

A Meta-Model for Cooperation Systems

Nils Jeners¹, Wolfgang Prinz², and Sebastian Franken¹

¹ RWTH Aachen University, Informatik V, Ahornstr. 55, 52056 Aachen
{nils.jeners, sebastian.franken}@rwth-aachen.de

² Fraunhofer FIT, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
wolfgang.prinz@fit.fraunhofer.de

Abstract. Today's enterprise information and cooperation systems are numerous and diverse. Users have problems of information overload, task interruption and media discontinuity. A unification approach of existing systems would be supportive to overcome these issues by reducing the variety and thereby the complexity. A process of unification starts with the comprehension of existing systems belonging to the virtual environment, by modeling them. The second step is the unification on the model layer followed by the last step, the design of the user interface. This paper will present the first step, a meta-model for cooperation systems, which applies concepts used in professional groupware systems as well as social media platforms. Major elements and concepts of cooperation systems will be identified to provide a basis for the unification of the virtual environment.

Keywords: cooperation system, meta-model, conceptual model, unification.

1 Introduction

The virtual environment of knowledge workers consists of cooperation systems, social media, and personal as well as enterprise information systems. It tends to grow and become complex. With increasing diversity of those systems, complexity for users increases equally. This complexity leads to problems of information overload [1], task interruption [2] and media discontinuity. Although groupware research [3] has yielded a number of productive and successful systems, it appears that the current social media trend [4] provides tremendous alternatives to traditional cooperation support systems such as email or shared workspaces. The shift from document links to people links was already observed in 1998 [5], so that today's groupware systems such as Lotus Notes, BSCW [6], and SharePoint blend with current social media platforms such as Facebook, and Twitter [7]. Furthermore, new kinds of systems like Yammer, Jive, and Cisco WebEx Social arise.

Nowadays, numerous systems exist side by side with overlapping functionality and disjoint contexts. However, discussing which system might be more appropriate for particular cases, does not help users to overcome this challenge. Even professionals are overwhelmed with this situation, so that a future-oriented approach is to accept the diversity of existing systems and utilize them in one single interface which provides the user every needed function.

In this paper, we present the first step of the unification approach: A coherent model which illustrates that social media systems apply the same basic concepts as other collaborative systems, but with a different adoption, thus creating a different user experience. The major elements and concepts of cooperative systems will be identified to provide a basis for the comparison and later on the integration of existing systems. The presented meta-model helps to reflect the features of groupware systems as well as social media platforms with the aim to identify the building blocks of those systems and to understand in which respect they overlap or differ from each other.

2 Background

Conceptual information models provide application modeling in a direct and natural way by offering semantic terms, such as entity, activity, generalization, aggregation, etc. [8]. There are several generic modeling techniques, such as (Extended) Entity-Relationships Models [9], UML [10], Telos [11], and ConceptBase [12]. They are generic and developed for the purpose to support developers and system architects with the design of a respective system.

One of the earliest model of information systems, the Zachman framework [13], focuses on architectural models. Other CSCW models focus on concepts and basic building blocks of existing systems [14, 15]. These models cover the basic concepts, but they are neither intended to classify systems nor to discuss with users. Thus they explain the concepts, but they do not assess the different characteristics of these concepts. Two further models are presented in [16] and [17]. The presented conceptual models are very detailed, but they do not help to focus on one aspect. They rather focus on an architectural perspective. The reference architectures proposed in [18, 19] identify several layers as well as architectural components and support to cooperate in virtual enterprises (VERAM). The basic services realize access to the underlying data structures which implement the concepts identified in this paper. The modeling framework ARCON helps to understand, design and implement collaborative networks [20]. Although many models exist, they do not help to focus on how the basic underlying concepts are perceived by users. The presented model approach tries to stay coherent and comprehensive. Applied models support cooperation processes between professionals and users.

3 Meta-Model of Cooperative Systems

Cooperative work is characterized by people and artifacts as well as communication between people and activities performed by people on artifacts [21]. Cooperative systems (groupware and social media) support activities of this kind. The proposed model uses the two entities (artifact and person), and establishes relationships between themselves and one another, resulting in a condensed model of cooperation systems (cf. Fig. 1). Although this model applies an extended entity-relationship model notation, it is not intended to model the database structure in the backend of particular systems. It rather tries to cover the mental model [22] users are developing from just viewing the graphical user interface while working with the system and using its functions.

