End-to-end, no-code business process compliance framework for the banking industry

Nigel Adams[1,*], Adriano Augusto[1], Michael Davern[1] and Marcello La Rosa[1]

[1]University of Melbourne, Australia

Abstract

The aim of business process compliance (BPC) is to ensure that business processes are executed in accordance with a prescribed set of rules. In practice, the evidence would suggest that achieving this goal is challenging. Penalties and subsequent remediation costs in the Australian banking industry amounted to over A$10bn between 2017 and 2022. Much of the BPC research has focused on developing methods and languages to extract, interpret and formalize compliance requirements to be checked against business process models at design time. However, regulatory reports published in relation to recent BPC-related events in the Australian banking industry provide insights that suggest some of the underlying research challenges may be addressed by re-framing how to source, formalize, and evaluate compliance requirements. Leveraging previous research results and existing process mining solutions, we designed a comprehensive, end-to-end, "no-code" Compliance Center Framework that addresses industry needs, in particular the banking industry, as well as BPC research gaps. We implemented said framework on top of the Apromore process intelligence platform and evaluated it by assessing feedback from senior industry executives and two industry focus groups, demonstrating that our solution has the potential to simplify, consolidate, and support BPC in the banking industry and beyond.

Keywords

Business process compliance, process mining, banking

1. Introduction

The aim of business process compliance (BPC) is to ensure that business processes are executed in accordance with a prescribed set of rules or norms [1]. The evidence would suggest that this is challenging in practice. Between 2017 and 2022, Australian regulators issued two of the four major domestic banks with penalties exceeding A$2bn for breaching the AML/CTF Act[1] [2, 3], a Royal Commission was held into misconduct in the industry [4], and the banks spent more than A$8bn on remediation [5]. The study in [5] identified 23 factors that underpin the challenges faced by banking industry practitioners. The factors fall into three broad categories: i) the extent of complex, frequently changing, compliance requirements, ii) impenetrable process spaghetti, and iii) significant organizational barriers to implementing a sustainable case for change.

Academic interest in the field of BPC traces its roots to corporate scandals at organizations such as Enron, HIH, AIG, and Société Générale and the subsequent legislative changes (e.g., Dodd-Frank, Sarbanes-Oxley) at the turn of the millennium [6]. While there are studies that aim to support different strategies over the BPC lifecycle (i.e., design time, run time, and post execution), much of the research is focused on extracting, interpreting, and formalizing compliance requirements to be checked against a process model at design time – commonly known as "compliance by design" [1]. Progress has been made, but the research community has also recognized that there are many challenges associated with BPC [7]. For example, i) fully automating BPC may be beyond reach [8]; ii) identifying, expressing and reasoning about natural language compliance requirements is complex [9]; iii) not all compliance requirements can be evaluated at design time [10]; and iv) some of the proposed solutions demand a

PLC - Processes, Laws and Compliance workshop, in conjunction with ICPM 2024, October 14, 2024, Lyngby, Denmark

*Corresponding author.

naadam@student.unimelb.edu.au (N. Adams)

0000-0002-1187-3099 (N. Adams); 0000-0001-7970-5246 (A. Augusto); 0000-0002-9572-812X (M. Davern); 0000-0001-9568-4035 (M. L. Rosa)

[1]Anti-Money Laundering and Counter-Terrorism Financing Act

level of technical expertise unlikely to be found in a commercial setting [11].

Despite the extent of the academic research and a technology sector targeting regulatory monitoring, reporting, and compliance problems (RegTech), a comprehensive, "bundling" of interoperable BPC approaches to support compliance digitization is yet to be developed [12] and the plethora of alternative solutions, e.g., the logic choice for the formalism, is potentially confusing for the non-technical, business user [13]. Conformance checking provides a good example. It is covered in the BPC literature as a compliance auditing approach that can also be applied at run time [7], but not at design time [14]. It (traditionally) checks an event log against a process model [14], but it does not verify whether the process model adequately reflects the necessary compliance requirements. There are also known limitations with conformance checking such as a focus on the control perspective and potentially long execution times [15]. Together, this puts the onus on the business user to determine the right BPC solution for the specific scenario, and there is little to guide them in their choice [13]. To make matters worse, the research community places more emphasis on the compliance management part of the equation, whereas industry practitioners place more emphasis on the process management side.

Trying to reconcile these two perspectives suggests that a comprehensive BPC solution should: i) focus predominantly on checking the compliance of mature processes; ii) not add to the complexity of the existing business operating environment; iii) be capable of scaling to handle transaction volumes in the many millions per day and compliance requirements in the hundreds per process; iv) be easy to setup and maintain for the business user; v) have a very short commercialization timeline; and vi) be capable of addressing the technical compliance challenges identified by the research community. With this in mind, we took a design science approach to developing a comprehensive, "no-code," process mining-based Compliance Center Framework using the following research questions as a starting point:

1. What are the requirements of a comprehensive BPC Compliance Center Framework for the banking industry?
2. How should the Compliance Center Framework be architected and implemented?
3. How will the Compliance Center Framework's user flows ensure that it is easy to use for the business user?

While the development of the Compliance Center Framework was motivated by events in the Australian banking industry, we do not believe that this precludes organizations from other industries or jurisdictions from using it.

The remainder of this paper is organized as follows. In Section 2, we summarize related work. Section 3 introduces the requirements and implementation of the *Framework* and provides a running example of the user flows. In Section 4, we discuss our evaluation, its findings, directions for future research and the threats to validity. In Section 5, we present our conclusions.

2. Background and Related Work

To address our research question, an understanding of both the BPC and process mining literature is required. Here, we provide a summary of both.

