Shifts in Significance: How Group Dynamics Improves Group Awareness

Christoph Oemig, Tom Gross

Faculty of Media, Bauhaus-University Weimar, Germany

Abstract

Appropriate awareness support has been an issue in the area of Computer-Supported Cooperative Work (CSCW) for quite a while. Yet, many challenges like the dual trade-off between awareness and privacy and awareness and disruption still remain. In this paper we present our human-centred approach on how to utilise group dynamics and team development specific information patterns to control awareness information in ubiquitous sensor-based environments. The consolidated results from our concept, study and prototype demonstrate how an information item's shift in significance can contribute improvements to the user's overall awareness experience.

1 Introduction

Team members need information about each other, about their workspace, and about the overall group development in order to achieve their best results. This knowledge gained from various sources is usually referred to as group awareness (e.g., Begole et al. 1999). In face-to-face situations, the information needed is ready at hand. However, work groups distributed in space and time afford technological awareness support (Dourish & Bellotti 1992). These systems in turn introduce their own challenges, for instance expressed in the dual trade-off between awareness and privacy and awareness and disruption (Hudson & Smith 1996).

The user's awareness experience strongly depends on the kind and quality of information. In this paper we especially address one attribute of information dynamism to improve it: its changing significance to the user over time. As an example, we can observe this change using instant messaging (IM) applications like ICQ (ICQ Inc. 2007): not engaged in a conversation one only sees the other users' online status. But once conversing, information about the others' online status decreases in significance rapidly. Now, signals about the chat partners' typing activity turn out to be more important. Finishing the chat reverses the picture again. In this little example the changed information need is due to shifts in significance of typing and presence information when the situation switched from “trying to make contact” to “being actively engaged in a conversation” and vice versa.

Putting this on a larger scale, emerging sensor-based ubiquitous environments impose an even greater need for this kind of functionality to support awareness efficiently. But how do we know when what information becomes more important than another? In this paper we present our approach on how to utilise shifts in significance combined with the knowledge on group dynamics and its team development models to filter information in ubiquitous sensor-based environments, regarding a user’s and also her team’s current situation. That is, the application adapts itself to a team's current situation in respect to the information it presents to the user. Before we start with the details, we briefly describe the related work surrounding our idea. Then we describe our concept’s corner stones. In the following sections we then present our composite approach (study and prototype) as well as its results. Finally, we offer conclusions and aspects of future work.

2 Related Work

The related work originates mainly from the areas of patterns in collaboration and collaborative information filtering both directly linked to awareness issues. Begole et al. (2002) describe their analysis of location, calendar and email activity to detect rhythms or temporal patterns in user behaviour by observing their daily work. They focus on the detection of recurring periods of inactivity and location transition. Their Awarenex client gathers the information needed. Actograms visualise the resulting patterns of activity. Besides temporal patterns Fisher and Dourish (2004) also analyse social patterns. They visualise these structures and outline typical ones. Further, they explain how structural and temporal patterns may augment awareness with context information derived from them. Opposed to structural and temporal patterns we seek to utilise information item patterns that depend on the team’s developmental progress. These describe stage-specific information requirements.

On the other hand we find related systems for collaborative filtering where more than one user’s information is taken into account for information selection mostly to decrease disruption. As an example, MatchBase (Gross et al. 2006a) consists of MBSens sensor components, MBAct actuator components, and the MBMatch inference engine. These allow to analyse the contactors’ and contactees’ contexts, and then to match these contexts in order to minimise the contactors’ effort and the contactees’ interruption. In its initial approach MatchBase is used to control the submission of email messages. Most recently, FeedMe (Sen et al. 2006), a general alert management system, uses techniques from machine-learning in addition to traditional rule-based alert filtering to infer alert preferences based on user feedback. They utilise a series of naïve Bayes algorithms. The major difference concerning our work is that we especially consider team stage specific information requirements to realise a more human-centred approach on awareness information filtering.

