Towards Speech-Act-Based Adaptive Case Management

Johannes Tenschert and Richard Lenz
Institute of Computer Science 6 (Data Management)
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
{johannes.tenschert,richard.lenz}@fau.de

***Abstract*—Knowledge workers already face a broad range of tools to support their work, e. g. adaptive case management systems, tailored information systems, groupware, and other (process) support systems. Case data is scattered across many systems, and the overlapping structured, semi-structured, and ad-hoc processes involved further impede keeping track of related data and activities. Organizations are socio-technical entities, and interactions have significant impact on their success. Today, around 50% of the work in the US is knowledge work, and other countries show a similar tendency. Improving integration of appropriate tools for knowledge work and augmenting support for interactions therefore offers to increase productivity in a very influential part of the workforce. Knowledge workers are well aware of the pragmatic intention of their communicative acts, but currently their systems are not. We suggest to use Speech Act Theory to enable useful inferences and to improve integration of the various tools for knowledge work. A focus on interactions raises awareness for the pragmatic intention and commitments in particular. It can help providing line markings for knowledge workers by facilitating compliance monitoring for interactions and artifacts stemming from many participating systems and manual documentation. Interactions already tie many separate systems together, and standardizing as well as partially automating them can therefore further simplify integration. Speech-act-based adaptive case management offers to increase process transparency, enable useful inferences, and integrate structured, semi-structured, and ad-hoc processes.**

***Index Terms*—adaptive case management, speech act theory, integration, compliance, knowledge-intensive business process**

I. INTRODUCTION

In the last decades, the share of knowledge work rapidly increased. Knowledge work involves the creation, distribution, or application of knowledge [1], and is furthermore characterized by abstractly defined tasks 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]. In summary, interactions are significant for the success of today's organizations.

Knowledge workers already have a broad range of tools to support their work. They use groupware to manage contacts and emails of their cases. They probably have access to an adaptive case management system (ACMS), some tailored domain-specific information system, or a wiki. They share case-related documents per e-mail or with file synchronization software. And they probably automate some process fragments with task automation services [7] or tailored BPM systems.

Hence, case data often is scattered across many systems and integrating all these tools is often performed by manually updating data. Moreover, structured, semi-structured, and ad-hoc processes often have overlapping responsibilities, and cross-cutting concerns of an organization cannot be attributed to a single BPMS. Therefore, integrating the scattered case data and process support systems is still an open question.

We use Speech Act Theory to classify communicative acts in order to share and process interactions with a standardized representation. We focus on interactions and their intention, since the importance for the case's success would already require to view interactions as first-class entities.

Speech acts have been proposed to improve the design of interactive systems for decades. Early prototypes were isolated applications with limited inferencing capabilities. We propose to use speech-act-based classification for integration of systems. Business rules benefit from a standardized representation of interactions, and facilitate tying together different systems. Compliance monitoring can benefit from emitting warnings and vetoes. If possible, compliance of an interaction is checked prior to its performance. If the system executing the interaction understands vetoes, a rule can act as a guardrail and prohibit the performance. This can facilitate automation of simple tasks, since exceptional cases leading to incompliant behavior are stopped before damage is done. Compliance rules are not strictly enforced. Systems and knowledge workers decide on their own, whether they follow vetoes or act on warnings. Therefore, the rules act like line markings indicating a sensible path but allowing deviant behavior if the knowledge worker deems them necessary. Finally, our approach is intended to integrate structured, semi-structured, and ad-hoc processes which requires the integration of various BPMSs and information systems.

In the following sections, we briefly introduce Speech Act Theory and related work. Section IV explains the contents of a case, the representation of speech acts, and it introduces speech act libraries. Section V outlines the architecture of an integrated ACMS, how line markings could be provided, and how participating systems are integrated. In Section VI we discuss our approach, and in Section VII 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. Some types of interactions adhere to typical patterns, e. g. questions are usually followed by an answer. A speech act consists of an illocutionary force, or rather the intention of the speaker, 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]. Not only utterances are speech acts, but rather all activities with the intention to send a message. The speaker is well aware of this context and of his pragmatic intention, but the systems supporting him currently are not. Searle distinguishes the following five illocutionary points of illocutionary forces:

- **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.

