2009-06-01

A Context Quality Model to Support Transparent Reasoning with Uncertain Context

Susan MCKeever
Technological University Dublin, susan.mckeever@tudublin.ie

Juan Ye
University of St. Andrews

Lorcan Coyle
University of Limerick

Simon Dobson
University of St. Andrews

Recommended Citation

McKeever S, Ye, J, Coyle L, Dobson, S (2009) A Context Quality Model to Support Transparent Reasoning with Uncertain Context, Proceedings of QuaCon 2009, June 2009, Stuttgart, Germany

A Context Quality Model to Support Transparent Reasoning with Uncertain Context *

Susan McKeever, Juan Ye, Lorcan Coyle, and Simon Dobson

susan.mckeever@ucd.ie

Abstract. Much research on context quality in context-aware systems divides into two strands: (1) the qualitative identification of quality measures and (2) the use of uncertain reasoning techniques. In this paper, we combine these two strands, exploring the problem of how to identify and propagate quality through the different context layers in order to support the context reasoning process. We present a generalised, structured context quality model that supports aggregation of quality from sensor up to situation level. Our model supports reasoning processes that explicitly aggregate context quality, by enabling the identification and quantification of appropriate quality parameters. We demonstrate the efficacy of our model using an experimental sensor data set, gaining a significant improvement in situation recognition for our voting based reasoning algorithm.

1 Introduction

The information used by context-aware systems to recognise different contexts is often imperfect. Sensor data is prone to noise, sensor failure and network disruptions. Users actions can contribute to degradation of information quality, such as the failure of users to carry their locator tags. Further uncertainty can be introduced in the reasoning process, such as the use of fuzzy functions to quantify vague context or difficulty in defining accurate inference rules [14]. Existing work in the area of context quality focuses on two main areas: (1) The *qualitative* identification of context quality parameters, often as part of a context modelling exercise, such as the work done by [4,6,5]; and (2) the *quantitative* use of reasoning techniques that incorporate context uncertainty such as Bayesian networks [10] and fuzzy logic [7].

The qualitative work provides a useful vocabulary for identifying and modelling context quality. However, such measures are usually associated with 'context', without specification of quality for each *layer* of context. Quality issues for low level sensor data are different to those at higher levels of context and a context quality model must reflect this [9]. Also, the *aggregation* of quality across the layers must be addressed in order to produce a meaningful and useable indicator of context quality to applications. This aggregation will support

* This work is partially supported by Enterprise Ireland under grant number CFTD 2005 INF 217a, and by Science Foundation Ireland under grant numbers 07/CE/I1147 and 04/RPI/1544

reasoning schemes that can propagate uncertainty from sensor level upwards. For example, Dempster Shafer [13] or voting algorithms [2] for context reasoning can incorporate explicit quantification of uncertainty of context sources.

This paper presents a UML-based structured model of context quality for each layer of context. We also include an aggregation model that contains a general set of quality measures and their propagation across context layers. Designers of context-aware systems can use our combined models to (1) identify and model context quality issues and (2) to support the specification of quality aggregation. In particular, context-aware systems using transparent reasoning techniques that aggregate quality from sensors upward will benefit from our modelling approach. We demonstrate our work by generating quality parameters for an experimental dataset. We incorporate these quality parameters into a voting-based reasoning algorithm. Our results show that situation recognition significantly improves with the inclusion of our modelled context quality than when quality is not used.

This remainder of this paper is organised as follows: Section 2 describes related work by other researchers; Section 3 details our structured quality and aggregation models and their relevance to context reasoning schemes; In Section 4, we demonstrate and critique our model with an experimental dataset. Finally, in Section 5, we conclude our work and define our future research direction.

2 Related work

Previous work on modelling context quality provides various well documented parameters for context quality, such as context *confidence* [11,3,10] to indicate probability of correctness and *freshness* [3,4,8,1] to indicate the degradation of information over time. Lower level sensor quality measures such as *precision, accuracy* and *resolution* [4,1] are used to define sensor data issues. Such work provides useful semantics for exploring the nature of context quality issues. Other modelling approaches include placeholders for quality parameters within structured models of context. For example, Henricksen and Indulska's [6] Object Role Modelling context model associates context facts with zero or more quality parameters and associated metrics. Similarly, Gu *et al.* [5] describe a context model that includes a quality ontology with specific parameters and metrics. They include a set of commonly used parameters in their ontology. Both of these models use similar modelling constructs for quality. However, sensor and situation quality parameters are not separately identified.

