CONTEXT MODELLING FOR INFORMATION RETRIEVAL—REQUIREMENTS AND APPROACHES

Tom Gross
Fraunhofer Institute for Applied Information Technology—FIT
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
tom.gross@fit.fraunhofer.de

Roland Klemke
bureau42 GmbH
Rathausallee 10, D-53757 Sankt Augustin, Germany
roland.klemke@bureau42.de

ABSTRACT

With Internet technology and other information systems that are available today users can often easily find the information needed. However, there are still several situations, in which users do not succeed to get the information required. In this paper we suggest context models to help users finding information in adequate quantity and quality. In general, context models analyse the current situation a user is in, compare it to the available information, and provide the user with information that is of most value in this situation. In this paper we discuss requirements for context modelling, present two approaches we have applied for our own applications—one with a highly sophisticated single-user model and one with a light-weight cooperative model—and discuss how these two approaches can be applied to information retrieval.

KEYWORDS

WWW, Information Seeking, Information Retrieval, Context, Context Modelling.

1. INTRODUCTION

With Internet technology and other information systems that are available today users can often easily find the information needed. Particularly when the user knows how to formulate the question in a format that is understandable for the system. In other situations users may have difficulties in finding the information needed. For instance, this is the case when users are in an anomalous state of knowledge, in which they know that they need more information, but are not yet in the position to properly formulate the query (Belkin 1980). In this paper, we suggest context models to help users in this latter type of situations: context models analyse the current situation a user is in, compare it to the available information, and provide the user with information that is of most value in this situation. Cooperative context models, furthermore, depart from the idea of supporting users in groups. They analyse the current situation of different users and then stimulate contacts, information exchange, and mutual help among users who are in similar situations.

In the following sections we will discuss requirements for modelling information seekers' contexts. The context models are then used to identify the information needs in correspondence to the information seeker's current situation. We present two different approaches for context modelling—one with a highly sophisticated single-user model and one with a lightweight cooperative model—and discuss how these two approaches can be applied to information retrieval.

2. CONTEXT MODELLING REQUIREMENTS

The modelling and use of contexts for the adaptation of the information to the user's current situation and current information needs has a great potential for improving services for the users, but at the same time can cause several new challenges concerning the creation and the maintenance of the context model. Therefore, it is important that context models meet the following requirements.

Requirement 1: A context-modelling framework has to identify all relevant contextual dimensions. A context may be defined as 'any information that can be used to characterise the situation of an entity; [where] an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves' (Dey & Abowd 1999). We define contextual dimensions as relevant if they:

- successfully characterise a given situation and distinguish it from others
- allow efficient storage
- allow the definition of a set or range of possible values of sufficient accuracy
- allow the measurement of the similarity for each pair of given values
- allow the use of indexing strategies to simplify retrieval

In (Agostini et al. 1996) organisational context is defined along three dimensions (organisation, process, space). These dimensions are hierarchically refined in (Lenat 1998) resulting in twelve dimensions for describing contexts (absolute time, type of time, absolute place, type of place, culture, sophistication/security, topic, granularity, modality/disposition/epistemology, argument-preference, justification, domain assumptions). Following this definition of relevance of a contextual dimension, 'outside temperature' is probably not a relevant dimension of an organisational context model while 'time' and 'process' are.

Approaches in the information retrieval community that try to make use of context knowledge in order to improve retrieval results typically use long-term user interest profiles (created explicitly by the user) or analyse the user's retrieval history. In general, they only look at the consumption side of the information retrieval process to make use of context. They ignore the production side. For general-purpose information retrieval systems contextualisation of information at production time would not be appropriate as producers and consumers of information are distinct groups—making their contexts incomparable. This situation changes when we look at virtual communities. Virtual communities use information systems that contain information produced and consumed by the same group of people—the members of the community. Thus they share a limited number of possible contexts, which should be used. This leads us to the next requirement:

Requirement 2: In a context-enhanced community information system, context knowledge has to be associated to information when it is produced and has to be used during information retrieval. This requirement is based on the idea that knowledge about the current context of a user may be used to enhance any information created, modified, or published by the user, and to offer information created, modified, or published in contexts similar to the current user's context.

