Group context-based adaptations for recommendation

Björn Joop
Department of Computer Science and Applied
Cognition Science
Interactive Systems and Interaction Design
University of Duisburg-Essen, Germany
bjoern.joop@uni-due.de

Jürgen Ziegler
Department of Computer Science and Applied
Cognition Science
Interactive Systems and Interaction Design
University of Duisburg-Essen, Germany
juergen.ziegler@uni-due.de

ABSTRACT

In groupware or community based applications the user interface is usually static or tailored to the individual user's needs. Newer developments try to adapt the user interface automatically in regard to user contexts. Even though these techniques are proven useful, there exists no context-adaptive system taking the current context of a group or community in regard. In this paper, we briefly discuss the problems of defining context and present our understanding of context as a subset of the current information state. We provide an exemplary scenario to present different approaches how to compute group contexts based on semantic models and user contexts, and the consequences for the adaptation goals - in the interface or through changes at system functionalities or tools. We additionally discuss the problems occurring at evaluating adaptations and the value of group context for collaborative work.

Author Keywords

context, user context, group context, group context-based adaptation, content recommendation, semantic models.

ACM Classification Keywords

H.3.3 Information Search and Retrieval

INTRODUCTION

Whereas single user adaptations are nowadays well established (cf. [2, 5, 6]), adaptations in regard to group interests are insufficiently researched. Present approaches use context-based adaptations to change user interface elements in regard to contextual information. This includes information from within the system as well as external information such as the physical world.

Even though single-user contexts are being used for adaptations, the usage of group context is not established at all. The notion of shared context has been analyzed to some

extent [11], but the combination into a shared context is still an open issue. Most Computer Supported Collaborative Work (CSCW) applications adapt to the individual user needs or allow manual tailoring. The main purpose of using group context for adaptation purposes is, that additional information in regard to the group are useful for a single user. This mainly focuses on collaborative work or community support. For this goal we understand recommendation as a useful example for content adaptation in user interfaces.

DEFINITIONS OF CONTEXT

The word "context" shows its meaning inherently: *con* (meaning: with) *text*. This definition has its origin in linguistics for describing the surrounding situation allowing an easier interpretation [7, 1]. In computer science the notion of context has played an increasingly important role especially in the area of ubiquitous computing with the aim of developing context-aware systems [2]. Context-aware systems usually use time, location, users and available resources as contextual information representing aspects of the physical world [8].

A general definition was provided by Dey et al [3, 4] defining context: "*context is any information characterizing the situation of an entity*". Another popular definition was given by Winograd [10] defining context as an operational term for characterizing its role in communication. This means something is context because it is used in communication for interpretation and not due to its inherent properties. Winograd further distinguishes between context and setting. In his opinion, the setting includes all fixes information about users nearby, the place an action takes place, etc. On the other hand, context represents all information helping understanding a communication.

Approaches modeling context range from simple key-value models over graphical or hierarchical models up to sophisticated ontology-based context models which support validation and reasoning (cf. [9]).

OUR UNDERSTANDING OF CONTEXT

We agree with Winograd [10], that we need a distinction between context and setting. But in our understanding, we need contextual (meaning: surrounding) information in regard to a focus (center of attention). Context depends on
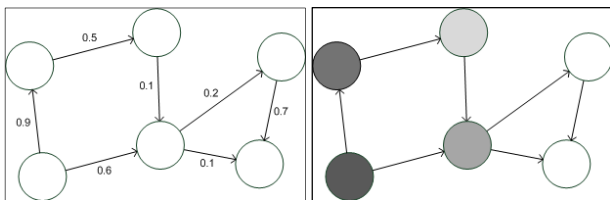
the current person's perspective, known facts and the state of the "real" world and applications. But this also means, that context is ever-changing and dynamic. These changes can be easily captured and interpreted by humans; context aware applications on the other hand have to use sensors to recognize these changes and have to interpret it due to predefined rules.

