Pervasive and Mobile Computing 6 (2010) 21–42

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Supporting pervasive computing applications with active context fusion and semantic context delivery

Nirmalya Roy [a,*,1], Tao Gu [b], Sajal K. Das [c]

[a] *Department of Electrical and Computer Engineering, The University of Texas at Austin, TX, United States*
[b] *Department of Mathematics and Computer Science, The University of Southern Denmark, Denmark*
[c] *Department of Computer Science and Engineering, The University of Texas at Arlington, TX, United States*

ARTICLE INFO

Article history:
Received 4 June 2008
Received in revised form 24 June 2009
Accepted 2 October 2009
Available online 30 November 2009

Keywords:
Pervasive computing
Context awareness
Context fusion
Bayesian networks
Ontology
SunSPOT

ABSTRACT

Future pervasive computing applications are envisioned to adapt the applications' behaviors by utilizing various contexts of an environment and its users. Such context information may often be ambiguous and also heterogeneous, which make the delivery of unambiguous context information to real applications extremely challenging. Thus, a significant challenge facing the development of realistic and deployable context-aware services for pervasive computing applications is the ability to deal with these ambiguous contexts. In this paper, we propose a resource optimized quality assured context mediation framework based on efficient context-aware data fusion and semantic-based context delivery. In this framework, contexts are first fused by an active fusion technique based on Dynamic Bayesian Networks and ontology, and further mediated using a composable ontological rule-based model with the involvement of users or application developers. The fused context data are then organized into an ontology-based semantic network together with the associated ontologies in order to facilitate efficient context delivery. Experimental results using SunSPOT and other sensors demonstrate the promise of this approach.

1. Introduction

Emerging pervasive computing technologies transform the way we live today by embedding computation in our surrounding environments. To reduce user distraction and allow users to focus on their tasks, pervasive computing applications must be aware of the context in which they run. These applications should be able to learn and dynamically adapt their behaviors to the current context such as the current state of the user, the physical environment and the computational environment, so that the user can focus on her/his current activity [1]. To achieve this, both users and applications need to acquire and manage their context information "anywhere and anytime". Hence, a fundamental problem for building pervasive computing applications will be how to minimize the ambiguity of context information – context fusion and deliver the information to thousands of diverse applications – context delivery, while scaling to a large number of sources, users and applications.

In reality, sensed and derived contexts are often heterogeneous and ambiguous. For example, a user location can be sensed using active badges, video cameras or even pressure sensors in the floor. All of these sensors have some degree of ambiguity in the data they sense. The ambiguity may become worse when deriving high-level contexts (i.e., user activity)

* Corresponding author. Tel.: +1 682 433 1904; fax: +1 817 272 3784.
E-mail addresses: nirmalya.roy@mail.utexas.edu (N. Roy), gu@imada.sdu.dk (T. Gu), das@uta.edu (S.K. Das).

1 The work of the first author was done while he was a Ph.D. student in Computer Science and Engineering and a member of the Center for Research in Wireless Moblity and Networking (CReWMaN) at the University of Texas at Arlington.

1574-1192/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2009.10.002

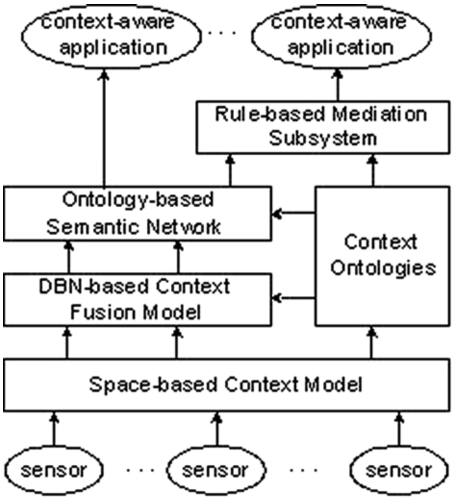

Fig. 1. Overview of our proposed framework.

from sensor outputs. This fundamental problem can be challenging from the following two aspects. Firstly, the data collected from multiple sensors always associated with an inherent ambiguity and the situation is often dynamic and unfolds over time. Though the sensing technology is becoming more ubiquitous and accurate, the interpretation of sensed data as context is still imperfect. The mapping from sensors' output to context information can be complicated due to the complex situation of an environment. We need a dynamic model to reflect changes, capture the beliefs of the current events, and predict the evolution of different situations. An adaptive system is therefore needed so that it cannot only systematically handle raw sensory data of different modalities, but also more importantly, reason over time to reduce context ambiguity during the interpretation of the situation. Secondly, the underlying context delivery mechanism can be critical due to the dynamic characteristics of context information and the diversity of computational environments. In such environments, context data stored in a context provider may be changed rapidly and context providers may leave or join the network frequently.

Aiming to address this fundamental problem, this paper proposes a framework to support both context fusion and context delivery for pervasive computing applications. We propose an active context fusion technique based on Dynamic Bayesian Networks (DBN). This technique is able to fuse ambiguous contexts, resolve conflicts in information overlap and ensure the conformance to the application's Quality of Context (QoC) bound [2,3]. This layered and modularized fusion technique adopts a space-based context model [4] in which context information is represented by different hierarchies. The space-based context model will then be associated with ontologies and contexts can be further mediated using a composable rule-based mediation subsystem with the involvement of users or application developers. For context delivery, we explore the semantics of context data in the form of context ontologies and organize context data semantically into clusters in a semantic network. This way, context information can be stored and retrieved according to their semantics, and context delivery can be more efficient in the presence of dynamic contexts. Fig. 1 gives an overview of our framework.

In summary, this paper makes the following contributions:

- We state an optimization problem using a generic QoC function to determine the optimal set of sensors that satisfy the specified quality of context at a minimum communication cost.
- We propose an active context fusion technique based on DBN for fusing multimodal fragments of sensor data, and a composable rule-based mediation model to infer the situation space based on a hierarchy of contexts.
- We propose an efficient context delivery technique where context data can be stored and retrieved according to their semantics by combining our fusion model with context ontologies.
- We perform our experiments using a combination of simulation traces and real data collected from SunSPOT sensor. The experimental results demonstrate that our framework is capable of reducing the sensor overhead (communication cost) while ensuring the acceptable context accuracy, predicting a user's situation, organizing and delivering contexts semantically.

The rest of the paper is organized as follows. Section 2 discusses related work with an example scenario. Section 3 presents the context model and QoC. Section 4 describes the active context fusion model based on DBN and ontology. A rule-based model has also been discussed in Section 4. We present semantic context delivery in Section 5. The performance of our proposed system is evaluated and the results are presented in Section 6. Finally, Section 7 concludes the paper.

2. Related work

Dey [5] proposed a toolkit which enables the integration of context data into applications and supports context-aware applications. The context fusion is based on simple name transformation and its context delivery assumes the priori knowledge about the presence of a widget or a context broker. Dey extended his work with the distributed mediation of ambiguous contexts that allow the user to correct ambiguity in sensed input [6]. Multimodal Maps [7], a map-based application for travel planning addresses ambiguity by using multimodal fusion to combine different inputs and then prompt the user for more information to remove the remaining ambiguity as much as possible. Remembrance Agent [8] uses context to retrieve information relevant to the user and explicitly addresses ambiguity in its manual interface. Chen, et al. [9] proposed a platform, named Solar, to support data fusion services and context dissemination to context-aware applications. Their fusion mechanism is based on an operator graph model, in which context processing is specified by application developers in terms of sources, sinks and channels. Solar also provides a policy driven data dissemination service based on a multicast tree. However, their fusion mechanism does not consider different type of contexts, and building a multicast tree may incur large overhead in the presence of node changes. Hong et al. [10] proposed Confab which includes a flexible and distributed data store and a context specification language. The context storage consists of a logical context data model which provides a logical representation of context information, and a physical data store where the context data is actually stored. While our context delivery service shares the similar idea of distributed context storage of Confab in which the context data is kept close to where it was generated and where it is likely to be used, our emphasis is on how to provide a scalable semantic delivery service for applications. Gaia [11] is an infrastructure supporting the construction of applications for smart spaces. It consists of a set of core services for building distributed context-aware applications. Different from the context service in Gaia, we focus on providing a semantic delivery service where context information can be shared in a semantic manner. However, these systems do not consider a formal context fusion mechanism which can fuse high-level contexts for different applications so that the common module for fusing context can be viewed as a shared and reusable service.

Many efforts have been made to develop middleware systems that can effectively support context-aware applications in the presence of resource constraints (e.g., sensor networks), also considering requirements for sensory data or information fusion from middleware perspective. For example, DFuse [12] is a data fusion framework that facilitates transfer of different areas of application-level information fusion into the network to save power. DFuse does this transfer dynamically by determining the cost of network using cost functions. The tradeoff between communication overhead and the quality of the reconstructed data was first studied in [13], which envisioned the effect of tolerance ranges on the relative frequency of sink-initiated fetching vs. source-initiated proactive refreshes. The focus, however, is on snapshot queries and not on *continually satisfying* the QoC bound of a long-standing subscription. The idea of exploiting temporal correlation across the successive samples of individual sensors for reducing the communication overhead for snapshot queries is addressed in [14], which used training data to parameterize a jointly-normal density function. While a precursor to our work, the focus there was on meeting the QoC requirements for a class of aggregation queries, whereas our focus is on arbitrary relationships between a context variable and its underlying sensor data. Adaptive middleware [15] has been proposed for context-aware applications in smart homes. In this scheme, the middleware matches the QoC requirements with the available QoC of the sensors. The application's QoC requirements are mapped to a utility function using the QoC attributes of the sensors available. Similar to the adaptive middleware scheme, in MiLAN [16], quality of service (QoS) requirements of applications are matched with the QoS provided by the sensor networks. However, in this scheme, the QoS requirements of the applications are some predetermined numbers, which the applications should know in advance in addition to the quality associated with the type of sensors it can use. In dynamic and pervasive computing environments, the number and types of sensors available to the applications may vary. Therefore, it is impractical to include knowledge about all the different sensor nodes that an application can potentially make use of. In addition, all these sensors come at various levels of cost and benefit to the application. Dielmann et al. [17] proposed a statistical approach using DBN to infer the situation in a meeting scenario. They used both a two-level hidden Markov model (HMM) and a multistream DBN, and demonstrated that the DBN architectures are an improvement over a simple baseline HMM, with the multistream DBN with counter constraints producing an action error rate of 6%. Brdiczka et al. [18] proposed both a deterministic approach based on Petri nets and a probabilistic one based on HMM to represent abstract context in the situation model. Both approaches are well adapted for particular applications: Petri nets for parallelism and HMM for erroneous or uncertain input. MidFusion [19] discovers and selects the best set of sensors on behalf of applications (transparently), depending on the QoS guarantees and the cost of information acquisition. They also provide a sensor selection algorithm to select the best set of sensors using the principles of Bayesian Networks (BN) and Decision theories. Even though, the computation for all combinations of sensors requires only one set of BN inferencing for all the sensors, the computational complexity is still exponential in terms of the number of sensors. Therefore, the best sensor set selection approach is specifically suited to sensor networks hierarchically organized into groups or clusters of sensors, with one sensor node designated as the interface to the services provided by these groups or clusters. Therefore we investigate this cluster-based approach for semantic-based context delivery by considering the DBN-based context fusion model as our baseline.

2.1. Example scenario

As an example, let us take the scenario of a home care patient after hospitalization for cardiac infarction. Although such a patient should be guaranteed a good quality of life and wellness management services in an independent way, he/she still needs to be in constant contact with an expert physician so that his/her cardiac activity (e.g., the heart rate and peripheral blood pressure), body temperature and breathing frequency can be continuously monitored. However, the health condition of a patient can only be partially evaluated through his vital signals and must be mediated and integrated by other signals and information coming both from personal characteristics (risk factors, degree of disease, age, sex, family history, psychological features, etc.) and from the environmental context (e.g., whether in bed or mobile, by him/herself or in company, at work or at home, the season and the temperature, etc.). The monitoring system should be able to deduce the context from the available data to provide a feedback to the patient as well as notifying his status to somebody else, such as a relative, the family doctor, or the hospital, depending on the degree of alert detected, and possibly adapting the level of service (i.e., the intensity of the monitoring activity).

