Semantic Email as a Communication Medium for the Social Semantic Desktop

Simon Scerri, Siegfried Handschuh, and Stefan Decker

Digitial Enterprise Research Institute,
National University of Ireland Galway,
IDA Business Park, Galway, Ireland
{Simon.Scerri,Siegfried.Handschuh,Stefan Decker}@deri.org

Abstract. In this paper, we introduce a formal email workflow model based on traditional email, which enables the user to define and execute ad-hoc workflows in an intuitive way. This model paves the way for semantic annotation of implicit, well-defined workflows, thus making them explicit and exposing the missing information in a machine processable way. Grounding this work within the Social Semantic Desktop [1] via appropriate ontologies means that this information can be exploited for the benefit of the user. This will have a direct impact on their personal information management - given email is not just a major channel of data exchange between desktops, but it also serves as a virtual working environment where people collaborate. Thus the presented workflow model will have a concrete manifestation in the creation, organization and exchange of semantic desktop data.

Keywords: Semantic Email, Workflow Patterns, Speech Act Theory, Semantic Web, Social Semantic Desktop.

1 Introduction

Despite sophisticated collaboration environments being around for a long time already, email is still the main means for distributed collaboration. People still use it for maintaining to-do lists, tracking, documentation, organization, etc. The major reasons for this may be grounded in the ease of use, the negligible learning effort to be able to use it, the universal availability, and that literally everyone on the Internet uses it and can be contacted via email. Email is the social communication backbone of the Internet and is a great example of the fact that often not the number of functionalities decide on the success of a technology, but its flexibility and how people use it (a similar example from the telecom world, would be text messaging which already produces larger revenues than other mobile services).

However, the many ways in which people use email are not well supported as the uses of email are beyond its original intended design [2]. Functionalities such as deadlines and reminders, tasks and task tracking, prioritizing etc. are missing or only supported to a limited degree. Emails in a conversation and their content are related by mere identifiers or mail subjects and simple extensions ("Re:", "Fwd:", etc.). What

S. Bechhofer et al.(Eds.): ESWC 2008, LNCS 5021, pp. 124 – 138, 2008.

would actually be needed would be support for simple action item management, i.e., task definition, time-lines, dependencies of tasks, checks if the action items have been completed, etc., in a seamless way. This could well be achieved by identifying and placing patterns of communication into a structured form.

From a technical perspective this can be seen as support for ad-hoc workflows [3], i.e., it helps users coordinate and track action items that involve multiple steps and users. Users can leverage ad-hoc workflows to better manage less-structured processes in a better way than via traditional rigid workflow systems. In this paper, we introduce a workflow model, based on traditional email, which enables the user to define and execute ad-hoc workflows in an intuitive way. By eliciting and semantically annotating implicit, well-defined workflows we collect missing information in a semantic, i.e., machine processable way. By grounding this work within the Semantic Desktop through ontologies, this information can be exploited for the benefit of the user via a tool that supports the user with personal information management.

In the Semantic Desktop paradigm [1], personal information such as address book data, calendar data, email data, folder structures and file metadata, etc. is lifted onto an RDF representation. Hence, information items on the desktop are treated as Semantic Web resources and ontologies and vocabularies allow the user to express such desktop data formally. The NEPOMUK Project developed an infrastructure of ontologies for this purpose[1], including NRL - a representational language [4] resembling RDF/S best practice. Other ontologies tackle different aspects of the semantic desktop. One such ontology in development is the Personal Information Model Ontology (PIMO) [5], which acts as a formal representation of the structures and concepts within a knowledge workers mental model. An instance of this model will include representations and inter-relationships between information elements on the user's desktop. The Information Element Ontologies provide a basis for these representations and are able to represent anything from files (NFO), contacts (NCO), calendar data (NCal) and so forth. The social aspect of the semantic desktop is dependent on data exchange and communication. Messages exchanged between desktops are rarely self-contained, and commonly refer to items in the user's PIMO like people, projects, events and tasks. The purpose of the NEPOMUK Message Ontology (NMO) is to introduce the message concept to the semantic desktop so that these links can be made explicit. These messages are exchanged between desktops within a workflow. Thus, the workflow model will have a concrete manifestation in the creation and organisation of semantic desktop data.

Our workflow model supports standard workflow patterns, i.e., templates for typical interactions, as defined by van der Aalst et al.[2] with a standard notation and well defined operational semantics. This semantics lays the foundation for automatic execution (this is beyond the scope of this paper). While our model incorporates a set of such templates, this template library can be extended by the user. Semantic annotations in conjunction with workflows not only support the user, but in the long-run also can be used in business environments for semantic business process mining, i.e., the identification of hidden, currently unsupported workflows and workflow consistency and compliance checking.

---

[1] http://www.semanticdesktop.org/ontologies/

[2] http://www.workflowpatterns.com/

The paper is organized as follows. In Section 2 we discuss related work. In Section 3 we present the main contribution of this work - a workflow model that supports and gives semantics to Email ad-hoc workflows. In Section 4 we provide a proof of concept for our work by applying our model to four common email workflows. We also introduce our prototype Semantic Email application. Finally in Section 5 we give an outline of our future work before concluding.

