Context-aware and resource efficient sensing infrastructure for context-aware applications

Peizhao Hu*†, Suan Khai Chong‡, Jadwiga Indulska*† and Shonali Krishnaswamy‡

* *The University of Queensland, Australia,*
School of Information Technology and Electrical Engineering
† *National ICT Australia (NICTA)*
‡ *Caulfield School of Information Technology,*
Monash University, Australia
Corresponding author email: Peizhao.Hu@nicta.com.au

***Abstract*—Middleware for wireless sensor networks and middleware for context-aware applications both provide information abstraction and programming support for gathering, pre-processing, and managing sensor data. However the former mostly concentrates on optimising the operations of the resource constrained hardware and simplifying access to the raw sensor data while the latter focuses on gathering sensor data, pre-processing it to the abstract context information required by the applications and providing reasoning on this data. In this paper, we explore the idea of enhancing middleware for context-aware applications with solutions from sensor networks middleware to allow resource efficient and context-aware management of sensing infrastructure. The decisions on which sensor data needs to be delivered to the middleware for evaluation are based on current contextual situations. The approach allows to trade the level of confidence in context information for resource efficiency in context provisioning without a detrimental effect on the functionality of context-aware applications.**

I. INTRODUCTION

Wireless sensing technologies are deployed in many real-world applications, such as automation, surveillance and inventory management [14]. The last decade has also seen a great progress on pervasive computing and in particular on context-aware applications that can adapt their behaviour to changes in the computing environment, user environment, user activities and preferences. Decisions about such adaptions are based on evaluation of context information which has to be gathered, evaluated and reasoned upon. Data produced by sensors is one type of context information that these applications may require.

To ease the complexity of software engineering of context-aware applications, the research in pervasive computing resulted in the development of various types of middleware. These middleware solutions encompass a rich set of features, including (i) general middleware operations e.g., context information gathering and management,(ii) support for advanced reasoning on context, e.g., first order logic or description logic (ontology) reasoning, and (iii) programming abstractions that ease development of context-aware applications. However, these solutions typically do not provide resource efficient context information gathering from large groups of sensors or large scale sensor networks in which individual sensors may not have global identification [1]. On the other hand, the scalability issues have been explored by the wireless sensor networks (WSN) community. There exist middleware solutions that have been specifically developed to cater for large scale sensor network deployments. The goal of these middleware solutions is primarily to facilitate data extraction from, potentially hundreds of thousands of resource constrained sensing devices [10]. Due to the scale of deployment, efficiency of operations and management are the main concern in the design and development of these middleware solutions. However, they usually lack awareness of the application requirements and operational objectives.

These two types of middleware were developed in separation by different communities and with different objectives. However, enhancing one's functionalities with the incorporation of techniques from another [10] can bring unique advantages. For example, providing application-level situation awareness to the low-level sensing infrastructures that support middleware for context-aware computing will improve efficiency of resource management. It will allow to trade the level of confidence in context information for resource efficiency in context provisioning without a detrimental effect on the functionality of context-aware applications.

In this paper we present an enhancement of a middleware for context-aware applications that uses a resource-efficient data collection algorithm developed for sensor networks and allows the middleware to gather context information from single sensors and from sensor networks while meeting the application requirements for context data quality. We have designed this extension and also implemented part of it as a proof of concept prototype. The design and prototype are developed as an extension of the ACoMS middleware [11] that is a middleware for reliable provisioning of context information. While the proposed enhancement is described as an extension of the ACoMS, the solution is generic and would be suitable for most logic-based middleware for

978-1-4244-5328-3/10/$26.00 ©2010 IEEE

context-aware applications (i.e. middleware that uses context information models, logic based reasoning on contextual situations, preference models, and sensor models).

The structure of the paper is as follows. Section II presents an example scenario that demonstrates the usefulness of our proposed middleware enhancement. Section III briefly describes the two primary components in this investigation, ACoMS and an algorithm developed for sensor networks to efficiently collect sensor data. Section IV describes necessary extensions to the middleware and the algorithm in order to achieve context-aware collection of sensor data. Section V reviews existing work focussing on resource preservation in the existing middleware for context-aware applications. Finally, we conclude in Section VI.