3 Concept

In our concept we seek to exploit an information item's shift in significance to improve the user's awareness experience. In general, shifts may occur in two major variations: Firstly, while switching from uncoupled to coupled interaction and vice versa, as in the introductory ICQ example (we call these *vertical shifts*). Secondly, during the course of longer lasting relationships (we call these *horizontal shifts*). The first ones are rather spontaneous and ephemeral in nature while the second ones usually result from developmental processes (e.g., learning). In this paper we focus on horizontal shifts especially in collaborative settings where we are able to leverage existing knowledge on group dynamics (i.e., team development processes). Here, knowledge is usually transferred into models, which typically split a team's overall development into subdivisions, so called *stages*. As our grounding hypothesis we assume, that the set of information items needed at a certain stage is typical for the stage itself. We define this set of stage-specific information items as *information pattern*. We also assume that a changing information pattern implies a stage change. These changes are usually due to shifts in significance while groups evolve through these stages. In detail, being able to distinguish team stages by information patterns allows doing the following:

- **Stage recognition and indication**: on the basis of typical patterns we are able to recognise a team's current stage. Additionally, we are able to indicate the stage.
- **Awareness information filtering and application adaptation**: having recognised the stage we are able to select appropriate awareness information utilising the predefined need for that particular team stage and adapt the application accordingly by adding and removing information items from the screen.
- **Gap analysis and information recommendation**: we expect each team member to develop slightly differently—that is, a team's development can be seen as aggregation of its members' development. Analysing the gap between a member's individual stage and the team's stage allows offering recommendations about missing awareness information to minimise the gap. Further, it allows demanding missing information from the user in order to increase the team's overall efficiency. Indicators can be used to visualise the gap thus offering additional awareness information.

The concept's central requirement is that stages can be distinguished by their information needs—that is, they have to be distant enough. To achieve this easier we use a fixed set of information items and binary values indicating if they are part of a pattern or not. This imposes another, rather logical requirement: there have to be enough information items in order to be able to offer a sufficient amount of possible combinations to represent all stages of the model. Table 1 shows how information patterns are constructed. It also allows demonstrating the difference between *input information patterns* and *output information patterns*. Input information patterns are used for stage recognition; output information patterns represent the information set eventually sent to the user. Table 1 shows that information item B is present in every stage and thus does not contribute to any stage differentiation. Therefore, item B is not part of the input pattern, but part of the output pattern, since it still is a required information item for all stages.

	Stage x	Stage y	Stage z
Information item A	-	X	-
Information item B	**X**	**X**	**X**
Information item C	X	-	X
Information item D	X	-	-

Table 1: An abstract information pattern example: X means the information item is part of the stage. The grey-shaded area outlines the information pattern for stage x. The input information pattern consists of items A, C, and D, the output information pattern of A, B, C, and D.

Figure 1 finally depicts input and output information patterns in action for stage recognition, information filtering and application adaptation. Another additional step is shown aggregating individual stages into a team stage. The output pattern is selected for the team stage adapting the user's application accordingly.

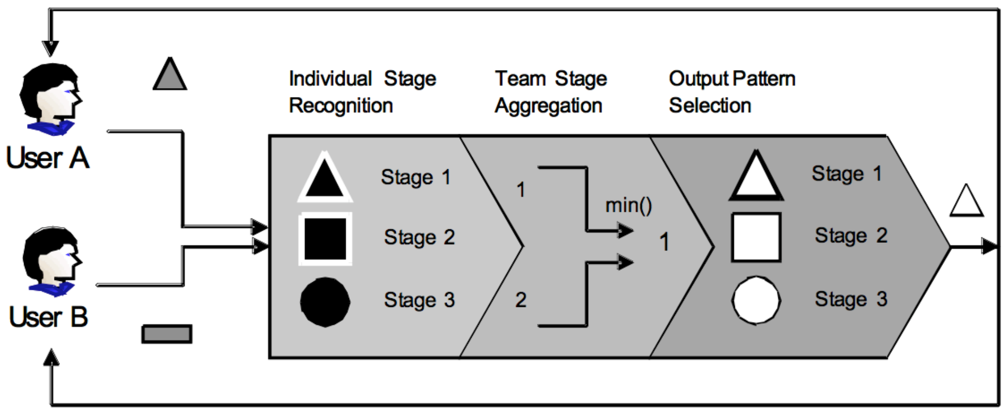

Figure 1: The process of stage recognition, filtering and application adaptation using input and output information patterns. A team's stage is determined by the minimum of its members' stages.