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 [11], and this library can act as a framework to document interactions. In the $F(P)$ framework [9], with the illocutionary force F and propositional content P, $F(P)$ can be the propositional content of some other speech act. For example, $F_2(F_1(P))$ 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. Related Work

In Casebook [12], [13], cases, processes, tasks, and artifacts are first-class entities in a social network environment. The social network contains profiles for both people *and* cases, and – similar to its obvious eponym – it provides activity feeds in these profiles. Cases are social. Case planning is further supported by task extraction [13] that relies on Speech Act Theory, natural language processing, and machine learning to analyze email and chat conversations in regard to commitments and their lifecycle [14]. The system provides recommendations based on the actions in similar cases. Knowledge workers may create case and process templates and by analyzing similar cases, the system suggests improvements for these templates. Even though Casebook emphasizes the social aspects and Speech Act Theory is actually applied for task extraction, the system does not further adopt it.

Cognoscenti [15], [16] is intended to support knowledge workers that need to collaboratively work with sensitive information. Projects or rather cases contain documents, tasks, meetings, notes, and a history. Cognoscenti supports federated case handling. It does not explicitly consider Speech Act Theory for interactions in cases. However, it implements[1] features to support the Sociocracy [17] method for consent-based decision making, which includes the special interactions *Proposal*, *Consent*, and *Object*. Hence, some interactions are classified and their pragmatic intention is used to support documentation of cases and usability by participants. The focus of Cognoscenti is completely on ad-hoc processes and the system therefore does not provide techniques for integration with structured or semi-structured processes. Hence, the main differences between Cognoscenti and the proposed architecture are that the classification of interactions according to Speech Act Theory is supported and encouraged for all interactions, and that speech acts are used to facilitate integration of various structured and completely ad-hoc fragments of a case.

Darwin Wiki [18] is a wiki-based ACMS. Cases consist of wiki pages that are extended with tasks and attributes. Tasks may request mandatory work results to be stored in attributes in order to generally add responsibilities and due dates to attributes. Types of wiki pages are templates to dynamically determine attributes and tasks of a case. Dependencies of tasks in a case type can be modeled via CMMN. Modeling experts may define dependencies and improve templates. End-users need to edit unstructured rich text, perform tasks, and fill attributes. They use predefined work templates, but are not required to know CMMN. The approach does not focus on interactions. Nonetheless, it empowers knowledge workers with limited modeling capabilities and introduces inference opportunities for structured attributes of a case.

The Coordinator [19] is a communication tool that already applied Speech Act Theory in 1987. It supports generating, transmitting, storing, retrieving, and displaying messages that are records of move in conversations [19]. Conversations are divided into the categories conversation for action, clarification, possibilities, and orientation. Conversations contain speech acts such as accept, promise, decline, counter-offer, cancel, and more. The communications tool makes the different types of conversations visible to the user and suggests appropriate types of speech acts to initiate and continue them. It makes useful inferences from the classification of single interactions: It displays missing responses of other participants as well as the current user's pending promises, offers, requests, and commitments with their due dates. Sadly, the approach does not yet cover integration of BPMSs, compliance, and other participating systems. It focuses on interactions and provides guardrails with the suggested interactions for different types of conversations. Medina-Mora et al. [20] introduced the Action Workflow approach that characterizes workflows by identifying and constructing loops of action in which a performer completes an action to the satisfaction of an internal or external customer. The loop proceeds in the phases proposal, agreement, performance, and satisfaction. Any phase might consist of additional actions. Loops can be connected to depict

[1] https://github.com/agilepro/cognoscenti

a hierarchy, interdependencies, and different participants for the customer and performer roles. Loops model coordination among people, and they are invariant to the applied technology.

Dietz [21], [22] analyzes and models business processes taking Speech Act Theory into account. His DEMO methodology considers two types of acts: Coordination acts (speech acts) and production acts. Making this dichotomy of actions explicit allows considering dependencies in coordination and production separately. The methodology can help to identify the deep structure of business processes invariant of the applied technology [22]. It encourages modelers to adhere to the workflow loop [20]. "Practitioners seek combinations and interfaces between DEMO methodology and other methods and techniques" [23]. Our approach intends to provide such an interface: It focuses on speech acts to integrate various BPMSs, ACMSs, BRMSs, and other systems used by knowledge workers to achieve their goals. A system implementing the DEMO methodology could exchange and use speech acts as well as documented production acts.

Bider et al. [24], [25] analyze business process support and ACM systems in regard to the user's speech acts. They propose a framework for analyzing communication models that currently covers general communication capabilities in regard to expressing the illocutionary points for any system, and communication capabilities related to business processes with a more detailed classification of the relevant communication acts [25]. They conclude that the Language/Action perspective is an appropriate tool to analyze communication models of systems. We further see it as an appropriate tool to integrate communication models of systems.

Van der Aalst et al. [26] introduced a framework for lightweight speech-act-based interacting workflow processes based on Petri nets which they call Proclets. Proclets are instances of proclet classes that describe the lifecycle of instances analogous to a workflow schema. Proclets exchange performatives. All performatives are stored in a knowledge base for inferences and decisions. An implementation of the approach could easily exchange performatives with a speech-act-based ACMS and Proclets are therefore considered for integration.

