Context- and Situation-Awareness in Information Logistics

Ulrich Meissen, Stefan Pfennigschmidt, Agnès Voisard, and Tjark Wahnfried

Fraunhofer Institute for Software and Systems Engineering (ISST),
Mollstrae 1, 10178 Berlin, Germany
{ulrich.meissen,stefan.pfennigschmidt,agnes.voisard,
tjark.wahnfried}@isst.fraunhofer.de

Abstract. In order to deliver relevant information at the right time to its mobile users, systems such as event notification systems need to be aware of the users' context, which includes the current time, their location, or the devices they use. Many context frameworks have been introduced in the past few years. However, they usually do not consider the notion of characteristic features of contexts that are invariant during certain time intervals. Knowing the current situation of a user allows the system to better target the information to be delivered. This paper presents a model to handle various contexts and situations in information logistics. A context is defined as a collection of values usually observed by sensors, e. g., location or temperature. A situation builds on this concept by introducing semantical aspects defined in an ontology. Our *situation awareness* proposal has been tested in two projects.

1 Introduction

Information logistics aims at providing a subscriber with the right information at the right time and at the right place (see for instance [4]). Two of its representative applications are ongoing projects at Fraunhofer ISST, namely *Personalized Web Services for the Olympic Games 2008 in Beijing* [10] and *MeLog* ("Message Logistics") [13], which consists in delivering mobile users their personal electronic messages according to their relevance with respect to the users' current *situation*. In such applications, beyond the classical dimensions of time and place, content represents a major challenge. Indeed, the information need that will turn into delivered content is a dynamic concept, i. e., a function of time, space, and preferences of the user, among other parameters.

Location-based services have emerged a few years ago to allow end users to obtain information based on some location, usually the position of the user [16]. Such services, for instance mechanisms to answer a query such as "Where is the nearest subway station?" or "What are the exhibitions in the city today?" are currently receiving a great deal of interest. They manipulate the common aspects of location and time but also more complex notions such as the profile of a user. In most event notification systems (ENS) designed so far, the profile of a user consists of more or less static definitions of personal data such as name and address as well as preference data (cf. [9]). With the dynamic incurred by the time and location components, the profile of the user has to be defined

W. Lindner et al. (Eds.): EDBT 2004 Workshops, LNCS 3268, pp. 335–344, 2004.

in a highly dynamic way, i. e., as a function of the other dimensions. In other words, the user demand may change rapidly according to such dimensions. A type of user demand can be gathered in time intervals. For instance, if a user is in the car, he or she would like to be kept updated on the (current) traffic situation in some area. *Situation awareness* is a solution to this problem; the fact of being in the car represents a certain *situation*.

This paper focuses on a model to handle user situations as well as the surroundings of the user – including time and current location – and other attributes referred to as the *context* of the user. The idea is to abstract from sensors and derive semantics as much as possible. Only then the user demand may be satisfied, i. e., information filtered and personalized. Even though some of these notions have been studied in the past few years, we are not aware of any model that encompasses all these notions in a unified framework. The notion of situation has been studied in different fields of computer science such as computational linguistics (situation theory [2, 3]) and robotics (situation calculus [11, 12]). Although there are similarities to our situation definition, the scope of application of these approaches is different. Our situation model complements the area of information logistics [4] by a formal description of the user's environment and its influences on the information need of the user. The definition of our situation model is based on definitions that had been established in the fields of artificial intelligence [12] and context awareness [1, 14]. Especially the interpretation of context data and their aggregation as studied in [8] is closely related to our work. Research done in the field of semantic networks and ontologies [6, 7] plays an essential role in our model, in order to interpret real situations.

This paper is organized as follows. Section 2 gives the example of a typical application in information logistics (MeLog). This example serves as a reference throughout the paper. Section 3 presents our model of context and situation. In Sect. 4, we get back to the application and we describe it using our model. Finally, Sect. 5 draws our conclusions and presents future perspectives.

