The Role of Probabilistic Schemes in Multisensor Context-Awareness

Waltenegus Dargie

Abstract—**This paper investigates the role of existing "probabilistic" schemes to reason about various everyday situations on the basis of data from multiple heterogeneous physical sensors. The schemes we discuss are fuzzy logic, hidden Markov models, Bayesian networks, and Dempster-Schafer theory of evidence. The paper also presents a conceptual architecture and identifies the suitable scheme to be employed by each component of the architecture. As a proof-of-concept, we will introduce the architecture we implemented to model various places on the basis of data from temperature, light intensity and relative humidity sensors.**

Index Terms—**Context, Context-Aware Computing, Context Reasoning, Context Modelling**

I. Introduction

Three essential conditions should be satisfied for human beings to adapt to their surrounding and to carry out their everyday life with ease: perception of what is taking place, relating the perceived phenomena with previous experience or present expectation, and duly reacting to the perceived change in the environment. During the perception process, the human brain presents the real-world, not as a collection of raw data, but wholly, conceptually, and meaningfully [1]. Moreover, the above conditions are not a one-time fulfilment, but a continual process which includes the learning of new phenomena and the revision of existing beliefs regarding the real-world. Depending on the perception faculties and their experience (world model), human beings can establish a shared understanding of their surrounding at various levels of abstraction.

Likewise, the necessary precondition for developing self-managing systems (devices, applications, and networks) should entail the capability to perceive what is taking place in and outside of a computing environment. Only then are self-managing systems able to adapt meaningfully and exploit resources which are available in their surrounding. In other words, these systems should become context-aware. However, there is a gap between the type of awareness that is required for self-management and the sort of real-world aspects which can be captured by employing sensors. This gap is characterised by incompleteness, inexactness, and ignorance. Incompleteness arises due to the fact that there are always certain aspects of the real world which cannot be captured by employing physical sensors. Inexactness is less a concern, since it can be dealt with by considering the reports of multiple sensors which monitor one and the same situation. Ignorance can have multiple aspects: (1) we may not be able to judge how much we should trust a report of a given sensor because there is no technical specification to tell us about the accuracy, resolution, sensing range, etc., of the reporting sensor; (2) since physical sensors can be influenced by external factors (such as ambient noise or even the enclosing of the sensor), we may not be able to easily resolve conflicting reports from similar sensors; and (3) more importantly, in mobile environment, we may not be able to foresee what sensors might be available at any given time – in other words, we may not be able to determine at design time what aspects of a situation of interest we might be able to capture.

Subsequently, a context capturing task should deal with the problem of ignorance at various stages. In this paper, we shall investigate the role of some probabilistic reasoning schemes in tackling ignorance and in capturing a context as an abstraction of a dynamic real-world situation. We shall also propose a conceptual framework for context reasoning, and share the experience learned during the implementation of the architecture.

The remaining part of this paper is organised as follows: in section II, we discuss related work; in section III, we investigate the role of some probabilistic schemes in context reasoning; in section IV, we introduce the components of our conceptual architecture and illustrate its implementation; and finally, in section V, we give a brief conclusion.

II. Related Work

In the recent past, several architectures and frameworks have been proposed for context-aware computing. Most of these proposals can be categorised into three main groups. The first group focuses on separating the acquisition of a context from its consumption. It takes advantage of lessons learned in software engineering, decomposing the design of context-aware applications into various concerns. The second group takes a conceptual approach towards context acquisition, without prescribing to a particular technology or scheme. The third group delves into the process of context acquisition and

Manuscript received on October 13, 2006 and revised on December 22, 2006.

Waltenegus Dargie is with the Chair for Computer Networks, at the Faculty of Computer Science, Department of Computer Networks, Technical University of Dresden, D-01062, Dresden, Germany. (Phone: +49 351 463 38352; fax: +49-351-463-38251; e-mail: waltenegus.dargie@tu-dresden.de.

takes advantage of experiences learned in artificial intelligence, image processing, and speech recognition to compute a context. The works of Schilit [2] and Dey [3] belong to the first group. Gellersen et al. [4] and Coutaz et al. [5] are some examples of the second group. The work of Chen et al. [6], Gu et al. [7], wang et al. [8], Peltonen et al. [9] Korpipää et al. [10], Mäntyjärvi et al. [11], and Wu [12] are examples of the third group.