Some approaches associate information with organisational models, software engineering process models or general workflow process models (e.g., (Prinz 1993; Wargitsch et al. 1998)). These approaches have shown that information may be retrieved in a context-based way—that is, a user who is in a certain context can view, browse or retrieve the corresponding contextualised information. This leads to:

Requirement 3: Context information has to be used as explicit query to the community information system. If the contextual information of the user is used as an explicit query to the context-enhanced community information system, we can apply similarity measures to the context models. This is a prerequisite to retrieve information from similar yet different contexts.

Another advantage of having explicit context models is that additional retrieval strategies can be provided that are based on combinations of content-based and context-based queries (e.g., match query and similar context, match query and complementary context, match query only, or match context only). This leads us to:

Requirement 4: Context-based and content-based retrieval of information should be possible both independently of each other and in combination. While a priori the modelling of contexts and the corresponding implementation mechanisms to exploit context information in an information system are appropriate for a domain with clearly structured work processes that remain stable over a long period of time, more flexible approaches are needed for other domains.

Requirement 5: A context should basically be recognised automatically; however, the system should allow users to explicitly provide context information at the same time (thus providing additional context information that cannot be detected automatically).

Requirement 6: The system has to consider all partially matching contexts and merge them into a coherent presentation of the information. While the user is associated with a unique context at any time, this context will not match exactly with the stored contexts. Instead, a set of partially matching contexts has to be considered.

Requirement 7: The context-modelling framework has to allow the dynamic ranking of important contextual dimensions used to perform the similarity match. In each information retrieval situation the individual contextual dimensions are of specific relevance to the user. For instance, a user waiting for a specific email wants to be notified immediately, whereas a user urgently finishing a paper does not want to be disturbed. In this example, the process dimension (waiting for an email vs. finishing a paper) gets a very high priority, whereas other dimensions such as location are less important. User control is a very important aspect in many systems that perform event-based automatic user notification. It is often critical to the success of the system. This leads us to the following requirement:

Requirement 8: The user notification with relevant events has to consider user preferences (like notification frequency, notification channel).

Requirement 9: The modelling effort for modelling and maintaining context models should clearly pay off in terms of improved access to information and increased working efficiency. A context modelling system—as a benefit—should improve the individual's awareness of relevant events. The cost is directly related to the modelling effort put into the system. Although it is often difficult to analyse the costs and benefits in advance, this requirement is important.

Requirement 10: The time spent on recognising the current context and on retrieving information relevant to this context has to be reasonably small. Finally, one of the goals of a context modelling approach is to give context-related relevant information to the user while she is in that context. This means, that the recognition of the current user context and the retrieval of information relevant to that context has to be done in reasonable time.

In order to find out which modelling effort is appropriate in which situation, we will compare two modelling approaches we currently perform in two different projects: SAiMotion and TOWER.

3. SAIMOTION

Continuous mobile information and communication systems require innovative user interfaces and context adapted modelling attempts for the selection and presentation of information. The Situation Awareness in Motion (SAiMotion) project aims at developing situation-driven information and interaction on mobile devices.

For situation-adapted information presentation and interaction the development of a comprehensive context model is vital. It determines the individual information needs on the basis of location, environment, user, task and activity features. Beyond approaches of location-aware computing where primarily the location of the user is been considered SAiMotion aims at using all relevant situative parameters for proactive information supply and user interaction. The main purpose of context modelling in SAiMotion is the presentation of information and services adapted to the user's context. As we focus on mobile scenarios with handheld computers, there is a trade-off between the amount of information potentially relevant to the user and the amount of information displayable on the screen. Therefore, we use a context-modelling framework to efficiently reduce the amount of presented information. In SAiMotion we use techniques from knowledge engineering to design context ontologies that offer the required modelling precision.

Another important aspect of the SAiMotion scenario is to look at both sides of an information system: the supply of information by a provider and the information consumption by an end user. Figure 1 shows the overall information flow within SAiMotion.

