Towards Speech-Act-Based Compliance

Johannes Tenschert*, Georg Michelson†, and Richard Lenz‡
*Institute of Computer Science 6 (Data Management)
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
johannes.tenschert@fau.de
†Chair of Ophthalmology
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
georg.michelson@uk-erlangen.de
‡Institute of Computer Science 6 (Data Management)
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
richard.lenz@fau.de

***Abstract*—Todays organizations are socio-technical entities, and interactions have significant impact on their success. Around 50% of the work in the US is knowledge work, and other countries show a similar tendency. Compliance monitoring of course can only cover processes and tasks that are modeled or documented somehow. Therefore, today it faces the problem that half of the workforce performing ad-hoc interactions and tasks cannot be supported properly. This paper introduces speech-act-based business rules that focus on the pragmatic intention of interactions instead of events and artifacts. No process schema is necessary. The rules can be applied on annotated tasks or artifacts in BPMSs as well as documented ad-hoc interactions. The approach requires some mapping of interaction artifacts to speech acts, and except for cross-cutting concerns a common process or case identifier to formulate rules on associated artifacts. Speech-act-based inference relying on Searle's $F(P)$ framework enables for example finding commitments including unfulfilled promises, pending interactions, and patterns of interaction. The introduction of Speech Act Theory into compliance monitoring can facilitate the integration of structured, semi-structured and ad-hoc processes. Integrating Speech Act Theory into common approaches to compliance monitoring is examined: LTL and CTL rules, Rete-based business rule management systems represented by JBoss Drools, and complex event processing. Rules are not restricted to validate compliance, they also emit tasks, recommendations, and warnings to translate legal requirements and corporate objectives into daily business.**

Keywords*—business process compliance, speech act theory, business rules, adaptive case management, knowledge-intensive business process

I. Introduction

In recent years, organizations had to cope with a rising need for monitoring compliance of their business processes, as well as an ad-hoc nature of many tasks in these processes with the rising share of knowledge work. Knowledge work involves the creation, distribution, or application of knowledge [1], and is furthermore characterized by abstractly defined tasks opposite to the manual workers' clearly defined activities, and the knowledge workers' responsibility for his own contribution in terms of quantity and quality [2]. Today, around 50% of the work in the US is knowledge work [3]. Interaction-based work, which represents a large share of knowledge work, accounts for 41% of the work in the US and 37% in Germany [4]. Organizations are socio-technical entities [5], and they are constituted as a network of conversations [6]. Therefore, interactions are significant for the success of organizations.

The techniques to automatically monitor and ensure regulatory compliance obviously can only cover tasks that are modeled or documented somehow. Neither inflexible BPM systems nor information systems waiving compliance checking are sufficient [7]. Therefore, ad-hoc interactions of knowledge workers and probably up to 50% of the actual work can not be monitored and supported properly. Moreover, knowledge workers have to deal with many different collaboration systems as well as domain-specific tools, and therefore compliance has to deal with their events and artifacts as well.

We propose to use Speech Act Theory for documenting ad-hoc interactions, supporting semi-structured processes, and annotating structured processes. The focus on interactions first of all conforms to the actual work that is performed. It allows to integrate ad-hoc activities into compliance monitoring. Thereby, it can make compliance the solution to integrating today's structured and ad-hoc work.

This paper introduces speech-act-based business rules that focus on interactions instead of events and artifacts. The rules can be applied equally on annotations in structured processes and ad-hoc interactions. The introduction of Speech Act Theory into common approaches to compliance monitoring was examined: Linear temporal logic (LTL) and computation tree logic (CTL) rules, JBoss Drools as a representative for Rete-based business rule management systems (BRMS), and complex event processing. Rules may check compliance prior to the actual performance of interactions to facilitate the integration of different systems by emitting a *veto* to possibly prevent many compliance breaches from happening. Moreover, rules may create tasks, recommendations and warnings to translate legal requirements and corporate objectives into daily business.

In the following sections, we briefly introduce Speech Act Theory, motivational scenarios, and related work. Section V introduces the requirements for speech-act-based inference in compliance rules, and heuristics to be used in business rules engines. Section VI outlines how speech-act-based inference can be integrated into common approaches for business rules

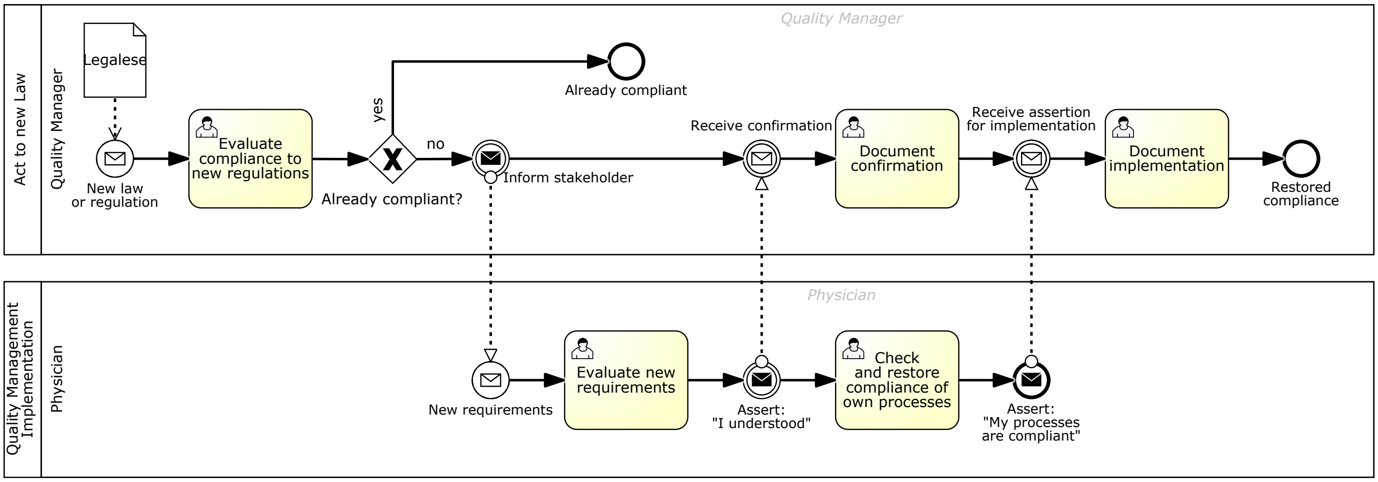

Fig. 1: Quality Management in Hospitals: Acting to new laws or regulations

in compliance, and Section VIII provides some examples on how to use speech-act-based business rules. In Section IX we discuss our approach, and in Section X we conclude and describe further questions to be answered.