II. Motivating scenario

In this section, we describe an example scenario to elaborate the enhancement of the proposed middleware for context-aware applications (that is, making the sensing infrastructure context-aware and resource efficient while ensuring the Quality of Information of produced sensor data). The scenario illustrates the idea of dynamically adapting behaviour of the sensing infrastructure to the applications' operational objectives. As a result the middleware supports efficient resource management and addresses scalability issues with regard to adapting large scale sensor networks as the sensing infrastructure for pervasive computing.

A building is being constantly monitored for detecting potential fire hazard. The building is equipped with sensing devices for monitoring object movement, temperature and smoke density level. In the daily operations when readings are normal, the selection of a sensor set should be optimised (i.e., putting some sensors into sleep mode) to preserve limited resources, such as communication bandwidth and battery life. In the case of emergency (detected by abnormal sensor readings), the system should adapt the set of sensors to improve situation awareness; that is, the system may in turn wake up additional sensors and collect their readings or even fuse sensors' data to increase the confidence of their observations. The set of sensors chosen for the monitoring task should be dynamically adapted to the applications' operational objectives (in this example scenario, preserve energy or improve certainty of situation recognition).

III. Main system components

Our goal is to enhance the middleware supporting context-aware applications in order to make its sensing context-aware and resource efficient. We used the PACE/ACoMS middleware for this enhancement. There are two reasons for this choice:

- The PACE middleware [8], [9] is a platform for gathering, evaluating and disseminating context information to context-aware applications. Its aim is to ease the development of context-aware applications through its comprehensive modelling techniques and programming support/abstraction. The formal models of context, and also situation and preference models allow to move the evaluation of context information from applications to middleware simplifying development of context-aware applications. The PACE has been already extended to the ACoMS [11] that can provide reliable context provisioning, i.e. can dynamically replace sensors when the sensors fail or the Quality of Information (QoI) of context information they provide does not meet the application requirements. The replaced sensors can be of different kind provided that the data they provide can be pre-processed to the abstract context information required by the applications (e.g., a location technology can be at run time replaced by a different location technology). This extension added, among others, sensor models and models of preprocessing of sensor data to the original PACE models. As the platform is model based it can be further extended to provide a situational awareness at the level of sensor data gathering.
- We have access to the PACE/ACoMS middleware[1] and therefore we are able to develop a proof of concept prototype that shows how some algorithms for energy efficient sensor data retrieval in sensor networks can support the middleware for context-aware applications.

In this section, in order to provide a background for the description of the proposed enhancement, we briefly describe the models and architecture of the PACE/ACoMS middleware (which for simplicity we will call ACoMS in the rest of the paper). We also describe an algorithm developed for energy efficient retrieval of data in sensor networks that will be used in our proposed middleware enhancement presented in Section IV.

A. The ACoMS middleware

The architecture of the ACoMS middleware is shown in Figure 1. In the ACoMS, application developers describe context information used in the application in the form of context models (denoted as `Application Context Model` in Figure 1) using a modelling language called CML (Context Modelling Language), which is developed based on ORM (Object Role Modelling) [7]. The CML modelling approach leverages the graphical notations to represent the information and their relationships. An example of the CML context model is provided in the following section. In addition, the middleware uses models of (i) contextual situations (higher level abstraction defined on context facts), (ii) user preferences that need to be evaluated when such situations are detected, and (iii) adaptation rules triggered by the situations (shown as `Situation Models`, `Preferences` and `Adaptation Rules` in Figure 1, respectively).

