Evidence-based compliance engineering

Joris Hulstijn

University of Luxembourg, Esch sur Alzette, Luxembourg

Abstract

Most research on compliance checking is about business processes, focusing on the order and presence of activities, roles, and temporal properties. However, legislation demands evidence of legal conditions. An organisation is considered to be compliant to a piece of legislation, when it can demonstrate that the corresponding legal requirements continue to hold for the entire set of cases to which the legislation applies. In this paper we propose a new perspective: *evidence-based compliance engineering*. We sketch how to utilise existing tools for automated theorem proving and natural language understanding, to allow automated verification of legal objectives based on evidence documents. To demonstrate compliance, the system maintains an invariant for the set of cases, that involves legislation, legal requirements, cases and evidence documents. Whenever a change would disrupt this invariant, processes are started to select and analyse documents, update the cases, update the evidence and run the necessary proofs. The compliance status is monitored continuously and can be made visible on a dashboard. The applicability of the approach is illustrated by two examples.

Keywords

compliance checking, evidence, invariant

1. Introduction

Compliance checking involves a lot of administrative work. For example, legislation against money laundering [1, 2, 3] demands that financial institutions must 'know their customer' (KYC). For every transaction, they must verify customer identity, trace their funding, and find the ultimate beneficiary. Solving these issues is complex and many banks fail. For example, Rabobank was fined for failing to uphold customer due diligence [4]. These difficulties in meeting compliance demands are common [5]. Software tools exist for know-your-customer and for anti-money laundering tasks. For example, Computer Assisted Subject Examination and Investigation Tool (CASE*it*), Customer Due Diligence Tool (CDD), tools for name-entity matching, data analysis tools for fine-tuning the suspicious activity detection, and a quick reference guide to track the relevant legislation in various countries [6]. However, each tool only covers part of the investigation task. There is no unifying vision on compliance.

Compare this varied landscape of compliance tools to the academic literature on compliance checking. We observe that most approaches focus on compliance of business processes [7, 8]. Compliance is interpreted as a set of formal properties, expressed in a form of logic, which is then verified for all traces generated by the business process. See for example Governatori and colleagues [9, 10], but also [8, 11] and later [12]. Here is a recent overview [13].

Proceedings of the 17th International Workshop on Value Modelling and Business Ontologies (VMBO 2024), February 26–27, 2024, 's Hertogenbosch, The Netherlands

joris.hulstijn@uni.lu (J. Hulstijn)

In these works, the legal problem of compliance checking – to verify whether the outcome of a process conforms to objectives demanded by law – is reduced to the computer science problem of conformance testing – to verify whether the traces generated by a process specification satisfy formal properties, e.g. [14]. From a computer science point of view, such a reduction is understandable, but from a legal or business point of view, there are several problems:

1. *conformance, not compliance.* Compliance with the law is reduced to conformance to formal properties. These formal properties are defined over the execution traces of a process specification, so they cover the order of activities, constraints on roles or resources, and temporal conditions. Other aspects of compliance management, such as organisational practices, lawfulness of a revenue model, governance, risk and controls and legal evidence, are not immediately covered, e.g. [15].
2. *no interpretation.* In conformance testing, the properties to test are supposed to be specified upfront. The decision which interpretation of the law must be selected, how that translates into formal properties, and how these properties must be measured or monitored, are not taken into account [16, 17, 18]. See also recent NLP approaches [19, 20].
3. *process, not cases.* In the process-based view, the object of compliance is a specific business process. Instead, as the know-your-customer example shows, the object of the law is often a set of cases, representing customers or transactions. These must be shown to comply to requirements, that correspond to the law, e.g. [21]. The process is only a means to that end. Of course, there is also procedural law, e.g. [22], that does put constraints on processes. Still, procedural law is applied to the handling of cases or dossiers.
4. *no evidence* Many administrative processes collect evidence in a dossier, to demonstrate that some legal conditions are met. Such evidence usually takes the form of documents, often in digitised form (proof of income; proof of identity). In the process-based view, no records are stored after automatic verification of a property.
5. *no data analysis.* Compliance checking has benefited from process mining [11], but makes insufficient use of advances in databases, ERP systems (e.g. SAP S4 HANA), and data analytics [23], especially for compliance testing [24]. Processes can assess properties of new or updated cases, but only a data-analysis can verify properties for all current cases.
6. *no risk management.* Compliance is never enough. In interpreting the law and implementing controls, managers must make a trade-off between the costs of compliance and the risk of violation [25]. This is a form of risk management. Assisted by compliance officers, management is ultimately responsible for the residual risks [26], but, the process-based view of compliance reduces it to an operational level.

In this exploratory paper, we propose a new perspective: *evidence-based compliance engineering*, to address some of these problems. The approach centres on legal interpretation and risk management, and does account for legal evidence, and it aims to combine process-based and data-based approaches to compliance management in a unified way. The approach is called 'compliance engineering' by analogy with requirements engineering [27]. The central idea is *requirements traceability* [28, 27]. All (legal) requirements lead to specific properties of a system (forward). Conversely, all system properties are motivated by (legal) requirements (backward).

