A Platform for the Temporal Evaluation of Team Communication in Distributed Design Environments

Matthias Uflacker, Alexander Zeier
Hasso Plattner Institute for Software Systems Engineering, Potsdam, Germany
{uflacker|zeier}@hpi.uni-potsdam.de

Abstract

We present a flexible approach to the observation of multi-modal communication streams over the course of project-based collaboration such as engineering design. We introduce team communication networks to formalize the occurrence and evolution of actors, resources, and semantic relationships in distributed design information spaces. Analyzing the digital traces of IT-mediated collaboration, we computationally construct team communication networks in an unobtrusive and progressive manner, thus providing a live view into the digital communication and information sharing activities of design teams. A service-based team communication network platform has been implemented and applied in distributed engineering design projects to enable the temporal analysis of information sharing activities and to support in the identification of characteristic communication patterns in distributed collaboration groups.

Keywords: CSCW, Design Information Spaces, Team Communication Networks, Semantic Web

1. Introduction

The progressing transition to digital team interaction and IT-supported collaboration creates heterogeneous landscapes of distributed and disconnected clusters of project-relevant information. The multi-modality of team communication channels is obviously supported by the individual selection of appropriate and available tools for each particular design situation. Email, Wikis, and online services for sharing pictures, videos or general documents only represent a selection from the field of applications used for disseminating information across team members and other stakeholders [1].

This heterogeneity in the representation and location of project information contributes to the complexity and diffusion of team information spaces, challenging the team with the achievement of common ground and information context [2]. Moreover, the management and analysis of the team process is hampered by the difficulty to observe and assess multi-modal, distributed communication activities, especially in larger and long-running projects. Having insight into the history and status quo of communication metrics such as individual participation, internal or external influences and relationships would allow to reveal hidden communication signatures and to identify undesirable trends early in the process.

Our work presents a software tool 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 store and query the appearance and relationships of 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 shared information growing, different traces of communication become available for the automated extraction of associations to other (hyperlinked) information and actors. We are able to translate arbitrary types of relationships into ‘team communication networks’ that represent a set of statements about the content and semantics of an information space.

The automated creation of a formal graph model for typed and time-annotated information resources and relations presents an unobtrusive approach to analyze communication patterns and the evolution of complex, multi-modal information sharing activities in collaboration processes. We have implemented d.store, a Web-based service platform for team communication networks, which has been applied to capture and evaluate the communication behavior of eleven distributed engineering design projects. This paper will introduce team communication networks and presents their application in the d.store platform.

2. 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. It identifies multiple correlations between
3. Team communication networks

We introduce team communication networks as a graph representation to express directed relationships (edges) between actors and information resources (nodes). Literal value properties can be assigned to nodes and edges in form of attributes. All network entities (nodes, edges, and attributes) are typed, meaning that they are instances of one, possibly more semantic concepts. Through the individual and flexible typification of nodes (e.g. person, email, wiki) and relations (e.g. sent, reply_to, author) a formal understanding about the contents and relationships of a shared information space is generated.

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 $(cls, t_i, t_e)$ with $cls$ being a list of assigned node types $c \in T_v$, and two timestamps $t_i, t_e \in N$ and $t_i \in N \cup \infty$ with $t_i > t_e$ 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 $(s, r, o, t_i, t_e)$ where $s \in V$ is the source node, $r \in T_e$ is the relation type, and $o \in V$ is the target node. Analogically, $t_i$ 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 $(s, p, val, t_i, t_e)$, $s$ 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. $t_i$ and $t_e$ are defined as before and define an attributes validity period.

Figure 1 shows a small example for a team communication network. In 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 in email $b$.

With the introduction of two timestamps $t_i$ 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 $t_i$ 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, $t_i$ 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 communication network's evolution over the course of a project and enables the exploration of previous states of an information space.

3.1. 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 of nodes, edges and attributes, the semantic of contained entities is provided by a set of domain-specific ontologies. The ontologies describe concepts and relationships for individual applications used for communication. Based on these definitions, a formal conceptual model for the existence and constraints of the network building blocks is created, which allows to infer 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 sections, we give examples for domain ontologies and inference rules for Web-based collaborative editing of Wiki pages and email communication.

3.2. 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 to project partners in order to disseminate insight and knowledge across the team. This usually happens in different ways, often by sharing URLs via emails or discussion groups. Thus, Web resources do not only explicitly refer to other hyperlinked Web resources, but are also linked from arbitrary other resources. By defining two relationship types hyperlink 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. Wiki applications that are keeping logs of page revisions and changes allow the examination of relationships between created content and contributors. Inspecting account names assigned to the revisions can identify authors having created or edited a Wiki page. Additionally, by inspecting the delta of two revisions, the individual contribution of an user becomes evident.

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.

3.3. Email communication

Email messaging is one of the most prominent technologies used today to digitally exchange information in an asynchronous and reliable manner. Studies reveal that global virtual teams largely rely on daily email communication [6]. 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 the ontology that defines the 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 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_to relationship types. Email addresses are assigned to persons via the address attribute. Bi-directional relations between an email and sending or receiving messages without needing to organize and participate in synchronous interactions.

![Figure 2. Two domain ontologies for basic Web and Wiki-related information sharing concepts.](http://hgs-web.de/ws/dstore/web/0.1/)

![Figure 3. A domain ontology for email communication.](http://hgs-web.de/ws/dstore/email/0.1)
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.