2 An Example Scenario

In order to illustrate our needs and motivate our approach towards an integrated model on situations, let us consider a real world scenario: Mr. Busy is a project manager in a large scale distributed enterprise. He spends half of is working time out of his office on business travel. Due to his traveling activities he has a logistic problem with messages: In the average, an amount of about 60 e-mail messages, 5 faxes and 5 voice messages are usually addressed to his office every day. Some of these messages are very important for him to get during his travel, because they might contain useful information for the next business meeting or just for traveling purposes. To solve this problem he has to check these messages regularly which is often rather inconvenient and sometimes impossible.

Let us consider a small snapshot of the business travel plan of Mr. Busy in order to make the task clear. Based on various information sources like his organizer or the travel management unit we can describe his travel in a sequence of situations:

```
 until 12:00 at the office, working
12:00 - 12:30 taking a taxi to the airport 12:30 - 13:30 at the
airport Berlin Tegel 13:30 - 15:00 flight 452 to London Heathrow
```

```
15:00 - 15:20 at the airport London Heathrow 15:20 - 16:30 in a
car with Ms. Miller 16:30 - 19:00 project meeting in London 19:00
- 19:30 taking a taxi to Hotel Comfort ... 10:30 - 07:00 flight
608 to Bejing
```

From the time he left several messages arrived in his office. Now let us presume that Mr. Busy has a perfect virtual assistant who selects only the messages that are relevant to the known situations during his travel. This perfect assistant decides the relevance to the incoming messages as follows:

```
 voice:   the car is repaired and
ready to collect           (not yet relevant) e-mail:  report on
project P1                               (relevant for meeting)
e-mail:  virus alert from IT support                        (not
yet relevant) e-mail:  invitation to a birthday party (not yet
relevant) e-mail:  report on project P3 (not yet relevant) e-mail:
better connecting flight from Bejing available     (relevant for
travel) fax:     night events in London from Hotel Comfort
(relevant for leisure) e-mail: letter from the board about last
year activities   (not yet relevant)
```

Being aware of the current and preferably the future situations of his client is essential for such a perfect assistant. The *MeLog* application described in Sect. 4 utilizes situation awareness by comparing situation patterns with observed situation sequences in order to provide the functionality of such an assistant. This approach is based on a situation model described in the following section.

3 Situation Model

This section is concerned with our situation model, the kernel of our approach. We introduce the concept of a situation and describe its associated operators. These operators allow to handle many real-life situations and to use them in order to deliver the right information at the right time.

3.1 From Context to Situations

A situation is defined in [12] as "the complete state of the universe at an instant of time". However, in order to describe someone's individual situation we do not need the whole state of the universe but rather use a subset that is considered relevant [5]. A state – called *context* in our model – is a collection of *context variables*, each representing one relevant observable real world parameter, e. g.,

$$\begin{aligned} \texttt{gpsLocation} &= (52.5264, 13.4172)\ , \\ \texttt{velocity} &= 1.8\,\mathrm{km/h}\ . \end{aligned}$$

A context can be considered as a snapshot or instantiation of all context variables at some point in time. The observation may physically be done via any kind of sensor function which do not play any role here. The value of a context variable (e. g., gpsLocation) will slightly change from time to time, i. e., from context to context, whereas we would not say that a slight movement of a participant within the conference room really affects the situation of the people attending the meeting. We use the notion of *characteristic features* of a context to get properties that are more stable over time. A characteristic feature – or *characteristic* for short – is a logical proposition about a context or a subset of its components, i. e., its context variables:

$$\texttt{organizationalLocation}\,(\texttt{conferenceRoom})\ ,$$
$$\texttt{kindOfMovement}\,(\texttt{slow})\ .$$

The mapping between context variables (e. g., velocity) and characteristic features (e. g., kindOfMovement) is defined using application-dependent aggregation rules which are also not discussed in detail here. From the examples used in the previous sections one can see the possible existence of a generalization/specialization relation. That means, one characteristic feature can be inferred when knowing another one. If we know, for example, that a project meeting takes place Tuesday, we can say also that it takes place weekdays. To utilize this, we use concept graphs (directed acyclic graphs), where the nodes are connected by *subsumes*-relations (Fig. 1). We utilize these kind of graphs or taxonomies because they are simple and reflect common ways of human thinking and structuring. Throughout this paper we will refer to such kind of graphs as *dimension structures*. Context and its characteristics encompass many *dimensions* or aspects [14], e. g., time, location, activities, or kind of movement, which should be handled separately. A dimension can be viewed as the type of a characteristic feature and is represented by a predicate and a dimension structure. For many of these aspects ontologies representing common knowledge already exist and can be used to express context characteristics.