How to develop an evidence-based approach to compliance engineering?

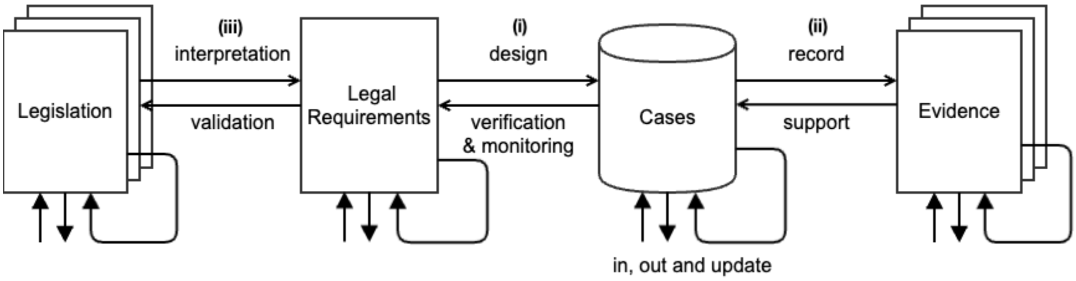

Figure 1: Conceptual model for Evidence-based Compliance Engineering. The model maintains an *invariant*: all cases comply to the legal requirements, which correspond to relevant legislation, and where compliance of cases is supported by evidence.

Consider a conceptual model that involves legislation (text), legal requirements (formal representation), and evidence (documents, traces of verified queries), all linked to a database of cases. Such a model is sketched in Figure 1. The model maintains a set of *invariant* properties:

(i) all cases conform to the legal requirements,
(ii) compliance of cases is supported by evidence, and
(iii) legal requirements correspond to relevant legislation.

Automated reasoning, as well as data base and data analytics queries, allow verification of the legal requirements on the data set that represents all cases. Whenever a change would temporarily disrupt the invariant, for example when laws are changed or when cases are added, semi-automatic processes will start, to restore the invariant. These processes are shown in the diagram as *in, out* or *update* arrows. For example, when a new client is entered, due diligence checks are performed and evidence is collected and analysed, to verify the client conforms to the legal requirements, and thus the organisation conforms to the law. Similarly, when a new law is adopted, the interpretation process is supported by tools for natural language processing, to derive legal requirements that correspond to the legal interpretation chosen. At any time, the compliance status can be automatically verified by automated reasoning. Cases are monitored regularly and their status made visible on a dashboard.

In this exploratory paper we aim to introduce this vision of compliance engineering. Later, the vision must be further worked out, in the form of an architecture, design principles, and a formal semantics, that makes it possible to specify and the invariant properties, and add tools and techniques to verify them. In addition, specific tools and techniques must be selected to 'populate' the various modules in the architecture.

The research method is a form of design science [29]. We present an artefact, a conceptual model for evidence-based compliance. The potential usefulness of the approach is illustrated by two scenarios: (1) know-you-customer, and (2) a building permit.

The remainder of the paper is structured as follows. First, section 2 contains the scenarios. Section 3 provides a theoretical background. Section 4. The paper ends with discussion of the research limitations and suggestions for further research.

2. Scenarios

We analyse two simplified scenarios: (1) know-your-customer, and (2) building permit. Data was collected from publicly available documents. Summaries are given in Table 1 and Table 2.

2.1. Know your customer

In various countries, legislation against anti-money laundering demands that financial institutions must 'know their customer' (KYC) [1, 2, 30] . For every transaction, these organisations must verify customer identity, trace their funding, and find the ultimate beneficiary. Subsequently, they must report any suspicious activities to the regulator. There are two main obligations: (1) to investigate all financial transactions, identify the clients, trace the source of funding, and identify the ultimate beneficiaries of the transaction. Here the main purpose is to establish a proper record of each transaction. (2) On the basis of these records, monitor and report any unusual or suspicious transactions, to the regulator. Note that the unusual nature of a transaction may only be found by comparing series of transactions, by advanced data analysis. Anomaly detection has been used for this purpose [31, 32].

In the Netherlands, under the WVWFT [1, §16], businesses must immediately report any unusual transaction to the Financial Intelligence Unit. What is considered unusual is defined by indicators in an administrative decree. Banks and other financial institutions must interpret these indicators and map them onto their data structures and business processes. For companies that violated this reporting obligation [5], it appears that a mistake in this interpretation and mapping process is more crucial, than a mistake in the execution of the processes.

We can make some observations. First, *different processes* are involved, with different departments (legal; compliance; risk and control; IT). Some processes are more strategic; others are more operational. Second, these processes need *different sources* of data. Sources can be structured in databases, but can also be textual, or contain outcomes of tests or analytics scripts. Third, these processes run at *different time-scales*. For example, in executing a transaction, there is an immediate time-scale, but in monitoring trends, there is a long-term time scale. Summarising, in the know-your-customer case, a single business process is the wrong unit of analysis for compliance purposes.

