INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

ANALYSIS OF VIRTUAL DESIGN COLLABORATION WITH TEAM COMMUNICATION NETWORKS

Matthias Uflacker[1], Philipp Skogstad[2], Alexander Zeier[1] and Larry Leifer[2]
(1) Hasso Plattner Institute, Germany (2) Stanford University, USA

ABSTRACT

We present a design information system for persistent team communication capture and the analysis of trends and relationships in multi-modal, digital team information spaces. Based on traces and streams of IT-mediated collaboration, the platform computationally constructs communication networks, a data structure representing the semantic concepts and relationships of shared information and individual actors in the process. We provide a set of ontologies to serve as a flexible template for the concepts and associations in design communication. Using the data from eleven distributed engineering design projects, we highlight the utility of our tool in the areas of design information retrieval, design management, and design research. For design research, we focus on identifying trends, key differentiators, and common signatures in the communication behavior of high-performance design teams.

Keywords: computer-supported cooperative work, team communication networks, design information capture, design research, design management

1. INTRODUCTION

Shaped by multi-disciplinary and distributed team setups, collaboration activities in engineering design face increasing challenges with regard to information sharing, dissemination and re-use [16]. Being both an enabler and a barrier for distributed collaboration, the transition to IT-supported team interaction and virtual communities creates heterogeneous, distributed, and disconnected clusters of project-relevant information [15]. The dispersal of knowledge representation and the multi-modality of team communication channels is further provoked by the individual selection of the most appropriate tools for each discipline and design situation [1,17]. But remote collaboration also has come to profit from the persistent nature of digital conversations. Today, email, wikis, forums, and online services for sharing pictures, videos or other documents are popular applications used to conveniently document and share design information across team members and stakeholders [14].

However, virtual collaboration groups are hindered by the augmented complexity and diffusion of digital team information spaces in achieving common ground and information context [2]. At the same time, the management and analysis of team processes is hampered by the difficulty to observe and assess multi-modal and distributed online communication activities, especially in larger and long-running projects. A consolidated and centralized representation of the current entities and associations in an information space could facilitate collaboration and empirical research design. Having insight into the history and status quo of communication metrics such as individual participation, internal or external influences and relationships between people and information creates potential to reveal hidden communication signatures and to identify or predict undesirable trends early in the process.

We have designed and implemented a design information system that supports the temporal evaluation of trends and patterns in digital information sharing activities of design teams. It accounts for the multi-modality of communication channels and provides a uniform model interface to capture and query the appearance and relationships between actors and heterogeneous information resources. Our approach emanates from the fact that more and more project information is distributed and available online, rendering shared information objects as uniquely locatable resources on the World Wide Web. With the digital footprint of design conversations growing, different traces of persistent communication become instantly available for the automated extraction of associations to actors and information. With the services provided by our software, we are able to computationally translate arbitrary types of relationships into *team communication networks*, which represent a set of semantic

statements about individual actors and resources of an information space. Through the integration of customizable ontologies for the communication channels being observed, the platform becomes flexible and adaptable to different design environments and application landscapes.
With the automated generation of a formal network-based model for typed and time-annotated information resources and relations, we establish an unobtrusive and non-intruding approach to analyze the communication and the temporal evolution of complex, multi-modal information sharing activities in collaboration processes. Three classes of use-cases are highlighted for the application of such an information system: 1) team support for context-rich information retrieval during the design process, 2) support for design management and education through real-time observation of virtual communication, 3) support for design research and empirical evaluation of influential factors.
In this work we present the foundations for team communication networks and their implementation in a Web-based service platform *d.store*, which has been applied to capture and evaluate the communication behavior of distributed engineering design teams over the course of eleven long-term projects. The ontologies used for the interpretation of information shared in emails and wiki pages are explained. We will further discuss the application of our system in this project setting, providing examples and results for each of the three use-cases specified above.

2. TEAM COMMUNICATION NETWORKS

We introduce team communication networks as a graph representation to express directed relationships (edges) between actors and information resources (nodes). Arbitrary properties (i.e. literal values) can be assigned to nodes and edges in form of attributes. All network entities (nodes, edges, and attributes) are typed, i.e. instances of one, possibly more semantical concepts. Through the individual and flexible classification of nodes (e.g. as *person*, *email*, *wiki*) and relations (e.g. *sent*, *reply_to*, *author*) a formal understanding about the contents and relationships in a shared information space is created. We continue with a formal definition for team communication networks.