There is a significant similarity of approach in the first group, and depending on how the real world is viewed, they can further be divided into two subgroups: In the first subgroup, the components of a given architecture or framework map to a real-world object or situation – for example, an aggregator representing a person or a place [3]. In the second subgroup, context is retrieved from a blackboard by using a declarative query request. Context sources simultaneously post context values on the blackboard and applications, without the need to deal with the task of binding to individual sources, declaratively request and utilise a context. In general, the approaches above consider a context as a piecemeal construct, and do not address the question how it actually is acquired. A mechanism for its detection is implied to be available, and its scope and usefulness is implied to be known by all applications which consume it.

Gellersen et al. and Coutaz et al. offer conceptual architectures to help context-aware application developers deal with the actual task of context acquisition. The architecture of Coutaz et al. consists of a sensing layer, a perceptual layer, a situation and context identification layer, and an exploitation layer. The sensing layer generates numeric observables; the perception layer is responsible for providing symbolic observables at the appropriate level of abstraction; the situation and context identification layer identifies the current situation and context from observables.

Similarly, Gellersen et al. propose a layered conceptual framework for sensor based computation of context. The lowest layer is the sensor layer, which is defined by an open-ended collection of sensors capturing some aspects of the real world. The middle layer is occupied by the cue layer, which introduces cues as abstraction from raw sensory data. This layer is responsible for extracting generic features from sensed data, hiding the sensor interfaces from the upper layer, which is the context layer responsible for manipulating the cues obtained from the cue layer and computing context as an abstraction of a real world situation.

Neither of the conceptual architecture prescribes to any particular algorithm or scheme to illustrate the implementation of the architectural layers.

The third group can be divided into two subgroups: those which employ logic- (as well as rule-) based schemes to reason about a context of interest and those which employ probabilistic schemes. In this paper, the term 'probabilistic schemes' is used broadly, and encompasses all schemes which deal with uncertain data as opposed to factual data. The works of Wang et al., Chen et al., and Gu et al. belong to the first subgroup whereas the remaining works belong to the second subgroup. Since this paper focuses on probabilistic schemes and because of the lack of space, we do not discuss logic-based reasoning schemes any further.

An essential aspect of a probabilistic scheme as a reasoning tool for computing a context is its capability to deal with the problem of ignorance discussed in section I. Peltonen et al. classify auditory scenes into predefined classes by employing two classification mechanisms: 1-NN classifier and Mel-frequency cepstral coefficients with Gaussian mixture models. Subsequently, they could be able to recognise a physical environment by using audio information only. The audio scene comprises several everyday outside and inside environments, such as streets, restaurants, offices, homes, cars, etc. The limitation of this approach is the absence of a reusable framework or architecture.

Korpipää et al. propose a multilayer context-processing framework to carry out a similar recognition task. The bottom layer is occupied by an array of physical sensors which measure physical properties. The other layers in the context processing hierarchy include a feature extraction layer incorporating a variety of audio signal processing algorithms from the MPEG-7 standard; a quantisation layer based on fuzzy sets and crisp limits; and a classification layer employing a naïve Bayesian classifier which reasons about complex contexts.

Wu applies Dempster-Schafer's theory of evidence to deal with uncertainty associated with context sensing. In his implementation, an Aggregator receives video and audio features from a camera and a set of microphone widgets to determine the likelihood of a participant's focus of attention in a meeting.

Mäntyjärvi et al. proposes a four-layered framework for higher-level context recognition. At the lowest level there are context information sources. These sources deliver sampled raw measurements which map to physical properties. The middle layers are occupied by the context measurement and context atoms extraction unites, respectively. After sampling, raw signals are pre-processed. In the case of sensor measurements, signal values are calibrated and rescaled. Pre-processed signals are used as input to various feature extraction methods in time and frequency domains producing features to describe context information. For example, the root mean square (RMS) value of an audio signal describes the loudness of a surrounding. The first task in context extraction is to abstract raw sensor signals and to compress information by using different signal processing and feature extraction methods. The features to be extracted are chosen depending on how well they describe some aspects of a context of interest. Extracted features are called context atoms since they contain the smallest amount of context information. The upper layer is occupied by the context information fusion unit, which process the context atoms and computes a higher-level context.