2 Related Work

The problems of Email and its impact on personal information management has been discussed a large number of times, most notably by Whittaker et.al.[2][6]. Giving the Email process a semantics via some formal standard would enable machines to understand it. As a result, machines could guide and support the user in the communicative process, automate the flow between trivial states of the communication, and in general participate actively in the communication alongside the user. This idea was extensively dealt-with in the research which first introduced the notion of Semantic Email [7]. Here a broad class of Semantic Email Processes was defined – to manage simple but tedious processes that are handled manually in non-semantic email. An email process was modelled as a set of updates to a dataset, via logical and decision-theoretic models. This research is suitable for a number of processes that are commonly executed in email collaboration. The processes need to be pre-meditated and have to follow fixed templates for each individual process (e.g. a template for meeting scheduling). However, email is frequently multi-purpose, and thus multiple processes should be able to execute within one email. A template-only based system would only support one process per email. Our rationale favours a system where the nature of the ad-hoc workflow can be elicited from the text of an email in a new thread. Workflows defined via a template selection would then complement processes detected on-the-fly. In [7] the possibility of establishing interactions between semantic email and other semantic web applications to support more sophisticated reasoning techniques was pointed out. Grounding Semantic Email within the Semantic Desktop enables the realization of this prospect.

In order to be able to elicit the nature of a process from the email content, an improved semantic email system needs to resort to some linguistic analysis that is, at least partially, automatic. A relevant standard theory on which this analysis can be based is Speech Act Theory [8]. The theory has been very influential in modeling electronic communicative patterns. The core idea behind is that every explicit utterance (or text) corresponds to one or more implicit actions. The theory was applied to the Email domain a number of times: in particular to ease task management for email-generated tasks [9][10] as well as for email classification [11][12][13]. In previous related work, efforts in this area where aligned with the concept of semantic email. This resulted in sMail - a conceptual framework for Semantic Email Communication based on Speech Act Theory [14]. This conceptual framework is also influenced by research that investigated the sequentiality of speech acts in conversational structure [15] as well as their fulfillment[16]. In this paper we will model the workflow of email collaboration based on this framework, using formal workflow patterns that serve as a conceptual basis for process technology.

The main contributions of this framework are the definitions and conceptualization of the *Email Speech Act Model* and the *Email Speech Act Process Model.* An Email Speech Act is defined as a triple (v,o,s) - where v denotes a *verb*, o an *object* and s a *subject*. The Email Speech Act Model contains instances for these speech act parameters. Verbs (*Request, Commit, Propose, Suggest, Deliver, Abort and Decline*) define the action of the speech act. The Object refers to the object of the action. Objects are classified in *Data* - representing something which can be represented within email (*Information*, *Resource, Feedback*); and *Activities* - representing external actions occurring outside of email (*Task*, *Event*). The subject is only applicable to speech acts having an activity as their object, and it represents who is involved in that activity – i.e., the *Initiator* (e.g. "Can I attend?"), the *Participant* (e.g. "You will write the document") or *Both* (e.g. "Let's meet tomorrow").

The Email Speech Act Process Model considers each speech act as a separate process. In essence, it outlines the expected reaction from both initiator and participant of a speech act, on sending it and on receiving it respectively. It assigns the *Initiator Expected Reaction* [IEA] and the *Participant Expected Reaction* [PER] to each speech act combination, and is applied to the Speech Act Model as (v,o,s) [IEA] → [PER]. The IEA refers to the status or action of the speaker, or in this case, the initiator on sending a speech act (*Expect*, *None*). The PER refers to the reaction expected from the hearer, or in this case the participant upon receiving and acknowledging a speech act (*Reply, Perform, None)*.

3 A Behavioural Model for the Email Process

The sMail Conceptual Framework [14] on which we base this work refers to an Email Speech Act Workflow (ESAW) which models sequences of speech acts within email messages in email threads. Thus this model is equivalent to the *Email Speech Act Process Flow Model.* The ESAW was however never formally defined. In this work we explicitly model it in a standardised workflow language. Its workflow patterns can

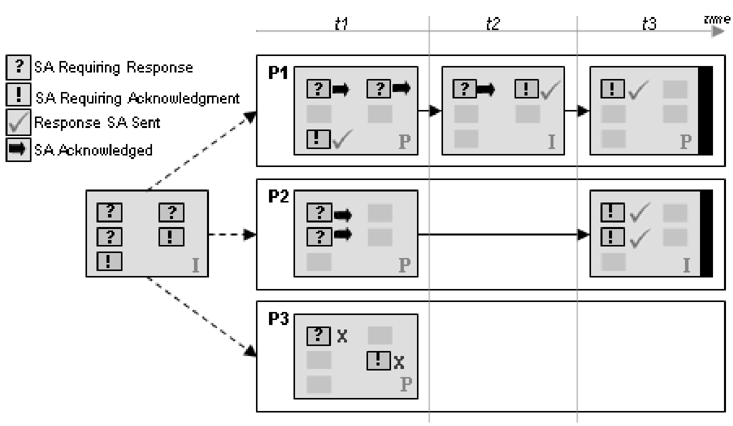


Fig. 1. Email Breakdown into seven ESAWs within three 1-1 transactions

be given semantics through their translation to YAWL[3] and subsequently Petri Nets[4]. Given this semantics, Semantic Email is not merely fancy email that is able to transport semantic content via Semantic Web technologies. The Email process itself is given semantics, by breaking it down into a number of speech act processes, each of which can execute according to a formal workflow.