Definition. A team communication network *TCN* is a directed graph $G_{TCN} := (T_V, T_E, T_A, V, E, A)$, where

- T_V is a collection of vertex types to classify the information or actors that are represented by network nodes.
- T_E represents a set of relationship types, which denote the semantic classes of individual network edges. Relationships are directed and can exist between any two nodes $v_i, v_j \in V$. Every edge is of exactly one relation type $r \in T_E$.
- T_A is a set of attribute types that can be instantiated to assign literal or numerical data values to the nodes and edges of a network.
- V is a set of typed network node instances. A node $v \in V$ is a 3-tuple $\langle cls, t_s, t_e \rangle$ with *cls* being a list of assigned node types $c \in T_V$, and two timestamps $t_s \in \square$ and $t_e \in \square \cup \infty$ with $t_e > t_s$ to mark the beginning and the expiration of a node's validity period in the network.
- E is a set of typed edge instances, representing the relationships between nodes. An edge $e \in E$ is defined as a 5-tuple $\langle s, r, o, t_s, t_e \rangle$ where $s \in V$ is the source node, $r \in T_E$ is the relation type, and $o \in V$ is the target node. Analogical, t_s and t_e are again two timestamps to define the time period of the relationships existence in a network.
- A is a set of typed attribute instances. An attribute is a 5-tuple $\langle s, p, val, t_s, t_e \rangle$. s is either a node $v \in V$ or an edge $e \in E$, the node or edge to which the attribute is assigned to. $p \in T_A$ is an attribute type and *val* is a literal or numerical data value of the attribute instance. t_s and t_e are defined as before and define an attributes validity period.

Figure 1 shows a simple example for a team communication network. Within the set of node instances $V := \{a,b,c,d\}$ nodes a and c represent two persons, b depicts an email that has been sent by person a, and d presents a wiki page that has been created by person c and is referenced by email b.
With the introduction of two timestamps t_s and t_e for the annotation of primary network entities, the point of time at which an individual instance (a node, relation, or attribute) became part of or has been removed from a network is stored in the model. The value of t_s indicates the point of time at which an entity was created in the network. t_e is set to *infinity* as long as the entity is present in the network. After removal, t_e is set to the date of its deletion, preserving all the information about an entity, but excluding it from subsequent representation. This way, the model maintains the traceability of a

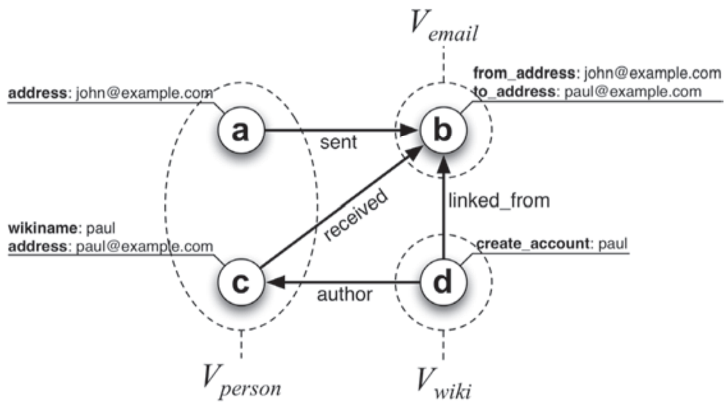

Figure 1. Team communication network with typed nodes, relations, and attributes.

communication network's evolution over the course of a project and enables the exploration of previous states of an information space.

2.1 Domain Ontologies and Inference

With a structured semantic representation of typed information objects and associations, we can contemplate the role of individual information and actors in a temporal, contextual manner. For the presented graph structure consisting of nodes, edges and attributes, the range and semantic of instance types is provided by a set of domain-specific ontologies. Each ontology describes concepts and relationships for particular information sharing applications. Based on these definitions, a formal conceptual model for the existence and constraints of the network building blocks is created, which allows inferring additional relationships out of asserted network properties. Inference is guided by a set of matching rules, which, e.g., define a *sender* relationship between a person and an email if the *address* value of the person matches the *from_address* attribute of an email (cf. Figure 1). By adding additional *address* attributes to a person that has multiple email addresses, sender and receiver relationships can be resolved instantaneously and in a flexible manner.

In the following, we give examples for domain ontologies and inference rules for Web-based collaborative editing of Wiki pages and email communication. Lacking standards for the graphical notation of RDF/OWL-based ontology models, we use a custom presentation language to describe the elements of domain ontologies. The notation is borrowing concepts from different graphical presentation languages for RDF and OWL, but has been tailored to address the requirements in the context of this work. The ontology namespace along with a number of prefix definitions for reused ontologies is listed on top of the graphical representation. Class definitions are represented as boxes and property types are represented as flat, directed shapes. Associated superordinates for classes and properties are stated above the particular definition, while additional type allocations are appended below. The domain and range types of a property are indicated by inbound and outbound connections to the associated resources. In the case of data-valued properties, the type of the value range is printed on the outbound side of the property. With these graphical primitives, an adequate overview of the ontology specifications that form the foundation of team communication networks can be provided.