2.2. Scenario 2. Building permits

In municipalities, a building permit is required before one is allowed to construct or adjust a building. To get a building permit, a procedure must be followed. A permit is only granted, when it is motivated by documents (building plans; safety assessment), and when these documents indicate that specific quality criteria for the building will be met. The purpose of the procedure is to make sure that (1) the municipality has records of all buildings and their construction, and (2) the municipality can maintain quality and safety criteria for all buildings. Buildings must not collapse under weather influence, be resistant to fire, and increasingly, be energy efficient. The specific quality criteria depend on the development plan for the neighborhood: agriculture, housing, shopping, industry (NL: bestemmingsplan). It is easier to regulate permits, then actual construction of buildings. In case of a mistake in a building permit, the building may have to be demolished or altered. This is the main sanction, although fines may also be used.

Table 1
Summary of Scenario 1. Know-your-Customer

law:	WVWFT [1]
domain:	financial
object of compliance:	transactions
legal objectives:	(1) relevant details of all transactions are recorded (2) all and only transactions that are deemed unusual according to WVWFT administrative decrees are identified and reported
business objectives:	(3) uninterrupted flow of transactions (4) confidentiality of client details is respected (5) acceptable compliance risk
(some) requirements:	(1) Ensure records of transaction: identity, bank account, authentication and authorisation of sender; identity, bank account, and authentication of receiver; source of funding, identity of ultimate beneficiary. (2) Maintain indicators of 'unusual'. Develop corresponding criteria and automated tests. Run tests before committing transactions. Hits are reviewed by compliance officer. Regularly review criteria and tests. (3) % investigated transactions < threshold; duration investigation < threshold; other monitoring carried out off-line (4) Access to sensitive data on need-to-know basis only; most data handled by automated processes; search is blocked; cases reviewed by compliance officers are made anonymous (if possible). (5) number and severity of unusual cases < threshold. number and severity of confidentiality breaches < threshold. all known unusual cases found.

Table 2
Summary of Scenario 2. Building Permit

law:	Municipal Environment plan, Zoning plan, Omgevingswet [33]
domain:	construction
object of compliance:	buildings
legal objectives:	(1) relevant details of all buildings are recorded (2) all buildings meet quality and safety criteria
business objectives:	(3) unobstructed flow of housing and renewal projects (4) owners demands are respected
(some) requirements:	(1) Ensure records of identity and residence of owner, architect, and construction company, a detailed blue print and construction plan (2) Ensure that buildings that meet quality and safety criteria get a permit (3) % stalled procedures < threshold; average duration < threshold

In the Netherlands, the legal framework for permits has changed. After years of delays due to late IT systems, the Omgevingswet (Environment Act)[33] tries to bundle many types of permits: buildings, safety, national heritage, etc, and harmonize the procedures of municipalities. To accommodate this change, stakeholders have to invest in new ways of working, and in IT.

Summarizing, for one construction project, often many different procedures are involved, and many types of data are needed (blue prints; safety assessments). These documents must be archived, so they can be retrieved later. Note that the object of compliance is the body of housing, not the permit. The procedure is only a means to maintain quality and safety criteria.

3. Theoretical background

In this section, we provide some theoretical background for the vision shown in Figure 1. Due to lack of space, this overview must necessarily remain sketchy.

3.1. Information integrity

The core of the vision is formed by a database of cases, that is designed to uphold *integrity constraints* [34, 25]. Integrity constraints are conditions on the data or information in a database or information system, that must remain true (invariant), based on their type or meaning. Modern database management systems and ERP systems have built-in tools and techniques to uphold integrity constraints. For example, entry-level controls filter out typing mistakes.

Some integrity constraints are about data types or syntactic constraints. For example, February 30, is not a valid date. Other integrity constraints are semantic. For example, an interest rate is a percentage, so in principle negative interest rates should not exist. Some integrity constraints are relationships. For example, the total income in a year calculated as the sum of all monthly incomes, should equal the income calculated as the sum of incomes generated per division. In accounting, these relational constraints are called reconciliation relations [25].

According to [35], transactions (TPs) on databases are designed and implemented in such a way that (1) they ensure the integrity constraints (IVP) for new input data, and (2) they preserve the integrity constraints (IVP) for existing data. In addition, there are procedural and organizational controls. All transactions must be logged, to allow traceability and thereby foster accountability. Only qualified users are authorized to execute a transaction, further reducing the risk. Similar ideas are built into ERP systems and database management systems [34].

3.2. Traceability in requirements engineering

Another source of inspiration is requirements engineering, in particular *traceability* [36], see also [27]. *Forward traceability* is the property that each part of a system specification leads to a configuration of lower-level components that implement it. This requires maintenance of traceability links in the *how*-direction, which is down the aggregation hierarchy. *Backward traceability* is the property that for each component of a system at some level, it is clear why it was included, in terms of the specification of the level above. This requires links in the *why*-direction, which is up the aggregation hierarchy [27, p. 26].

