The Path towards Semantic Email: Summary and Outlook

Simon Scerri[1], Brian Davis[1], Siegfried Handschuh[1]

[1] 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 provide a summary of work that has been pursued in the area of Semantic Email, with a particular focus on our work in the area. The aim of this paper is to provide a status quo for this topic, as well as to generate ideas and discussions that could evolve the topic and take it to new heights. We finish off by outlining future directions for evaluation, improvement as well as extension of our current technologies.

Introduction

The motivation for our work arises from a vision of a framework that supports the exchange of semantically-annotated email in order to expose implicit knowledge about the following to machines:

- artefacts of the email communication process (people, tasks, etc.)
- evolution of the communication (thread structure)
- ad-hoc workflows taking place within email (information requests, meeting scheduling, etc.)

Machines will then be able to support email users with correctly interpreting, handling and keeping track of email messages. Thus, Semantic Email can have direct benefits to known problems like Email Visualisation, Personal Information Management [1] and Workflow Management.

In this paper, we provide a summary of work to data and an outlook for the future of Semantic Email. After a summary of earlier work, we introduce the theoretical aspect of our research, which is encapsulated within the sMail Conceptual Framework. This framework has been applied in Semanta - a plug-in/extension to existing mail user agents to support Semantic Email. After presenting Semanta, we discuss future plans and provide some concluding remarks.Related Work

In this section we give a brief look at earlier initiatives on which we built our work. Speech Act Theory [2] states that states that every speech (or written text) has one or more associated explicit acts. The theory has been influential in areas involving the Language-Action perspective or the Pragmatic Web. It has been applied to email (to the textual content) a number of times, in particular to classify email based on the sender's intent[3][4][5], detect focus of threaded email discussions[6], predict actions on email messages[7] and ease task management arising through email[8][9] amongst others. Other related work in this area included an investigation on speech act fulfillment [10] and on the sequentiality of speech acts in conversation [11].

From another perspective, other relevant work involved the introduction and formalization of Semantic Email processes [12]. This involved an attempt to study and formalize a number of common scenarios, or workflows, that occur in email conversations.

Our work combines ideas from both perspectives. We are also interested in studying and modeling email ad-hoc workflows. Conventional email allows for specific subworkflows to take place routinely and spontaneously within the email communication. The whole email communication process can then be considered as an abstract ad-hoc workflow composed of a set of these well-defined subworkflows. We apply speech act theory to elicit knowledge about these ad-hoc workflows. However we allow for this knowledge to be created, transported, modified and retained in all stages of email communication, thus enabling the possibility of email workflow and personal information management.

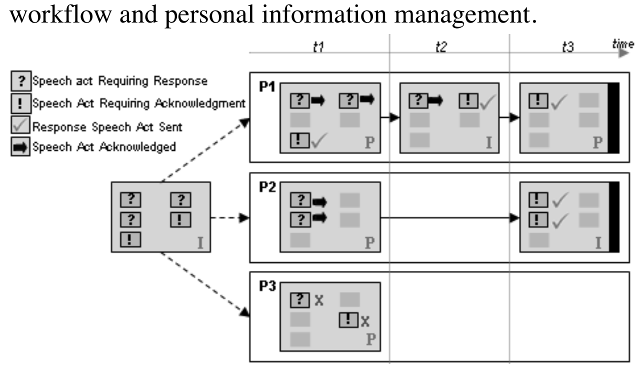

Figure 1 : Email breakdown into 1-1 processes

sMail Conceptual Framework

The sMail (Semantic Mail) Conceptual Framework[1] applies speech act theory to the email communication process, in order to give a more formal structure and semantics to ad-hoc workflows which are characteristic of

[1] http://smile.deri.ie/projects/smail/

email communication. Our approach is based on the notion that the content of any email message in a thread can be summarized into a number of predefined speech acts. Every initial email in a thread is conceptually broken down into a number of 1-1 transactions between the *Initiator* and the *Participant*. 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 (we exclude the possibility of mailing lists at this stage), then this amounts to *n* transactions. Each transaction can include zero or more speech acts. In turn, each of these speech acts initiates a workflow with clear intents and expectations. This concept is demonstrated in Fig. 1 – a timeline demonstrating how one email is broken down into separate, independent 1-1 transactions (dependent cases will be discussed later). Although the email received by the three participants (*P1*, *P2*, *P3*) is identical and contains five speech acts (gray boxes), the speech acts are addressed to different participants(denoted as an outlined gray box):