We see contextualization as a selection process: In a complex situation contextualization mechanisms are used to extract the most relevant or dominant elements in regard to the current center of attention.

FROM STATE TO CONTEXT

In [5] we presented a multi-layered framework for context-based adaptation: We use predefined semantic knowledge to create a directed and weighted graph. Using sensors capturing internal and external information, the graph can be modified to represent the current situation, both from the system and the physical world. We call this graph the current state.

To determine which information from the state is contextual important or not, we use contextualization techniques (e.g. spreading activation or rules) in regard to a given focus (cs. Figure 1). A focus is a non-empty subset of the state representing the application's (and thus the user's) current center of attention.

Figure 1. *Left:* **Directed, weighted graph representing the current state before contextualization.** *Right:* **Same graph after contextualization. The more important an information node is, the darker it is painted.**

The derived context information is a subset of the state graph and can be further filtered and used for adaptation.

SCENARIO

For an exemplary scenario we look at research processes and article creations in a newspaper. Multiple users (authors) research different topics for their current task of writing an article for a predefined topic.
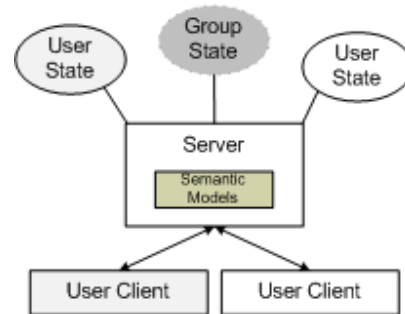
We describe three persons more detailed: Sarah is specialized in politics and writes articles about current developments within the government. Alex is sports journalist and Eve is writing articles describing economy trends.

Even though these three persons work for the same company and for the same newspaper edition, they work on different tasks and topics, thus a system supporting them has to take different requirements into account. The system offers a web-frontend with document management and search functionalities.

At a specific time, Sarah writes an article about the newest president's political scandal. At the same time Alex, sports editor, writes an article about the last European soccer tournament and Eve, the economist expert, researches material about the bankruptcies of global banks.

EXEMPLARY SYSTEM ARCHITECTURE

A centralized server manages users and client applications (e.g. the web-frontend) for collaborative work as well as the different user states. Upon requesting the context for a user or an application, the server returns a list of the most relevant information after contextualization.

Figure 2. Exemplary system architecture for client/server connection. The server holds different user states as well as group based state(s) for computing the context upon request.

Figure 2 shows simplified system architecture. Different user clients can request context information separately from the server. The server manages each user separately in own user state graphs based on the semantic models. Additionally one or several group states are hosted using each user's state as information sources.

User clients can also be represented through session-based web frontends used to display new, for the accessing user contextually useful content (usually in form of links or documents) or any other application type (e.g. text editors, chats, etc.).

ADAPTATION GOALS

In case of a single-user application, it is sufficient to simply use part of the current knowledge and surrounding for the state. For whole user groups, this is not sufficient. Therefore it is necessary to include all information from all users and their particular surroundings to determine a group state, which then can be used to gain the group's context.
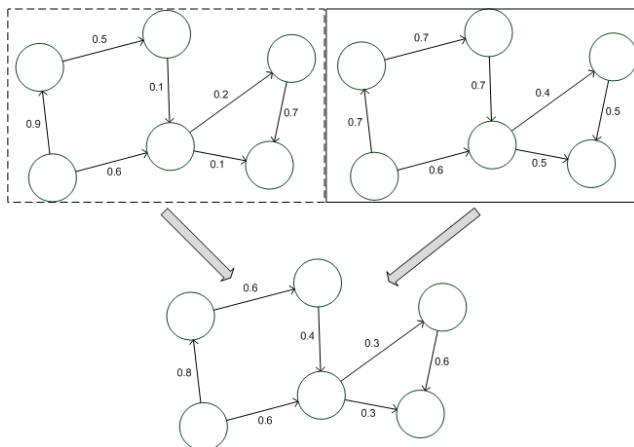
In regard to different adaptation goals, we need to use different approaches to handle group information in one (or several) state(s) for recommendation.