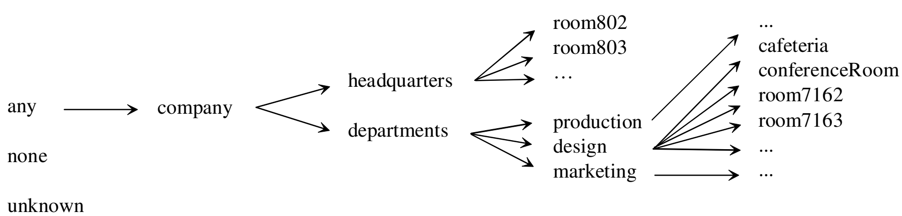

Fig. 1. Example of a dimension structure

We are now able to give a more formal definition of situation. We use the concept of characteristic features described previously. A situation in our model will be formed by a sequence of contexts with invariant characteristics and is described as a triple

$$S = (t_{\text{beg}}, t_{\text{end}}, cs)$$

where
t_{beg} is the starting time of the situation (i. e., the time of the first context of the sequence),
t_{end} is its end time (i. e., the time of the last context of the sequence), and
cs is a set of characteristic features which are invariant throughout the sequence. cs is interpreted as the conjunction of all characteristic features: $cf_1 \wedge \ldots \wedge cf_n$.

This definition offers a rich concept that enables us to describe the activity and the location of the user, such as being “at home”, “at the office”, “in the train”, or “on the phone”. We would like to emphasize two major features. First, the notion of situation may encompass many dimensions as one can be both in a taxi and on the phone. To handle this aspect efficiently we chose not to mix dimensions and to consider them separately. Second, the generalization/specialization notions of our characteristics enable the description of situations on different levels of granularity. It can be general, e. g., “traveling”, or more precise, e. g., “in a taxi going down the Champs Elysees”.

Situation Awareness. Recognizing and identifying situations is a central requirement for applications in information logistics. Applications that make use of situations and are able to handle changes such as entering or leaving a situation are denoted *situation-aware applications*. There are two ways for such an application to utilize the situation model: (1) Analysis of past and current situations, where context information is available. (2) “Situation construction” or planning of situations with assumed characteristics, where context information is not available, but where the characteristics (which are propositions about the contexts) impose restrictions on context variables. These restrictions can be used afterwards to check whether the planned situation actually takes or took place, or not. In addition, one can define *typical situations* and check whether an actual situation complies with a certain definition. This last point will be further discussed as an example in Sect. 4.2. To define typical situations we additionally use the notion of *situation patterns*, as logical propositions about situation characteristics.

3.2 Operators

The situations defined above are handled through operators. We distinguish operators that manipulate whole situations from the ones that work on characteristics of situations.

Operators on Characteristics. The following three operators have in common that they deal with similarities or analogies between characteristics:

$\texttt{generalize}\,(cs_1, cs_2) \rightarrow cs_\mathrm{r}$

This operator takes two sets of characteristics cs_1, cs_2, and finds the most specific set of characteristics cs_r that is common for both. In order to do that it utilizes the subsumes relation of the dimension structures used and finds the least common ancestor.

$\texttt{fulfills}\,(cs, p) \rightarrow \{\texttt{true, false}\}$

This operator determines whether a set of characteristics cs complies to the conditions of a situation pattern p.

$\texttt{compare}\,(cs_1, cs_2) \rightarrow [0, 1] \subset \mathbb{R}$

This operator computes the similarity between two given sets of characteristics cs_1, cs_2 by applying a similarity metric on the subsumes-paths within the dimensions (e. g., semantic distance [15]).

Operators on Situations. We use the notion of *situation sequences* to denote a series of directly subsequent situations.