In order to model our ESAW, every email is conceptually broken down into a number of 1-1 transactions between the *Initiator* and the *Participant* agents. The initiator refers to the agent that starts the email thread, whereas the participant is any other agent implicated in the email. Both agents can play the role of a sender or a recipient of an email in a thread in their own time. If the initiator addresses an email to n participants, then this amounts to n transactions. Each transaction can have zero or more speech acts which can be modeled using n distinguishable ESAWs. This concept is demonstrated in Fig. 1 – a timeline demonstrating how an email is broken down into separate, independent 1-1 transactions (cases where these transactions are not independent will be discussed later). Although the email received by the three participants (P1, P2, P3) is identical, the speech acts in the email are addressed to different participants. The first transaction between the Initiator and P1 consists of three speech acts (i.e. 3 out of the 5 speech acts are addressed to P1). Each speech act can execute along an independent ESAW. At time interval *t1*, P1 terminates one of these workflows, but sends control back to the initiator for the other two workflows. The activities in Fig. 1 are marked P or I depending on which agent has control. At time *t2*, the initiator terminates the second of these two, and returns control to P1 for the first. At time *t3* P1 terminates the last ESAW and as a direct result this 1-1 transaction terminates. The same happens with the transaction between the initiator and P2, although at a different time. However the third transaction does not terminate by time *t3* since P3 has stalled both ESAWs by not reacting to them.

The ESAW we present is grounded on key research in the area of control flow workflow patterns. The Workflow Patterns Initiative [3] was established with the aim of delineating the fundamental requirements that arise during business process modelling on a recurring basis and describe them in an imperative way. The contribution of this work is a set of patterns describing the control-flow perspective of workflow systems. Our ESAW uses 9 patterns out of the 40 documented patterns. Table 1 enlists these patterns (excluding the 10^{th} which is our customized version of the 1^{st}) together with a short description[5] and the graphical notation, used based on UML 2.0 Activity Diagram notations[6].

The workflow is graphically represented in Fig. 2. A swimlane splits the figure vertically to distinguish between the initiator and the participant(s) in separate 1-1 transactions. Whereas the initiator is always the agent that starts the workflow, its termination can depend on the initiator, the participant, or both. This is reflected in the workflow figure given the termination symbol is outside of both agent spaces. We will now give a brief walkthrough description of the workflow. The workflow will be further explained through a number of Semantic Desktop use-cases in Section 4.

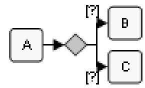
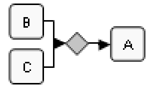
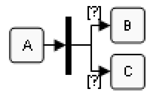
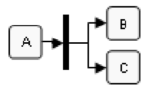
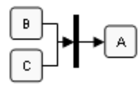
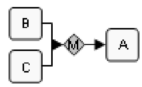
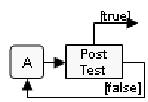
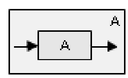
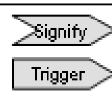
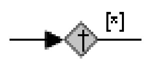
3 http://www.yawl-system.com/

4 http://www.informatik.uni-hamburg.de/TGI/PetriNets/

5 The given descriptions are neither formal nor complete. Refer to [3] for the complete specifications.

6 http://www.uml.org/

Table 1. Patterns used in the ESAW

Name	**Description**	**Graphical Notation**
Exclusive Choice (XOR-split)	The thread of control is immediately passed to exactly one outgoing branch based on the outcome of a logical expression associated with each.	
Simple Merge (XOR-join)	Convergence of two or more exclusive branches into a single branch. The active incoming branch passes control over to the outgoing branch.	
Multi-Choice (OR-split)	The thread of control is passed to one or more of the outgoing branches based on the outcome of logical expressions associated with each.	
Parallel Split (AND-split)	Branch diverges into two or more parallel branches each of which executes concurrently.	
Structured Synchronizing Merge	The thread of control is passed to the subsequent branch when each active incoming branch has been enabled.	
Multi-Merge	The thread of control is passed to the subsequent branch immediately when just one of the active incoming branches is enabled.	
Structured Loop: Post Test	Executes an activity or sub-process repeatedly. Post-test is evaluated at the end of the loop to determine whether it should continue.	
Recursion	The ability of an activity to invoke itself during its execution or an ancestor in terms of the overall decomposition structure with which it is associated.	
Persistent Trigger	An activity is triggered by a signal from the external environment. These triggers are persistent and are retained by the workflow until they can be acted on.	
Simplified XOR-Split	The majority of exclusive choices in the workflow have two common choices. These are abstracted with this symbol and expanded later in Fig. 2.	

An agent can initiate a speech act sequence by sending one of the following speech acts (shown below with their brief description and their intended effect).