The modelling approaches described do not model or aggregate quality parameters at each layer of context. We address this as follows: (1) We provide a structured (UML) extendable context quality model that includes quality parameters for sensor, abstracted context and situations, as illustrated in Figures 1 and 2. (2) We provide an aggregation model that aggregates quality across each layer of context, as shown in Figure 3. To illustrate our work, we provide a demonstration of how context quality can be used to support the reasoning process, as explained in Section 4.

3 Our Approach to Modelling Context Quality

Context-aware applications are usually decoupled from the intricacies of lower-level sensor data, Instead, they liaise with higher-level human-understandable situations [14], such as a user 'preparing breakfast' at home. For such applications, a usable expression of quality at situation level will be useful, rather than having to interpret lower level measures such as sensor precision. We approach our context quality modelling work with the following principles in mind: (1) Quality issues at each layer of context (sensor, abstracted context, situation) are different [9], requiring parameters to be specified for each layer; (2) Known aggregations between quality parameters along the layers should be shown to assist in quantifying quality. However, prescription of explicit aggregation formulae is avoided to allow for system-specific calculations; (3) A model of context quality should be extendable to allow for system-specific requirements; (4) A standard modelling technique should be used to maximise use of the structured model. We incorporate these principles into our structured (Figures 1 and 2) and aggregation models (Figure 3). These models are general and flexible enough to suit different types of sensors and context and their associated quality issues. We use UML for our structured model.

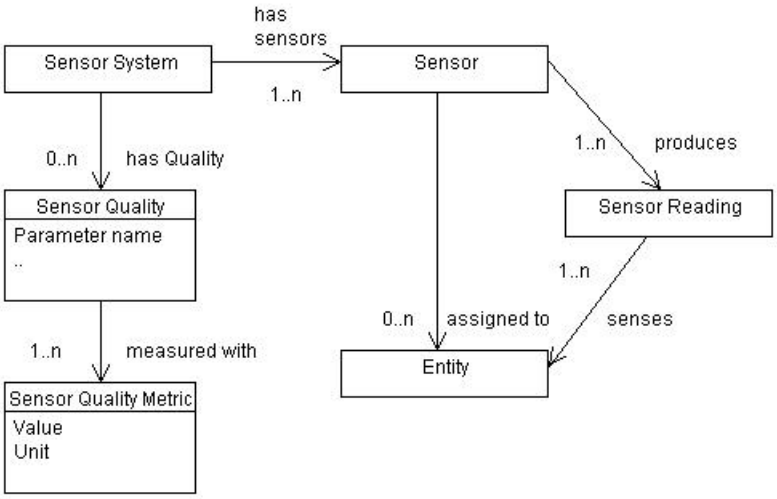

Fig. 1. Sensor quality modelling in UML.

3.1 Modelling sensor quality

In our UML model, sensors represent any physical (e.g. Ubisense [1]) or virtual data source (e.g. tracked user calendar) that provides dynamic information about an entity, such as location of a user. Sensors are heterogeneous. Therefore, it is challenging to provide a set of quality parameters that applies across all sensors. Our structured model is generic and flexible, describing placeholders for modelling quality without prescribing what parameters to model. Our sensor quality model as shown in Figure 1 shows zero or more quality parameters for each sensor. For each quality parameter in our sensor class, one or more metrics may be stored. Each sensor system may have one or more sensors, and each sensor tracks one or more entities. A quality parameter is an instantiated sensor quality class and (at least one) sensor quality metric class. E.g., our structured model of our in-house Ubisense system includes a quality parameter of '`precisionx`' (for x-axis), with a metric value of `1.65` and unit of `metres`.

Our aggregation model (Figure 3) describes aggregation of quality from sensor level to situation. We specify a base set of sensor quality parameters that are used to determine quality at higher levels of context abstraction. This set may be reduced or expanded according to the individual sensor system. *Precision* indicates the range within which a sensor reading or part of a sensor reading is believed to be true; *Accuracy* indicates the error rate or frequency of correctness of sensor readings, for a given precision*; Frequency* of sensor readings can be used to support calculation of a freshness measure of abstracted sensor readings. Other commonly used parameters such as resolution and coverage are not included here as we do not use them to determine higher level quality, but they can be instantiated in the structured model for system specific scenarios.