2.1. Business Process Compliance

Governatori and Sadiq describe BPC as "a relationship between two sets of specifications: the normative specifications that prescribe what a business has to do, and the process modeling specification describing how a business performs its activities" [1]. This is achieved through verifying a representation of the norms against a formal specification of the process [1]. The BPC lifecycle, comprising three phases: design-time checking; run-time checking; and audit (post execution) checking [16], represents the interaction between the compliance management lifecycle [17] and the business process management lifecycle [18]. Most solutions proposed by the research community focus on developing an approach to address a specific phase of the BPC lifecycle [12].

Managing BPC at design time is a preventative strategy, concerned with ensuring that processes comply with relevant rules and regulations before execution – either during the design process [1] or post-design but pre-execution [19]. Debate has centered on approaches and languages that are expressive enough to handle the range and complexity of compliance requirements, but are seen to be technically complex, and those approaches that are easier to use for non-technical users [7, 11]. Zasada et al., suggest that the language to capture compliance requirements should be: "As complex as necessary ... as simple as possible" [13].

Run-time methods verify compliance during the process execution, and typically address aspects of BPC that cannot be verified and validated at design-time, e.g., segregation of duties and deadlines for completion [10]. Proposed solutions fall into two broad categories: reactive, where compliance verifies progress-to-date [10]; and proactive monitoring, where progress-to-date knowledge is used to predict compliance outcomes [20].

Auditing is a post-execution strategy. Traditionally both manual and sampling-based, there is now a shift to continuous auditing [21]. Some approaches covered in the BPC literature are based on process mining techniques [21], which benefit from reviewing a population of transactions instead of a sample. Database-driven solutions have also been proposed [7].

While progress has been made applying BPC research techniques in real-world scenarios, challenges remain to automate aspects of BPC such as formalizing laws [9] and we are not aware of an end-to-end, commercial solution that can check compliance at all phases of the BPC lifecycle within a complex, banking process environment.

2.2. Process Mining

The goal of process mining is to design methods and techniques that can automatically analyze process execution data, i.e., process event logs, to facilitate business process management activities by extracting actionable process knowledge [18]. For the past 25 years, the research community has predominantly focused on techniques for addressing the problem of i) automated process model discovery; ii) process conformance (and compliance) checking; iii) variant analysis; and iv) process monitoring.

Automated process discovery techniques aim to analyze process event logs and generate a representation of the process behavior as either a Petri net or a BPMN model [18]. Conformance checking aims to detect differences between the prescriptive behavior captured in a process model and the observed behavior of that process, as recorded in the process' event log [14]. Conformance checking techniques predominantly operate on historical process execution data, however, some research efforts have focused on the design of online conformance checking [14]. Online conformance checking is a backward-looking, run-time approach. It compares, in real-time, completed process execution data from a partial trace against prescriptive process behavior [14]. Process monitoring techniques aim to analyze real-time process execution data, i.e., while the process unfolds, but unlike online conformance checking they are forward looking [22]. There are two types: i) predictive monitoring, e.g., How long will the process take to complete? [22]; and ii) prescriptive monitoring, e.g., what task should be executed to reach a positive outcome [23]. Variant analysis techniques aim to identify differences in process behavior – predominantly control-flow and process performance differences – by comparing two (or more) event logs from the same process [18].

3. The Compliance Center Framework

In this section, we describe our approach to designing and implementing our Compliance Center Framework (henceforth, the *Framework*). Our summary analysis of the literature and regulatory reports concluded that the banking industry urgently requires a solution to address its BPC challenges across the BPC lifecycle. Recognizing the technical challenges outlined by the research community [13, 7], we also note that satisfying the need for urgency will likely lead to a compromise in terms of the degree of automation. Hence, our objective for the *Framework* is to: Create a comprehensive framework for checking business process compliance that is easy to use for business users, which can be deployed

Table 1
Compliance Center Framework requirements summary

Cat	RID	The Compliance Center Framework must ...
i.	R1	Automatically discover a process model from an event log.
ii.	R2	Enable access to an existing risks, obligations and controls register.
	R3	Enable a non-technical user to create a *Control*.
	R4	Enable a non-technical user to create a *Risk* and/or *Obligation*.
iii.	R5	Enable a non-technical user to formalize a *Control* with complex requirements.
	R6	Allow a single control to be associated with one or more *Risks* and/or *Obligations*.
	R7	Allow multiple instances of a *Control* to be applied to a process.
iv.	R8	Evaluate whether a *Control* has been violated.
	R9	Ensure that at least one *Control* is assigned to each *Risk* and/or *Obligation*.
	R10	Ensure that a non-technical user can check compliance, irrespective of the BPC lifecycle stage, without technical guidelines.
v.	R11	Be able to report the number of control violations across multiple processes.
	R12	Align to the taxonomies of the organizational risk management framework.
	R13	Be able to report the number of control violations by *Risk*/*Obligation*.
	R14	Ensure that reports always reflect the most current process data.
vi.	R15	Notify users when a control violation has been detected.
	R16	Support and check the effectiveness of violation recovery actions.
	R17	Enable the user to visualize the cause of any violation and access the underlying case and event data for any violated control.
vii.	R18	Provide guidance and support to ensure a broad base of non-technical users can set up, run, and maintain the *Framework*.
	R19	Be able to identify and notify a user of any change to a risk, obligation, and/or control.
	R20	Ensure that each risk, obligation, and risk have an assigned owner.

rapidly and widely within an organization, which addresses the challenges outlined in previous work but acknowledges that there may be trade-offs. To meet this objective, we first present the requirements for the *Framework*, followed by a discussion of our implementation approach, and a running example.

