A Context Framework for Process-Oriented Information Logistics*

Bernd Michelberger[1], Bela Mutschler[1], and Manfred Reichert[2]

[1] University of Applied Sciences Ravensburg-Weingarten, Germany
{bernd.michelberger,bela.mutschler}@hs-weingarten.de
[2] Institute of Databases and Information Systems, University of Ulm, Germany
manfred.reichert@uni-ulm.de

Abstract. A continuously increasing data overload makes it a challenging task for knowledge-workers and decision-makers to quickly identify relevant information, i.e., information they need when executing business processes. To tackle this challenge, *process-oriented information logistics* is a promising approach. The basic idea is to provide the right process information, in the right format and quality, at the right place, at the right point in time, and to the right people. To achieve this, it becomes particularly important to take the work context of process participants into account. In fact, knowing and utilizing context information is a prerequisite to effectively provide relevant process information to process participants. This paper provides a sophisticated *context framework* for enabling context-awareness in process-oriented information logistics.

Keywords: process-oriented information logistics, context-aware delivery of process information.

1 Introduction

Nowadays, enterprises are faced with a continuously increasing amount of data [1]. Knowledge-workers and decision-makers suffer from this data overload, since it makes it difficult for them to identify and access the needed information to perform their current tasks in the best possible way [2]. In the following, we call this information "process information", i.e., process information is information supporting process participants when working on business processes. Examples include e-mails, office files, best practices, or process descriptions. In practice, however, this alignment is difficult to accomplish since process information is typically handled separately from business processes and their execution [3].

To close this gap, *process-oriented information logistics* (POIL) is a promising approach. Goal is to provide the right process information, in the right format and quality, at the right place, at the right point in time, and to the right people. More precisely, POIL enables the process-driven, context-aware delivery of

* This paper was done in the niPRO research project. The project is funded by the German Federal Ministry of Education and Research (BMBF) under grant number 17102X10. More information can be found at http://www.nipro-project.org.

W. Abramowicz et al. (Eds.): BIS 2012, LNBIP 117, pp. 260–271, 2012.

process information to knowledge-workers and decision-makers. POIL is particularly suitable for knowledge-intensive business processes involving large amounts of process information, user-interaction, and decision-making.

Various approaches have been proposed to enable POIL, including *data warehousing* and *business intelligence*. However, these approaches have not primarily been designed with POIL in mind. Data warehousing, for example, rather focusses on the creation of an integrated database [4]. Opposed to this, POIL focuses on the management of process information flows to support the execution of business processes. Traditional business intelligence, in turn, addresses data analytics and is typically completely isolated from business processes execution [3]. Moreover, information supply is often restricted to decision-makers. Conversely, POIL focuses on integration and analysis of process information as well as their delivery to both knowledge-workers and decision-makers.

What has been neglected so far is the support of knowledge-workers and decision-makers by providing personalized and contextualized process information. The latter is required to address the different needs of process participants. For example, less experienced process participants need more detailed information than experienced ones. To enable such differentiation, a process participant's context needs to be identified. For this purpose, his or her situation is described according to its characteristics, so-called *context information*. Besides process-related context information (e.g., process step, temporal process constraints), user-related context information (e.g., user name, role, experience level), device-related context information (e.g., display size, bandwidth), location-based context information (e.g., position), time-based context information (e.g., current date), and environment-related context information (e.g., temperature, noise level) may be considered as well. This paper proposes a context framework for POIL and the handling of context information to support the context-aware delivery of process information being relevant for process participants.

The presented research is performed in the niPRO project. In this project we apply semantic technology to integrate process information within *process information portals*. Our overall goal is to support knowledge-workers and decision-makers with the process information needed depending on their current working context. Key challenges include the provision of contextualized process information, flexible visualization of process information [5], and the development of design approaches for different levels of process information quality [2].

This paper is organized as follows. Section 2 gives a motivating example. Section 3 motivates the need for context-awareness in POIL and section 4 presents our context framework. Section 5 discusses related work. Finally, section 6 concludes the paper with a summary and an outlook.

