Toward Uncertain Business Intelligence

The Case of Key Indicators

Enterprises widely use decision support systems and, in particular, business intelligence techniques for monitoring and analyzing operations to understand areas in which the business isn't performing well. These tools often aren't suitable in scenarios involving Web-enabled, intercompany cooperation and IT outsourcing, however. The authors analyze how these scenarios impact information quality in business intelligence applications and lead to nontrivial research challenges. They describe the idea of uncertain events and key indicators and present a model to express and store uncertainty and a tool to compute and visualize uncertain key indicators.

Carlos Rodríguez, Florian Daniel, and Fabio Casati
University of Trento, Italy

Cinzia Cappiello
Politecnico di Milano, Italy

The increased use of IT to support business operations, as well as advances in business intelligence (BI) techniques, let enterprises monitor and analyze processes to help them understand which aspects of their business aren't performing well and how to improve. Enterprises have used such techniques for some time in the context of single departments and processes but are extending them to BI applications that integrate data from multiple departments and even multiple companies. Common examples include the now omnipresent Enterprise Data Warehouse,[1] which aggregates process data across departments and geographies; business process outsourcing scenarios, in which a business delegates process execution to other companies; and intercompany cooperation, in which multiple companies share data and processes.

Although BI applications are often complex and comprise multiple kinds of analyses, one widely used metaphor is that of key indicators (KIs),[2] a set of values that summarizes critical business operations' performance. Companies use KIs to detect problems and trigger business decisions.

Despite their importance to business, few researchers have devoted attention to KIs' expressiveness if they're computed from low-quality data or how KIs can communicate possible uncertainties to BI analysts. Even in closed scenarios, many possible sources of uncertainty

 1089-7801/10/$26.00 © 2010 IEEE

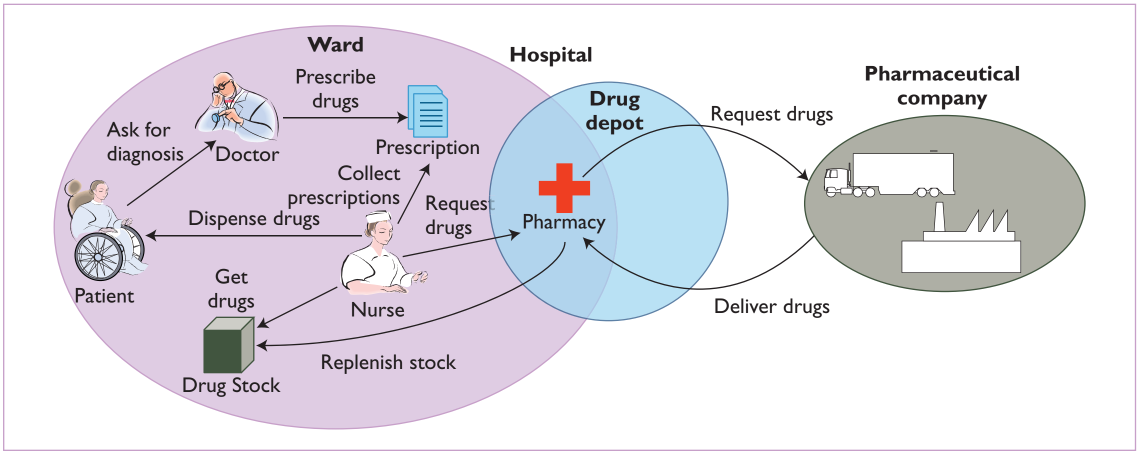


Figure 1. Outpatient drug dispensation in a hospital. This process is executed every time a patient arrives to the hospital's wards and is thus subject to both performance and compliance requirements, oriented to assure the service's quality and compliance with regulations, respectively.

exist in BI applications,[3] and this 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, we can easily predict or detect uncertainty (for instance, a partner doesn't send data on time, or a source has an inherently unreliable data collection method), whereas in others, the problems are occasional and harder to recognize. Here, we aim to understand how to deal with this lack of comprehensive knowledge about organizational business processes and compute meaningful indicators, despite uncertainty in the underlying data.