The above scenario requires the integration of patients' vital signs monitored by different sensory medical devices, of environmental data acquired by sensors located near the patient, of patient data available from the electronic medical records stored by the hospital. Although the current technologies offer the necessary means to support this kind of health care, in our opinion without a contextual realization that tailors the available data into usable information, the healthcare applications will become practically unusable. Contextual information deals with information about the user environment (e.g., location, activity) that enables this tailoring and reduces efforts required to develop healthcare applications.

Application scenarios of the type presented above give rise to several issues. The sensory devices constantly attached to the patient produce huge streams of physiological data which must be collected and related to environmental conditions and should be delivered to the proper healthcare providers according to their underlying semantics. To achieve this we need a technique that can fuse acquired data with different modalities to infer the current context state (activity) and situation space (behavior or sickness) associated with the monitored person and group data in semantic cluster for efficient context delivery. Again the sensors should be light and portable to reduce their impact on the patient's well-being and thus must be constrained in terms of energy capacity. Consequently, the amount of information transmitted to the sensor fusion mediator (the data aggregator) should be minimized in order to prolong its lifetime by selecting the structure of an optimal set of sensors based on the QoC guarantees and cost of information acquisition.

3. Context model

Context-aware data fusion in the face of ambiguities is a challenging research problem as the data sent to the sensor fusion mediator collected from a network of multiple sensors is often characterized with a high degree of complexity due to the following challenges: (i) data is often acquired from sensors of different modalities and with different degrees of uncertainty and ambiguity, (ii) decisions must be made quickly, and (iii) the situation as well as sensory observations always evolve over time. We use the space-based model as our underlying context model, which consists of context attribute, context state and situation space. To facilitate our active fusion model, we extend the basic model with QoC attributes [29]. The extended model incorporates various intuitions for context inference to achieve a better fusion result.

3.1. The basic model

The space-based context model defines the following concepts:

Definition 1 (*Context Attribute*). A context attribute, denoted by a_i, is defined as any type of data that is used in the process of inferring situations. A context attribute is often associated with sensors, virtual or physical, where the values of the sensor readings denote the context attribute value at a given time t, denoted by a_i^t. *The body temperature* "100 °F" *of a patient measured by ith sensor at a given time t is an example of a context attribute.*

Definition 2 (*Context State*). A context state describes the application's current state in relation to a chosen context and is denoted by a vector S_i. It is a collection of N context attribute values that are used to represent a specific state of the system at time t. Thus, a context state is denoted as $S_i^t = (a_1^t, a_2^t, \ldots, a_N^t)$. *Suppose the body temperature is* "100 °F" *and the location is in "gym", then the context state of the patient is "doing physical exercise".*

Definition 3 (*Situation Space*). A situation space represents a real-life situation. It is a collection of ranges of attribute values corresponding to some predefined situation (sickness, normal behavior) and denoted by a vector space $\mathcal{R}_i = (a_1^R, a_2^R, \ldots, a_{\mathcal{M}}^R)$ (consisting of $\mathcal{M}$ acceptable ranges R for these attributes). An acceptable range a_i^R is defined as a set of elements V that satisfies a predicate $\mathcal{P}$, i.e., $a_i^R = V|\mathcal{P}(V)$. *For example the context attribute body temperature can take values within* "98 °F" *to* "100 °F" *when the patient context state is "doing physical exercise" with predefined situation space "normal". But if the context attribute body temperature takes values within this range* "98 °F" to "100 °F" *when the patient context state is "lying on the bed" then the situation space is "not normal". Again if we can use context attribute "PDA location" of a user to infer "user in a meeting" situation, then the accepted region for this attribute might be the location information of the PDA such as "at home", "at office" or "at meeting room".*

3.2. Quality of context model

Despite recent development in sensing and network technology, continuous monitoring of individuals vital signs (e.g., the heart rate and peripheral blood pressure, body temperature and respiratory rate) and environmental context (e.g., whether in bed or mobile, by him/herself or in company, at work or at home, the season and the temperature, etc.) in normal setting is still challenging due to the resource constrains of sensor networks. We define the Quality of Context (QoC) [20] as a metric for minimizing resource usage (e.g., battery life, communication bandwidth) while maintaining a minimum quality of the data received. QoC is essential to our model in choosing the best data values among the monitored ones for reporting a specific type of context. For example, if the blood pressure of an inhabitant in a smart home monitoring environment lies in between the predefined normal range (120/80 mm Hg), or frequency of getting up from the bed at night is (2–3 times) then the sensor need not to report that value to the sensor fusion mediator again. But if the aggregated value computed at the mediator is beyond the tolerance level of QoC (± 10 mm Hg for BP or >5–6 times for Frequency), then the sensor needs to report its samples back to the mediator.

We extend the basic model with QoC and relate the extended model with some specific application scenarios. We define properties, called QoC attributes that characterize the quality of the context data received. In context fusion, the sensor fusion mediator ensures over the entire query lifetime that the aggregated value computed by it does not diverge from the true reading by more than a specified "tolerance". The key is to have the mediator communicate a precision range or interval to an individual sensor, with an idea that a sensor does not need to report its samples back to the mediator as long as they fall into this specified range. Such tolerance is expressed in terms of the QoC metric, and is especially useful for applications issuing aggregation queries. We assume that the information fusion issues an aggregation query with its QoC specified by a precision range Q, this implies that the aggregate value computed at the mediator at any instant should be accurate within Q.

Our objective is to evaluate the update cost of a sensory action A for a given task while ensuring the conformance to the application's QoC bound. Let us denote the update cost (in terms of communication overhead) as $\mathcal{U}_i^j$ if indeed sensor B_i has to report its sample value at time j to the mediator. Then, the objective is to minimize

$$\sum_{i \in B_m} \mathcal{U}_i(q_i) \tag{1}$$

where $\mathcal{U}_i$ denotes the expected average update (reporting) cost and explicitly indicates its dependence on the specified precision interval q_i (tolerance range). Intuitively, $\mathcal{U}_i$ is inversely proportional to q_i, since the value of the reporting cost would be high as the precision interval continues to shrink.

But this update cost also depends on the hop count h_i, the length of the uplink path from sensor B_i to the mediator. Accordingly, the update cost can be rewritten as:

$$\text{minimize} \sum_{i \in B_m} \mathcal{U}_i(q_i, h_i). \tag{2}$$

If the underlying data samples evolve as a random-walk model, we have $\mathcal{U}_i \propto \frac{h_i}{(q_i^2)}$ resulting in the following optimization function:

$$\text{minimize} \sum_{i \in B_m} \frac{h_i}{(q_i^2)}. \tag{3}$$

For example, as shown in Fig. 2 with data from a respiratory sensor, we can capture the respiratory rate within a ± 5 bound (say) 90% of the time (i.e., with 0.9 probability). In Fig. 2 each arc from a sensor to the context attribute is associated with a function of three parameters: q (accuracy range of sensor data), Q (accuracy range of derived context attribute) and $\wp$ (fidelity of the context attribute being derived lying with the range Q). Thus, the fidelity function is $\wp = f_1(q_1, Q)$ for sensor B_1. In other words, given tolerances on q_1 and Q, we can say how often (in an ergodic sense), the fused context attribute estimation will lie within the true accuracy range $\pm Q$. Similarly, when we consider two sensors B_1 and B_2 jointly, the fidelity function should be $\wp = f_{12}(q_1, q_2, Q)$. In this way, for m sensors, there are $2^m - 1$ (all possible combinations except no sensors) functions $f(\cdot)$, indicating the relationship between context attribute, context fidelity and precision range. Given these continuous functions, the 'application' now says that it needs a precision bound (on the context attribute) of $\acute{Q}$ with a fidelity of at least $\acute{\wp}$.
Then, the problem is:

Problem 1. Find the combination of $q_1, q_2, \ldots, q_m$ that satisfies $f_{1,\ldots,m}(q_1, q_2, \ldots, q_m, \acute{Q}) \geq \acute{\wp}$, and yet minimizes $\sum_{i \in B_m} h_i/(q_i)^2$.

Of course, in some cases, the context attribute value may be Boolean or binary. In this case, it may be hard to associate a bound Q with the accuracy of the context. However, this is a special case — a model for the more general case is developed here.

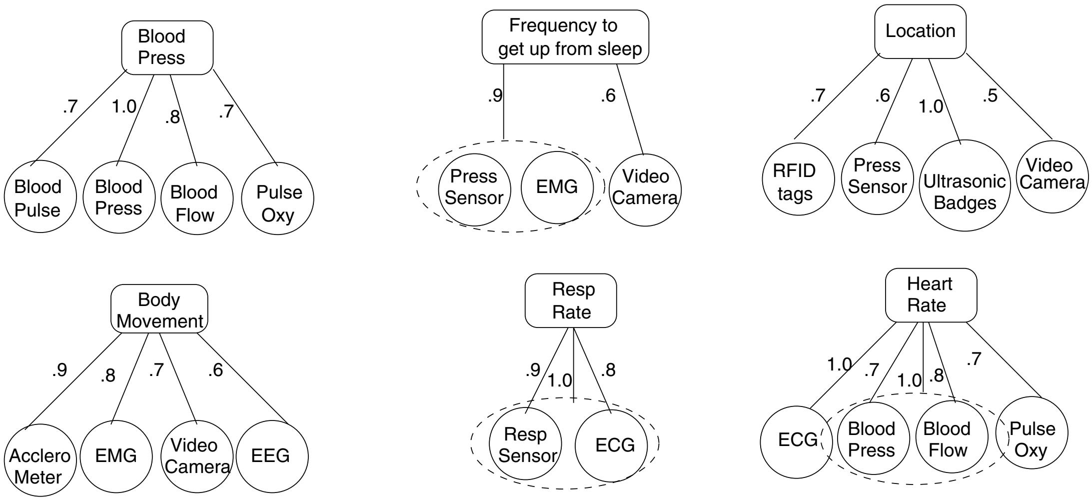

Fig. 2. Subset of sensors with required QoC.

The problem of optimally computing the q_i values can be represented by the Lagrangian:

$$\text{minimize} \sum_{i \in B_m} \frac{h_i}{q_i^2} + \lambda \times \left[f_{1,\ldots,m}(q_1, q_2, \ldots, q_m, \acute{Q}) - \acute{\wp} \right]. \tag{4}$$

Finding an exact solution to Eq. (4), for any arbitrary $f(\cdot)$ is an NP-complete problem [21]. While a completely arbitrary $f(\cdot)$ function requires a brute-force search, there are certain forms of $f(\cdot)$ that prove to be more tractable. In particular, a particularly attractive case occurs when the ith sensor's individual fidelity function is represented by the form

$$\mathbf{f}_S(i) = \nu_i * \exp^{-\frac{q_i^2}{\eta_i}}, \tag{5}$$

where η_i and ν_i are sensitivity constants for sensor s_i. A larger value of η_i indicates a lower contribution from sensor s_i to the inference of context state S. Moreover, for a selection of m sensors, the resulting $f(\cdot)$ function has the form:

$$f_S(m) = 1 - \prod_{i \in B_m} (1 - \mathbf{f}_S(i)). \tag{6}$$

We solve this by taking the Lagrangian optimization i.e, we solve for

$$\text{minimize} \sum_{i \in B_m} \frac{h_i}{q_i^2} + \lambda \left[1 - \prod_{i \in B_m} \left[1 - \left(\nu_i * \exp^{-\frac{q_i^2}{\eta_i}} \right) \right] - \acute{\wp} \right] \tag{7}$$

and prove Lemma 1.