When information is submitted to the information system, its use by an end user has to be anticipated in terms of the context in which it will be useful. Two strategies are possible here: (1) the context of the information provider is automatically associated to the submitted information, implicitly assuming that the context of use is similar to the context of submission. (2) The provider explicitly states values for all relevant contextual dimensions and thus explicitly tries to anticipate contexts of use.

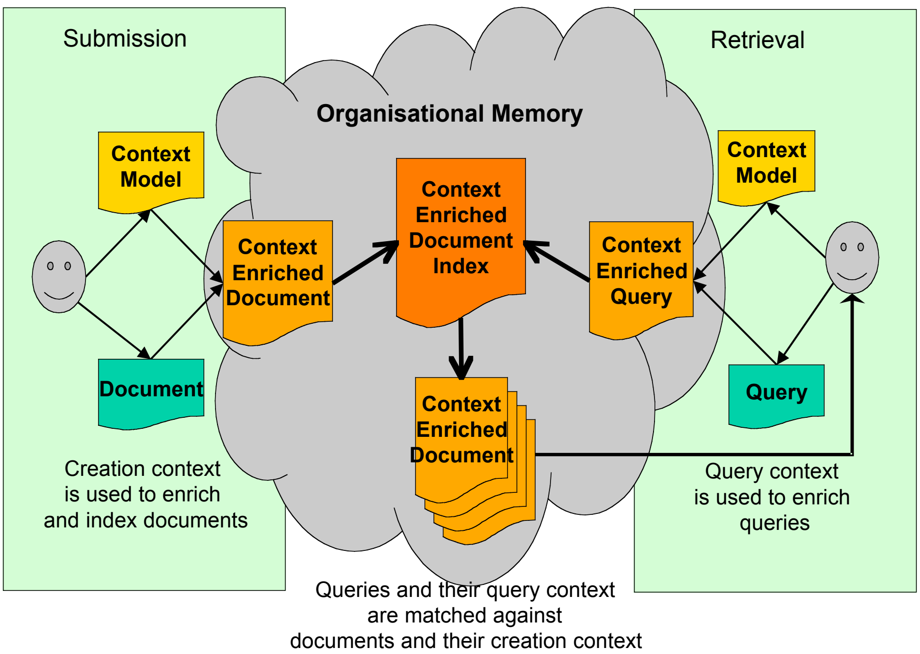

Figure 1. Context enhanced information flow in SAiMotion.

Based on requirement 1 and our definition of relevance of a context dimension we can identify basic dimensions of an organisational context (see Figure 2 and (Klemke 2000) for the results of a literature study about important organisational context dimensions):

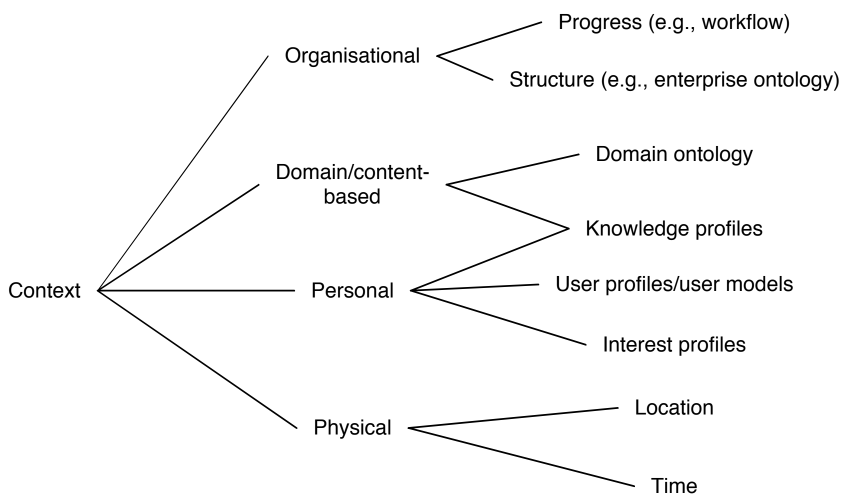

Figure 2. Context typology.

- A *person* is uniquely identified by an ID or a name. A person's context is further characterised by her position within the organisation, her roles, her skills, her interests and experience.
- An *activity* is defined by the task a person has to fulfil (e.g., embedded in a process), by the tools used, files opened, and so forth.