$\texttt{previous}\,(s) \rightarrow s_\mathrm{p}$

This operator determines the predecessor s_{p} of a given situation s (*nil* if there is no situation known).

$\texttt{next}\,(s) \rightarrow s_{\mathrm{s}}$

This operator determines the successor s_{s} of a given situation s (*nil* if there is no situation known).

$\texttt{combine}\,(seq) \rightarrow s_{\mathrm{r}}$

This operator tries to find a generalized situation s_{r} covering the whole time interval of a situation sequence seq. The resulting situation will be built such that, the begin time equals the begin time of the first situation and the end time equals the end time of the last situation of the sequence. The characteristic of the resulting situation will be the most specific generalized characteristic of all situations. If the sequence contains no situations the result will be *nil*. If the sequence contains only one situation the result will be this situation.

4 Application

In this section, we present the *MeLog* system as a functional prototype of a situation-aware application in information logistics. MeLog is short for "message logistics" and was developed within the scope of a research project at Fraunhofer ISST in 2002 and 2003.

4.1 The MeLog System

MeLog gives automatic decision support in order to deliver electronic messages such as e-mail, converted fax, or voice messages at the right time. Based on user situations such as "at the airport", "eating at a restaurant", and "during the project meeting", the system recognizes the most relevant topics and delivers messages that have high information value for these or the following situations. In order to do that, MeLog in a first step calculates the *relevance* between a message and the user's situations via his or her topics of interest. Based on relevance, time-dependent *utility* and *acceptance* functions are then taken into account in order to calculate the time of highest information value. The strategies and algorithms of these calculations are not discussed here in detail.

The MeLog system consists of several components. As shown in Fig. 2 they can be divided into components of the system kernel layer and components of the data model layer. The kernel layer encompasses all functional components. That is, this layer is responsible for the recognition of context data, the aggregation of that data, and the derivation of situations.

The data layer manages all information sources necessary for predicting the information value, such as dimension ontologies, situation history, etc. One concrete example of dimension data we used is the user's address book. The set of topic descriptions known as overall "interesting" to the user also belongs to the data layer. Examples of topic structure sources are the user's e-mail folder structure or similar structures found in document management systems.

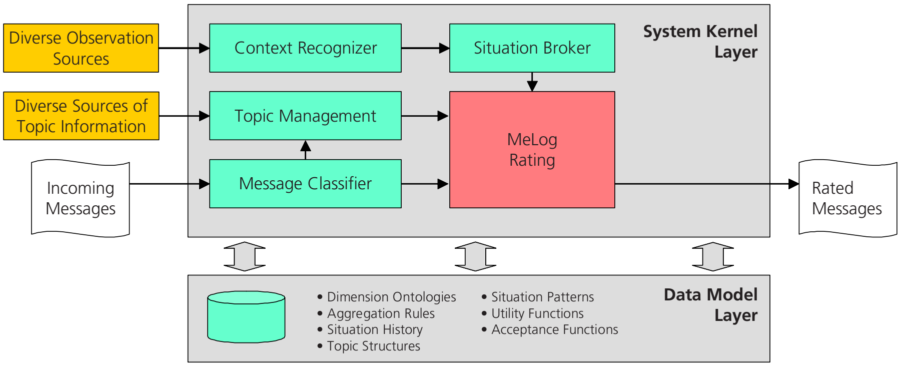

Fig. 2. MeLog system architecture

Another category of input is context data, gathered by different observation sources. Besides other sources MeLog takes advantage of the entries in the user's electronic diary, such as appointments. These entries serve as information about situations that are planned by the user. Any kind of sensor data incorporated by the system is also an observation source in this view. Thus it is a major capability of the approach to consider aspects of planned situations as well as real world sensor data.

4.2 Back to the Example Scenario

In the following short example we want to find out whether actual situations, in this case the flight to London and the project meeting, fit to typical situations defined by the user, in this case traveling and meetings for project P1. Afterwards the relevance of messages can be calculated on the basis of these typical situations and their associated information demand.

In our example context data is available from the following sensors: (1) location sensors; decimal geo coordinates (type: WGS84), (2) electronic organizer; entry keywords (type: string), (3) laptop; used documents (type: string).