II. Speech Act Theory

Speech Act Theory was first introduced by Austin [8] and further elaborated by Searle [9], [10]. Saying something is an action with a particular intention of the speaker. Not only utterances are speech acts, but rather all activities with the intention to send a message. Some types of interactions adhere to typical patterns, e. g. questions are usually followed by an answer. The speaker is well aware of this context and of his pragmatic intention, but the systems supporting him currently are not. A speech act consists of an illocutionary force and propositional content. The illocutionary force can uniquely be characterized by its illocutionary point, preparatory conditions, mode of achievement, degree of strength, propositional content conditions, sincerity conditions, and degree of strength of sincerity conditions [10]. Searle distinguishes the following five illocutionary points of illocutionary forces, i. e. the intention of a speaker:

- **Assertive:** Commit the speaker to something being the case, e. g. assert, inform, remind.
- **Commissive:** Commit the speaker to some future course of action, e. g. commit, promise, accept.
- **Declarative:** Change the reality according to the propositional content, e. g. approve, decline, judge.
- **Directive:** Attempt to cause the hearer to take some particular action, e. g. request, ask, order.
- **Expressive:** Express the attitude or emotions of the speaker, e. g. thank, congratulate, apologize.

The English language alone contains at least 4800 speech act verbs [11] which may have different meanings in different domains. Selections of speech acts for a specific domain or purpose may be called a speech act library. For example, in the context of a hospital, 21 illocutionary forces were identified [12], and this library may act as a framework to document interactions in this domain.

In the $F(P)$ framework [9], where F is an illocutionary force and P is the propositional content, $F(P)$ can be the propositional content of some other speech acts, for example $F_2(F_1(P))$ which may represent informing F_2 about some promise F_1 to do P. There is no conceptual limit on how deep nesting can be.

III. Motivational Scenarios

Introducing Speech Act Theory into Compliance is useful for sectors which emphasize interaction both in their daily work and their corresponding laws and regulations. For completely structured processes implemented in one BPMS the benefits of explicitly modeling and auditing speech acts probably are negligent. The following domains contain processes with routine fragments that involve knowledge workers and in practice require several BPMS and information systems.

In Germany, doctor's offices and hospitals are legally obligated to implement a system for quality management to ensure continually improving patient care. This obligation is based on §135a (2) SGB V [13] and the ÄQM-RL[1] guideline on quality management [14]. The contents of quality management rules are continuously changing. Thus, a system is needed to monitor and assure the appropriate changes. Unsurprisingly, the guideline lists the design of communication processes as one of its basic elements.

The process for quality management consists of many subprocesses, including acting to new laws or regulations. When new regulations are required to put into practice, the quality manager evaluates compliance to them and decides which stakeholders need to be informed. After this evaluation, it is either deemed irrelevant or the appropriate stakeholders need to be informed about derived requirements for their processes. A stakeholder needs to confirm that he understood these requirements and after he implemented changes or

[1]Qualitätsmanagement-Richtlinie vertragsärztliche Versorgung [14]

positively checked his processes for compliance he also needs to confirm that his processes now are compliant. The quality manager receives and documents both confirmations. This process fragment is depicted in Figure 1. Of course, this only describes a happy path. The process actually may contain many ad-hoc interactions to clarify requirements and to resolve conflicts with existing practices.

In the process of patient information and consent, the physician has to properly inform his patient about his medical condition, his choices of treatment, and corresponding complications of the selected treatment. On an abstract level, this process is rather fixed and typically ends with the patient signing a form to acknowledge informed consent. However, if high risks (e. g. complication rate, mortality rate, side effects) are involved, this conversation may take several meetings and many ad-hoc questions, answers, assertions, requests, and expressives. Appropriate questions have to be answered to not contradict informed consent. Many elements of the process could be covered by declarative notations.

If conflict management is considered, a documented history of speech acts may help in finding contradictions, unfulfilled promises, and patterns for success and failure. Generally, conflict management is knowledge work that typically consists of unstructured ad-hoc tasks. Keeping ones promises, answering requests on time, and avoiding contradictions helps in resolving conflicts. A flexible representation of ad-hoc speech acts enables to actually formulate rules for recommendations, warnings, and strict compliance.

IV. Related Work

Governatori [15] proposes compliance checking on annotated business processes. The *Regorous* approach contains three definitions for compliance: In *fully compliant processes*, all possible executions are compliant. For *partially compliant processes*, it is possible to execute the process without violating any norms. A trace is *weakly compliant* iff every violated norm has been compensated for. To check whether a definition holds for a process or trace, the activities in a model of a business process are annotated with corresponding states. Rules check compliance for the states. The approach is primarily aimed at supporting the design of compliant business processes. Obviously, checking whether a process is fully compliant is not possible if no model of this process is available. Hence, it is not applicable for adaptive case management. However, adding semantic annotations to structured process fragments can still be used for run-time compliance checking. We annotate speech acts instead of states to structured processes, model speech-act-based process templates, and allow knowledge workers to document their interactions [16], [17]. Since inference on speech acts *is* possible, structured and unstructured interactions of the same case can be checked by the same engine.

Koetter et al. [18] propose a generic compliance descriptor to link laws and regulations, compliance requirements, and their implementation. They assume and acknowledge that different compliance techniques are applied in the lifecycles design-time, deployment and run-time, that laws and regulations may change, and that IT implementations change. The assumptions are not completely applicable in adaptive case management, since here these phases often *are* run-time. However, there are different approaches for business rules that are applicable and appropriate for different types of business requirements. Hence, in order to integrate and manage different business rules engines the approach of a generic compliance descriptor is also useful in adaptive case management. Generally, different implementations in actual business landscapes for different types of business rules indicate that Compliance rules containing speech acts should not only be investigated for only one approach. Therefore, we elaborate how extensions for speech-act-based compliance rules can fit in various models.

Sem et al. [7] analyzed various adaptive case management systems used in practice in regard to how they combine flexibility and compliance considerations. Generally, they divide compliance support into before work, *operational compliance*, retroactive proof of compliance, and compliance as a decoupled service. They divide operational compliance requirements for adaptive case management into *guardrails* against violation and assistance in making laws and regulations visible. The measures to ensure compliance can be categorized into making the state of a case visible, making laws and regulations visible, requiring completion of tasks and suggesting tasks. However, advising not to do tasks that most likely are not allowed in a situation, creating warnings for potential violations, or even automatically informing appropriate stakeholders about potential violations seem not to be applied in adaptive case management. In Section VI we include warnings. For integration with adaptive case management we also prefer guidance over guardrails.