1. (Suggest, Activity, *): Activity suggestion – No reply or action required
2. (Deliver, Data, Ø): Deliver unrequested data – No reply or action required
3. (Propose, Activity, *): Dependent (1-n) – n replies required
4. (Abort, Activity, *): Notification of an aborted activity – Action possible
5. (Commit, Activity, *): Notification of a commitment – Action possible
6. (Request, Activity, *): Independent (1-1) – Reply required, action possible
7. (Request, Data, Ø): Data request – A reply is required

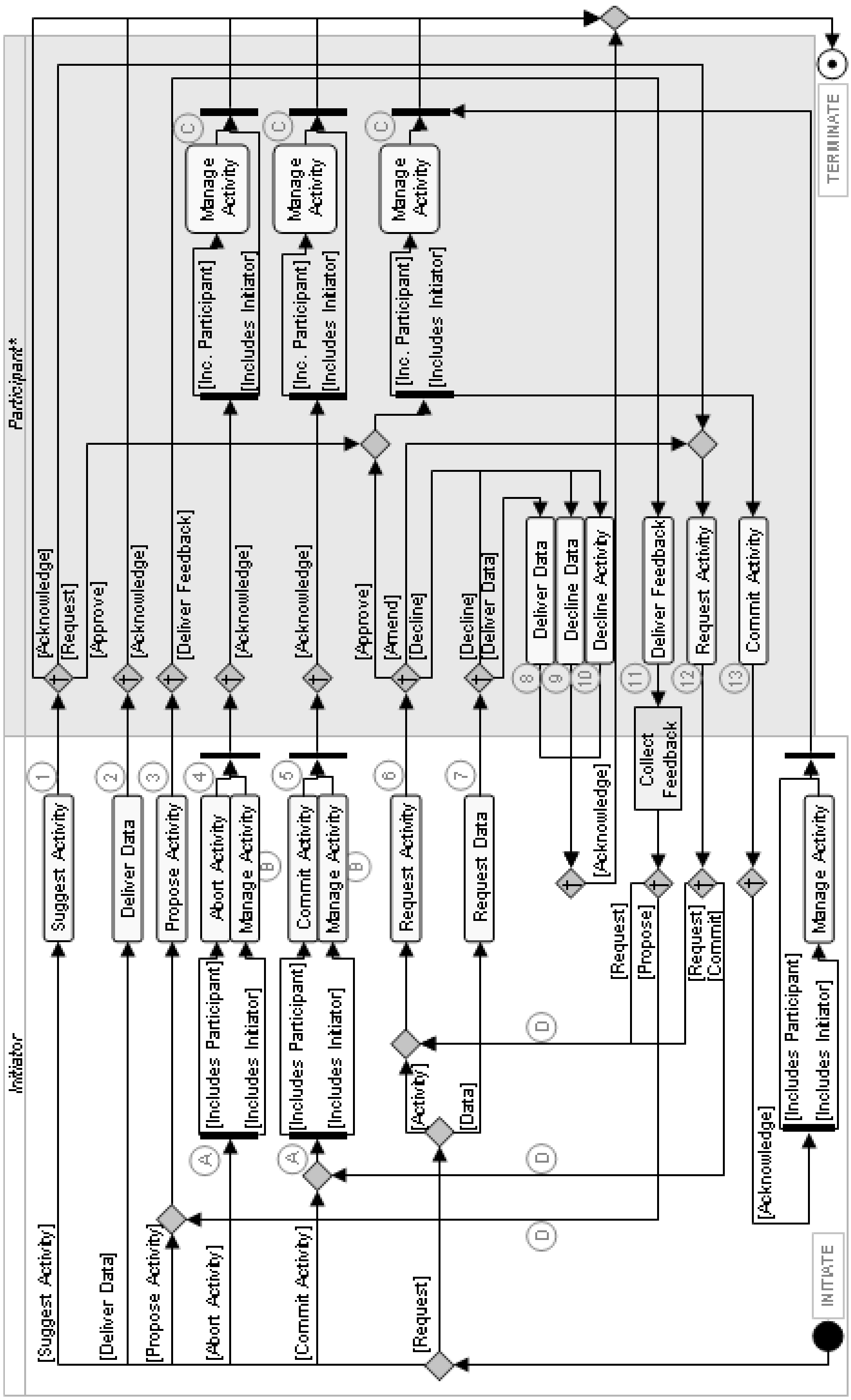


Fig. 2. Speech Act Process Flow Model

– where ‘*’ denotes any subject (i.e. the event or task implicates the initiator, participant or both). These seven speech acts are marked 1-7 in Fig. 2. The ESAW shows that the choice of some of these initiative speech acts (marked A), namely (Abort, Activity, *) and (Commit, Activity, *) results in a multi-choice split. If the subject of the activity includes the initiator (e.g. “I will do something!”) the respective path is executed. If it includes the participant (e.g. “You will do something!”) the alternative path executes. If both are included (e.g. “We will do something!”), both paths execute simultaneously. If the activity implicated an action by the participant, the process continues simply by sending the speech act. If the activity implicated an activity by the initiator, the initiator is expected to manage the generated activity (marked B), e.g. represent it on the semantic desktop as an event or task.

On receiving a speech act (grey half of Fig. 2), the participant is presented with a choice of activities according to the type of speech act received. Whereas some of these choices lead to the termination of the ESAW, e.g. ‘Acknowledge’, others execute a more interesting path. Some lead to activity management for the participant (marked C), whereas a number of paths make it necessary for the participant to send one of six reply speech act to the initiator. These six speech acts are marked 8-13 in Fig. 2. A brief description of these reply speech acts and of their context (in reply to [i.r.t]) is given below:

8. (Deliver, Data, Ø): Deliver requested data – i.r.t. request for data
9. (Decline, Data, Ø): Decline requested data – i.r.t. request for data
10. (Decline, Activity, *): Decline participation – i.r.t. an activity request
11. (Deliver, Feedback, Ø): Deliver feedback – i.r.t. activity proposal
12. (Request, Activity, *): Same as before – amending an activity request
13. (Commit, Activity, *): Same as before – i.r.t. an activity request

On sending these reply speech acts, control is returned to the initiator (white half of Fig. 2). At this point, some paths lead to the termination of the workflow (e.g. via Acknowledge) whereas other paths restart the loop by returning control to the participant via another speech act process (marked D).