- A point in *time* may be described as absolute time. A further characteristic important for the contextual description of time is the type of time (e.g., something happened on a Monday morning).
- A *location* of a person is not only characterised by its co-ordinates (absolute location) but also by further characteristics such as name (e.g., room number) and function (e.g., office vs. meeting room).

A simple, formal definition of a context model that is suitable for organisations is: `Context = (Person, Activity, Time, Location).` Through ontological refinement and association these dimensions cover all identified contextual aspects from the context typology shown in Figure 1. Each of the attributes that further define the basic context dimensions can either be represented by primitive values (like a timestamp, an ID, or a name) or be complex values (e.g., a categorisation hierarchy to classify organisational roles or interests).

In order to model contexts flexibly and comprehensively, we use an ontological modelling technique developed in previous projects (see (Jarke et al. 2001)). It allows defining basic concepts, features, categories, and entities in order to achieve effective information filtering results according to adaptive personal interest models. We extended the domain-modelling framework with spatial, personal, task, and temporal models (see Figure 3).

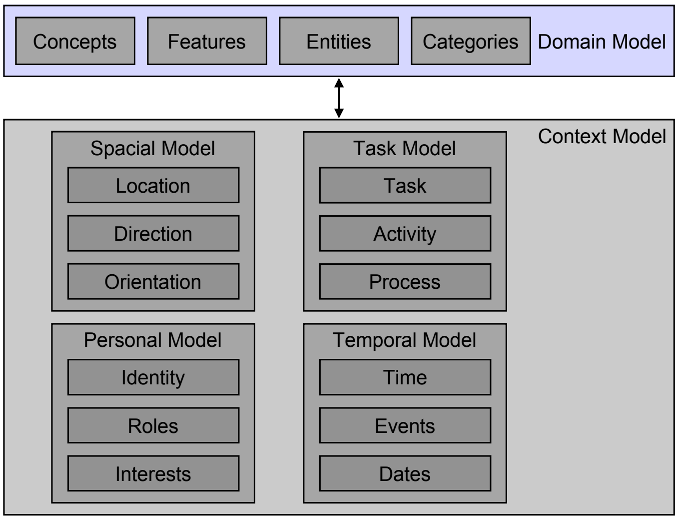

Figure 3. Context-modelling techniques.

Using this structure, we can effectively model complex contexts on a fine level of granularity. We use the same context representation to enrich the contents contained within the SAiMotion system and to represent the user's current situation. The definition of distance measures on instances of context models allows to select the most appropriate information for the user's current context and to deliver it accordingly. However, standard static distance measures are inappropriate here: the distance of two contexts itself is context dependent. We account for this fact using a simple heuristic: the contextual dimension changed most recently is ranked most important in the assessment of the distance of the user's context and the context of information to be evaluated. This emphasizes the fact that the user's focus of attention is usually oriented towards the recent changes.

4. THEATRE OF WORK ENABLING RELATIONSHIPS—TOWER

In cooperative settings the motivation for supporting contexts is slightly different. Whereas in SaiMotion contexts are used to improve the information supply for the single user, in TOWER they are used to provide geographically distributed group members with a common frame for orientation. This common frame is also known as common ground (Clark & Brennan 1991). In the CSCW literature the pervasive knowledge of who is around, what these other users are doing, how available they are, what they are doing with electronic artefacts, and so forth is often called awareness (or *group* awareness (Begole et al. 1999; Gross 2001) or *workspace* awareness (Gutwin et al. 1996)). A common frame for orientation in the search process stimulates spontaneous contacts, serendipitous encounters, and information exchange among information seekers.

TOWER is an environment that aims at providing these types of awareness information. Activities of the users are captured with various sensors in the electronic and in the physical environment. Sensors in the electronic world can detect information such as whether a user is logged in, what applications the user is currently using, which documents the user has currently open. Sensors in the physical world such as audio sensors, video sensors, infrared gates capture information such as if a user is in her office, how much activity takes place in a shared coffeeroom. This information is then presented in the electronic environment and in the physical environment. In the electronic environments awareness indicators include a 3D multi-user environment displaying users as avatars and documents as buildings; tickertapes that run over the computer screen, and pop-up windows. In the physical world indicators such as a fish tank releasing bubbles, a fan blowing air, robots performing gestures, and ambient sounds are available (Gross 2002).