[1]Refer to http://sourceforge.net/projects/pace-framework/

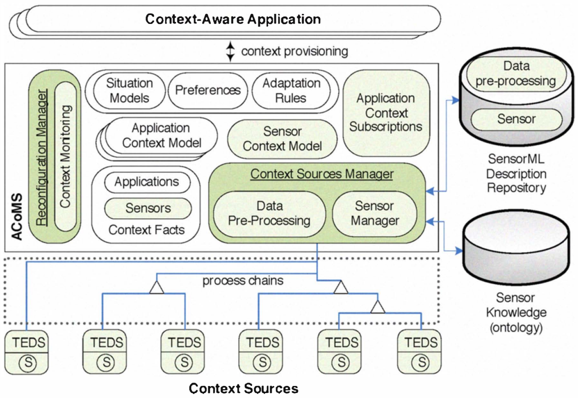

Figure 1. The ACoMS system architecture.

The application context models abstract raw data gathered from the sources of context information (i.e., sensors). Such raw data may need to be pre-processed to acquire the form defined by the context model (abstract context fact). If run-time replacement of a context information source is needed or the sources of context information are to be dynamically configured/activated when the applications start, then a mapping is required from the context facts to the appropriate sources of raw context data (i.e., sensors), through the appropriately assigned data pre-processing models. The sensor and pre-processing models required to support this mapping are shown in Figure 1 in grey and are described in [11].

The system maintains context models for each application it serves. Heterogeneous context sources (i.e., sensors) provide the system at run-time with fact instances (i.e., abstracted from sensor data) that conform to the application context models. This allows reuse of context information by many applications and reduces the burden on resource-constrained sensors and communication networks. For example, if many applications of the same user require location information, each application will include location context fact in its model but they will share one fact instance (location reading for the user).

B. The HiCoRE algorithm

The HiCoRE [4] is a mining algorithm developed for sensor networks that mines for highly correlated rules from gathered sensor data at aggregation points (i.e., base station). A highly correlated rule signifies the relationships between attributed sensor nodes, which can be used to infer sensors' data and reduce the amount of sampling required (as long as correlation rules hold). The HiCoRE algorithm is presented, as *pseudo-code* in Algorithm 1.

The HiCoRE algorithm takes a batch of frequent transactions, b_j, (that is, the set of sensor data that is fed into the algorithm) and computes a correlation rule R. The **steps 1-2** initialise the algorithm and define variables

Algorithm 1 The HiCoRE Algorithm: Miner

1: SET `FT_List, Top_FT;`
2: SET `maxSupport, highestSupport, thresholdSupport;`
3: Obtain energy levels of sensors in S and sort them in ascending order of energy levels, sorted energy lists $Energys_S = e_0, e_1, ..., e_q$.
4: Generate the covariance matrix, C_{matrix}.
5: Using a bitmap, initialise two sensors in S with greatest probability measure from C_{matrix} and $Energys_S$.
6: Transpose continuous transaction values in b_j to discrete values.
7: **for** $i = 1$ to $length(\texttt{FT\_List})$ **do**
8:     $\texttt{currentSupport} = \frac{\texttt{FT\_Count}(t_i)}{length(b_j)}$
9:     **if** `currentSupport` > `maxSupport` **then**
10:         `maxSupport = currentSupport`
11:     **end if**
12: **end for**
13: **if** `highestSupport >= thresholdSupport` **then**
14:     $R = getRules(\texttt{Top\_FT})$
15: **else if** Number of bits set > 2 **then**
16:     **if** all bits set **then**
17:         Reset all bits to 0
18:     **else**
19:         Remove one bit reflecting current highest correlation in matrix C_{matrix}
20:     **end if**
21: **end if**
22: Return R.

that are needed to store the frequent transactions and their respective counts; these include `FT_List` that is a list of the frequent transactions, `FT_Count` (t_i) that represents the occurrence count of a frequent transaction t_i, and `Top_FT` that denotes the most frequent transaction in the `FT_List`. Following this, **step 3** is responsible for keeping track of the energy levels of sensor nodes in the sensor group, S, which is updated at each algorithm iteration. The covariance matrix, C_{matrix}, for all attributes of sensors in the group is generated in **step 4**. In **step 5**, a binary bitmap is used for the algorithm to give preference to highly correlated sensor attributes in the covariance matrix generated and sensors with the biggest variance in their energy levels. Here, we also wish to give preference to choosing antecedent sensors that have the greatest energy level to conserve energy on low-energy consequent sensors. In **step 6**, continuous sensor values are transposed to discrete values to generate rules and to reduce processing complexity. The **steps 7-12** then obtain the `highestSupport` from transactions already in the `FT_List`. In the final pass (**steps 13-21**), the rules that meet the user-defined `thresholdSupport` are generated. After these rules have been generated, they can then be filtered by a rule confidence threshold and filtered rules could then be used by the aggregator to control operations of sensors that send data to it.