Gap analysis and information recommendation is realised by comparing a user's input information pattern to the currently used output information pattern. Overall, the above can be conceived as framework: it is agnostic of the eventually utilised team development model and its respective information patterns. Our subsequently described approach verifies the above concept and suggests one model-patterns combination.

4 Approach

We decided on a composite approach (see Figure 2) for mainly two reasons: we wanted to verify our hypotheses but also check on the general feasibility of our idea. Having conceptualised a framework allowed us to do so. Thus we achieved results in a shorter period of time as opposed to a sequential approach. After setting up the general concept, we conducted a small study and realised a prototype. For the former we picked two models from the field of group dynamics and checked their application as part of a survey trying to identify their information patterns (as top down approach, see section 4.1). As a second and rather opportunistic part of our approach we chose to implement our framework right away based on our existing knowledge (as bottom up approach, see section 4.2). We had no particular model in mind but tried to explore how far we could get with available information items in a sensor-based environment.

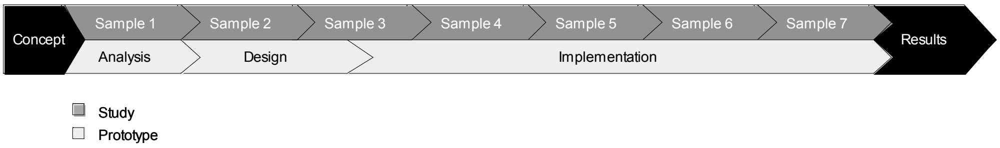

Figure 2: Overview of the composite approach.

4.1 Study

In a small survey we asked four student project teams (up to five students each) to regularly fill out a questionnaire, which was to determine their current team stage according to two selected models as well as their current information need. Information items (personal information, workspace information, information about a user's availability and location) had to be rated in their importance on a scale from 1 to 10. All teams started and finished their projects at the same time. A sample was taken every two weeks over a 14-week time span yielding seven samples per team member.

4.1.1 Team Development Models

As one of the most famous we selected Tuckman's (1965) developmental stage sequence for small teams known as "Forming, Storming, Norming, Performing" (FSNP) and Drexler et al.'s (1988) Team Performance Model (TPM). Tuckman outlines a team's development as a four-staged process. The "Forming" stage stands for the orientation phase when people are set up as a team for the first time. The "Storming" stage represents intra-group conflicts in response to the new group and task demands. "Norming" outlines the settlement of the aforementioned conflicts. Group cohesiveness develops and the new roles become adopted. In the "Performing" stage structural issues have been resolved and solutions emerge. Tuckman later added the "Adjourning" stage, which represents the group's dissolution (Tuckman & Jensen 1977). The Team Performance Model comes along more fine-grained. It proposes seven stages: In the "Orientation" stage team members find out why they are here and start trying to find a place within the group. During the "Trust Building" stage team members try to find out about others, about their plans and their expectations. Then the team clarifies what the team needs to do in the “Goal Clarification” stage. Once the goals are clear the team needs to determine which way to achieve them in the “Decision Making” stage which is followed by the “Implementation” stage where the key concern is the sequence of work and who does what, when and where. Finally, most of the work gets done in the “High Performance” stage. The “Renewal” stage concludes the team’s development. It is quite obvious that both models mainly differ in their granularity.