This work was in part motivated by the incompleteness of the probabilistic approaches discussed so far. Even though attempts were made to identify the various stages of a context computing process and the necessary schemes for it, we discovered that certain steps were missing to make the process wholesome. For example, in most of the frameworks, pre-processing and feature extraction layers were identified as essential steps, but no particular schemes were proposed to carry out these tasks.

One may intuitively argue that it is possible to employ some of the richly available algorithms from image processing or speech recognition fields. This might hold true for some context recognition tasks, however, some context processing tasks pose peculiar challenges of their own which must be uniquely addressed. For example, both image processing and speech recognition tasks deal with a single sensing element; or even if multiple sensing elements are employed, the features extracted from each sensing elements are similar; subsequently, a similar set of algorithms can be used for each sensing element. This may not be the case when we deal with heterogeneous sensing elements, for example, when we attempt to capture the activity of a person by processing data from temperature, light intensity, humidity, and accelerometer sensors.

Our aim is not to propose an entirely new architecture and entirely new context processing algorithms; rather, it is to propose a comprehensive framework in which functional components are identified, the task of each component is properly explained, and the potential processing schemes are proposed. In this pursuit, we attempt to bring together the various approaches and identify their suitable place for generalizing a context processing task. In the next section, we will give a summary of the most frequently employed schemes. The treatment we give, however, is neither exhaustive nor comprehensive.

III. Probabilistic Schemes for Context Processing

In this section, we discuss fuzzy logic, hidden Markov models, Bayesian Networks, and the Dempster-Shaffer theory of evidence. Our selection of these schemes depends on the potential of each scheme to tackle a particular challenge during a context processing task. Therefore, our approach must not be understood as a comparative investigation.

A. Fuzzy Logic

L.A. Zadeh introduced the fuzzy set theory claiming that many sets in the real-world are defined by a non-distinct boundary [13]. His theory extends two-valued logic by allowing intermediate values such that a gradual transition from falsehood to truth, and vice versa, is possible. Consequently, notions like cold, warm, visible etc. can be mathematically formulated and presented to computers to achieve more human-like decisions.

The fuzzy member function $\mu(x)$ – a precise but subjective measure which depends on the context of use – attaches a numerical value to each element x of a fuzzy set y in order to describe the degree of membership of x in y. The range of $\mu(x)$ lies between [0, 1]; where 0 denotes that x is certainly not an element of y; and 1 denotes that x has full membership, that is, it is certainly included in y. The intermediate values reflect the relative level of membership associated with the item. One of the advantages of a fuzzy set is its allowance to define linguistic variables which makes sense to human social and conceptual settings. For example, the thermal characteristic of a room can be described as: *very cold, cold, medium, hot, very hot*. The set of values a linguistic variable can take is called the variable's *term set*.

The elements of two or more fuzzy sets can be combined (fused) to create a new fuzzy set with its own membership function. In the same way crisp sets are manipulated using intersection, union, complement, and other set operations, fuzzy sets can be manipulated using conjunction, disjunction, complement, and containment operations. Another operation particular to fuzzy logic is the *modifier* operation. A modifier modifies the meaning of a linguistic variable as well as its term set. For example, in the term "*very cold*", the modifier *very* modifies the fuzzy term '*cold'*. By chaining primitive operations and modifiers, more complex fuzzy sets can be generated. Even though the precise interpretation of a modifier is application-specific, intuitively, it has either an intensifying or a lessening effect on the term it is applied to.

B. Hidden Markov Models (HMM)

A Markov chain or process is a sequence of events (called states) the probability of each of which is wholly dependent on the event immediately preceding it. Given a sequence of states: $Q = \{q_1, q_2, ..., q_n\}$, the state of q_n is determined as:

$$p(q_n \mid q_{n-1}, ..., q_2, q_1) \approx p(q_n \mid q_{n-1}) \quad (1)$$

A Hidden Markov Model (HMM) represents stochastic sequences as Markov chains; the states are not directly observed, but are associated with observable evidences, called emissions, and their occurrence probabilities depend on the hidden states. The generation of a random sequence is the result of a random transition in the chain.

HMM has been applied to a wide variety of dynamic systems, the most salient applications being the ones dealing with speech recognition. For these applications, the hidden states are the smallest units of a speech called *phonemes*. Every word is thus built from phonemes, which are identified with the hidden states. After different hidden Markov models are trained on examples, one can run each HMM separately on a new word to be recognized. Then the likelihood of every HMM is computed on the new word and the highest likelihood is chosen.