3.2 Modelling abstracted context quality

As shown in Figure 2, sensor readings are abstracted to more meaningful context by passing one or more sensor readings through a static mapping or filter. e.g. a Ubisense coordinate of *(12.3, 32.4, 34.1, ID34)* may be mapped via a building map to 'John's desk'. Abstracted context is modelled as a UML association class that describes the relationship (e.g. has location, has temperature) between an entity class and a context filter class at a particular point in time. A context event is an instantiation of the association class for an instantiated entity class and context filter class for a particular time t. A context event may also be modelled as an association between two entities, such as 'John located near Susan'. Context confidence is included for each context event. It will be derived from zero or more context event quality parameters as used for the system in question. Our structured model therefore allows for any type of context relationship and associated quality to be modelled.

In our aggregation model (Figure 3), we capture context event quality in a base set of quality parameters; these quantify the imperfections of vague context

[1] Ubisense is a networked location system : `www.ubisense.net`

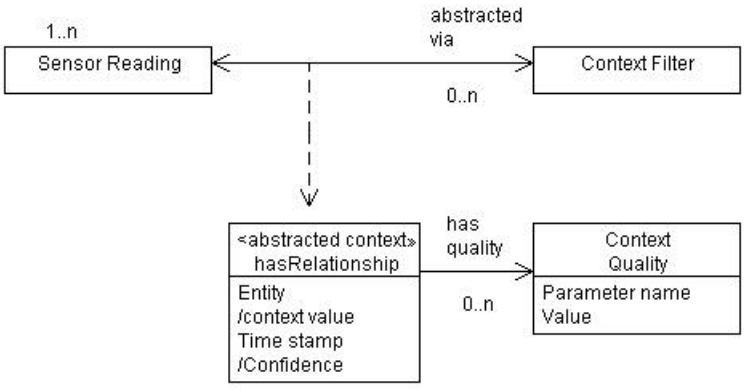

Fig. 2. Context event quality modelling in UML

(fuzzy membership), erroneous or conflicting context (reliability), imprecision (precise membership) and out of dateness (freshness). These values are combined to produce a final context event *confidence* value. As shown in Figure 3, context event confidence is a function of the other context event quality parameter values, such as the product of their values as per [2] or as their averaged value. *Fuzzy membership* is used when fuzzy context filters are used in the abstraction process. In our experimental work, a computer activity sensor that tracks keyboard/mouse activity classifies its context values as 'active' or 'inactive'. We apply a fuzzy membership value using a linear fuzzy membership function for the context filter. For context filters with crisp boundaries such as location, *precision membership* of the bounded value is derived using the sensor reading precision. This captures the impact of general sensor precision for a particular context instance. For example, we use Ubisense precision to define a bounded area within which the true Ubisense coordinate should occur, as described in [9]. This area may intersect more than one location (e.g. desk), leading to quantifiable membership of each desk location. e.g. User John is 0.8 in Desk 1, and 0.2 at Desk 2. *Reliability* captures the error rate associated with a context event. The reliability can incorporate any sources of error, including user error and sensor system accuracy. It can be measured objectively by training or observation. *Freshness* indicates the extent to which time has eroded the credibility of the context event. It can be calculated in various ways, such as use of a decay function [9], or a valid lifetime, derived from sensor reading timestamp or frequency. We avoid prescribing explicit hard-coded formulae for freshness and other parameters as their calculation may differ from one scenario to another.

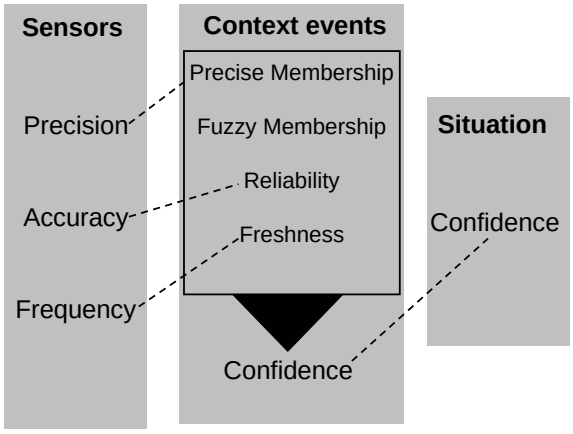

Fig. 3. Aggregation model: quality parameters and aggregations. The lines indicate how quality is aggregated. e.g accuracy at sensor level contributes to the reliability of a context event