3.1. Design Requirements

Informed by the work in [19, 11, 24] and the industry perspective, we specified requirements in seven distinct categories, summarized in Table 1 and justified (referring to their IDs) as follows:
i) Specifying the process representation (R1): The regulators identified mature processes as the area of greatest concern [4, 2, 3]. These processes are complex, with many variants, and a manually discovered process model is unlikely to describe an accurate representation of the actual process [5].
ii) Identifying the compliance requirements (R2-R4): Principle 2 of the Basel Committee's[2] "Revisions to the Principles for the Sound Management of Operational Risk" [25] requires banks to "develop, implement and maintain an operational risk management framework (ORMF) that is fully integrated into the bank's overall risk management processes". A bank's ORMF will typically include a bank's risks, obligations and controls register [26]. The control description in the register is a natural language description of the compliance requirement, as such, we considered it to be our primary source of compliance requirements, and the *Framework* must enable a business user to either import risks, obligations and controls from an existing register or manually create them.
iii) Formalizing the compliance requirements (R5-R7): In practice, the controls that serve as the source of the compliance requirements are frequently more nuanced than referenced in the literature [5]. For example, controls may: i) be composite controls that represent multiple process perspectives; ii) be controls that refer to multiple processes; iii) apply to specific activities within a process or the entire process; and iv) apply to more than one process or be applied multiple times within a process.

[2]The Basel Committee on Banking Supervision (BCBS) is the primary global standard setter for the prudential regulation of internationally active banks.

iv) Specifying the compliance checking approach (R8-R10): The fundamental requirement for any BPC solution is to check that the behavior represented in the process satisfies its compliance requirements, i.e., controls are in place for each *Risk* and/or *Obligation* and that there are no violations [1]. The literature emphasizes the need for a BPC framework that satisfies this requirement at each stage in the BPC lifecycle [24, 11, 7]. Both the literature [13] and the regulatory reports [2, 3] make it clear that the people responsible are likely to be non-technical, business users.
v) Reporting and governance (R11-R14): In terms of reporting & governance, one of the Board priorities proposed in [3] stated that monitoring is required at an enterprise level, a governance level, and a transactional level. This requires a BPC framework that can align to the broader enterprise operational risk framework, including its inherent taxonomy, aggregate compliance reporting across multiple processes, filter by *Risk* and/or *Obligation*. In addition, the regulatory report in [27] stresses the importance of reports reflecting the most current process data **(R14)**.
vi) Violation handling (R15-R17): The findings in [3, 2] recommend a more proactive approach to risk management, hence the *Framework* must notify users when a violation has been detected **(R15)**, support and check the effectiveness of violation recovery actions **(R16)** and improve the quality of root cause analysis **(R17)** [27].
vii) Other (R18-R20): Requirement **R18** acknowledges the technical [13] and domain knowledge gaps [3] of the *Framework*'s users. Both the literature [7] and the regulatory reports [4, 2, 3] highlight the extent of compliance requirements change **(R19)**. Finally, in Australia, ASIC's and APRA's Financial Accountability Regime (FAR),[3] ensures that accountability is attributed to "directors and [the] most senior and influential executives" hence **R20**.

3.2. Implementing the Compliance Center Framework

To introduce our implemented solution, we follow the same structure as Section 3.1, addressing each requirement in turn and then summarizing the *Framework* architecture and points of difference with the current paradigm. Where, in the following Section, we describe user flows at the UI/UX level, we do so to highlight the simplicity of the implemented framework.
Satisfying R1: Given the urgency, we determined that the *Framework* should leverage an existing, commercial process mining solution, and selected the Apromore process intelligence platform on the basis that: i) automated process discovery is a core capability of the platform; ii) it has both an Academic version accessible through the Academic Alliance and a commercial solution that is currently licensed by many clients in the banking sector; iii) it is a "no-code" solution with an intuitive interface (we discuss this point in more detail below); iv) it would reduce the development scope of the *Framework* without compromising the main contribution; and v) the platform has existing ETL (extract, transform and load) functionalities to address the potential data issues.
Satisfying R2-R4: The *Framework* should provide both an "import" form and manual forms to add a new *Risk*, *Obligation*, and/or *Control*, displaying each item in a general library displayed on a main panel of the user interface.
Satisfying R5-R7: Formalizing a compliance requirement is the heart of our *Framework*. Our approach is based on applying First-Order Linear Temporal Logic over Finite Traces (FO-LTL-FT). Once a *Control* has been created, this should follow a three-step approach to translate the *Description* of a *Control* into a parameterized query ready to filter/analyze an event log: a) identify the *Control Type* and applicable *Control Template*; b) assign the *Control* to a *Risk* and/or *Obligation*; c) assign the *Control* to a process (via its event log) and instantiate the *Control Templates* linking them to the corresponding event log data (i.e., attributes and their values). The approach is summarized in Figure 1. Here we describe the three steps, which we will elaborate further in our running example.

a) The *Framework* should "recommend" one or more *Control Types* based on the control *Description* – this is achievable via traditional natural language processing techniques. The user should be able to edit the recommended option. Valid *Control Type* options should be any combination of "Control Flow,"

[3] ASIC (Australian Securities & Investments Commission) is Australia's corporate, markets and financial services regulator, APRA (Australian Prudential Regulation Authority) is Australia's prudential supervisor.

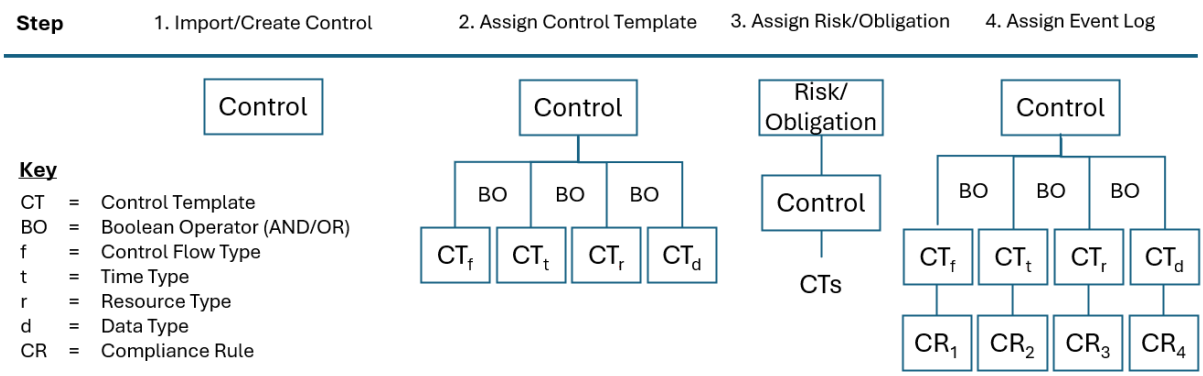

Figure 1: Compliance Center Framework approach to formalizing a compliance requirement