Adaptation goal: main group interests

For displaying the current main interests of the group, e.g. the most dominant tasks, shared interests or documents, one of the easiest ways to create a group state is to integrate each individual state into a single one through adding all nodes and edges, summing up all weights and normalizing them (cf. figure 3). This leads to a group state which represents all interests within the group in regard to their

occurrence. The main disadvantage of this approach is that new information will not be considered as useful context information and thus cannot be recommended. Only information most of the users regard useful will be displayed.

Another way of handling main group interests would be through using sensor mechanisms to analyze each state and then transfer inferred knowledge (edges and relations) to a separate group state. This approach is highly depended on the sensor implementation and must be specialized to the application domain. Unfortunately, the main disadvantage is the inability to easily transfer new, sensors unknown information such as new task-related content.

Figure 3. Combining two different user states from the same semantic model to a group state: adding all nodes and edges into one graph, then summing up and normalizing all weights.
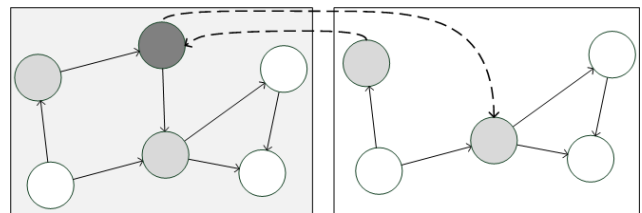
Example:
Even though Eve, Sarah and Alex write on different articles covering diverse topics, they share some task-dependant interests such as the current deadline for this edition, the need of specialized tools for editing the article or images. Thus capturing the current main interests within thins small group we can support them directly through recommending tools or newspaper-dependant information (e.g. deadlines).

Adaptation goal: information transfer
It is often useful to recommend information one user added to his/her interest profile or their newly created documents to other users within the group. Information one user acquired may be highly useful to tasks or process other users are involved in. Therefore it is necessary to transfer information from one user state to another. Additionally, the new information must be semantically fitting integrated so that it can be presented to the users upon request.

One approach is to enrich each user's individual state with additional information representing the group information. This may happen if different users have the same semantic graph structure from the same semantic models, but have totally different information represented in them. This can be realized through dynamically linking nodes from each state to other states (cf. figure 4). This way every existing
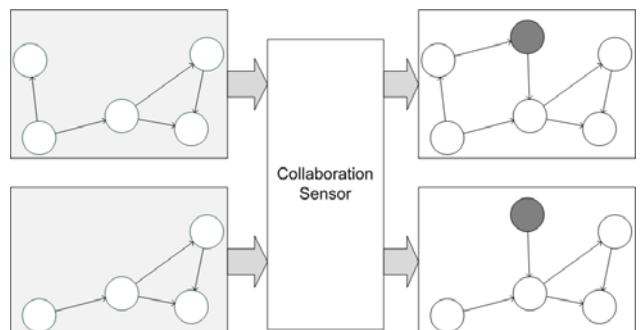
state has to be compared to each other state for missing relations and nodes which then have to be created interlinking the states.

Figure 4. Two user states are dynamically linked. Both states are slightly different even though they base on the semantic models. To enrich the right state with additional information, connections (dotted) to and from "missing" nodes (dark node left) are included.

The main disadvantage lies here in the need to update every state in regard to changes one user made.

Another way to enrich user states with information from other states is to replicate missing information nodes and relations and persistently store them upon need. For this, collaboration sensors have to analyze the existing user states and find or infer information which may be highly interesting for other users. Collaboration sensors have to react in regard to predefined sensing mechanisms or rules (e.g. from data mining, graph merging algorithms, etc.). Then the inferred knowledge can be added to each user's state and be persistently stored. The quality of the inferred information is highly dependent on the quality of the semantic models – thus the information in the state graphs - and on the collaboration sensor implementation.