The Knowledge Query and Manipulation Language (KQML) is a language for programs or rather agents to communicate attitudes about information, e. g. querying, offering or subscribing [27], [28]. KQML contains an extensible set of predefined performatives, i. e. speech acts sent between agents. Kimbrough et al. [29] introduced a speech-act-based formal language for business communication (FLBC) in the context of an army office. Both KQML and FLBC support nested messages, i. e. they adhere to Searle's $F(P)$ framework. They both show (i) that electronic and traceable communication can be semantically enriched with speech acts, (ii) that Searle's $F(P)$ framework has practical use, and (iii) that inference on speech acts is useful for integration of different systems and interacting human users.

Hisarciklilar et al. [30] support design communication using Speech Act Theory and Semantic Web annotations. They combine interactions with domain-specific information, which is adopted in our approach. Speech acts may contain arbitrary key-value pairs for annotations, and are optionally linked to various artifacts, i. e. information for a particular domain can easily be added or referenced.

Sem et al. [31] analyzed various adaptive case management systems used in practice in regard to how they combine flexibility and compliance considerations. They divide operational compliance requirements for ACM 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 visible, requiring completion of tasks, and suggesting tasks. Additionally, we try to enable advising not to do tasks that most likely are not allowed, creating warnings and automatically informing appropriate stakeholders about potential violations.

For semi-structured processes, the approach to support flexibility and to ensure compliance is often compliance by design in declarative notations. 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 [32]. Examples for these notations are DCR graphs [32], Declare [33], and CMMN [34]. In these approaches, the constraints are 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 their pragmatic intention. In our architecture, these BPMSs can be integrated into a case. A mapping for speech acts should be provided, and semi-structured processes could share coordination and production artifacts with an ACMS. Since the ACMS does not enforce rules, the restrictions could be leveled off to line markings.

IV. Approach

Depending on the domain, the contents of a case and a representation of speech acts may vary. This section introduces one definition of the contents of a case, a representation for instances of speech acts, and speech act libraries to classify illocutionary forces of speech acts.

A. Case

Each case consists of an ID, a name, an owner, a flag indicating whether the case is active, interactions (speech acts and annotations), contacts, tasks, documents, notes, tags, super and sub cases, related processes, and master data.

Speech acts can be linked to artifacts (e. g. tasks, documents) of a case. The focus of our approach lies in interactions. Hence, a case should contain participating contacts. The model of a contact conforms to the vCard[2] format. A contact also has a role for the case, e. g. "patient", "assigned judge", etc. This role should not be confused with user access control. It is intended to help knowledge workers to manage the people involved in a case regardless of access to the case files. For similar cases and templates, the role can help in automatic

[2]RFC 6350 vCard Format Specification

document generation. Tasks consist of a name, priority, status, due date, and related interactions. Their model is similar to VTODO in the iCalendar[3] format to facilitate interoperability. Documents have a name, potentially multiple versions, related interactions, and annotations. Notes can be appended to the case itself and to many artifacts.

Tags are intended to help knowledge workers classify their cases and for suggesting appropriate related structured processes or process fragments. Super and sub cases as well as related processes of BPMSs also are referenced in a case. Since the master data of a case may vary substantially between cases, we use user-defined key-value annotations. The annotations can be organized in tabs and arranged freely according to the knowledge worker's preference.

B. *Representation of Speech Acts*

A speech act is represented by speaker, hearer, illocutionary force, propositional content, context, an optional timestamp, and optional preparatory conditions. Propositional content consists of logically connected coordination as well as production acts and adheres to Searle's $F(P)$ framework. A set of key-value annotations is added to represent values required by individual illocutionary forces, e. g. deadlines for a promise, and similar to [30] for domain knowledge. One should optionally be able to attach the utterance which performed the speech act, e. g. a reference to the e-mail, or free-form text. The context is represented by the case or process in which the speech act has been performed. The ID should include a hint to the system responsible for the process or case. At least some mapping needs to exist, which could be provided by an URL. For speech acts that were gathered implicitly (e. g. "He promised that..."), the actual time or performance may be unknown. Since an illocutionary force consists of seven components that are 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.

C. *Speech Act Library*

Creating one generic speech act library for all potential use cases and domains is not sensible, since the English language alone contains at least 4800 speech act verbs [35], and the meaning of one verb may differ between domains. Therefore, we propose to use domain-specific speech act libraries. It would be sensible to create one basic library which is extended for different domains to standardize basic inferences. An entry of a speech act library should contain a name (e. g. "promise"), the illocutionary point (e. g. "commissive"), optionally a superordinate illocutionary force, and a flag indicating whether it asserts propositional content.