2 Motivating Example

We use a scenario from the clinical domain, to motivate our approach. This scenario is based on lessons learned during an exploratory case study we performed at a German university hospital [6]. In this case study we analyzed the process of an unplanned, stationary hospitalization, including patient admission, medical indication in the anesthesia, surgical intervention, post-surgery treatment, patient discharge, and financial accounting & management.

Our scenario (cf. Fig. 1) focusses on the ward round. First, the ward round is prepared (1), i.e., the doctor scans patient information and current medical instructions (e.g., endoscopic investigations, physical therapies). After finishing initial preparations, the doctor visits his patients. The doctor communicates with a patient and asks for information about his status (2). This information is written down by a nurse in parallel (3). Afterwards, the patient is examined (4). This activity includes the analysis of blood values and further follow-up diagnosis. Then the doctor creates medical orders (5). Finally, a nurse updates patient information and initiates further medical orders (6).

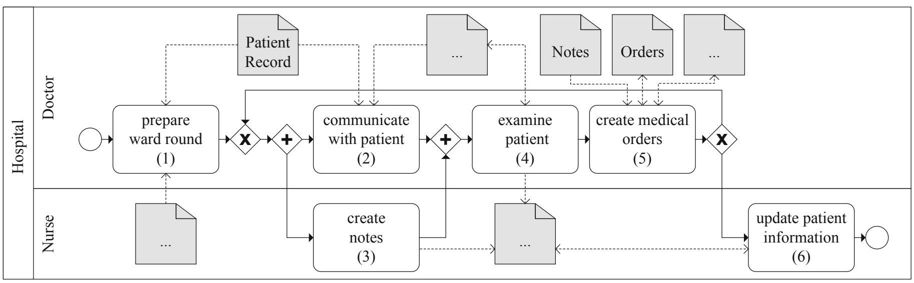

Fig. 1. Motivating example: Ward round

For each of the six process steps a variety of process information is needed. For example, to perform the process step "create medical orders" (5) a doctor needs access to blood values, notes, and current medical orders. Note that the mentioned process information only constitutes a small part of all processed information. In practice, there exist numerous different process information distributed across data sources (e.g., databases, shared drives, Intranet portals) [6]. Typical process information include, for example, process descriptions, working guidelines, operational instructions, forms, checklists, and best practices (e.g., documented in text documents, spreadsheets, and e-mails) [2].

As discussed, it is a big challenge to align process information with business processes. To reach this goal, different facets of POIL have to be addressed. In a first step, we have investigated issues related to process information quality [2]. In this paper, we take a closer look at context-awareness in POIL and propose a context framework.

3 Context-Aware Information Delivery

Context-awareness in POIL aims at the context-aware delivery of process information being relevant for a process participant. We adopt the notion of Dey [7] and define *context-awareness* in POIL in a general way: POIL uses context information to deliver relevant process information to process participants, where relevancy depends on the participant's task and process information quality requirements such as completeness or granularity [2].

Generally, context-awareness in POIL comprises three basic aspects: *sensors*, *context*, and *situation* (cf. Fig. 2). Reconsider our motivating example (cf. Fig. 1) and assume that a doctor performs process step "communicates with patient". Thus, the *situation* (S) at hand could be described as follows: "Doctor Peter Miller communicates with a patient on Monday, 12th November, 2011, in room number 301 using a tablet computer". Based on this we are able to characterize the situation using context information. For example, process-related context information (e.g., process step: "communicates with patient"), user-related context information (e.g., first name: "Peter", last name: "Miller", role: "doctor"), time-based context information (e.g., weekday: "Monday", day: "12th", month: "November", year: "2011"), location-based information (e.g., room number: "301"), and device-related context information (e.g., used device: "tablet computer") can be utilized.