2.2 Web and Wiki Resources

The Web of today has become a collaboration platform to support team interaction and information sharing activities. Information that is gathered in arbitrary resources is shared and distributed among process participants in order to disseminate insights and knowledge across the team. This can be done in many different ways, e.g., verbally or by sending emails with hyperlinks or attachments. Obviously, Web resources do not only reference other hyperlinked Web resources, but are also linked from other resources, which are not necessarily Web resources. By defining the two relationship types *hyperlink*

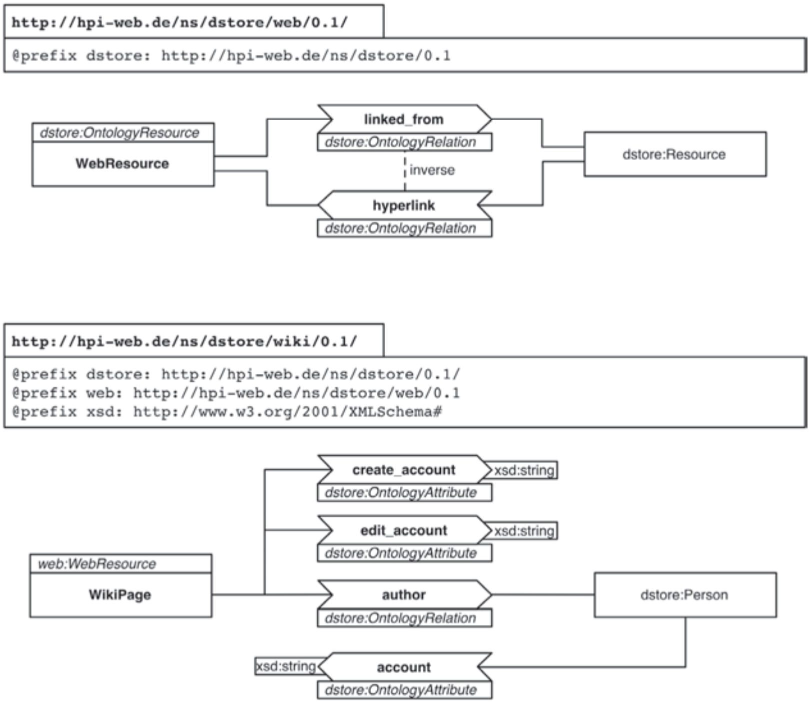

Figure 2. Two ontologies for Web and Wiki-related concepts in team communication networks.

and *linked_from* as inverse properties, a back reference from and to hyperlinked resources can be inferred.
Wikis are widely adopted in professional scenarios and project-based collaboration to support in information management and organizational tasks. The content is generally contributed, accessed, and incrementally revised by an interacting community of registered users, which makes this medium applicable to the analysis of collaboration practices [4]. Wiki applications that are keeping logs of page revisions and changes allow the examination of relationships between created content and contributers. Figure 2 shows the basic association types in conjunction with Wiki-based collaboration. As a specialization of a Web resource, a Wiki page naturally is also domain and range of hyperlink relations. The ontology defines two attribute types *create_account* and *edit_account* for the resource type *WikiPage*. These attributes are used to appoint the Wiki account name of the creator or of one of potentially many editors of a Wiki page. The ontology further defines an attribute type that is used to assign Wiki names to a person.

2.3 Email Communication

Email messaging is one of the most prominent technologies used for the digital exchange of information in an asynchronous and reliable manner. Studies reveal that global virtual teams largely rely on daily email communication [7]. Several advantages have let to a strong pervasiveness of email usage in virtual collaboration. Today, access to email systems is available almost anytime and anywhere, allowing asynchronous, ad-hoc transmission and retrieval of messages without needing to organize and participate in synchronous interactions.

Figure 3 shows an ontology that defines node, relationship, and attribute types used to describe email communication activities in team communication networks. The *Email* class is the central node type for transmitted email documents. Instances of this type have a number of lexical attributes such as