Inference already yields useful insights with an abstract view on interactions, i. e. some types of similar questions, assertions etc. can be modeled in a hierarchy starting with the five illocutionary points and adhering to the Liskov Substitution Principle [36]. XML Schemata and namespaces show that such ontological commitments in fact are applicable in practice. This allows detailed domain-specific information for users and inference engines as well as generic rules that are easy to reuse. Since for example a complaint about X usually indicates X, a flag indicating whether the speech act is assertive can be useful for inference. Furthermore, templates of interactions can be stored in the library, e. g. to autofill attributes or to relate artifacts.

V. Architecture

The architecture of the speech-act-based ACMS is intended to integrate structured, semi-structured, and ad-hoc processes and tasks to support knowledge work. The system has to manage cases and associated interactions, tasks, documents, and other artifacts, but it also has to interact with existing BPMSs and domain-specific information systems. Figure 1 indicates these interactions with external systems. To our knowledge, except for our prototype there currently are no solutions for speech-act-based business rules [37]. Therefore, inference currently is within the bounds of the ACMS, but in the future it *should* be an external system used by many ACMSs and BPMSs. Speech-act-based micro processes [38] also are placed within the bounds of the ACMS, but this approach on semi-structured processes or rather their structured fragments could also be used by more than one ACMS.

A. *Case Data Store*

The core module of a speech-act-based ACMS is the case data store that manages cases, speech acts, tasks, documents, and other artifacts. Speech acts and tasks are separated from the case store for their particular characteristics: Speech acts document interactions and may consist of other speech acts. The only sensible reason to modify speech acts is to correct errors made in haste during manual documentation. Tasks on the other hand *have* a state indicating the progress that should change until they were performed or aborted. External systems may only have slight knowledge of case-related data, but they might be able to create or modify tasks. The case store links tasks and speech acts with related artifacts of a case. Moreover, it needs to relate cases to arbitrary artifacts and processes in external systems.

B. *Inference and Line Markings*

Inference on case data and speech acts in particular can be divided into three categories: Reminders, compliance, and event-condition-action (ECA) rules. Reminders can support explicitly known due dates of tasks or calendar entries. However, they could also be friendly reminders to potential compliance breaches in the future, e. g. if reaction times have to be fulfilled or a large amount of pending tasks would suggest imminent problems. These reminders not backed by explicitly known dates could for example stem from a BRMS.

Speech-act-based compliance monitoring could improve the translation of legal requirements and corporate objectives that involve interactions into daily business [37]. The interactions gathered in many systems are standardized and rules are

[3] RFC 5545 Internet Calendaring and Scheduling Core Object Specification (iCalendar)

