In: Nolte, A., Prilla, M., Rittgen, P. and Oppl, S.: Proceedings of the International Workshop
on Models and their Role in Collaboration at the ECSCW 2013 (MoRoCo 2013)

Cooperation on Models and Models for Cooperation

Tom Gross, Christoph Beckmann
Human-Computer Interaction Group, University of Bamberg, Germany
(<fistname>.<lastname>(at)uni-bamberg.de)

Abstract. In this paper we would like to propose a Janus head perspective on cooperation and models: on the one hand cooperation on models is a very important type of activity for groups who want to create shared models that are accepted by the group members; on the other hand models for cooperation are an essential basis to develop user-centred cooperative systems.
Keywords. Cooperation on models; models for cooperation; patterns.

1 Introduction

The organisers of this workshop on 'MoRoCo – Models and their Role in Collaboration' at the European Conference on Computer-Supported Cooperative Work - ECSCW 2013, point out in their call for papers that 'using visual representations for work or business processes can be considered a common practice in modern organisations. These models serve a large variety of different purposes such as documentation of current practices, or informing and planning change or software development.' (Nolte *et al*. 2013). Indeed, models play an important role in Computer-Supported Cooperative Work (CSCW) as shared artefacts in teams that are conceived, developed, and maintained by the teams.

Besides cooperation on models, models that structure the cooperation process are an essential part of cooperation technology. Developing software that supports teams cooperating—this software is often referred to as groupware—is a challenging task and has been researched for more than two decades (Gross 2013; Marca & Bock 1992). Groupware often has a strong influence on how teams will work together. And, in fact, the effectiveness and efficiency of the teamwork as

well as the satisfaction of the individual team members strongly depend on the quality of the concepts underlying the respective cooperative technology. Schmidt (2011, p. vii) points out: 'the development of computing technologies have from the very beginning been tightly interwoven with the development of cooperative work'. Schmidt (2011, p. vii) continues that: 'our understanding of the coordinative practices, for which these coordination technologies are being developed, is quite deficient, leaving systems designers and software engineers to base their system designs on rudimentary technologies. The result is that these vitally important systems, though technically sound, typically are experienced as cumbersome, inefficient, rigid, crude'.

In the light of this Janus head perspective—that cooperation on models is an important part of CSCW, and that the models underlying the cooperative technology do fundamentally influence its success—this paper looks at the role of models for cooperation that can be used as basic concepts for cooperative technology that in return is used for cooperation on models. In the next section we give a brief overview of the history of models and patterns. We then introduce and suggest as a point of departure and the framework of Erving Goffman (esp. (1959)) who studied social interaction among humans and their use of their technical environment for several decades and derived a framework for social interaction. Finally, we summarise our contribution.

2 Models and Patterns

Models and patterns have a long tradition. They have early been used in architecture, most prominently by Christopher Alexander (1977). Alexander used introduced a pattern language to describe solutions that were repeatedly applied to reoccurring design challenges in the design of buildings.

Later, in Software Engineering design patterns serve a similar purpose—design patterns here have been considered as a successful approach for documenting and reusing knowledge providing a 'way of supporting object-oriented design' (Sommerville 2007, p. 422). Software design patterns basically have the following structure: a pattern name, a description of the problem, a description of the solution, and the consequences of the use of the pattern (Gamma *et al*. 1994).

Design patterns are also used for documenting knowledge and experience with the development of cooperative technology. Schuemmer and Lukosch (2007, p. 22) write that 'developers building groupware applications are challenged with technical problems that are outside the focus of average software developers'. Martin and Sommerville (2004) analysed social interaction and translated their results into the format of design patterns. They (2004, p. 61) point out that 'patterns of cooperative interaction highlight similar findings across studies related to particular socio-technical configurations, and the accompanying activities given those configurations. They start to address the question of how we

generalise from ethnographic studies to provide guidance for system designers and other users' and 'patterns can be of relevance and practical use to researchers and practitioners from technical or social scientific backgrounds who have an interest in social aspects of systems design'.

All these patterns provide valuable input for generating models underlying cooperative technology. And they are interesting artefacts to study when developing tools that aim at supporting teams working on them.

Yet, software design patterns primarily help structuring software, and cooperative design patterns are primarily based on the analysis of existing cooperative systems or on some ethnographical studies. In the next section we introduce Goffman's framework of social interaction, which is based on decades of observations.

3 Goffman's Framework of Social Interaction

Goffman's framework of social interaction is based on decades of observations and study of related work of Goffman and provides a substantial peace of knowledge and insight into the way social interaction among humans works.