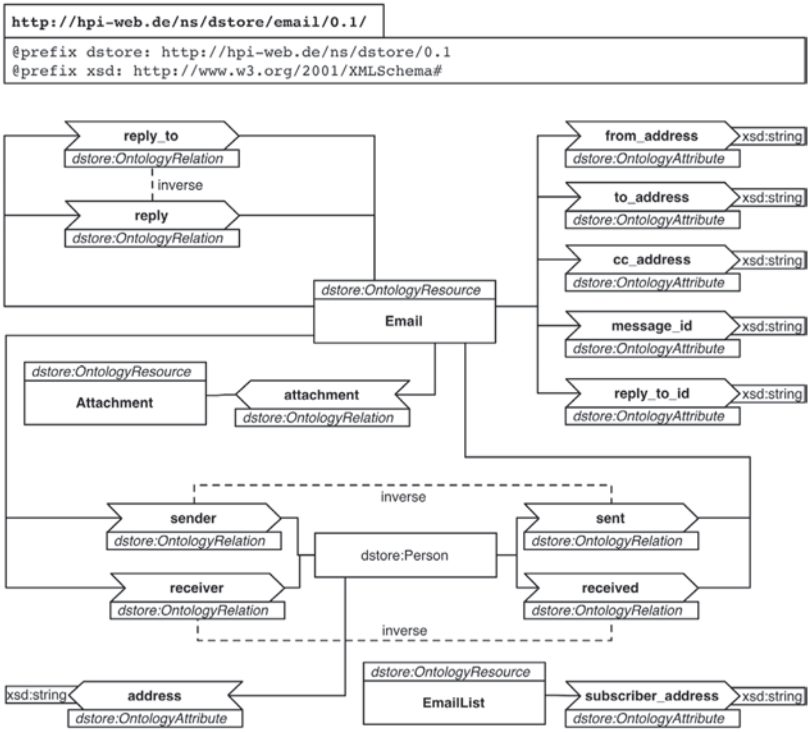

Figure 3. A domain ontology for email communication.

mailbox addresses listed in the *to* and *cc* fields of an email, a unique message ID, or the ID of a previous message to which an email replies to. This dependency between two emails is expressed through the *reply*/*reply_to* relationship types. Email addresses are assigned to persons via the *address* attribute. Bi-directional relations between an email and sending or receiving persons are expressed via the *sender*/*sent* and *receiver*/*received* relationship types. Additionally, the concept of an *EmailList* is used, which is characterized by having a number of email recipients signed up to the distribution list via the *subscriber_address* attribute.

2.4 Rule-based Inference of Node Relationships

With a set of node attributes and the help of inference rules, relations between information objects and persons can be derived computationally and instantaneously. This automatic inference of relationships decreases the number of relations that need to be explicitly stated in the graph, leading to increased flexibility and less overhead in the manipulation of networks, nodes and attribute values. A number of inference rules have been specified to dynamically derive types and relationships in a team communication network. These rules are defined as pairs of preconditions and implied postconditions, presented here in the form *antecedent* ® *consequent*. If the left-hand precondition of a rule is met for any combination of nodes in a network, then the right-hand postcondition is inferred to hold true as well.

Using the ontological concepts defined for email communication, an inference rule to establish *sender* relationships between emails and people can be specified in the following form:

from_address(x,y) *address*(z,y) ® *sender*(x,z)

It states that for any node *x* with an attribute *from_address* of value *y* and any node *z* with an *address* attribute of the same value *y*, a *sender* relationship between *x* and *z* is inferred. Note that, due to the inverse specification of the *sender* relationship, *z* instantly has an inferred *sent* relation to node *x*. Analogical rules can be constructed to spawn direct relations from an email message to its recipients:

$$\mathit{to\_address}(x,y) \wedge \mathit{address}(z,y) \rightarrow \mathit{recipient}(x,z)$$
$$\mathit{cc\_address}(x,y) \wedge \mathit{address}(z,y) \rightarrow \mathit{recipient}(x,z)$$

Accordingly, associations can be inferred that are relating person nodes to wiki resources, which have been edited or created by this person:

$$\mathit{create\_account}(x,y) \wedge \mathit{wikiname}(z,y) \rightarrow \mathit{author}(x,z)$$

Inference rules are also used to instantiate relationships between two information objects that feature logical interdependencies. One example is the *reply_to* relationship between an email and a preceding message to which the sender of the email replies. On message data level, this interdependency is encoded in the email header via unique message IDs and the value of a *reply-to* field that features the ID of a foregoing message. The following rule triggers the inference of *reply_to* relationships (and the inverse *reply* relations) with the occurrence of two email nodes with matching ID attributes:

$$\mathit{reply\_to\_id}(x,y) \wedge \mathit{message\_id}(z,y) \rightarrow \mathit{reply\_to}(x,z)$$

More complex dependencies between nodes can be expressed by adding additional prerequisites to the antecedent of a rule, such as:

$$\mathit{received}(w,x) \wedge \mathit{subscriber\_address}(w,y) \wedge \mathit{address}(z,y) \rightarrow \mathit{received}(z,x)$$

This rule stipulates that all emails, which have been sent to an *EmailList* also have been received by the persons who are subscribed to that list. Referring back to the network representation in Figure 1, it becomes apparent that only the attributes and the *hyperlink* relationship need to be modeled explicitly in the graph. All other relationships and node types can be inferred by a team communication network system with the help of the presented rules and the attribute values that are assigned to the four nodes.

3. PLATFORM IMPLEMENTATION

A system of team communication networks features a set of computer-processable ontology definitions and network instances, which are defined and encoded in a standardized format. The Resource Description Framework (RDF) [8,9] and the Web Ontology Language extension OWL [10] provide a suitable framework to encode the ontological elements required for this work. Both standards are used for the specification of ontologies and for the representation of concept and instance models in the platform implementation. The RDF/OWL framework supports a logical separation of concepts and instances through namespaces and the integration of multiple concept models through the import of other namespaces. This way, a decoupling of concept models and instance models across multiple team communication networks can be achieved, which allows independent customization and adaptation of models on individual network level without affecting the models of other networks.

