From Groupware to Social Media
A Comparison of Conceptual Models

Nils Jeners[1] , Wolfgang Prinz[2]

[1] RWTH Aachen University, Informatik V, Ahornstr. 55, 52056 Aachen,
nils.jeners@rwth-aachen.de
[2] Fraunhofer FIT, Schloss Birlinghoven, 53754 Sankt Augustin, Germany,
wolfgang.prinz@fit.fraunhofer.de

Abstract. Although Groupware research has yielded a number of productive and successful systems, it appears that the current social media trend is somewhat out ruling the traditional cooperation support systems such as email or shared workspaces. In this paper we propose a conceptual model that identifies major elements and concepts of cooperative systems to provide a basis for their comparison. We will illustrate that social media systems apply the same basic concepts as other collaborative systems but with a different adoption creating a different user experience.

Keywords: groupware, social media, meta model, conceptual model

1 Introduction

Groupware has a long history [1]. All systems belonging to this class of applications are trying to help workers to organize their work and get it done. A lot of research and developments has been made for professional scenarios. Social Network Sites which rise since 1997 [2], evolve from leisure or non-professional use cases within groups of friends or people with same interest. These systems do not try to represent business workflows or support organizational tasks. These systems try to keep leisure cooperation simple, e.g. arrange an evening with friends, or share pictures from a party.

Nowadays these systems are also used in professional environments. Not for marketing purpose only, but also for organizational communication (yammer.com) and setting project meetings (doodle.com) for example.

Big companies such as Cisco copy social network concepts of systems like Facebook and many others. Business applications like Cisco's Quad[1] and Jive[2] evolved trying to bring social systems into a professional environment.

If we anticipate that there are groupware systems on the professional side, and social media on the leisure side, we want to identify the influences each of the system has to one another. Further we want to ask whether these two types remain on each

[1] www.cisco.com/web/products/quad/
[2] www.jivesoftware.com/

side or merge and evolve to one system class that is either for professional and leisure usage scenarios.

Therefore we developed a generic model which can represent all the systems and tools that we are using in daily work and private cooperation. This generic model consists of abstract classes that are instantiated with each analyzed system. With the generic model in mind we are able to compare the systems component by component and find out whether the systems are based on similar concepts with different shapes, or not.

Common underlying concepts provide the potential to transform one system in to another. Legacy groupware systems can evolve and be enhanced to social systems, keeping the same set of features with another shape or different UI. To identify the key concepts and point on possible transformations we try to apply the meta model.

In the following we first present related approaches in modeling cooperative applications. This is followed by the presentation of our conceptual model and its instantiation for two example systems. Afterwards we will compare the particular adoption of core concepts by groupware and social media systems.

2 Background

A lot of models help to classify groupware in general. The Space/Time matrix of Johanson [3] and the 3C-Model of Teufel [4] are the most famous of them. These models are suitable as a taxonomy, however an instantiation of these models is not possible.

The Zachman Framework is an early framework for information systems architecture [5] that can also be applied on groupware systems. Its basic concepts are roles and perspectives, which also fit on groupware and social media. The framework of Zachman is quite useful in terms of planning and developing applications. Like a pattern language it provides a mutual understanding for all stakeholders.

The basic building blocks of group communication support systems are roles, message objects, functions and rules [6]. A so called CSCW system with the above mentioned building blocks access a common underlying system which provides services to applications and user access. TOSCA [7] and MOCCA [8] understand CSCW systems as a heterogeneous collection of applications, paradigms and models and not a single system. Within this environment there are models that specify the environment. The four presented models (informational model, organizational model, workspace model, and room model) are considered as perspectives with an abstract view on the environment functionality.