For semi-structured processes, the approach to compliance is often compliance by design in declarative notations for flexible business processes. It is based on the idea that one should only model constraints of a process required by law or objectives and derive possible paths from the constraints [19]. Examples for these notations are DCR graphs [19], Declare [20], and CMMN [21]. In these approaches, the constraints are not *line markings*, but strictly enforced. The systems deal with an ad-hoc nature of flexible processes as long as they do not require ad-hoc tasks the BPMS is not aware of. Moreover, even though the actual process models may highly depend on interactions, the systems are not aware of the pragmatic intention. But they could easily be extended with speech act annotations to use their events and artifacts in speech-act-based compliance rules and further facilitate integration.

Dietz [22], [23] analyzes and models business processes and organizations taking Speech Act Theory into account. His methodology can help to identify the deep structure of business processes invariant of the applied technology while unveiling coordination and production acts and their dependencies. The approach is not directly aimed at ensuring Compliance, but it explicitly captures dependencies and encourages modelers to adhere to the workflow loop [24].

V. Inference

In order to define the necessary attributes of speech acts for compliance rules, potential questions to be answered need to be identified.

A. Basic Questions

The questions or patterns in this section are a non-exhaustive list derived by the motivational scenarios.

For patient information and to facilitate informed consent, questions of patients need to be answered. The illocutionary forces for questions and answers do not need to be unique neither in a predefined speech act library nor in practice. Therefore this basic pattern already requires to express a set of related illocutionary forces. Moreover, questions need to be *followed* by answers, hence sometimes the order of speech acts is significant. After a patient information, the physician is also required to gain a signature for informed consent in which the patient asserts the understanding of the physician's explanation. Therefore, checking the existence of a specific speech act is also required.

For the quality management process that acts to new laws and regulations in hospitals, there are also many requirements: All appropriate stakeholders need to be informed, all processes should immediately become compliant after new regulations are enacted or come into force, questions of stakeholders on the new regulations need to be clarified, feedback (e. g. warnings, complaints) of physicians needs to be considered and dealt with properly. These rules need to query speakers and hearers, different types of illocutionary forces, and also the order of acts.

For conflict management, possible questions are which promises and requests are unfulfilled, whether stakeholders have been informed, which offers had been accepted or declined, and whether contradictions of utterances could be unveiled. Moreover, some assertions might be based on other people or institutions, e. g. "John warned that…", so for inference one could rely on the $F(P)$ framework.

B. Representation of Speech Acts

A speech act can be represented by speaker, hearer, illocutionary force, an optional timestamp, propositional content, context, and optional preparatory conditions.

Since an illocutionary force consists of seven components that are probably difficult to impossible to correctly specify for a domain, but easy to understand by domain experts, we propose to use domain-specific speech act libraries in which illocutionary forces are named and categorized into the five illocutionary points. Moreover, an entry should include a flag indicating whether the propositional content is asserted. For example, complaining about X often indicates X [10]. And since many illocutionary forces to be deployed in a speech act library might be similar, e. g. questions in patient information and administrative questions, or probably even templates with further restrictions or rather preset values of speech acts, they should have a relationship adhering to the Liskov Substitution Principle [25] in order to allow addressing generic illocutionary forces as well as highly specific subtypes regardless of how the speech act library evolves in the future.

The timestamp of an utterance or speech act is useful to check whether two coordination or production acts were performed in the right order. However, especially for speech acts that were gathered implicitly (e. g. "He promised that …") the actual time or performance may be unknown. The propositional content consists of logically connected coordination and production acts and adheres to the $F(P)$ framework. The illocutionary force may impose restrictions on the propositional content. For simplicity, the propositional content can be built from free-form text, the prototype of a coordination or production act, or the reference to a coordination or production act, as well as logical connectives $\neg$, $\vee$ and $\wedge$ on these (composite) elements. Moreover, a set of key-value annotations may be added to represent values required by individual illocutionary forces, for example a deadline for a promise. The context is represented by the case or process in which the speech act has been performed. One should optionally be able to attach the utterance which performed the speech act.

C. Heuristics for Inferences

This section introduces basic inferences for speech acts to be used in the business rules engines. Since data will be missing in actual processes as well as in the representation of illocutionary forces, and the general representation of speech acts was simplified, these formulas are heuristics.

$$\tau(s,F) \hat{=} \text{illocutionary force of } s \text{ is subtype of } F \tag{1}$$

$$\hat{=} \tau(\text{force}(s), F) \tag{2}$$

$$\tau(f,F) \hat{=} \text{illocutionary force } f \text{ is subtype of } F \tag{3}$$

$$\hat{=} \begin{cases} true & , f = F \\ false & , f \neq F \wedge \neg\exists \text{parent}(f) \\ \tau(\text{parent}(f), F) & , f \neq F \wedge \exists \text{parent}(f) \end{cases} \tag{4}$$

$$s \models X \hat{=} s \text{ implies } X \text{ (} X \text{ is speech act or other fact)} \tag{5}$$

$$\hat{=} (s = X) \vee (\text{isAsserting}(s) \wedge \text{content}(s) \models X) \tag{6}$$

The first rule relates illocutionary forces of speech acts to super classes according to the Liskov Substitution Principle (LSP). The value of the expression $\tau(s, F)$ indicates whether the illocutionary force of speech act s is of type F, e. g. whether a specific utterance during patient information is assertive or whether an utterance is *some* type of question if many illocutionary forces for questions with varying degree of strength were modeled. Similarly, $\tau(f, F)$ indicates whether the illocutionary force f is of type F. The function parent(f) returns the direct super class of an illocutionary force f, e. g. parent(parent($weak\_question$)) = parent($question$) = $directive$. The three rules of $\tau(f, F)$ adhere to the LSP.

The second heuristic for $s \models X$ is simplified and depends on the representation of propositional content (e. g. whether $\vee$, $\wedge$, $\neg$ and other elements are supported) and the desired maximum computational effort. A simple heuristic is to find a complete X within the propositional content and to ensure that X only has assertive parents in conjunctive combinations without negation. For many questions, this simple heuristic will suffice.