Goffman uses the metaphor of a theatre stage and points out that humans in any kind of social interaction do a performance in front of other humans who are listening and watching and interpreting the performance. Goffman writes: 'for the purpose of this report, interaction (that is, face-to-face interaction) may be roughly defined as the reciprocal influence of individuals upon one another's actions when in one another's immediate physical presence' and 'a "performance" may be defined as all the activity of a given participant on a given occasion which serves to influence in any way any of the other participants' (1959, p. 15).

His concepts that are most relevant with respect to modelling social interaction as basis for cooperative technology can be grouped into three categories: primary participants, performance, and secondary participants. Figure 1 depicts these three categories and the concepts they contain respectively.

Primary participants are humans who act according to their social status (i.e., socio-economic standing in the society). They perform a routine (i.e., a 'pre-established pattern of action which is unfolded during a performance' (Goffman 1959, p. 16)). According to Goffman humans have kinds of ideal interactions with each other: the optimistic ideal of full harmony (i.e., being in harmony with oneself and with others), which according to Goffman is hard to achieve; and the pragmatic ideal as a projection that should be in accordance with reality and that others can accept—at least temporarily—without showing deep and inner feelings of the self.

An interaction takes place between at least one performer and one person in the *audience*. The performer defines a situation through a projection of reality as expressions of a character bound to a certain social role in front of the audience.

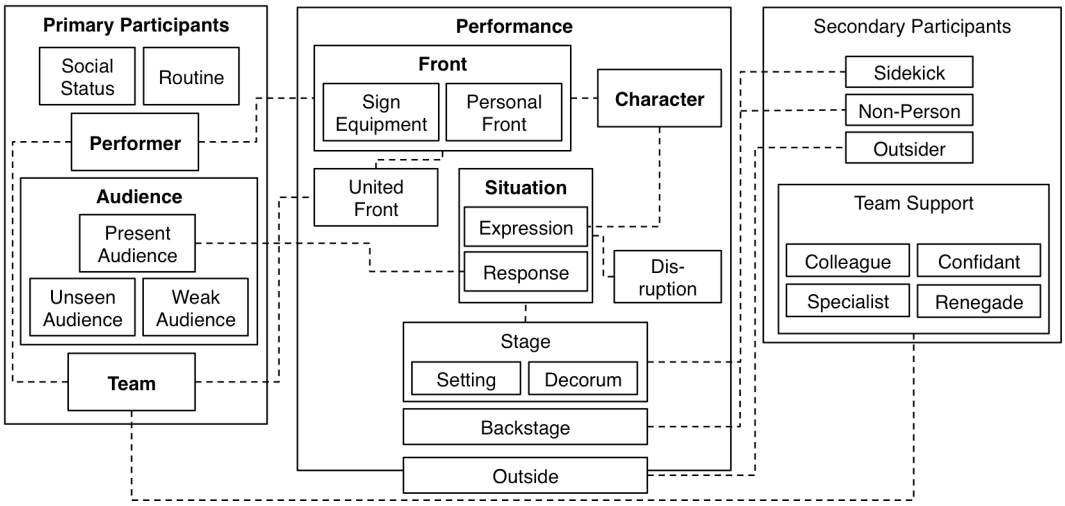

Figure 1. Concepts of social interaction from Goffman's framework of social interaction.

The performer anticipates the audience and continuously adapts the performance accordingly. Audiences can be of three different types: Present audience refers to persons who attend the performance, receive expressions, verify these in accordance to the projected situation and reality, and respond accordingly. Unseen audience are imaginary persons; the performer can use them in order to anticipate a performance. Finally, week audience are real persons who are not present at the performance (e.g., other performers giving similar performances).

Multiple performers can act as a *performance team*. The members of a performance team need to fit together as a whole—to either present similar individual performances to amplify a projection, or to present dissimilar performances that complement to a projection.

For a performance, performers prepare a set of *fronts* shown to the audience. Fronts consist of material and immaterial parts. The material part is the sign equipment and are all properties required to give a convincing performance. The immaterial part is the personal front and refers to a performer's types of behaviour such as speech patterns. During interaction performers appear on stage through a character. A character as figure is composed of a 'front', which is adapted to the audience and performance. In a performance team, the team as whole has a united front (e.g., according to a professional status) and each member has a character with an associated front to invoke during staging.

A performance as social interaction is a finite cycle of *expressions* to define a situation and responses to feedback the validity of the expressions. Characters plays routines during performances to convey acceptable and to conceal inacceptable expressions—in a performance team multiple characters will follow this behaviour. Expressions are information that is communicated by a character, which use 'sign-vehicles' (i.e., information carriers). There are wanted

expressions that are acceptable and foster a situation as a valid projection of reality, and unwanted expressions that are inacceptable and inappropriate for a given performance in front of a particular audience. In order to manifest a performance that is coherent, a performer strives to communicate expressions consistently through their characters towards an audience. Thus a performer's character endeavours to conceal unwanted expressions. Responses are feedback from the audience, which continuously verifies the performance according to the defined situation and the overall reality as well as to the front of the character, and responds the result to the performer.