Three aspects of groupware concept models defined by Ellis et al. [9] are the ontological model, the coordination model and the user-interface model. The ontological model consists of objects and the operations on these. Objects are modeled with attributes and values. Values can either be atomic or other objects. The operations are divided in four classes, namely view, create, modify, and destroy. Objects own an intended semantics or an operational semantics. The coordination model covers the dynamic aspects in terms of activities. Activities are performed by

actors with a specific role. Procedures are sets of activities. The third aspect is the user interface, the appearance of the system, the user experience.

The reference architecture proposed in [10] identifies several layers as well as architectural components. Most relevant for our approach are the basic services that realize access to the underlying data structures which implement the concepts identified in this paper. The generic CSCW model, proposed by Farias et al. [11], consists of four concepts, namely activity, actor, information and service, whereas activity is the core that connects everything. An activity consists of a goal and a state. It is performed by an actor, uses information and supports services. The ARCON framework provides help in order to understand, design and implement collaborative networks [12]. A reference architecture that helps enterprises to cooperate in virtual enterprises is VERAM [13].

In summary, several approaches to identify generic architectures and building blocks exist. However, a comprehensive conceptual model that enables the modeling and comparison of different collaborative applications is yet missing. We will present our approach in the next section.

3 Conceptual Models

In our approach, we first defined elementary actions the user can perform at existing system like Email, Twitter, etc. We categorized these activities to generalize them in a meta model which covers these system. We instantiated the abstract classes with the applied techniques of the existing systems.

3.1 Meta Model

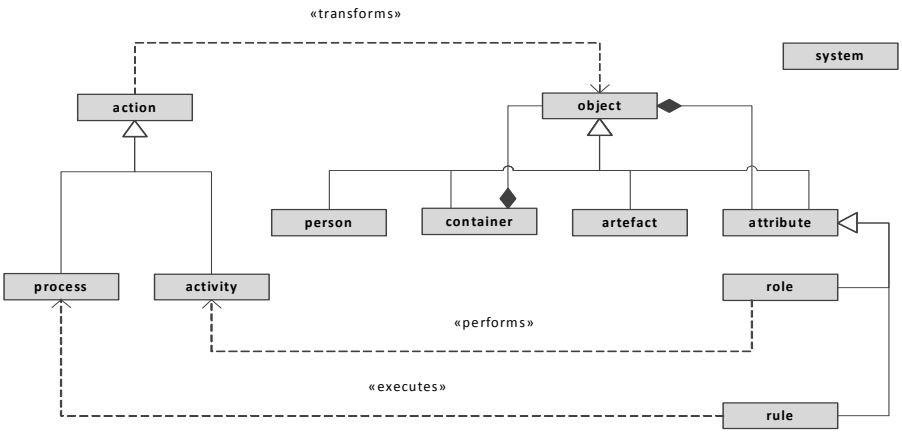

Fig. 1. Meta model of cooperative systems.

Our proposed meta model (see fig. 1) consists of abstract classes, that describe the generic system for cooperative applications. Each system implements the abstract

classes and inherits from them. The three top-level classes are *action*, *object* and *system*. *Action* denotes all activities and processes performed by either users of the system or the system itself. *Object* represents all virtual entities of the system namely *person, container, artifact and attribute*. The system class contains all systems represented in the meta model. Every implemented class of the model belongs to either one or several systems.

The *action* class is specialized in the two subclasses *activity* and *process*. *Activities* can be performed by persons who have a specific role and *processes* can be executed by the system itself, controlled by a rule. Both subclasses can transform (create, modify, and destroy) specific objects, e.g. create a new container, modify an attribute, or destroy an artifact.

Object is the abstract parent class of *person*, *container*, *artifact* and *attribute*. *Person* stands for every humane actor in the system, i.e. user, specified by *roles*. *Containers* are collections of objects. They can contain containers itself, persons and artifacts. *Artifacts* denote the basic entities of the system such as messages, files or any other objects the users can interact with. *Attributes* belong to objects and cannot stand alone. They are always attached to other objects (person, container, and artifact. The two attributes, already implemented are *role* and *rule*. *Roles* specifies person in terms of the activities they can perform. We distinguish between four kinds of roles: Organizational roles describe people in the context of their work hierarchy, e.g. boss, colleague, and partner. Activity roles specify the activities one person can perform, e.g. author, reader. Right roles show the rights a particular person has, e.g. manager, owner. The last kind of roles is the cultural roles. *Rules* are expressions that can stick to objects and if this rule is valid, a *process* will be performed. An example for this kind of *rule* is an autoreply mechanism in email systems or a notification mechanism in shared workspaces.