Since uncertainties are involved, often the solution process is also relevant to reasoning. For example, while an inference engine has no problem to prove X relying on a nested set of "P_1 asserted that P_2 asserted that ... P_n asserted X" statements, a judge probably will depending on n and the P_i involved. Therefore, heuristics might need to be adapted to the actual domain and might need to emit a proof.

Temporal relationships between artifacts and events already are supported in most business rules and complex event processing engines. These existing systems can be used by either calling a speech-act-based heuristic in a rule or emitting annotated results from heuristics as events. For references between speech acts or more advanced heuristics, the engines may need to be extended to natively support speech acts.

VI. Integration

This section outlines how speech-act-based inference can be integrated into LTL and CTL rules, Business Rules represented by Drools, and complex event processing.

A. LTL

LTL formulas [26] are commonly used to check compliance of processes [27]. At design-time of a process and with a finite number of speech acts, extended LTL rules with selectors on speech acts can be translated to pure LTL rules.

The set Σ is the finite set of potential speech acts of a process. The function δ translates an extended LTL rule. The function $\sigma(\alpha)$ searches for some speech act in Σ and is true for every speech act s where $s \vDash \alpha$. The propositional variables AP need to be extended by Σ to indicate whether a speech act has been performed, i. e. $AP' = AP \cup \Sigma$, while $AP \cap \Sigma$ should be empty.

$$\delta(p) = p, p \in AP \tag{7}$$

$$\delta(\sigma(\alpha)) = \begin{cases} p \wedge \neg p & l = \varnothing, p \in AP' \\ s & l = \{s\} \\ s_1 \vee ... \vee s_n & l = \{s_1, ..., s_n\} \end{cases} \tag{8}$$

$$\text{for } l = \{s | s \in \Sigma \wedge s \vDash \alpha\} \tag{9}$$

$$\delta(\neg\varphi) = \neg\delta(\varphi) \tag{10}$$

$$\delta(\varphi \vee \psi) = \delta(\varphi) \vee \delta(\psi) \tag{11}$$

$$\delta(\mathrm{X}\varphi) = \mathrm{X}\delta(\varphi) \tag{12}$$

$$\delta(\varphi \text{ U } \psi) = \delta(\varphi) \text{ U } \delta(\psi) \tag{13}$$

The function δ may call itself recursively only on smaller formulas, i. e. δ terminates. Only formula 8 actively translates the filtering of speech acts to LTL by disjunction of potential speech acts or rather their propositional variables. Therefore, at least at design-time of structured processes, LTL rules extended by filtering of speech acts can be translated to pure LTL rules to check for compliance. Obviously, for integration with arbitrary speech act histories in a case that cannot be stated a priori, this translation is not possible. Also, references going beyond just filtering is not possible if only $s \vDash \alpha$ is checked. For example, the pool Quality Management Implementation of Fig. 1 has to ensure that after every incoming requirement (e. g. $s_1 \vDash \alpha_1$) the physician can assert that his processes are compliant (e. g. $s_2, s_3 \vDash \alpha_2$). The extended rule φ could be formulated by globally asserting that if there is a requirement, there eventually needs to be the assertion (with finally $\psi = F\psi =$ true U ψ, globally $\psi = G\psi = \neg F \neg\psi$):

$$\varphi = G(\sigma(\alpha_1) \rightarrow F\sigma(\alpha_2)) \tag{14}$$

$$\delta(\varphi) = G(s_1 \rightarrow F(s_2 \vee s_3)) \tag{15}$$

However, an analogous translation can be performed on LTL-FO$^+$ [28], which extends LTL with first-order-logic to quantify data inside traces of sent messages. For example, if someone promises (X) to perform speech act Y in the future, he should actually perform it. In an extended variant of LTL-FO$^+$ this can be written as $\forall_\Sigma x.\sigma(x, \alpha) \rightarrow (\exists_\Sigma y.\sigma(promise(x, y)) \wedge F\ y)$, while σ still can be translated statically for a finite Σ.

In summary, LTL formulas could be used to describe and check temporal relationships of speech acts. While existing tools could be used for checking compliance a priori with a finite set of all potential interactions, this is no longer true for run-time checking and arbitrary ad-hoc interactions.

B. CTL

CTL formulas [29] are also used in model checking. Similar to LTL, speech-act-based CTL formulas that are checked at design-time of a process and only need to cover a finite set of speech acts can also be translated to pure CTL formulas.

Again, the set Σ is the finite set of potential speech acts of a process, the function δ translates an extended CTL rule to pure CTL, the function $\sigma(\alpha)$ searches for all speech acts s in Σ where $s \vDash \alpha$. Also, the propositional variables AP are extended by Σ, i. e. $AP' = AP \cup \Sigma$ and $AP \cap \Sigma = \emptyset$. The following formulas characterize δ:

$$\delta(p) = p, p \in AP \tag{16}$$

$$\delta(\sigma(\alpha)) = \begin{cases} p \wedge \neg p & l = \varnothing, p \in AP' \\ s & l = \{s\} \\ s_1 \vee ... \vee s_n & l = \{s_1, ..., s_n\} \end{cases} \tag{17}$$

$$\text{for } l = \{s | s \in \Sigma \wedge s \vDash \alpha\} \tag{18}$$

$$\delta(\neg\varphi) = \neg\delta(\varphi) \tag{19}$$

$$\delta(\varphi \wedge \psi) = \delta(\varphi) \wedge \delta(\psi) \tag{20}$$

$$\delta(AX\varphi) = AX\delta(\varphi) \tag{21}$$

$$\delta(EX\varphi) = AX\delta(\varphi) \tag{22}$$

$$\delta(A[\varphi \text{ U } \psi]) = A[\delta(\varphi) \text{ U } \delta(\psi)] \tag{23}$$

$$\delta(E[\varphi \text{ U } \psi]) = E[\delta(\varphi) \text{ U } \delta(\psi)] \tag{24}$$

The function δ translates extended CTL recursively to pure CTL, while only Formula 17 actively translates speech acts to propositional variables. In summary, CTL formulas can also be applied to check processes for compliance in a speech-act-based environment.