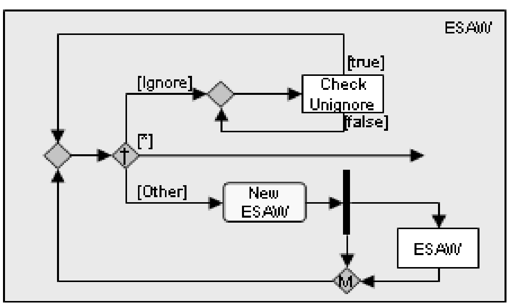


Fig. 3. *Other* and *Ignore* choices in simplified XOR-split

As we mentioned earlier, the workflow in Fig. 2 has been simplified. In reality, each exclusive choice that an agent must consider after receiving a speech act (simplified XOR-split in Table 1) has two extra choices – ‘Ignore’ and ‘Other’. Fig. 3 shows the behaviour of these choices.

The ‘Ignore’ path leads to a structured loop pattern that uses a post test which continuously checks for a reactivation signal. If this is detected it leads back to the start of the subactivity. The ‘Other’ path is more interesting since it uses the recursion pattern. Given this path is enabled, a new ESAW is initiated as a sub-workflow of the current ESAW. In the meantime, control is returned to the start of the subactivity via a structured synchronizing merge. This means that via the ‘Other’ path, *n* sub-workflows can be initiated. The parent ESAW does not need any of these sub-workflows to terminate in order to regain control. However, it does need all *n* sub- workflows which branched-off to terminate in order to terminate itself.

We also mentioned that there are cases when a speech act addressed to *n* participants does not branch in *n* simultaneous independent workflows. This is the case with speech acts with the *Propose* verb. In this case, the workflow will stall until the initiator gets the responses of all *n* participants. In Fig. 2 this behaviour is represented with a ‘Collect Feedback’ activity. This sub-activity is expanded in Fig. 4. Here we see that when each of the *n* participants delivers the required feedback to the initiator, a signal is fired. On the initiator’s side, each time a signal fires a post-check structured loop checks whether all participants have submitted their feedback. If this is the case the initiator can decide on which action to take, depending on the desired level of consensus in the cumulative feedback.

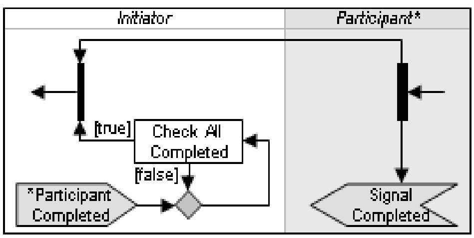


Fig. 4. *Propose* Speech Acts

4 Semantic Email on the Semantic Desktop

We have implemented a prototype of an application [17] that supports semantic email and the workflow model presented in this paper. The application, called *Semanta* (Semantic Email Assistant), is an add-in to a popular Mail User Agent[7]. The sMail models discussed in 2.3 have been encapsulated and are available within a Semantic Email Ontology[8]. Linking this ontology to those provided by the Semantic Desktop was a step towards email and desktop information unification. *Semanta* uses these ontologies to enhance email messages with semantics. Annotation of the email is performed both automatically via a speech act extraction web service, and manually via an appropriate wizard. The metadata is transported alongside the email content via a specific RDF MIME extension in the email headers. In this section we will provide

[7] Currently only Microsoft Outlook is supported but an add-in for Mozilla Thunderbird is in the pipeline.

[8] http://smile.deri.ie/ontologies/smail.rdf

a proof of concept for our ESAW and demonstrate how it can benefit the social aspect of the social semantic desktop. We will do this via four common use-cases that take place between the semantic desktops of four employees in a hypothetical company. The employees communicate via email, supported by *Semanta*. The employees have different levels of authority - Marilyn is a director, James a manager whereas Frank and Grace are regular employees.

4.1 Data Request

Marilyn requires a recent photograph of two employees - James and Frank. She selects an email Request Template where she asks the two contacts to deliver her the required data – a resource. The email has one speech act, a (Request, Resource, Ø), addressed to both participants. As shown in Fig. 5 at time interval *t1*, the speech act is broken down into two separate processes, which follow the ESAW independently:

i. P1 (Marilyn-Frank): When Frank (P) receives the email from Marilyn (I) at time *t1*, *Semanta* will recognize its purposes and will mark it as having pending action items until Frank follows a path in the ESAW which terminates the workflow. When Frank, clicks on the annotated speech act *Semanta* presents him the following choices: *Deliver*, *Decline*, *Ignore* and *Other*, as defined in the ESAW. He selects the Deliver option and, supported by the knowledge stored in his PIMO, he finds an appropriate photo in seconds. On selecting it, *Semanta* generates an automatic reply email. This is locally linked to the previous email in the thread, as well as to the instance of the photo in Frank's PIMO. Some auto-generated text in the email is automatically annotated with the speech act (Deliver, Resource, Ø). Frank does not need to add anything to the conversation so *Semanta* sends the reply. When Marilyn receives the reply at *t2*, *Semanta* shows her the following options: *Acknowledge*, *Ignore*, *Other*. She acknowledges the receipt and given the nature of the speech act, *Semanta* prompts her to save the file. When she does, an instance of the file is created in her PIMO and linked to the email message that introduced it to her desktop. This workflow instance terminates.
ii. P2 (Marilyn-James): At time interval *t1*, James does not have a more recent photo available. He selects the Decline option, and *Semanta* prompts him to provide the justification as input text to an automatic reply email. This is