Lemma 1. *The combination of* $q_1, q_2, \ldots, q_m$ *which satisfies the fidelity function* $f_{1,\ldots,m}(q_1, q_2, \ldots, q_m, \acute{Q}) \geq \acute{\wp}$ *and minimizes the objective function in* (3) *is*

$$\frac{h_1 * \eta_1 * \left(1 - \nu_1 * \exp\left(-\frac{q_1^2}{\eta_1}\right)\right)}{q_1^4 * \nu_1 * \exp\left(\frac{-q_1^2}{\eta_1}\right)} = \cdots = \frac{h_m * \eta_m * \left(1 - \nu_m * \exp\left(-\frac{q_m^2}{\eta_m}\right)\right)}{q_m^4 * \nu_m * \exp\left(\frac{-q_m^2}{\eta_m}\right)}.$$

Proof. The above expression follows immediately by taking partial derivatives of the Lagrangian in Eq. (7) and setting them to 0 as shown below. In our case:

$$\textbf{minimize} \quad \sum_{i \in B_m} \frac{h_i}{q_i^2}$$

$$\textbf{subject to}: \ 1 - \prod_{i \in B_m} \left[1 - \nu_i * \exp^{-\frac{q_i^2}{\eta_i}} \right] \geq \acute{\wp}. \tag{8}$$

Taking log we can rearrange the constraint of Eq. (8), as

$$\log(1 - \acute{\wp}) \geq \sum_{i \in B_m} \log \left(1 - \nu_i * \exp^{-\frac{q_i^2}{\eta_i}} \right). \tag{9}$$

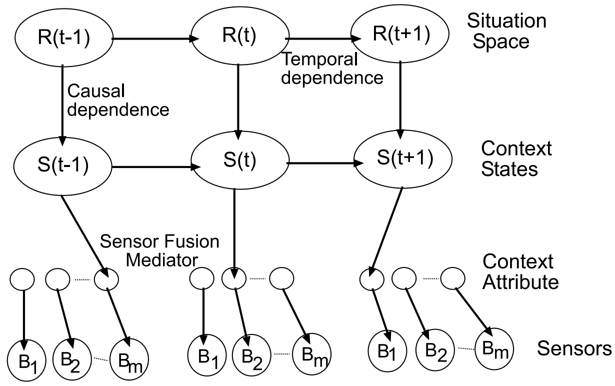

Fig. 3. Context-aware data fusion framework based on dynamic Bayesian networks.

Considering this, we form the Lagrangian constraint,

$$\text{minimize} \sum_{i \in B_m} \frac{h_i}{q_i^2} + \lambda \left[\log(1 - \acute{\wp}) - \sum_{i \in B_m} \log \left(1 - \nu_i * \exp^{-\frac{q_i^2}{\eta_i}} \right) \right]. \tag{10}$$

Taking the partial derivative of the Eq. (10) w.r.t q_i and equating it to 0, we find

$$\lambda = \frac{h_i * \eta_i * \left(1 - \nu_i * \exp\left(-\frac{q_i^2}{\eta_i}\right)\right)}{q_i^4 * \nu_i * \exp\left(\frac{-q_i^2}{\eta_i}\right)} \tag{11}$$

which proves the optimal choices of q_i as stated in Lemma 1. ■

This optimization problem helps us to chose the values of $q_1, q_2, \ldots, q_m$ for a given a set of sensor m, such that we minimize the total update cost while ensuring the required accuracy level is achieved.

4. Active context fusion

A characteristic of a sensor-rich smart healthcare environment is that it senses and reacts to *context*, information sensed about the environment's occupants and their daily activities, by providing context-aware services that facilitates the occupants in their everyday actions. Here we develop an approach for sensor data fusion in context-aware healthcare environment considering the underlying space-based context model and a set of intuitions it covers. In the case of context-aware services, it is really difficult to get an accurate and well-defined context which we can classify as 'unambiguous' since the interpretation of sensed data as context is mostly imperfect and ambiguous. For example, a video camera system which identifies the user posture based on the current position (sitting, standing, lying on the floor in distress) has a different probability of correctness than using pressure sensors in furniture. The ambiguity problem becomes worse when the application derives implicit higher-level context state (activity of the person) based upon those inputs. For example, an application may infer that a person is lying in distress. However, there may be other explanation of this phenomenon such as the person might be lying to perform normal exercises. Again these applications are often constrained by limited time and resources, and thus require taking fast decisions. It implies that an effective fusion model should be active, purposive and adaptive.

An active context fusion model involves several difficult issues: a query/task must be presented in the system and the system must use this representation to operate in a goal-oriented manner. It must select the most informative context attribute which maximize its solution towards our goal to predict the situation space. It also needs to evaluate the cost of a sensory action for a given task towards the QoC bound of the application. We use a typical DBN-based data fusion algorithm to develop a context-aware model which provides a mathematical framework to gather knowledge from sensor data. The model performs context fusion in both top-down and bottom-up inference mechanisms. The top-down inference can be used to predict the utility of a particular sensory action with respect to a goal at the top, e.g., for a given context state it will fuse the most relevant context attribute. The bottom-up inference allows the integration of the context attributes from a sensory action and updates each node about the context state in the network.

4.1. Dynamic Bayesian Network based model (DBN)

A DBN-based model, is presented in Fig. 3, consists of a situation space, context states, context attributes, a sensor fusion mediator and network of information sensors [22].

The selection of an information source (i.e., sensor) or the activation of a process to compute new information is simply regarded as a set of actions available to the decision maker in decision theory. For example, the value of "location" attribute can be measured by ultrasonic badges, RFID tags, video cameras or pressure sensors in the floor. In our case, the information module needs to determine the next optimal context attributes and the corresponding sensory action such as triggering ultrasonic badges or RFID tags or video cameras or pressure sensors. Selecting an action always has a consequence which can be measured by the cost of information acquisition. If we can devise a cost measure to each possible consequence, this can be used by the system to decide what action to perform, and what sensor to activate.

Let us assume that we have a situation space $\mathcal{R}_i$ to confirm using the sensory information sources $B = \{B_1, \ldots, B_m\}$ which is a set of measurements taken from sensors labeled from 1 to m as shown in Fig. 3. The context attribute which is most relevant in our case should decrease the ambiguity of the situation space a_j^R the most; and we will select the one that can direct the probabilities of the situation space to near one (for maximum) and zero (for minimum). Let $\mathcal{V}_i$ be the ambiguity reducing utility to the situation space $\mathcal{R}_i$. Then the expected value of $\mathcal{V}_i$, given a context attribute a_i^t from a sensor B_i, which has K possible values, can be represented as

$$\mathcal{V}_i = \max_{i=0}^{K} \sum_{j=0}^{N} [P(a_j^R|a_i^t)]^2 - \min_{i=0}^{K} \sum_{j=0}^{N} [P(a_j^R|a_i^t)]^2 \tag{12}$$

where $i \in \{1, 2, \ldots, m\}$ is a sensor tag which identifies the sensor that provides the context attribute. This context attribute can be measured by propagating the possible outcome of an information source, i.e.,

$$P(a_j^R|a_i^t) = \frac{P(a_j^R, a_i^t)}{P(a_i^t)}. \tag{13}$$

However, quantification of this conditional probability needs a detailed model depending upon the usage of different types of sensors and their applications. Consider, for example, an audio sensor. Evaluating the benefit of using audio in disambiguating whether a person is moaning in pain or singing, is really hard. It depends on how far the person is from the microphone, which way the person is facing, the time of day (at night it is more quiet so sounds can be heard more clearly), the state of other potentially interfering audio sources (such as air conditioning, TV, radio, refrigerator), etc. Computing the disambiguating utility therefore, needs very detailed models of how the above factors affect the efficacy of the audio sensor.

Considering the information update cost from Eq. (2) and ambiguity reducing utility from Eq. (12), the overall utility can be expressed as

$$U_i = \alpha \mathcal{V}_i + (1 - \alpha)(1 - \mathcal{U}_i) \tag{14}$$

where $\mathcal{U}_i$ is the update cost to acquire the information by the sensor with tag i with a knowledge of QoC bound, and α denotes the balance coefficient between the ambiguity reduction and the cost of information acquisition. Eq. (14) represents the contribution to ambiguity reduction and the cost associated with information retrieval to achieve the desired level of confidence to the situation space. We can observe from Eq. (14) that the utility value of a context attribute a_i increases with the ambiguity reducing utility and decreases as the cost to acquire that attribute increases. So the most economically efficient disambiguation sensor action A^* can be chosen with the help of the following decision rule:

$$A^* = \arg\max_A \sum_j U(B, a_j^R) P(a_j^R|B) \tag{15}$$

where $B = \{B_1, \ldots, B_m\}$ is a set of measurements taken from sensors labeled from 1 to m at a particular point of time. By incorporating the temporal dependence between the nodes as shown in Fig. 3, the probability distribution of the situation space we want to achieve can be generally described as:

$$P(\mathcal{R}, A) = \prod_{t=1}^{T-1} P(S_t|S_{t-1}) \prod_{t=1}^{T-1} P(\mathcal{R}_t|B_t) P(\mathcal{R}_0) \tag{16}$$

where T is the time boundary. This sensor action strategy must be recalculated at each time slice since the best action varies with time. The ambiguous context mediation algorithm is presented in Fig. 4.

4.2. Ontology-based fusion model

We present our context ontology for modeling context fusion in Fig. 5. We design our ontology in a hierarchical manner from a top-level ontology to a domain-specific ontology. The upper ontology is a high-level reusable ontology structure which captures general features of basic contextual entities. Domain-specific ontologies are a collection of ontology set which define the details of general concepts and their features in each sub-domain. The context model is structured around a set of abstract entities, each describing a physical or conceptual object including location, Person, Computational Entity as well as a set of abstract sub-classes. Each entity is associated with its context attributes (represented in owl:DatatypeProperty) and relations with other entities (represented in owl:ObjectProperty). Another flexible OWL

Procedure ACMA

1. $t = 0$, Compute ambiguity-reducing utility $\{\mathcal{V}_1^t, \ldots, \mathcal{V}_m^t\}$ by Eqn.12
2. Calculate utility value $\{U_1^t, \ldots, U_m^t\}$ using Eqn.14
3. Select the most economically efficient disambiguation sensor action based on Eqn.15
4. Instantiate the subset of corresponding sensors B
5. Run ACMA inference algorithm to update probability distribution $P(\mathcal{R}, A)$ of situation space
6. If $P(\mathcal{R}, A^*) \geq$ confidence threshold, then terminate; otherwise
7. Add a new time stamp $t = t + 1$, and go to step 1

Fig. 4. Ambiguous Context Mediation Algorithm (ACMA).

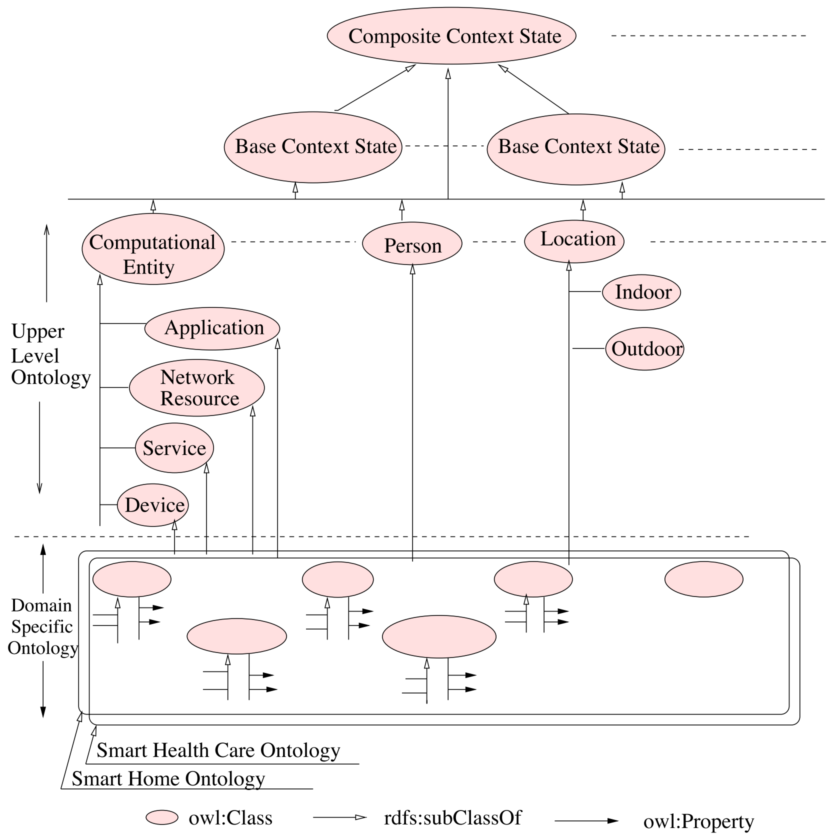

Fig. 5. Context fusion upper ontology.

property (owl:subClassof) helps us to add new concepts in a hierarchical manner that are required in a specific domain without any rigor. Fig. 6 presents a partial context ontology for a healthcare application. The abstract class "device" has been classified into few sub-classes such as BSN, TV and Cell phone, and BSN has been sub-grouped into two different sub-classes such as Ambient Sensor and Wearable Sensor. Context attributes (e.g., location, person, time, computational entity) are most fundamental contexts for taking final decision about the executing situation. These contextual attributes not only form the skeleton of the context state but also conglomerate with a base context state to derive a composite context state.