In our case, each condition to be tested on the cases, is motivated by a legal requirement, which must in turn be motivated by an interpretation of a legal text or clause (backward). Conversely, each interpretation of a legal text or clause, must lead to some legal requirements, which translate into formal conditions, that can be tested on the cases in the database (forward).

3.3. Compliance policies

Compliance management involves more than operational aspects. A large part of the effort goes into motivating strategic choices about compliance [25]. Such choices are laid down into company policies [15]. These policies are what drive subsequent decisions and efforts, for example to invest in IT or in personnel. Part of the strategic choices are driven by legal demands, such as laws and regulations, or by internal controls and measures that are in turn driven by legal demands. Other choices are driven by business objectives, such as cutting costs or adding value to the customer. According to the Lean business process improvement philosophy [37], compliance efforts do not seem to add value to the customer. They are necessary, but do not add a competitive edge.

In the trade-off between compliance and business value, managers, assisted by compliance officers, have to make a specific interpretation of the law (see below) and relate that to the business environment, existing policies, costs of compliance, and business objectives. Typically, a manager will not invest in compliance, unless there is a moral reason to be compliant (solidarity; care for customers), or unless the likelihood of being caught in a violation multiplied by the expected costs (reputation risk; sanctions) are larger than the saved costs.

3.4. Compliance by design

In compliance, there are two roles for information systems: (i) *source*: measure, represent and store data about compliance behaviour, that can be used as a source of evidence, and (ii) *analysis*: analyse various sources of data, and determine whether the company is compliant. In both cases, the reliability of the system itself must be part of the evaluation. For this reason, the system must be designed with compliance in mind: compliance by design. In particular, *internal controls* must be implemented, to ensure reliability of evidence [38, 39]. Here reliability means correctness (all data corresponds to reality) and completeness (all relevant aspects of reality are stored as data). The idea is to get the data directly from the source, without possibilities for manipulation, and to put controls on all subsequent processing (see integrity).

Lu et al [9] present *compliance by design* as essentially a preventative approach. Compliance is built into the business processes that support it. We use the term in a broader sense, including trade-offs between compliance risk and other business objectives. To achieve compliance-by-design, several steps have to be followed [9]. (1) *Maintain a controls directory*: this is a representation of relevant control objectives, and indicators to evaluate them, (2) *Analyse*, compare actual business processes to (indicators of) the control objectives, (3) *Respond*, redesign the process to mitigate the risks deficiencies found, or accept deficiencies, and (4) *Monitor*, regularly test if objectives are still valid. This is similar to the plan-do-check-act cycle [40], used in risk and control. See also [41] for a similar compliance framework. Actually, there are two different time scales: (1) urgent, when a risk is identified, and controls must be designed and implemented, and (2) monitoring, to regularly evaluate the risks and effectiveness of controls.

We also mention an approach called *lawfulness by design* [21]. It suggests design patterns to ensure lawfulness of IT artefacts, specifically data sets. By following design patters that work, the cognitive load on the developers of systems can be reduced, and can be used for other tasks.

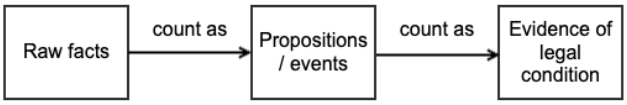

Figure 2: From raw data (source), to cleaned up information (propositions and events), to evidence of a legal condition (decision)

3.5. Legal interpretation

Much work on legal informatics (e.g. [42]), wrongly assumes that the law is captured in a single document, that can be translated into a formal representation, to be checked automatically. In fact, there are many legal documents (e.g. national law, administrative degrees, policies), and per document, many different interpretations (e.g strict versus lenient), that will lead to different implementations [16][17]. To decide about these implementations, it is crucial to maintain an intermediate (formal) representation for each separate interpretation, so different interpretations can be analysed and compared. For example, which alternative is cheaper?

The law is often ambiguous [18]. This is deliberate. The law must be future-proof and not be written for a specific technology. This is called the *open texture of the law* [43]. Especially terms that express a valuation, are subject to debate. For example, in the GDPR art. 5.1(f), what is meant by 'appropriate security' crucially depends on the context: on the type of data and on purpose for collecting the data. Black [44] compares the debate about contested terms to *regulatory conversations* among stakeholders, that settle what is considered acceptable.

3.6. Legal rules and evidence

Data has to be aggregated, cleaned and interpreted, before it has meaning as information. Furthermore, information provides evidence of some legal condition that is relevant to some decision (Figure 2). For example, a unique footprint in the mud means that a suspect was present at the crime scene, which is enough evidence to hold the suspect in custody. In this diagram, the meaning relationship is 'counts-as' [45], see also [46]. Such counts-as rules are known as constitutional rules, which include legal terms and definitions. In a sense, these rules specify what conditions constitute (generate) the institutional facts. Constitutive rules are contrasted with regulative rules, which specify obligations and permissions, and possibly sanctions in case of a violation. Regulative rules use the terms and conditions, specified in constitutional rules.