1. A request for a resource, addressed to P1, P2, P3
2. A request for information, addressed to P2
3. A notification, addressed to P1
4. A request for information, addressed to P1
5. A notification, addressed to P3

The three transactions execute concurrently. The first transaction (between initiator and P1) starts with three speech acts. These initiate three sub-workflows and control is passed to P1. At time interval *t1*, P1 receives the email and addresses the three speech acts. Some speech acts require only acknowledgment (gray box with an exclamation mark) whereas others require a response (gray box with a question mark). Whereas the second (the notification) requires no further action (i.e. sub-workflow is terminated), P1 reacts to the other two by sending a response to the initiator. This response is in the form of another two speech act within a reply email. This means that for these two sub-workflows, control is passed back to the Initiator at t2 (activities in Fig. 1 are marked P or I depending on who has control). At t2, the second speech act requires no further action (sub-workflow terminates), whereas the initiator returns control to P1 for the first. At time t3 the last sub-workflow taking place between the initiator and P1 is terminated. This signals the termination of this 1-1 transaction (denoted by a black bar). In the meantime, the other two transactions have also been executing. The transaction between the initiator and P2, also terminates by time t3. However the third transaction does not terminate by then since P3 has not yet reacted to the speech acts (unattended speech acts are marked with an 'x').
The speech acts described in the scenario are instances of the *sMail Speech Act Model* - a speech act conceptualization specific to the electronic communication domain with a particular emphasis on email. An exchanged speech act amounts to a Speech Act Process. Such a process act implies an effect (to various degrees) on both the sender and recipient of the speech act. The *sMail Speech Act Process Model* outlines a number of different effects and applies them to pre-defined instances of the speech act model. Given the process model, every single speech act elicited from within a new email message can be seen as the start of a separate sub-workflow. Therefore, any email in any thread can contain any number of speech acts - all of which represent separate sub-workflows in different stages of lifetime. The *sMail Speech Act Process Flow Model* outlines the dynamic behaviour of these sub-workflows within email communication over time, given the expected effects outlined in the process model.
Collectively, these three models make up the sMail Conceptual Framework. We will now have a more detailed look at each of them individually, and present and outline performed and future plans for evaluation respectively.

Email Speech Act Model

This model is based on a succession of previous work in the area, most notably by Carvalho & Cohen [5]. We define the *sMail Speech Act* as the triple (*a*,*o*,*s*); where *a*

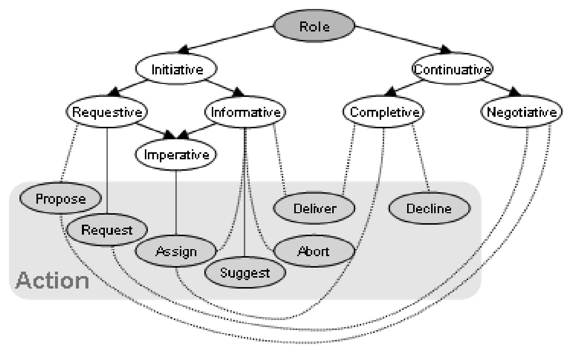

Figure 2 : Speech Act Actions

denotes an *Action*, *o* the *Object* of the action and *s* the *Subject* of the action. The Speech Act Model contains instances for these speech act parameters. Speech act actions (*Request, Assign, Suggest, Deliver, Abort, Propose* and *Decline*) can have varying *Roles*, e.g. *Initiative* for actions used to initiate discourse and *Continuative* otherwise (Fig. 2). Actions may serve particular roles in

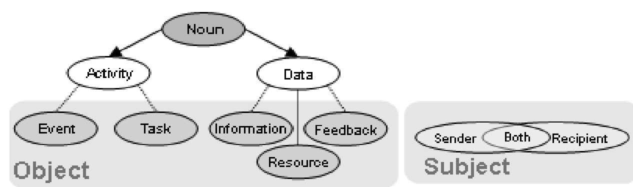

Figure 3 : Speech Act Object, Subject

different situations (e.g. Deliver can be *Responsive* as a response to a request or *Informative* otherwise). The speech act object represents instances of *Nouns* (Fig. 2), the nouns being *Data* - representing something which can be represented within email (*Information, Resource,*