A cooperation system is modeled by a quadruple $\mathcal{C} := (\mathcal{P}, \mathcal{A}, \mathcal{Rel}, \mathcal{Rul})$ with a set of *People* and a set of *Artifacts* as well as a set of *Relations* and a set of *Rules*.

People $\mathcal{P} := \{p_1, p_2, \dots, p_n\}$ are natural persons, who use a system, i.e. users. Instances of a person can be single persons but basically representatives of a type of person, called role. A role contains certain characteristics, such as the right to perform particular activity. Author, admin, manager, and owner are exemplary roles. Everything that is not a person is belonging to the set of artifacts $\mathcal{A} := \{a_1, a_2, \dots, a_m\}$. Instances of an artifact can principally be every working artifact that is virtually represented by the system. The whole system itself is an artifact, documents of different data types or attributes of documents can be artifacts. This depends on the intended granularity of the model.

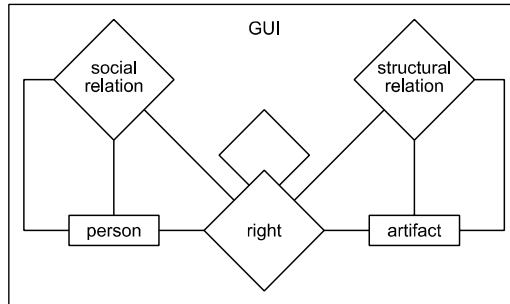


Fig. 1. Meta-model of Cooperative Systems

In the set of the relationships $\mathcal{Rel} := \{R_1, R_2, \dots, R_o\}$, the model includes three types of relationship: social, structural, and rights.

The relationship between two artifacts is called structural relation. This relation establishes a structure of content. It namely structures the artifacts by relating one to another, e.g. “a folder contains a file” indicates a structural relation between these two artifacts. A social relation is the analogous relation between two persons. It establishes a structure of persons. This relation can be symmetric, e.g. “Alice is a friend of Bob”, or asymmetric, e.g. “Alice is following Bob. Social relations between roles are inherited to the individual persons.

Every entity (person, artifact) can own the right to change either an entity or a relation. A person has particular rights to perform certain activities on artifacts. For example, the authorship of an artifact gives a person the right to edit the artifact. Furthermore, a person can change a social relation (e.g. becoming friend of another person), a structural relation (e.g. saving a file in a certain folder), or a right itself (e.g. inviting a person as an editor to a document). Rights of artifacts performing activities on persons are less common. Persons primarily are the active part of the system and artifacts are supposed to be passive objects. Nevertheless, artifacts are becoming active in terms of notifications and other kind of system performed processes.

Persons and artifacts can have the right to change social relations, structural relations, or rights themselves. A change of a social relation for example is to follow another person, or accept a request for friendship. An example for a change of a structural relation would be to move a file into a shared folder. A change of rights

themselves would be to invite a person to a shared folder, which give this person access rights to this particular folder.

The set of rules $\mathcal{Rul} := \{\varphi_1, \varphi_2, \dots, \varphi_p\}$ is modeling dependencies of relations. Rights can be derived from roles of persons, social relations, or structural relations. Social relations determine rights, e.g. a friend of an author has the right to read the artifacts the author created. Structural relations determine rights, e.g. every member of a shared folder has access to all subfolders and containing files.

The model mainly focuses on the internal context of a cooperative system, but the interface between user and system must not be forgotten. Therefore, the model is framed by the graphical user interface (GUI) to clarify the importance of the interface. Moreover, an appropriate GUI is essential to provide features of cooperative systems, e.g. presence and awareness, and make them accessible to users.

Fig. 2 illustrates the basic symbols of the proposed meta-model: persons, artifacts, and the various types of relations. Although it is possible to model everything with the generic symbols, it seems to be more comprehensible to differentiate between different persons and different artifacts, if necessary. Therefore we added a simple artifact (file, document), which holds information and an advanced artifact (folder, container) to apply structural information, like a folder hierarchy. Beside the symbols for entities, Fig. 2 also illustrates the symbols for relations. First, it shows the structural relation between two artifacts, second the social relation between two persons and third the right between respective entities (person, artifact) or between entities and relations.