A useful perspective on evidence can be found in assertion-based auditing [47]. Here, an audit dossier contains the outcomes of various tests, or assertions, that together make up the truth of an audit objective. In a similar way, we aim to add assertions to the evidence dossier. Whenever cases are updated, the assertions have to be re-tested.

3.7. Continuous assurance

Prevention is not enough. No organisation is ever fully compliant, and therefore needs to run regular tests, to find any remaining violations and solve issues. This is the task of compliance monitoring. Our compliance engineering philosophy is inspired by ideas from continuous control monitoring and continuous auditing [48, 49, 24].

Kocken and Hulstijn [50] show that it is possible in principle to provide a continuous assurance service, on the basis of a continuous auditing system (monitoring specific properties of the data stream) combined with a continuous control monitoring system (monitoring operational effectiveness of internal controls), which both feed into a dashboard for showing the current status. Finally, a workflow triggered by a request for a written assurance document, summarises the last months of data in the platform, for those clients who need written proof.

3.8. Data analytics and data mining

Data mining has been used extensively in fraud detection and compliance monitoring [51, 52]. Fraud detection is an analytic task, similar to classification. Based on the attributes of a case, a case is classified as 'fraud' or 'non-fraud'. Monitoring can be seen as repeated detection. A classic problem is to distinguish violations from exceptional cases [53]. So in practice, suspected violations must first be verified by a human expert ('trained eye').

Recent approaches to fraud detection, identify outliers or anomalies in the data, as indicators of fraud [32]. Data is clustered based on similarity. Outliers indicate potential fraud. These techniques have been tested in an audit context [31]. Anomaly detection is a form of unsupervised learning, which removes the need for human experts to annotate data up front. However, suspected cases must still be verified. So, in a first phase unsupervised techniques select potential fraud cases, which are then filtered down in a second phase, by supervised techniques, trained for a specific application domain [32]. The remaining suspicious cases are verified by a compliance officer. So these techniques must be deployed as part of an integrated system in which the distribution of human effort and data mining are mutually optimised. Effectiveness of such tools must be frequently verified, as part of organisational learning.

3.9. Natural language processing

As compliance checking is essentially a legal activity, it is no surprise that documents and texts play an important role. Natural Language Processing is a huge area, that has managed to automate basic linguistics tasks: part-of-speech tagging (segment meaningful chunks in text), parsing (translate sentences into a meaning representation), named entity recognition (e.g. to identify parties in a contract), co-reference resolution (e.g. to identify cross references between legal clauses), topic modelling (determine what a text is about) and sentiment analysis (determine the attitude conveyed by a text). These tasks ca be combined into useful tools to support legal reasoning, including compliance checking [53].

In our vision, there are three functionalities that can be supported by natural language processing. First, the interpretation task can be partly automated [16], so that various versions of a law or policy can be converted into a meaning representation, which can then be compared, simulated and tested against models of operational systems and business processes, to determine which one is more efficient or more secure, for example. Second, in all kinds of monitoring and verification tasks, NLP techniques can analyse unstructured data that involves text elements. For example, named entity recognition can trace the ultimate beneficiary of a financial transaction, as part of the know-your-customer case. Here, the same caveat holds as above: such components can only predict suspected cases; they still need to be verified. Third, in many administrative tasks, evidence documents can be parsed and tested, if they ‘count as’ the right type of evidence. For example, identification documents can be scanned for name and social security number.

Recently, NLP approaches to compliance checking have been proposed, that explicitly address differences in legal interpretation. However, these systems do not require an intermediate meaning representation, removing also the need to maintain such representations, but instead identify changes and differences in the legal texts themselves, before being mapped onto the controls and business processes in which they apply [20, 19].

3.10. Data quality

A general problem in compliance checking, is that one depends on the *data quality*. Data for making compliance decisions is often missing or wrong. For that reason, all automated data-driven decisions must be cross-verified, with the subject or with an expert [53].

This went terribly wrong in the RoboDebt case [54], in Australia. Here, a government information system was used to calculate and collect the ‘debt’ of citizens, meaning the amount of social security benefits, that citizens had received but were not entitled to. When accurate data about a citizen’s income was missing, the debt was calculated based on averages. Note that people with low income, who need social benefits, often also have an unstable income. So an average over the previous period, is not a good way to estimate an income. In a large number of cases this resulted in huge debts that were not justified, leading to financial and social harm for the people involved. This situation was made worse by the reluctance of officials to handle complaints. This illustrates the importance of regular checks, and of allowing feedback [55].

Note in this respect, that many compliance applications assume *negation-as-failure*: absence of data or evidence means that the property is considered to be false. For example, in the building permit case, if no evidence of a safety assessment is recorded, officials may act on the assumption that no assessment was done. This assumption may only be made, if all cases go through a strict entry process, that ensures completeness (see integrity). In practice however, omissions and mistakes are made, so decision making must be made robust to uncertain or incomplete data. That is why frequent monitoring and data analysis remains necessary [56].