Our situation characteristics are described by the following dimensions: (1) transportation in use, (2) organizational location, (3) kind of movement, (4) person in presence, (5) document in use.

Behind each of these dimensions a domain-dependent dimension structure is used in order to describe meta knowledge about the values within the dimension. In our example Mr. Busy has defined characteristics of two typical situations as well as the information demand associated with these patterns:

```
 p("traveling")  =
[transportationInUse(any), kindOfMovement(any/../significant)]
p("meeting P1") =
[organizationalLocation(any/../company/departments/design)
                   OR organizationalLocation(any/../company/headquarters),
                  personInPresence(any/../staff/design/project teams/team P1)
                   OR personInPresence(any/../partners/companyX/marketing),
                  kindOfMovement(none),
                  documentInUse(any/../work/projects/project P1)]
```

The first one means *"when Mr. Busy is traveling, he uses transportation and he is moving significantly"*. The second means *"when Mr. Busy is in a meeting about Project P1 he is either in the headquarters or design department, people from the project team or marketing partners are in presence, he is not moving, and he uses documents for project P1"*. The *generalize* operator can be used to derive such patterns from past situations, e. g., if several "traveling" situations already took place the common characteristics of a typical situation can be derived from generalizing these observed situations.

Table 1. Context data gathered during business travel

Time	Coordinates from location sensor	Keywords from electronic organizer	Documents from laptop
...	...	...	...
11:57:45	(52.5264, 13.4172)	office, working	design-plan-alpha-2b.tex
11:57:46	(52.5263, 13.4171)	office, working	none
...	...	...	...
15:42:01	(52.4259, 12.6045)	flight, berlin, london	project-P4-report.pdf
...	...	...	...
17:35:21	(51.5851, 0.0351)	meeting, miller, turner	project-P1-milestone.ppt
...	...	...	...
19:31:22	(51.5626, 0.0687)	hotel	none
...	...	...	...

Table 1 shows a part of the raw sensor data the application gets before and during the business travel. According to predefined aggregation rules, nodes within the dimensions structures can be derived from that data. Based on that the characteristics of an observed situation can be inferred. Such rules tell the application, for instance, which of the observed coordinates correspond to which node within the organizational location ontology (cf. Fig. 1). Based on this aggregation MeLog derives two situations from the sensor data, situation s_1 (*flight*, 13:30 - 15:00) and situation s_2 (*meeting*, 16:30 - 19:00). The characteristics of these situations are shown below.

```
 cs(s1) =
[transportationInUse(any/../airplane),
          kindOfMovement(any/../significant),
          documentsInUse(any/../work/projects/project P4/project-P4-report.pdf)]
cs(s2) =
[organizationalLocation(any/../company/departments/design/room7163),
          personInPresence(any/../staff/design/project teams/team P1/Ms. Miller),
          personInPresence(any/../staff/design/project teams/team P1/Mr. Turner),
          kindOfMovement(none),
          documentInUse(any/../work/projects/project P1/project-P1-milestone.pdf)]
```

Now the application uses the *fulfills* operator in order to decide whether these observed situations fit to the predefined situation patterns. Checking whether the characteristics of the "flight" situation fulfill the characteristics of the "traveling" pattern gives the following term:

```
 fulfills(cs(s1), p("traveling"))
  = fulfills(transportationInUse(airplane), transportationInUse(any)) AND
    fulfills(kindOfMovement(significant), kindOfMovement(significant))
  = TRUE
```

Analogically follows:

```
 fulfills(cs(s2), p("traveling"))  =
FALSE fulfills(cs(s1), p("meeting P1")) = FALSE fulfills(cs(s2),
p("meeting P1")) = TRUE
```

According to the outcome of the operations above the information demand associated with a typical "travel" situation can be also associated with the observed "flight" situation. The demand associated with a typical "project P1 meeting" situation can be associated with the observed "meeting" situation.