Key Assurance Indicators in Healthcare

As part of the EU project master (*M*anaging *A*ssurance, *S*ecurity, and *T*rust for Services; www.master-fp7.eu), we're developing diagnostic algorithms to assess and report on compliance, even in the presence of uncertain data. We use so-called key assurance indicators (KAIs) to measure performance against compliance requirements, such as those deriving from a privacy law. We're testing algorithms in collaboration with Hospital San Raffaele (in Milan), which provides the necessary, distributed business context: its outpatient drug-dispensation process. Figure 1 summarizes this process.

The process begins with a patient's visit to a doctor in the hospital's ward. If the patient needs any treatment, the doctor sends an accordant prescription for drugs to the nurse, and the patient can ask that nurse to dispense the drugs. The nurse collects all drug prescriptions and checks whether the necessary quantities are in stock. If yes, he or she can immediately dispense the drugs to the patient. If not, the nurse must issue a drug request to the hospital's pharmacy, which is then responsible for providing the requested drugs. If, in turn, the pharmacy is running out of stock, the personnel in charge issue a request to the pharmaceutical company that provides the drugs. By law, the hospital must guarantee that all patient data are anonymized throughout this process, and the hospital's internal policy states that the pharmacy must replenish drug supplies within two business days. To control, for instance, this latter aspect, the hospital wants to compute a KAI called average replenishment duration (ARD), which lets it monitor the time it takes to refill the ward's drug stock.

From an IT perspective, several Web service-based information systems interact inside a service-oriented architecture (SOA) to support this drug-dispensation process. For instance, Web services exist for issuing drug requests within the hospital's various dependencies, and the pharmaceutical companies the hospital cooperates with accept drug requests through Web service interfaces. To retrieve the data the hospital's BI application has requested, the business process engine generates suitable events during process execution that the IT infrastructure can log and analyze.

Uncertain Events

The drug-dispensation process describes a BI scenario in which a business sources data from multiple cooperating entities and companies.

Trust and Reputation in Web-Based Collaboration

Trust and reputation are concepts studied in different fields, such as economics, sociology, computer science, and biology. In our research, we specifically study trust in the context of business intelligence applications. Although a growing literature exists on the theory and applications of trust and reputation systems, definitions aren't 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;[1,3] reputation is the general opinion about a person, company, or object. So, whereas trust derives from personal and subjective phenomena, we can consider reputation to be 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 for supporting decisions in Internet-based service provisioning. Reputation especially can drive the relationships of individuals and firms in online marketplaces.[4,5] For instance, they might use collaborative filtering systems to judge a party's behavior and assist other parties in deciding whether to start business with that party. A reputation system collects, distributes, and aggregates feedback about participants' past behavior and discourages unfair behavior.[6] The cross analysis of different reputation systems enables us to realize mechanisms and methods for online reputation monitoring and improvement.[7]

References

1. D.H. McKnight and N.L. Chervany, The Meanings of Trust, tech. report MISRC Working Paper Series 96-04, Management Information Systems Research Center, Univ. of Minnesota, 1996.
2. A. Jøsang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support Systems*, vol. 43, no. 2, 2007, pp. 618–644.
3. D. Gambetta, *Trust: Making and Breaking Cooperative Relations*, Basil Blackwell, 1988.
4. M. Fan, Y. Tan, and A.B. Whinston, "Evaluation and Design of Online Cooperative Feedback Mechanisms for Reputation Management," *IEEE Trans. Data and Knowledge Eng.*, vol. 17, no. 2, 2005, pp. 244–254.
5. G. Zacharia and P. Maes, "Collaborative Reputation Mechanisms in Electronic Marketplaces," *Proc. 32nd Hawaii Int'l Conf System Sciences*, IEEE CS Press, 1999, p. 8026.
6. P. Resnick et al., "Reputation Systems," *Comm. ACM*, vol. 43, no. 12, 2000, pp. 45–48.
7. C. Ziegler and M. Skubacz, "Towards Automated Reputation and Brand Monitoring on the Web," *Proc. 2006 IEEE/WIC/ACM Int'l Conf. Web Intelligence*, IEEE CS Press, 2006, pp. 1066–1072.