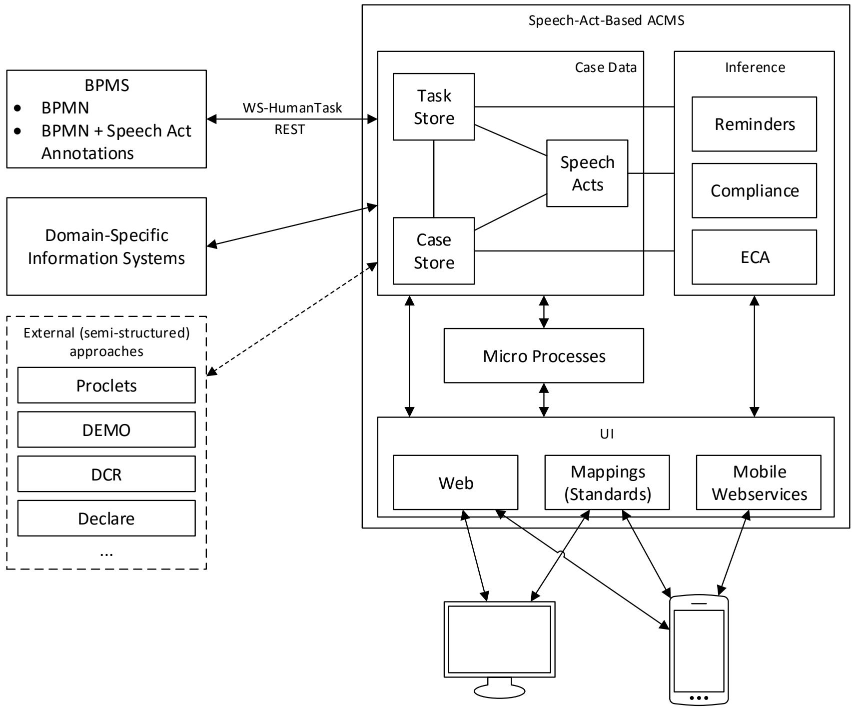

Fig. 1: Proposed Architecture of a Speech-Act-Based ACMS

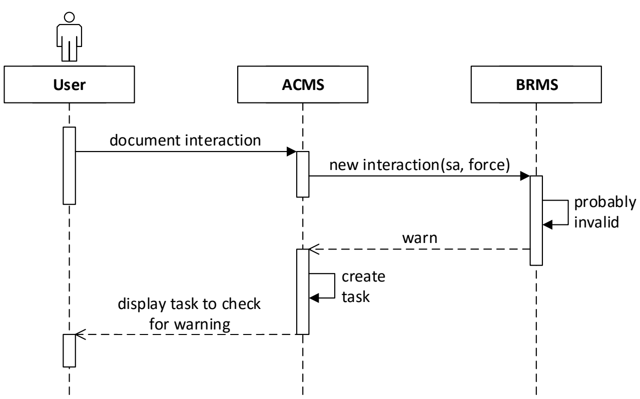

Fig. 2: Implement line markings instead of guardrails

formulated on the pragmatic intention instead of a large amount of similar events. The same could obviously also be achieved with a mapping of similar artifacts to one generic form. For interactions, this generic mapping would most likely lead to groups similar to their appropriate illocutionary force.

Line markings can be provided by compliance monitoring. They guide knowledge workers through the process, but unlike guardrails, line markings are not enforcing behavior. As soon as potential problems are detected, and the relevant interactions have already been performed, warnings can be emitted to the user. A knowledge worker is free to ignore the warning if his approach prevents an even bigger problem or if the rule in question is not applicable for the specific case. Moreover, if interactions are checked for compliance prior to their actual performance, then compliance monitoring could also emit a *veto*. Similar to a warning, knowledge workers and systems are free to ignore the veto. But if a participating system is aware of it, it can stop automation that was modeled for a happy path and is no longer applicable for the specific case, or plainly malfunctioning processes. A knowledge worker can flag the interaction in question to enforce documentation, but he might reconsider if he detects an error on his part. Figure 2 shows an example for line markings. A user documents an interaction he has already performed. His ACMS can therefore only enforce documentation, and hands the speech act to a BRMS for compliance checking. The BRMS detects the interaction may probably be invalid behavior for this case and warns the ACMS which in turn creates a task to resolve the warning. The task can finally be considered by the knowledge worker. The same approach for monitoring with a BRMS can also be applied for ECA rules. Today, knowledge workers often know ECA rules as task automation services [7], and if those rules can be created by technophile knowledge workers and shared with others, they can also be useful to reduce manual tasks and integration efforts.

C. ACM and Groupware

Groupware already improves interactions on desktops, laptops and mobile devices, and should be considered for displaying and modifying tasks, contacts, appointments, and notes of a case. There are commonly used web standards for those artifacts, e. g. vCard, iCalendar and the protocols CardDAV and CalDAV. Knowledge workers today typically use more than one device. The UI therefore needs to consist of a mapping to some web standards for artifacts, a web interface for desktop operating systems, and a web interface as well as web services particularly useful for mobile devices.