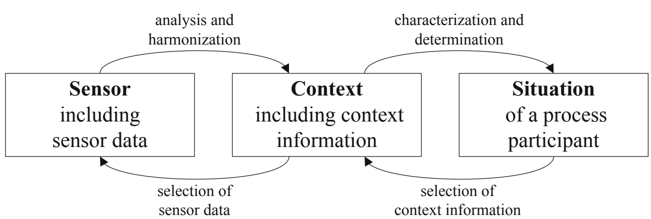

Fig. 2. Relationship between situation, context, and sensors

Particular context information (e.g., last name, role, day) is called *context aspect* (CA). Context aspects are further classified into different categories (e.g., process, user, time) denoted as *context factors* (CF). The set of all context factors is called *context* (C) (cf. Fig. 3). In order to determine specific values of context aspects, *sensors* (SE) are required. For example, to determine the context aspect "temperature" the sensor "thermometer" can be used. In the following we take a closer look at the three basic aspects (cf. Fig. 2).

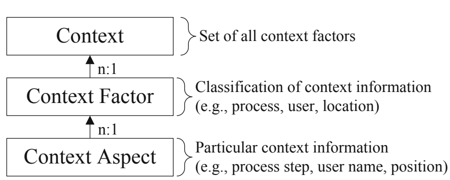

Fig. 3. Relationship between context, context factors, and context aspects

3.1 Sensors

The term *sensor* is specified by Haseloff as "any hardware or software systems that provides data about the entire or a part of the context of one or more entities" [8]. According to Indulska and Sutton [9] sensors can be classified into three categories: *physical sensors* (e.g., thermometer, microphone), *virtual sensors* (e.g., keyboard input, touch display movement, database trigger), and *logical sensors* (e.g., detect a process participant's position by analyzing logins at devices and a mapping of devices to locations). The main task of a sensor is to provide sensor data representing the initial value of the context aspects (e.g., the lightning sensor identifies the value of the context aspect lightning).

Based on this characterization, we can give a formal definition of the term *sensor*. Let SE be the sensor and v be the sensor value. We distinguish between *simple* (cf. Formula (1)) and *logical* (cf. Formula (2)) sensors. For example, a simple sensor can be a Global Position System (GPS) module determining the current position of a process participant. A logical sensor, in turn, can be a software system determining the user name based on first name and last name.

$$SE_{\text{simple}} := v \tag{1}$$

$$SE_{\text{logical}} := \{v_1, v_2, ..., v_{\text{n}}\},\ n \geq 2 \tag{2}$$

3.2 Context

The notion of *context* as defined by Schilit et. al [10] or Pascoe et. al [11] is too specific and is based on an explicit set of context factors. Hence, these definitions become problematic when additional, so far unconsidered context factors need to be considered. In our research we need a more dynamic composition of context factors depending on the situation and on available sensors.

Depending on the process participant's situation different context factors are relevant. For example, to be able to update patient information (6) through the nurse it is not important to know where the task is performed. Conversely (e.g., in a case of an emergency) it is important to know which doctor is nearest to the emergency department. Therefore, we define context in a more general way according to Dey [7]: Context is any information that may be used to characterize the situation of an entity. The latter may be a person, location, or object being considered as relevant for the interaction between a process participant and a process information portal including the process participant and process information portal themselves.

We can now provide a formal definition of the terms *context*, *context factor*, and *context aspect*: Let C be the context, CF be the context factor, CA be the context aspect, and SE the sensor. C, CF, and CA can be defined as follows:

$$C := CF_1 \cup CF_2 \cup ... \cup CF_{\text{n}} \tag{3}$$

$$CF := \{CA_1, CA_2, ..., CA_{\text{n}}\} \tag{4}$$

$$CA := \{SE_1, SE_2, ..., SE_{\text{n}}\} \tag{5}$$

3.3 Situation

Finally, a *situation* can be characterized as "the world state at an instant of time" [12,13]. Haseloff more accurately says that "a situation is a part of the world state at a specific point in time or within a specific time interval" [8]. In other words, a situation represents the instantiation of the context at an instant of time. However, to describe the situation of a process participant we do not need the whole world state, but only those parts which might be relevant for POIL. Let S be the situation, C the context, t_{start} the starting time of the situation, and t_{end} its end time. S can then be defined as follows:

$$S := \langle C, t_{\text{start}}, t_{\text{end}} \rangle \qquad (6)$$

The next section presents our context framework to enable context-aware delivery of process information.