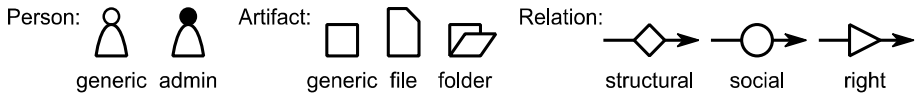


Fig. 2. Symbols of the cooperative system meta-model

4 Model Application and Evaluation

In the following the conceptual differences between cooperation systems will be identified by instantiating the above mentioned meta-model. Every aspect will be completely covered to establish a classification. As aforementioned the instances of the model do not try to model an internal database structure or a technical architecture of a particular system. The instances depict the mental models which are gathered during system usage. The focus is on the two following aspects: The different characteristics of social relationships and the different metaphors of information exchange in virtual environments.

4.1 Social Relationships

Cooperation always happens between multiple people, therefore cooperation systems support multiple users (co-located or distributed). Cooperation always implies a relationship between cooperation partners. This relationship is mapped to the

application logic of the system. To cover the variety of real world relationships, there is a variety of virtual relationships implemented in different existing systems. In the following a classification of virtual relationships is given by presenting the distinguishing attributes. In general we identified three basic types of relationship: friends, followers and groups (cf. Fig. 3). But there are hybrid forms combining the features.

Existing classifications emerged from a graph theory and social network analysis perspective. Virtual relationships of users can form chains (process oriented), stars (egocentric, dominant member), hierarchies (organogram of a company), and networks (project oriented) [23]. But every system which supports relations between users can be applied to form the above mentioned types. These types rather describe the usage or the user group of one system, but it does not encounter the core of the character of social virtual relationships.

A **friend network** is applied in Facebook. Its virtual network $\mathcal{N} := (P, R)$ consists of a set P of persons and $R := P \times P$ an undirected relation between two persons. Two persons p_i and p_j are friends if and only if $(p_i, p_j) \in R \wedge (p_j, p_i) \in R$, with $p_i, p_j \in P$ and $p_i \neq p_j$. In other words, a friendship is mutual if both persons agreed; the relation of a friend network is symmetric and irreflexive. In Fig. 3 (1) persons A, B, and C are friends.

A **follower network** is applied in Twitter. Its virtual network $\mathcal{N} := (P, R)$ consists of a set P of persons and $R := P \times P$ a directed relation between two persons. A person p_i is follower of a person p_j if and only if $(p_i, p_j) \in R$, with $p_i, p_j \in P$ and $p_i \neq p_j$. Person p_j is a so called followee of person p_i . The relation is irreflexive. In Fig. 3 (2) persons A and B are following each other, B is following C and C is following A.

A **group network** exists in Facebook groups, Yammer networks, or folders in shared workspace systems. The attached functions differ from concept to concept, but the underlying base concept remains the same. One abstract artifact is linked to people who interact or share something. The relation between persons is no more direct; it is indirect via an artifact, e.g. discussion thread, files.

The group network $\mathcal{N} := (P, G, R)$ consists of a set P of persons, a set G of groups, and $R := P \times G$ a relation between persons and groups. A person $p \in P$ is member of a group $g \in G$ if and only if $(p, g) \in R$. The relation R is called bipartite, because persons and groups do not have a relation among themselves. In Fig. 3 (3) persons A, B, and C are members of the folder f and thereby they have an indirect relation via this folder.

Group networks that are not bipartite also exist. A network $\mathcal{N}' := (P, G, R, S)$, with P, G , and R analogous to the group network \mathcal{N} . Additionally a relation $S := G \times G$ between groups exists. With it a group hierarchy could be established. A group $g_i \in G$ inherits its members to a group $g_j \in G$ if and only if $(g_i, g_j) \in S$. All p_i with $(p_i, g_i) \in R$ are members of group g_j too, but not vice versa. An example for this type of group network is the shared workspace metaphor.

Hybrid networks contain several of the above mentioned features in parallel. For example, Facebook has a combined friend and group network. Furthermore, being a

fan of a celebrity applies to a follower network. In Fig. 3 (4) persons A and B are friends, C is following A, and B and C are sharing a folder f .