On a whole the TOWER system consists of several components (cf. Figure 4): *sensors* capturing and recognising user activities; an *Internet-based event and notification infrastructure* storing, administrating, and distributing the captured events; a *space module* dynamically creating and updating a 3D multi-user environment that represents the information and artefacts of the group; a *symbolic acting module* creating and animating avatars of the users in the 3D multi-user environment according to their respective actions; a *docu drama* allowing the replay of scenes of the 3D multi-user environment; and various *ambient interfaces* presenting information in the whole physical environment of the users.

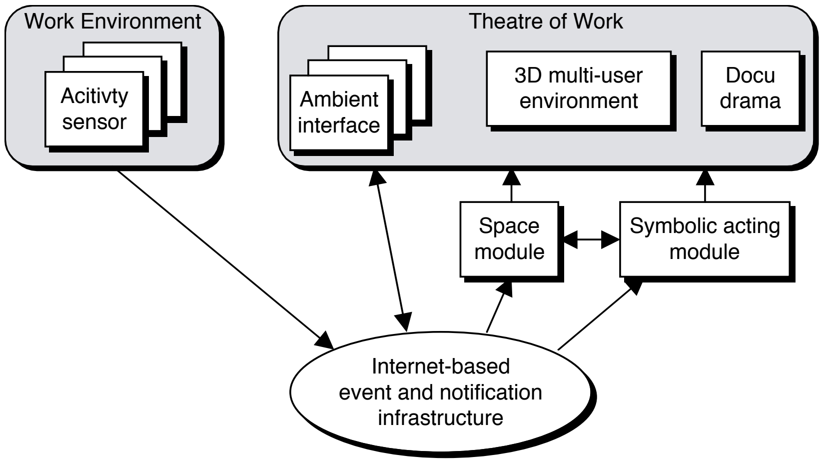

Figure 4. TOWER architecture.

In the TOWER environment contexts are implemented as an extension of the event and notification infrastructure (ENI). Events are the basis for the processing of the awareness information and the context information in TOWER. The events are produced by sensors, which are associated with actors, shared material, or any other artefact in the electronic and physical environment that might be interesting for the individual user or the social context. Events are described as strings of attribute-value pairs. For instance, `producer=tom.gross&artefact=Deliverable7.1`. The sensors send the events they capture to the ENI server.

The ENI server stores and administrates the events. The context module and the situation module of the ENI server are responsible for the context processing. The context module analyses the attributes of incoming events and compares these attributes with the context descriptions in the context database. If all or some attributes match, the context module attaches a context attribute to the incoming event (e.g., `event-context=ProjectX`). On the other side the situation module analyses the attributes of the events a user produces through her specific behaviour and tries to reason about the current work context of the respective user. The system can then compare the user's current work context with the incoming events' context of origin and provide the user with information that is important in her current situation—that is, the ENI server sends the respective events to the users' indicators.

Both, the descriptions of the contexts of origin in the context database and the descriptions of the current work contexts in the situation module are represented as attribute-value pairs. Having a syntax analogous to the individual events makes the comparison easy. Table 1 shows the attributes of the context descriptions.

Table 1. Attributes of awareness contexts.

Attribute	**Description**
`context-name`	Name of the context
`context-admin`	Human or non-human actor who created the context
`context-member`	Human members of a context
`context-location`	Physical locations related to a context
`context-artefact`	Artefacts of a context
`context-app`	Applications related to a context
`context-event`	Events relevant to a context
`context-acl`	Access control list of a context
`context-env`	Related contexts

These attributes are used to describe awareness contexts. For instance, an awareness context could be defined for a project and would then contain the project's name, the administrator, who creates and maintains the awareness context; the project's members, locations, artefacts, applications, event types such as read, write, delete, and the access control list that contains the access rights to information related to the project as well as the relations to other awareness contexts. So, it is the responsibility of the person producing the respective context description to make sure that the above requirements are met (esp. requirement 1).