4 Context Framework

Our context framework aims at the context-aware delivery of relevant process information to process participants. It has been influenced by mobile and ubiquities computing and is therefore applicable to mobile scenarios as well (e.g., a ward round). The fundamental difference between our framework and existing ones is the explicit consideration of business processes. In fact, existing frameworks strongly focus on geographic services (e.g., provide the local temperature based on a current position) but do not address important ideas of POIL as discussed in Section 3. When compared to existing frameworks, our context framework does not directly provide any context information to applications (e.g., a weather application). Instead, it utilizes context information to determine the process information a process participant needs. More precisely, our context framework deals with gathering, representing, storing, analyzing, and providing of context information in order to enable the context-aware delivery of process information. As a consequence, existing frameworks can only be partially transferred. For the remainder of this paper we introduce the architecture of the framework and an ontology-based context modeling approach handling and representing context information in a machine-interpretable form.

4.1 Context-Framework Architecture

Generally, a context framework can be based on different architectures depending on business requirements. Chen [14], for example, distinguishes between three different architectural designs: *direct access to sensors*, *context server*, and *middleware infrastructure*. We adopt the latter viewpoint for several reasons, e.g., the reduced complexity resulting from the reduced number of data connections as well as the separation of business logic from the presentation layer and the data layer. Figure 4 illustrates the layered architecture of our context framework.

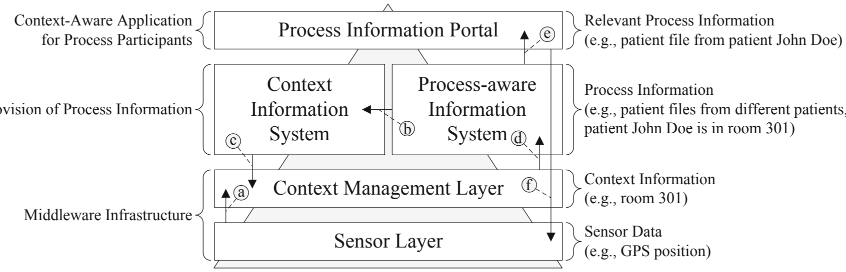

Fig. 4. Architecture of the context framework

The *sensor layer* is responsible for the management of sensor data (e.g., temperature, keyboard input) collected by different sensors (e.g., thermometer, keyboard). The sensor layer provides logical functionality, for example, functions to identify the role of a user by analyzing his or her access rights. Furthermore, the sensor layer allows for adding, removing, and switching (e.g., the GPS module will be replaced by a radio-frequency identification (RFID) system) sensors as well as encapsulating sensor communication (i.e., applications do not directly access sensor data).

The *context management layer* manages context information. Its main components include a context management layer interface, a context analytic engine, and a context model (not shown in Figure 4). The context management layer interface enables retrieval of sensor data from the sensor layer and provision of context information to higher layers via public interfaces. The context analytic engine allows for reasoning, interpreting, and aggregating context information (e.g., instead of GPS coordinates, the specific room number is given) [15]. Finally, the context model is responsible for storing and handling context information (cf. Section 4.2).

The *context information system* provides process information (e.g., which device belongs to which user, hospital map) to enrich available context information. The context information system can be seen as a support application. In the area of mobile computing, a geographic information system (GIS) has similar goals, but is limited to geographically information.

The *process-aware information system* (PAIS) contains integrated process information to support the execution of business processes, i.e., by delivery of process information to process participants. Its task is to gather process information from different data sources (e.g., databases, applications, shared drives), to analyze this process information (e.g., by using text similarity, usage pattern), and to offer it via public interfaces to other applications. Other functions include monitoring, event handling, and process information security.

The *process information portal* is responsible for the context-aware delivery of relevant process information to knowledge-workers and decision-makers.

Figure 4 also shows the dependencies between the different architectural layers. The sensor layer provides sensor data (e.g., user name, current process step) to the context management layer (a). Simultaneously, the context information system provides certain process information (e.g., inventory lists, building maps) to the context management layer (b). The context information system obtains its process information from the PAIS (c). Based on the data/information flows of (a) and (b), the context management layer identifies context information and makes it available to the PAIS (d). The PAIS, in turn, uses this context information to identify relevant process information. The latter is then provided to the process information portal (e), which offers relevant process information to employees. Besides, the process information portal can be a sensor for the sensor layer (e.g., in order to gather user actions, clickstreams) (f).