In order to model a process with an HMM, the following elements should be available: (1) The number of states in the model, *N;* (2) the number of observation symbols, *M,* as well as a probability distribution matrix, *B*, in each of the states describing the occurrence of observable evidence; and (3) the state transition probabilities described by a square matrix, *A*.

Operations with Hidden Markov Models are often carried out with three aims in mind: (1) given the model $\lambda = (A, B, \pi)$, and a sequence of observations *O*, one might want to compute the likelihood of the observed sequence *O*, i.e., $p(O|\lambda)$; (2)

given $\lambda = (A, B, \pi)$ and an observation sequence O, one might want to determine an optimal state sequence for the underlying Markov process – in other words, the aim is to uncover the most likely hidden states of the HMM; or (3) given an observation sequence O and the dimensions N and M, one might want to determine $\lambda = (A, B, \pi)$ that maximizes the probability of O – this is useful for training the model to best fit the observed data.

One particular aspect of a HMM is the presence of transition from one hidden state to another. This aspect could be very useful to predict human behaviour or a sequence of activities. However, to determine a single complex activity by considering various correlated observable aspects (primitive contexts which can be captured by employing sensors), an HMM may not be optimal.

C. Bayesian Networks

Bayesian networks apply Baye's theorem and satisfy the Markov's condition – a node A is independent of node B given its parents – to model probabilistic relationships among distinctions of interest in an uncertain-reasoning [14]. The networks are directed acyclic graphs (DAG) where nodes represent random variables and a directed arrow represents conditional dependence between the random variables. A particular configuration of a Bayesian network refers to an instantiation of the random variables with values from a two dimensional value vector; the possibility of this configuration is determined by its joint probability.

Operations in BN can be decomposed into inference and learning. Inference refers to the computation of a posterior probability distribution over a model (or parameters). The precondition for inference is that the structure of the network is known and the prior probability distribution is already available. Learning can refer to the structure of the model, or the parameters, or both. Furthermore, learning may take place in the presence of either fully or partially observed variables. In any case, the goal of learning is to find a single model (or set of parameters) which best explains the observed evidence.

Because a Bayesian Network does not necessarily require a transition from one state to another in order to compute the global or local state of the network, it can be an excellent scheme to compute a single higher level context as an abstraction of numerous primitive contexts.

D. Dempster-Shaffer Theory of Evidence

Traditional probabilistic schemes dealing with uncertainty are established upon two basic assumptions: (1) that the analyst has knowledge of the probabilities of all events; where there is no sufficient knowledge with regards to some of the events, the *Principle of Insufficient Reason* is applied, and, (2) that the axiom of additivity is satisfied. The two basic assumptions work fine for *aleatory* uncertainty. This type of uncertainty, also called irreducible uncertainty, class A uncertainty, or stochastic uncertainty, results from the fact that a system can behave randomly. On the other hand, if we want to encode uncertainties resulting from an inconsistent report of two or more independent sources which observe one and the same phenomenon, the assumptions above may not be reasonable assumptions[1]. The Dempster-Schafer theory of evidence (DST) offers an alternative to traditional probabilistic theory by providing schemes to encode epistemic uncertainty into the model of a system [15].

A DST is defined as an undirected graph with associated belief functions. For every variable or node A in the graph, the frame of decrement $\Omega(A)$ is the set of exhaustive and mutually exclusive propositions or hypothesis about A. If X is a node which consists of several individual nodes, $\Omega(A)$ is the Cartesian product of the frames of all conjoining members. Making a decision in DST means to choose the best proposition from Ω. Only one proposition can be true, otherwise the frame has to be redefined.

Let $2^{\Omega(A)}$ denote the power set of $\Omega(A)$. A *basic probability assignment*, *m,* is a primitive of DST; it defines the mapping of the power set to the interval [0, 1]. The value of m for the set A, expresses the proportion of all relevant and available evidence supports on the claim that a particular element of X (the universal set) belongs to the set A, but to no particular subset of A. All the remaining subsets of A are represented by separate mass functions. The total belief in the subset $B \subseteq \Omega$, where $B \in 2^{\Omega(A)}$, is measured by the belief function, $Bel(A)$, which is defined as:

$$Bel(A) = \sum_{B \backslash \mathrm{B} \subseteq \mathrm{A}} m(B) \tag{2}$$

The Belief function sums up all the basic probability functions of the proper subsets B of A. It follows from equation (2) that for any singleton $B \in \Omega(A)$, $Bel(B)$ is equal to $m(B)$, while for the entire frame $\Omega(A)$, $Bel(\Omega(A)) = 1$. The Plausibility function, $Pl(A)$, is calculated by summing all the basic probability functions of the set B that intersects the set of interest A.