"Resource," "Time," and "Data," i.e., the four process perspectives. Selecting a *Control Type* should enable a set of applicable *Control Templates* at the bottom of a form. Each *Control Template* should determine the pattern to be filled when formalizing a *Compliance Rule*, which should be based on the compliance patterns referred to in the compliance request language meta-model of Elgammal et al. [11].

The user should be able to select Boolean operators (e.g., AND, OR) to express the relationship between multiple *Control Templates*. For example, to specify that the "Credit Check" task should precede the "Create Offer" task and be completed by different resources (segregation of duty), would require both a "Control Flow" template and a "Resource" template joined by an "AND" operator. To address **R7**, the user should be able to add/edit/delete additional *Control Templates* and apply Boolean operators to express the relationships. For example, where a task may be performed by more than one resource type an *OR* operator would be used within the "Resource" *Control Template*.

b) To assign the *Control* to one or more *Risks* (or *Obligations*) the user should be able to select and edit the relevant *Risk* (or *Obligation*) from the library. Each *Risk* and *Obligation* should have a list of assigned *Controls*, and the user should be able to assign or remove a *Control* from this list **(R6)**.

c) The user should be able to complete this step via the "Assign control to process" form. After selecting both a *Control* and associated event log, the remaining part of the form should display the *Control Templates* that define the selected *Control* and a range of text boxes and dropdown boxes that allow the user to complete the specification of the *Compliance Rule(s)* by instantiating the *Control Templates*. The *Control Template* created within a *Control* should now be formalized by substituting the generic operands with either variables and/or event log attributes and their values. The user should select/input the event log attributes and their values from either dropdown boxes populated from the event log data, e.g., the task name as documented in the event log, or text boxes for the user to input a variable **(R5)**.

Satisfying R8-R10: The *Framework* should evaluate all *Compliance Rules* of a *Control*, by querying the event log using the query parameters of each *Compliance Rule*. The query should return the violations per process instance of the *Compliance Rules* and their Boolean operators for the *Control* **(R8)**. To satisfy **R9**, the user should be able to review at any time a *Risk* or *Obligation* to observe the current list of assigned controls. To ensure the approach is suitable for checking compliance at different BPC lifecycle stages, a BPC lifecycle, stage-specific event log should be selected. Design time for banking processes is primarily concerned with enhancing and modifying existing processes, not modeling new processes from scratch. As such, during the testing cycle for a process enhancement – post-design and build, but pre-implementation – process simulators may generate event logs, which should be the input for design-time compliance checking. If a "To Be" model exists, simulation parameters can also be applied to the model to generate an event log to be checked at design time. For run-time event logs, the *Framework* should adopt a near real-time approach (refreshing event log data and running the compliance checks every 10-15 minutes). This would keep the *Framework* design relatively simple, by not having to incorporate event stream monitoring, without significantly compromising the intent of

run-time checking. This is an example of a trade-off. There are requirements that banks must monitor in real time, e.g., fraud monitoring, sanctions screening, however, large, mature banks already deploy a wide range of bespoke applications, hence the limitation is mitigated. To complete the BPC lifecycle, given that some compliance rules cannot be checked at run time (requiring cases to be completed before checking – e.g., reconciliations, aggregated SLA adherence), the *Framework* should allow the checking of post-execution event logs **(R10)**.

Satisfying R11-R14: The *Framework* should be able to report the number of compliant and violating cases whenever an event log is updated, or on demand, and allow the user to filter by item (process, *Risk*, *Obligation*, and *Control*) as well as the categories used to describe the items.

Satisfying R15-R17: The *Framework* should notify users when new violations have been detected and assess the effectiveness of recovery actions. For instance, a *Control* including two *Control Templates* connected by an "OR" would ensure that if the compliance rule associated with the first *Control Template* is violated but the compliance rule associated with the second *Control Template* is not, then the *Control* would not be flagged as violated. To satisfy **R17**, the *Framework* operational dashboard should display at least the number of violations, a Pareto chart of the violations, a table listing the process instances that violate one or more controls, and a one-click drill-down functionality to inspect each process instance.

Satisfying R18-R20: To satisfy **R18** the *Framework* should ensure that the interface is easy to use for the business user, i.e., a "no-code" approach, a common look and feel across all aspects of the interface, text boxes and drop-down boxes for user input, context-sensitive action menus, hover buttons describing the underlying object, a dedicated *Compliance Center* folder within the user's workspace, search and help functionalities. To satisfy **R19-R20** the *Framework* should check for changes whenever a risks, obligations and controls register is refreshed, notifying users and assigned owners of any variance.

Compliance Center Framework architecture: The architecture for the *Framework* is described in Figure 2. The architecture reflects both our design and our implementation of the *Framework* within the Apromore process intelligence platform.[4] The *Framework's* essential components are: the risks, obligations, and controls repository; a query engine; the violations repository; the user interface (repository navigation UI and dashboarding UI). The ETL engine and the event logs repository, while being fundamental components for our *Framework*, came natively from the Apromore platform.