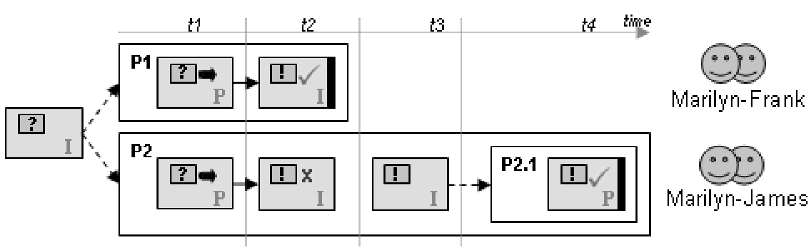


Fig. 5. Data Request and Task Delegation

automatically annotated as (Decline, Resource, Ø). When Marilyn receives the reply at *t2*, she is assisted with choosing from *Acknowledge*, *Ignore* or *Other*. She selects the latter, since she wants to tell James that he should get a new photo as soon as possible. *Semanta*'s wizard assists her with creating and annotating this reply (Figure 6). The results is a (Commit, Task, Participant) addressed to James. On sending it at *t3*, she sets off a new sub-workflow (P2.1). Before this new workflow terminates, the parent workflow, P2, cannot terminate.

As opposed to previous work [7], even after selecting a seemingly fixed data request template, the ensuing workflow is still to a great extent ad-hoc. For example, when James or Frank got the data request, they could have replied to Marilyn with a request for information asking why she needs their photo – and wait for the answer, before actually delivering the resource.

4.2 Task Delegation

At time interval *t4* (Sub-workflow P2.1 in Fig. 5) James receives Marilyn's latest email with the speech act committing him to a task. *Semanta* assists him with choosing from Acknowledge, Ignore and Other. He selects the first option, thus accepting the task. The subject of the speech act in Marilyn's email is the Participant, so according to the ESAW, given that James is the participant in the transaction, he has to manage the generated activity (activity management is not shown in the time graphs). *Semanta* prompts him with the activity manager (Figure 6). Unless he chooses to dismiss the generated task, the *Semanta* can support him with adding the task as an object on his semantic desktop (i.e task list). In his PIMO, the task would be linked to the email in which it was generated and to the contacts involved, i.e. Marilyn. When James is done managing the task, the workflow terminates.

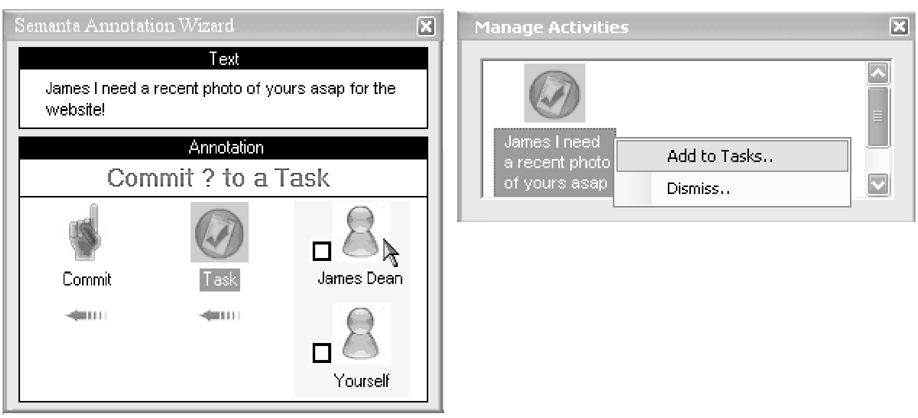


Fig. 6. *Semanta*'s Annotation Wizard and Activity Manager

4.3 Appointment Scheduling

James wants to organize a meeting with Grace and Frank. At time interval zero, he looks into his calendar to find an appropriate date and time. From his calendar application, James generates a meeting proposal for Tuesday at 11am, addressing it to Grace and Frank. *Semanta* generates an automatic email with one speech act. Since James is implicated in the event, the speech act equivalent to this proposal is (Propose, Event, Both). James sends the proposal, thus initiating two parallel instance of the ESAW as shown in Fig. 7. Until James gets the requested feedback, the email item will show in a list of items pending a response. If James feels that too much time has elapsed, *Semanta* can assist him with sending a reminder.

i. P1: When Grace receives the email, at time *t1*, *Semanta* recognizes the speech act within it as being a meeting proposal. She is shown her calendar and the commitments for the proposed day. *Semanta* then shows her the following options: *Deliver Feedback*, *Ignore*, *Other*. Since Grace is free at the proposed time, she expresses her availability and delivers the feedback.
ii. P2: Unlike Grace, Frank has another commitment. In his feedback at *t1*, he informs James that on Tuesday he is only available after 1pm.

