

# Agora - Speech-Act-Based Adaptive Case Management

Johannes Tenschert and Richard Lenz

Institute of Computer Science 6, University of Erlangen-Nuremberg,  
{johannes.tenschert,richard.lenz}@fau.de

**Abstract.** Today's work is increasingly characterized by unpredictable collaborative processes called knowledge work. Some types of knowledge work are supported by case management tools which typically provide regulated access to case-related information. Knowledge workers are well aware of the pragmatic dimension of their communicative acts, but the systems they are using currently are not. This demo paper presents a prototype of a speech-act-based adaptive case management tool we call Agora. In Agora, we focus on interactions of a case and enable flexible documentation of ad-hoc interactions and activities as well as support for (semi-)structured processes. Interactions are linked to appropriate artifacts of the case. The approach enables useful inferences by the user and automatically by a system from all documented interactions and case-related data. For example, it can help in unveiling assertions and commitments, finding unfulfilled promises, and automatic reactions to certain interactions. Thereby, the approach facilitates integration of structured, semi-structured and ad-hoc processes.

**Keywords:** adaptive case management, speech act theory, knowledge-intensive business process

## 1 Background and Significance to BPM

Business process management is aimed at supporting cooperative work and shows its advantages especially in optimizing structured processes. However, in recent years the share of knowledge work has increased rapidly. Knowledge work involves the creation, distribution, or application of knowledge [2]. In contrast to a manual worker's clearly defined activities, knowledge work is characterized by abstractly defined tasks and the knowledge worker's responsibility for his own contribution in terms of quantity and quality [3]. In the US, around 50% of today's work is knowledge work [6]. Currently, interactions in knowledge work are supported by providing groupware and collaboration tools. Knowledge workers may also use an adaptive case management system (ACMS), tailored information systems, and other (process) support systems. Therefore, case data is scattered across many systems, and the overlapping structured, semi-structured and ad-hoc processes involved in actual knowledge work further impede keeping track of related data, activities and interactions.

Our speech-act-based approach focuses on interactions of a case, and enables flexible documentation of ad-hoc interactions and activities, as well as support

for (semi-)structured processes. It relates interactions to artifacts and makes the pragmatic intention of interactions explicit to facilitate useful inferences by the user or automatically by the system. Examples for these inferences are unveiling all assertions and commitments, finding unfulfilled promises, checking whether important stakeholders have been informed, or automatically reacting to certain interactions and situations, e.g. informing legal guardians about certain interactions. Moreover, the model enables compliance monitoring on interactions [9]. No process schema is necessary for a case, but if a more detailed schema can be given, the possibilities for support and inference increase. This demonstration introduces Agora, the prototype implementing our speech-act-based approach.

We use Speech Act Theory to classify and represent interactions. Speech Act Theory was first introduced by Austin [1], and further elaborated by Searle [4, 5]. 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 its illocutionary force  $F$ , i.e. the intention of the utterance, and its propositional content  $P$ . Searle's  $F(P)$  framework allows that propositional content may also be a speech act, i.e. speech acts can be nested ("Alice informed me that bob promised...").

One example use case of Agora is writing a conference paper. Finding appropriate results for publication is not a structured process and varies from case to case. However, there are some similarities between cases: Usually, several authors are involved, case-specific agreements occur, and these agreements contain speech acts. By documenting and classifying these interactions, useful inferences and relations can be displayed. One could start by giving the project some working title and collecting artifacts (sketches, emails). The actual writing of the paper will be predominantly ad-hoc, but there are many interactions: questions, promises ("I'll write Section 2."), complaints ("I don't like the abstract!"), requests for meetings, and so forth. Some tasks could be automatized as a micro process [7], for example creating a git repository. This process may contain the creation of the repository, storing its location in the case's master data, and emails to stakeholders of the case to inform them about the repository. In most cases, the process of submitting a paper to a conference only differs slightly. A micro process might create some tasks to find deadlines and to document them, informing all authors, submitting the abstract, and submitting the paper. If it is accepted, more structured tasks are triggered: "Hooray" to all authors, submitting a camera ready version, registering at the conference, and booking the trip. At the conference, many contacts and interactions can be documented in Agora. After the trip, a structured process to settle travel expenses is triggered.