Action	Noun	Subject	Description	Role	IEA	PER
Request	Activity	Recipient	Request/Amend recipients' activity performance	Requestive/Negotiative	Expect	Reply
		Sender	Request/Amend performance of personal activity	Requestive/Negotiative	Expect	Reply
		Both	Request/Amend performing joint activity	Requestive/Negotiative	Expect	Reply
Propose	Activity	Recipient	Propose recipients' activity performance (multiple & dependent)	Requestive/Negotiative	Expect	Reply
		Sender	Propose performance of personal activity (multiple & dependent)	Requestive/Negotiative	Expect	Reply
		Both	Propose performing joint activity (multiple & dependent)	Requestive/Negotiative	Expect	Reply
Assign	Activity	Recipient	Permit/Assign recipient's activity	Completive/Imperative	None	Perform
		Sender	Approve/Announce personal activity	Completive/Informative	Perform	None
		Both	Approve/Assign joint activity	Completive/Imperative	Perform	Perform
Suggest	Activity	Recipient	Suggest an activity for indirect consideration	Informative	None	None
		Sender	Suggest a personal activity for indirect consideration	Informative	None	None
		Both	Suggest a joint activity for indirect consideration	Informative	None	None
Decline	Activity	Recipient	Decline permission for recipients activity	Completive	None	None
		Sender	Decline performing personal activity	Completive	None	None
		Both	Decline performing a joint activity	Completive	None	None
Abort	Activity	Recipient	Abort recipient's preset activity	Informative	None	Perform
		Sender	Abort own preset activity	Informative	Perform	None
		Both	Abort preset joint activity	Informative	Perform	Perform
Request	Data		Request data from recipient	Requestive	Expect	Reply
Deliver	Data		Deliver data	Completive/Informative	None	None
Decline	Data		Decline delivering data	Completive	None	None

Table 1 : sMail Speech Acts and their associated IEA and PER

Feedback); and *Activities* - representing external actions occurring outside of email (*Task, Event*). The speech act subject (Fig. 3) is applicable to speech acts with activity as their object and represents who is involved in that activity – e.g. for a meeting, the *Sender* ("Can I attend?"), the *Recipient* ("You have to write the document") or *Both* ("Let's meet tomorrow").

Email Speech Act Process Model

The Email Speech Act Process Model considers an exchanged speech act as a separate process. Speech act the--ory highlights three forces of speech - the Locutionary (literal meaning); the Illocutionary (social function the speaker is performing); and the Perlocutionary (the result or effect on the hearer in the given context). The sMail Speech Act Process Model models the latter two forces for all combinations of speech acts given in our model.It assigns the Initiator Expected Action (IEA) and the Participant Expected Reaction (PER) to each speech act combination, and is applied over the Speech Act Model as:

$$(a,o,s) : [IEA] \rightarrow [PER]$$

where the IEA refers to the status or action of an agent on sending a speech act (*Expect*, *None*). The PER refers to the reaction expected from an agent upon receiving and acknowledging a speech act (*Reply*, *Perform*, *None*). *Expect* denotes that the communicator is expecting further communication. *Reply* denotes that the communicator is expected to further the communication, whereas *Perform* denotes that on sending or receiving a speech act the communicator is expected to perform an external action as a direct result of the speech act (e.g. attend an event/perform a task). Table 1 shows valid combinations (Action, Noun - generalization of the Object, and Subject where applicable) of speech acts in the sMail speech act model with the associated IEA and PER.

Email Speech Act Process Flow Model

This model (a.k.a Workflow Model) investigates possible sequences of speech acts and other actions within email discourse or threads, given the speech act process model. It can be considered as a well-defined representation of the ad-hoc workflows taking place within email communication, e.g. Meeting Scheduling, Task Delegation, Data Request, Event Announcement etc.

