Towards Integration of Uncertain Sensor Data into Context-aware Workflows

Matthias Wieland[1], Uwe-Philipp Käppeler[2],
Paul Levi[2], Frank Leymann[1], Daniela Nicklas[3]

[1]Institute of Architecture of Application Systems, Universtät Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
lastname@iaas.uni-stuttgart.de

[2]Institute of Parallel and Distributed Systems, Universität Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

[3]Department of Computer Science, Carl von Ossietzky Universität Oldenburg
Escherweg 2, 26121 Oldenburg, Germany
dnicklas@acm.org

Abstract: The integration and usage of uncertain sensor data in workflows is a difficult problem. In this paper we describe these difficulties which result from the combination of very distinct areas. On the one hand, applications from area of measurement engineering manage sensors that capture data and annotate the data with technical meta data. On the other hand, context-aware workflows from the BPM area place high level requirements for the quality of context data that is derived from the sensor data. Between those two areas exists a gap that has to be closed by a context management and mediation system, supporting the handling of Quality of Context (QoC). To achieve this the paper presents an QoC aware architecture based on an extension of the existing Nexus Platform and a first approach for matching the workflow requirements with the sensor annotations.

1 Introduction

Workflow Management Systems, which monitor and execute business processes, are often triggered and controlled by external data events, e.g., the arrival of an order, the result of a computation, or, as in the example in figure 1 on the left side (Loan Approval), the amount of money at the stake in a loan approval process. The workflow execution assumes that this external data is completely accurate and there are no inaccuracies because of the quality of the data used. Based on such external data, well-defined decisions are taken within the workflow execution.

If the processes modeled and executed by the workflow management system depend on dynamic states of the physical world (like in context-aware workflows [WKNL07]), external data (the context information) is often gathered by sensor systems. Almost all sensors

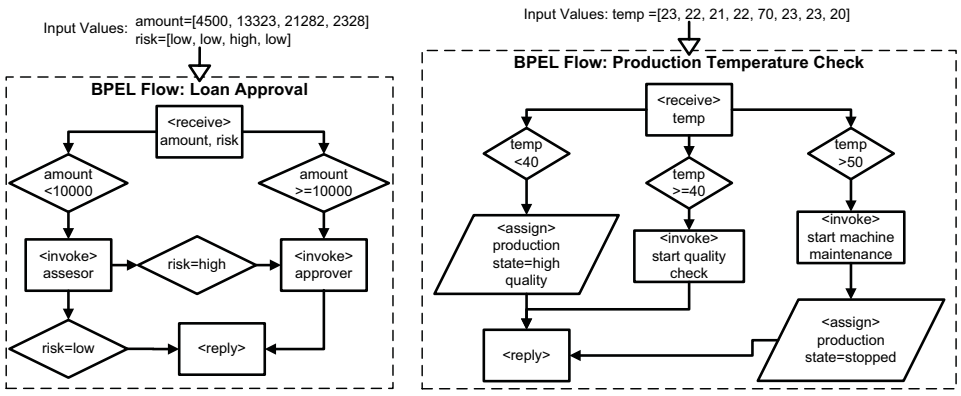

Figure 1: Example workflows: “Loan Approval” handling exact data, “Production Temperature Control” handling uncertain sensor data

produce data that is—to a certain degree—inaccurate and noisy, which leads to aberrant measurements. To assess these inaccuracies, context information has to be annotated with meta data about the Quality of Context (QoC). This quality meta data can then be used by workflow management systems for a more suitable treatment of the external data within the workflow decisions. The context-aware workflow shown in figure 1 on the right side (Production Temperature Check) is used as example throughout the paper. The following problem may occur if QoC is not considered: if an aberrant measurement (e.g., the fifth input value: 70°C) is received by the workflow, the right part of the workflow is be activated and ”start machine maintenance“ is invoked. This should be avoided because in that case the production would stop. This should only happen if there is a high certainty in the data this action is based on.