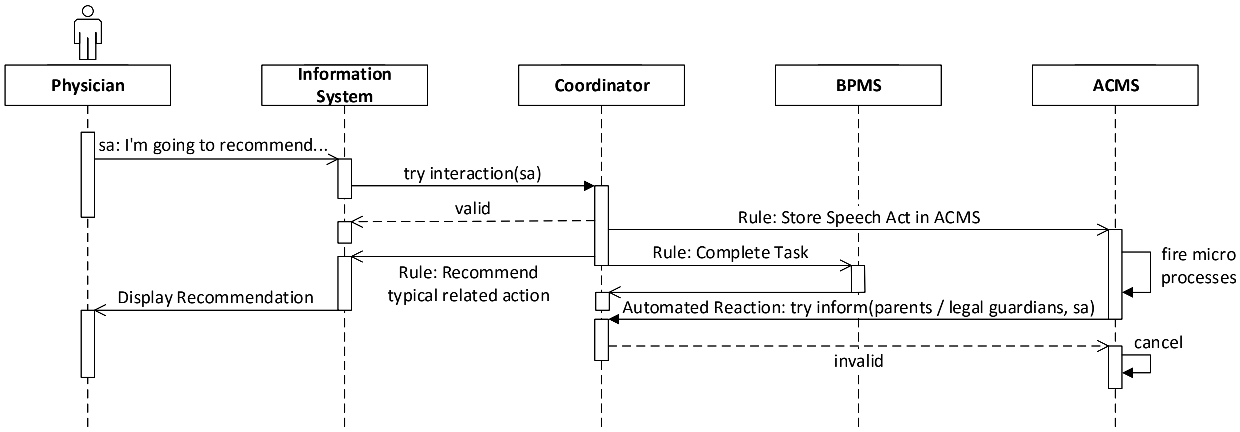

Fig. 3: Integration via ECA rules

D. Integration of Processes and Systems

Typically, BPMSs offer some API, e. g. a proprietary REST interface or WS-HumanTask [39], to share process status information or human tasks between systems. In order to integrate speech-act-aware BPMSs with our approach, it will be necessary to implement individual interfaces or a mapping from artifacts to speech acts.

One coordinator should take care of forwarding production and coordination artifacts between the systems involved. Moreover, the coordinator should provide the veto and warning mechanism for line markings. Both a speech-act-aware BRMS and ACMS are suitable to act as the coordinator. Figure 3 shows how ECA rules could tie different systems together. The fictional scenario of an information system for patient information, a BPMS for hospital administration, and an ACMS of a pediatric ward starts with a physician preparing for a meeting. He intends to recommend some specific treatment and documents this intention sa into the patient information system. The information system does not need to be aware of the full history and only needs to forward the intention to the coordinator to check for compliance. It is valid behavior, and therefore the interaction is stored. The coordinator now expects that sa was performed and triggers appropriate actions. One rule informs the ACMS about sa, which in turn fires micro processes. The ACMS is aware of vetoes, and one inappropriate automated reaction can be canceled before damage is done. Another rule informs the BPMS that it finally can close some task for the recommendation step. The last rule fired generates some recommendation of common related steps which is transferred to the information system. The information system displays the recommendation.

WS-HumanTask and most APIs of BPMSs do not include a representation of speech acts. Stil, mapping annotated tasks to their appropriate speech acts can be performed at the coordinator. Therefore, systems that are not aware of coordination acts do not need to become aware. However, in order to relate cases to the various processes involved, some context information has to be provided by a BPMS. All participating systems that are aware of speech acts should adhere to some common speech act library, either by design or with a mapping.

VI. Discussion

This approach actually integrates BPMN and BPMN-like languages with ACM, even though [40] deemed them incompatible. The arguments for the conclusion of incompatibility are not challenged. In fact, we share the same positions. But the arguments are based on the assumption that an ACMS *depending* on processes designed in BPMN-like languages will fail. In our approach, no process schema is necessary and the ACMS does not depend on one. It facilitates interaction *with* structured processes that share a context and artifacts. Knowledge workers are not required and not even encouraged to model complete process diagrams. But often structured processes overlap with or trigger knowledge work, and this interaction should be supported by the ACMS. Additionally, knowledge workers are offered techniques to (semi-)automate routine process fragments [38], and process specialists may annotate BPMN with speech acts. Therefore, BPMN and ACM might still be incompatible, but the assumptions are not applicable in our approach.

A speech-act-based ACM approach might additionally support structured processes that require many interactions. For example, the *IT Infrastructure Library* (ITIL) process framework contains 26 processes and explicitly defines 216 interfaces between them, even excluding processes requiring interfaces to *all* other processes [41], [42], [43], [44], [45]. Moreover, 66% of the 150 process triggers described in the core publications can be attributed directly or implicitly to speech acts, and in practice a large share will be triggered by e-mail or phone. Implementers of ITIL have to consider a broad range of interfaces between processes involving different tailored information systems, BPMSs, ACMSs, and groupware. Our approach would standardize and classify the interactions between processes, simplify triggering and management of related processes for a case, and for inferences on interactions.

Our approach can also facilitate Compliance. Introducing speech acts into business rules increases expressibility of a BRMS and allows to formulate constraints on the pragmatic intention of interactions, e. g. commitments, directives, and even apologies, to find unfulfilled promises, pending interactions, and to check patterns of interactions required by law and internal regulations. The focus on speech acts allows a sensible mapping of similar events to a unified and semantically enriched format. Rules are applicable both for cases and cross-cutting concerns. Rules should not be enforced, but rather deployed as *line markings*, that emit warnings or suggestions and probably notify additional stakeholders. Vetoes acting as guardrails could be used to only automate happy paths and check for compliance problems prior to performing coordination and production acts. With a veto, deviant instances could be automatically canceled and dealt with by the knowledge worker. This would also benefit task automation services. Moreover, this approach can facilitate integration of autonomous agents that collect and infer information of a case. For example, Kalia et al. [14] identify tasks and commitments in email and chat conversations. These artifacts can be monitored by a BRMS, and warnings could be resolved by the user, either by correcting inaccurate commitments or by acting on justified warnings.