4.1.2 Results

The questionnaire-based stage recognition revealed: Two teams went from initial to final stages, one team was performing well from the start (the team members knew each other and the topic beforehand) and one never reached high performance stages which strongly correlated to its final result. These developments directly represent typical scenarios our application has to cope with in the future. The distances (Euclidean distance) between stages determined on the basis of different information item ratings revealed the following: The selected Team Performance Model turned out to be too fine grained to offer distant enough information patterns for its stages using the given information item set (Stage 3, 4 and 5 were hard to distinguish as well as 1 and 2). Tuckman’s developmental sequence was harder to distinguish in the first three stages due to the same reason. Some stages could rather be characterised as milestones (i.e., a point in time) as opposed to longer lasting phases a team passes through. TPM’s “Decision Making” stage is such a milestone candidate, which was even hard to capture by our survey. We found slight transitions rather then clearly distinct stages. However, the questionnaire by itself clearly identified the stages and their changes. Therefore, we concluded that we did not use the proper information items for theses models, which leads to the following three options: first, pick another (third) model and try again. Second, pick different information items and try again, or, third, cluster the stages to make them fit the given set of information items (i.e., to build a third model ourselves). We decided on option three, since we were not aware of an existing third model and were limited to the selected information items having our sensor-based infrastructure in mind. We clustered stages using single linkage clustering which became the basis for our result shown in section 4.2.2.

4.2 Prototype

Our prototype (see Figure 3) was implemented as user interface plugin on top of the instant messaging platform PRIMI (Gross & Oemig 2005). Its goal was to realise our concept’s framework and to gain experience with simple information patterns and stage recognition (application feasibility). Information items and patterns were stored as sensor data within SensBase (Gross et al. 2006b), an implementation of the Sens-ation platform.

4.2.1 Five-Staged Team Development Model

We decided to implement a five-staged prototype:

- Stage 1: Orientation. Project and people are set up. Users and project leaders add initial information.

- Stage 2: Trust building. Additional, more sensitive information is added by users, like a photo according to principles of building “Trust without Touch” (Zheng et al. 2002).
- Stage 3: Work package construction and responsibilities. We integrated a small project management tool for work package planning to derive valuable information about the team’s ongoing work.
- Stage 4: Work. Work packages are finished one by one.
- Stage 5: Adjourning. All working packages are completed.

A user may be a member of multiple teams, each possibly with a different stage. A project team stage is determined by the minimum stage of the individual team members.

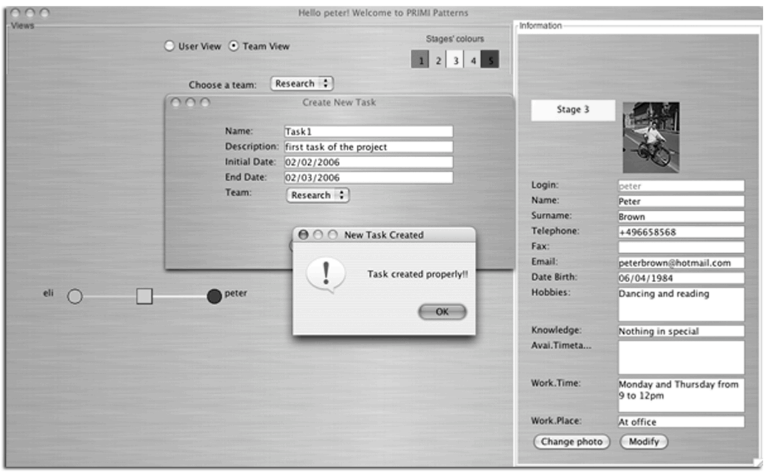

Figure 3: A screenshot of our prototype showing the team view. There is a team of two users, currently in stage 3 creating tasks. Teams are represented by boxes, team members as circles. Colours indicate current individual and team stages. Information items are placed on the right.