In this paper, we present a multi-level architecture that first gathers and processes uncertain sensor data, second matches the available data with the declared needs of a workflow execution (expressed by policies), and last executes quality of context-aware workflows on the top-most level. The straightforward solution would be to use the sensor data and QoC meta data directly in the workflow. However, this is not a practical solution because workflows are typically modeled by domain experts and not by technical experts. Despite the fact that sensor quality descriptions are often complex and hard to understand, domain experts would prefer to use simple percentage values for the attributes they are interested in (e.g., actuality, correctness, ...), or even only want to specify an overall required minimum quality for the used context data in the workflow.

By Web Service conventions, non-functional requirements are typically expressed by the service requester using policies. In addition, the service provider expresses its capabilities with policies. Then a middleware, e.g., an enterprise service bus intersects both policies and calculates the effective policy. If both partners (the workflow and service) agree on that effective policy, this is like a contract between them and every following interaction has to adhere to that contract. We adapt this general approach for our solution.

As we see, there is a gap between the technical sensor meta data description and the policy

requirements of the workflows. This gap has to be bridged by a system matching both and calculating if the sensor data fulfills the requirements. The contribution of our paper is to describe this problem in detail and point out the steps towards a solution of that problem based on an extension of the existing Nexus Platform, and by using the well established workflow modeling and execution language BPEL (Business Process Execution Language).

The remainder of the paper is organized as follows: in Section 2, we describe related work and the foundations this paper builds on. Section 3 presents the architecture of our proposed solution. Sections 4–6 describe the three layers of the architecture in more detail. Section 7 summarizes and concludes the paper.

2 Foundations and Related Work

Quality of Context (QoC), its modeling, usage, and relevance for context-aware applications, has been already addressed by many researchers (e.g., [KH05, TB03]). The definition of context quality requirements presented in this paper is based on that related work. However, to the best of our knowledge, we are the first to show how such requirements can be expressed as policies to be used by Web Services and workflows. In the area of mediating ambiguous context, a mediation subsystem was introduced in [DMAC02]. This system could be used as an improvement in the implementation of the filter operator described in Section 6.

A BPEL workflow is a recursive aggregation model for Web Services. Thus, the BPEL workflow itself is accessible as Web Service from externally. Internally, it orchestrates different Web Services in order to perform its work. Hence, all quality handling in workflows is based on the Web Service standards. To describe non-functional properties of a Web Service, the WS-Policy [W3C07b] standard is used. A policy is a collection of *alternatives* that are composed out of a set of different *assertions*. WS-PolicyAttachment [W3C07a] defines how to attach policies to entities in a Web Service based system.

To include the handling of quality in workflows, it is necessary to process the quality of the workflows' input. Here, we focus on information based on measurements. Therefore, the quality of the data is mostly given by the physical restrictions of the according sensors. Erroneous measurements and sensors themselves can be described using *SensorML* and *Observations and Measurements*. Both standards are part of the *Sensor Web Enablement* [BPRD07], which describes an architecture and protocols that define an access to sensors and measurements via network communication. In [KDKK05], we adopted these protocols for the Augmented World Model Language (AWML) to be used within the Nexus Platform [NGS+01]. Metrics for data quality often rely on probabilities as described in [CCX08], which is necessary for example to reason situations based on measurements and context information. For the handling of uncertainty in workflows, we extend the description of uncertainty to several domains of degradation.