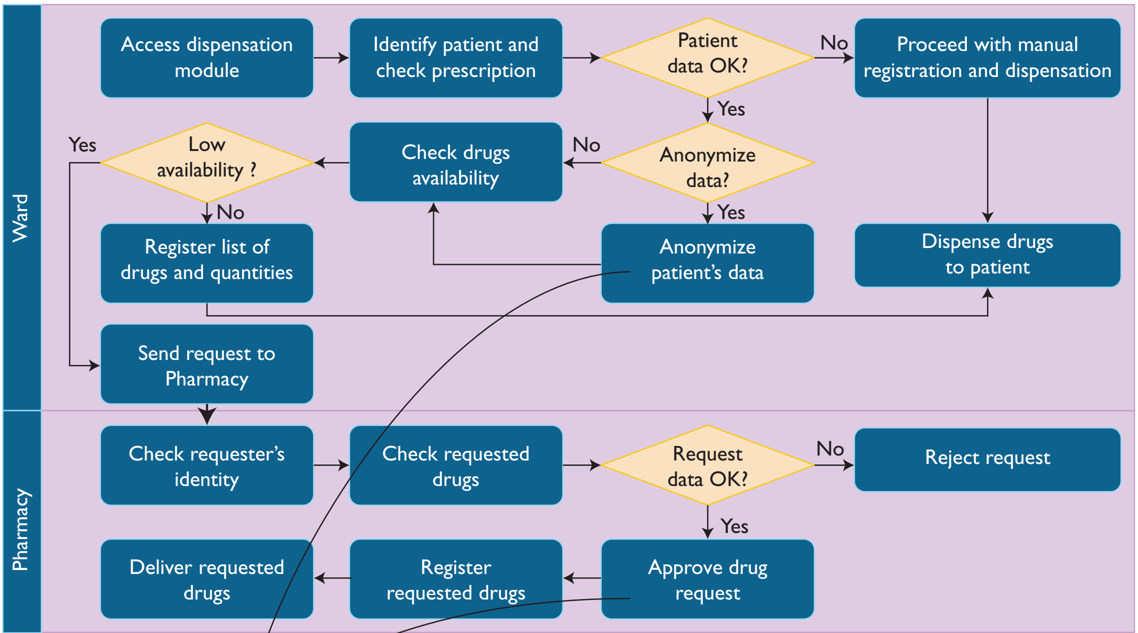
Such a scenario is typically characterized by different levels of visibility into partners' business activities and different levels of trust in the visible data that the business can obtain from each partner .

In the case of cooperative processes (those that span organizational domains,[4] such as the ward, stock management, and the pharmaceutical company), we can distinguish three kinds of business events:

- Internal events stem from activities that are under the company's control (the ward) and consequently are completely visible and trustable.
- Shared events originate in activities shared with the integrated partner (stock management). Depending on the technical solution adopted to implement the cooperative part of the process, the visibility into this partner's internals might be lower than that into the organization's own activities. Similarly, trust in events might be lower.
- External events are part of the partner's internal processes; these events are typically hidden from the company, and we can't analyze them (for example, we don't have access to stock management's internal processes). Similarly, we can associate visibility and trust levels with outsourced processes (the pharmaceutical company's production and shipment of drugs), but both are typically lower than in the cooperative process scenario.

Visibility into shared or outsourced processes typically has structural or organizational roots (for instance, the use of incompatible IT systems or privacy restrictions) that don't often change over time. Trust in partners and the information they provide might instead vary with faster dynamics, such as those based on trust-assessment systems that automatically assess trust values for partners from past interactions (see the "Trust and Reputation in Web-Based Collaboration" sidebar for more details).

Using Web services and a SOA moves cooperative processes to the Web. The consequent reliability problems raise information quality issues as regards collecting the events on which BI algorithms can perform their analyses. In this context, we identify some issues that are strongly related to how the IT infrastructure col-



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	1.000 units	

* We are not sure the real world event happened, though we have an event in the log.
† A real world event happened, but we didn't register it.
‡ We are sure an event happened, but we have doubts about its values. With probability 0.40 the quantity of the drug is 100ml and with probability 0.60 the quantity is 70ml.