4. Towards an architecture

We will start with a general vision. After that, we will outline the various components, and finally, detail how these components can be connected.

Take the two views that we have discussed: the *data view* and the *process view*. The data view looks at evidence of states of affairs being compliant. The process view looks at newly created states of affairs or at transactions, which update those states of affairs. We believe that these two views can be combined into a unified whole, if we make use of an *invariant property* [57]. Compliance, like safety or security, is a property that must always hold. So compliance is specified as an invariant property. Changes or incidents threaten to disrupt compliance. So that triggers a research question: can we ensure by a suitable architecture (system and procedures) that a company remains demonstrably compliant?

Consider the diagram in Figure 1 again. We start from the set of cases. Now suppose that the organisation is actually compliant: they have evidence to demonstrate that all cases in the

Table 3
Components

	Components	Purpose
1.	database of cases	store and retrieve properties of cases, integrity constraints, control mechanisms to maintain integrity constraints
2.	evidence repository	store and retrieve evidence, verify validity of evidence, summarise evidence documents
3.	requirements repository	store and retrieve requirement specifications, trace dependencies
4.	legislation repository	store and retrieve legal sources, trace dependencies

Table 4
Links between components (arrows in Figure 1) and useful compliance tools to play that role

	Links	Tools
1-2	interpretation	NLP, search, indexing, RE tools
2-1	validation	RE tools
2-3	design	compliance-by-design, process mining
3-2	verification	DBMS, DB query, data mining
	monitoring	continuous audit, anomaly detection
3-4	record	archiving, indexing, search
4-3	support	NLP, automated verification

set satisfy the legal requirements. So the invariant property, for all cases legal compliance is demonstrated by evidence, is true. This can be broken down into three sub-properties (page 3): (i) all cases conform to the legal requirements, (ii) compliance of cases is supported by evidence, and (iii) legal requirements correspond to relevant legislation.

What components are needed to ensure these properties? A summary is given in Table 3.

How can such invariant properties be verified and demonstrated? We look at the six arrows in the diagram. (i) To verify compliance, run a set of queries on the cases, that corresponds to the legal requirements (verify and monitor, right to left). The database must be designed in such a way, that it has attributes than can answer such queries (design, left to right). (ii) Moreover, for all claims or assertions stored in the case database, there must be a valid reference to a piece of evidence to support it: a document, a credential, or the outcome of query (support, right-to-left). Conversely, whenever a claim or assertion is tested, either as part of the regular monitoring or as part of processes that handle transactions or changes, outcomes of such tests are recorded (record, left-to-right). (iii) Analogous to forward traceability, for all relevant interpretations of legislation, there must be one or more requirements, that represent them (interpretation, left-to-right). Conversely, for all requirements in the repository, there most be a preferred interpretation of a legal source, that motivates it (validation, right to left).

Now consider all the possible ways in which this invariant can be (temporarily) breached. For example, cases may be added, updated or deleted; evidence may be lost; data may be wrong or incomplete; laws may change, or be re-interpreted, and the IT infrastructure in which these processes is maintained, may change too. For all these potential threats to the invariant property of compliance, adequate business processes must be designed, that will restore the invariant.

In the diagram, these potential changes are shown by three arrows labelled *in*, *out* and *update*, inspired by system input, output and change. In business process management, one often refers to the four basic operations of persistent storage: *create*, *read*, *update* and *delete* (CRUD). They have a similar role. Ideally, read would not seem to alter the invariant, but that is wrong, if we allow for uncertain or incomplete data. Suppose we do a random check, and it appears data is added or removed. In that case, a read will change the outcome.

5. Conclusions

In this paper we have presented a more ambitious and legally more plausible conceptual model for compliance: *evidence-based compliance engineering*. By contrast to the dominant view in information systems research, compliance is not predominantly about business processes, but rather it is both about processes *and* about data. Processes collect and update the data that counts as evidence of continued compliance, seen as an invariant property.

This may be called the *dual nature of compliance*: one pertaining to continued adherence to the rules, as shown by data, and the other to the enforced following of legal procedures.

Such an evidence-based conceptual model of compliance, is more natural for legal experts. It would potentially take trade-offs between compliance and business objectives, into account. It would also allow for differences in legal interpretation, using NLP tools. It allows for the advances in data analytics and data mining to be deployed. If done well, frequent analysis and monitoring should make it robust for incomplete and uncertain data.

So far, this is only a vision. A lot of work remains to be done. First, the vision must be developed into an enterprise architecture, consisting of components and interfaces. Given a proper semantics of the functionality of the components in terms of assertions (Boolean statements about compliance), and a semantics of the interfaces in terms of transactions, it must in principle be possible to demonstrate, that the invariant can be maintained, when all components continue to function, and when all in, out and update processes, continue to perform. Second, the architecture must be populated with real tools and techniques, that can be used to perform the functionality on the arrows. Here, requirements engineering may be used to pick the best solution for a given component. Third, the usefulness and adequacy may then be demonstrated, for one or more real-life cases, such as the scenarios discussed.