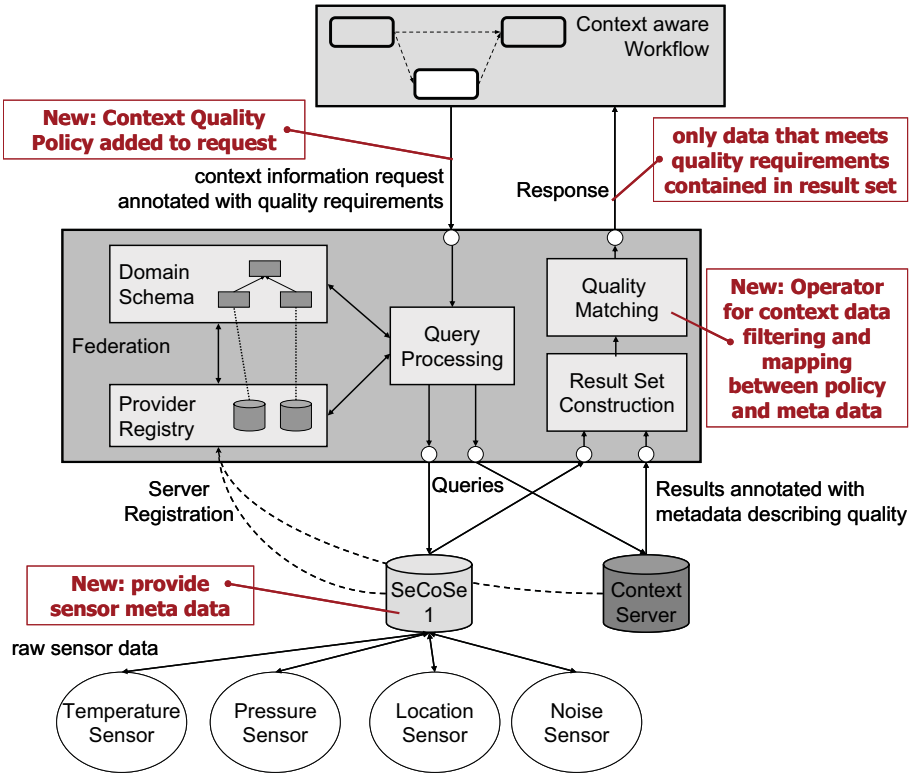

Figure 2: Extensions needed in the Nexus Platform for handling uncertainty

3 Architecture

The goal of the Nexus context management platform is to integrate various local context models to a common view for applications. The Nexus project [SFB] developed the platform over the last years. It uses a federation approach and bases on the so-called Augmented World Model (AWM) that serves as a common, yet extensible integration schema [NM04]. The AWM is object-based and covers four different types of context information: geographical context (map data), dynamic context (sensor data), information context (documents and virtual information), and technical context (sensors, networks, devices, etc.). Figure 2 shows the architecture of the Nexus platform and the extensions necessary to handle QoC for context-aware workflows. In the following, we shortly describe the three layers of the architecture, and what functionalities have to be added in order to handle QoC requirements and matchmaking.

The lowest layer is the data layer formed by the context servers. Different types of context servers are available for optimized management of diverse types of context information [GBH+05] (dynamic mobile objects, static objects and sensor data). On this layer meta data about the quality of the sensed information has to be added. This is described in

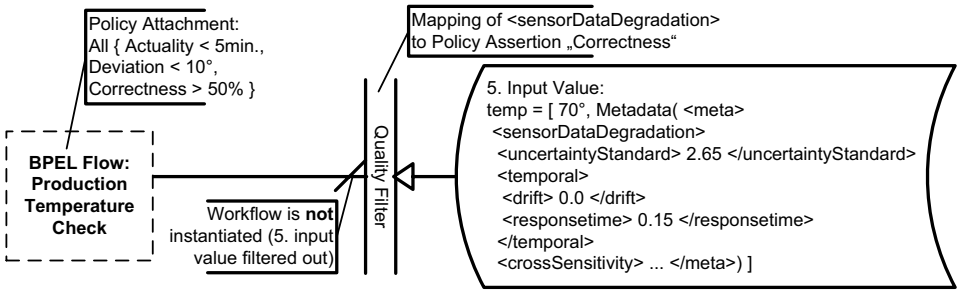

Figure 3: “Temperature Control” workflow using the extended Nexus Platform

Section 4 in more detail.

The middle layer is the federation layer. This layer forms a uniform interface for all context servers and manages the common data schemas and the extended *Domain Schemas*. It also manages the different available context servers and their service areas in a *Provider Registry*. Incoming queries are processed by the *Query Processing* component. The queries are forwarded to all context servers that could have context data available for the area of the query. The results are combined together in one result set by the *Result Set Construction* component. After that, the newly introduced *Quality Matching* component has to filter the context data for the required quality by removing all entries in the result set that are not sufficient for the quality requirements. This is described in Section 6 in more detail.