Given the semantics of a Propose, the two workflow instances generated by James are not independent. In fact, the separate parallel instances of the ESAW merge back together once both participants have delivered their feedbacks as stated in Fig. 4 and seen in Fig. 7 at time interval *t2*. James is now shown two options: Propose – to propose again and Request – to call for the meeting. Since James requires everyone to attend the meeting and no consensus was reached, he proposes another time for the meeting – Tuesday at 2pm. When he sends the new proposal at time *t3*, the cycle is repeated, and the control flow splits again into two dependent activity instances:

i. P1: Grace is available at the new time and delivers her positive feedback at *t4*.
ii. P2: Frank is also available and also delivers the positive feedback at *t4*.

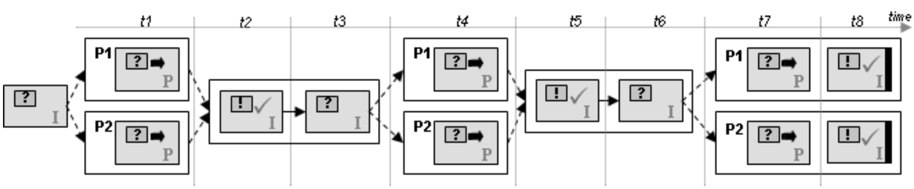


Fig. 7. Appointment Scheduling

At time *t5*, James gets both Grace's and Frank's feedback. Since consensus was reached this time around, he chooses the request option in *Semanta*'s user prompt. This translates into a (Request, Event, Both) addressed to both Frank and Grace. After sending at *t7*, the workflow splits again into two, this time independent, instances:

i. P1: At *t7*, Grace gets the request and she can choose from: *Approve*, *Amend*, *Decline*, *Ignore* or *Other*. She approves her participation and given the ESAW, the workflow follows two parallel paths. In the first *Semanta* assists Grace with managing the email-generated activity. In the second path, *Semanta* sends an automatic email to James with a (Commit, Event, Both) speech act. On receipt at *t8*, James acknowledges the commitment to the event. Since the subject of the speech act includes him, *Semanta* provides him with the possibility of adding the event to his desktop. If he does, a representation for the event is created in the semantic desktop's RDF repository, with a link to the source email amongst other metadata. This exposes the email-generated event to all semantic desktop applications. The workflow instance terminates.
ii. P2: At *t7*, Frank repeats the process performed by Grace, and is supported by *Semanta* with adding the event to his semantic desktop and sending a (Commit, Event, Both) back to James. On getting this at *t8*, James again acknowledges the commitment and is prompted to manage the generated event. This time he chooses to dismiss the event, since he already has a representation for this event on his desktop. The workflow instance terminates.

4.4 Event Announcement

James writes an email to inform Marilyn of the upcoming meeting so if she wants to, she can attend. *Semanta*'s Speech Act Extraction service recognizes a sentence as being either a suggestion or a request related to a new event involving Marilyn. James confirms that this is a suggestion, and *Semanta* creates a (Suggest, Event, Participant) addressed to Marilyn. On sending, an instance of the ESAW is created - P1 in Fig. 8, and control passes over to the Marilyn at time interval *t1*. *Semanta* offers to show her the known commitments for the said time and day. When considering the speech act, Marilyn is presented with the following options: *Acknowledge*, *Approve*, *Request*, *Other* and *Ignore*. Since the speech act is a suggestion, she does not have to take any action or even reply – in which case she would just acknowledge it. However she decides that she can attend the event, and chooses the Approve option. Given her selection, the workflow follows two parallel paths. Since the subject of the speech act she approved included herself, the first path sees *Semanta* assisting her with managing the generated event. The second sees *Semanta* generating a (Commit, Event, Participant) to be sent in an auto-reply to James. When James acknowledges this at *t2*, the path terminates because, being the initiator, he is not included in the subject of the speech act. Given both paths terminated, the workflow instance terminates at time interval *t2*.

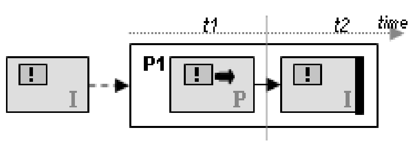


Fig. 8. Event Announcement

5 Conclusions and Outlook

In this paper, we have introduced the *Email Speech Act Process Flow Model* (ESAW) – a workflow model based on traditional email, which enables the user to define and execute ad-hoc workflows in an intuitive way. Given this model we can semantically annotate implicit, well-defined workflows, thus making them explicit and exposing the missing information in a machine processable way. Grounding this work within the Semantic Desktop through appropriate ontologies means that this information can be exploited by various semantic desktop applications for the benefit of the user. This will have a direct impact on their personal information management

Our main contribution is the formal description of the ad-hoc workflows of email communication. To the best of our knowledge, this was never formally described. We believe that defining this model using formal work flow patterns is equivalent to providing a formal semantics for email collaboration. After having defined the ESAW workflow model, we provided the corresponding implementation, namely the *Semanta* Email Client add-in. We believe that the human effort required to support *Semanta* with the semi-automatic annotation of email is minimal compared to the support provided by *Semanta* throughout the complete execution of the ensuing workflow. Alternatively our one-click semantic email process templates are still an improvement over previous work, since even after selecting a seemingly fixed template the workflow is still to a great extent ad-hoc.