5. COMPARISON

In this section we will check if and in how far the two approaches in SaiMotion and in TOWER meet the requirements from section 2, and we will compare the two approaches with each other.

5.1 Comparison with Requirements

We will briefly check if and in how far the two approaches meet the requirements from section 2.

Requirement 1: A context-modelling framework has to identify all relevant contextual dimensions. In SAiMotion this strongly depends on the ontology available and on the knowledge of the person who creates the context model. In many situations, the quality of the framework is very good at the beginning and may get outdated over time. In TOWER the context model can either be created by a user or automatically by the system. Whereas in the first case the situation is the same as with SaiMotion; in the latter case the automatically generated context model might miss some contextual dimensions, but has the advantage of always being up-to-date through automatic updates.

Requirement 2: In a context-enhanced community information system, context knowledge has to be associated to information when it is produced and has to be used during information retrieval. This is the case for both approaches. The difference is that in SaiMotion the context information for incoming information is stored in a central repository, whereas in TOWER the context information for incoming information is attached to the information in the form of a context-attribute.

Requirement 3: Context information has to be used as explicit query to the community information system. In both systems this is the case.

Requirement 4: Context-based and content-based retrieval of information should be possible both independently of each other and in combination. In both systems this is the case.

Requirement 5: A context should basically be recognised automatically; however, the system should allow users to explicitly provide context information at the same time (thus providing additional context information that cannot be detected automatically). In both approaches contexts can be recognized automatically. However, only in SaiMotion the system is able to ask the user for context information; in the current state, TOWER solely depends on the automatic processing.

Requirement 6: The system has to consider all partially matching contexts and merge them into a coherent presentation of the information. This is true for both approaches. In SaiMotion partly matching information is always displayed in distinct indicators. In TOWER, users are free to specify whether the information stemming from different contexts should be displayed separately or with one indicator.

Requirement 7: The context-modelling framework has to allow the dynamic ranking of important contextual dimensions used to perform the similarity match. In both systems this is the case.

Requirement 8: The user notification with relevant events has to consider user preferences (like notification frequency, notification channel). In SaiMotion the information is always provided as additional information added to the search results of users. In TOWER users are free to specify when and how they want the information to be presented.

Requirement 9: The modelling effort for modelling and maintaining context models should clearly pay off in terms of improved access to information and increased working efficiency. The modelling effort in SaiMotion is very high, but also the matching precision is very high. In TOWER, the modelling effort is low, but the automatic matching can sometimes be less precise than in SaiMotion.

Requirement 10: The time spent on recognising the current context and on retrieving information relevant to this context has to be reasonably small. Both systems compute their results in reasonable time. So far, the systems have been successfully used with dozens of users at the same time. However, the systems' scalability has yet to be tested for user numbers topping 100 and more.

5.2 Comparison of SaiMotion and TOWER

For comparing the two approaches we use the following dimensions: the modelling technique used, the persistence of a context, the similarity assessment, the way a context is triggered, the context modelling purpose, the modelling responsibility, the modelling effort, the required resources for retrieval, and the modelling precision (see Table 2).

Table 2. Comparison of SAiMotion and TOWER.

Feature	**SaiMotion**	**TOWER**
Modelling Technique	Based on an ontology	Based on attribute-value pairs
Context persistence	Dynamic configuration of contextual dimensions	Persistent, yet evolving descriptions
Similarity assessment	Context-dependent	Comparison of attributes and values
Context triggering	Contexts are similar if similarity measure is above threshold	Users enter/leave contexts according to matching attributes and values
Context modelling purpose	Contextualisation of information to improve information supply	Contextualisation of working situations to improve group awareness
Modelling responsibility	Assigned role	Distributed modelling
Modelling effort	High	Low
Retrieval resources	High	Low
Modelling precision	High	Low

Modelling Technique. While in SAiMotion ontology-based techniques are used to model organisational contexts, TOWER uses attribute-value pairs to model different contextual dimensions.