3.2 Instantiated Systems

We instantiated the proposed meta model exemplary with two common cooperation system. We compare email (Fig. 2) on the one hand with the social media system Yammer (Fig. 3) on the other. The instantiated models of the systems are on a more generic level, to not get lost in details and fit into this paper.

Fig. 2 shows the basic building blocks of an IMAP email system like it is implemented in Outlook or Thunderbird. It has mails, contacts, folders. Mails are structured in several folders called inbox, outbox, etc., depending on their status (send, received, etc.) and have attributes e.g. from, to, subject, body, etc. There are three roles: author, sender, and receiver, which perform certain activities like write, send, and read.

Fig. 3 shows the enterprise social network called Yammer. It mainly consists of posts in certain networks (groups). Posts can contain polls, events, embedded images of videos and different other content. Posts can be liked (as in Facebook) and tagged (called topic) to be searchable. There also are three roles called admin, author, reader, which perform activities like read, write, tag, invite.

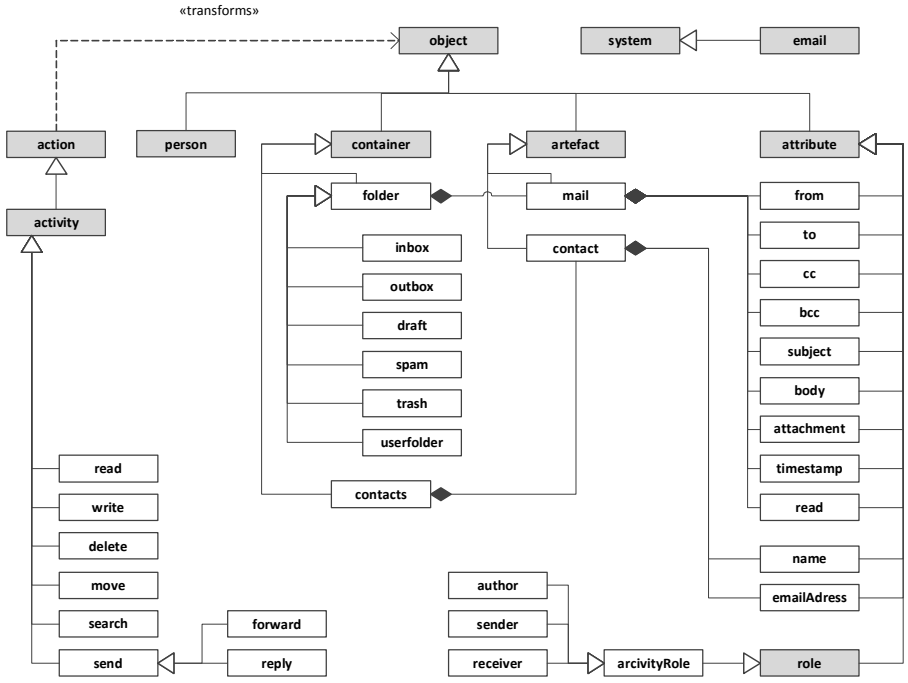

Fig. 2. The instantiated meta model with email.