Other examples that require handling cases in a flexible way and involve many interactions are treating and informing patients, handling legal cases, and supervising theses. Agora integrates well-known and well-established BPM techniques for structured processes with speech-act-based support for ad-hoc cooperation.

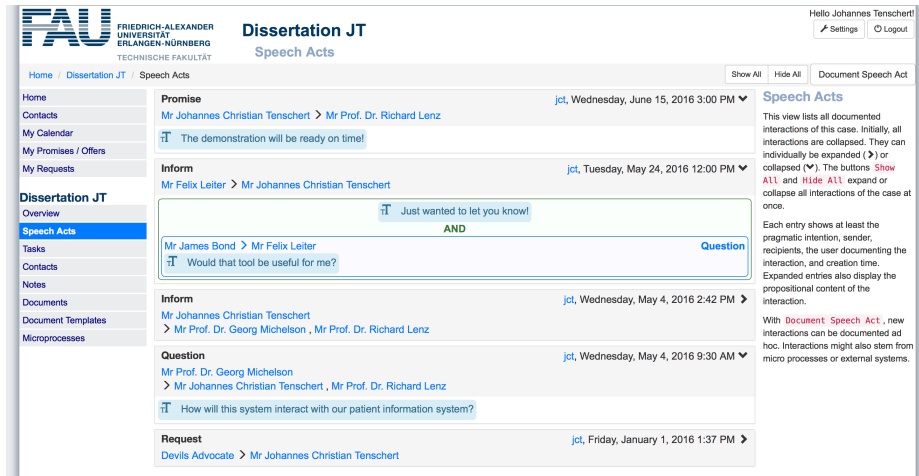


Fig. 1: Overview of speech acts of a case

## 2 Model and Implementation

This section introduces the representation of cases in our approach, outlines the key features of the prototype Agora, and briefly describes its implementation.

### 2.1 Cases

Agora manages a set of cases. Each case consists of interactions, contacts, tasks, documents, notes, related process instances, and key-value-based master data. Interactions are documented speech acts with a specific creator of the documentation. Speech acts contain a sender, one or more recipients, an illocutionary force, and propositional content in the form of text, speech acts or production acts for artifacts, and logically connected text and acts adhering to Searle's F(P) framework. Speech acts can be linked to artifacts of a case, e. g. documents and tasks. Figure 1 shows how interactions of a case can be displayed. Here, users can expand and collapse propositional content of all or individual speech acts for an overview of all interactions of a case as well as for detailed information. Related process instances currently are restricted to micro processes [7]. In the near future, they will also contain structured processes in an external BPMS, e. g. to settle travel expenses or grade and properly archive a thesis.

### 2.2 Features

Agora provides flexible key-value annotations for cases and artifacts. These annotations can be organized in tabs and arranged freely according to the knowledge worker's preference. In order to make certain interactions involved in the generation of documents or completing tasks visible to the user and an inference engine,

interactions are either automatically or manually linked to appropriate artifacts. Since Agora focuses on interactions, contacts involved in a case should be easy to maintain. Often, stakeholders are shared across many cases, e. g. attorneys and judges. Therefore, contacts are created once and can then be referenced. Contacts can be annotated with one or many roles in a case. The roles are not intended to be used like the traditional roles of BPM systems. They are intended to help managing the stakeholders involved, and *can* be used e. g. for simplifying document generation or preselecting contacts in a micro process.