Context Persistence. In SAiMotion contexts are dynamic configurations of contextual dimensions. A configuration represents a singular context. Such a configuration of contextual dimensions is only made persistent, if it is associated with information that is newly inserted into the repository. In TOWER a context is a persistent object that is created by an administrator. A context knows a set of members that are able to enter and leave this context based on the events that are produced by their everyday activities.

Similarity Assessment. SAiMotion assesses the similarity of contexts based on weighted dynamic similarity measures that are combined out of distance measures for the individual dimension. TOWER uses a straightforward similarity measure based on the number of matching dimensions and attributes within each individual dimension.

Context Triggering. In SAiMotion users are in similar contexts, if the similarity measure is above a certain threshold. In TOWER users can enter or leave predefined contexts. A user enters a context when a set of contextual dimensions matches with the predefined dimensions of that context. Therefore, users can be in several contexts simultaneously.

Context Modelling Purpose. SAiMotion uses context-modelling techniques to associate contextual knowledge with information in order to improve information supply processes. In TOWER, the main purpose of context modelling is to improve the situated awareness of users with similarities in order to facilitate spontaneous contacts, serendipitous encounters, information exchange, and coordination among groups.

Modelling responsibility. In SAiMotion maintaining a context modelling framework (i.e., a set of contextual dimensions and their respective range of possible values) is assigned to a centralised role. In TOWER every user can create own contexts (the creator is automatically the administrator for that context) and specify the set of dimensions used within this context.

Modelling effort. SAiMotion uses complex structures (ontologies) that require a high modelling effort. TOWER relies on attribute-value structures that can be modelled with fairly little effort. TOWER supports the automatic generation and evolution of contexts. For instance, if a shared workspaces systems such as BSCW (Bentley et al. 1997) is used, the system contains a number of users, a number of shared artefacts, tools, and so forth. TOWER can take this information as inputs for a context description—so all members of the shared workspace are members of the context, all artefacts are parts of the contexts, and so forth. When changes in the shared workspace occur (e.g., a new user joins the shared workspace), the context description can be updated automatically.

Retrieval Resources. The similarity measure used in SAiMotion is a weighted combination of individual distance measures. TOWER simply counts the number of exactly matching dimensions. The effort for retrieving similar contexts is significantly higher in SAiMotion.

Modelling Precision. The benefit for higher modelling and retrieval efforts in SAiMotion compared to TOWER is, that the modelling precision is higher.

6. SCENARIO

In an information seeking scenario the two approaches can be combined easily and complement each other meeting the requirements elaborated above. The combination of SAiMotion and TOWER allows modelling a comprehensive framework with several dimensions that are important for adequate support for information seekers. In SAiMotion organisational, domain and content-based, personal, and physical information is modelled; in TOWER the social and project-related information is modelled.

Both systems SAiMotion and TOWER *per se* associate context knowledge to information and events at production time, it is very easy to use this information to support information seekers later on.

Explicit context models in both systems allow context- as well as content-based information retrieval.

SAiMotion allows the user to explicitly provide information about her current context. This information can be used in the combined system to improve reasoning about the user's current context.

The TOWER system offers a whole range of indicators for presenting context information. They can be used to present the information that stems from various different contexts that can be relevant for the user in a specific situation. For instance, a user searching for Java programming environments for Mac OS X might be presented with information about Java programming environments in one tickertape, and with information on programming on Mac OS X in another tickertape. In a third window a list could show other users who are looking for similar information.

Furthermore, the shape of these different indicators can reflect the importance of contextual dimensions. For instance, for users without thorough knowledge about search engines, the information about other users who might be able to help will be displayed more prominently. In both systems users can specify notification preferences. TOWER users can specify relationships between contexts of origin and current work contexts, and can state the preferred type and timing of presentation. They could e.g., specify to receive information about other users looking for similar information in a list when it is discovered. Additionally, they might want to receive information about other users looking for programming in general only in a small window, updated in a frequency of ten minutes. The modelling effort is high in SAiMotion and rather low in TOWER. So, users can decide which type of modelling they want to use. If they prefer to take more effort, they have the advantage of more precise matching between contexts. In the same vain, the reasoning of SAiMotion is slower than the one of TOWER. Users can, again, decide if they prefer speed or precision.