4.2.2 Results

When conducting first developer tests we learned from our prototype that its first two phases were hard to distinguish as users added most of the information at once, not separate. We also found one missing stage: the “prior to first contact” stage where users mentioned a different information need than in the first implemented stage. Team development models lack this stage completely while offering a stage for “Renewal” (TPM) or “Adjourning” (FSNP). Our prototype also showed that users could add or remove information items implicitly introducing a great convenience: while he/she has to enter a telephone number directly, workspace activity is captured implicitly by user action. Further, the team stage was determined as the minimum stage of all team members, which, however, turned out to have effects on the recommendation capabilities: a recommendation was based on the difference between the user's current stage and the team stage. It was supposed to be issued when a user falls behind in her development. Yet, using the minimum no one ever receives any recommendations, since even the one with the lowest stage is on team stage level. Even worse, another side effect using the minimum caused advanced team members not to receive the information they needed since it was partially not user stage-specific. Consequently, the personal pattern needed to be applied if it matched to a higher stage than the team pattern. Utilising the average or maximum to determine a team stage does not appear to be that appropriate either, since not all team members have reached that stage. Thus our recommendation mechanism needs to be redesigned. Another issue that came up using the prototype: sometimes not only the team stage decides on a user's information need but also his/her role in the project. Consequently, we need information patterns for regular project member and different role specific pattern sets (e.g., for project leaders). Table 2 shows the consolidated stage model and information patterns gained from part one and two of our approach.

	Prior first contact	**Orientation and Trust building**	**Working/Performing**	**Adjourning**
Nickname	X	X	X	X
Full name	X	X	-	X
Email address	-	X	X	X
Telephone	-	X	X	-
Role	-	X	-	X
Project description	-	X	-	X
Shared workspace information	-	-	X	-
Code repository information	-	-	X	-
Location	-	-	X	-
Availability	-	X	X	-

Table 2: Consolidated stages and their specific information patterns depicting clustered and new stages resulting from our composite approach.

4.3 Discussion

From our composite approach we obtained very valuable insights either part by itself could not have delivered. Our empirical basis is rather small but served its purpose to provide us a direction and a foundation to ask further questions. The information items available in the used ubiquitous sensor-based environment were not able to identify the stages of the two selected group dynamics models, since some of the needed stage information is generally hard to grasp: for instance, how do you capture hidden agendas with physical or software sensors? The gap between the questions of the questionnaire and the questions sensors can answer was too large. Therefore, the eventually used model had to come closer to the available information basis. We did not expect a one-to-one mapping and by our adaptation the model became too simple to be ever used in the context of social psychology. However, it served our purpose to filter awareness information efficiently and thus supported our central hypothesis. Checking on the concept's feasibility also revealed quick results. Since our idea was conceptualised as framework we were able to gain some experience in dealing with many incarnations of the concept's corner stones and captured early feedback by exposing them to users. The testing of slight adaptations and variations became very easy this way.

5 Conclusions and Future Work

In this paper we presented our idea and composite approach on leveraging horizontal shifts in significance and group dynamics to improve group awareness. We demonstrated how to use information patterns to determine the awareness information presented to the user. Previously, a user's awareness focused on what others were doing. It has never been put into perspective to the user's own activities offering true group awareness about the team's state. Yet, we only made one step concerning team stage analysis and its incorporation with awareness information management. Further future work includes the usage of different sets of information items to see if they even improve the system's functionality by better differentiating team stages. The design of transitions between stages—that is, an information pattern's change in the user interface, remains an interesting issue. The same accounts for trying novel indicators for team stage awareness and gap analysis as well as indicators to visualise multiple teams including their stages (e.g., for project managers). Here, one can think of portfolio chart type of indicators showing the "poor dog" and "cash cow" project teams. Another work item would be the integration of further information like work rhythms to better determine a team's stage to eventually offer more fine-grained information. Yet, one of our immediate next steps is to integrate vertical shifts of significance as in our introductory example of ICQ. This offers another dimension of filtering for even further awareness experience improvements.

6 References

Begole, J.; Rosson, M.B.; Shaffer, C.A. (1999): Flexible Collaboration Transparency: Supporting Worker Independence in Replicated Application-Sharing Systems. In: ACM Transactions on Computer-Human Interaction 6, 6 (June 1999). N.Y.: ACM Press. pp. 95-132.