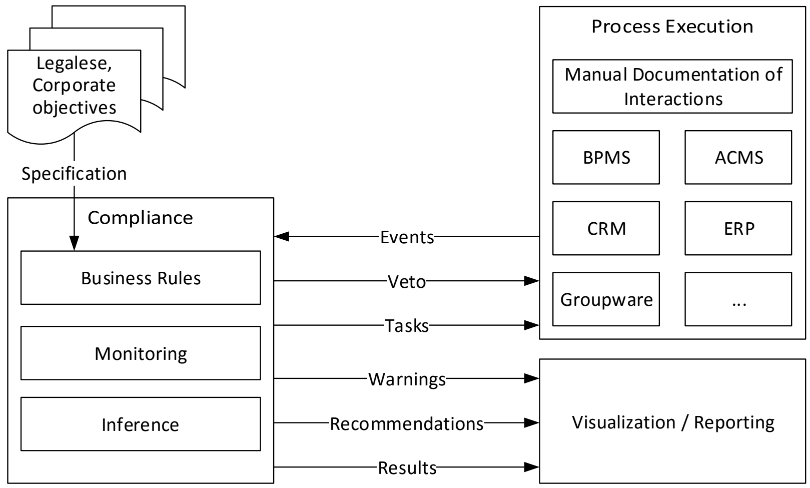

Fig. 2: Speech-Act-Based Compliance and Business Rules

C. Business Rules

A business rule typically consists of a condition over various entities and some resulting action. For performance, business rules engines rely on the Rete algorithm. Business rules can be used to check compliance, e. g. to set some value on whether it passed a constraint or to report breaches of compliance. Moreover, business rules can also contain actions that support compliant behavior, e. g. emit warnings or recommendations, open tasks, and actively reminding to prevent compliance problems from happening. Finally, some business rules may act like an agent to automate some specific aspect of work.

All of these types of business rule can already support inference on speech acts, if they allow calling functions (i. e. an inference engine) in their conditions. These function calls need to check constraints on the illocutionary force and implied facts from propositional content. JBoss Drools[2] allows calls to getters and setters as well as arbitrary function calls and was therefore used to check whether formulating and deploying speech-act-based business rules is possible.

Since determining whether arbitrary formulas are valid is undecidable for many logical systems, one has to either restrict inference to decidable logical systems or apply heuristics. We do not restrict the approach for inference, but for the implementation of Drools examples in Section VIII, we rely on heuristics for the propositional content of a speech act and on relations among illocutionary forces that adhere to the Liskov Substitution Principle.

D. Complex Event Processing

In complex event processing (CEP), “incoming information items are viewed as notifications of events happening in the external world” [30] and used to detect higher-level events. CEP can be considered an enabling technology for compliance monitoring [27].

[2]http://www.drools.org

Drools Fusion provides CEP capabilities with events as first class citizens, pattern matching, temporal constraints, temporal relationships, sliding windows, and more [31]. The examples in Section VIII do not show all of these features. Esper [32] can be considered the leading open source CEP provider [30]. Similar to drools, events in Esper can be Java objects and method invocation in EPL queries is allowed [32]. Hence, Esper may also call an inference engine in a query and does not need to be aware of speech acts at all to support them.

In summary, CEP systems can easily be used in a speech-act-aware environment.

VII. Architecture

Regardless of what type of speech-act-based engine is used, the proposed architecture can be summarized by Figure 2. Regulatory norms as well as corporate objectives are translated into monitoring rules that check the actual compliance to these requirements in order to emit vetoes or report potential compliance breaches, and to action rules that may automate interactions, modify artifacts, or emit tasks, warnings, or recommendations. For process execution, we expect that many systems are involved, for example BPMSs for structured processes, adaptive case management systems (ACMS) to support ad-hoc tasks and interactions, CRM and ERP systems, as well as various collaboration tools including Groupware. Not all of these systems will be aware of speech acts which may result in additional rules for mapping or more complex rules to cover the respective events and artifacts. Moreover, warnings, recommendations, and various auditing results need to be visualized.

For incoming speech acts, the BRMS also needs two modes of operation: One that checks in advance whether a speech act or a pattern of past interactions is compliant to regulations and objectives in order to provide vetoes, and a second one that is free to act on potentially incompliant interactions that are documented because they were already performed. Allowing - but reporting and acting on - incompliant behavior is necessary

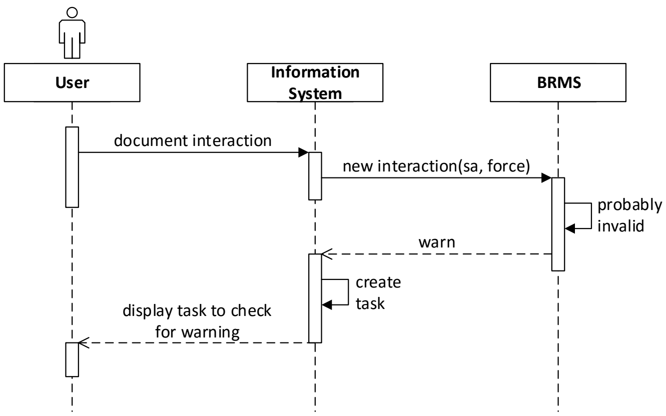

Fig. 3: Implement line markings instead of guardrails

since both systems and most importantly knowledge workers will not always accept vetoes. Not all systems are able to act on vetoes and cannot cancel damage that has already been done. And knowledge workers will only accept and probably welcome *line markings* that guide them through warnings and recommendations [33], but do not enforce behavior. Moreover, systems and knowledge workers may decide to perform incompliant interactions regardless of whether they were warned, and the system should continue being able to support and monitor the process.

Figure 3 is an example sequence for a knowledge worker documenting an interaction he has already performed, for example a recommendation or request during patient information. The information system the interaction is documented in passes the speech act sa to the BRMS and adds the `force` flag that differentiates both modes. With the `force` flag, the BRMS is free to act on this interaction, for example to perform compensating actions or inform appropriate stakeholders. In this case one heuristic identified a potential problem and warns the information system that a knowledge worker needs to decide the appropriate course of action. The information system creates a task and actually warns the user.

Figure 4 shows the behavior in the case of many systems being involved. In the following fictional scenario, the information system for patient information is connected to a BPMS of a hospital administration as well as an ACMS of a pediatric ward. A physician prepares for a meeting and documents that he intends to recommend some specific treatment. The information system is aware of its past behavior, but it may not be aware of the case history in the BPMS and ACMS and should not need to be. It therefore first asks the BRMS whether the interaction to be performed is compliant behavior. After the BRMS confirms this, the information system performs the speech act, and the BRMS continues checking rules that are allowed for the `force` flag. It fires a rule that informs the ACMS which performs some actions using micro processes [16]. One automated reaction to inform parents or legal guardians is performed, but it shows invalid behavior, e. g. informing stakeholders may not be allowed for all diseases. The ACMS is aware of the BRMS capability to check behavior before it happens and can therefore cancel this interaction. The BRMS also informs the BPMS that is able to complete some task for the recommendation step. The last fired rule generates some recommendation of common related steps for the physician that is transferred to the information system. Finally, the information system displays recommendations and potentially results of the BPMS and ACMS.