Figure 2. Typical data quality problems in Web-based distributed business processes. The figure shows a simplified example of the most important issues we have identified. In real settings, any combination of these issues might arise at any point in the process.

lects events (Figure 2 illustrates this situation):

- Case A. We registered an event in the log, yet we aren't sure the corresponding real-world event really happened. For instance, the system might not be able to successfully anonymize a patient's data due to a failure in the algorithm. If the anonymization component doesn't register this failure properly, we register a wrong anonymization event.
- Case B. A real-world event happened, but we couldn't register it in the log. In a running production system, numerous events might be published by the cooperating partners concurrently (via an Enterprise Service Bus), and due to network overloads or system downtimes, for example, events might get lost.
- Case C. A real-world event happened, but we have conflicting alternatives for it. For instance, a doctor might prescribe a specific quantity of drug (80 ml), but only 100 ml or 70 ml doses are available. During data cleaning (before running the BI algorithms), the system might detect the mismatch and track it by keeping both options and associating probabilities to them, trying to reflect the doctor's actual intent (see the "Uncertain or Probabilistic Data Management" sidebar for details on uncertain data management).

Having identified the kind of business events we want to deal with and the types of uncertainty we might find in them, we next move on

Uncertain or Probabilistic Data Management

In traditional data management, such as 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). In contrast, in probabilistic/uncertain data management (UDM), we don't take this for granted and consider the existence and values of data items to be probabilistic events. Consequently, also answering a query over these data becomes probabilistic.

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

We can group research on UDM into two big areas: *uncertain data modeling*[1] and *query processing on uncertain data*.[2] In the former area, the focus is on modeling uncertain data in such a way that it can be kept rich and useful for applications that use them, while maintaining 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. Researchers have proposed several tools for uncertain data management, such as Mystic,[3] Trio,[4] Orion,[5] and MayBMS.[6] In our work in the main text, we focus on the modeling of and computations with uncertain data.

References

1. T. Green and V. Tannen, "Models for Incomplete and Probabilistic Information," Data Eng. Bulletin, vol. 29, no. 1, 2006 //pages?//.
2. R. Cheng, D. Kalashnikov, and S. Prabhakar, "Querying Imprecise Data in Moving Object Environments, IEEE Trans. Knowledge and Data Eng., vol. 16, no. 9, 2004, pp. 17–24.
3. C. Re and D. Suciu, "Managing Probabilistic Data with MystiQ: The Can-Do, the Could-Do, and the Can't-Do," LNCS 5291, Springer Verlag/Heidelberg, 2008, pp. 5–18.
4. O. Benjelloun et al., "An Introduction to ULDBs and the Trio System," IEEE Data Eng. Bulletin, vol. 29, no. 1, 2006, pp. 953–964.
5. S. Singh et al., "Orion 2.0: Native Support for Uncertain Data," Proc. 2008 Int'l Conf. Management of Data, ACM Press, 2008, pp. 1239–1242.
6. L. Antova et al., "Fast and Simple Relational Processing of Uncertain Data," Proc. 2008 IEEE 24th Int'l Conf. Data Engineering (ICDE 08), IEEE CS Press, 2008, pp. 983–992.

to modeling these events by taking uncertainty into consideration.

Dealing with Uncertainty in Event Logs

We've seen that the data underlying distributed BI are characterized by several data deficiencies — that is, 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 still lets us perform meaningful analyses. For this purpose, we propose a notion of uncertainty that's composed of three attributes: trust, completeness, and accuracy.

If we model the ideal, or certain, event log as an ordered sequence of events $\overline{L} = [\overline{e}_i]$ (the bar indicates certain data) and an event with k_i data parameters as $\overline{e}_i = \langle d_{i1}, ..., d_{ik_i} \rangle$, the three attributes let us deal with the deficiencies Figure 2 describes as follows.

Case A describes a meaningless state — that is, an event that doesn't match with any real-world event. Without additional controls, such as additional events or certificates from cooperating partners, that specifically aim to identify this kind of discrepancy, we can't deal with this situation. What we can do, however, is leverage the trust we have in the partner that produced the event. That is, we use a trust measure $t_i \in [0,1]$ to indicate the probability that an event registered in the log is true.

Case B shows an incomplete representation of the real world — that is, the lack of an event. This affects the completeness of how the real-world process is represented and refers to the whole event log. We know about missing events in the log because we know the models of the processes we monitor and the expected sets of events those processes generate. 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 L's subsets, such as 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 event's accuracy because we can't provide a single description. That is, each event might have a set of "possible worlds" (alternatives) for its parameters $\{d_{ij1}, ..., d_{ijk_i}\}$, where the index j identifies each alternative. To keep track of each possible world's likelihood, we associate to each world j a probability p_{ij}, where $\Sigma_{j=1}^{J_i} p_{ij} = 1$ 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 = $\langle [e_i], \text{comp} \rangle$ (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 log's completeness. We model uncertain events as $e_i = \{\langle d_{ij1},...,d_{ijk_i}, p_{ij}, t_i \rangle\}$, where the parameters d_{ijk} are the event parameters (such as a product's cost) or event metadata (such as an event's identifier or time stamp), p_{ij} are the probabilities of the possible worlds, and t_i is the trust level associated with the event.