3.3 Modelling situation quality

The provision of *situation confidence* allows adaptive applications to assess the risk associated with responding to a situation. Our structured model therefore explicitly includes at least one situation confidence parameter. Situation confidence is modelled as an attribute in a situation association class. The calculation of situation confidence will depend upon the reasoning scheme employed. For schemes that aggregate quality from sensor level upwards, situation confidence will be derived by fusing confidences of causal context events, as shown in our aggregation model (Figure 3).

3.4 Using Context Quality in the Reasoning Process

Reasoning with uncertain context typically involves reasoning schemes from the AI domain employed within context middleware, such as Bayesian networks [10], probabilistic logic [10], neural networks [11], Dempster Shafer [13], voting [2], and fuzzy logic [7]. The selection of one reasoning scheme over another will depend on a variety of factors such as: the availability of training data versus domain knowledge; the requirement for end users to understand the reasoning process; and the level of re-training required due to flux in the environment. Such reasoning schemes typically produce a quantified confidence in each of the possible situations. Applications can then use confidence to safeguard their adaptation strategy, such as accepting situations above a particular threshold value.

The extent to which sensor and context quality is *transparently* incorporated into the reasoning process depends on the reasoning mechanism involved.

Bayesian networks can be naively trained using a 'black box' approach. In this way, there is no transparent quantification or aggregation of sensor or context event quality. The degree of uncertainty appears as 'output' via the root node variable's probabilities. Alternatively, other finer grained reasoning mechanisms can specifically aggregate lower level sensor and context quality in a way that is transparent to the systems designer. Examples of such mechanisms are voting [2], Dempster Shafer [13] and fuzzy logic [10]. Uncertain reasoning processes that transparently aggregate quality of context up to situation level *need to identify, model and aggregate appropriate quality parameters* along each layer of context. Our model supports the identification of quality issues and quality aggregation as part of such transparent reasoning processes.

4 Demonstration of Model

We used our model to identify and aggregate quality parameters for a sensed data set. We wanted to determine whether the the use of our quality parameters improved the situation recognition rates when we used a transparent reasoning scheme based on voting. We contrasted two approaches to reasoning: : (1) Basic reasoning, where quality was not used, selecting the user activity situation that received the most 'votes' from its causal context events (2) The same voting algorithm but with quality parameters used to attentuate the contribution of each vote.

4.1 Approach

We collected a data set tracking a user in our office. We wanted to determine at any point in time whether a user was 'in' any one of three possible situations: 'busy' at their desk, on a 'break' or at a 'meeting'. To support this, we needed to know where the user was located, whether they were using their computer, and where they were scheduled to be at a meeting. We used three sensors in our dataset: (1) Ubisense location tag sensor that tracks the user's location (2) a computer activity sensor on the user's computer that monitors keyboard and mouse activity and (3) a calendar sensor that indicates whether a meeting is scheduled or not in the user's diary. For example, the situation of user 'busy ' is occurring when the user is at their desk, using their keyboard or mouse and no meeting is scheduled in their diary: 'user hasLocation *desk*', 'computer hasStatus *active*', and 'calendar hasSchedule *free*'. When each of these context events is detected, each contributes a vote towards the 'busy' situation. Situation checking occurs every 30 seconds. The user maintained a diary to track the actual ground truth situations.

When quality parameters are used, each vote is attenuated by the confidence value of its associated context event. In order to determine context event confidence, we needed to determine the underlying sensor and context event quality. We analysed data from the sensors and context events, using our aggregation model in Figure 3. as a basis for finding which quality parameters we should use.