The workflow models sequences of speech-acts occurring in email conversations, as shown in Fig. 1. It attempts to generalise all possible scenarios ensuing the exchange of a speech act over email. In general there are three possible scenarios taking place after the recipient acknowledges an incoming speech act:

1. The incoming speech act needs to simply be acknowledged, no further action required (e.g. the recipient acknowledges some delivered information)
2. Upon acknowledgment, the recipient needs to respond to the speech act with another speech act (e.g. providing requested information, annotated as a Deliver-Information)
3. As a consequence of acknowledgment, the recipient is expected to do something (e.g. the recipient accepted a task, or acknowledged an event to be attended).

The workflow model allows for flexible reactions given the ad-hoc nature of these workflows (i.e. there is no way to deterministically predict how a recipient will react to a request in an incoming email). Thus situations where a new, unrelated speech act is sent back to the sender as a response may arise. For example, upon receiving a meeting request (which is an instance of the workflow model), the recipient might ask whether it is urgent and wait for an answer before making a decision on their attendance. This 'ask and wait' process can be considered as a secondary sub-workflow taking place within the original workflow.

The workflow model is extendable, and new processes can be easily modelled and integrated. If new speech acts are represented in the Speech Act Model and the Speech Act Process Model, the workflow model could easily be extended to reflect the changes. The workflow model presented in [13] is based on workflow patterns defined in [14]. These patterns can be given semantics through their translation to YAWL[2] and subsequently Petri Nets[3]. Via this model, the Email process can be given semantics by being broken down into a number of speech act processes that can execute along the workflow model.

Evaluation

The sMail Speech Act Model was evaluated [15] via statistical means. We employed the use of the κ-statistic, in order to calculate the inter-annotator agreement between human annotators annotating segments of a corpus of emails with one or more speech acts. κ varies between -1 – signifying complete disagreement; to 1 - perfect agreement; 0 signifying that there was no agreement above that expected by chance. The experiment was also performed on a similar, preceding model provided by Carvalho&Cohen [5] for comparison. The corpus for the experiment was comprised of a random selection of 50 email threads from the Enron corpus. After various considerations, we obtained different values for κ, varying between 0.756-0.830 for the earlier model (their own experiment yielded values of between 0.720-0.850) and between 0.811-0.836 for our model. This result confirmed that despite adding further parameters to our Speech Act Model in order to be able to give more structure and semantics to email discourse, our model still resulted in a high-level of agreement between human annotators – even if the difference was marginal. Thus, we deem that our model is an improvement over previous taxonomies that modeled speech acts in the email domain. From a practical perspective, some categories (speech acts) used for the annotation are irrelevant. If the sMail models are to be used in a user-supportive application, some speech acts will not have any visible effect in terms of machine support (e.g. an information delivery would not require the machine to do anything except bring the information to the recipient's attention). Re-calculating κ after the omission of such categories, we obtained values of 0.623 versus 0.511, in advantage of our model. Therefore, for our Semantic Email vision, the sMail Email Speech Act Model was found to be significantly better than any other existing model. The model was adjusted after considering observations of the main causes for annotator disagreement plus other feedback from the annotators.

The remaining two models will be evaluated collectively, via an experiment in which we will observe the sequentiality of speech acts as they occur in email threads. This evaluation will take into consideration research which investigated the sequentiality of speech acts in conversational structure [11]. As an outcome of this study we will devise probabilistic Bayesian Networks which highlight the probability of which speech act is most likely to succeed another speech act, if at all. Similar work was also carried out in [16]. Another means of evaluation for the sMail Conceptual Framework as a whole will take place via the evaluation of Semanta – an extension to existing email clients which takes advantage of the framework. Semanta and its evaluation plans are introduced and discussed in the next section.

[2] http://www.yawl-system.com/

[3] http://www.informatik.uni-hamburg.de/TGI/PetriNets/

Semanta

Semanta[4] (Semantic Assistant) is an add-in/extension to popular Mail User Agents[5]. Based on the sMail Conceptual Framework, it assists the user with handling common workflows, e.g. Meeting Scheduling, Task Delegation, Event Announcement, Information Exchange etc (co)executing in email threads. In this section we will explain how the knowledge in the framework is exposed to the application level via the sMail Ontology[6]. We will then have a look at the kind of support that Semanta provides to the user as well as at an important component – the text analytics service which provides semi-automatic annotation of email content. Although Semanta has many benefits as a stand-alone application, the full power of Semantic Email is taken advantage of when it is integrated within the Social Semantic Desktop (SSD) [17]. We will thus start this section with an introduction about the SSD.