Begole, J.B.; Tang, J.C.; Smith, R.B.; Yankelovich, N. (2002): Work Rhythms Analysing Visualisations of Awareness Histories of Distributed Groups. In: Proceedings of the ACM 2002 Conference on Computer-Supported Cooperative Work - CSCW 2002. N.Y.: ACM Press. pp. 334-343.

Dourish, P.; Bellotti, V. (1992): Awareness and Coordination in Shared Workspaces. In: Proceedings of the Conference on Computer-Supported Cooperative Work – CSCW 1992. N.Y.: ACM Press. pp. 107-114.

Drexler, A. B.; Sibbet, D.; Forrester, R. H. (1988): The Team Performance Model. In: W. B. Reddy & K. Jamison (eds.), Team Building: Blueprints for Productivity and Satisfaction. Alexandria, VA & San Diego: National Institute for Applied Behavioural Science & Pfeiffer. pp. 45-61.

Fisher, D.; Dourish, P. (2004): Social and Temporal Structures in Everyday Collaboration. Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI 2004, N.Y.: ACM Press. pp. 551-558.

Gross, T.; Braun, S.; Krause, S. (2006a): MatchBase: A Development Suite for Efficient Context-Aware Communication. In: Proceedings of the Fourteenth Euromicro Conference on Parallel, Distributed, and Network-Based Processing - PDP 2006. Los Alamitos: IEEE Computer Society Press. pp. 308-315.

Gross, T.; Egla, T.; Marquardt, N. (2006b): Sens-ation: A Service-Oriented Platform for Developing Sensor-Based Infrastructures. International Journal of Internet Protocol Technology (IJIPT) 1, 3 (2006). pp. 159-167.

Gross, T.; Oemig, C. (2005): PRIMI: An Open Platform for the Rapid and Easy Development of Instant Messaging Infrastructures. In: Proceedings of the 31st EUROMICRO Conference on Software Engineering and Advanced Applications - SEAA 2005. Los Alamitos: IEEE Computer Society Press. pp. 460-467.

Hudson, S.E; Smith, I. (1996): Techniques for Addressing Fundamental Privacy and Disruption Trade-offs in Awareness Support Systems. In: Proceedings of the ACM 1996 Conference on Computer-Supported Cooperative Work - CSCW'96 (Nov. 16-20, Boston, MA). N.Y.: ACM Press. pp. 248-257.

ICQ Inc. ICQ.com - Community, People Search, and Messaging Service! http://www.icq.com/, 2007. (Accessed 9/3/2007).

Sen, S.; Geyer, W.; Muller, M.; Moore, M.; Brownholtz, B.; Wilcox, E.; D.R. Millen (2006): FeedMe – A Collaborative Alert Filtering System. In: Proceedings of the ACM 2006 Conference on Computer-Supported Cooperative Work. N.Y.: ACM Press. pp. 89-98.

Tuckman, B. (1965): Developmental Sequence in Small Groups. Psychological Bulletin, 63, pp. 384-399.

Tuckman. B. W.; Jensen, M. A. C. (1977): Stages of Small Groups Development Revisited. Group & Organization Studies, 2 (4), pp. 419-427.

Zheng, J., Veinott, E., Bos, N., Olson, J.S. and Olson, G.M. Trust without Touch: Jumpstarting Long-Distance Trust with Initial Social Activities. In Proceedings of the Conference on Human Factors in Computing Systems - CHI 2002. ACM Press, N.Y., 2002. pp. 141-146.

Acknowledgements

We would like to thank our colleague Tareg Egla and the Cooperative Media Lab (CML) students—especially Roser Agúndez, Yunlu Ai, and Laia Isern—for contributing to the concepts and implementation of this work. We also thank the anonymous reviewers for providing stimulating comments on earlier versions of this paper.

Contact

Prof. Dr. Tom Gross, tom.gross(at)medien.uni-weimar.de
Christoph Oemig, M.Sc., coemig(at)acm.org