Figure 2 also displays how the components interact. The ETL engine uploads and appends process data to the event logs repository. If controls are assigned to the process whose data has been updated, the ETL engine notifies the query engine of the change. The latter will trigger a reassessment of the controls and store in the violation repository all the identified violations. Identified violations are then displayed on a custom dashboard tailored for compliance reporting.

Our *Framework* differs from the current BPC paradigm, presenting business users with a common look and feel, in a "no-code" environment, with a consistent approach to evaluation over the BPC lifecycle, using artifacts with which they are already familiar (e.g., the risks, obligations, and controls register). Basing the *Framework* on a commercial process intelligence platform also means that it can be deployed rapidly and widely, while leveraging several orthogonal process intelligence functionalities can support and extend its effectiveness and value. In this respect, the *Framework* satisfies our objective. It is flexible enough to evaluate more complex and nuanced controls, including compensating for violations, and its ability to report violations across processes, risks, obligations, and controls supports the broader governance requirements outlined by

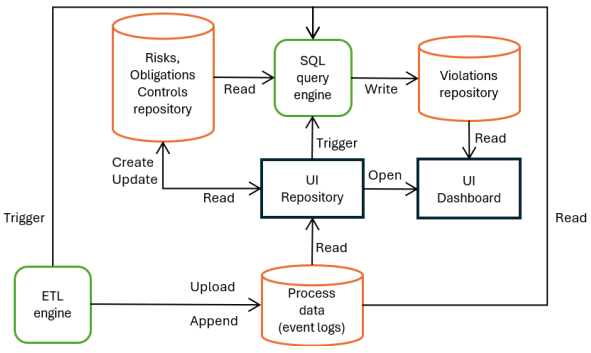

Figure 2: Compliance Center Framework components architecture – as implemented

[4]Available on the Academic Alliance instance from late October 2024.

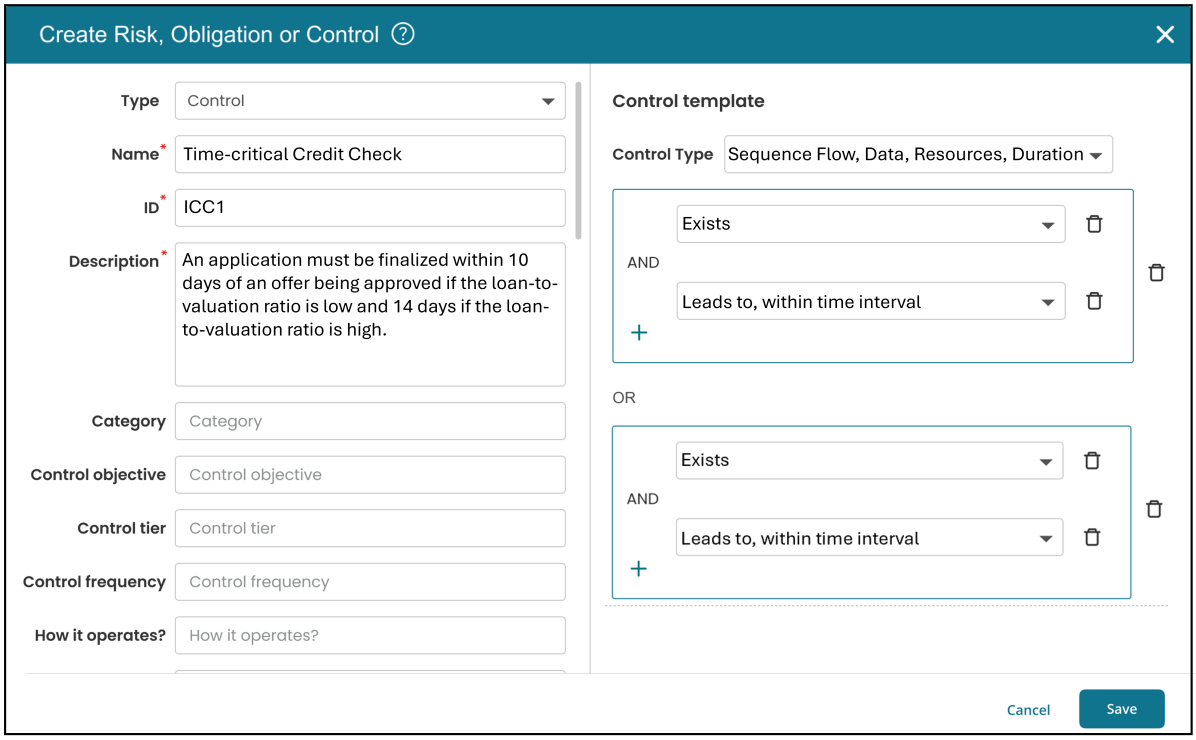

Figure 3: Creating a *control* (Step 2)

the regulators, e.g., Recommendation 16 in [2].

3.3. User Workflow with a Running Example

We illustrate the core *Framework* workflow of formalizing a compliance requirement with a running example. We assume that a user has identified a "Secured Lending Credit Default" *Risk*, and the user has created a "Time-critical Credit Check" *Control* with the following description: "An application must be finalized within 10 days of an offer being approved if the loan-to-valuation ratio is low and 14 days if the loan-to-valuation ratio is high". The *Control* should be assigned to the "Secured Lending Credit Default" *Risk* and the "Loan Origination" event log. The workflow enables business users to i) create a *Risk* and/or *Obligation*; ii) create a *Control*; iii) assign a *Control* to a *Risk* and/or an *Obligation*; iv) assign a *Control* to an event log by instantiating it (i.e., creating a compliance rule); v) evaluate the *Compliance Rules*; and vi) review the outcome of the compliance evaluation via a dashboard. In Figure 3 we provide an example of how the user creates a *Control* (step 2) and, in Figure 4, how the user assigns a *Control* to an event log by instantiating it (step 4). However, for reasons of space and clarity, we decided to showcase this running example in a video tutorial available online: dx.doi.org/10.6084/m9.figshare.26636308.