Disruptions can result from wrong or undefined projection—a consequence of a false or doubtful projection of reality based on contradictive expressions or discrediting actions. To prevent accidental disruptions a performer and an audience can agree on: the 'working consensus' as an agreement on the definition of the situation to describe a temporal value system among all participants; 'reciprocity' that means that performers guise their characters to act according to the situation (i.e., provoke neither intentionally nor factually misunderstandings) and that the audience responds to performance according to the situation (i.e., allege neither consciously nor unconsciously false behaviour); and 'interactional modus vivendi' that describes that an individual in the audience only responds to expressions that are important for the individual; the individual in the audience remains silent in things which are only important to others.

Stages provide a setting for the interaction and are embroidered with decorative properties (i.e., decorum). They support performers when fostering a situation. Both performers and the audience have access to the stage. The backstage is a region, which only performers can access to prepare and evaluate their performance. Also team members suspend backstage. The outside region denotes to neither stage nor backstage. Although it will be excluded in a performance, performers will prepare and use a dedicated front for the outside (e.g., the façade of buildings of a company).

We put other participants of Goffman's framework who are of minor important for cooperative technology in the category *secondary participants*. Participants who are involved, but are not participating in a performance are: team support (colleagues who constitute the weak audience, training specialists that build up a desirable performance and service specialists that maintain a performance, confidants that listen to a performer's sins, and renegades that preserve a idealistic moral stand that a performer or team did not kept), and sidekicks that support a single performer during performance, but in a subordinate role. Non-persons are present but neither participate nor are involved in a performance (e.g., servants). Outsiders are neither performers nor audience and have little or no knowledge of the performance. They can access the outside region; however they can invade a performance and cause a collision of performances: the outsider sees a

performance that eventually is reserved for the future when the outsider is part of the audience.

Overall, Goffman's framework provides an inspiring point of departure when conceiving of basic concepts for a model of cooperation. These concepts can be brought together in a shared model that can then—in a cooperative endeavour—be worked on in a group. The group can work on a model for any domain or business, but it can also work on a model that represents its own structure and roles of actors and ways of interaction among actors and with third parties.

4 Informing the Design of Modelling

The framework of social interaction of Goffman provides multifarious insights that have the potential to positively influence cooperation on models as well as models for cooperation.

Cooperation on models—based on the concepts above—can be characterised as follows. During the cooperation process there are typically active group members and passive group members. A group members' expressions in terms of activities can include oral or written communication, new additions to models, changes of their own parts of a model, changes of parts of the model that have been created by others, and so forth. Passive group members might watch the active person and respond (e.g., confirm that changes to their parts of the model are welcome). On the other hand the active group members might have sophisticated routines that allow them not only to concentrate on their own communication and activities, but also on the others' reactions. Active members can tightly cooperate with other active members in performance teams. The team support might include lab administrators who are responsible for maintain the distributed modelling software and hardware. Researchers have only very recently started looking at these subtleties of users' performances and others responses to them. For instance, Birnholtz et al. presented a study of collaborative writing and point out that: 'people are also concerned about how their behaviours—and they themselves—will be perceived by others' (2013, p. 961). Despite the fact that this study was on collaborative writing and not editing models, it showed interesting evidence that active users in team do care about other users responses to their performance.

Models for cooperation should use Goffman's notions as input for entities. According to Goffman several roles need to be considered by modellers of cooperative processes: performers who actively communicate and change artefacts, performance teams which consist of multiple performers, as well as audiences which can be present and visible to the active performer, unseen and weak audiences which are absent yet important. Furthermore, models for cooperation might foresee secondary participants such as team support or outsiders. In early cooperative systems and early research (cf. e.g., Rodden 1991

for an overview) the notion of a role was clear-cut to and distinct. For instance, a chair-person has specific rights and duties, and a participant has others. More recently—and in accordance with Goffman—roles have been seen as emerging and evolving over time (Finholt *et al*. 2012). Schmidt (2011, p. 31) writes: 'the apparent stability of organizational roles and patterns of communication is a superficial hide … Cooperative work arrangements should rather be conceived as emerging formations that change dynamically in accordance with the requirements of the situation, and cooperative work involves, inescapably, the vicissitudes of distributed decision making. These characteristics have important implications for CSCW systems design'.

As these short examples show, it is important for system designers with respect to cooperation on models and models for cooperation, to find a balance between having a structured, effective, and efficient process and providing lightweight adequate adaptability, flexibility, and spontaneity (Gross & Marquardt 2010; Schirmer & Gross 2011). This has been pointed out very early in the CSCW literature (esp. Bannon & Schmidt 1989), but neglected by some system designers.