The Social Semantic Desktop

On the SSD, personal information such as address book data, calendar data, email data, folder structures and file metadata, etc. is lifted onto an RDF representation. 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[7] developed an infrastructure of ontologies[8] which tackle different aspects of the semantic desktop. The Personal Information Model Ontology (PIMO) [18], acts as a formal representation of the structures and concepts within a knowledge workers mental model. An instance of this model includes 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), messages (NMO) and so forth. Providing Semanta as a Semantic Desktop application means that the semantic knowledge it captures and creates can be exploited by other intelligent desktop applications. Given Semanta already represents semantic knowledge in RDF, integration of Semanta on the SSD requires:

1. modifying the sMail Ontology to extend concepts in the semantic desktop ontologies (e.g. nmo:Email).
2. storing and accessing generated semantic knowledge on the desktop's RDF repository.

[4] http://smile.deri.ie/projects/semanta/

[5] Microsoft Outlook and Mozilla Thunderbird

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

[7] http://www.nepomuk.semanticdesktop.org/

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

Architecture

Fig. 4 illustrates the general architecture for Semanta on the SSD. Business logic is separate from the GUI and is available within the Semantic Email service. Hence only the GUI is dependent on the Mail User Agent (MUA) being targeted. Thus since Semanta uses RDF for knowledge representation, it can support different users using different MUA's on different platforms. The Semantic Email service is just one of the services provided on the SSD. Another service is the Text Analytics service, which is used by the Semantic Email service to provide semi-automatic annotation of email content. Both services have access to the knowledge of the sMail models via the sMail Ontology. In turn, this ontology is based on concepts from NEPOMUK Ontologies.

The MUA is still responsible for creating and sending email messages. Semanta is responsible for annotating and processing outgoing and incoming semantic email respecti-

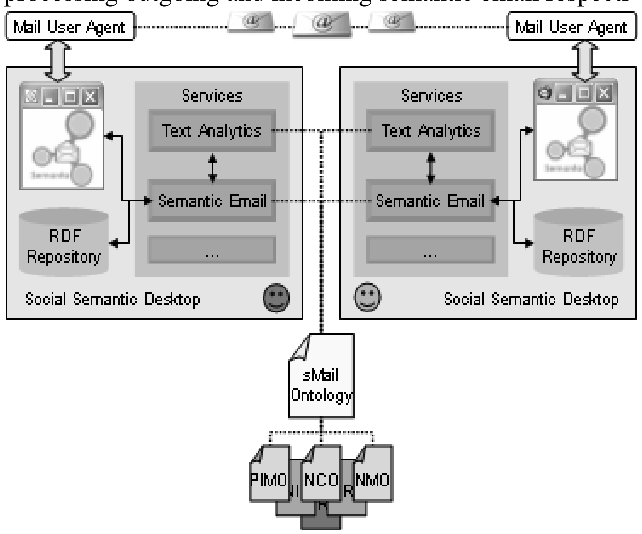

Figure 4 : Semanta on the SSD – Architecture

-vely. Semantic knowledge that is captured and processed by Semanta is stored in the RDF Repository on the desktop. This enables information integration between email-generated data and items on the user's desktop. These items, or representations of people, files, appointments, tasks, projects etc, are frequently the artifacts of email communication workflows. Thus data generated by Semanta can be used by other Semantic Desktop applications and vice versa. Additionaly, grounding Semanta on the SSD means that relationships between the given email artefacts (e.g. a task attended to by a group of people) and other data structures (e.g. A folder) and information elements (e.g. files in the same folder or a project related to the folder) can be deducted.

Intelligent User Interface

Given its knowledge of the sMail Framework via the sMail Ontology, Semanta can assist and support the user with a number of tasks. While relevant speech acts are semi-automatically elicited via the Text Analytics Service, the user can review, change or create new annotations via the Semanta Speech Act Annotation Wizard (Fig. 5). These annotations, together with metadata regarding email thread information, are represented in RDF and invisibly transported alongside the email content in the email headers. Semanta scans incoming email messages for speech acts (action items) and flags them appropriately.