4. Evaluation

In this section, we present our evaluation design, followed by the feedback we received concerning our design science evaluation criteria (relevance, validity, utility, quality, and efficacy of the *Framework* [28]). We then discuss this feedback, before commenting on the threats to validity.

4.1. Evaluation Design

Given the exploratory nature of the evaluation, we chose a semi-structured focus group as our evaluation method, recruiting senior bankers from the primary researcher's personal network as participants. We chose to run two focus groups to cover both new and mature processes. The first group, comprising

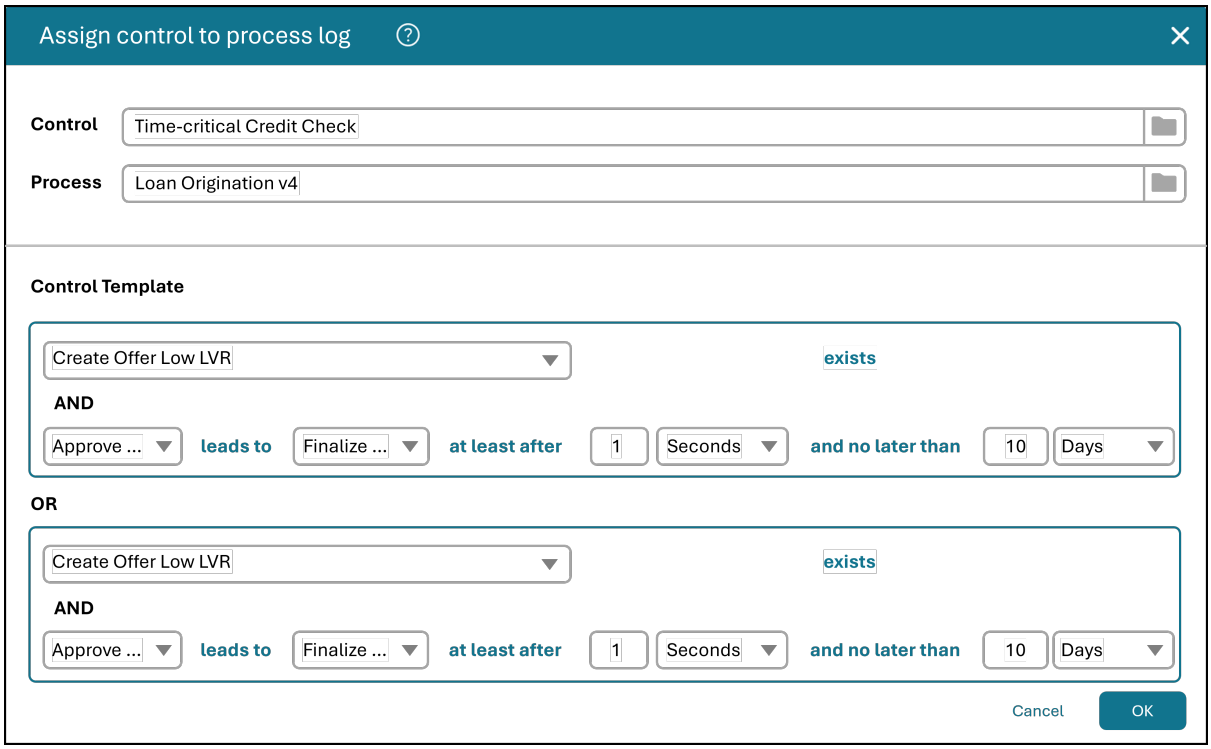

Figure 4: Creating and instantiating a *Control* (Step 4)

two risk business partners and a senior process excellence manager, had 14 years' banking experience (on average). They were part of a project team focused on re-imagining and designing new processes. The purpose of selecting this group was to evaluate the validity of using test logs to check compliance at design time. The second group was an operations team working with existing, mature processes. The five participants had a mix of operations, risk, projects, and business analyst expertise, with over ten years' banking experience (on average).

The focus groups lasted one hour and were conducted on bank premises. Two researchers were present at each focus group. We asked participants to discuss their current BPC challenges and current compliance evaluation approach. We then presented a wireframe of the user workflows and directed the conversation to capture the group's feedback on the evaluation criteria. We recorded the responses and present the feedback below.

The primary researcher also conducted three 30-minute interviews with senior banking executives including a Chief Operating Officer, a Chief Compliance Officer, and a Divisional Chief Risk Officer, as well as two interviews with consulting Principals (each with more than ten years' experience, one in process excellence, the other in technology). Each executive was briefed on the research and the proposed *Framework*. The interviewees were asked to comment on the underlying problem being addressed and the relevance of the *Framework* to their organization or clients.

4.2. Focus Group and Interviewee Feedback

Relevance: Both focus groups reinforced the challenges outlined in [5]. They referenced the extent and complexity of the requirements that they must comply with and their ambiguity, e.g., a requirement to check for a "material change to circumstances." The second group highlighted the complexity of operating a common process across multiple legal jurisdictions, with slight variations in each, making the operating environment for staff particularly confusing. The second group also referenced the spaghetti nature of the process environment and the challenge of extracting poor quality data from multiple legacy systems. Two of the executives referred to the increasing cost of compliance within the industry. Each executive referenced specific problems in their scope directly relevant to the problem

Table 2
Summary of evaluation feedback on the *Framework*

Criterion	Summary
Relevance	Accurately reflects the practical challenges of monitoring compliance in a complex environment.
	Aligns with operational resilience regulatory guidelines.
	The cost of compliance is increasing.
Validity	Integration with risks, obligations and controls register is critical.
	Improves on current "compliance-by-design" approach.
	Reinforces the importance of configuring and checking composite controls.
Utility	Faster identification of violations is a significant benefit.
	It is an Important communications tool for providing feedback to senior stakeholders.
	It has broad applicability among a wide range of users.
Quality	Addresses the shortcomings of reactive, sample-based checking.
	Integrates compliance with other performance analytics.
	Enables more precise compliance rule definitions.
Efficacy	More cost effective than alternatives.
	Customizing dashboard views to suit the needs of specific audiences is critical.
	Control templates help drive standardization.

the *Framework* was trying to solve. In terms of the consultants, one referenced the "loose technology coupling" (the use of event logs as the process representation rather than monitoring tools placed directly in the process) would facilitate technology change and that it would require C-suite ownership. The other consultant referenced the regulatory guidelines on operational resilience that had recently been introduced in Australia and required organizations to specifically monitor process compliance.