In the near future we would like to perform two kinds of evaluation for our work. First we will perform a statistical evaluation by considering a corpus of real email messages. We will manually apply the ESAW model to conversations in this corpus, by breaking them down into a set of ESAW workflow patterns. This will determine the applicability of our workflow model to real conversations and point out any particular scenarios that the current model does not support. This evaluation will also serve as a study into the sequentiality of activities in email conversations, via probabilistic graphical models like Bayesian networks represented as transition diagrams. From a more formal point of view, we will analyse properties of the ESAW model mathematically, in an attempt to prove them as well as to determine the complexity of various patterns. Similar work has been done in follow-up work to [7].

Secondly we will investigate ways to evaluate the benefits of using *Semanta* to exchange semantic email, i.e, trial-based evaluation. We are currently improving *Semanta* and integrating further its functionalities within the Semantic Desktop. The ESAW we presented is completely extensible and new patterns can easily be introduced in the workflow. From an application point of view, *Semanta* can easily be extended to support custom workflow patterns required by a user or an organization. From another perspective, given its flexible nature support for the ESAW can be implemented on top of any existing protocol and email client. We will investigate the possibilities for integration of ESAW within widely-used email services like Gmail[9].

We believe that the here presented workflow model will have a concrete manifestation in the creation and organisation of semantic desktop data. Semantic annotations in conjunction with workflows not only support the user, but in the long-run also can be used in business environments for semantic business process mining,

---

9 http://mail.google.com/mail/

i.e., the identification of hidden, currently unsupported workflows and workflow consistency and compliance checking.

Acknowledgments. We would like to give our special thanks to Prof. Manfred Hauswirth and Armin Haller for sharing their knowledge on workflow patterns. The work presented in this paper was supported (in part) by the Lion project supported by Science Foundation Ireland under Grant No. SFI/02/CE1/I131 and (in part) by the European project NEPOMUK No FP6-027705.

References

1. Groza, T., Handschuh, S., Moeller, K., Grimnes, G., Sauermann, S., Minack, E., Mesnage, C., Jazayeri, M., Reif, G., Gudjonsdottir, R.: The NEPOMUK Project - On the way to the Social Semantic Desktop. In: 3rd International Conference on Semantic Technologies (ISE-MANTICS 2007), Graz, Austria (2007)
2. Whittaker, S., Bellotti, V., Gwizdka, J., Email and PIM: Problems and Possibilities. In: CACM07 (2007).
3. Voorhoeve, M., Van der Aalst, W.: Ad-hoc workflow: problems and solutions. In: 8th International Workshop on Database and Expert Systems Applications (DEXA07) (1997)
4. Sintek, M., van Elst, L., Scerri, S., Handschuh, S.: Distributed Knowledge Representation on the Social Semantic Desktop: Named Graphs, Views and Roles in NRL. In 4th European Semantic Web Conference, Innsbruck, Austria (2007)
5. Sauermann, L., van Elst, L., Dengel, A.: PIMO - a Framework for Representing Personal Information Models. In: I-Semantics'07, pp270–277. (2007)
6. Whittaker, S., Sidner, C.: Email overload: exploring personal information management of email. In: CHI '96 Conference, Vancouver, BC Canada, April, pp. 276–283. (1996)
7. McDowell, L., Etzioni, O., Halevey, A., Levy, H.: Semantic Email. In: 14th international conference on World Wide Web, New York, NY, USA (2004)
8. Searle, J.: Speech Acts. Cambridge University Press (1969).
9. Khoussainov, R., Kushmerick, N.: Email task management: An iterative relational learning approach. In: Conference on Email and Anti-Spam, CEAS'2005 (2005).
10. Corston-Oliver, S., Ringger, E., Gamon, M., Campbell, R.: Task-focused summarization of email. In: Text Summarization Branches Out workshop, ACL 2004 (2004).
11. Khosravi, H., Wilks, Y.: Routing email automatically by purpose not topic. Natural Language Engineering, Vol. 5, pp. 237–250. (1999)
12. . Goldstein, J., Sabin, R.E.: Using Speech Acts to Categorize Email and Identify Email Genres. In: System Sciences, HICSS2006 (2006)
13. Carvalho, V, Cohen, W.: On the collective classification of email speech acts. In: SIGIR-2005, pp. 345–352. (2005)
14. Scerri, S., Mencke, M., Davis, B., Handschuh, S.: Evaluating the Ontology underlying sMail Ű the Conceptual Framework for Semantic Email Communication. In: 6th Language Resources and Evaluation Conference (LREC 2008), Marrakech, Morocco (2008)
15. Jose, P.: Sequentiality of speech acts in conversational structure. In: Journal of Psycholinguistic Research. Vol.17, Number 1, pp. 65Ů88. (1988)
16. Singh, M.: A Semantics for Speech Acts. In: Annals of Mathematics and Artificial Intelligence. Vol. 8, Numbers 1-2, pp. 47Ů71. (2006)
17. Scerri, S.: Semanta - Your Personal Email Semantic Assistant. In: Demo Session at the International Conference on Intelligent User Interfaces (IUI08). Island of Madeira, Canary Islands, Spain. (2008).