By observing existing systems the above classes of networks were discovered. The dimensions of relations are as follows. *Reciprocity* indicates whether a relationship is mutual (friends) or one-way (follower). *Directedness* indicates whether a direct relationship exists (friends) or an indirect one via an artifact (groups, shared folders). *Visibility* denotes whether a relation is visible to just one person (lists, circles), visible to every involved person (private groups), or visible to everybody (public friendship). *Cardinality* indicates whether a relation involves only two persons (friends), or several persons (group). The found network types are not convertible into each other without loss of information.

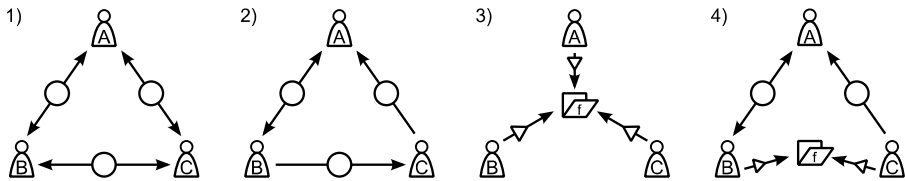


Fig. 3. Four classes of social relationships and virtual networks in cooperative systems: 1) Friend network, 2) Follower network, 3) Group network, and 4) Hybrid network

4.2 Information Exchange

If the network of people is the basic social structure of cooperative systems, the exchange of information is the base concept in terms of activities within these systems. Independent of performing communication, cooperation, or coordination, every activity can be reduced to steps of information exchange. Within this concept, we differentiate between two sub concepts: Sending and sharing.

Sending includes every activity which transfers the sent artifact to the communication partner, i.e. it is copied. Fig. 4 (left) illustrates an instance of a communication system. Person A owns a folder o (outbox) which contains a created message m . While A is sending m to person B, a copy of m , is saved in the inbox (i) of B. This is a typical procedure in e.g. email. As opposed to this, sharing includes every activity which only affects the rights from a particular person to a particular artifact. The same situation as before with sharing (cf. Fig. 4 right): Person A saves message m in the conversation folder c . If person B has already joined this conversation, the right to read m can be derived. If not, the activity of person A leads to the change of the right person B has in respect to the conversation.

Sending concepts can be found in messaging and chat systems, like email, but they are not bound to communication. Sending a file attachment definitely belongs to cooperation and applies to the sending concept too. Sharing concepts can be equally found in communication and cooperation scenarios. Facebook and Twitter, for example, apply this concept in their support how people communicate. The users are able to store messages on their server and people share different containers, e.g. walls, where messages are aggregated. People can post messages to these containers, i.e. they share a message, in contrast to send a message to a specific person. In cooperation scenarios the sharing concept is more obvious. A shared folder like we have in Dropbox indicates that several persons have access to it and can store files for the others.

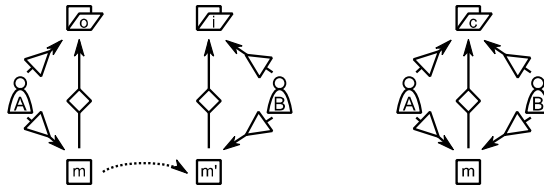


Fig. 4. Two classes of information exchange in virtual communities: Sending and sharing

These two concepts exist in different forms and can be characterized by several dimensions. *Reciprocity* indicates whether a communication is bilateral (conversation) or one way (broadcast). *Directedness* indicates whether a message is directed to a person (push) or whether it is not directed, e.g. stored in a folder (pull). *Visibility* denotes whether a conversation is visible to just two persons or a group (private) or visible to everybody (public). *Cardinality* indicates how many persons are involved.

5 Conclusion

This paper presents a meta-model of cooperation systems enabling the instantiation for different collaborative applications and thus their comparison. We have illustrated the classification of social relationships and people networks as well as the difference between the concepts of sending and sharing. The comparison has shown that the basic concepts resemble by a variety of different systems, only differing in their appearance. We believe that this paper contributes to a more systematic understanding of the core elements of collaborative applications. This work is just the first step for a unification approach of several cooperation systems. Our meta-model and with it the modeling language sets the basis for a precise understanding of cooperation systems, their comparison and their integration.

Our next steps will focus on further applications of the model with the aim of further refinement and validation. Thereby we aim to bridge cooperative use cases to technical systems and further to the design of user interfaces.