Figure 5. From different states (left) collaboration sensors infer new knowledge which can then be added to the particular states for later contextualization.

Upon computing the context for the user, we are now able to use contextualization techniques using the newly added information to create context-based recommendations.

Example:
During Alex's research regarding the last European soccer tournament, he added a report from a journalist describing a big party at the soccer stadium during the final game. This party was thrown by a number of highly ranked bank officials. Because of a number of revealing pictures and a detailed description, Alex added some additional
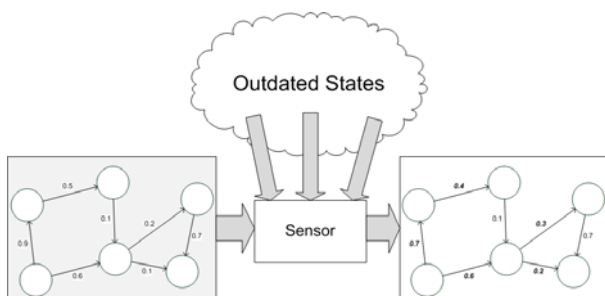
information and stored it in his content management system and thus in his current state.

On the other hand, Eve does research for an article about high ranked bank officials whom misused their status and the bank's expenses for themselves and their friends.

Using collaboration sensors, the adaptive system recognizes similarities between the tasks these two users are currently working on and identifies the differences – in this case, the information added by Alex. Upon enriching the individual user's states before contextualization, this information will be added to Eve's state and semantically fitting included using relations with a high relation strength connecting this new information with her current task. This way, upon looking at her current recommendations, she will be informed that there exists a report and pictures about such an abuse of expenses in the system and she can look at it and use it for her current work.

Adaptation goal: tracing changes over time

Within a group, the quality and the distribution of information can change over time. To trace these changes, e.g. within shared documents, upon solving tasks, etc., it is necessary to create either time dependent information in every state, or to store outdated states for comparison (cf. figure 6).

Figure 6. Modification of a user's state in regard to outdated states. This may lead to increasing or decreasing relation strengths.

In either case, we need specialized sensor metrics to identify and react to changes over time. Before contextualization, sensors have to change the relation strength in regard to the changes over time; e.g. if a task is solved and thus has no longer a high priority, meaning the relation strength to this node is further decreasing, used information and content may already part of the shared knowledge and can thus be used for similar tasks later on. This way either less used information, which possesses high relation strength, can be further distributed as well as old, outdated information can be "forgotten".

Further adaptation possibilities

Content recommendation is not the only possibility to support group activities. Using group context-based models for adaptation, it is possible to retrieve the group's main interests or to transfer knowledge from one user to another without the need for direct interaction. A web frontend is not the one possible way to present recommendations to help group activities. Depending on the modeled knowledge, it is also possible to use recommendations to react to current tasks or interests to change the client applications layout, enable or disable functions or to change the overall to trigger applications for specific purposes.

Example:
Eve and Sarah both need to make an interview in another town at the same day. Even though both do not necessarily recognize the fact that both have this appointment, the system can compute the group's current context and inform both about this fact. Even more, it may change the calendar to inspire them to contact each other – maybe even through big red markings at the appointment. In case both are currently working with the system, it is also possible to offer a direct communication channel (e.g. Skype, chat, etc.) and a reason (the appointment) why to talk to each other.

EVALUATING GROUP CONTEXT BASED ADAPTATION

Evaluating the success of adaptation in regard to group contexts is more complex than in regard to single user context. The reason is, that not a single person has to profit from the adaptation, but whole groups. The user satisfaction is depended to the adaptation purpose and thus the method for computing the group states and group context.

The success for computing the group's main interests, it is necessary to evaluate each user's interests separately and then compute stochastically to find the groups real main interests (e.g. in regard to their occurrence). Upon successful adaptation most users should be satisfied with the changes and recommendations, even though – because of heterogeneous interests within the group – some users will most likely be displeased. To increase satisfaction it will be necessary to combine both single-user and group context-based adaptations.