Validity: In terms of validity, both focus groups confirmed that leveraging and integrating with their organization's risks, obligations, and controls register would be a prerequisite. They reinforced the more nuanced nature of controls and the validity of taking a composite control approach. The first group confirmed the validity of the *Framework* using test logs at design time, and that their current "compliance by design" approach lacked specific support tools. They also confirmed that "compliance by design" is not foolproof and both run-time and post-execution control testing is required. The second group was less familiar with process mining techniques, but recognized that techniques to extract, transform, and load the event logs from a complex data environment would be critical.

Utility: The reporting timeliness of the *Framework* was identified as a significant benefit. One specific violation the first group referred to would not have been detected for up to one year, if not discovered by chance. With a wider range of backgrounds, the second group's discussion highlighted the utility of the *Framework* as a communication tool for presenting to senior stakeholder groups. The second group indicated that the solution has broad applicability among business users, e.g., risk assurance managers reviewing regulatory change, team leaders managing day-to-day compliance, and business analysts responsible for process change.

Quality: Both groups indicated that control testing is currently reactive, and sample based. A "whole of population" solution, tested particularly in near real-time, would be a significant improvement in quality. When the second group was shown the reporting dashboard, the conversation moved from checking compliance to other aspects of business performance that could also be measured more effectively, e.g., specifying rules to help define current subjective views of quality. They commented that using the control templates and compliance rule approach would: i) push them to have a deeper discussion on the definition of quality, because they would have to formalize the *Compliance Rules*; ii) allow them to select *Compliance Rules* to tighten over time; and iii) allow them to use the dashboard as a training and coaching tool across processing centers.

Efficacy: The first group suggested that the approach may be more cost effective than the alternatives, and with digitized processes comprising more steps and more controls to check, this could generate ongoing benefits. The first group also commented that they had been considering developing "bots" to monitor the automated controls, but the *Framework* approach would be far more cost effective. The

second group highlighted the criticality of report customization by identifying multiple perspectives: the team leader, the quality control analyst, the performance improvement analyst, and senior executive perspective. The questions they asked also covered using the control templates to help drive standardization and measure performance, which would provide additional benefits.

4.3. Discussion

The *Framework* design satisfies our objective of creating a comprehensive framework for checking BPC that we believe is capable of being run by business users. It can be deployed at design time, near real time, and post execution. It can evaluate 21 different unique *Control Templates*, and many more when composite rule permutations are considered, without users having to learn a different approach for each BPC lifecycle stage. Further work is required to understand the coverage of the templates. Furthermore, by leveraging an existing process intelligence platform, the *Framework* can be deployed both rapidly and widely and does not add to the organizational complexity.

We addressed RQ1 by detailing the requirements and features of a comprehensive *Framework* for the banking industry in Table 1. Section 3.2 describes how the *Framework* is architected (RQ2) and the figures and video in Section 3.3 show how the "no-code" approach addresses RQ3. The *Framework* also addresses the findings in [8] that BPC is not just a technical problem to be solved but concerns organizational and social aspects that cannot be fully automated. The focus group feedback was encouraging, and the senior executives and consultants recognized the relevance of the *Framework*.

The feedback helped identify opportunities for further research. The focus group participants referenced the challenges of managing inconsistencies in requirements. A related opportunity for further research is identifying redundant and duplicate controls, "[there are] controls built on top of controls built on top of controls" [5]. Ongoing research in predictive and prescriptive monitoring, specifically the likelihood of violating controls covering all process perspectives, will help the industry move to a more proactive approach to BPC.

There are a number of threats to validity. The *Framework* assumes that the risk, obligations, and controls register is a complete and accurate representation of the compliance requirements, interpreted correctly in the control description. This assumption has not been empirically validated, beyond anecdotal evidence in the focus groups. We do not have empirical evidence that the 21 compliance templates are capable of formalizing all compliance requirement types within the banking industry. We also acknowledge the potential limitations of FO-LTL-FT as a logic for formalizing compliance requirements. The *Framework* has been developed for business users operating in the BPC domain, and the UI/UX design is intended to be easy to use, however, we have not yet determined the extent of training required for business users to operate it effectively. Additional potential limitations that we have not yet tested include the *Framework's* ability to scale to support the high transactional volumes common in a major bank, its ability to assess compliance in a complex ecosystem, or the number of attributes that can included without impacting performance.

5. Conclusion

BPC is a multi-faceted problem [8] that cannot be solved yet with an end-to-end, automated approach [11]. By re-framing the problem, we were able to design a comprehensive BPC solution that is easy for business users to run. The critical assumptions that informed the design of our *Framework* were that: i) an existing risks, obligations and controls register would act as the single source of compliance requirements; ii) regulators were primarily concerned with existing, mature processes; iii) the user base would be non-technical; and iv) choosing an event log as the process representation would allow us to leverage an existing commercial process intelligence platform. On this basis our *Framework*, designed as a "no-code" solution, enables users to simply: i) define a process by importing an event log; ii) import risks, obligations and controls from existing registers; iii) apply control templates to facilitate compliance rule setting; iv) generate compliance violations dashboards and reports; and v) drill down to the underlying cases generating the violations. Industry focus groups and executive feedback was

favorable. While the focus for the design of the *Framework* has been on the Australian banking industry, we do not see any reason why other heavily regulated industries, or banking in other jurisdictions cannot leverage the *Framework*. Our next step is to test it in a case study environment.