Speech acts are documented regardless of whether the speaker and hearer are knowledge workers or automated agents or systems. However, the documented interactions need to be relevant for the case. For example, the REST message initiating a process or querying information will not be documented, but sending an email with master data typically should be.

For our prototype, we also emphasize groupware functionality. Both because cases should contain their stakeholders, interactions, tasks, and dates, but additionally because users will not document their interactions unless they "have good ways to mark the illocutionary forces and to inscribe the propositional contents of their utterances" [46]. Usability is therefore a major concern and groupware functions facilitate selecting and entering interactions and the people involved. Usability for today's knowledge work also necessitates supporting mobile devices and thereby imposes additional requirements.

Still, the proposed approach does not yet emphasize on data security and user access control. Currently, roles primarily describe the various stakeholders involved in a case regardless of whether they have access to case data. For user access control, the approach of Cognoscenti [16] for federated cases could simplify this for the knowledge workers involved. Nonetheless, the two types of roles for stakeholders and users would need to be integrated in a sensible way.

In summary, we believe that taking integration and semi-automation into account, a speech-act-based approach on ACM systems may finally gain traction to be deployed and used in practice. In contrast to isolated applications, centralized inference and automatic reactions for processes and document generation could make the advantages of documenting and using the pragmatic intention more apparent.

VII. Conclusion

Knowledge workers are faced with scattered information across many business process support systems, ACMSs, and groupware. This is partially caused by the fact that the structured, semi-structured and ad-hoc processes involved have overlapping responsibilities and typically are not supported by one centralized system. Our approach could facilitate their integration. We classify communicative acts to share and infer on interactions with the standardized representation of a speech act adhering to a domain-specific speech act library. No process schema is necessary. However, a case may involve many structured or semi-structured processes across several BPMSs. The ACMS can provide a common context containing interactions, activities, artifacts, and stakeholders involved.

With a common representation of interactions and a shared context, we further support providing knowledge workers with line markings to guide them through their work. Compliance monitoring can consider interactions and their pragmatic intention, and rules may be formulated on different levels of granularity. For line markings, we provide warnings. Systems and knowledge workers may ask compliance monitoring in advance whether an action or interaction is compliant, and receive a veto if it is not. The rules are not enforced. If a system is not aware of vetoes, or if a system or knowledge worker decides it should ignore the veto, he or it is free to do so. The approach facilitates the integration of structured, semi-structured, and ad-hoc processes that now share common coordination and production artifacts. For automation of process fragments and using the veto mechanism, modeling for the happy path only may suffice and this could reduce the effort for users.

Still, there are many open questions. The prototype that combines ACM, speech-act-based business rules, semi-structured process fragments, and interacts with BPMSs, groupware and domain-specific information systems is under active development. Solutions for subproblems already offer to initiate documentation of interactions, relating corresponding case data, document generation based on case information, compliance checking, and more. But ensuring that external systems are not required to offer additional speech-act-based interfaces and still being able to apply speech act annotations might prove to be difficult.

Our architecture is useful only if knowledge workers actually decide to accept documenting their interactions, and this requires user interfaces with a high usability and low effort to distinguish illocutionary forces [46] as well as additional benefits. Incorporating a broad range of tools and processes requires to display a small amount of appropriate options. But knowledge workers should also be able to apply options that were not suggested and need to be able to find them. Knowledge workers are not restricted to desktop computers, they also use mobile devices. Business rules for compliance monitoring and automated actions require that the domain experts are encouraged and able to easily formulate and share rules. Data security and user access control also has to be provided in a usable way. Roles for user access and as classification of stakeholders of a case need to be combined in a way that facilitates management of the stakeholders involved in a case while reducing the effort to share case data properly. Ensuring usability under conflicting requirements will therefore raise many questions and compromises.

In summary, speech-act-based adaptive case management can facilitate integration of structured, semi-structured, and adhoc processes that are involved in a case as well as the many participating support systems. It enables useful inferences and line markings guiding through knowledge work.

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] M. Coronado and C. Iglesias, "Task automation services: Automation for the masses," *Internet Computing, IEEE*, vol. PP, no. 99, pp. 1–1, 2015.

[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] 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.

[12] H. R. Motahari-Nezhad, C. Bartolini, S. Graupner, and S. Spence, "Adaptive case management in the social enterprise," in *Service-Oriented Computing*. Springer, 2012, pp. 550–557.