Document templates may use certain artifacts (e. g. contacts) and master data (e. g. file reference, working title) of a case to generate word documents or emails. For example, while dealing with a thesis and if the student is a documented stakeholder of the case, the registration at the exam office can be generated. Also, travel expense settlement forms for attending a conference could be pre-populated. Micro processes [7] are tightly integrated into the prototype to automate routine fragments. They allow parallelism, missing attributes of interactions and actions that are clarified during execution, and flexible user input for several tokens of a process instance at the same time. Inferences currently are hard-coded into the system, but already allow for example to find unfulfilled promises and to display all assertions. The system is prepared to provide automatic reactions for interactions according to the type of interaction and the case's master data, e. g. to inform legal guardians about activities. The Agora client is a single-page web application. It synchronizes certain artifacts (e. g. contacts, cases, current view) continuously to facilitate collaboration.

### 2.3 Architecture and implementation

The architecture of our approach is described in [8]. On the server side, Agora is implemented in Java servlets. The servlets provide REST interfaces that return JSON. Clients access these REST interfaces with a single-page web application based on HTML5, AngularJS, and Bootstrap. Case data and processes are stored in a relational database and accessed with Hibernate.

Moreover, a JSON library has been developed to easily map objects to different profiles, i. e. to not reimplement the same classes for specific views (confidentiality, volume). An additional layer for templates of word documents has been created based on docx4j. This document generation layer handles merge fields, parameters, and dynamically generated formatted paragraphs and runs.

## 3 Maturity and Future Work

The Agora prototype is intended to demonstrate how interaction artifacts and the pragmatic intention of interactions can be integrated into an adaptive case management system. It has not yet been evaluated with end users. Agora supports interactions that are part of structured processes, emerge in semi-structured processes, or are documented in an ad-hoc fashion. The system provides a flexible data model that allows knowledge workers to create arbitrary annotations for

cases and artifacts, and to relate interactions to artifacts. A screencast is available at <http://www6.cs.fau.de/people/johannes-tenschert/bpm-2016>.

Future versions of Agora will improve inferences, usability, and traceability of interactions. Currently inferences are hard-coded. There is active development to integrate business rules according to [8, 9] for integration of structured, semi-structured and ad-hoc processes as well as for compliance monitoring. It is intended to allow user-definable domain-specific business rules. The data model of artifacts in Agora is flexible, and the model of artifacts typically managed with a smartphone and groupware (e. g. contacts, tasks, and notes) is very similar to the appropriate web standards vCard and iCalender in order to facilitate synchronization in the future. Therefore, knowledge workers can choose their preferred gadgets and tools to work on a case. Finally, the model and implementation will be improved to further support traceability of interactions. Currently, a case is one conversation, and the interactions forming logical connections, e. g. accepting a specific proposal, are not explicitly linked together. Ideally, finding the reason of performing certain (speech) acts should not require to read the whole conversation, and typical patterns of interaction could be supported without extensive manual documentation.

The vision for a final prototype is that even though case data may still be distributed across certain devices and BPMSs, the ACMS would be a system of record that includes and references all (interaction) artifacts of a case. It should suggest appropriate activities, integrate with the tools knowledge workers actually use, reduce the time to hunt for information and creating routine artifacts, and provide useful inferences for dealing with the case at hand.

## References

1. Austin, J.L.: How to do things with words. Oxford university press (1975)
2. Davenport, T.H.: Thinking for a Living: How to Get Better Performances and Results from Knowledge Workers, chap. What's a Knowledge Worker, Anyway?, pp. 1–24. Harvard Business Press (2005)
3. Drucker, P.F.: Knowledge-worker productivity: The biggest challenge. *California Management Review* 41(2), 79–94 (1999)
4. Searle, J.R.: Speech acts: An essay in the philosophy of language. Cambridge university press (1969)
5. Searle, J.R., Vanderveken, D.: Foundations of illocutionary logic. Cambridge University Press (1985)
6. Swenson, K.D.: Robots don't innovate - innovation vs automation in BPM (May 2015)
7. Tenschert, J., Lenz, R.: 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)
8. Tenschert, J., Lenz, R.: 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)
9. Tenschert, J., Michelson, G., Lenz, R.: Towards speech-act-based compliance. In: 2016 IEEE 18th Conference on Business Informatics (2016)