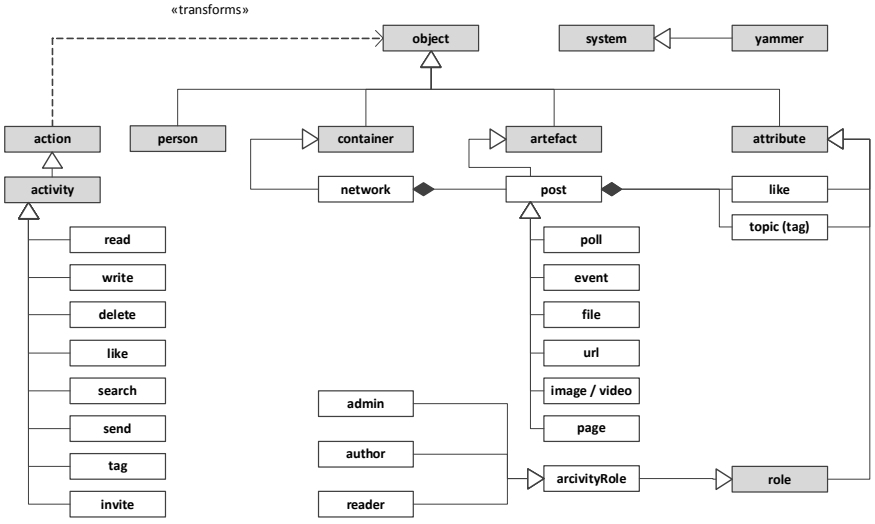

Fig. 3. The instantiated meta model with Yammer.

The two systems are mainly used to exchange certain information with others. Email applies a sending metaphor, where the author writes a message and sends it to the desired receiver. The data (mail) is not owned by the author or the system. The author and the receiver, they both hold a copy. Yammer provides a sharing metaphor.

The author writes a post, and saves it in the system. The data (post) is stored in the centralized system and everybody (with respective rights) can access the data.

While the appearance of these two systems is different, the purpose of the systems is the same. Whether it is called mail or post, does not change the entity which holds the information, but it indicates a different concept.

3.3 Comparison of Core Concepts

In this section we compare the particular instantiation of core concepts and their effect. First we look at the people concept group as well as the relation of people into larger sets. The following table 1 identifies four relations.

Table 1. Relation of people.

	mutual	container	visibility	purpose
group	yes	yes	public	Access management
friends	yes	no	public	Notification management of sender/receiver, publicity two-way
follower	no	no	public	Notification management of receiver, publicity one-way
circle/lists	no	yes	private	Notification management of sender

The classic group concept is primarily used in shared folder or teamroom systems, mainly for the purpose of defining access rights. Social media systems relate people by friends and follower networks or in circles (Google+). The difference is that groups are symmetric, while the networks can be asymmetric. This means that all members of a group know each other, i.e. each user knows that the information he provides is accessible to all group members. Friends and follower networks are not as transparent, since they are created individually by each user. Thus two users who belong to the same network of another user do not necessarily know each other. Thus information sharing is more directed to the personal network of a user and not to a symmetric group. In an organizational context this can cause problems as it is often not clear if important information is received by all required people in case that they did not configure their own network accordingly.

Another aspect is the relation and organization of objects into container illustrated in the following table 2. Again the classic approach is the folder approach that organizes objects into a hierarchical order.

Table 2. Container of objects.

	cardinality	container	visibility	purpose
folder, workspace	1:m	yes	public	access management, hierarchy, group
tags	m:n	no	public	filtering

Most social media systems apply tags to organize and structure object. The advantage is an easier networking of information as long as tags are applied in a disciplined way.

This brief comparison of the implementation of relations of two core concepts indicates that social media systems apply a network-based relationship while classic applications apply a group- or folder-based approach.

Table 3 shows how information is shared among the different systems. In common shared workspace systems, a group respectively a community is the entity in which information is shared. The members of a group are known to everybody within the group. This concept is used to map existing groups, e.g. project groups. A follower network is established by people who are interested in posts from the author. This concept is applied in Twitter and can be compared to subscriber systems of newsfeeds. Walls like in Facebook often tend to be semi-public. A wall belongs to one person and everybody can write to the walls of friends which is not the same as sending a message how it is done in inbox systems. A sender actively decides who the message receives.