In this article, we don't focus on how to compute individual uncertainties for events. Rather, we tackle the problem of how to represent uncertainty and compute with it.

Modeling, Computing, and Visualizing Uncertain Key Indicators

KIs are typically associated with specific business processes, such as these processes' execution time or the delay between two activities. To specify a KI, we therefore imagine having a view over the event log that filters out the events of the process we're interested in and groups them according to executed process instances. The result is a set of event traces $\{t_l\} = \{[e_{l1},...,e_{ln_l}]\}$, where n_l is the number of events in each trace. This lets us obtain KIs in the form of $\overline{KI}(\{\overline{t_l}\}) = v$, where $v \in \mathbb{R}$ is the indicator's scalar value.

With uncertain data, interpreting KIs as simple, scalar values is no longer appropriate. We propose the idea of an uncertain key indicator (UKI) as a means to convey to business analysts both a value for the indicator and the uncertainty associated with it. We can define a UKI as

$$\text{UKI}(\{t_l\}) = \langle \{\langle v_m, p_m \rangle\}, conf, comp \rangle.$$

The set $\{\langle v_m, p_m \rangle\}$ represents the possible worlds for the indicator's values v_m; p_m is the probability for each alternative. The number of possible worlds for the values v_m depends on the number of possible worlds for the events involved in computing the indicator. Specifically, the indicator will have $\Pi_n J_n$ possible worlds, where J_n refers to the number of possible worlds of the event e_n in the event traces.

Instance	Duration	Par	Par	…	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

(a)

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)

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

(c)

Figure 3. Example computation of the average replenishment duration (ARD) indicator. We can see (a) the events generated by a business process (including uncertain events); (b) one of the possible worlds that stem from the uncertain events and that are used for computing the possible values for the corresponding UKI; and (c) the final set of possible values for the (uncertain) ARD indicator.

The parameter conf $\in [0,1]$ represents the confidence we have in the computed possible worlds' correctness; we compute this confidence by aggregating the trust levels of the events the indicator considers. The parameter comp is the completeness of the data over which we compute the UKI.

Let's consider the ARD indicator, which we compute as the average time in hours needed to replenish drugs in the ward's drug stock. Figure 3a 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 subprocess of the drug-dispensation process) that we can use to compute indicators: each tuple corresponds to an executed process instance. The column Duration tells us how many hours each replenishment took; we express its value as a set of pairs $\{\langle \text{duration}_{ij}, p_{ij} \rangle\}$ obtained during extraction-transform-load (ETL) and data cleansing. The column AvgTrust contains the average of