The vision is promising, but there are also various challenges: *data quality* (wrong, missing or uncertain data), *legal interpretation* (comparing various representations), and *trade-offs* (balancing compliance and business objectives). Each of these challenges will also lead to fruitful research directions.

References

[1] T. K. der Staten Generaal, Wet ter voorkoming van witwassen en het financieren van terrorisme (Wwft), Technical Report, Koninkrijk der Nederlanden, 2008.

[2] Parliament, The Money Laundering and Terrorist Financing (Amendment) (No. 2), Technical Report No. 860, United Kingdom, 2022.

[3] FinCEN, Beneficial Ownership Information Reporting Requirements, Technical Report, Financial Crimes Enforcement Network, US Treasury, 2022.

[4] Reuters, Rabobank faces punishment over customer anti-money-laundering checks, 2021.

[5] A. Merz, ‘it could have been us’. peer responses to money-laundering violations in the dutch banking industry, Crime, Law and Social Change (2023). doi:https://doi.org/10.1007/s10611-023-10120-y.

[6] Pwc, Using the right tools for anti-money laundering compliance, 2023. URL: https://www.pwc.com/us/en/industries/financial-services/financial-crimes/anti-money-laundering/compliance-tools.html.

[7] M. Dumas, W. van der Aalst, H. M. ter Hofstede Arthur, Process-Aware Information Systems, John Wiley and Sons., 2005.

[8] A. F. S. A. Elgammal, O. Türetken, W. J. A. M. van den Heuvel, M. Papazoglou, Formalizing and applying compliance patterns for business process compliance., Software and Systems Modeling 15 (2014) 119–146.

[9] R. Lu, S. Sadiq, G. Governatori, Measurement of Compliance Distance in Business Work Practice, Information Systems Management 25 (2009) 344–355.

[10] G. Governatori, S. Sadiq, The journey to business process compliance, in: Handbook of Research on Business Process Management, IGI Global, 2009, pp. 426–445.

[11] L. T. Ly, F. M. Maggi, M. Montali, S. Rinderle-Ma, W. M. P. v. d. Aalst, Compliance monitoring in business processes: Functionalities, application, and tool-support, Information Systems 54 (2015) 209–234.

[12] M. Hashmi, G. Governatori, H.-P. Lam, M. T. Wynn, Are we done with business process compliance: state of the art and challenges ahead, Knowledge and Information Systems 57 (2018) 79–133. doi:10.1007/s10115-017-1142-1.

[13] J. Carmona, B. van Dongen, M. Weidlich, Conformance checking: Foundations, milestones and challenges, in: W. M. P. van der Aalst, J. Carmona (Eds.), Process Mining Handbook, LNBIP 448, Springer, 2022, p. 155–190.

[14] S. C. Tosatto, G. Governatori, P. Kelsen, Business process regulatory compliance is hard, IEEE Transactions on Service Computing 8 (2014) 958 – 970.

[15] M. Eggert, Compliance Management in Financial Industries, Springer, 2014.

[16] G. Boella, M. Janssen, J. Hulstijn, L. Humphreys, L. van der Torre, Managing Legal Interpretation in Regulatory Compliance, in: B. Verheij (Ed.), Proceedings of the XIV-th International Conference on Artificial Intelligence and Law (ICAIL 2013), ACM, Rome, 2013, pp. 23–32.

[17] S. Ghanavati, J. Hulstijn, Impact of legal interpretation in business process compliance, 2015.

[18] A. K. Massey, E. Holtgrefe, S. Ghanavati, Modeling regulatory ambiguities for requirements analysis, in: H. C. Mayr, et al. (Eds.), Proceedings of the 36th International Conference on Conceptual Modeling (ER 2017), LNCS 10650, Springer, 2017, pp. 231–238.

[19] H. Mustroph, M. Barrientos, K. Winter, S. Rinderle-Ma, Verifying resource compliance requirements from natural language text over event logs, in: Business Process Management (BPM 2023), Utrecht, The Netherlands, LNCS14159, Springer, 2023, pp. 249–265.

[20] M. Barrientos, K. Winter, J. Mangler, S. Rinderle-Ma, Verification of quantitative temporal compliance requirements in process descriptions over event logs, in: Advanced Information Systems Engineering, volume 13901 LNCS, 2023, p. 417 – 433.

[21] M. S. Ernestine Dickhaut, Andreas Janson, J. M. Leimeister, Lawfulness by design –

development and evaluation of lawful design patterns to consider legal requirements, European Journal of Information Systems 0 (2023) 1–28.

[22] U. Parliament, Criminal procedure and investigations act 1996, 1996.

[23] H. Chen, R. H. L. Chiang, V. C. Storey, Business intelligence and analytics: From big data to big impact, MIS Quarterly 36 (2012) 1165 – 1188.