Our basic context model represents context attribute as Predicate (subject, value). We extend the basic model to derive a context state or a set of context states by combining the Boolean Operator such as union, intersection, complement etc. We also extend the basic context model to incorporate probabilistic information for representing uncertain contexts. It has the form of Prob(Predicate(subject, value)), in which the probability measurement takes a value between 0 and 1. For example, Prob(BodyTemperature(John, HIGH)) = 0.9 means the probability that John's body temperature is currently high is 0.9.

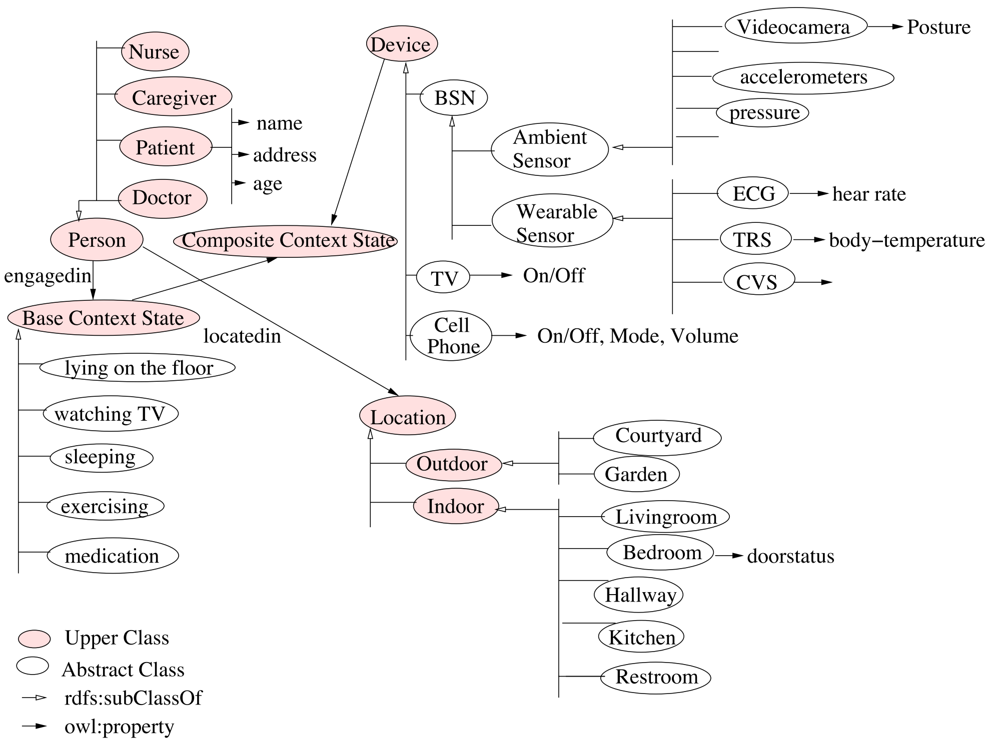

Fig. 6. Context fusion lower ontology.

Table 1
User defined context state derivation rules.

Context state	Reasoning rules
Doing physical exercise	(?x LocatedIn Gym) ∧ (x BodyTemp HIGH) ∧ (x RespiratoryRate HIGH) ⇒ (?x contextstate exercise)
WatchingTV	(?x LocatedIn LivingRoom) ∧ (TVSet LocatedIn LivingRoom) ∧ (TVSet status ON) ⇒ (?x contextstate watchingTV)
Lying on the floor	(?x LocatedIn Kitchen) ∧ (x posture lying) ∧ (ElectricOven status ON) ⇒ (?x contextstate lying on the floor)
Walking	(?x LocatedIn Courtyard) ∧ (x motion ON) ∧ (House doorstatus OPEN) ⇒ (?x contextstate walking)
Going to restroom	(?x LocatedIn Bathroom) ∧ (Flush status ON) ∧ (Bathroom doorstatus CLOSED) ⇒ (?x contextstate going to restroom)
Lying on the bed	(?x LocatedIn Bedroom) ∧ (Bedroom lightlevel Low) ∧ (Bedroom doorstatus CLOSED) ⇒ (?x contextstate lying on the Bed)
Talking	(?x LocatedIn LivingRoom) ∧ (y LocatedIn LivingRoom) ∧ (x sound HIGH) ⇒ (?x contextstate talking)

We apply Bayesian network to enable learning causal dependencies between various context events, and obtaining probability distributions. In our model, each node corresponds to a context event and directed arcs between nodes represent causal relationships between the contexts. The uncertainty of the causal relationship is represented by the conditional probability $P(a_j^R|a_i^t) = \frac{P(a_j^R,a_i^t)}{P(a_i^t)}$. After getting this conditional probability, we are able to compute the probability distribution of the BN. This DBN-based model can lead to a well-defined context fusion mechanism with the consideration of the inherent ontology associated with different states. With the help of ontology, we can make uncertain context easily shared by different applications and enable context fusion module to derive high-level composite context more accurately and timely. Context attribute (CA) are the instance of the ontology and they can itself fuse to derive the context state (CS) or can be combined with a base context state to derive the composite or high-level context state. Hence, we have, $CA_1 \wedge CA_2 \cdots \wedge CA_n \Rightarrow CS_i$ and $(CA_1 \wedge CA_2 \cdots \wedge CA_n) \wedge (CS_1 \wedge CS_2 \cdots \wedge CS_n) \Rightarrow CS_i$. The user defined context state derivation rules are shown in Table 1. Context states are derived only with the help of the different context attributes. In Table 2, the context fusion rules are defined from single/multiple context states and context attributes; to predict the situation space.

Table 2
User defined context fusion rules.

Situation state	Fusion rules (Derived from a single/multiple context state and context attribute)
Sickness	(?x contextstate lying on the floor)$\wedge$ (x BodyTemp HIGH) $\wedge$ (x RespiratoryRate HIGH) $\Rightarrow$ (?x situationstate sick)
Behavior	(?x contextstate walking) $\wedge$ (currenttime status midnight) $\Rightarrow$ (?x situationstate abnormal)
Behavior	(?x contextstate restroom) $\wedge$ (?x contextstate watchingTV) $\Rightarrow$ (?x situationstate normal)
Sickness	(?x contextstate going to restroom) $\wedge$ (?x contextstate lying on the Bed)$\wedge$ (?x leavingbed HIGH) $\Rightarrow$ (?x situationstate sick)

4.3. Rule-based mediation subsystem

In pervasive computing environments, the ambiguity of context may vary among different applications and users. If we can directly involve applications and users to the model to mediate the current associated ambiguity with the context, unambiguous context can be realized to some extent. An application can choose to ignore the ambiguity and take some action (e.g., act on the most likely choice) or can use rule-based techniques to ask the end user about his/her actual intent. We propose a rule-based mediation subsystem as follows. We define the following rule format derived from active database for identifying ambiguous contexts.
on ⟨context attribute⟩
if ⟨context state⟩
do ⟨situation space⟩
The basic idea is as follows. The evaluation of a situation should be done if a particular context state is true on capturing some context attributes. Context attribute in our framework can be primitive (sensors in the smart home being activated) or non-primitive (complex or composite) which are derived from some other attributes as well. For example, the primitive ones can be location, time, etc and non-primitive ones can be body temperature, respiratory rate, blood pressure, etc. They can also be instantaneous or durative. To illustrate, consider the following rules that can be used to actively monitor the elderly person in the smart home [23].

Rule 1:
on Context Attribute (CA) : ⟨?user hasBodyTemperature ‘VALUE1’⟩, ⟨?user hasRespiratoryRate ‘VALUE2’⟩, ⟨?user hasBlood-Pressure ‘VALUE3’⟩
if Context State (CS) : ⟨?user hasActivity ‘EXERCISE’⟩, ⟨?user hasActivity ‘LIEDOWN’⟩
do Situation Space (SS): ⟨?user inSituation ‘SICKNESS’⟩

Suppose using some specific sensors in an in-door smart home environment, we obtain the value of context attributes such as body temperature, respiratory rate and blood pressure of a person. Now, if these measured values are higher than a specified range, we can infer about the situation of the person by observing the context state. If the context state is ⟨?user hasActivity ‘EXERCISE’⟩ – ‘‘doing physical exercise’’, the situation is normal; otherwise, if the context state is ⟨?user hasActivity ‘LIEDOWN’⟩ – ‘‘lying on the bed’’, the situation is abnormal.

Let us consider another rule where we consider the context attribute as time instant, time span and location of the inhabitant in the home.

Rule 2:
on CA: ⟨?user time ‘6AM’⟩, ⟨?user locatedIn ‘BACKYARD’⟩
if CS: ⟨?user hasActivity ‘WALKING’⟩
do SS: ⟨?user inSituation ‘NORMAL’⟩
If it is 6 am in the morning and the location of the person is at backyard where the context state is ‘‘walking’’, then we can conclude that the behavior of the person is normal. But if it is 2 am in the morning and the location of the person is outside the home where the context state is ‘‘sleeping’’, then we can conclude that the behavior of the person is not normal. We can make different variants of this rule by considering different context attributes with a different context state. Thus detecting that a person has been engaged in a specific activity for an unusual time may be an indicator of a health problem or a potentially hazardous domiciliary situation.

5. Context delivery

Contexts output from our context fusion model are expected to be queried and utilized by various applications. Based on our context ontology model, we organize a large number of context providers into a semantic overlay network. The network groups semantically similar context data into a cluster to facilitate efficient routing. Each context provider maintains a local repository which supports semantic context queries using an RDF-based query language such as RDQL [24].

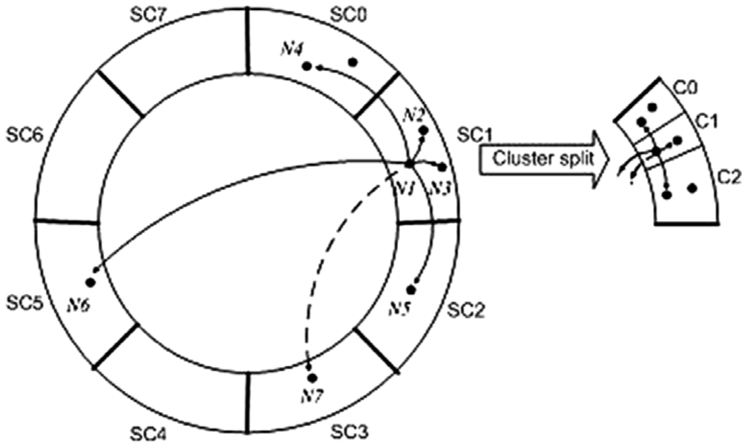

Fig. 7. Semantic network construction.

5.1. Context data mapping

To register context data to the network, we need to extract the semantics of context data so that each context provider can be placed in an appropriate semantic cluster in the network. This is done by mapping context data to semantic cluster(s) based on ontology and counting the number of triples corresponding to each semantic cluster.