To validate our approach, we have implemented *d.store*, a resource-oriented platform for team communication networks [11]. The platform provides a REST-based [12] service interface for accumulating information and actor relationships from observed collaboration archives. Every network instance and every individual node in a network is a uniquely identifiable Web resource provided by the service platform. Additional resources manage the collections of entities, such as the collection of nodes in a network, and the attributes or relationships for a particular node instance. The unified HTTP/1.1 interface of the resources allows independent and distributed retrieval and manipulation of network states, such as exploring node properties or creating a new resource, via the four standard calls *GET*, *PUT*, *POST*, and *DELETE*.

The organization of system- and network-specific RDF/OWL graph models in *d.store* is visualized in Figure 4. A global instance model organizes the collection of network instances. The basic concepts that are needed to describe team communication networks are defined in a system-wide concept model of the platform. For every communication channel that the system considers for the analysis of team collaboration, the system model imports a domain-specific ontology model that describes the relevant information and relationship types for that channel. We have given examples for email and Wiki communication, but any other medium can be described and imported. For each network instance in the system, two RDF/OWL models for network-specific concepts and instances specify the individual configuration of a network. The concept model of a network imports the global type definitions and hence the domain ontologies configured in the system. Additional ontologies can be imported into the conceptual model of a network without affecting the state and type configuration of other networks.

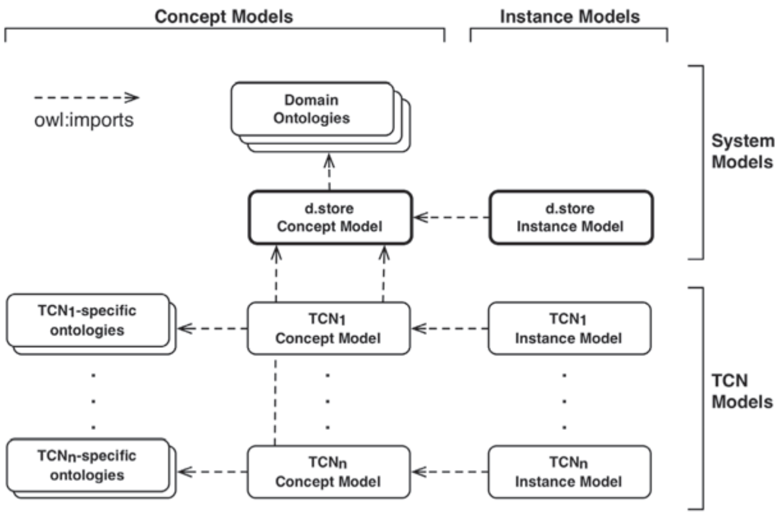

Figure 4. Organisation of RDF/OWL graphs in d.store.

Our system is built on top of the Jena Semantic Web framework[1], which is used to organize the concept and instance models. The application of the inference rules is based on the Pellet[2] reasoner library and a rule engine that interprets our inference rules encoded in the Semantic Web Rule Language format SWRL[3]. Team communication networks are gradually generated through posting identified objects and associations to the resource-oriented service interface provided by the platform. The internal RDF representations for the networks are decomposed into triple statements, which are processed in main memory for performance reasons. A persistent and consistent copy of the asserted statements is stored in relational tables. Adopting the Jena framework to our needs, the notion of an RDF triple has been extended by two additional timestamps, which mark the beginning and the end of a statement's validity period. These additional two values are used to model the timestamps t_s and t_e, which are defined for the elements of team communication networks. The resulting time-annotated RDF graph provides the basis for reasoning on and analysis of the constructed networks.

A number of helper applications to feed information and relationships identified in email archives or wiki logs have been implemented to automate the data import. Read and write access to the networks is provided via the unified interface of the HTTP/1.1 protocol [13]. d.store supports a number of representation formats for the provided resources of a network, including RDF/XML, the JavaScript object notation JSON, GraphML, and HTML. Clients can make use of the *Accept* header in HTTP/1.1 messages to negotiate the representation type that is returned from the server upon request.

[1] http://jena.sourceforge.net
[2] http://pellet.owldl.com
[3] http://www.w3.org/Submissions/SWRL/

4. APPLICATION

Starting to apply the platform for the temporal evaluation of multi-modal communication behavior of distributed design teams, we are using the data collected from eleven global engineering projects, which were running for a period of nine month. The projects were placed in a joint academic partnership between a US-based University and six universities in Europe and South America, with each project team distributed and composed of two groups of global and local students. Teams had a