Regarding information transfer from one user context to another, success could be measured through the number of accesses to the recommended information. Even though group information is being used to enrich a single user context, the benefits will be shown for collaborative scenarios and can be measured using the number of data access or communication attempts, or at comparing the efficiency at solving tasks and the result's quality.

DISCUSSION

Single user context-based adaptations are able to support and inform single user applications through information or automatic functions. Group contexts offer a lot of potential in the area of collaborative work – especially in regard to dynamically changing requirements or workflows. Instead of tailoring a collaborative application towards supporting single tasks or workflows, group context aware applications can adapt to these changes and still maintain eligible support increasing productivity and better information transfer.

As we presented, there exist different ways to approach group states and thus group context. Even though we identified differences for the creation of group states influencing the application's adaptation possibilities, we are sure this list is still not complete. One open issue is the possibility to incrementally create and build a more natural group context in regard to user's action, knowledge and documents. This approach is mostly influenced by people's everyday interactions: upon meeting and interacting with others, we offer ourselves information to them and vice versa. Therefore no automatic mechanisms creating group information will be used, but instead focusing on analyzing collaborative interactions.

Unsolved issues regarding security and privacy policies are also still open. Usually, users are not to be monitored regarding their actions during work. This leads to the need to include different public views to the user's state and knowledge to limit the access from others.

CONCLUSION AND FUTURE WORK

In this paper we briefly presented our understanding of context and different ideas for creating group states as starting points for context-based adaptation for groups. Our intention is to recommend content to users depending not only on the individual user context (or user interests), but for or in regard to whole user groups. In our opinion, group context-based adaptations offer a wide range of possibilities to directly support users through recommending information from other users, in regard to the group's main interests or in regard to changes within the group over time.

We briefly discussed evaluation methods and limits for group context-based adaptation methods regarding the different adaptation goals. For further studies, a concrete implementation and evaluation of the presented ideas and approaches is required and will be presented in upcoming papers. Additionally, it is necessary to investigate how group context information changes over time and how these changes can be made traceable and understandable for users.

REFERENCES

1. Austin, J.L. How to Do Things with Words (Oxford: O.U.P., 1962).

2. Bunt, H. Context and Dialogue Control. *THINK Quarterly*, Vol. 3, 1994.

3. Dey, A.K., Abowd, G.D., Brown, P.J. , Davies, N.D. , Smith, M. and Steggles, P. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, 1999, pp. 304-307.

4. Dey, A.K. Understanding and using Context. *Personal Ubiquitous Computing Nr. 1 Vol. 5* (2001), pp. 4-7.

5. Haake, J., Hussein, T., Joop, B., Lukosch, S., Veiel, D., Ziegler, J. Context Modeling for Adaptive Collaboration. *Technical Report 2009-02*, University of Duisburg-Essen, 2009, (http://duepublico.uni-duisburg-essen.de/servlets/DocumentServlet?id=20510)

6. Hussein, T., Linder, T., Gaulke, W., Ziegler, J. (2009), Bergmann, L. (Hrsg.). Context-aware Recommendations on Rails. *Proceedings of the 2009 Workshop on Context-aware Recommender Systems (CARS 2009)*

7. Malinowski, B. The Problem of Meaning in primitive Languages. *The Meaning of Meaning*, 1923, pp. 146-152.

8. Schilit, B., Adams, N. and Want, R. Context-Aware Computing Applications. *(IEEE) Workshop on Mobile Computing Systems and Applications*, 1994.

9. Strang, T. and Linnhoff-Popoen, C. A Context Modeling Survey, Workshop on Advances Context Modeling Reasoning and Management. *The Sixth International Conference on Ubiquitous Computing*, 2004.

10. Winograd, T. Architectures for Context. *Human-Computer Interaction, Vol. 16* (1991), pp. 401-419.

11. Zimmermann, A., Lorenz, A. and Oppermann, R. *An Operational Definition of Context*. Modeling and Using Context (2007), pp. 558-571.