The next section deals with the representation and handling of context information in ontology-based context model.

4.2 Context Model

The *context model* constitutes a fundamental part of our context framework. We use the context model to store and handle context information in a machine-interpretable form. Table 1 summarizes fundamental requirements (R1-R10) for such context models, which were gathered based on a literature study, two exploratory case studies [6], and an additional online survey [16].

Table 1. Requirements for a context model in POIL

#	Requirement
R1	The context model should represent all context information being relevant to the process participant's situation.
R2	The context model should be able to hide irrelevant context information in specific situations (e.g., location in non-mobile scenarios).
R3	The context model should be flexible and scalable to cope with the challenges of different update rates of context information.
R4	The context model should enable an efficient context analytic (e.g., reasoning, interpreting, aggregation) of context information.
R5	The context model should allow storing and handling historical context information.
R6	The context model should allow an efficient handling (e.g., fast processing, easy accessibility) of context information.
R7	The context model should be combined with the process information in order to provide contextualized process information
R8	The context model should be able to interpolate context information, to cope with incomplete context information.
R9	The context model should be easy to use, so that applications designers can easily translate real-world information to context information.
R10	The context model should store context information taking into account privacy and security issues.

Generally, a context model for POIL should represent all context information being relevant in the current situation of a process participant. However, a context model has to be restricted because the set of context information is infinite [17]. Indeed, any context modeling approach can only capture some parts of all possible context information. Hence, what we require is a classification of context information which allows reducing the complexity of context modeling. For instance, context information (e.g., first name, last name) in the same category (e.g., user) can be processed using the same or similar algorithms. We use the following six context factors (cf. Fig. 5) in POIL.

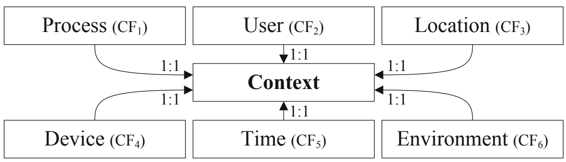

Fig. 5. Our Context Factors

- *Process* (CF_1) includes process-related context aspects and reflects on what is currently (and in the past) happening. This includes, for example, general process aspects (e.g., process schemas, process instances, goals), time-based process aspects (e.g., durations and time lags between process steps, restricting execution times), responsibility-based process aspects (e.g., process owner), and data-based process aspects (e.g., input and output files).
- *User* (CF_2) includes user-related context aspects and reflects who is involved in a certain situation. Thereby, we distinguish between explicit user aspects (e.g., user name, first name, last name, birthday, department) and implicit ones (e.g., experience, interests).
- *Location* (CF_3) includes location-based context aspects and reflects where a situation takes place. This context factor includes both physical (e.g., GPS coordinates, Geolocation, and RFID systems) and logical (e.g., meeting room or office room) location aspects.
- *Device* (CF_4) includes device-related aspects and reflects which devices are used in a certain situation. It includes device type aspects (e.g., personal computer, notebook, tablet, smartphone), hardware aspects (e.g., processor, disk space, and display size), software aspects (e.g., operating system, installed applications), and others (e.g., display properties, bandwidth).
- *Time* (CF_5) includes time-based aspects like current time, virtual time, time zone, business days, and calendar week.
- *Environment* (CF_6) includes environment aspects and reflects what environmental aspects influence a situation. We distinguish between physical aspects (e.g., noise level, lightening), organizational aspects (e.g., cooperate culture, enterprise policies, cooperate identity guidelines), legal requirements (e.g., privacy policy, regulations), and others.

Based on these context factors, it becomes easier to model a context. Different context modeling approaches can be used for this purpose: *key-value models*, *markup scheme models*, *graphical models*, *object-oriented models*, *logic-based models*, and *ontology-based models* [18]. In our framework, we use ontology-based models (cf. Table 2), since there exists powerful tool support for ontologies. Furthermore, partial validation and distribution of context information becomes possible and ontologies allow for an easy linking to other ontology-based models (e.g., ontology-based process information models and business process models). Finally, ontologies have strengths relating to normalization and formality. Several authors (e.g., [18]) share our assessment that ontology-based models provide a promising approach to deal with the challenge of context modeling.