[13] H. R. Motahari-Nezhad, S. Spence, C. Bartolini, S. Graupner, C. Bess, M. Hickey, P. Joshi, R. Mirizzi, K. Ozonat, and M. Rahmouni, "Casebook: A cloud-based system of engagement for case management," *Internet Computing, IEEE*, vol. 17, no. 5, pp. 30–38, 2013.

[14] 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.

[15] K. Swenson, "Demo: Cognoscenti open source software for experimentation on adaptive case management approaches," in *Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), 2014 IEEE 18th International*, Sept 2014, pp. 402–405.

[16] K. D. Swenson, "My sandbox, your sandbox," in *Thriving on Adaptability*, L. Fischer, Ed. Future Strategies Inc., 2015, pp. 19–30.

[17] J. Buck and S. Villines, *We the people: consenting to a deeper democracy: a guide to sociocratic principles and methods*, 2007.

[18] M. Hauder, R. Kazman, and F. Matthes, "Empowering end-users to collaboratively structure processes for knowledge work," in *Business Information Systems: 18th International Conference, BIS 2015, Poznań, Poland, June 24-26, 2015, Proceedings*, W. Abramowicz, Ed. Cham: Springer International Publishing, 2015, pp. 207–219.

[19] T. Winograd, "A language/action perspective on the design of cooperative work," *Hum.-Comput. Interact.*, vol. 3, no. 1, pp. 3–30, Mar. 1987.

[20] 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.

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

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

[23] M. Dumay, J. Dietz, J. Mulder, G. Goldhuhl, M. Lind, and S. Haraldson, "Evaluation of demo and the language/action perspective after 10 years of experience," *Proceedings of LAP*, 2005.

[24] I. Bider and E. Perjons, "Reviving language/action perspective in the era of social software: Research in progress," in *Short Paper Proceedings of the 5th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling, Rostock, Germany, November 7-8, 2012*, 2012.

[25] I. Bider, "Analyzing communication capabilities of cm/acm systems," in *Thriving on Adaptability*, L. Fischer, Ed. Future Strategies Inc., 2015, pp. 91–103.

[26] W. M. Van Der Aalst, P. Barthelmess, C. A. Ellis, and J. Wainer, "Proclets: A framework for lightweight interacting workflow processes," *International Journal of Cooperative Information Systems*, vol. 10, no. 04, pp. 443–481, 2001.

[27] T. Finin, R. Fritzson, D. McKay, and R. McEntire, "Kqml as an agent communication language," in *Proceedings of the Third International Conference on Information and Knowledge Management*, ser. CIKM '94. NY, USA: ACM, 1994, pp. 456–463.

[28] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, D. McKay, S. Shapiro, J. McGuire, R. Pelavin, and C. Beck, "Specification of the kqml agent-communication language," 1994.

[29] S. O. Kimbrough and S. A. Moore, "On automated message processing in electronic commerce and work support systems: Speech act theory and expressive felicity," *ACM Trans. Inf. Syst.*, vol. 15, no. 4, pp. 321–367, Oct. 1997.

[30] O. Hisarciklilar and J.-F. Boujut, "A speech act theory-based information model to support design communication through annotations," *Comput. Ind.*, vol. 60, no. 7, pp. 510–519, Sep. 2009.

[31] 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.

[32] 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.

[33] 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.

[34] 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

[35] 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.

[36] 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.

[37] J. Tenschert, G. Michelson, and R. Lenz, "Towards speech-act-based compliance," in *2016 IEEE 18th Conference on Business Informatics*, 2016.

[38] 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.

[39] OASIS, *Web Services Human Task (WS-HumanTask) Specification Version 1.1*, OASIS Std. [Online]. Available: http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cs-01.pdf

[40] K. D. Swenson, "Position: Bpmn is incompatible with acm," in *Business Process Management Workshops: BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, M. Rosa and P. Soffer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 55–58.

[41] Great Britain Cabinet Office, *ITIL Service Strategy 2011*, ser. Best Management Practices. London: TSO (The Stationery Office), 2011.

[42] ——, *ITIL Service Design 2011*, ser. Best Management Practices. London: TSO (The Stationery Office), 2011.

[43] ——, *ITIL Service Transition 2011*, ser. Best Management Practices. London: TSO (The Stationery Office), 2011.

[44] ——, *ITIL Service Operation 2011*, ser. Best Management Practices. London: TSO (The Stationery Office), 2011.

[45] ——, *ITIL Continual Service Improvement 2011*, ser. Best Management Practices. London: TSO (The Stationery Office), 2011.

[46] F. Flores and J. J. Ludlow, "Doing and speaking in the office," *Decision support systems: Issues and challenges*, vol. 11, pp. 95–118, 1980.