Logs

We have seen that the data underlying distributed BI is characterized by a number of data deficiencies, i.e., unconformities between the data we have in the event log and what happened in the real world [5]. The challenge is to deal with deficiencies in a way that allows us to perform meaningful analyses, despite the deficiencies. For this purpose, we propose a notion of *uncertainty* that is composed of three attributes: *trust*, *completeness*, and *accuracy*.

If we model the ideal, i.e., certain, event log as an ordered sequence of events $L=e_i$ (we use the bar to indicate certain data) and an event with k_i data parameters as $e_i=d_{i1},\dots,d_{ik_i}$, the three attributes allow us to deal with the deficiencies describe in Figure 2 as follows:

- Case (a) describes a *meaningless state*, i.e., an event that does not match with any real-world event. Without additional controls, e.g., additional events or certificates from cooperating partners, that specifically aim at identifying this kind of discrepancy, we cannot deal with this situation. What we can do, however, is leveraging on the *trust* we have in the partner that produced the event. That is, we use a trust measure $t_i \in [0..1]$ as an indicator of the probability with which an event registered in the log is true.
- Case (b) shows an *incomplete representation* of the real world, i.e., the lack of an event. This affects the *completeness* of the representation of the real-world process and refers to the whole event log. We know about missing events in the log since we know the models of the processes we monitor and the expected sets of events generated by them. In order to keep track of missing events, we associate a completeness measure $comp \in [0..1]$ to L. If we need to report or run algorithms only on subsets of L, e.g., by analyzing data from a given month or year, $comp$ will refer to the particular subset.
- Case (c) proposes two different *alternatives* for the same real-world event. This leads to a problem with the *accuracy* of the event, since we cannot provide a single description

but only a set of possible alternatives for the event. That is, each event may have a set of "possible worlds" (the alternatives) for its parameters $\{d_{ij1},\dots,d_{ijk_i}\}$, where the index j identifies each alternative. To keep track of the likelihood of each possible world, we associate to each world j a probability p_{ij}, where $j=1 J_i p_{ij}=1$ and J_i is the number of alternatives. Each possible world has its own probability of being the right description of the real world.

In summary, we represent an uncertain event log as a tuple $L=e_i,comp$ (we omit the bar for uncertain data), with e_i being the chronological sequence of uncertain events stemming from all the business processes we want to analyze and $comp$ being the completeness of the log; and we model uncertain events as $e_i=\{d_{ij1},\dots,d_{ijk_i},p_{ij},t_i\}$, where the parameters d_{ijk} are the parameters of the events (e.g., the cost of a product) or event meta-data (e.g., the identifier of the event or its timestamp), p_{ij} are the probabilities of the possible worlds, and t_i is the trust level associated with the event.

In this article we do not focus on how the individual uncertainties for events are computed. We rather tackle the problem of how to represent uncertainty and how to compute with it.

Modeling, Computing and Visualizing Uncertain Key Indicators

KIs are typically associated with specific business processes, e.g., the execution time of a process or the delay between two activities. In order to specify a KI, we therefore imagine having a view over the event log that filters out the events of the process we are interested in and groups them according to executed process instances. The result is a set of event traces $t_l=\{e_{l1},\dots,e_{ln_l}\}$ with n_l being the number of events in each trace. This allows us to obtain KIs in the form of $KI t_l=v$ with $v\in\mathbb{R}$ being the scalar value of the indicator.

In the case of uncertain data, it is no longer appropriate to interpret KIs as simple, scalar values. We propose the idea of *uncertain key indicator* (UKI) as a means to convey to the business analyst both a value for the indicator and the uncertainty associated with it. A UKI can be defined as:

$$UKI t_l=v_m,\ p_m,\ conf,comp$$

The set $\{v_m,\ p_m\}$ represents the possible worlds for the values v_m of the indicator, and p_m is the probability for each of the alternatives. The number of possible worlds depends on the number of possible worlds of the events involved in the computation of the indicator. Specifically, the indicator will have $n J_n$ possible worlds, where J_n refers to the number of possible worlds of the event e_n in the event traces. The parameter $conf\in[0..1]$ represents the confidence we have in the correctness of the computed possible worlds; we compute this confidence by aggregating the trust levels of the events considered by the indicator. The parameter $comp$ is the completeness of the data over which the UKI is computed.

Let us consider the case of the *ARD* (Average Replenishment Duration) indicator, which is computed as the average time in hours needed to replenish drugs in the Ward's drug stock. Figure 3(a) shows an excerpt of the data warehouse we use to store event data for reporting and analysis. Specifically, the table shows the parameters extracted from the event traces of the *drug replenishment process* (a sub-process of the *drug dispensation process*) that are used to compute indicators: each tuple corresponds to an executed process instance. The column *Duration* tells us how many hours each replenishment took; its values are expressed as a set of pairs $\{duration_{ij}, p_{ij}\}$ obtained during ETL and data cleansing. The column *AvgTrust* contains the average of the trust values associated with the events in each trace.