Once the correlation rule is generated, the consequent sensor readings can be inferred, as long as the antecedent sensor values remain. For example, let us assume the following rule exists in the system that consists of two multi-modality sensors (capable of measuring light intensity and ambient temperature), sensor S_1 and S_2. The system can then decide whether any of these sensors can be put into sleep to preserve local or global resources (e.g., communication bandwidth, power).

$$S_2^{Temperature} : mid \wedge S_2^{Light} : low \rightarrow S_1^{Light} : low$$

As long as the antecedent sensor values, $S_2^{Temperature}$ and S_2^{Light}, hold, the system can then infer sensor S_1^{Light} readings and put it into sleep mode. Therefore, less communication bandwidth as well as transmission power are required to propagate raw data from individual sensor nodes to the higher level components. More information about HiCoRE and its performance evaluation is described in [4].

IV. A CONTEXT-AWARE AND RESOURCE EFFICIENT CONTEXT MANAGEMENT SYSTEM

To achieve context-aware provisioning of sensor data we designed the ACoMS extension; the HiCoRE algorithm is one element in this extension. Figure 2 shows the new architecture — ACoMS+ which consists of three components:

The HiCoRE algorithm mines correlated rules from observed sensor data and ranks these discovered rules based on the *ranking metrics*. These ranking metrics are defined by application designers for a set of application-specific objectives and are a function of sensor's physical characteristics and specifications (e.g., energy level, power consumption, sensitivity). In addition, application designers may also specify the *fusion logic* to deal with cases when more than one sensor fulfil the requirements of the information provisioning task and high confidence of context information is needed.

The information quality evaluator calculates the run-time information quality of actual sensor data using techniques of information fusion; for example, computing the entropy or certainty of resulting information against the information quality requirements specified by applications. Information about the quality evaluation (including techniques of choice and adaptation thresholds) is optionally specified as the *QoI policies* by application designers.

The ACoMS framework provides context information provisioning services to multiple context-aware applications(as described in the previous section) and in addition provides such services to the HiCoRE algorithm and the information quality evaluator, for their context-aware operations.

The addition of the HiCoRE algorithm and the information quality evaluator allows to capture the multidimensional information quality metrics of a sensor network; that is, it allows the ACoMS+ to rank the discovered correlation rules based on sensors' specifications, and it supports verification of information quality based on sensors' real-time observations. This hybrid approach is needed for sensor driven systems, as the QoI of sensing data not only depends on properties of the sensor fusion algorithm, but also depends on the quality of raw data received from individual sensor nodes [13], [15]. The quality of raw sensor data in turn depends on various sensor's physical characteristics and specifications, such as sampling rate, accuracy.

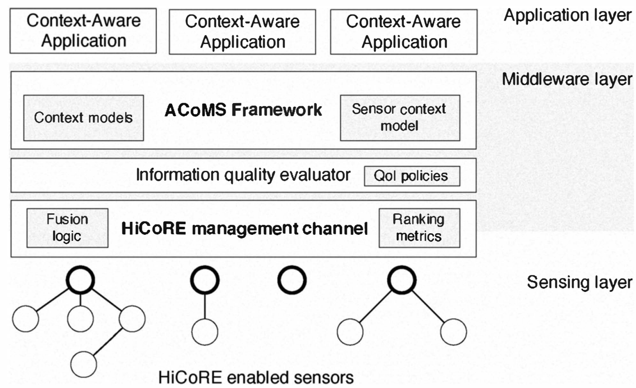

Figure 2. The ACoMS+ system architecture.