In summary, this architecture acts proactive preventing incompliant behavior, and retroactive based on warnings, recommendations, reports, and compensating actions to minimize damage.

VIII. Examples

The goal of speech-act-based compliance is primarily to support ad-hoc interactions and integration of different process support systems. Hence, in examples we focus on run-time rules. JBoss Drools represents Rete-based BRMSs and also provides complex event processing. Therefore, the following examples were implemented and tested in JBoss Drools. Additionally, one query is formulated in Esper. We provide some examples checking whether the course of action is valid,

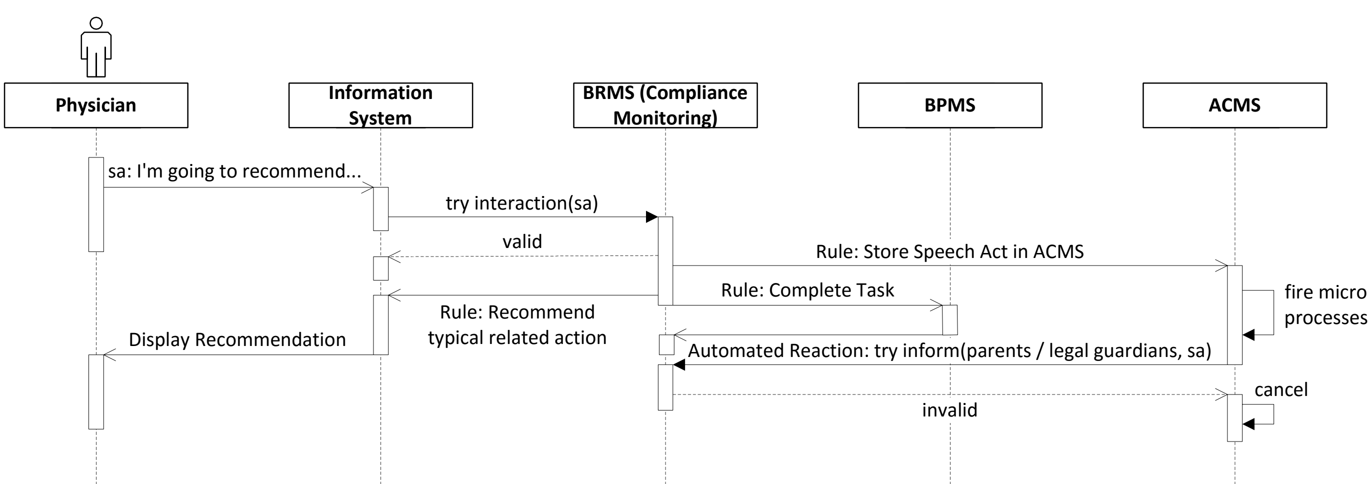

Fig. 4: Enforce rules prior to performing incompliant behavior

how one could formulate warnings, and some action rules to automate behavior.

A. *Checking compliance*

Listing 1 tries to find potentially incompliant behavior for the motivating example of patient information and consent. Physicians have to act on questions by their patients, and as long as they have not performed this, the behavior is incompliant. First, the rule matches on speech acts that are some sort of question. They can either be a `question` or some more detailed subtype. The rule also tries to match a speech act of the commissive illocutionary point that indicates the question has been resolved. The propositional content needs to indicate the question to properly match question and answer or question and resolving the question somehow.

```
rule "Check for unresolved questions"
  when
    $question : SpeechAct(
      force.withinConstraint(
          "directive.question"))
    not (SpeechAct (
      content.heuristicImplies($question)
      && force.withinConstraint(
          "commissive.resolved")))
  then
    Compliance.notValid($question);
end
```

Listing 1: Marking unanswered questions as invalid

The illocutionary force is of type commissive to also cover inappropriate or completely unrelated questions that a physician may reject. If he properly answers the question, some other rule may fire the `commissive.resolved`, and otherwise he asserts to the system that he appropriately acted on the question. If the second speech act cannot be matched, the rule fires the action to mark the question as invalid.

Since between documenting a question and acting on it the question is marked invalid, some rule has to set it back to compliant behavior. Listing 2 performs this task. If more than one rule checks compliance on questions, a single valid flag does not suffice.

```
rule "Question resolved"
  when
    $question : SpeechAct(
      force.withinConstraint(
          "directive.question"))
    $answer: SpeechAct (
      content.heuristicImplies($question)
      && force.withinConstraint(
          "commissive.resolved"))
  then
    Compliance.valid($question);
end
```

Listing 2: Marking answered questions as valid

B. *Warnings and recommendations*

Listing 3 warns about unfulfilled promises. This rule is especially for the motivational examples of conflict management and quality management. All interactions representing promises are matched. If no speech act indicating that the promise were kept is documented or can be detected properly, the BRMS emits a warning for the promise. Recommendations can be formulated in the same way.

```
rule "Warn about unfulfilled promises"
  when
    $promise : SpeechAct(
      force.withinConstraint(
          "commissive.promise"))
    not (SpeechAct (
      content.heuristicImplies($promise)
      && force.withinConstraint(
          "commissive.keptPromise")))
  then
    Compliance.warn($promise);
end
```

Listing 3: Warning about unfulfilled promises

Listing 4 is formulated in Esper and emits all questions that have not been answered within three days.

```
select Q
from pattern[
  every Q=SpeechAct ->
    (timer:interval(3 days)
    and not SpeechAct(Answering = Q.ID))]
where
  Q.withinConstraint("directive.question")
  and Q.Speaker = Q.getCase().getPatient()
```

Listing 4: Find all questions not being answered in three days

C. *Action rules*

Finally, action rules of an information system for conflict or quality management matches ad-hoc tasks to documented or future speech acts. Two rules should automate the appropriate documentation, i. e. if the system is aware that the planned interaction has been performed, it can close the respective task, and if the knowledge worker indicates that he performed the task, the system may document the appropriate speech act.

```
rule "Close Task after Speech Act"
  when
    $t: Task(status == Status.OPEN,
            PrototypeAct != null)
    SpeechAct(
      $t.PrototypeAct.implies(this))
  then
    $t.status = Status.CLOSED;
    insert(new CloseTaskEvent($t));
end
```

Listing 5: Closing a task after its annotated interaction or a speech act implying the prototype was performed

If the rule of Listing 5 matches an open task indicating some planned interaction (`PrototypeAct != null`) and a speech act representing (or implying) the planned interaction, it closes the task and emits an event for other rules.