(a) Data warehouse table used to store parameters from uncertain events and to compute UKIs

Process Instance ID	Duration	Par_1	Par_2	…	AvgTrust
72665	{<10.0,0.05>,<15.0,0.90>,<20.0,0.05>}	…	…	…	0.70
72666	{<38.0,1.0>}	…	…	…	0.81
72667	{<10.0,1.0>}	…	…	…	0.45
72669	{<24.5,1.0>}	…	…	…	0.63
72670	{<3.0,0.10>,<4.0,0.80>,<5.0,0.10>}	…	…	…	0.94
72672	{<27.0,1.0>}	…	…	…	0.72
72673	{<15.5, 1.0>}	…	…	…	0.99

Proc. Inst.ID	Duration	Probability
72665	**10.0**	**0.05**
72666	38.0	1.0
72667	10.0	1.0
72669	24.5	1.0
72670	**3.0**	**0.10**
72672	27.0	1.0
72673	15.5	1.0
	= 18.3	**= 0.005**

(b) One of the possible worlds of the input data (out of the available nine we have for the *Duration* parameter)

(c) Possible values (with respective probabilities) of the ARD indicator

ID	Value	Prob.
1	18.3	0.005
2	18.4	0.04
3	18.6	0.005
4	19.0	0.09
5	**19.1**	**0.72**
6	19.3	0.09
7	19.7	0.005
8	19.9	0.04
9	20.0	0.005

Figure 3 Example computation of the ARD indicator

In order to compute ARD, it is necessary to consider individually each possible world that emerges from the data in Figure 3(a). For instance, Figure 3(b) shows one possible world constructed by using the first alternatives for both tuples 72665 and 72670 and a first value for ARD ($v1=avgDuration=$ 18.3) with its probability ($p1=Proc.Inst.IDProbability=$ 0.01). Applying the same logic to the other eight possible worlds allows us to compute all possible worlds of ARD as shown in Figure 3(c). The combination *19.1, 0.72* is the most likely, though the other combinations cannot be excluded.

In order to obtain the overall confidence ($conf$) we have in the indicator as computed in Figure 3, we average the *AvgTrust* values in Figure 3(a), which gives us a value of $conf$=0.75. Finally, in Figure 3(a) we lack two tuples, i.e., process instances. The completeness for ARD is therefore *comp=79=0.78*. Thus, the uncertain representation of ARD is:

ARD = <{<18.3,0.01>,<18.4,0.04>,<18.6,0.01>,<19.0,0.09>,<19.1,0.72>,
<19.3,0.09>,<19.7,0.01>,<19.9,0.04>,<20.0,0.01>};0.75;0.78>

But how do we compute and visualize UKIs in practice? Figure 4(a) shows a simplified version of the *infrastructure* being developed in the context of the MASTER project: process definitions instrumented with compliance annotations feed one or more runtime environments (e.g., operated by different partners) that execute the processes and signal, monitor, and enforce behaviors according to the annotations. Doing so produces events, which we log and periodically load into a data warehouse, where we also check the compliance of executed processes. We store all execution data for reporting (in the reporting dashboard) and analysis (key indicators, root cause analysis, protocol mining).

Figure 4(b) illustrates an excerpt of the *dimensional data warehouse model* [6], showing how we physically store uncertain data and uncertain key indicators in the warehouse. Fact tables are shaded gray, dimension and uncertainty meta-data tables are white. The *Event Fact* table stores the events loaded from the event log. Dimensions that can be used to perform queries and multidimensional analysis are, e.g., *Component Dimension*, *Process Instance Dimension*, and *Date Dimension*. The auxiliary *Attribute Uncertainty* table stores uncertainty meta-data for the attributes of the *Event Fact* table. UKI values are stored in the *Key Indicator Value Fact* and *Key Indicator Values* tables. The former can be joined with the dimension tables it is associated with to support queries and multidimensional analysis. The latter is again an auxiliary table that stores the actual (uncertain) indicator values. The computation of an UKI therefore translates into a set of SQL statements evaluated over the data warehouse.