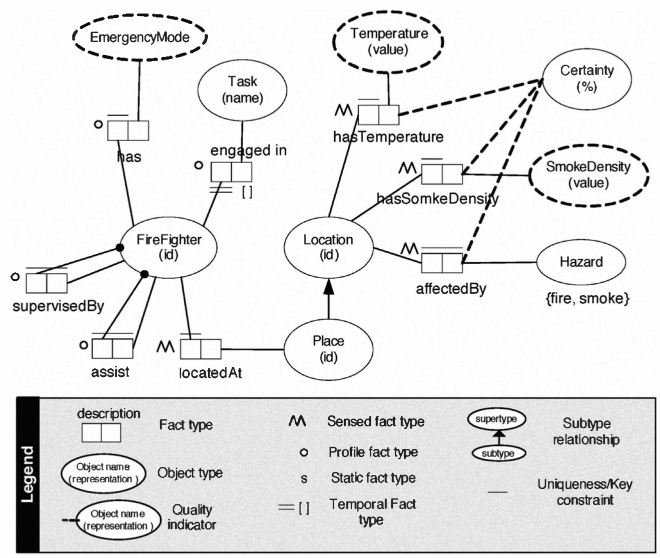

Figure 3. The application context model (*simplified*).

In the ACoMS+ architecture, the ACoMS framework supports correlation rule mining and information quality verification by providing relevant application context information (via *application context models*) and sensor specification metadata (via the *sensor context model*).

The following sections discuss each of the three components in more detail.

A. Scenario based ACoMS models

We will use the scenario presented in Section II to illustrate how the ACoMS framework can provide context-awareness to the sensing infrastructure. Figure 3 illustrates the application context model for this scenario.

This application context model captures a range of context information including: (i) the relationship between entities (people, places), and (ii) the properties and activities of entities. It has context facts of two types: *profiled* and *sensed* fact types. Profiled information is user-supplied, and is therefore initially very reliable, but often becomes out of date, while sensed context information is usually highly dynamic and prone to noise and sensing errors. This classification of information types allows context information to be managed and processed according to the characteristics of its type. Another important property of the context modelling approach that is shown in this example is its ability to capture Quality of Information. The *certainty* metadata of a context fact type (location `hasTemperature` temperature) indicates the required confidence of the individual fact instance gathered from the sensors. The information quality evaluator uses this quality requirement to verify whether information supplied by the given set of sensors (selected by a correlation rule) fulfils the needs of the applications.

The ACoMS framework allows definition of *situations* that require adaptation. Each situation is defined using the basic context fact types from the application context models. The ACoMS uses a variant of first order logic for defining situations as illustrated below in two example situations for the presented scenario:

$$
\begin{aligned}
fireHazard(loc) :\ & \\
& \exists event \bullet affectedBy[loc, event] \\
& \bullet\ event = \text{“}fire\text{''} \\
& \wedge\ hasTemperature(loc, temp) \\
& \wedge\ temp > 60degC \\
highSmokeLevel(loc) :\ & \\
& \exists smkDensity \\
& \bullet\ hasSmokeDensity[loc, smkDensity] \\
& \bullet\ smkDensity = \text{“}high\text{''}
\end{aligned}
$$

The ACoMS framework also provides a way for developers to specify preferences. Example preferences for the scenario are given below:

$$
\begin{aligned}
p1 = & \\
& \texttt{when}\ emergency(event) \\
& \ \wedge\ highSmokeLevel(loc) \\
& \texttt{rate}\ 0.8 \\
p2 = & \\
& \texttt{when}\ emergency(event) \\
& \ \wedge\ fireHazard(loc) \\
& \texttt{rate}\ \overline{\wedge}
\end{aligned}
$$

where $\overline{\wedge}$ stands for *obligation.*

These preferences tune the degree to which each element contributes to emergency situations — $highSmokeLevel$ and $fireHazard$. The preference model allows customised tuning of individual applications to different circumstances. For example, the preference $p2$ states that it is certainly an emergency given that there is a fire hazard at loc.