Listing 6 documents the speech act implied by the closing of a task if no documented speech act already represents or contains its planned interaction. The task must provide some prototype to construct the documentation.

```
rule "Document Speech Act of a Task"
  when
    $e: CloseTaskEvent (t : Task )
    not (SpeechAct (
      t.PrototypeAct.implies(this)
    ))
  then
    SpeechAct sa = t.PrototypeAct.build();
    sa.setSource("Business Rules Engine");
    t.getCase().addTask(sa);
    insert(sa);
end
```

Listing 6: Documenting speech act after a task promising the act was closed

IX. Discussion

First of all, introducing Speech Act Theory into business rules increases expressibility of a BRMS and can improve the translation of legal requirements and corporate objectives that involve interactions into daily business. The approach allows to formulate constraints on the *pragmatic intention* of interactions, e. g. on commitments, directives, and even apologies, to find unfulfilled promises, pending interactions, and to check for patterns of interaction required by law.

As shown in Section VI, existing approaches on compliance checking and business rules can easily be extended to support Speech Act Theory. Design-time checks with speech-act-based LTL or CTL rules can be translated to pure LTL and CTL to apply existing solutions. Run-time checks in typical Rete-based BRMSs or CEP systems either support calling methods for inferencing on illocutionary forces and propositional contents, or need to be extended to call those engines. The approach is to improve the matching on interaction events and this only requires more sophisticated selectors, but not to introduce a completely different model.

Of course, our approach does not really improve design-time checking. If all potential speech acts are known in advance, which is required for our translation functions for LTL and CTL rules, then the constraints would probably be formulated on these known events and artifacts. But we primarily intend our approach to not be restricted to a priori checks, we want to support knowledge work and facilitate the integration of structured, semi-structured, and ad-hoc processes.

Here, the focus on interactions offers a sensible mapping of artifacts originated from various systems and manual documentation to a unified and semantically enriched format. Ly et al. [27] already suggest that due to the heterogeneity of data sources required in compliance monitoring, an integrated target event format is desirable, which our approach offers at least for interactions. This allows to choose for each case the process support system that best fits the current case or even orchestrate various systems. Moreover, reasoning on semantically annotated artifacts or processes facilitates searching for cross-cutting concerns [34], and thereby checking compliance on features that are characterized by being spread across many processes and systems. Moreover, in realistic settings one cannot always assume the existence of a complete process model [27], but for incomplete process models organizations also need to ensure compliance to laws and regulations.

The approach also allows to integrate autonomous agents that support processes and tasks but are not visible in any process model. For example, Kalia et al. [35] automatically identify tasks and commitments in email and chat conversations based on Speech Act Theory, natural language processing and machine learning. Since their evaluated precision for realistic conversations is 90% in email and 80% in chat conversations, this could help in automatically gathering actual work that would not be properly documented and also check for potentially incompliant behavior.

Knowledge workers are also supported by not strictly enforcing rules, but rather introducing them as *line markings*. Knowledge workers will not accept vetoes that prohibit them from their work. When necessary, they will find a workaround or not use the support systems at all. After all, in his area of expertise the knowledge worker will always have a better understanding of the operations than deployed business rules. This expertise does not prevent small oversights, but a warning, a recommendation, or a preventive *"Really?"* may help.

X. Conclusion

In this paper, we introduced Speech Act Theory into Compliance and business rules in particular. A focus on the pragmatic intention of interactions offers to check compliance for processes requiring ad-hoc interactions and processes scattered across many systems, i. e. cross-cutting concerns and knowledge work.

We show that inferencing on interactions is already possible in BRMSs, at least for those based on LTL and CTL rules, typical Rete-based BRMSs and complex event processing. LTL and CTL rules containing selectors for speech acts can be translated to pure LTL and CTL. Rete-based BRMSs and CEP engines that allow method invocation can easily import inference engines for the pragmatic intention to improve selectors. For temporal relations, they should continue to use their own means. Moreover, we introduce an architecture in which compliance facilitates the integration of structured, semi-structured and ad-hoc processes. No process schema is necessary. Interactions and common events emitted by the many systems involved are processed by the BRMS to emit tasks, warnings, recommendations, and *vetoes*. Knowledge workers and participating systems are not required to understand vetoes and are also allowed to ignore some if they do. Vetoes act as line markings and may help in cancelling incompliant behavior before damage is done. If knowledge workers can rely on compliance monitoring to cancel and report rogue automation, they can provide simplified automated process fragments without worrying about all possible exceptions.

Introducing Speech Act Theory into compliance requires that the participating systems emit at least events and artifacts to a business rules management system. Events and artifacts need to contain some process or case identifier to relate associated actions and interactions. Some mapping of interactions of an arbitrary system-specific format to a speech act has to be implemented. This can happen directly in the system or has to be performed in the BRMS or an intermediary mapping tool.

Still, there are open questions. Inference may happen in many ways and we presented some simple heuristic that may need to be adapted for different domains. Moreover, the approach requires to document the pragmatic intention of interactions, and for structured processes (e. g. BPMN and CMMN) they need to be mapped or annotated. The architecture is only useful if knowledge workers actually decide to accept documenting their interactions, and this requires systems with a high usability and low effort to distinguish illocutionary forces. Some additional benefit should be provided, e. g. automatically creating documents out of user-defined templates and case data.

In summary, a landscape of speech-act-aware business process support systems and especially speech-act-based compliance offers to improve the support and productivity of knowledge work and the integration of the many systems and approaches involved while translating legal requirements into daily business.

References

[1] T. H. Davenport, *Thinking for a Living: How to Get Better Performances and Results from Knowledge Workers*. Harvard Business Press, 2005, ch. What's a Knowledge Worker, Anyway?, pp. 1–24.

[2] P. F. Drucker, "Knowledge-worker productivity: The biggest challenge," *California Management Review*, vol. 41, no. 2, pp. 79–94, 1999.

[3] K. D. Swenson. (2015, May) Robots don't innovate - innovation vs automation in BPM.