Sensor	Sensor quality	Context event quality
Ubisense	PrecisionX (1.65 m), PrecisionY (1.11m), accuracy (0.8)	Precision membership, reliability (0.72)
Calendar	Precision (10 mins)	Precision membership, reliability (0.6)
Comp. Activity	None used	Fuzzy membership, reliability (0.95)

Table 1. Quality parameters used

Quality parameters for our experiment are shown in Table 1. Values for constant parameters such as reliability are shown. Parameters that are dynamically calculated from sensor readings, such as fuzzy membership, are listed without an accompanying value. The parameters and their values are derived as follows:

- For the Ubisense sensor, precisions (1.65 x-axis, 1.11 y-axis) and accuracy (0.8) were captured using training data where we gathered readings and compared actual location versus sensor readings. We used these precision numbers to generate a precision membership for 'has Location' context events, as described in Section 3.2. An accuracy of 0.8 for Ubisense was degraded at the context event level to a reliability of 0.72 because the user neglected to carry their tag during the data set collection 10% of the time.
- For the calendar sensor, we observed that over a period of a month, the user adhered to 22 out of 36 total meetings in her diary. Therefore, reliability of the calendar sensor is 0.6 based on 60% adherence to diary entries. Start and end times of attended meetings during this time were imprecise by an average of 10 minutes.
- For the computer activity sensor, no noise in sensor readings was observed. However, when abstracted to context event, we applied a fuzzy function to capture the gradual move from active status to inactive and vica versa. The function degraded linearly from fully active (fuzzy membership value of 1) if activity was detected in the last 30 seconds, reducing to 0 after 3 minutes of no activity.
- *Context confidence* for each event was calculated as the product of the context event quality parameter values, as also used by [2]. For example, if at a particular point in time t, a context event of 'user hasLocation desk' has a calculated precision membership of 0.9, and Ubisense reliability is 0.72, the overall confidence of the 'user hasLocation desk' context event at that point in time is 0.65.
- *Situation confidence* for each of the three user situations was calculated as the average of the context event confidences for that situation. For example, at time t, the context events relevant to the user 'busy' situation have the following context event confidences: 'user hasLocation *desk*, conf 0.65', 'calendar hasSchedule *free*, conf 0.9' and 'computer hasStatus *active*, conf 0.4'. Therefore, the confidence of the user 'busy' situation at this point in time

t is the average of the three underlying context events, 0.65. At time t, the situation with the highest confidence is deemed to be occurring.

4.2 Analysing our results

Situation recognition rates (see Table 4.2) were better when we used our quality model parameters: 90% of situations that could not be identified by the basic reasoning technique were identified when our quality parameters were included in the reasoning process. Basic reasoning failed when the sensor data was noisy (e.g. imprecise Ubisense reading) or user errors were encountered (e.g. user forgot to carry locator tag). The 'meeting' recognition rates were the same because the underlying sensor information was of good quality. I.e. The user always had a meeting scheduled when the meeting was on, the user remembered to wear their Ubisense locator tag to the meeting, and the sensor readings were accurate. Reasoning *with quality parameters* failed for 7% of situation checks for three reasons: (1) The ground truth was captured in a manual diary, with the user rounding up to minutes. This led to mismatches at situation start and end times between ground truth and situation detection. Annotation of ground truth with time in seconds will be important to avoid this problem in future experiments. (2) Some situation rules were inadequate. For example, when a user is reading at their desk without using their computer, the 'busy' at desk situation is not fulfilled because the computer activity sensor is inactive. The inactive computer activity sensor has a higher vote due because its context events are more reliable than the Ubisense related context events, so at 'break' was selected (3) The fuzzy membership of the activity context events led to the system *gradually* recognising that the computer was no longer active. In reality, the user finished the 'busy' situation in an abrupt manner as recorded in the ground truth diary. This requires some thought as to how specific situation transitions and associated application behaviour should work: are instantaneous changes versus gradual changes in situation or behaviour required? Such analysis will help to determine the appropriateness of using fuzzy and decay functions for underlying contexts.

Situation	**# of readings**	**basic reasoning: % identified**	**reasoning with quality: % identified**
User busy	720	52	96
User on break	199	75	85
User at meeting	53	91	92

Table 2. Situation recognition results: basic and quality reasoning

Our results highlighted a number of weaknesses that may be resolved by more a robust reasoning mechanism rather than changes to our quality model. At various points, it was difficult to identify the winning situation because the situation confidences were so close. e.g. How meaningful is it to choose situation

of *(busy, conf 0.45)* over (*break, conf 0.44*)? Therefore, an algorithm that can support greater convergence on situation confidence is desirable. It was also clear that conflict between the sensors was manifested by a lack of a clear winner on the voting process. However, the source of conflict is not identified by our voting algorithm. Finally, it was not obvious how to allocate the vote uncertainty. E.g. the contexts events from the calendar sensor had a reliability of 0.6, resulting in 0.4 of the vote being unallocated, which seems intuitively incorrect. We propose that these weaknesses may be addressed by using Dempster Shafer theory as our reasoning mechanism, as discussed in Section 5.