The highest layer consists of the context-aware applications. They can be implemented as normal applications or they can be modeled as Context-aware Workflows (CaW). In this paper, we focus on the second type: CaW based applications. To cope with QoC, the context queries in the CaWs are annotated with a Context Quality Policy (CQP). If the workflow engine executes a CaW it attaches the CQP to all requests in form of Policy Attachments. Therefore, the federation is enabled to filter all context information in order that all returned data meets the required CQP. How a CQP is expressed is described in Section 5.

After describing the extensions needed for the Nexus architecture, figure 3 shows the effects of the extensions on the execution of the example workflow: The fifth input value (70C) is filtered out hence the value for correctness is below the required 50%. The correctness is calculated in the example by `1-<uncertaintyStandard>/5` of the standard deviation, which can be obtained by comparing several sensors [KBZ+08]. In this example correctness is compared to the calculated value of (1-2.65/5)=0,47 or 47% , the workflow is not activated and “start machine maintenance” is not invoked only because of an erroneous measurement. By that unnecessary stops of production are avoided.

4 Data Layer: Degradation of Sensor Data

To provide environmental models for context-aware applications it is not only necessary to continually extend the models and integrate more and more data. The models also need to adapt to the current state of the real world. For both cases, it is necessary to integrate sensor data into the environmental models. However, when integrating scalar sensor data or complex measurements, we have the problem of noise in each measurement. Therefore, it is necessary to use sensor data fusion to reduce inconsistencies, systematic errors and noise of measurements for a reliable sensor data integration [KKN+05]. In addition, context-aware applications and their designers have to consider inconsistencies and erroneous information in the environmental models. To support the processing of information from these models, providers can collect and determine grades of quality in several aspects to objects and their attributes.

4.1 Data Fusion

Errors in sensor data result from systematic errors and stochastic errors [Hof07, Die91]. The stochastic error can be reduced by sensor fusion combining several measurements to one piece of information. This can be done by measuring the same phenomenon repeatedly over time, or using many sensors measuring the same information, which is the only possibility when observing dynamic phenomena. The more measurements are combined to one piece of information, the more it is possible to reduce noise and to increase accuracy.

We evaluated procedures that combine a multitude of measurements to a single result that can be integrated in the shared context model in. The statistically optimized procedure based on ratings of the participating agents is enhanced using normalized weighted arithmetic mean [KBZ+08] which prevents the system from running into singularities caused by the positive feedback from the ratings. The singular behavior occurs when one of the sensors used for the data fusion incidentally is rated very good several times consecutive. This agent gets rated better and better and in the end all negotiations lead to the assumption of the measurement from this agent as the result of the data fusion and the system never recovers from this state. These problems make it impossible to use the statistically optimized algorithm in an open and worldwide system where it is not possible to reset all sensor ratings repeatedly during runtime and where most of the time only a few measurements are provided for a data fusion. Normalizing the ratings of each sensor prevents the system from singular behaviors and it allows combining measurements of different sensors with different standard deviations.

The normalized weighted arithmetic mean method is combined with an additional pre-processing based on fuzzy clustering that detects aberrant measurements which can be excluded from further processing [BKZ+08].

4.2 Meta Data Description and Quality Domains

For a context-aware application that has to handle information of different qualities, it is necessary to obtain ratings concerning quality aspects like consistency and accuracy. For defining the degradation that results from the accuracy of sensors we defined meta data elements [HKNS05] that describe different domains of degradation [Käp08]:

- *Empirical Uncertainty*: This domain describes a relative, absolute or standard deviation, probability distribution or accuracy.
- *Cross sensitivity*: This domain lists errors that result from environmental conditions of a sensor. Each condition (e.g., temperature) is listed with the according relative or absolute error.
- *Processing*: Errors resulting from data processing such as digitalization of an analogue signal, amplification or linearization are listed in this domain.
- *Reliability*: In the reliability domain, timestamps and details to the last maintenance and calibration can be stored together with a resulting error from abrasion in the current state of the sensor.
- *Temporal aspects*: In the domain of temporal aspects details to attributes of the sensor concerning drift, hysteresis, response time, repeatability and sampling rate can be listed.
- *Resolution*: The resolution meta data can contain the upper and lower bound of the range of the used sensor, and its quantification and discrimination.