References

[1] G. Governatori, S. Sadiq, The journey to business process compliance, IGI global, 2009.

[2] APRA, Prudential inquiry into the commonwealth bank of australia (cba) final report, 2018. URL: https://www.apra.gov.au/news-and-publications/apra-releases-cba-prudential-inquiry-final-report-and-accepts-enforceable.

[3] Westpac, Board governance of aml/ctf obligations at westpac, 2020. URL: https://www.westpac.com.au/about-westpac/media/media-releases/2020/4-june/.

[4] K. Hayne, Royal Commission into misconduct in the banking, superannuation and financial services industry, Royal Commission into Misconduct in the Banking, Superannuation and Financial Services Industry, 2019.

[5] N. Adams, A. Augusto, M. Davern, M. L. Rosa, Why do banks find business process compliance so challenging? an australian perspective, in: International Conference on Business Process Management, Springer, 2022, pp. 3–20.

[6] M. Rosemann, M. zur Muehlen, Integrating risks in business process models, in: ACIS 2005 Proceedings, volume 50, 2005.

[7] 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.

[8] A. Barnawi, A. Awad, A. Elgammal, R. Elshawi, A. Almalaise, S. Sakr, An antipattern-based runtime business process compliance monitoring framework, framework 7 (2016) 551–572.

[9] I. A. Amantea, L. Robaldo, E. Sulis, G. Governatori, G. Boella, Business process modelling in healthcare and compliance management: a logical framework, Journal of Applied Logics—IfCoLog Journal of Logics and their Applications 9 (2022).

[10] F. M. Maggi, M. Montali, M. Westergaard, W. M. Van Der Aalst, Monitoring business constraints with linear temporal logic: An approach based on colored automata, in: International Conference on Business Process Management, Springer, 2011, pp. 132–147.

[11] A. Elgammal, O. Turetken, W.-J. Van Den Heuvel, M. Papazoglou, Formalizing and appling compliance patterns for business process compliance, Software & Systems Modeling 15 (2016) 119–146. URL: https://dx.doi.org/10.1007/s10270-014-0395-3. doi:10.1007/s10270-014-0395-3.

[12] S. Sackmann, S. Kuehnel, T. Seyffarth, Using business process compliance approaches for compliance management with regard to digitization: evidence from a systematic literature review, in: ICBPM, Springer, 2018, pp. 409–425.

[13] A. Zasada, M. Hashmi, M. Fellmann, D. Knuplesch, Evaluation of compliance rule languages for modelling regulatory compliance requirements, Software 2 (2023) 71–120.

[14] J. Carmona, B. van Dongen, A. Solti, M. Weidlich, Conformance checking, Springer, 2018.

[15] A. Augusto, R. Conforti, A. Armas-Cervantes, M. Dumas, M. La Rosa, Measuring fitness and precision of automatically discovered process models: A principled and scalable approach, IEEE Transactions on Knowledge and Data Engineering (2020).

[16] M. El Kharbili, A. K. A. de Medeiros, S. Stein, W. M. van der Aalst, Business process compliance checking: Current state and future challenges, Modellierung betrieblicher Informationssysteme (MobIS 2008) (2008).

[17] E. Ramezani, D. Fahland, J. M. van der Werf, P. Mattheis, Separating compliance management and business process management, in: International Conference on Business Process Management, Springer, 2011, pp. 459–464.

[18] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, Fundamentals of business process management (Second Edition), Springer, 2018.

[19] M. El Kharbili, S. Stein, I. Markovic, E. Pulvermüller, Towards a framework for semantic business process compliance management, Proceedings of GRCIS 2008 (2008).
[20] S. Rinderle-Ma, K. Winter, J.-V. Benzin, Predictive compliance monitoring in process-aware information systems: State of the art, functionalities, research directions, Information Systems 115 (2023) 102210.
[21] W. M. van der Aalst, K. M. van Hee, J. M. van der Werf, M. Verdonk, Auditing 2.0: Using process mining to support tomorrow's auditor, Computer 43 (2010) 90–93.
[22] I. Verenich, M. Dumas, M. L. Rosa, F. M. Maggi, I. Teinemaa, Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring, ACM Transactions on Intelligent Systems and Technology (TIST) 10 (2019) 1–34.
[23] Z. D. Bozorgi, I. Teinemaa, M. Dumas, M. La Rosa, A. Polyvyanyy, Prescriptive process monitoring for cost-aware cycle time reduction, in: 2021 3rd International Conference on Process Mining (ICPM), IEEE, 2021, pp. 96–103.
[24] L. T. Ly, F. M. Maggi, M. Montali, S. Rinderle-Ma, W. M. P. Van Der Aalst, Compliance monitoring in business processes: Functionalities, application, and tool-support, Information Systems 54 (2015) 209–234. URL: https://dx.doi.org/10.1016/j.is.2015.02.007. doi:10.1016/j.is.2015.02.007.
[25] B. C. on Banking Supervision, Revisions to the Principles for the Sound Management of Operational Risk, Report, 2021. URL: https://www.bis.org/bcbs/publ/d515.pdf, (Accessed on 10/17/2024).
[26] A. Chapelle, Operational risk management: Best practices in the financial services industry, John Wiley & Sons, 2019.
[27] ASIC, Rep 594 review of selected financial services groups' compliance with the breach reporting obligation, 2018. URL: https://download.asic.gov.au/media/4879889/rep594-published-25-september-2018.pdf.
[28] A. R. Hevner, S. T. March, J. Park, S. Ram, Design science in information systems research, MIS quarterly (2004) 75–105.