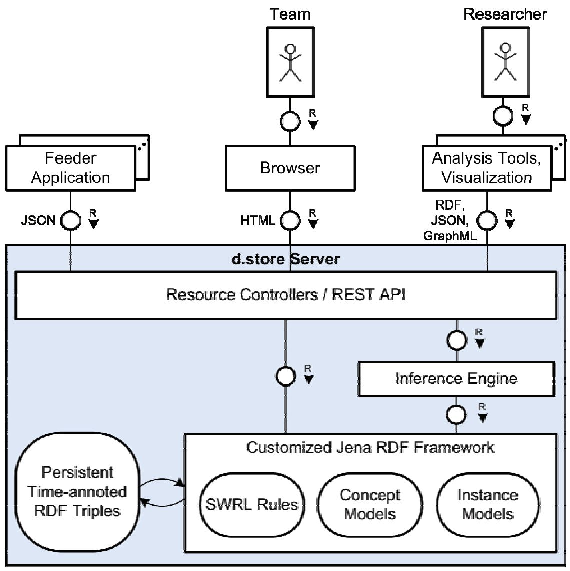

Figure 5. Platform and different clients accessing network resources provided by the service interface.

multi-disciplinary setup, involving students with backgrounds in mechanical engineering, software engineering, economics, product and industrial design. The project teams have been working independently on engineering design challenges that were provided by global enterprises and with guidance of corporate liaisons under close-to-realistic project conditions, budget and time constraints. The design process involved early need-finding activities, user observations, iterative prototyping and evaluation, and finished with a fully functional and documented prototype. All projects were synchronized with regard to start and end dates, milestone presentations and deadlines, which allows the comparison of information sharing activities across different teams.

The email archives, wiki and server log files that have been generated in the projects now constitute the basis for ongoing research into asynchronous information sharing activities via email, wikis, and document sharing systems. We started with the generation of team communication networks out of approx. 8700 project-related emails (containing more than 2900 hyperlinks and 1700 file attachments), 1200 wiki resources and shared documents. The average ratio between emails being sent and the number of emails with direct relationships to other information resources submitted in the form of attachments and hyperlinks was approx. 1 to 0.6. Obviously, emails are commonly used to share project information that is not only contained in the message itself, but is provided in files or in hyperlinked information resources, a fact that supports the creation of a multi-modal, networked view on team communication and information sharing processes.

We highlight three use-cases for the utilization of information provided by the *d.store* platform to exemplify the application for the three point-of-views *Design Information Classification & Retrieval*, *Design Management & Education*, and *Design Research*.

4.1 Support of Design Information Classification & Retrieval

From a team perspective, the retrieval of information in a shared information space presents an important and frequently repeated design activity. The *d.store* Web interface supports the search for electronically shared project information through the provision of contextual meta-information, assigned automatically to information resources in form of annotated types and relationships. The domain-specific types can be complemented ad-hoc by custom, user-defined annotations, resulting in a tagging mechanism known from other social collaboration tools. Tags can be used to filter and sort resources based on semantic concepts. We have applied information extraction tools for the automatic tagging and classification of information resources based on identified keywords and noun phrases.

The bi-directional nature of relationships in team communication networks allows for rapid navigation between related information resources on different communication channels. For example, the user can easily discover emails and other resources that have been mentioned or refer to a particular webpage by visiting the relationships *linked_from* for this resource (cf. sec. 2.2).

The Web user interface of the *d.store* platform provides access to the nodes of a team communication network, the assigned meta-data, types, attributes and hyperlinks for all related items. Resources can be filtered and sorted by time of appearance in the network, providing designers with a temporal understanding of the communication process.

4.2 Support of Design Management & Education

In both professional and academic design settings insight into the distribution of information sharing activities to individual participation and communication channels can be beneficial for the management and guidance of a team process. The *d.store* platform allows for the measurement of communication and the identification of communication peaks or distinctive features in the communication behavior of individual persons in real-time. Thresholds could be setup to automatically provide warnings to managers or educators in order to help correct anomalies or adjust the process accordingly. Therefore, the tool can be used as a non-intrusive tool that allows the identification of non-contributing team members early on so that managers or educators can intervene before potential damage is done to the design team's identity.

4.3 Support of Design Research

Design research is particularly concerned with an understanding of how ideas are created, communicated and evolve. A quantitative observation of the information space over the course of a project can be accomplished with *d.store* by considering nodes of a network that satisfy desired requirements with regard to types, relationships or attributes. In one case we were interested in the number of emails received by individual actors and the number of replies that were sent back to the team. Figure 6 shows data provided by our information system visualized in an interactive motion chart to track the evolution from the beginning of a project to its end. Each circle represents a person, color-coded by its individual role in the project (e.g. local or global team member, corporate contact). The two dimensions represent the number of emails a person has received (x-axis) and the number of responses sent (y-axis). The size of a circle denotes the number of hyperlinks a person has shared in the emails that were sent to the team list. The two highlighted node traces represent an obviously highly involved and responsive corporate contact person (left) and an actively communicating member of the local project team (right). The visualization shows that most other actors in the project are only marginally involved in the email communication. In a second case, the temporal distribution of information and actors being added to the information space was in the center of our team observations. Figure 7 displays a graph generated out of *d.store* data, which visualizes node additions in the communication network, separated by the respective node types. The figure not only shows the domination of email messages in the information space of the observed team, but also reflects noteworthy communication peaks towards the approach of project milestones.