This work of this paper has been partially funded by the European Commission under the Seventh Framework Programme as part of the GRANATUM project.

References

1. Eppler, M.J., Mengis, J.: The Concept of Information Overload - A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines. In: Meckel, M., Schmid, B.F. (eds.) *Kommunikationsmanagement im Wandel*, pp. 271–305. Gabler, Wiesbaden (2004)
2. Speier, C., Valacich, J.S., Vessey, I.: The influence of task interruption on individual decision making: An information overload perspective. *Decision Sciences* 30, 337–360 (1999)
3. Ellis, C.A., Gibbs, S.J., Rein, G.: Groupware: some issues and experiences. *Communications of the ACM* 34, 39–58 (1991)

4. Boyd, D.M., Ellison, N.B.: Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication* 13, 210–230 (2007)
5. Hoschka, P.: CSCW research at GMD-FIT: From basic groupware to the Social Web. *ACM SIGGROUP Bulletin* 19, 5–9 (1998)
6. Appelt, W., Busbach, U.: The BSCW system: a WWW-based application to support cooperation of distributed groups. Presented at the
7. Prinz, W., Kolvenbach, S.: From Groupware to Social Media — Extending an Existing Shared Workplace System with Social Media Features. *Information Technology* 54, 228–234 (2012)
8. Mylopoulos, J.: Information modeling in the time of the revolution. *Information Systems* 23, 127–155 (1998)
9. Chen, P.P.S.: The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS)* 1, 9–36 (1976)
10. Object Management Group: *OMG Unified Modeling Language™ (OMG UML), Infrastructure, Version 2.3* (2010), <http://www.omg.org/spec/UML/2.3/Infrastructure>
11. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: Representing knowledge about information systems. *ACM Trans. Information Systems (TOIS)* 8, 325–362 (1990)
12. Jarke, M., Gallersdörfer, R., Jeusfeld, M.A., Staudt, M., Eherer, S.: ConceptBase—a deductive object base for meta data management. *J. of Intelligent Information Systems* 4, 167–192 (1995)
13. Zachman, J.A.: A framework for information systems architecture. *IBM Systems Journal* 26, 276–292 (1987)
14. Prinz, W., Pennelli, P.: Relevance of the X.500 directory to CSCW applications: directory support for computer based group communication. *Studies in computer supported cooperative work*, pp. 267–283. Holland Publishing Co., Amsterdam (1990)
15. Ellis, C., Wainer, J.: A conceptual model of groupware. In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pp. 79–88 (1994)
16. De Farias, C.R.G., Pires, L.F., van Sinderen, M.: A conceptual model for the development of CSCW systems. In: *Proceedings of 5th International Conference on the Design of Cooperative Systems COOP*, pp. 189–204 (2000)
17. Jeners, N., Prinz, W.: From Groupware to Social Media - A Comparison of Conceptual Models. In: *Camarinha-Matos, L.M., Xu, L., Afsarmanesh, H. (eds.) PRO-VE 2012. IFIP AICT*, vol. 380, pp. 416–423. Springer, Heidelberg (2012)
18. Peristeras, V., Martínez-Carreras, M.A., Gómez-Skarmeta, A.F., Prinz, W., Nasirifard, P.: Towards a Reference Architecture for Collaborative Work Environments. *International Journal of e-Collaboration* 6, 14–32 (2010)
19. Zwegers, A., Tolle, M., Vesterager, J.: VERAM: virtual enterprise reference architecture and methodology. In: *VTT Symposium*, pp. 17–38 (2003)
20. Camarinha-Matos, L., Afsarmanesh, H.: Towards a reference model for collaborative networked organizations. In: *Shen, W. (ed.) Information Technology for Balanced Manufacturing Systems. IFIP*, vol. 220, pp. 193–202. Springer, Heidelberg (2006)
21. Dix, A.: *Human-computer interaction*. Pearson/Prentice-Hall, Harlow (2003)
22. Norman, D.A.: *The design of everyday things*. Basic Books, New York (2002)
23. Katzy, B., Zhang, C., Löh, H.: Reference Models for Virtual Organisations. In: *Camarinha-Matos, L.M., Afsarmanesh, H., Ollus, M. (eds.) Virtual Organizations*, pp. 45–58. Kluwer Academic Publishers, Boston (2005)