5 Conclusion

This paper presented a model to handle various contexts and situations in information logistics. A context is defined as a collection of values extracted from the environment at a certain time (e. g., location and speed extracted by sensors). A situation builds on this concept by introducing propositions about context data, which form characteristic features that are stable over a time interval. Semantical aspects in form of ontologies are used to enable interpretation of situations by applications. When the system is able to deduce situations from the context, it is implicitly able to infer the user's information demand. This enables the delivery of information relevant at a certain point in time.

Currently we focus on the prediction of situations. By mining situation sequences, one is able to find patterns in situations and to predict future situations, hence anticipate the information that will be of interest to the user in the future. A second research direction is to use our model as a means to precisely describe and analyze scenarios in information logistics, in order to derive application needs in that field. Grouping people according to the situation they share combined with the analysis of appropriate levels of individualization are points of particular interest in such applications.

Acknowledgements

We wish to thank the anonymous reviewers for their useful comments, as well as Andreas Billig, Stefan Decker, Rüdiger Gartmann, and Norbert Weißenberg for interesting discussions.

References

1. G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In H.W. Gellersen, editor, *Handheld and Ubiquitous Computing: First International Symposium, HUC'99*, volume 1707 of *LNCS*, Berlin/Heidelberg/New York, 1999. Springer Verlag.
2. V. Akman and M. Surav. The use of situation theory in context modeling. *Computational Intelligence*, 13(3):427–438, 1997.
3. R. Cooper, K. Mukai, and J. Perry, editors. *Situation Theory and its applications*, volume I of *CSLI Lecture Notes No. 22*. CSLI: Center for the Study of Language and Information, Stanford University, California, 1990.

4. W. Deiters, T. Löffeler, and S. Pfennigschmidt. The information logistics approach toward user demand-driven information supply. In D. Spinellis, editor, *Cross-Media Service Delivery*, pages 37–48, Boston/Dordrecht/London, 2003. Kluwer Acadamic Publishers.
5. A. K. Dey. *Providing architectural support for building context-aware applications*. PhD thesis, College of Computing, Georgia Institute of Technology, 2000.
6. T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Boston/Dordrecht/London, 1993. Kluwer Academic Publishers.
7. M. Gruninger and Jintae Lee. Ontology applications and design. *Communications of the ACM*, 45(2):39–41, 2002.
8. S. Haseloff. Context gathering – an enabler for information logistics. In P. Chamoni, W. Deiters, N. Gronau, R.-D. Kutsche, P. Loos, H. Müller-Merbach, B. Rieger, and K. Sandkuhl, editors, *Multikonferenz Wirtschaftsinformatik (MKWI) 2004, Band 2*, pages 204–216, Berlin, 2004. Akademische Verlagsgesellschaft.
9. A. Hinze. *An Adaptive Integrating Event Notification Service*. PhD thesis, Free University of Berlin, Computer Science Institute, 2003.
10. B. Holtkamp, R. Gartmann, and Y. Han. Flame2008: Personalized web services for the Olympic Games 2008 in Beijing. In *e-2003, e-Challenges Workshop*, 2003.
11. Y. Lespérance, H. J. Levesque, Fangzhen Lin, D. Marcu, R. Reiter, and R. B. Scherl. A logical approach to high level robot programming – a progress report. In B. Kuipers, editor, *Working notes of the 1994 AAAI fall symposium on Control of the Physical World by Intelligent Systems*, New Orleans, LA, 1994.
12. J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. Report Memo AI-73, Department of Computer Science, Stanford University, Stanford, California, 1968.
13. U. Meissen, S. Pfennigschmidt, K. Sandkuhl, and T. Wahnfried. Situation-based message rating in information logistics and its applicability in collaboration scenarios. In *Proceedings of the 30th Conference on EUROMICRO*. IEEE Computer Society, 2004 (to appear).
14. D. Petrelli, E. Not, M. Zancanaro, C. Strapparava, and O. Stock. Modelling and adapting to context. *Personal Ubiquitous Computing*, 5(1):20–24, 2001.
15. P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In Chris Mellish, editor, *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, San Francisco, 1995. Morgan Kaufmann.
16. J. Schiller and A. Voisard, editors. *Location-based Services*. Morgan Kaufmann, San Francisco, 2004.