[24] F. Huang, W. G. No, M. A. Vasarhelyi, Z. Yan, Audit data analytics, machine learning, and full population testing, The Journal of Finance and Data Science 8 (2022) 138–144.

[25] R. Christiaanse, J. Hulstijn, Control automation to reduce costs of control, International Journal of Information System Modeling and Design 4 (2013) 27 – 47.

[26] IIA, The three lines of defense in effective risk management and control, 2013.

[27] R. J. Wieringa, Requirements Engineering: Frameworks for Understanding, Wiley, 1996.

[28] J. D. Palmer, Traceability, in: R. H. Thayer, M. Dorfman (Eds.), Software Requirements Engineering, IEEE Computer Society Press, 1997, p. 364–374.

[29] R. J. Wieringa, Design science methodology for information systems and software engineering, Springer Verlag, London, 2014.

[30] U. K. Parliament, Proceeds of crime act, 2002.

[31] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, B. Reimer, Detection of anomalies in large scale accounting data using deep autoencoder networks, CoRR abs/1709.05254 (2017).

[32] T. Pourhabibi, K.-L. Ong, B. H. Kam, Y. L. Boo, Fraud detection: A systematic literature review of graph-based anomaly detection approaches, Decision Support Systems 133 (2020) 113303. URL: https://www.sciencedirect.com/science/article/pii/S0167923620300580. doi:https://doi.org/10.1016/j.dss.2020.113303.

[33] T. K. der Staten Generaal, Omgevingswet 2016, 2016.

[34] P. W. P. J. Grefen, P. M. G. Apers, Integrity control in relational database systems- an overview, Data and Knowledge Engineering 10 (1993) 187–223.

[35] D. D. Clark, D. R. Wilson, A comparison of commercial and military computer security policies, in: IEEE Symposium on Security and Privacy, IEEE, 1987, pp. 184–194.

[36] A. Egyed, P. Grbacher, Supporting software understanding with automated requirements traceability, International Journal of Software Engineering and Knowledge Engineering 15 (2005) 783–810.

[37] J. P. Womack, D. T. Jones, Lean Thinking: Banish Waste and Create Wealth in Your Corporation, Free Press, 1996.

[38] COSO, Internal Control - Integrated Framework, Technical Report, Committee of Sponsoring Organizations of the Treadway Commission, 1992.

[39] M. B. Romney, P. J. Steinbart, Accounting Information Systems, 14th ed., Pearson Education, 2018.

[40] W. E. Deming, Out of the Crisis, MIT Center for Advanced Engineering Study, 1986.

[41] J. Hulstijn, Compliance by design: Het inbouwen van regelgeving in bedrijfsprocessen en informatiesystemen, RegelMaat 27 (2012) 88–100.

[42] T. Breaux, A. Anton, Analyzing regulatory rules for privacy and security requirements, IEEE Transactions on Software Engineering 34 (2008) 5–20.

[43] H. L. A. Hart, The Concept of Law, Clarendon Press, Oxford, 1961.

[44] J. Black, Regulatory conversations, Journal of Law and Society 29 (2002) 163–196.

[45] J. R. Searle, The Construction of Social Reality, The Free Press, 1995.

[46] A. J. I. Jones, M. Sergot, A formal characterisation of institutionalised power, Journal of the Interest Group in Pure and Applied Logic 3 (1996) 427–443.

[47] D. A. Leslie, S. J. Aldersley, D. J. Cockburn, C. J. Reiter, An Assertion Based Approach to Auditing, University of Kansas, Kansas, 1986, pp. 31–64.

[48] M. Alles, G. Brennan, A. Kogan, M. A. Vasarhelyi, Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at Siemens, International Journal of Accounting Information Systems 7 (2006) 137–161.

[49] A. Kogan, M. G. Alles, M. A. Vasarhelyi, Design and evaluation of a continuous data level auditing system., Auditing: A Journal of Practice and Theory 33 (2014) 221–245.

[50] J. Kocken, J. Hulstijn, Providing Continuous Assurance, in: H. Weigand (Ed.), Proceedings of the 11th International Workshop on Value Modeling and Business Ontologies (VMBO 2017), Luxembourg Institute of Science and Technology (LIST), Luxembourg, 2017.

[51] E. Ngai, Y. Hu, Y. Wong, Y. Chen, X. Sun, The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature, Decision Support Systems (2011) 559–569.

[52] B. Baesens, S. Höppner, T. Verdonck, Data engineering for fraud detection, Decision Support Systems 150 (2021) 113492.

[53] K. D. Ashley, Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age, Cambridge University Press, 2017.

[54] C. Holmes, Report of the Royal Commission into the Robodebt Scheme, Report, Commonwealth of Australia, 2023.

[55] K. Orr, Data quality and systems theory, Communications of the ACM 41 (1988) 66–71.

[56] L. P. English, Improving Data Warehouse and Business Information Quality, John Wiley & Sons, Inc., New York, 1999.

[57] L. Lamport, A new approach to proving the correctness of multiprocess programs, ACM Transactions on Programming Languages and Systems 1 (1977) 84–97.