Each application processing data of environmental models benefits from quality information. However, in an open system the data providers cannot be obliged to offer this meta data. Therefore, all the above meta data elements are optional.

5 Application Layer: Quality Requirements of Workflows

The Context-aware Workflows (CaW) are modeled by business experts or domain experts, e.g. an engineer in a factory. Because of that the following *general requirements* are requested for the *expression of the quality assertions* in a CQP.

- The quality assertions for *discrete sensor values* have to be easy to understand, that means not technical, clearly defined and measured by one numerical value with unit if possible.
- Quality assertions for *aggregated context information* that is derived out of different sensor measurements have to be provided.

- Finally, and most important, it has to be possible to express the context quality assertions not only absolutely, but also *relatively with a percentage value*. For expression of relative quality in percentage a *allocation base* has to be specified for all assertion types. This is described in detail in the thesis [Wei08].

Based on those requirements CQP was developed as a WS-Policy based language. CQP consists of following assertion types:

- *Actuality*: Temporal proximity between the usage of a context information and the point in time when it was captured by a sensor. The context provider must provide a time stamp meta data attribute in order to calculate the actuality. Furthermore, the clocks between the context provider and the middleware must be synchronized. Actuality has the unit s (seconds) and can be expressed absolute or relative. A lower actuality results always in a decrease of QoC.
- *Accuracy*: Describes how exact the context information is representing the reality. It is measured in the unit of the sensor, e.g. for a GPS device in m (meters) or for a temperature sensor in degree and can be expressed absolute or relative. For higher level aggregated context information accuracy is not applicable.
- *Resolution*: Describes the quality of digitalization of analog signals. It can also be used to specify the granularity of context information. That means if e.g. data about restaurants is listed in the resolution of city scale or per street scale. Because of that the unit of resolution can be metric or symbolic. If symbols are used for identifying granularity levels, they have to be defined manually in the CQP schema (like the allocation base). A higher resolution results in a higher QoC.
- *Correctness*: Describes the probability that the measured value is correct. This probability must be calculated by the context provider with statistical methods about the used sensors and their standard derivation. Correctness is a percentage value only and has no unit.
- *Deduction history*: Describes the quality of aggregated context information. It is characterized by the number of context information values that are used for the aggregation. The higher this number is the lower is the QoC. This value has to be given by the context provider after performing the aggregation because it cannot be calculated thereafter.
- *Reliability, Trust*: Describes the reliability of the context provider that manages the context information. This value is set by the federation or the application itself. It is needed to describe the influence of the context provider on the QoC. For example if a context provider often provides incorrect information the federation decreases its reliability. It is a value in percentage and has no unit.

An example for a CQP expressing an actuality assertion that requests - *The actuality should be >=90%* - is shown in listing 1. Here the basis <qoc:BasicQualityClasses> allows to give relative assertions in percentage. Without such basis all assertions would have to be

Listing 1: QoC Policy with actuality assertion

```
<wsp:Policy>
 <wsp:ExactlyOne>
  <wsp:All>
    <qoc:BasicQualityClasses>
        <BasicQualityClass>
            <Name>relative actuality</Name>
            <Equipartition>
                <PhysicalUnit>s</PhysicalUnit>
                <LowerLimit>115</LowerLimit>
                <UpperLimit>5</UpperLimit>
            </Equipartition>
        </BasicQualityClass>
    </qoc:BasicQualityClasses>
    <wsp:ContextRestriction>
        <tns:Actuality>
            <Clock>http://exampleclock.org</Clock>
            <PercentagedQualification>
                <minValue>90</minValue>
            </PercentagedQualification>
        </tns:Actuality>
    </wsp:ContextRestriction>
  </wsp:All>
 </wsp:ExactlyOne>
</wsp:Policy>
```

given in absolute values. In this example the basis for limits of the actuality is given with 5 seconds and 115 seconds, which is the maximum that the application could handle properly, a currently requested actuality of 90% would define a period of time of 16 seconds between the measurements timestamp and the processing of the information.