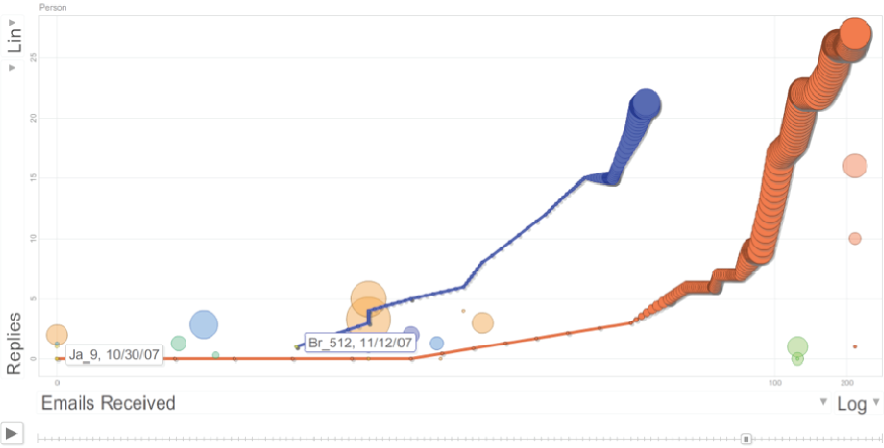

Figure 6. Interactive animation of email communication trends for individual project participants. The two dimensions contrast the number of emails a person has received (x) with the number of replies sent over time (y). The radius of a node indicates the overall number of emails sent by an individual.

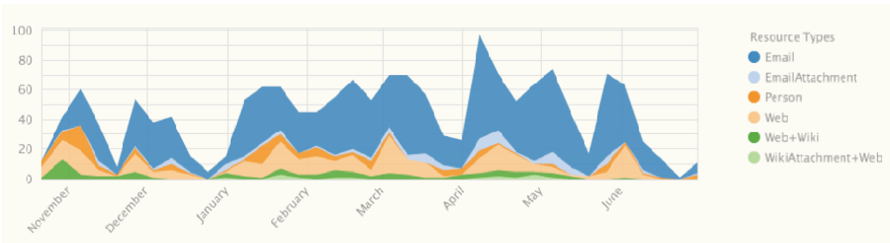

Figure 7. Generated graph for the temporal distribution of resource additions in a team communication network.

5. RELATED WORK

The assessment and analysis of communication signatures in collaborating groups and organizations has been the topic of other preceding works. Reiner [3] proposed a modeling framework to support collaboration and distributed knowledge management for design teams. His work demonstrates the use of design history as a source of insight into team design processes and identifies multiple correlations between historic design data and team performance in the domain of software engineering. The results are based on the application of a software tool to record design information through explicit interactions during (mostly local) collaboration processes.

A knowledge network management system has been presented in [18], which aims to provide mechanisms for retrieving, acquiring and classifying shared knowledge in collaboration groups. However, the taxonomy used to structure entities in a network remains relatively unspecific and static and computational creation of nodes and relationships is not considered.

Müller [4] presented a graph-based approach to analyze the evolution of wiki networks, processing server log files to create separate networks for the isolated analysis of either social relationships or link relationships. The focus of this research work, however, is set on self-organized knowledge management in wiki-based communities.

Gloor et al. have analyzed communication patterns in collaborative innovation networks (COINs) and presented a temporal communication flow visualization for social network analysis [5,6]. Their work is concentrated on mining email communication archives and the visualization of sender/receiver relationships over time to identify community structures and most active actors. With the team communication network platform presented in this work, we are targeting towards a unified representation of arbitrary, heterogeneous types of relationships, actors, and information in one single model. This increases the expressiveness of the created networks and allows the combined analysis of different communication channels and social relationships.

6. CONCLUSION

We have presented a non-interfering approach for the analysis of IT-mediated communication streams in distributed design teams. Team communication networks have been introduced, which provide a data structure that describes the occurrence and evolution of actors, resources, and their semantic and temporal relationships in distributed design information spaces. *d.store* has been implemented to capture and process conversations of design teams. Our approach does not require additional interaction or overhead to the workflow of process participants. The platform offers a resource-oriented service interface to team communication networks, providing central access to context and relationships of shared information objects and individual actors. While featuring general applicability through the flexible incorporation of domain ontologies, we have demonstrated the application of team communication networks in an academic context of eleven engineering design projects. Information that has been exchanged via email lists and wiki pages has been captured and translated to team-specific networks. Three potential use-cases for the utilization of the networks have been identified in the areas of design information retrieval, design management and education, and design research. Our research continues with a detailed exploration of the collected data and observed communication signatures. In particular, we are planning use the system to evaluate characteristic elements in the communication behavior of global design teams and will analyze the potential correlation between these signatures and team performance.