Table 3. Sharing / Messaging / Activity stream concepts.

	responsibility	visibility	purpose
groups, communities	admin or all members	public/private	common goal/interest
follower	user	private	user interest
walls	friends	friends	public friendship
inbox	contacts	private	1:1 messages

4 Conclusion

This paper presents a conceptual model enabling the instantiation for different collaborative applications and thus their comparison. We have illustrated the instantiation for email as well as the social media system Yammer. The comparison has shown that most model elements are applied by both, yet in a different manner.

The comparison has shown that the two systems do not differ in a great manner, but rather in small pieces how something is called at what core concept is followed.

With believe that this paper contributes to a more systematic understanding of the core elements of collaborative applications. Our next steps will focus on further applications of the model with the aim of further refinement and validation.

References

1. Ellis, C., Gibbs, S. and Rein, G. 1991. Groupware: some issues and experiences. Commun. ACM 34, 1 (January 1991), 39-58.
2. boyd, d. m. and Ellison, N. B. 2007. Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 13(1), article 11.
3. Johansen, R. 1991. Leading business teams. Reading. Addison-Wesley.
4. Teufel, S., Muelherr, T. and Bauknecht, K. 1995. Computerunterstützung für die Gruppenarbeit. Bonn: Addison-Wesley.
5. Zachman, J. 1987. A framework for information systems architecture, IBM Systems Journal, v.26 n.3, p.276-292, DOI = 10.1147/sj.263.0276
6. Prinz, W. and Pennelli, P. 1992. Relevance of the X.500 Directory to CSCW Applications. In Groupware: Software for computer supported cooperative work, D. Marca and G. Bock (eds.), IEEE Computer Society Press, pp. 209-225.
7. Prinz, W. 1993. TOSCA: Providing Organisational Information to CSCW applications. In Proceedings of ECSCW '93: Third European Conference on Computer Supported Cooperative Work, G.d. Michelis, K. Schmidt, and C. Simone eds., Milan, Italy, Kluwer Academic Publishers, Dordrecht, pp. 139-154.
8. Benford, S., Mariani, J., Navarro, L., Prinz, W., and Rodden, T. 1993. MOCCA: An Environment for CSCW Applications. In Proceedings of Conference on Organizational Computing Systems, S. Kaplan eds., Milpitas - California, ACM Press, 1993, pp. 172-177.
9. Ellis, C. and Wainer, J. 1994. A conceptual model of groupware. In Proceedings of the 1994 ACM conference on Computer supported cooperative work (CSCW '94). ACM, New York, NY, USA, 79-88. http://doi.acm.org/10.1145/192844.192878
10. Vassilios Peristeras, Maria Antonia Martinez-Carreras, Antonio F. Gomez-Skarmeta, Wolfgang Prinz, Peyman Nasirifard, "Towards a Reference Architecture for Collaborative Work Environments", in International Journal of e-Collaboration, 2010, Vol 6., No. 1
11. Farias, C. R. G., Pires, L. F. and van Sinderen, M. 2000. A Conceptual Model for the Development of CSCW Systems, In Designing Cooperative Systems: The Use of Theories and Models (Proceedings of COOP 2000), Sophia Antipolis (France), IOS Press, pp. 189--204.
12. Camarinha-Matos, L., Afsarmanesh, H. 2006. A Modeling Framework for Collaborative Networked Organizations. In: Network-Centric Collaboration and Supporting Frameworks, Vol. 224. 3-14. Springer Boston.
13. Zwegers, A., M. Tolle, and J. Vesterager (2003). VERAM: virtual enterprise reference architecture and methodology. In I. Karvonen, R. Van den Berg, P. Bernus, Y. Fukuda, and M. Hannus (Eds.), Global Engineering and Manufacturing in Enterprise Networks, Volume 224, pp. 17–38. VTT.