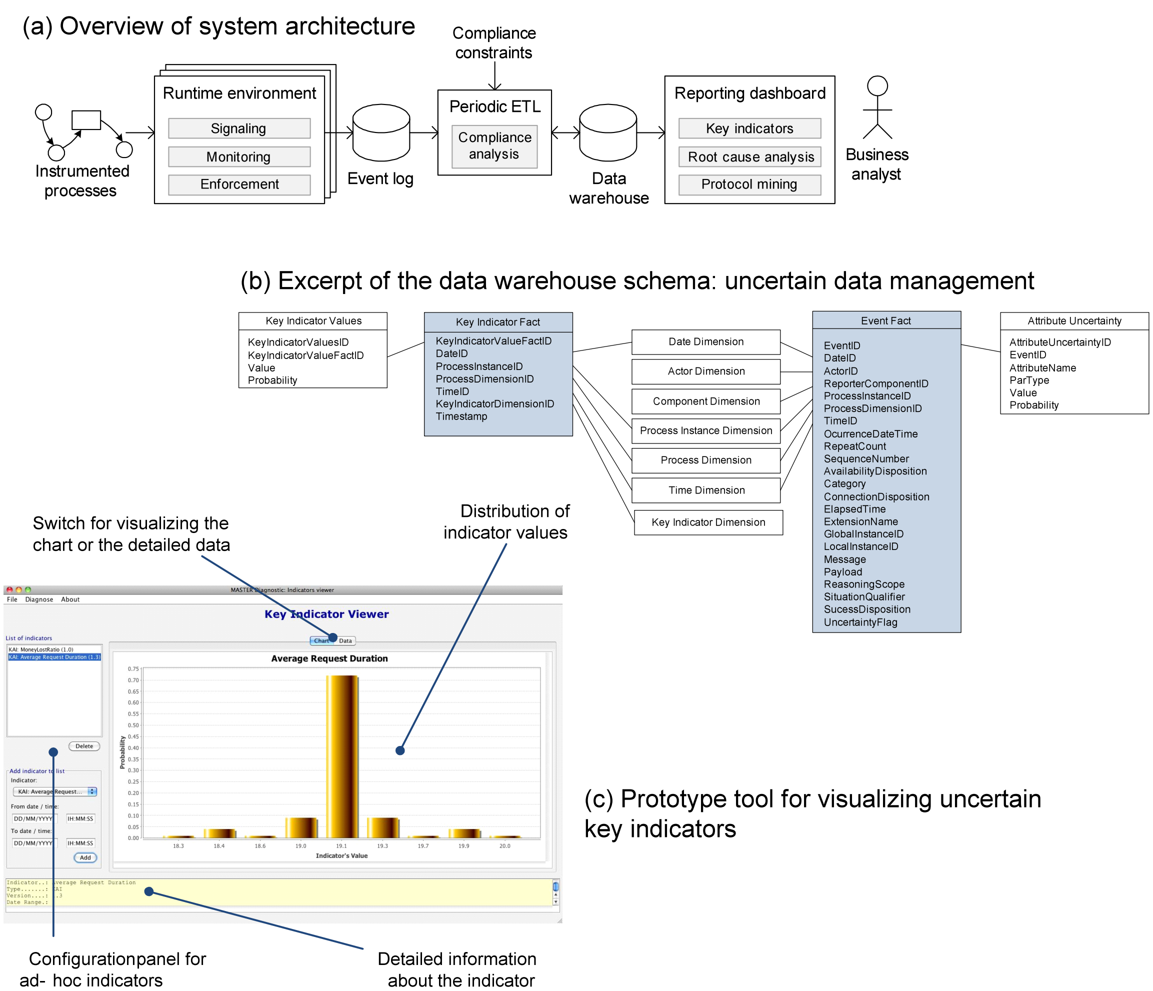


Figure 4 Storing events and computing and visualizing uncertain key indicators

Finally, it is important to properly visualize UKIs in a dashboard, where the important aspects of the monitored business processes can be inspected at a glance. The challenge is to convey the uncertainty of UKIs to the business analysts, while keeping visual metaphors as simple and concise as possible. We approach this problem in a parallel line of research [7][8] where we work on the development of effective reporting dashboards. In Figure 4(c) we show a screenshot of our tool for the visualization of UKIs, which the business analyst can start by drilling down on uncertain indicators in the dashboard. The tool allows the analyst to inspect all uncertainty aspects introduced in this paper (possible worlds, confidence and completeness) and to write ad-hoc queries to better understand the nature of the underlying data.

Conclusion and Outlook

The discussion in this article follows in a way the footsteps of other areas of science, mainly in physics, where uncertainty has become a key ingredient when modeling reality. We believe the same should be done in information engineering, recognizing that our ability to observe reality is not as "precise" as we would like.

The result of the work presented here is a model for representing this imprecision in terms of uncertain events and uncertain indicators, an approach to store uncertainty metadata and compute uncertain indicators, and a tool to communicate uncertainty to users. While this is useful in its own right, the main contribution lies however in providing a basis for uncertainty in BI applications, as this is the branch that is concerned with understanding and analyzing the real world. Indicators are just one (although significant) aspect of BI applications, but what organizations aim at is understanding and improving their processes. On the understanding side, we are now adopting the uncertain data model introduced in this article in the context of process discovery from uncertain data. On the improvement side, we are applying the model to analyze the root causes of compliance violations, specifically working toward techniques like uncertain decision trees and correlation analysis of uncertain data. The computation model presented in this article is the conceptual basis for the outlined research and a first step toward a theory of uncertainty in business intelligence in general.