To assist the mapping process, we denote the leaf concepts in the upper ontology shown in Fig. 5 as a set $E = \{Indoor, Outdoor, Person, Service, Device, \ldots\}$, and use them as semantic clusters. The mapping technique traces the hierarchy of OWL classes in the upper ontology and maps RDF triples to their associated classes. We create two structures — ClusterHierarchy and ClusterMap. We first map each triple to an OWL class using ClusterMap, and then map the triple to an appropriate semantic cluster using ClusterHierarchy. Let $SCn_{sub}, SCn_{pred}, SCn_{obj}$, where $n = 1, 2, \ldots$, denote the semantic clusters extracted from the subject, predicate and object of a triple respectively (Note: unknown subjects/objects or variables are mapped to E). If the predicate of a triple is of type owl:ObjectProperty, we obtain the semantic clusters using $(SC1_{pred} \cup SC2_{pred} \cup \cdots SCn_{pred}) \cap (SC1_{obj} \cup SC2_{obj} \cup \cdots SCn_{obj})$. If the predicate of a triple is of type owl:DatatypeProperty, we obtain the semantic clusters using $(SC1_{sub} \cup SC2_{sub} \cup \cdots SCn_{sub}) \cap (SC1_{pred} \cup SC2_{pred} \cup \cdots SCn_{pred})$. For example, give a triple—(Bedroom lightLevel 87), "Bedroom" is mapped to *IndoorSpace*, and "lightLevel" is mapped to both *IndoorSpace* and *OutdoorSpace*. Hence, the tripe is mapped to $IndoorSpace \cap OutdoorSpace = IndoorSpace$ finally. We name the semantic cluster corresponding to the highest triple count the major semantic cluster. To achieve this, we create a *HashMap* and iterate through all the triples of a context provider. Upon successful execution, we obtain a vector containing the semantic cluster IDs corresponding to all its local context data. The first element in this vector indicates the ID of its major semantic cluster. A context provider will then join its major semantic cluster and we will describe the joining process in the next section.

5.2. Building a semantic network

We apply the principle of a small world network model [25] and extend it with clustering operations to build our semantic network. After obtaining the semantics of its local data, nodes are organized in such a way that those have semantically similar data (i.e., maps to the same leaf concepts in the upper ontology) are grouped together in a semantic cluster. As a nodes data may correspond to multiple semantic clusters, a node joins its major semantic cluster and publishes the indices of its data (i.e., reference pointer) to its minor semantic clusters.

Routing table construction: Each node builds its routing table by creating a set of local contacts in its own cluster, a short-range contact in each of its neighboring clusters, and a small number of randomly chosen long-range contacts. The local contacts and the short-range contacts are served for intra-cluster routing whereas the long-range contacts are used for inter-cluster routing. Each newly joining node builds its routing table in the same way resulting in all the clusters being linked linearly in a ring fashion. To illustrate, as shown in Fig. 7, *Node* 1 builds two local contacts (*Nodes* 2 and 3) in *SC*1, two short-range contacts (*Nodes* 4 and 5) in *SC*0 and *SC*2, respectively, a long-range contact (*Node* 6) in *SC*5, and publishes its index to a random node (*Node* 7) in *SC*3.

Grouping context data according to their major semantic clusters can minimize the cost of node joining, leaving and data changes. A node will stay in its major semantic cluster as long as the majority of data does not change. However, keeping a large number of nodes in a semantic cluster may result in a scalability issue. We design the follow clustering operations to balance the load in each sub-clusters for better scalability.

Load balancing: When the number of nodes in a semantic cluster exceeds a certain size, cluster splitting occurs. Let M represent the maximum cluster size. If the size of a cluster exceeds M, the cluster is split into two. Each node maintains

Table 3
Calibrated accelerometer sample values for different context states.

Range (5–95th percentile) of tilt values (in degrees)	Context state
85.21 to 83.33	*Sitting*
68.40 to 33.09	*Walking*
28.00 to −15.60	*Running*

a CurrentLoad which measures its current load in terms of the number of triples and data indices it stores. When node x joins the network, it sends a join request message to an existing node, say y. If y falls into the same semantic cluster that x wishes to join, x joins the cluster by connecting to y if its cluster size is below M; otherwise y will direct the request to a node, say z, in the semantic cluster that x wishes to join, and x will connect to z if its cluster size does not exceed M. If the cluster size exceeds M, node y or z (called an initial node) will initiate the splitting process. The initial node first obtains a list of all the nodes in this cluster which is sorted according to their CurrentLoads. Then it assigns these nodes in the list to the two sub-clusters alternatively. After splitting, we obtain two clusters with relatively equal load. The initial node is also responsible for generating a new cluster ID for each of the two sub-clusters. To obtain a new cluster ID, each node maintains a bit split pointer which indicates the next bit to be split in the n-bit binary string. Initially, the bit split pointer points to the most significant bit of the n-bit string. When cluster splitting occurs, the bit pointed by the bit split pointer is split into 0 and 1 and move the pointer forwards to the next bit in the n-bit string. The same mechanism follows for the insertion of a new semantic cluster. A semantic cluster can be split into a maximum number of 2^n clusters. After splitting, a node updates its cluster ID, the bit split pointer as well as its local contacts and short-range contacts. When the number of nodes in a cluster falls below a threshold, cluster merging occurs. When node x leaves the network, it first checks whether its cluster size has fallen below a threshold $M_{\min}$. If the current size is above $M_{\min}$, x simply leaves the network by transferring its indices to a randomly selected node in its cluster. Otherwise, this cluster needs to be merged into one of its neighboring clusters within the same semantic cluster. The leaving node triggers cluster merging which is an inverse process of cluster splitting. If the last node in a semantic cluster leaves, it initiates two messages to all the nodes in its two adjacent clusters informing them to update their neighbor lists. Subsequently, the semantic cluster will be removed from the network.

Query routing: The query routing process involves two steps: inter-cluster routing and intra-cluster routing. Upon receiving a query, node x first obtains the destination Semantic Cluster ID (denoted as D). This is done following the context data mapping process. Then node x will check whether D falls into its own semantic cluster by comparing D against the most significant m-bits of its ClusterID. If that is the case, x will flood the query to all the local contacts and also forward the query to its short-range contacts in its adjacent clusters corresponding to D. The forwarding processes are recursively carried out until all the clusters corresponding to D have been covered and all nodes in each of the clusters are reached. If D falls into neither node x's own cluster nor its adjacent semantic cluster, x will rely on its long-range contacts to route the query across clusters. Each node maintains two long-range contacts, for example, we can partition a 2^m semantic space where $m = 3$ into four by creating two long-range contacts: one pointing to the opposite semantic cluster and another pointing to the semantic cluster located in a quarter of the ring space. Given the maximum cluster size M, the system can have a total of $M \times 2^{m+n-1}$ nodes when $M_{\min} = 1$. Let C_x denote the cluster where x resides in and SC_x denote the semantic cluster that C_x corresponds to. SC_x can be obtained by truncating C_x to m bits from the most significant bit. The two semantic clusters SC_{half} and $SC_{quarter}$ that x's long-range contacts point to are denoted as $(SC_x + 2^i) \bmod 2^m$, where $i = m-1, m-2$. To initiate a search, x obtains D based on a query and checks which cluster range (partitioned by x's long-range contacts) D falls into. Then node x forwards the query to the closer semantic cluster through its long-range contact. If D is closer to SC_x, node x will forward the query across its adjacent cluster towards D.

6. Experimental components and evaluation

We use the SunSPOT [26] (Sun Small Programmable Object Technology) device for context sensing and mediation, which is a small, wireless, battery powered experimental platform. Each free-range SunSPOT contains a processor, radio, sensor board and battery; the base-station SunSPOT contains a processor and radio only. The SunSPOT uses a 32-bit ARM9 microprocessor running the Squawk VM and programmed in Java, supporting the IEEE 802.15.4 standard. In our context sensing and performance evaluation we will use various built-in sensors available with SunSPOT sensor board.

6.1. Empirical determination of context estimates

We used the accelerometer to measure the tilt value of the SunSPOT (in degrees) when the monitored individual was in three different context states: *sitting, walking and running*. From the collected samples, we computed the 5th and 95th percentile of the tilt readings, corresponding to each state. Table 3 shows the resulting ranges in the accelerometer tilt readings observed for each of the three states. The results indicate that there is an observable separation in the ranges of the tilt values for the three different states. This suggests that the states can be distinguished reasonably accurately even under moderate uncertainty in the sensor's readings.

Table 4
Light sensor values (lumen) for different context states.

Avg. range of light level (lumen)	Context state
LightSensor.getValue() = 10 to 50	Turned on → active
LightSensor.getValue() = 0 to 1	Turned off → sleeping

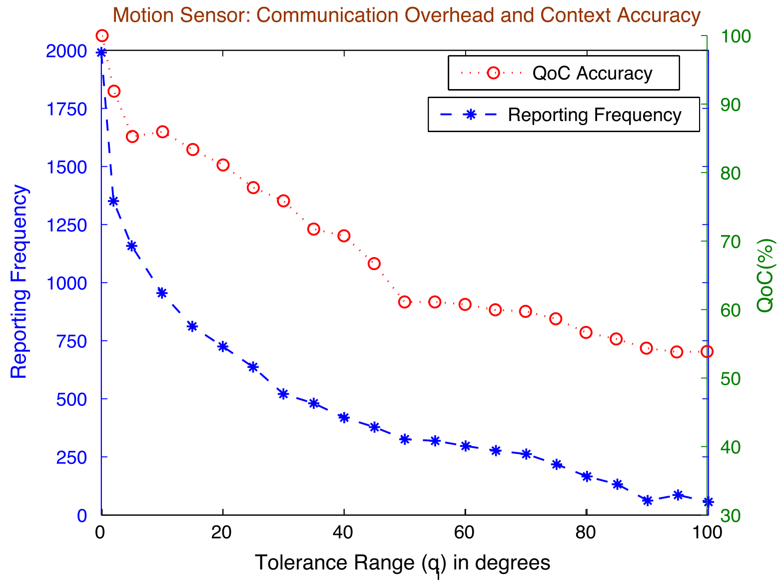

Fig. 8. Communication overhead and QoC accuracy vs. Tolerance range using motion sensor.

Similarly, we also used the SunSPOT light sensor to measure the light level for different user contexts. Intuitively, low values of ambient light intensity may be indicative of a *'sleeping'* state, while higher values of light intensity are likely to result when the individual is *'active'*. Table 4 shows the observed ranges for the light values for each of these two states. The accuracy of context from the light sensor is, however, much lower, as users may often be inactive (e.g., sitting), even under high illumination.

6.2. Measurement of quality of context accuracy and sensor overheads

To study the potential impact of varying the tolerance range on each sensor and the resulting tradeoff between the sensor reporting overhead, we collected traces for the SunSPOT motion and light sensors for a single user who engaged in a mix of three different activities (*sitting, walking and running*) for a total of ≈6 min (2000 samples at 5.5 Hz). We then used an emulator to mimic the samples that a sensor would have reported, given the trace, for a given q, and compared the context inferred from the values reported by the emulation against the ground truth. Fig. 8 shows the resulting plots for the 'total number of samples reported' (an indicator of the reporting overhead) and the corresponding accuracy (defined as 1-*error rate*) achieved, for different values of the tolerance range (q_m) for the motion sensor. Fig. 9 plots the corresponding values vs. the tolerance range (q_l) for the light sensor.

As the figures demonstrate, there is, in general, a continuous drop in the reporting overhead and the context accuracy as q increases. However, as seen in Fig. 8, a QoC accuracy of ≈80% is achieved for a modestly large q value of 40. Moreover, using this tolerance range reduces the reporting overhead dramatically by ≈85% (from 1953 → 248). This suggests that it is indeed possible to achieve significant savings in bandwidth, if one is willing to tolerate marginal degradation in the accuracy of the sensed context. A similar behavior is observed for the light sensor ($q = 4$ incurs a 5% loss in accuracy vs. a ≈65% reduction in reporting overhead). However, as the difference between the lumen ranges for *Active* vs. *Sleeping* is only ≈10 (Table 4), increasing q actually leads to a sharp fall in the accuracy.