Situation detection triggers adaptations and these can also include an adaptation of the sensing infrastructure. For example, the HiCoRE algorithm can be adapted from low energy sensing used for daily monitoring purposes (when only data from a small subset of sensors selected based on the HiCoRE correlation rules is delivered to the ACoMS+ middleware) to a thorough sensing required for situation awareness in case of emergency.

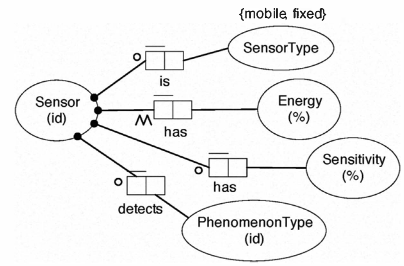

Figure 4. The sensor context model (*simplified*).

The ACoMS, in addition to the application context models also uses models of sensors. Figure 4 shows a simplified sensor context model which captures sensor's specifications and relevant physical characteristics. This information can be used by the HiCoRE algorithm to perform correlation rules mining and to estimate a quality score for each discovered correlation rule.

B. Context-aware HiCoRE

The extended HiCoRE algorithm uses context information provided by the ACoMS framework to perform correlation rule mining. The context-aware operations include: (i) the use of context information of individual applications to mine correlation rules that are dependent on the applications' operational objectives (e.g., selectively mine sensor data for particular information type required by applications). An application may define a number of operational objectives (such as to preserve energy or to improve information accuracy), therefore, the criteria for correlation rule mining should be appropriately adapted to the desired objective of the applications; (ii) the use of context information of sensors (e.g., specifications) to rank the discovered correlation rules based on each application operational objective; and (iii) the use of situation based triggering to change the objectives (e.g., from energy preservation to better situation awareness).

Figure 5 shows the relevant components in Figure 5(a) (with reference to Figure 2) and an example of the correlation rules mining and information quality evaluation in the ACoMS+, in Figure 5(b).

Figure 5(b) shows an example of the HiCoRE's correlation rule mining. In this example, we assume that there are n sensors $S_1, S_2, ..., S_n$, and each sensor produces a measurement of its type (e.g., temperature). Upon receiving sensor data from the sensor nodes, the HiCoRE algorithm mines correlation rules according to a set of application specific objectives (e.g., to preserve energy, O_1, or to improve information accuracy, O_2). Let assumes the discovered correlation rules are $R_{1,1}, R_{1,2}$ and $R_{1,3}$ for the energy

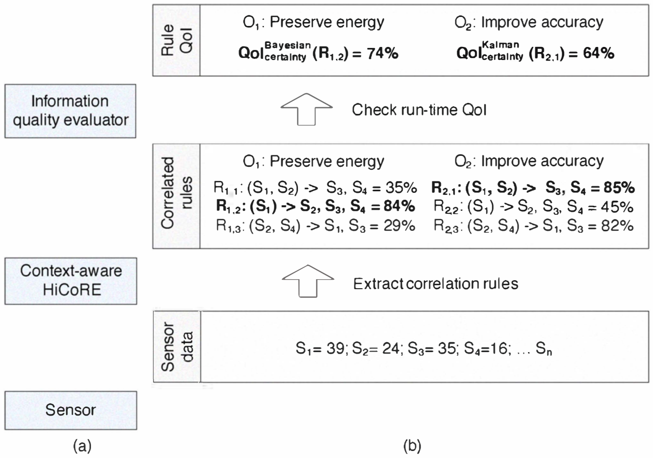

Figure 5. An example of context-aware operations in the ACoMS+.

conservation goal O_1. The rule $R_{1,2}$ states sensors S_1 and S_2, S_3, S_4 are correlated and we can infer the value of sensors S_2, S_3, S_4 from sensor S_1; therefore, only sensor S_1 is required to perform sampling.

Each correlation rule will likely affect more than one sensor. For example, although rule $R_{1,2}$ only need sensor S_1 to be in operation, the HiCoRE will still be able to provide the estimated values for sensors S_2, S_3, S_4 to upper system components upon request, as long as the correlation holds. In this case, the HiCoRE may choose to select the best, send all, or fuse sensor data for the information provisioning task. The *fusion logic* (shown in Figure 2) is a component for capturing these fusion decision of individual applications.