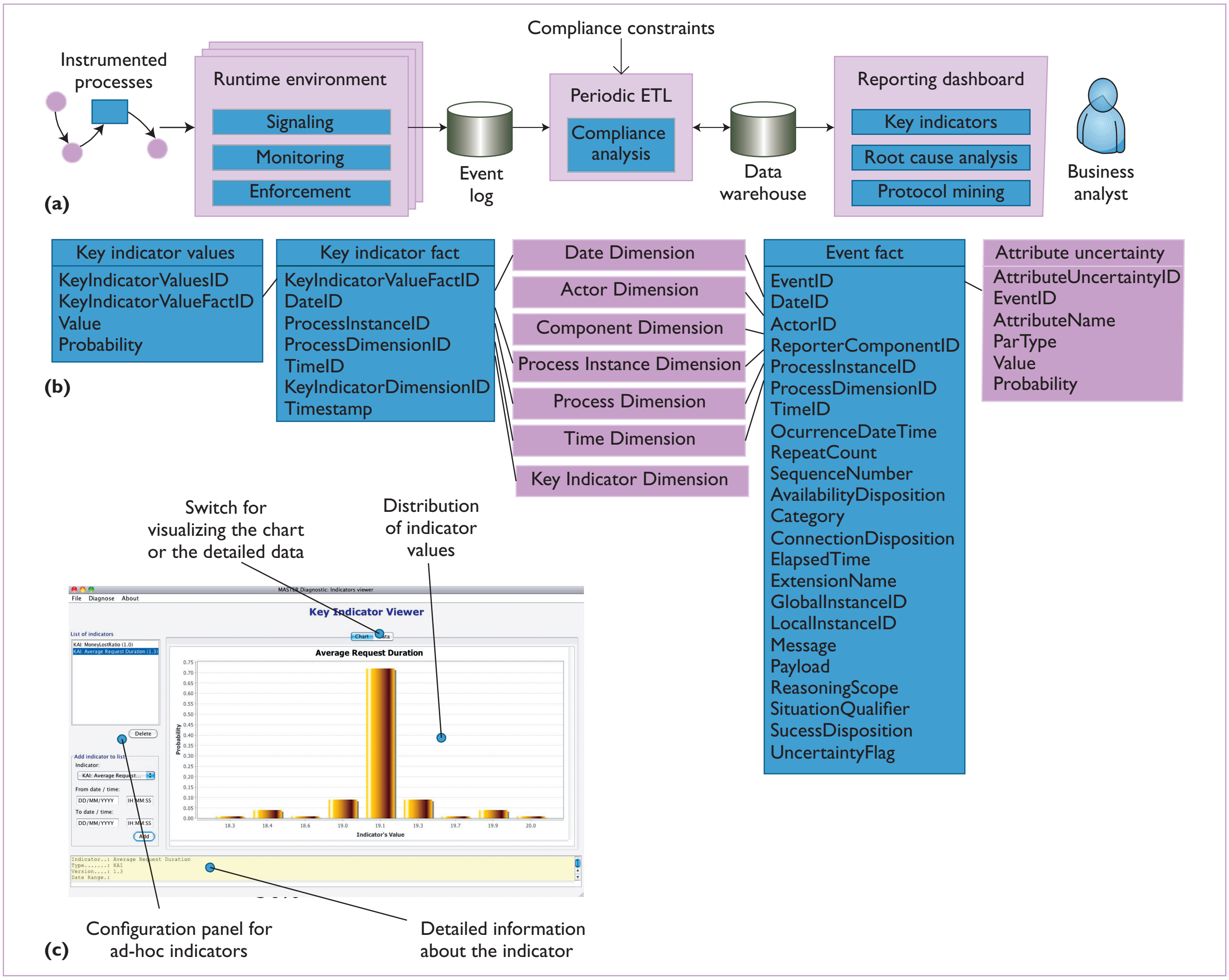


Figure 4. Storing events and computing and visualizing uncertain key indicators. We can see (a) a simplified picture of the architecture underlying this article, (b) an excerpt of the dimensional model used for our data warehouse schema, and (c) a screenshot of the indicator visualizer that's part of our diagnostic infrastructure.

the trust values associated with the events in each trace.

To compute ARD, we must individually consider each possible world that emerges from the data in Figure 3a. For instance, Figure 3b shows one possible world constructed using the first alternatives for both tuples, 72665 and 72670, and a first value for ARD ($v_1 = avg(Duration) = 18.3$) with a probability ($p1 = \Pi_{ProcInstID} Probability = 0.01$). Applying the same logic to the other eight possible worlds lets us compute all possible ARD worlds, as Figure 3c shows. The combination $\langle 19.1, 0.72 \rangle$ is the most likely, although we can't exclude the other combinations.

To obtain the overall confidence (*conf*) we have in the indicator as computed in Figure 3, we average the AvgTrust values in Figure 3a, which gives us a value of $conf = 0.75$. Finally, in Figure 3a, we lack two tuples — that is, process instances. ARD's completeness is therefore comp = 7/9 = 0.78. Thus, ARD's uncertain representation 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 4a 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 (for example, 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 executed processes' compliance. We store all execution data for reporting (in the reporting dashboard) and analysis (key indicators, root cause analysis, and protocol mining).

Figure 4b 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 blue and dimension and uncertainty metadata tables are purple. The event fact table stores the events loaded from the event log. Dimensions that we can use to perform queries and multidimensional analysis are, for example, the component dimension, process instance dimension, and date dimension. The auxiliary attribute uncertainty table stores uncertainty metadata for the event fact table's attributes. UKI values are stored in the key indicator fact and key indicator values tables. We can join the former with the dimension tables it's associated with to support queries and multidimensional analysis. The key indicator values table is again an auxiliary table that stores the actual (uncertain) indicator values. Computing a UKI therefore translates into a set of SQL statements evaluated over the data warehouse.