6.3. The benefit of joint sensing

We also investigated how the use of readings jointly from both sensors affects the inferencing accuracy vs. tolerance ranges. We consider the individual to be in a *sitting, walking or running* state whenever the motion sensor tilt values lie within the corresponding range and the light sensor values indicate an *active* state. Fig. 10 uses a three-dimensional plot to illustrate the observed inferencing accuracy when the tuple (q_m, q_l) is jointly varied. This confirms the QoC is now less

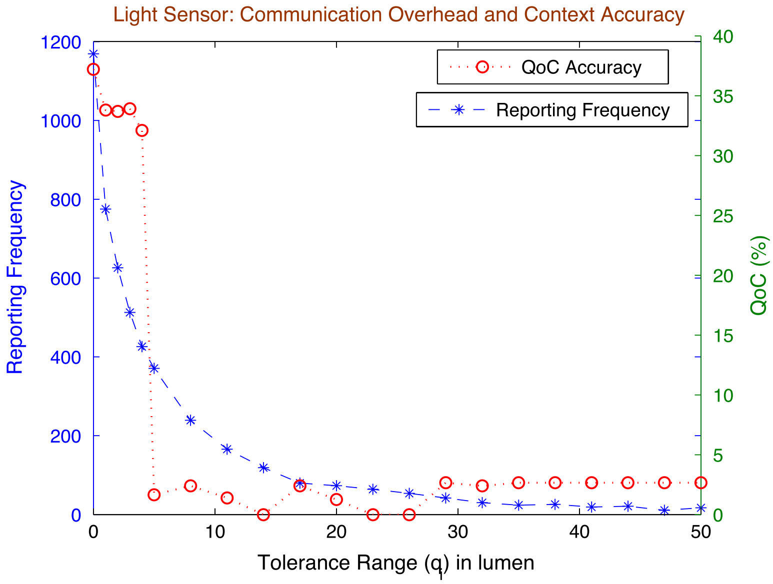

Fig. 9. Communication overhead and QoC accuracy vs. Tolerance range using light sensor.

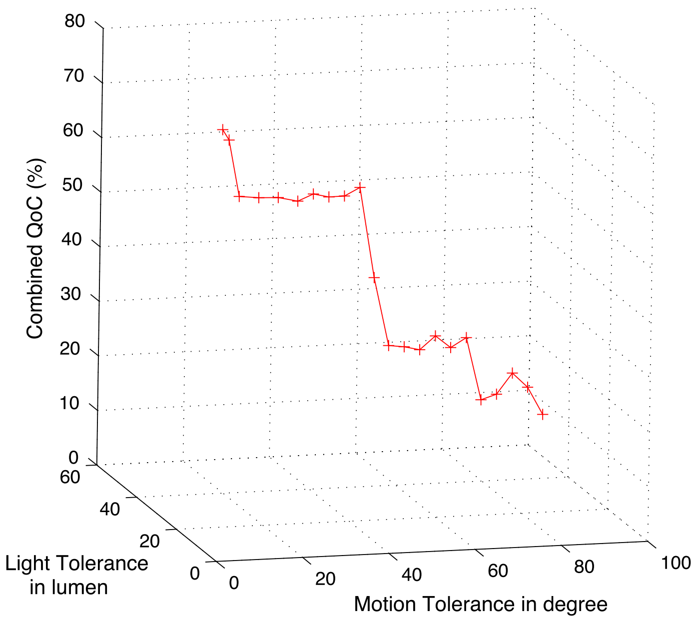

Fig. 10. QoC accuracy vs. Tolerance range using both motion and light sensor together.

susceptible to individual q variations. Fig. 11 confirms this benefit by plotting the QoC accuracy vs. q obtained using the light sensor against that obtained by using both light and motion sensors (the q ranges of both being identical). Clearly, the accuracy obtainable from the combination of the two sensors is much higher than that of a single sensor.

6.4. Evaluation of ambiguous context mediation

We also conducted experiments to evaluate the performance of the proposed ambiguous context mediation framework in a smart home health monitoring environment and report the results in this section. The ambiguous context mediation algorithm (ACMA) given in Fig. 4 was applied during our evaluation. In our application, the goal is to determine a set of sensors and the situation level (emergency or non-emergency) of a patient based on the most economically efficient disambiguation sensor action. Let us assume the situation level has three states, high, medium and low. Fig. 12 represents a snapshot of the Bayesian Network model for this application using Netica BN software [27]. In this figure we represent

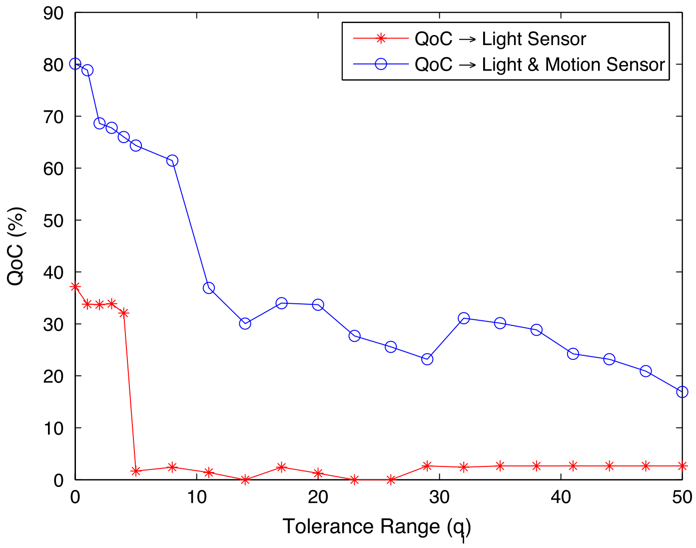

Fig. 11. Comparison of QoC accuracy improvement using multiple sensor.

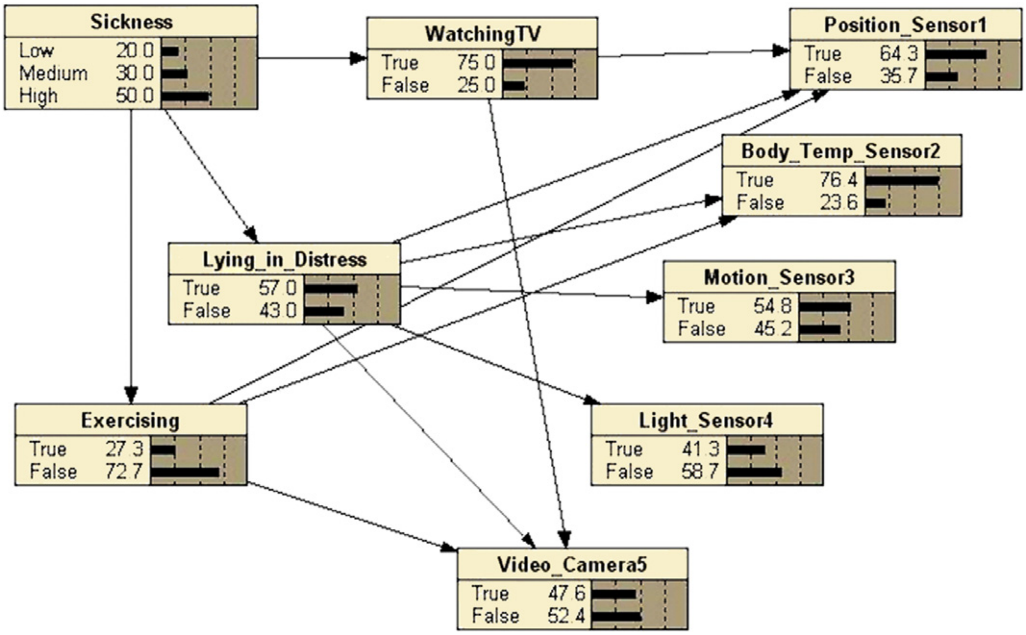

Fig. 12. Bayesian network.

a situation space sickness and three context states — WatchingTV, Lying_in_Distress and Exercising. The conditional probabilities of these context states are empirically derived using SunSPOT traces. The sensors selected by ACMA for this application are Position_Sensor1, Body_Temp_Sensor2, Motion_Sensor3, Light_Sensor4 and Video_Camera5. We empirically derived the conditional probabilities of the context attributes from the raw readings of the sensors, e.g., given the ground truth of patient exercising, how many times did the Body_Temp_Sensor report a reading above the threshold value. These conditional probability tables at a particular state of the application are all shown in Fig. 12.

From Fig. 13, we observe that the utility increases (reduces ambiguity) as the number of selected sensors increases for different states of the application. The initial utility is calculated using Eq. (12) considering a single sensor. The maximum utility values obtained by increasing sensor set size for three different states (different probability values). With different balance coefficients, the best set of sensors for an application having multiple states is also different. This confirms that the gain obtained by having more sensors exceeds the benefits of getting detailed information from each individual sensor in accordance to our fusion model.

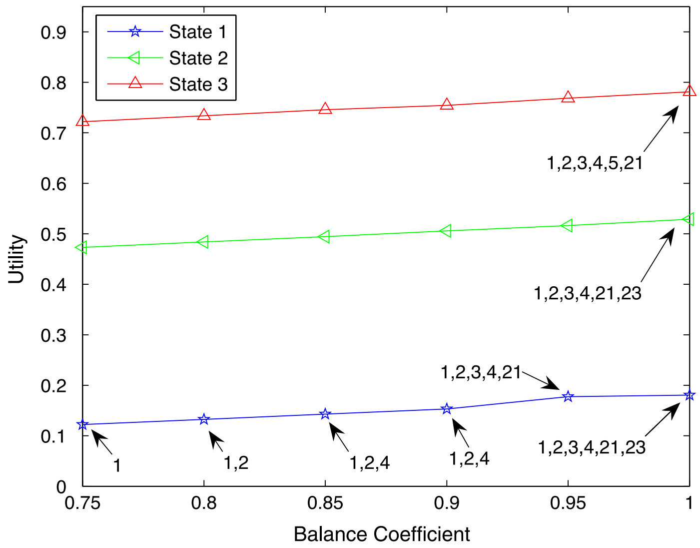

Fig. 13. Best sensor set for different values of balance coefficient α.

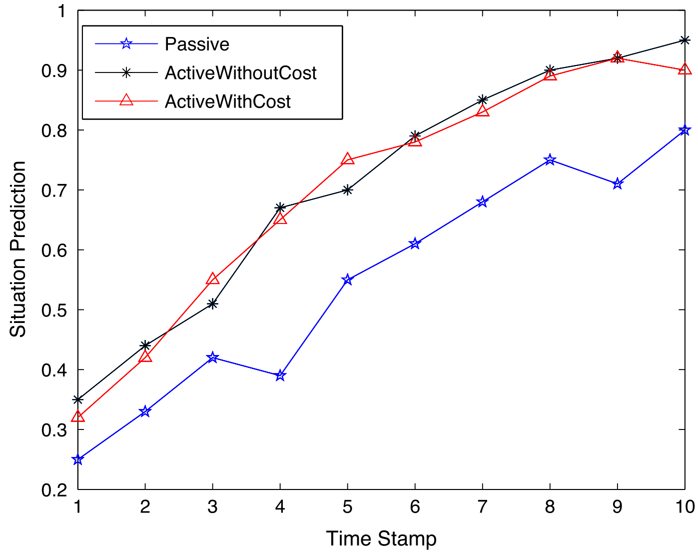

Fig. 14. Situation prediction probability using passive and active fusion.

Next we experimentally analyze the performance of active (context-aware) and passive (non-context-aware) fusion to illustrate how the proposed active fusion system basically works. The choice of which sensor to activate depends on the expected utility of each sensor. This repeats until we identify the situation type with sufficient confidence. We observe during our experiment that few sensors dominate in active fusion compared to the others. This repetition of sensors accelerates the decision to be taken on situation space compared to the passive fusion as shown in Fig. 14. But sometimes it leads to information redundancy if it repeats the same value of the attribute consecutively. However, it may be beneficial for reducing imprecision and increasing reliability.