6 Federation Layer: Matchmaking

The matchmaking can be done with any algorithm that is plugged into the presented architecture (in the Quality Matching Component, see Figure 2). In this section, we present a first algorithm that is easy to use but which does not use artificial intelligence or automatic learning. But if an interesting algorithm for the matchmaking task is invented it could be used. The aim of the currently running Nexus Quality Project is to provide a reference framework for context quality. Part of that framework will be to provide such matchmaking algorithms. Because of that we focus here on an easy solution that can be replaced later by something more sophisticated.

To perform matchmaking between the sensor meta data and the workflow policies, the federation layer has to understand both. Furthermore, it has to know how to map one to the other correctly. Our standard mapping of information from the measurements meta data to workflow policies is a predefined, fixed implementation.

The *fixed implementation* of this translation is simple and efficient. An example for this direct mapping is shown in Figure 3. Here <sensorDataDegradation> from the sensor meta data is mapped to the Policy Assertion Correctness. All other assertions are also mapped to one or more meta data fields. After the mapping the values can be directly compared with each other. If the comparison produces a false result the context object is filtered out. In our example the Policy Assertion Correctness $> 50\%$ is compared with <sensorDataDegradation> <uncertaintyStandard> which has the value of 2.65. Using the formula given in Section 3 a correctness value of 47% can be calculated and compared to the given policy. In this case the comparison produces a false result ($47\%>50\%$=false) and the object is filtered.

However, this fixed implementation prevents workflows from using complex sensor data fusion that relies on different sensor's qualities. Therefore, we propose an additional method for implementation of the quality matching operator that is easy to realize. We call it *configurable translation*. Here, an application designer or even workflows could automatically configure individual translations of the quality ratings given in the meta data to policies. A configuration for the operator should be defined either by complete formulas, key words according to the meta data, and policies, or by parameters for the predefined mapping. The configuration by parameters would provide a possibility to rate different domains of quality with different weights easily.

7 Conclusion

In this paper, we showed that it is important to cope with the quality of context information. We presented an extended architecture based on the Nexus Platform to handle uncertain sensor data. Furthermore, we described how sensor data can be annotated with meta data as basis for quality handling and we showed how applications based on context-aware workflows can make quality restrictions for requested context data. For the matchmaking, we proposed a fixed implementation that is easy to use and a configurable translation that is more flexible. Future work is to find algorithms for matchmaking based on artificial intelligence or automatic learning and implement the quality filter and matching operators in the Nexus federation. As a final result of the work we found that context-aware systems using QoC techniques get more complex and expensive but also more error resilient which could balance this out in future.

References

[BKZ+08] Ruben Benkmann, Uwe-Philipp Käppeler, Oliver Zweigle, Reinhard Lafrenz, and Paul Levi. Resolving Inconsistencies using Multi-agent Sensor Systems. In *Proceedings of the 8th Conference on Autonomous Robot Systems and Competition: Robotica 08;*. Universidade de Aveiro, 2008.

[BPRD07] Mike Botts, George Percivall, Carl Reed, and John Davidson. OGC Sensor Web Enablement: Overview And High Level Architecture, 2007. OGC White Paper.

[CCX08] Reynold Cheng, Jinchuan Chen, and Xike Xie. Cleaning uncertain data with quality guarantees. *PVLDB*, 1(1):722–735, 2008.

[Die91] Cornelius Frank Dietrich. *Uncertainty, Calibration and Probability, The Statistics of Scientific and Industrial Measurement*. Adam Hilger Imprint, 2nd edition, 1991.