[4] S. Lund, J. Manyika, and S. Ramaswamy, "Preparing for a new era of knowledge work," *McKinsey Quarterly*, vol. 4, pp. 103–110, 2012.

[5] M. Berg, J. Aarts, J. van der Lei *et al.*, "Ict in health care: sociotechnical approaches," *Methods Archive*, vol. 42, no. 4, pp. 297–301, 2003.

[6] T. Winograd and F. Flores, *Understanding Computers and Cognition: A New Foundation for Design*. Ablex Publishing Corporation, 1986, ch. Using computers: A direction for design, pp. 163–179.

[7] H. F. Sem, S. Carlsen, and G. J. Coll, "Combining compliance with flexibility," in *Thriving on Adaptability*, L. Fischer, Ed. Future Strategies Inc., 2015, pp. 59–71.

[8] J. L. Austin, *How to do things with words*. Oxford university press, 1975.

[9] J. R. Searle, *Speech acts: An essay in the philosophy of language*. Cambridge university press, 1969.

[10] J. R. Searle and D. Vanderveken, *Foundations of illocutionary logic*. Cambridge University Press, 1985.

[11] W. Brennenstuhl and T. T. Ballmer, *Speech act classification - A Study in the Lexical Analysis of English Speech Activity Verbs*, W. Levelt, Ed. Springer-Verlag Berlin Heidelberg, 1981.

[12] M. Schoop, "Habermas and searle in hospital: A description language for cooperative documentation systems in healthcare," *Proceedings of the Second International Workshop on Communication Modeling - The Language/Action Perspective*, 1997.

[13] Verpflichtung der Leistungserbringer zur Qualitätssicherung. [Online]. Available: http://www.gesetze-im-internet.de/sgb_5/__135a.html

[14] Gemeinsamer Bundesausschuss, "Qualitätsmanagement-Richtlinie vertragsärztliche Versorung - ÄQM-RL," *BAnz AT 09.12.2015 B4*, 2015. [Online]. Available: https://www.g-ba.de/informationen/richtlinien/18/

[15] G. Governatori, "The regorous approach to process compliance," in *Enterprise Distributed Object Computing Workshop (EDOCW), 2015 IEEE 19th International*, Sept 2015, pp. 33–40.

[16] J. Tenschert and R. Lenz, "Supporting knowledge work by speech-act based templates for micro processes," in *AdaptiveCM 2015–4th International Workshop on Adaptive Case Management and other non-workflow approaches to BPM*, 2015.

[17] ——, "Towards speech-act-based adaptive case management," in *AdaptiveCM 2016–5th International Workshop on Adaptive Case Management and other non-workflow approaches to BPM*, 2016.

[18] F. Koetter, M. Kochanowski, A. Weisbecker, C. Fehling, and F. Leymann, "Integrating compliance requirements across business and it," in *Enterprise Distributed Object Computing Conference (EDOC), 2014 IEEE 18th International*, Sept 2014, pp. 218–225.

[19] M. Marquard, M. Shahzad, and T. Slaats, "Web-based modelling and collaborative simulation of declarative processes," in *Business Process Management*, ser. Lecture Notes in Computer Science, H. R. Motahari-Nezhad, J. Recker, and M. Weidlich, Eds. Springer International Publishing, 2015, vol. 9253, pp. 209–225.

[20] M. Westergaard and F. M. Maggi, "Declare: A tool suite for declarative workflow modeling and enactment," in *Proceedings of the Demo Track of the Nineth Conference on Business Process Management 2011, Clermont-Ferrand, France, August 31st, 2011*, 2011.

[21] Object Management Group, *Case Management Model and Notation (CMMN)*, Std. formal/2014-05-05, Rev. formal/2014-05-05, 2014. [Online]. Available: http://www.omg.org/spec/CMMN/1.0

[22] J. L. G. Dietz, *Enterprise Ontology: Theory and Methodology*. Springer, May 2006.

[23] ——, "The deep structure of business processes," *Commun. ACM*, vol. 49, no. 5, pp. 58–64, May 2006.

[24] R. Medina-Mora, T. Winograd, R. Flores, and F. Flores, "The action workflow approach to workflow management technology," in *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work*, ser. CSCW '92. New York, NY, USA: ACM, 1992, pp. 281–288.

[25] B. H. Liskov and J. M. Wing, "A behavioral notion of subtyping," *ACM Trans. Program. Lang. Syst.*, vol. 16, no. 6, pp. 1811–1841, Nov. 1994.

[26] A. Pnueli, "The temporal logic of programs," in *Foundations of Computer Science, 1977., 18th Annual Symposium on*. IEEE, 1977, pp. 46–57.

[27] L. T. Ly, F. M. Maggi, M. Montali, S. Rinderle-Ma, and W. M. van der Aalst, "Compliance monitoring in business processes: Functionalities, application, and tool-support," *Information Systems*, vol. 54, pp. 209 – 234, 2015.

[28] S. Halle and R. Villemaire, "Runtime monitoring of message-based workflows with data," in *Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE*. IEEE, 2008, pp. 63–72.

[29] E. M. Clarke, E. A. Emerson, and A. P. Sistla, "Automatic verification of finite-state concurrent systems using temporal logic specifications," *ACM Trans. Program. Lang. Syst.*, vol. 8, no. 2, pp. 244–263, Apr. 1986.

[30] G. Cugola and A. Margara, "Processing flows of information: From data stream to complex event processing," *ACM Comput. Surv.*, vol. 44, no. 3, pp. 15:1–15:62, Jun. 2012.

[31] The JBoss Drools team, "Drools documentation," 2016. [Online]. Available: https://docs.jboss.org/drools/release/6.4.0.Final/drools-docs/html_single/#DroolsComplexEventProcessingChapter

[32] EsperTech Inc., "Esper reference," 2016. [Online]. Available: http://www.espertech.com/esper/release-5.4.0/esper-reference/html/index.html

[33] T. Shepherd, *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*. Meghan-Kiffer Press, 2010, ch. Moving from Anticipation to Adaptation, pp. 41–62.

[34] C. Di Francescomarino, C. Ghidini, M. Rospocher, L. Serafini, and P. Tonella, "Reasoning on semantically annotated processes," in *Service-Oriented Computing–ICSOC 2008*. Springer, 2008, pp. 132–146.

[35] A. Kalia, H. R. M. Nezhad, C. Bartolini, and M. Singh, "Identifying business tasks and commitments from email and chat conversations," Citeseer, Tech. Rep., 2013.