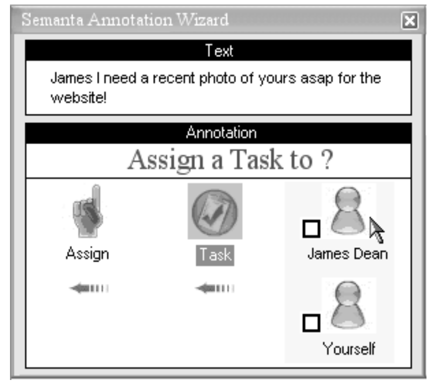

Figure 5 : Semanta's Annotation Wizard

When the user reads an incoming email, Semanta proposes a number of possible actions to the user (Fig. 6). Since Semanta can detect which email action items have been

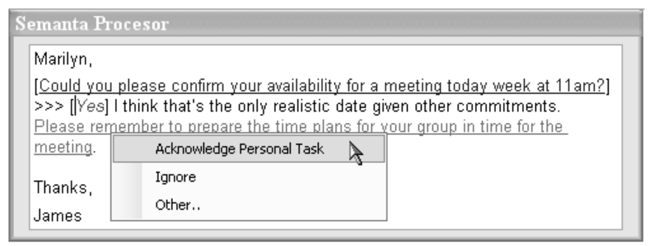

Figure 6 : Supporting the user with handling Action Items

given due attention or otherwise, the user can be

Given Semanta captures knowledge about artifacts of the e mail workflows, the user can be supported in their personal information management – particularly information generated via email. Semanta assists the user with exporting and saving events or tasks that were generated on-the-go as part of the email communication A semantic representation of these items is saved in the desktop's RDF store, thus enabling interoperation with other semantic applications and performing information integration by linking desktop to email data. This results in a more visual email environment, given that emails can be easily linked to threads, contacts, task, events, and any other information elements on the user's semantic desktop.

Text Analytics

The current text analytics service uses ontology-based information extraction techniques to elicit speech acts in email bodies. The information extraction is based on a declarative model which classifies text into speech acts based on a number of linguistic features like sentence form, tense, modality and the semantic roles of verbs[9]. The service deploys GATE [19] conditional corpus pipeline. The pipeline consists of a tokeniser, modified sentence splitter, POS tagger, keyphrase lookup via Finite State gazetteers and several JAPE [20] grammars. Some of these grammars are conditionally run based on the outcome of previous JAPE annotations and are ordered in priority to consume the longest matching annotation. Earlier work [4] implemented similar Knowledge Based (KB) approaches using earlier versions of GATE. The current service is at the beta testing stage whereby JAPE grammars are iteratively tuned based on the outcome of each test cycle. We are currently improving the recognition of persons involved in the email, taking advantage of existing structured sources such as Address books. We are also investigating the inclusion of co-reference resolution as well as conducting proper precision and recall measurements using an annotated extract of the ENRON corpus. We intend to employ a hybrid KB/Machine Learning option - a set of hand-coded JAPE rules that serve as input for a machine learning algorithm or as a backup strategy where a trained system fails.

Future Work

We are currently evaluating the applicability of the sMail Speech Act Process and Process Flow models on a corpus of email messages. We are interested in the realisation rate of the recipient's reactions to different speech acts, as outlined by these models. We would also like to apply formal methods to determine which properties of our workflow model can be proved. A similar analysis of particular email scenarios and their complexity was performed in [12]. After enriching Semanta with additional features and undergoing further testing we intend to perform a quantitative evaluation to establish whether Semanta provides any real benefits to the average email user – both on the semantic desktop and otherwise.

There are further ideas and directions which we intend to pursue. In particular, we would like to investigate the possibility of combining our technology with other established ideas in regards to Email, Personal Information Management and Workflow Management. One possibility involves Getting Things Done (GTD) [21] – a time/action management method which places particular attention on the context of tasks to be managed. This method is particularly relevant to our work given that it includes a workflow process which is used to gain control over all the tasks and commitments which one needs or wants to get done. GTD has already been applied to the email domain[10] and we want to investigate how our approach can benefit such initiatives. Although the sMail Conceptual Framework was originally targeted at the Email domain, it can easily be extended to apply to other forms of electronic communication, such as instant messaging. In fact, instant messaging is very similar to email communication – except for the fact that it is synchronous in nature and thus ambiguities can be cleared instantly. Nevertheless, questions are also ignored or not given due attention. More importantly, tasks are assigned and event are planned as much via instant messaging as via email. Thus, we would like to experiment with our models, the sMail Ontology and the Text Analytics service so that they can be extended and applied to this domain.