Fig. 14 shows no significant performance difference by considering the acquisition cost. So the information redundancy can be overcome by frequently alternating between the active sensors with almost the same performance gain. Fig. 15 represents the confidence of situation prediction for different specified QoC constraints. The confidence level achieves a higher value for a rigid QoC constraint compared to a flexible system. Though we achieved a better confidence level for a tight QoC constraint, more uniformity is achieved for the loosely bound system. This observation confirms that the participation of more sensors during the non-rigid QoC bound fusion process yields a more stable value though fails to achieve a higher confidence gain. Next we examine the situation prediction when we selectively choose the different sensors using the context mediation algorithm. Fig. 16 depicts the variation of situation prediction with different sets of context attributes from different sensors. In the first scenario, all context attributes are fused following the specified algorithm according

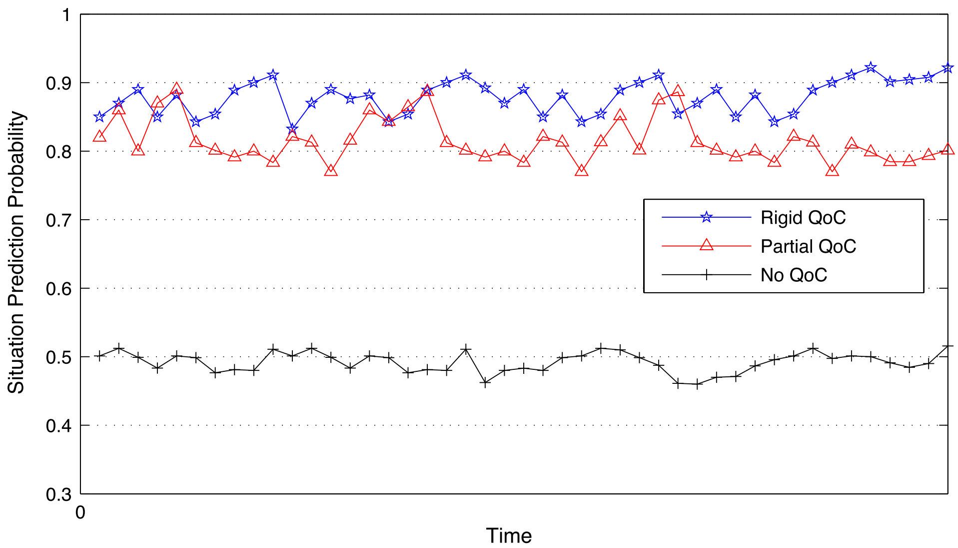

Fig. 15. Variation of situation prediction probability using QoC constraint.

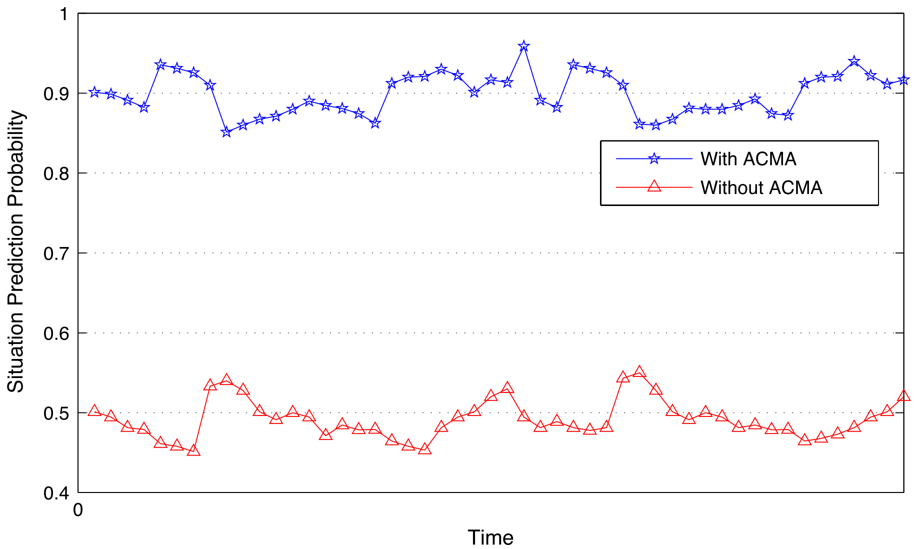

Fig. 16. Variation of situation prediction probability using sensor values.

to their QoC specification. In the second scenario, values are only partially satisfied due to their inherent inaccuracy and experimental settings. The fusion of selective context attributes yields better results compared to the non-selective one.

Through this evaluation we observed it is indeed possible to significantly reduce the sensors' resource usage while satisfying the application quality requirements in pervasive healthcare environments. It also attested the promise of context-aware multi-sensor fusion scheme for selecting a most economically efficient disambiguation action in a resource optimized quality assured model.

6.5. *Evaluation of context delivery*

We conduct simulations to evaluate the efficiency of context delivery as it requires a large number of nodes. The simulation is started by having a pre-existing node in the network generated in the AS model, and then performing a series of join operations invoked by new coming nodes. If a semantic cluster exceeds the maximum size M, it will be split into two and this operation may continue until the number of sub-clusters reaches 2^n. After the network reaches a certain size, a mixture of node joining and leaving are invoked to simulate the dynamic characteristic of the overlay network. We use the following performance metrics:

- *Search path length* is the average number of hops traversed by a query to the destination.
- *Search cost* is the average number of query messages incurred during a search operation.

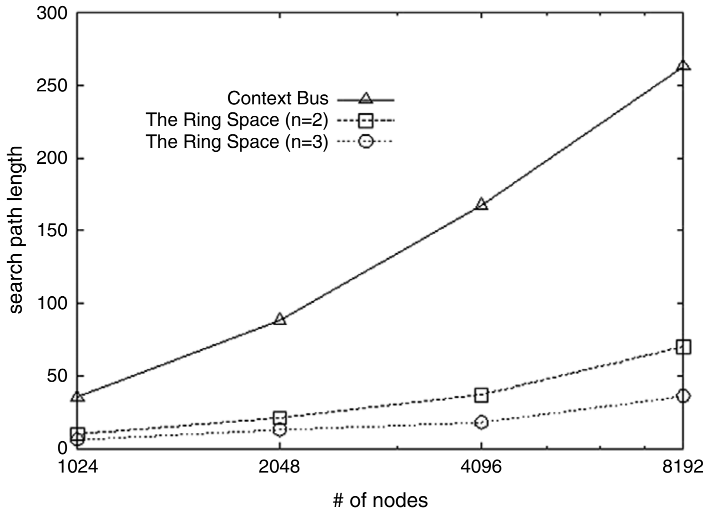

Fig. 17. Search effect.

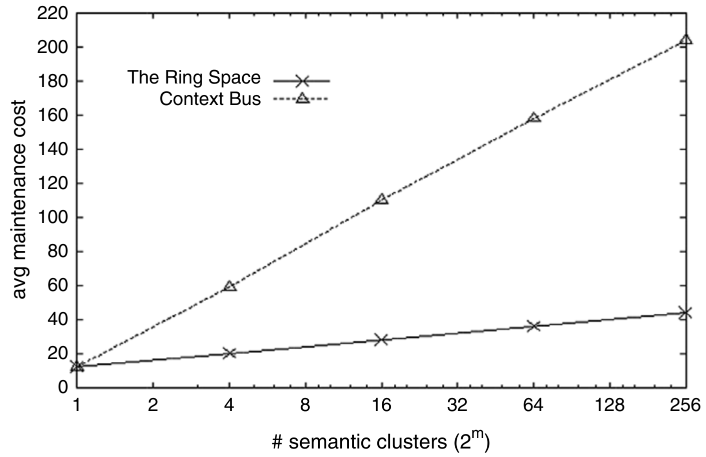

Fig. 18. Average maintenance cost.

- *Maintenance cost* is the average number of messages incurred when a node joins or leaves the network. It consists of the costs of node joining and leaving, cluster splitting/merging and index publishing. We measured these costs in terms of number of messages.

6.5.1. Search efficiency and overhead

The efficiency of executing a search request is captured in search path length during the search. In this experiment, we evaluate the search effect and compare it with ContextBus [28]. We set up a network with $m = 4$ and varied the network size from 210 to 213. We set $n = 2$ and 3 respectively, as a result a semantic cluster will be split into two when the size exceeds $N/25$ and $N/26$ where N is the network size. Fig. 17 shows that our system effectively reduces the search path length as compared to ContextBus.

We evaluate the average maintenance cost and show the result in Fig. 18. The maintenance cost for ContextBus increases rapidly when the number of semantic clusters grows. This is because the required number of outgoing degrees for a node in ContextBus increases in proportion to the dimension. The maintenance cost in our system also increases with respect to the dimension, but much more gradually. This confirms our design goal of reducing maintenance overheads.

6.5.2. Clustering effects

We evaluate the effect of clustering by varying the cluster size M from 20 to 210. We first evaluate the effect of cluster size on search path length by setting a network of size $N = 210$. Hence all clusters are semantic clusters. Fig. 19 plots the search path length in our system when M increases from 20 to 210. The search path length across clusters increases while the search path length within clusters decreases with larger cluster sizes (note that there are 210 clusters in the network

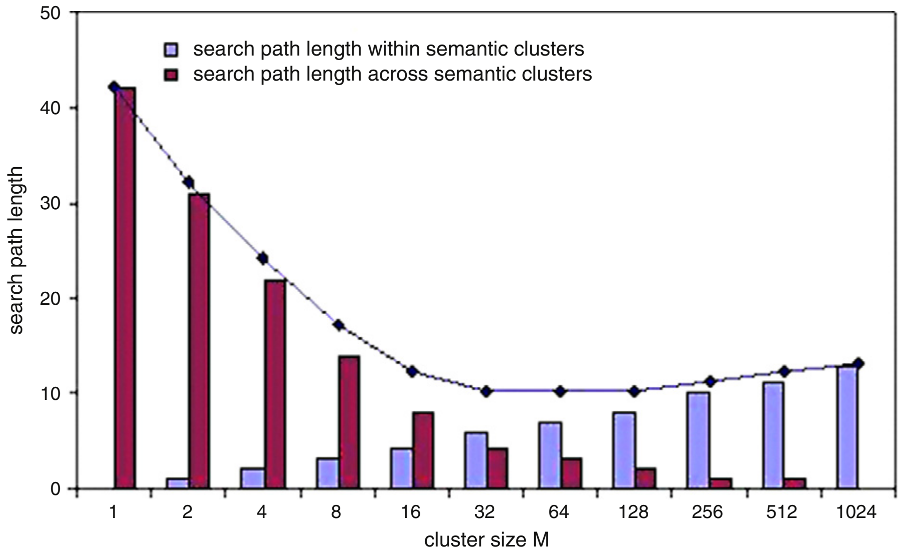

Fig. 19. Search path length vs. Cluster size M.

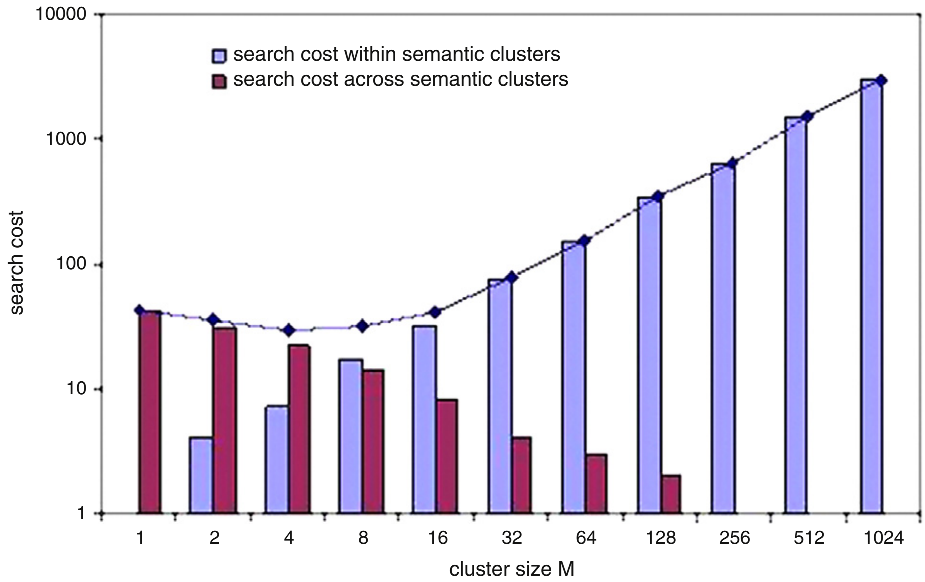

Fig. 20. Search cost vs. Cluster size M.

when $M = 1$ and only one cluster when $M = 210$). This is because with a fixed network size, the total number of clusters in our system decreases with larger cluster sizes.