7. CONCLUSIONS

In this paper we have discussed requirements for modelling the context of information seekers. We have presented two approaches for context modelling—one with a highly sophisticated single-user model and one with a lightweight cooperative model. Finally, we have discussed how these two approaches can be integrated to a comprehensive information-seeking scenario.

ACKNOWLEDGEMENTS

The research presented here was carried out and financed by the SAiMotion project (funded by the German BMBF, Funding-Key 01AK900A. and by the IST-10846 project TOWER, partly funded by the EC. We would like to thank our colleagues from the SAiMotion and the TOWER projects as well as the anonymous reviewers for their valuable comments.

REFERENCES

Agostini, A., De Michelis, G., Grasso, M.A., Prinz, W. and Syri, A. Contexts, Work Processes, and Workspaces. Computer Supported Cooperative Work: The Journal of Collaborative Computing 5, 2-3 (1996). pp. 223-250.

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

Belkin, N.J. Anomalous State of Knowledge for Information Retrieval. Canadian Journal of Information Science 5 (1980). pp. 133-143.

Bentley, R., Appelt, W., Busbach, U., Hinrichs, E., Kerr, D., Sikkel, S., Trevor, J. and Woetzel, G. Basic Support for Cooperative Work on the World Wide Web. Human-Computer Studies 46, 6 (June 1997). pp. 827-846.

Clark, H.H. and Brennan, S.E. Grounding in Communication. In Resnick, L.B., Levine, J.M. and Teasley, S.D., eds. Perspectives on Socially Shared Cognition. American Psychological Association, Washington, DC, 1991. pp. 127-149.

Dey, A.K. and Abowd, G.D. Towards a Better Understanding of Context and Context-Awareness. GIT-GVU-99-22, College of Computing, Georgia Institute of Technology, ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf, 1999. (Accessed 12/12/2000).

Gross, T. Towards Ubiquitous Awareness: The PRAVTA Prototype. In Ninth Euromicro Workshop on Parallel and Distributed Processing - PDP 2001 (Feb. 7-9, Mantova, Italy). IEEE Computer Society Press, Los Alamitos, CA, 2001. pp. 139-146.

Gross, T. Ambient Interfaces in a Web-Based Theatre of Work. In Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed, and Network-Based Processing - PDP 2002 (Jan. 9-11, Gran Canaria, Spain). IEEE Computer Society Press, Los Alamitos, CA, 2002. pp. 55-62.

Gutwin, C., Greenberg, S. and Roseman, M. Supporting Workspace Awareness in Groupware. In Proceedings of the ACM 1996 Conference on Computer-Supported Cooperative Work - CSCW'96 (Nov. 16-20, Boston, MA). ACM, N.Y., 1996. pp. 8-8.

Jarke, M., Klemke, R. and Nick, A. Broker's Lounge - An Environment for Multi-Dimensional User-Adaptive Knowledge Management. In Proceedings of the Thirty-Fourth Annual Hawaii International Conference on System Sciences - HICSS 2001 (Jan. 7-11, Kauai, HI). IEEE Computer Society Press, Los Almitos, CA, 2001.

Klemke, R. Context Framework - An Open Approach to Enhance Organisational Memory Systems with Context Modelling Techniques. In Proceedings of the International Conference on Practical Aspects of Knowledge Management - PAKM 2000 (Oct. 30-31, Basel, Switzerland). 2000.

Lenat, D. The Dimensions of Context-Space. http://www.cyc.com/publications.html, 1998. (Accessed 1998).

Prinz, W. TOSCA: Providing Organisational Information to CSCW Applications. In Proceedings of the Third European Conference on Computer-Supported Cooperative Work - ECSCW'93 (Sept. 13-17, Milan, Italy). Kluwer Academic Publishers, Dortrecht, NL, 1993. pp. 139-154.

Wargitsch, C., Wewers, T. and Theisinger, F. An Organisational-Memory-Based Approach for an Evolutionary Workflow Management System - Concepts and Implementation. In Proceedings of the 31st Conference on System Sciences (Los Alamitos, CA). 1998. pp. 174-183.