[9]Details of this aspect of text analytics are as yet unpublished and outside the scope of this paper.

[10] http://chandlerproject.org/

Conclusions

In this paper we provided an overview of the work that has been done on the topic of Semantic Email, as envisioned by us and others in previous work. Therefore this paper can be considered as a status quo for Semantic Email. We also provided a detailed insight about the architecture of a Semantic Email-supportive system for the SSD. Furthermore, we outlined future directions that we would like to pursue. These include further evaluation of the sMail models, extension and evaluation of Semanta, and improvements to the text analytics service. Additionally, we want to explore the possibilities of combining our technology with existing technology to further extend the applicability of Semantic Email. Finally we also outlined our intention to abstract the knowledge models and technology to take them to a higher level outside the domain of email.

References

1. Whittaker, S., Bellotti, V., Gwizdka, J.: Email and PIM: Problems and Possibilities. CACM.7 (2007).
2. Searle, J.: Speech Acts. Cambridge University Press (1969).
3. Goldstein, J., Sabin, R.E.: Using Speech Acts to Categorize Email and Identify Email Genres. System Sciences, HICSS2006 (2006).
4. Khosravi, H., Wilks, Y.: Routing email automatically by purpose not topic. Natural Language Engineering, Vol. 5 (1999) 237–250.
5. Carvalho, V., Cohen, W.: Improving Email Speech Act Analysis via N-gram Selection, ACTS Workshop, New York City, NY (2006).
6. Feng, D., Shaw, E., Kim, J., Hovy, E.: Learning to detect conversation focus of threaded discussions. Human Language Technology Conference, New York (2006).
7. Dabbish, L., Kraut, R., Fussell, S., Kiesler, S: Understanding email use: predicting action on a message. SIGCHI conference on Human factors in computing systems (2005).
8. Khoussainov, R., Kushmerick, N.: Email task management: An iterative relational learning approach. CEAS'2005 (2005).
9. Corston-Oliver, S., Ringger, E., Gamon, M., Campbell, R.: Task-focused summarization of email. Proceedings of the Text Summarization Branches Out workshop, ACL (2004).
10. Singh, M.: A Semantics for Speech Acts. In: Annals of Mathematics and Artificial Intelligence. Vol. 8, Numbers 1-2, pp. 47—71. (2006)

11. Jose, P.: Sequentiality of speech acts in conversational structure. In: Journal of Psycholinguistic Research. Vol.17, Number 1, pp. 65—88. (1988)
12. McDowell, L., Etzioni, O., Halevey, A., Levy, H.:Semantic email.World Wide Web, (2004)
13. Scerri, S., Handschuh, S., Decker, S.: Semantic Email as a Communication Medium for the Social Semantic Desktop. European Semantic Web Conference ESWC08 (to appear).
14. Van der Aalst, W., Voorhoeve, M.: Ad-hoc workflow: problems and solutions. 8th International Workshop on Database and Expert Systems Applications (1997)
15. Scerri, S., Mencke, M., Davis, B., Handschuh, S.: Evaluating the Ontology underlying sMail – the Conceptual Framework for Semantic Email Communication. Language Resources and Evaluation LREC08 (to appear).
16. Carvalho, V.R. and Cohen, W.W. On the collective classification of email speech acts. In Proceedings of SIGIR-2005, (2005) 345-352.
17. Decker, S., Martin, F: The Social Semantic Desktop. Technical Report (2004).
18. 5. Sauermann, L., van Elst, L., Dengel, A.: PIMO - a Framework for Representing Personal Information Models. In: I-Semantics' 07, pp. 270--277. (2007).
19. Allen, D.,: Getting Things Done (1996–2004).
20. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, (2002).
21. Cunningham, H., Maynard, D., Tablan, K.: JAPE: a Java Annotation Patterns Engine (Second Edition). Technical report CS--00--10, University of Sheffield, Department of Computer Science (2000).