[DMAC02] A. Dey, J. Mankoff, G. Abowd, and S. Carter. Distributed mediation of ambiguous context in aware environments. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 121–130. ACM Press New York, NY, USA, 2002.

[GBH+05] Matthias Großmann, Martin Bauer, Nicola Hönle, Uwe-Philipp Käppeler, Daniela Nicklas, and Thomas Schwarz. Efficiently Managing Context Information for Large-Scale Scenarios. In *Proc. of the Third IEEE Intl. Conf. on Pervasive Computing and Communications*, 2005.

[HKNS05] Nicola Hönle, Uwe-Philipp Käppeler, Daniela Nicklas, and Thomas Schwarz. Benefits Of Integrating Meta Data Into A Context Model. In *Proceedings of 2nd IEEE PerCom Workshop on Context Modeling and Reasoning (CoMoRea)*, 2005.

[Hof07] Jörg Hoffmann. *Handbuch der Messtechnik*. Carl Hanser Verlag Mnchen, 3rd edition, 2007.

[Käp08] Uwe-Philipp Käppeler. Modellierung der Degradierung von skalaren Sensordaten in ContextServer und SensorContextServer. SFB 627 Report Nr. 2008/01, Universität Stuttgart, Germany, 2008. *(In German)*.

[KBZ+08] Uwe-Philipp Käppeler, Ruben Benkmann, Oliver Zweigle, Reinhard Lafrenz, and Paul Levi. Resolving Inconsistencies in Shared Context Models using Multiagent Systems. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems: IAS-10; Germany*. Springer-Verlag, 2008.

[KDKK05] Uwe-Philipp Käppeler, Dominique Dudkowski, Georg Kindermann, and Darko Klinec. Modellierung von Sensoren und Sensordaten in der Nexus-Plattform. SFB 627 Report No. 2005/03, Universität Stuttgart, Germany, 2005. *(In German)*.

[KH05] Michael Krause and Iris Hochstatter. Challenges in Modelling and Using Quality of Context (QoC). In *Mobility Aware Technologies and Applications*, volume Volume 3744/2005 of *LNCS*. Springer, 2005.

[KKN+05] Uwe-Philipp Käppeler, Georg Kindermann, Daniela Nicklas, Nicola Hönle, and Dominique Dudkowski. Shared Dynamic Context Models: Benefits for Advanced Sensor Data Fusion for Autonomous Robots. In *Proceedings of Artificial Intelligence and Applications;*. IASTED, 2005.

[NGS+01] Daniela Nicklas, Matthias Großmann, Thomas Schwarz, Steffen Volz, and Bernhard Mitschang. A Model-Based, Open Architecture for Mobile, Spatially Aware Applications. In *Proceedings of the 7th International Symposium on Spatial and Temporal Databases*. Springer-Verlag, 2001.

[NM04] Daniela Nicklas and Bernhard Mitschang. On Building Location Aware Applications Using an Open Platform Based on he NEXUS Augmented World Model. *Software and Systems Modeling*, 3(4), 2004.

[SFB] SFB627. Nexus project web site. `http://www.nexus.uni-stuttgart.de/index.en.html`.

[TB03] Michael Schiffers Thomas Buchholz, Axel Kpper. Quality of Context Information: What It Is And Why We Need It. http://www.nm.ifi.lmu.de/pub/Publikationen/bks03/, 2003.

[W3C07a] W3C. Web Services Policy 1.5 - Attachment, W3C Recommendation, 2007.

[W3C07b] W3C. Web Services Policy 1.5 - Framework, W3C Recommendation, 2007.

[Wei08] Steffen Weik. Entwicklung einer Sprache zur Beschreibung von Qualitätseigenschaften kontextbezogener Dienste. Diplomarbeit, Universität Stuttgart, Germany, 2008. *(In German)*.

[WKNL07] Matthias Wieland, Oliver Kopp, Daniela Nicklas, and Frank Leymann. Towards Context-Aware Workflows. In *CAiSE07 Proc. of the Workshops and Doctoral Consortium Vol.2*. Tapir Acasemic Press, 2007.