5 Conclusions

This introduction of key concepts from Goffman's framework of social interaction is only a starting point towards a more comprehensive discussion of key concepts—in the sense of reoccurring design patterns—of models for cooperation underlying cooperative technology. Conversely, since these key concepts and their mutual relationships can evolve into complex models it would be great to have approaches and tools to cooperatively work on them. Goffman's framework is just one part of the overall picture; other researchers have been using other frameworks, most prominently activity theory (Kaptelinin & Nardi 1997; Nardi 1996) or distributed cognition (Hutchins 1995) (Perry 2003).

In this workshop I would like to share thoughts on how cooperation on models actually works in practice and how tools supporting this type of cooperation can be conceived, while at the same time—from a Janus head perspective—looking at the structure of this cooperation process on models and taking it as the shared artefact that the team is actually working on.

References

Alexander, C., Ishikawa, S. and Silverstein, M. (1977). *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, Oxford, UK.

Bannon, L.J. and Schmidt, K. (1989). CSCW: Four Characters in Search of a Context. In *Proceedings of the First European Conference on Computer-Supported Cooperative Work - ECSCW'89* (Sept. 13-15, Gatwick, UK). Elsevier, Dordrecht. pp. 358-372.

Birnholtz, J.P., Steinhardt, S.B. and Pavese, A. (2013). Write Here, Write Now! An Experimental Study of Group Maintenance in Collaborative Writing. In *Proceedings of the Conference on Human Factors in Computing Systems - CHI 2013* (Apr. 27-May 2, Paris, France). ACM, N.Y. pp. 961-970.

Finholt, T.A., Tellioglu, H., Inkpen, K.M. and Gross, T., eds. (2012). *Proceedings of the 2012 International ACM Conference on Supporting Group Work - Group 2012*. ACM, N.Y.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA.

Goffman, E. (1959). *The Presentation of Self in Everyday Life*. Doubleday Anchor Books, N.Y.

Gross, T. (2013). Supporting Effortless Coordination: 25 Years of Awareness Research. *Computer Supported Cooperative Work: The Journal of Collaborative Computing* 22, 4-6.

Gross, T. and Marquardt, N. (Sept. 2010). Creating, Editing, and Sharing Complex Ubiquitous Computing Environment Configurations with CollaborationBus. *Scalable Computing: Practice and Experience - Scientific International Journal for Parallel and Distributed Computing (SCPE)* 11, 3. pp. 289-303.

Hutchins, E., ed. (1995). *Cognition in the Wild*. MIT Press, Cambridge, MA.

Kaptelinin, V. and Nardi, B.A. *Activity Theory: Basic Concepts and Applications*. Presented at Tutorial #11 at the Conference on Human Factors in Computing Systems - CHI'97 (Mar. 22-27, Atlanta, GA). 1997.

Marca, D. and Bock, G., eds. (1992). *Groupware: Software for Computer-Supported Cooperative Work*. IEEE Computer Society Press, Los Alamitos.

Martin, D. and Sommerville, I. (Mar. 2004). Patterns of Cooperative Interaction: Linking Ethnomethodology and Design. *ACM Transactions on Computer-Human Interaction* 11, 1. pp. 59-89.

Nardi, B.A. (1996). *Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press, Cambridge, MA.

Nolte, A., Prilla, M., Rittgen, P. and Oppl, S. *Call for Papers: Workshop "MoRoCo - Models and their Role in Collaboration" at ECSCW 2013*. http://moroco2013.files.wordpress.com/2013/05/moroco-2013-call.pdf, 2013. (Accessed 7/5/2013).

Perry, M. (2003). *Distributed Cognition*. In Carroll, J.M., ed. HCI Models, Theories, and Frameworks - Towards a Multidisciplinary Science. Morgan Kaufmann Publishers, San Francisco, CA. pp. 193-223.

Rodden, T. (1991). A Survey of CSCW Systems. *Interacting with Computers* 3, 3. pp. 319-353.

Schirmer, M. and Gross, T. (Oct.-Dec. 2011). Lightweight Editing of Distributed Ubiquitous Environments - The CollaborationBus Aqua Editor. *International Journal of Distributed Systems and Technologies (IJDST)* 2, 4. pp. 57-73.

Schmidt, K. (2011). *Cooperative Work and Coordinative Practices - Contributions to the Conceptual Foundations of Computer-Supported Cooperative Work (CSCW)*. Springer-Verlag, Heidelberg.

Schuemmer, T. and Lukosch, S. (2007). *Patterns for Computer-Mediated Interaction*. Wiley, N.Y.

Sommerville, I. (2007). *Software Engineering 8*. Pearson Education Limited, Harlow, England.