5 Conclusion and future work

In this paper, we presented a general context quality model. The structured model in UML provides a flexible and generalised way for designers of context-aware systems to incorporate context quality into their design process; the aggregation model identifies quality issues and their aggregation across context layers. We applied our model to an experimental data set, using a transparent voting algorithm to identify situations. Situation recognition results improved when we used quantified quality measures in our reasoning process.

Currently, we are integrating our quality model work with a more robust uncertainty reasoning algorithm based on Dempster Shafer theory (DST). DST is a mathematical theory of evidence that propagates uncertainty values [12], providing an indication of the quality of inference. We anticipate that use of DST will allow us to address the weaknesses of the voting algorithm identified in 4.2. Early results from this work using a sample of our data set results in situation recognition rates similar to the voting algorithm. However, DST also provides a conflict metric quantifying the extent of context disagreement during situation recognition. Analysis of this conflict may allow us to detect a number of scenarios. We noted that conflict levels were high when sensors are in conflict, rule problems occur or situations are in transition. This is valuable information that we hypothesise can improve our ability to reason with context, particularly in dynamic environments subject to changes in sensors and situations.

References

1. T. Buchholz, A. Kupper, and M. Schiffers. Quality of context: What it is and why we need it. In *10th Workshop of the HP OpenView University Association (OVUA '03)*, 2003.
2. S. Dobson, L. Coyle, and P. Nixon. Hybridising events and knowledge as a basis for building autonomic systems. *IEEE TCAAS Letters*, 2007. To appear.
3. M. R. Ebling, G. D. H. Hunt, and H. Lei. Issues for context services for pervasive computing. In *Workshop on Middleware for Mobile Computing*, 2001.
4. P. Gray and D. Salber. Modelling and using sensed context information in the design of interactive applications. In *Engineering for HCI*, volume 2254/2001 of *LNCS*, pages 317–335, 2001.

5. T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. In *Proc' of Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 270–275, 2004.
6. K. Henricksen and J. Indulska. Modelling and using imperfect context information. *Pervasive Computing and Communications Workshops, 2004. Proc' of the 2nd IEEE Annual Conference on*, pages 33–37, 2004.
7. P. Korpipaa, J. Mantyjarvi, J. Kela, H. Keranen, and E. Malm. Managing context information in mobile devices. *Pervasive Computing, IEEE*, 2(3):42–51, 2003.
8. H. Lei, D. M. Sow, I. John S. Davis, G. Banavar, and M. R. Ebling. The design and applications of a context service. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(4):45–55, 2002.
9. S. McKeever, J. Ye, L. Coyle, and S. Dobson. A multilayered uncertainty model for context aware systems. In *Adjunct Proc' of the international conference on Pervasive Computing: Late Breaking Result*, pages 1–4, May 2008.
10. A. Ranganathan, J. Al-Muhtadi, and R. Campbell. Reasoning about uncertain contexts in pervasive computing environments. *Pervasive Computing, IEEE*, 3(2):62–70, 2004.
11. A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. V. Laerhoven, and W. V. D. Velde. Advanced interaction in context. In *Proc' of 1st International Symposium on Handheld and Ubiquitous Computing*, pages 89–101, 1999.
12. G. Shafer. *A mathematical theory of evidence*. Princeton Unverisity Press, 1976.
13. H. Wu, M. Siegel, and R. Stiefelhagen. Sensor fusion using dempster-shafer theory. In *Proc' of IEEE Instrumentation and Measurement Technology Conference*, 2002.
14. J. Ye, S. McKeever, L. Coyle, S. Neely, and S. Dobson. Resolving uncertainty in context integration and abstraction: context integration and abstraction. In *ICPS '08: Proceedings of the 5th international conference on Pervasive services*, pages 131–140, New York, 2008. ACM.