At the time when the correlation rules are discovered, the extended HiCoRE assigns a score (specific to a desired application objective) to each rule. The score is a measure of the 'goodness' of the rule in achieving the application objective and is calculated based on the characteristics of the sensors that are described in the correlation rule. These scoring schemes are described by application designers at design time as the *ranking metrics* (shown in Figure 2). A scoring schema is a function of various sensor's properties. For example, for energy conservation the function can be a combination of the battery level and power consumption.

In our example, the rule $R_{1,2}$ of the objective O_1 scores 84%, as only sensor S_1 will actually consume energy. By assigning a score to each rule, the solution provides a way to rank the alternatives of sensor selection for each application objective. In the same example, when an application changes the operational objective to improve accuracy of information, it may choose rule $R_{2,1}$, as the combination of sensors S_1, S_2 is able to achieve higher confidence of sensor data based on their specifications.

When a rule is selected (i.e., $R_{1,2}$ or $R_{2,1}$, depending on the application objective), the operations of sensors described in the rule will be adapted accordingly.

C. Information quality evaluator

In addition to supporting context-awareness and resource efficiecy of the sensing infrastructure, the ACoMS+ provides assurance of information quality by allowing application designers to optionally define the *QoI policies*, as shown in Figure 2. These QoI policies describe not only the evaluation metrics (e.g., certainty, timeliness), but also the choice of technique for the evaluation (e.g., Bayesian network, hidden Markov model, Kalman filter [6]). Profile of the quality evaluation can be described as a SensorML process model[2], which is used by the ACoMS to model pre-processing of raw sensor data. Through the QoI policies, the ACoMS+ allows the application designers to specifically define quality assurance strategies to evaluate fused context information that is critical to their applications.

Based on these information policies, the information quality evaluator checks the information quality of sensors' real-time observations, as shown in Figure 5(b). The result of the quality evaluation is associated with an objective-specific correlation rule.

Following the same example that we discussed in the last section, we have $QoI^{Bayesian}_{certainty}(R_{1,2})$ and $QoI^{Kalman}_{certainty}(R_{2,1})$. For objective O_1, the $QoI^{Bayesian}_{certainty}(R_{1,2})$ states the correlation rule $R_{1,2}$ has certainty level 74%, and Bayesian network is used for the evaluation. In this example, we assume the application decides to fuse data from the four sensors S_1, S_2, S_3 and S_4. It should be noted that only sensor S_1 value is gathered from the actual sensor node, while the other values are estimated according to the correlation rule, $R_{1,2}$. Should the certainty level be below the required certainty level specified by the applications, the ACoMS+ evaluates the alternative correlation rules recorded in the HiCoRE and performs the adaptation accordingly. The same evaluation processes apply when applications change their operational objectives (e.g., from preserving energy O_1 to improving accuracy of information O_2).

The information quality evaluator provides a way for the ACoMS+ to be application specific QoI-aware while keeping resource allocation transparent to the high-level context-aware applications.

V. Related Work

The efficient management of the sensing infrastructure is an important element lacking in most context-aware middleware solutions. Although there exist solutions that investigate this problem and provide some controls of the underlying sensing infrastructure, they fall short in various aspects. The ACoMS [11] is a middleware solution that has been extended based on the PACE framework to address sensor heterogeneity and fault-tolerant provisioning of context information. However, its approach to efficiently

[2]http://vast.uah.edu/SensorML

manage its sensing infrastructure is relatively primitive in comparison to the solution proposed in this paper. More specifically, in the ACoMS sensors are assumed to be single-adminstrative entities (i.e., a sensor network is treated as a single sensor). The RUNES middleware [5], as many others (e.g., the Gaia middleware [3], the Solar architecture [2], the PICO framework [12]), has similar "direct-to-sensor" assumption as the ACoMS middleware, despite the fact that a middleware of sensor network is used for managing the underlying sensors' operations. The main concern here is that efficient management of the sensing infrastructure depends on the individual underlying sensor network middleware solution, and application context information and operational objectives are not fully utilised for more adaptive and efficient management of the sensing infrastructure.