3.4. Rule-based inference of node relationships

We apply a set of rules to infer relations between information objects and persons, based on node attribute values. The computational inference of relationships decreases the number of relations that need to be explicitly stored in the graph, leading to increased flexibility and less overhead in the manipulation of networks, nodes and attribute values. Rules are defined as pairs of pre-conditions and implied post-conditions, presented here in the form antecedent ⇒ consequent. If the left-hand pre-condition of a rule is met for any combination of nodes in a network, then the right-hand post-condition 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 form:

\[ \text{from_address}(x,y) \land \text{address}(z,y) \Rightarrow \text{sender}(x,z) \]

The rule 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 relationship to node \( x \). Analogical rules can be constructed to spawn direct relations from an email message to its recipients:

\[ \text{to_address}(x,y) \land \text{address}(z,y) \Rightarrow \text{recipient}(x,z) \]
\[ \text{cc_address}(x,y) \land \text{address}(z,y) \Rightarrow \text{recipient}(x,z) \]

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

\[ \text{create_account}(x,y) \land \text{wikiname}(z,y) \Rightarrow \text{author}(x,z) \]

We can also define rules to instantiate relationships between two information objects. The reply_to relationship between an email and a preceding message to which the sender of the email replies is an example. 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, thus, the inverse reply relations) on the occurrence of two email nodes with matching ID attributes:

\[ \text{reply_to_id}(x,y) \land \text{message_id}(z,y) \Rightarrow \text{reply_to}(x,z) \]

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

\[ \text{received}(w,x) \land \text{subscriber_address}(w,y) \land \text{address}(z,y) \Rightarrow \text{received}(z,x) \]

This rule creates a received relation between emails sent to an EmailList and persons that 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 encoded explicitly in the graph. All other relationships and node types can be inferred by a team communication network system based on rules and the attribute values that are assigned to the four nodes.

4. Platform implementation

A system of team communication networks features a set of formalized ontology definitions and network instances, which are defined and encoded in a standardized format. The Resource Description Framework (RDF) [7,8] and the Web Ontology Language extension OWL [9] provide a suitable framework to encode the models. Both standards are used for the specification of domain 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 namespace imports. This allows us to decouple concept and instance models of multiple team communication networks and independent customization of network concepts without affecting other models.

To validate our approach, we have implemented d.store, a resource-oriented platform for team communication networks [10]. The platform provides a REST-based [11] service interface for accumulating information and actor relationships from observed collaboration archives. Every network instance and every individual node in a network is an 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 querying and manipulation of network states, such as exploring node properties or creating a new resource, by using the standard operations 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.

Our system is built on a customized semantic web framework, which is used to organize the concept and instance models. An OWL reasoner library interprets the inference rules, which have been expressed in the Semantic Web Rule Language SWRL. Team communication networks are gradually generated through posting identified objects and associations to the resource-oriented service interface provided by the platform. The information is encoded into asserted RDF statements and stored in a relational database. Adopting the semantic 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. The 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 model provides the basis for reasoning on and analysis of the constructed network.

A number of helper applications feed information and relationships to the platform, which have been identified by processing communication data. Read and write access to the networks is provided through the unified interface of the HTTP/1.1 protocol [12]. \( \texttt{d.store} \) supports a number of representation formats for the resources of a network, including RDF/XML, JSON, GraphML, and HTML. Clients can make use of the \textit{Accept} header field of HTTP/1.1 to negotiate the representation type that is returned from the server upon request.

### 5. Application

Starting to use the platform for the evaluation of multi-modal communication behavior of distributed design teams, we analyzed data collected in eleven global engineering projects, which were running for a period of nine month. The projects were placed in a joint academic partnership between an US-based university and six global institutions, with distributed project teams composed out of global and local students. The teams featured a multi-disciplinary set-up, involving students with backgrounds in mechanical engineering, software engineering, economics, or industrial design. The project teams have been working independently on prevailing engineering design tasks that were accompanied by global enterprises and corporate liaisons under 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 in terms of start and end dates, major milestones 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 continued research in asynchronous information sharing activities via email, Wikis, and document sharing platforms. 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 in public online folders. The average ratio between emails being sent and the number of direct relationships to other

![Figure 4. Import hierarchy of RDF/OWL graphs in the \( \texttt{d.store} \) platform.](image)

![Figure 5. Platform architecture with clients accessing the resources provided by the service interface.](image)
information resources submitted in the form of attachments and hyperlinks was approx. 1 to 0.6. Obviously, emails were 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, contextual view on team communication and information sharing processes.

To give an example for the utilization of the d.store platform, we scanned the individual sending behavior extracted from a team’s email list. Metrics we were interested in imply the individual number of sent and received emails as well as the number of replies and referenced Web resources. Figure 6 shows an interactive motion chart to track the evolution over the course of the project. Each circle represents a person, color-coded by its individual role in the project (e.g. local or global team member, corporate contact, etc.). 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 traced individuals represent an obviously highly involved and responsive corporate contact person (left) and an active member of the US-based project team. The visualization also shows that most other actors in the project are only marginally involved in email communication.

6. Conclusions

We have presented a non-interfering approach for the temporal observation of different IT-mediated communication streams in distributed design teams. Team communication networks provide the data structure required to describe the occurrence and evolution of actors, resources, and their semantic associations in distributed design information spaces. By processing data archives that are generated by using favored collaboration tools, no additional interaction and overhead is introduced to the workflow of process participants. The d.store platform offers a resource-oriented service interface to social information networks, providing distributed access to context and relationships of de-central information objects and individuals. We have demonstrated the application of team communication networks in realistic project settings, where we have concentrated on the evaluation of email and Wiki usage. However, the platform remains general applicability by being open and extensible to other domain ontologies. Our research continues with a detailed exploration of the collected data and communication signatures. In particular, we will use the system to evaluate characteristic elements in the communication behavior of global design teams and will analyze the potential impact that these signatures have on team performance.

References