REFERENCES

[1] Shneiderman, B., Creativity support tools: accelerating discovery and innovation. *Commun. ACM*, vol. 50, no. 12, pp. 20-32, 2007.

[2] Bannon, L. and Bodker, S., Constructing common information spaces. In *ECSCW'97: Proceedings of the fifth European Conference on Computer-supported Cooperative Work*. pp. 81-96, Kluwer Academic Publishers, 1997.

[3] Reiner, K. A., A framework for knowledge capture and a study of development metrics in collaborative engineering design. Ph.D. dissertation, Stanford University, Stanford, CA, USA, 2006.

[4] Müller, C., Graphentheoretische Analyse der Evolution von Wiki-basierten Netzwerken für selbstorganisiertes Wissensmanagement. Gito, April 2008.

[5] Gloor, P., Laubacher, R., Dynes, S. and Zhao, Y., Visualization of communication patterns in collaborative innovation networks - analysis of some W3C working groups. In *Proc. of the Twelfth International Conference on Information and Knowledge Management*. pp. 56-60, ACM New York, NY, USA, 2003.

[6] Gloor, P. and Zhao, Y., TeCFlow - a temporal communication flow visualizer for social networks analysis. In *ACM CSCW'04 Workshop on Social Networks*, 2004

[7] Lurey, J. and Raisinghani, M., An empirical study of best practices in virtual teams. *Information & Management*, vol. 38, no. 8, pp. 523-544, 2001.

[8] Brickley, D. and Guha, R., RDF vocabulary description language 1.0: RDF schema. W3C Recommendation, W3C, February 2004.

[9] Beckett, D., RDF/XML syntax specification (revised). W3C Recommendation, W3C, February 2004.

[10] Dean, M. and Schreiber, G., OWL web ontology language reference. W3C Recommendation, W3C, February 2004.

[11] Uflacker, M. and Zeier, A., d.store: Capturing team information spaces with resource-based information networks. In *IADIS International Conference WWW/Internet 2008*, Freiburg, Germany, 2008.

[12] Fielding, R. T., Architectural styles and the design of network-based software architectures.

Ph.D. dissertation, University of California, Irvine, 2000.
[13] Fielding, R. T. et al., Hypertext Transfer Protocol - HTTP/1.1. RFC, The Internet Engineering Task Force, http://www.ietf.org/rfc/rfc2616, 1999.
[14] Perry, M. J., Fruchter, R. and Rosenberg, D., Co-ordinating distributed knowledge: a study into the use of an organizational memory. *International Journal of Cognition Technology and Work, CTW*, vol. 1, pp. 142-152, 1999.
[15] Perry, M. J., Fruchter, R. and Spinelli, G., Spaces, traces and networked design. HICSS, 2001.
[16] Geisler, C. and Rogers, E. H., Technological mediation for design collaboration. In *IPCC/SIGDOC '00: Proceedings of IEEE international professional communication conference*, pp. 395-405, 2000.
[17] Eckert, C. and Stacey, M., Dimensions of communication in design. In *Proceedings of the 13th International Conference on Engineering Design: Design Management - Process and Information Issues*, Professional Engineering Publishing, pp. 473-480, Glasgow, 2001.
[18] Uelpenich, S. and Bodendorf, F., Supporting the knowledge life cycle with a knowledge network management system. In *Proceedings of the 36th Hawaii International Conference on System Sciences*, 2003

Contact: Matthias Uflacker
Hasso Plattner Institute
Enterprise Platform and Integration Concepts
August-Bebel-Str. 88
14482 Potsdam
Germany
Tel: Int +49 331 97992 566
Fax: Int +49 331 97992 579
E-mail: matthias.uflacker@hpi.uni-potsdam.de

Matthias is a post-graduate researcher at the Hasso Plattner Institute for Software Systems Engineering, University of Potsdam. Holding degrees in Computer Science from German and Australian Universities, he investigates knowledge-intensive collaboration processes in distributed design from an information technology and design research perspective.

Philipp is a Post-Doc at the Center for Design Research at Stanford University. He holds a Ph.D. in Mechanical Engineering from Stanford University and an MBA from the University of St. Gallen, Switzerland. His research interest is in understanding how to create and manage high-performance engineering design teams. His experience includes work for major tier one and two automotive manufacturers, in the defence industry and in teaching design thinking in the US and abroad.

Alexander Zeier is Deputy Professor of Prof. Hasso Plattner for the Enterprise Platform and Integration Concepts research group at the Hasso Plattner Institute, University of Potsdam since April 2006. He focuses on Enterprise Platforms, In-Memory-based Business Applications, RFID Technologies and Supply Chain Management.

Larry is a Professor in Mechanical Engineering at Stanford University and the Director of the Center for Design Research (School of Engineering). His design thinking and informatics research is concerned with understanding, supporting and improving design practice, including issues in research methodology, team dynamics (co-located and distributed), innovation leadership, interactive design spaces, collaboration technology, and design-for-wellbeing.