Acknowledgements: This work was supported by funds from the European Commission (contract N° 216917 for the FP7-ICT-2007-1 project MASTER).

References

1. B. H. Wixom, H. J. Watson An Empirical Investigation of the Factors Affecting Data Warehousing Success. *MIS Quarterly*, Vol. 25, No. 1 (Mar., 2001), pp. 17-41.
2. R.S. Kaplan, D.P. Norton. *The Balanced Scorecard: Translating Strategy into Action*. Harvard Business School Press, USA (1996).
3. M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer, 2007.
4. F. Daniel, F. Casati, T. Palpanas, O. Chayka. Managing Data Quality in Business Intelligence Applications. *QDB'08*, Auckland, New Zealand, 2008.
5. Y. Wand and R.Y. Wang. Anchoring Data Quality Dimensions Ontological Foundations. *Communications of the ACM*, 39(11), November 1996.

6. D. Hollingsworth. *The Workflow Reference Model. TC00-1003*, Workflow Management Coalition, January 1995.
7. C. Rodríguez, F. Daniel, F. Casati, C. Cappiello. Computing Uncertain Key Indicators from Uncertain Data. *ICIQ'09*, Postdam, Germany, 2009.
8. P.Silveira, C.Rodríguez, F.Casati, F. Daniel, V.D'Andrea, C.Worledge, Z.Taheri. On the Design of Compliance Governance Dashboards for Effective Compliance and Audit Management. *NFPSLAM-SOC'09*, Stockholm, Sweden, 2009.

Author Biographies

Carlos Rodríguez is a PhD student at the University of Trento, Italy. He is involved in the European research project MASTER (Managing Assurance, Security and Trust for Services) where he works on compliance assessment by analyzing and mining execution data obtained from business processes. His current research interests include uncertain data management and data mining, with special focus on analysis of business execution data and process discovery. Contact him at crodriguez@disi.unitn.it.

Carlos Rodríguez
Dipartimento di Ingegneria e Scienza dell'Informazione
University of Trento
Via Sommarive 14, I-38123 Povo (TN), Italy
Phone: +39 0461 882092
Fax: +39 0461 882093
E-mail: crodriguez@disi.unitn.it

Florian Daniel is a postdoctoral researcher at the University of Trento, Italy. His main research interests are mashups and user interface composition approaches for the Web, web engineering, quality and privacy in business intelligence applications. He has a PhD in information technology from Politecnico di Milano, Italy. He is actively involved in two European research projects on compliance management: MASTER (Managing Assurance, Security and Trust for Services) and COMPAS (Compliance-driven Models, Languages, and Architectures for Services). Contact him at daniel@disi.unitn.it.

Florian Daniel
Dipartimento di Ingegneria e Scienza dell'Informazione
University of Trento
Via Sommarive 14, I-38123 Povo (TN), Italy
Phone: +39 0461 883780
Fax: +39 0461 882093
E-mail: daniel@disi.unitn.it

Fabio Casati is Professor of Computer Science at the University of Trento. He got his PhD from the Politecnico di Milano and then worked for over 7 years in Hewlett-Packard USA, in the research program on business process intelligence. Fabio has also contributed (as software and data architect) to the development of several HP commercial products and solutions in the area of web services and business process management. In Trento, he is leading or participating to several FP7 projects, is active in many industry-funded projects, both local and international, and has over 20 patents. His passions are now in social informatics, or, informatics at the service of the community. His latest efforts are on prevention of non-communicable diseases, on remote and real-time healthcare, on collaborative programming, and on models for scientific disseminations that can help scientists work in a more efficient way. Contact him at casati@disi.unitn.it.

Fabio Casati
Dipartimento di Ingegneria e Scienza dell'Informazione
University of Trento
Via Sommarive 14, I-38123 Povo (TN), Italy
Phone: +39 0461 882044
Fax: +39 0461 882093
E-mail: casati@disi.unitn.it

Cinzia Cappiello is assistant professor at the Politecnico di Milano. Her research interests regard data and information quality aspects in service-based and Web applications, Web services, and sensor data management. She has a PhD in information technology from Politecnico di Milano. She has published several papers about data and information quality in international journals and conferences. She is also actively participating in the data and information quality community by co-chairing workshops and regularly serving as reviewer for international conferences and journals. A more detailed curriculum vitae and the list of publications can be found at: http://dei.elet.polimi.it/cappiell. Contact her at cappiell@elet.polimi.it

Cinzia Cappiello
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Via Ponzio 34/5, 20133, Milano, Italy
Phone: +39 02 23994014
Fax: +39 02 23993411
E-mail: cappiell@elet.polimi.it