Finally, we must be able to properly visualize UKIs in a dashboard, where we can inspect the monitored business processes' important aspects at a glance. The challenge is to convey the UKI's uncertainty to business analysts while keeping visual metaphors as simple and concise as possible. We approached this problem in a parallel line of research,[7,8] where we aimed to develop effective reporting dashboards. In Figure 4c, we show a screenshot of our tool for UKIs visualization, which business analysts can employ to drill down on uncertain indicators in the dashboard. The tool lets analysts inspect all uncertainty aspects introduced in this article (possible worlds, confidence, and completeness) and write ad hoc queries to better understand the underlying data's nature.

In a way, the discussion in this article follows in the footsteps of other scientific areas, mainly 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 isn't as precise as we would like.

The result of the work we present 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. Although this is useful in its own right, our main contribution is in providing a basis for uncertainty in BI applications because this branch of research is concerned with understanding and analyzing the real world.

Indicators are just one (although a significant) aspect of BI applications, but what organizations want is to understand and improve their processes. As part of the Master project, on the understanding side, we're now adopting the uncertain data model introduced in this article in the context of process discovery from uncertain data. On the improvement side, we're applying the model to analyze the root causes of compliance violations, specifically working toward techniques such as uncertain decision trees and correlation analysis of uncertain data. Our computation model is the conceptual basis for the outlined research and a first step toward a theory of uncertainty in BI in general.

Acknowledgments

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 and H.J. Watson, "An Empirical Investigation of the Factors Affecting Data Warehousing Success," MIS Quarterly, vol. 25, no. 1, 2001, pp. 17–41.
2. R.S. Kaplan and D.P. Norton, The Balanced Scorecard: Translating Strategy into Action, Harvard Business School Press, 1996.
3. F. Daniel et al., "Managing Data Quality in Business Intelligence Applications," Proc. Int'l Workshop on Quality in Databases (QDB 08), Univ. of Twente, 2008, pp. 133–143.
4. M. Weske, Business Process Management: Concepts, Languages, Architectures, Springer, 2007.
5. Y. Wand and R.Y. Wang, "Anchoring Data Quality

Dimensions Ontological Foundations," Comm. ACM, vol. 39, no. 11, 1996, pp. 86–95.
6. D. Hollingsworth, The Workflow Reference Model TC00-1003, Workflow Management Coalition, Jan. 1995.
7. C. Rodríguez et al., "Computing Uncertain Key Indicators from Uncertain Data," Proc 14th Int'l Conf. Information Quality (ICIQ 09), Univ. of Potsdam/MIT Press, 2009, pp. 106–120.
8. P. Silveira et al., "On the Design of Compliance Governance Dashboards for Effective Compliance and Audit Management," Proc. 3rd Workshop on Non-Functional Properties and Service Level Agreements Management in Service Oriented Computing (NFPSLAM-SOC 09), Springer, to appear in 2010.

Carlos Rodríguez is a PhD student at the University of Trento, Italy. He's 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. Rodríguez has a MSc in informatics engineering from the National University of Asuncion, Paraguay. Contact him at 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. Daniel has a PhD in information technology from Politecnico di Milano, Italy. He's coauthor of the book Engineering Web Applications (Springer, 2009) and co-organizer of the International Workshop on Lightweight Integration on the Web (ComposableWeb). Contact him at daniel@disi.unitn.it.

Fabio Casati is a professor of computer science at the University of Trento, Italy. His research interests are in social informatics and informatics at the service of the community, and his latest efforts are on prevention of noncommunicable diseases, remote and real-time healthcare, collaborative programming, and models for scientific disseminations that can help scientists work in more efficient ways. Casati has a PhD from the Politecnico di Milano. Contact him at casati@disi.unitn.it.

Cinzia Cappiello is an assistant professor at the Politecnico di Milano, Italy. Her research interests regard data and information quality aspects in service-based and Web applications, Web services, and sensor data management. Cappiello has a PhD in information technology from Politecnico di Milano. She cochairs several workshops and regularly serves as reviewer for international conferences and journals in the data and information quality area. Contact her at cappiell@elet.polimi.it; http://home.dei.polimi.it/cappiell.