With the same setting as in the previous experiment, we evaluated the search cost and its breakdown within clusters and across clusters with various cluster sizes. From Fig. 20, we observe that the search cost in our system increases rapidly from a point where $M = 16$. This is due to the effect of blind flooding within a cluster.

We plot the cost of node joining/leaving and cluster splitting/merging over different cluster sizes in Fig. 21. As there are lesser clusters with larger cluster sizes, a new node requires a smaller number of hops to join the network. Therefore the cost of joining/leaving decreases with respect to M. With a larger cluster size, cluster splitting and merging occur less frequently, resulting in a lower cluster splitting/merging cost.

7. Conclusion

This paper presents a framework which supports quality assured context fusion, semantic-based context delivery and user-centric situation prediction for pervasive computing applications. The proposed active context fusion model is able to selectively choose those information sources that are most informative with respect to the current context state while

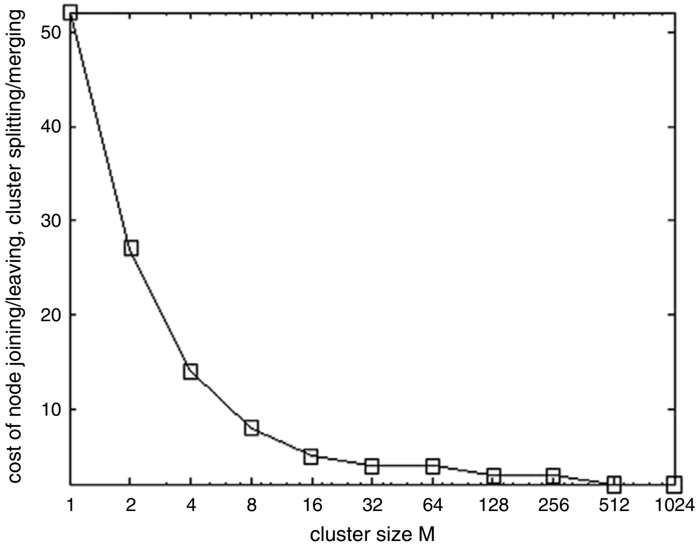

Fig. 21. The cost of node joining/leaving and cluster splitting/merging vs. Cluster size M.

minimizing the associated costs (i.e., computational complexity, time and required resources) and satisfying the QoC bound of an application. The context delivery model is able to perform intelligent clustering and delivery of context data according to the semantics. A SunSPOT context sensing system is developed and subsequent experimental evaluation is done.

Acknowledgements

This work is partially supported by the AFOSR Grant FA9550-08-1-0260, and NSF Grants IIS-0326505 and CNS-0721951. The work of S.K. Das is also supported by (while serving at) the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] N. Roy, A. Roy, S.K. Das, Context-aware resource management in multi-inhabitant smart homes: A nash H-learning based approach, in: Proc. of IEEE Int'l Conf. on Pervasive Computing and Communications, PerCom, March 2006, pp. 148–158. Mark Weiser Best Paper Award (Extended version appeared in Pervasive and Mobile Computing Journal, vol. 2, Issue 4, pp. 372–404, November 2006.).

[2] N. Roy, A. Misra, S.K. Das, C. Julien, Quality-of-Inference QoINF-aware context determination in assisted living environments, in: Proc. of ACM SIGMOBILE Workshop on Medical-Grade Wireless Networks WiMD 2009 in Conjunction with MobiHoc, May 2009.

[3] N. Roy, A. Misra, S.K. Das, Efficient long-term Quality-of-Inference QoINF-aware context determination in pervasive care environments, in: Proc. of ACM SIGMOBILE Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments HealthNet 2008, Poster Paper.

[4] A. Padovitz, S.W. Loke, A. Zaslavsky, Towards a theory of context spaces, in: Proc. of the Second IEEE Conference on Pervasive Computing and Communications Workshop, Orlando, Florida, March 2004.

[5] A.K. Dey, D. Salber, G.D. Abowd, A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, in: Context-Aware Computing, Human-Computer Interaction HCI Journal 16 (2–4) (2001) 97–166 (special issue).

[6] A.K. Dey, J. Mankoff, G.D. Abowd, S. Carter, Distributed mediation of ambiguous context in aware environments, in: Proc. of the 15th Annual Symposium on User Interface Software and Technology UIST 2002, October 2002, pp. 121–130.

[7] A. Cheyer, L. Julia, Multimodal maps: An agent-based approach, in: Proc. of the International Conference on Cooperative Multimodal Communication, Netherlands, May 1995, pp. 103–113.

[8] B. Rhodes, The wearable remembrance agent: A system for augmented memory, in: Wearable Computing, Personal Technologies Journal 1 (1997) 218–224 (special issue).

[9] C. Guanling, Solar: Building a context fusion network for pervasive computing, Ph.D. Dissertation, Department of Computer Science, Dartmouth College, August 2004.

[10] Jason I. Hong, James A. Landay, An infrastructure approach to context-aware computing, Human-Computer Interaction 16 (2001).

[11] M. Roman, C.K. Hess, R. Cerqueira, A. Ranganathan, Roy H. Campbell, K. Nahrstedt, Gaia: A middleware infrastructure to enable active spaces, IEEE Pervasive Computing (2002) 74–83.

[12] R. Kumar, M. Wolenetz, B. Agarwalla, J. Shin, P. Hutto, A. Paul, U. Ramachandran, DFuse: A framework for distributed data fusion, in: Proc. of the 1st International Conference on Embedded Networked Sensor Systems, November 2003, pp. 114–125.

[13] C. Olston, J. Widom, Offering a precision performance tradeoff for aggregation queries over replicated data, in: Proceedings of VLDB, Margan Kaufmann Publishers Inc., 2000, pp. 144–155.

[14] A. Deshpande, C. Guestrin, S. Madden, J.M. Hellerstein, W. Hong, Model-based approximate querying in sensor networks, Int'l Journal on Very Large Data Bases (VLDB Journal) (2005).

[15] M.C. Huebscher, J.A. McCann, Adaptive middleware for context-aware applications in smart-homes, in: Proc. of the 2nd Workshop on Middleware for Pervasive and ad-hoc Computing, 2004, pp. 111–116.

[16] W. Heinzelman, A.L. Murphy, H.S. Carvalho, M.A. Perillo, Middleware to support sensor network applications, IEEE Network (2004) 6–14.

[17] A. Dielmann, S. Renals, Dynamic Bayesian networks for meeting structuring, in: Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'04, 2004.

[18] O. Brdiczka, P. Reignier, J.L. Crowley, D. Vaufreydaz, J. Maisonnasse, Deterministic and probabilistic implementation of context, in: Proc. of IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOMW'06, 2006.
[19] H. Alex, M. Kumar, B. Shirazi, MidFusion: An adaptive middleware for information fusion in sensor network applications, Elsevier Journal of Information Fusion (2005).
[20] W. Hu, A. Misra, R. Shorey, CAPS: Energy-efficient processing of continuous aggregate queries in sensor networks, in: Fourth IEEE International Conference on Pervasive Computing and Communications, PerCom, 2006, pp. 190–199.
[21] A. Deshpande, C. Guestrin, S. Madden, Using probabilistic models for data management in acquisitional environments, in: Proceedings of the 2nd Biennial Conference on Innovative Data Systems Research, CIDR, January 2005, pp. 317–328.
[22] Y. Zhang, Q. Ji, C.G. Looney, Active information fusion for decision making under uncertainty, in: Proc. of the Fifth International Conference on Information Fusion, vol. 1, 2002, pp. 643–650.
[23] J.C. Augusto, C.D. Nugent, N.D. Black, Management and analysis of time-related data in smart home environments, in: Proc. of Mediterranean Conference on Medical and Biological Engineering Medicon'04, Ischia, Italy, 2004.
[24] RDQL, http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/.
[25] J. Kleinberg, The small-world phenomenon: An algorithm perspective, in: Proc. of the 32nd ACM Symposium on Theory of Computing, 2000.
[26] SunSpotWorld—Home of Project SunSPOT: www.sunspotworld.com/.
[27] Netica Bayesian network software: http://www.norsys.com.
[28] T. Gu, E. Tan, H.K. Pung, D. Zhang, ContextPeers: Scalable peer-to-peer search for context information, in: Proc. of the International Workshop on Innovations in Web Infrastructure IWI'05, in Conjunction with the 14th World Wide Web Conference WWW'05, Japan, May 2005.
[29] N. Roy, G. Pallapa, S.K. Das, A middleware framework for ambiguous context mediation in smart healthcare application, in: Proc. of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications WiMob 2007, 2007, p. 72.

Nirmalya Roy is currently a postdoctoral research fellow in Mobile and Pervasive Computing group in Electrical and Computer Engineering Department at the University of Texas at Austin. He received his MS and Ph.D. degrees in Computer Science and Engineering from the University of Texas at Arlington in 2004 and 2008, respectively. He received his B.E. degree in 2001 from Jadavpur University, Calcutta. His research interests include context-aware resource management in mobile, pervasive and grid computing environment. More information about him can be found at http://crewman.uta.edu/~nirmalya. He is a member of the IEEE.

Tao Gu is currently an Assistant Professor in the Department of Mathematics and Computer Science at University of Southern Denmark. He received his Bachelor's degree from Huazhong University of Science and Technology, and Master's degree from Nanyang Technological University, Singapore, and Ph.D. in computer science from National University of Singapore. His current research interests involve various aspects of pervasive computing and wireless sensor networks. More information about him can be found at http://www.imada.sdu.dk/~gu. He is a member of the IEEE.

Sajal K. Das is a University Distinguished Scholar Professor of Computer Science and Engineering and the Founding Director of the Center for Research in Wireless Mobility and Networking (CReWMaN) at the University of Texas at Arlington (UTA). He is also an E.T.S. Walton Professor of Science Foundation of Ireland; a Visiting Professor at the Indian Institute of Technology (IIT) at Kanpur and IIT Guwahati; an Honorary Professor of Fudan University in Shanghai and International Advisory Professor of Beijing Jiaotong University, China; and a Visiting Scientist at the Institute of Infocomm Research (I2R), Singapore. His current research interests include wireless sensor networks, mobile and pervasive computing, design and modeling of smart environments, pervasive security, smart health care, resource and mobility management in wireless networks, mobile grid computing, biological networking, applied graph theory and game theory. He has published over 400 papers and over 35 invited book chapters in these areas. He holds five US patents in wireless networks and mobile Internet, and coauthored the books "Smart Environments: Technology, Protocols, and Applications" (Wiley, 2005) and "Mobile Agents in Distributed Computing and Networking" (Wiley, 2008). Dr. Das is a recipient of several Best Paper Awards in such conferences as EWSN'08, IEEE PerCom'06, and ACM MobiCom'99. He is also recipient of the IEEE Technical Achievement Award (2009), IEEE Engineer of the Year Award (2007), UTA Academy of Distinguished Scholars Award (2006), University Award for Distinguished Record of Research (2005), College of Engineering Research Excellence Award (2003), and Outstanding Faculty Research Award in Computer Science (2001 and 2003). Dr. Das serves as the Founding Editor-in-Chief of Pervasive and Mobile Computing (PMC) journal, and Associate Editor of IEEE Transactions on Mobile Computing, ACM/Springer Wireless Networks, IEEE Transactions on Parallel and Distributed Systems, and Journal of Peer-to-Peer Networking. He is the founder of IEEE WoWMoM and co-founder of IEEE PerCom conference. He has served as General or Technical Program Chair as well as TPC member of numerous IEEE and ACM conferences. More information about him can be found at http://crewman.uta.edu/~das. He is a senior member of the IEEE.