Table 2. Comparison between context modeling approaches

Criteria	Key-Value	Markup	Graphical	Object	Logic	Ontology
Ease of use	++	+	o	o	-	o
Formalization	-	o	o	o	++	++
Expandability	--	+	o	+	-	++
Expressiveness	-	o	+	+	++	++

++: very good, +: good, o: neutral, -: bad, --: very bad

Altogether, the context model is responsible for storing context information. Based on this context information, a PAIS is able to better identify relevant process information for process participants.

5 Related Work

Bucher and Dinter [3] conducted a study to assess benefits, design factors, and realization approaches for POIL. Management challenges related to information logistics (IL) are discussed by Winter [19]. Heuwinkel and Deiters [20] demonstrate the possibilities and advantages of IL in the healthcare sector.

Context and context-awareness in general are discussed by Dey [7], Schilit et. al [10], and Pascoe et. al [11]. Context-awareness in IL is discussed by Haseloff [8], Meissen et. al [13], and Lundqvist et. al [21].

Further approaches have been proposed to deal with challenges of context-awareness and context modeling. Especially in the field of mobile computing a numerous of context frameworks have been proposed (e.g., Context Toolkit [22], Hydrogen [23]). More frameworks exist in the field of information retrieval (e.g., SAiMotion [24]). A broader view on context models supporting business process agility is given by Thönssen and Wolff [25].

Schilit et. al [10] investigate possible context factors. They distinguish between location, identity, and device. Dey et. al [7] state that location, identity, activity, and time are more important than other context factors. Kaltz et. al [26] propose user & role, process & task, location, device, and time as possible categorization of web application scenarios.

6 Summary and Outlook

This paper proposes a context framework for enabling context-awareness in process-oriented information logistics. We motivate the need for context-awareness and show why the handling of context information is success-critical with respect to the context-aware delivery of process information. Most important, we introduce the our context framework, which deals with gathering, representing, storing, analyzing, and providing of context information along executed business processes. More specifically, we describe the framework's architecture and introduce important context factors and context aspects (to be used in context modeling).

Future research will include a more detailed investigation of the presented context aspects (also in cross-organizational scenarios), the development of a proof-of-concept application implementing our context framework, and further research on the handling of context modeling.

References

1. Edmunds, A., Morris, A.: The Problem of Information Overload in Business Organisations: A Review of the Literature. Int'l J. of Information Management 20(1), 17–28 (2000)
2. Michelberger, B., Mutschler, B., Reichert, M.: Towards Process-oriented Information Logistics: Why Quality Dimensions of Process Information Matter. In: Proc. 4th Int'l Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2011), Hamburg. LNI, vol. 190, pp. 107–120 (2011)
3. Bucher, T., Dinter, B.: Process Orientation of Information Logistics - An Empirical Analysis to Assess Benefits, Design Factors, and Realization Approaches. In: Proc. 41st Annual Hawaii Int'l Conf. on System Sciences, pp. 392–402 (2008)
4. Lechtenbörger, J.: Data warehouse schema design. Infix Akademische Verlagsgesellschaft Aka GmbH, PhD Thesis, University of Münster (2001)
5. Reichert, M., Kolb, J., Bobrik, R., Bauer, T.: Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. In: Proc. 27th ACM Symposium On Applied Computing (SAC 2012), 9th Enterprise Engineering Track, Trento (accepted for publication, 2012)
6. Michelberger, B., Mutschler, B., Reichert, M.: On Handling Process Information: Results from Case Studies and a Survey. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. LNBIP, vol. 99, pp. 333–344. Springer, Heidelberg (2012)
7. Dey, A.K.: Providing Architectural Support for Building Context-Aware Applications. PhD Thesis, Georgia Institute of Technology (2000)
8. Haseloff, S.: Context Awareness in Information Logistics. PhD Thesis, Technical University of Berlin (2005)
9. Indulska, J., Sutton, P.: Location Management in Pervasive Systems. In: Proc. Australasian Information Security Workshop Conf. on ACSW Frontiers 2003, Adelaide, vol. 21, pp. 143–151 (2003)