In the design of a context-aware and resource efficient sensing infrastructure presented in this paper, we introduced two dimensions of quality metrics of the discovered correlation rules and a way to evaluate them. The design uses sensor's specifications to estimate the 'goodness' of the rule in achieving the application objective and uses sensor real-time observations to verify the resulting information quality. The idea is based on, so called, *local* and *global* information quality evaluation in the wireless sensor network community. For example, Zahedi et. al. [15] explores the idea of combining sensors' characteristics and properties of fusion algorithm to assure information integrity and quality of sensor networks.

VI. Conclusion and Future Work

In this paper, we presented the design of a context-aware and resource efficient sensing infrastructure for context-aware applications. The contribution is the enhancement of the sensing infrastructure of a model based middleware for context-aware applications. The proposed enhancement extends the HiCoRE algorithm and incorporates it into the middleware for context-aware applications to achieve resource efficient context information provisioning from large groups of sensors or large scale sensor networks. The HiCoRE mines sensor data to discover correlations that can be used to save energy and/or bandwidth (by putting sensors into low-power mode or by inferring sensors' data if correlations exist rather than requiring the sensors to perform sampling). We also described the information quality evaluator that can provide assurance of quality of information. The presented design has been already partially implemented and tested; the HiCoRE algorithm has been integrated with the middleware for context-aware applications. The full development, e.g. evaluation of context information quality, is still in progress.

Acknowledgement

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program; and the Queensland Government.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *IEEE Computer Networks*, 38(4):393–422, 2002.

[2] G. Chen, M. Li, and D. Kotz. Data-centric middleware for context-aware pervasive computing. *Journal of Pervasive and Mobile Computing*, 4(2):216–253, April 2008.

[3] S. Chetan, A. Ranganathan, and R. Campbell. Towards fault tolerance pervasive computing. *IEEE Technology and Society Magazine*, 24(1):38–44, Spring 2005.

[4] S. Chong, M. Gaber, S. Krishnaswamy, and S. Loke. Arts: Using adaptive rule triggers on sensors for energy conservation using coarse-granularity data. In *ICESS'08*, Chengdu, China, 2008.

[5] P. Costa, G. Coulson, C. Mascolo, G. P. Picco, and S. Zachariadis. The runes middleware: A reconfigurable component-based approach to networked embedded systems. In *PIMRC'05*, pages 11–14, Berlin, Germany, 2005.

[6] D. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[7] T. A. Halpin. *Conceptual Schema and Relational Database Design*. Prentice Hall, Australia, Sydney, 2nd edition, 1995.

[8] K. Henricksen and J. Indulska. A software engineering framework for context-aware pervasive computing. In *PerCom'04*, pages 77–86, 2004.

[9] K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. *Journal of Pervasive and Mobile Computing*, 2(1):37–64, 2006.

[10] K. Henricksen and R. Robinson. A survey of middleware for sensor networks: State-of-art and future directions. In *3rd International Workshop on Middleware for Sensor Networks*, pages 60–65, Melbourne, Australia, 2006.

[11] P. Hu, J. Indulska, and R. Robinson. An autonomic context management system for pervasive computing. In *PerCom'08*, pages 213–224, Hong Kong, 2008.

[12] S. Kalasapur, M. Kumar, and B. Shirazi. Dynamic service composition in pervasive computing. *IEEE Transactions on Parallel and Distributed Systems*, 18(7):907–918, 2007.

[13] E. Lefebvre, M. Hadzagic, and E. Bossé. *Advances and Challenges in Multisensor Data and Information Processing*, chapter On Quality of Information in Multi-Source Fusion Environments, pages 69–77. IOS Press, 2007.

[14] B. Mellor. A world of connections. *The Economist*, 2007.

[15] S. Zahedi, E. Ngai, E. Gelenbe, D. Mylaraswamy, and M. Srivastava. Information quality aware sensor network services. In *The 42nd Asilomar Conference on Signals, Systems and Computers*, pages 1155–1159, Pacific Grove, CA, USA, October 2008.