10. Schilit, B.N., Adams, N., Want, R.: Context-Aware Computing Applications. In: Proc. 1st Int'l Workshop on Mobile Computing Systems and Applications, Santa Cruz, pp. 85–90 (1994)
11. Pascoe, J., Ryan, N.S., Morse, D.R.: Human-Computer-Giraffe Interaction: HCI in the Field. In: Proc. 1st Workshop on Human Computer Interaction with Mobile Devices, GIST Technical Report G98-1 (1998)
12. McCarthy, J., Hayes, P.J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence. In: Machine Intelligence, pp. 463–502. Edinburgh University Press (1969)
13. Meissen, U., Pfennigschmidt, S., Voisard, A., Wahnfried, T.: Context- and Situation-Awareness in Information Logistics. In: Lindner, W., Fischer, F., Türker, C., Tzitzikas, Y., Vakali, A.I. (eds.) EDBT 2004. LNCS, vol. 3268, pp. 335–344. Springer, Heidelberg (2004)
14. Chen, H.L.: An Intelligent Broker Architecture for Pervasive Context-Aware Systems. PhD Thesis, University of Maryland (2004)
15. Baldauf, M., Dustdar, S., Rosenberg, F.: A Survey on Context-aware systems. Int'l J. of Ad Hoc and Ubiquitous Computing 2(4), 263–277 (2007)
16. Hipp, M., Mutschler, B., Reichert, M.: On the Context-aware, Personalized Delivery of Process Information: Viewpoints, Problems, and Requirements. In: Proc. 6th Int'l Conf. on Availability, Reliability and Security (ARES 2011), Vienna, pp. 390–397 (2011)
17. Klemke, R.: Modelling Context in Information Brokering Processes. PhD Thesis, RWTH Aachen University (2002)
18. Strang, T., Linnhoff-Popien, C.: A Context Modeling Survey. In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The 6th Int'l Conf. on Ubiquitous Computing, Nottingham (2004)
19. Winter, R.: Enterprise-wide Information Logistics: Conceptual Foundations, Technology Enablers, and Management Challenges. In: Proc. 30th Int'l Conf. on Information Technology Interfaces (ITI 2008), Dubrovnik, pp. 41–50 (2008)
20. Heuwinkel, K., Deiters, W.: Information logistics, e-healthcare and trust. In: Proc. Int'l Conf. e-Society (IADIS 2003), Lisbon, vol. 2, pp. 791–794 (2003)
21. Lundqvist, M., Sandkuhl, K., Levashova, T., Smirnov, A.: Context-Driven Information Demand Analysis in Information Logistics and Decision Support Practices. In: Proc. 1st Int'l Workshop on Contexts and Ontologies: Theory, Practice and Applications (2005)
22. Salber, D., Dey, A.K., Abowd, G.D.: The Context Toolkit: Aiding the Development of Context-Enabled Applications. In: Proc. SIGCHI Conf. on Human Factors in Computing Systems: the CHI is the Limit (CHI 1999), Pittsburgh, pp. 434–441 (1999)
23. Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J., Retschitzegger, W.: Context-Awareness on Mobile Devices - the Hydrogen Approach. In: Proc. 36th Annual Hawaii Int'l Conf. on System Sciences (HICSS 2003), vol. 9, pp. 292–301 (2003)
24. Gross, T., Klemke, R.: Context Modelling for Information Retrieval - Requirements and Approaches. IADIS Int'l J. on WWW/Internet 1(1), 29–42 (2003)
25. Thönssen, B., Wolff, D.: A broader view on Context Models to support Business Process Agility. In: Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications, pp. 337–358. IGI Global (2010)
26. Kaltz, J.W., Ziegler, J., Lohmann, S.: Context-aware Web Engineering: Modeling and Applications. Revue d'Intelligence Artificielle 19(3), 439–458 (2005)