$$Pl(A) = \sum_{B \backslash \mathrm{B} \cap \mathrm{A} \neq \phi} m(B) \tag{3}$$

The belief value for the hypothesis A may be interpreted as the minimum uncertainty value about A, and its plausibility value, which is also the "unbelief" value of the complementary hypothesis $\overline{A} : Pl(A) = 1 - Bel(\overline{A})$, may be interpreted as the maximum uncertainty value of A. Thus, uncertainty about A is represented by the values of the interval $[Bel(A),\ Pls(A)]$, which is called the belief interval. The length of the belief interval provides a measurement of the imprecision about the uncertainty value.

The DST is also known as combination theory, since it is often employed to combine the evidence gathered from two or more independent sources in order to minimise the effect of imprecision. The Dempster rule of combination is a generalisation of Baye's rule, and is given by:

[1] This type of uncertainty is called epistemic uncertainty, subjective uncertainty, class B uncertainty, or reducible uncertainty.

$$\begin{cases} m(A) = \dfrac{\sum_{B_1 \cap \ldots \cap B_n = A \neq \phi} \prod_{i=1}^{N} m_i(B_i)}{1-k} \\ K = \sum_{B_1 \cap \ldots \cap B_n = \phi} m_i(B_i) \end{cases} \quad (4)$$

In (4), $K \rightarrow (\in [0,1]$, represents a basic probability assignment function associated with conflict; it is determined by summing the products of the basic probability assignments of all sets where the intersection is null. *1-K* is interpreted as a measure of conflict between the different sources. The larger *K* is, the more the sources are conflicting, and the lesser the reliability of the combined data. When *K* equals 1, the orthogonal sum does not exist, the sources are totally contradictory, and it is no longer possible to combine them.

IV. The Conceptual Architecture and its Implementation

A context as a higher-level abstraction of a dynamic real-world situation can be computed in three steps. The first step deals with the capturing of atomic aspects (or primitive context types) by directly employing sensors. An atomic aspect maps directly to a measurable physical aspect. It can be acoustic aspects, thermal aspects, humidity, light intensity, etc. To make a primitive context meaningful to a human user, its states can be transformed such that they reflect human reasoning. For example, the measurement of a temperature sensor can be mapped to one of the following conceptual states: cold, lukewarm, warm, or hot. The conceptual states of a primitive context can be computed using fuzzy logic, as distinction between the states is gradual as opposed to well defined margins. The second step towards context computing deals with the aggregation of several primitive contexts which describe one and the same entity. This entity can be a place, a device, a person, etc.; it can also be another primitive context. In case of a primitive context, aggregation is done to disambiguate potential conflicting observations of independent sources, where these sources can be either sensors or those entities which process fuzzy algorithms. Since a number of heterogeneous sensors may observe one and the same phenomenon from different angles, they may deliver conflicting reports. A simple example is the thermal observations of a room from multiple temperature sensors which are distributed inside a room; depending on the accuracy and resolution of the sensors as well as their spatial position, each sensor may report a different measurement. Hence, an aggregation process is needed to appropriately model the room. As the sensors are independent of each other, the Dempster-Shaffer theory of evidence is the best tool for combining the different evidences. A DST processor takes as its input the fuzzy sets along with corresponding membership functions of the lower level, and combines them, using the combination rule given by equation (4). The sensors' accuracy or resolution or both, or some prior knowledge of the sensors distribution with regard to the scene being observed can be used to compute the probability mass function which is vital for combining the reports. The third and final stage in context computing is the use of a higher-level reasoning scheme to compute a context of interest, i.e., the higher-level context. Eligible schemes are hidden Markov models and Bayesian Networks. If multiple higher-level contexts are to be computed, and if there is a time correlation between the higher-level contexts, a hidden Markov model is suitable. If, on the other hand, we are interested to compute a single higher-level context as an abstraction of several primitive contexts, a Bayesian network is more plausible. Figure 1 shows our conceptual architecture.

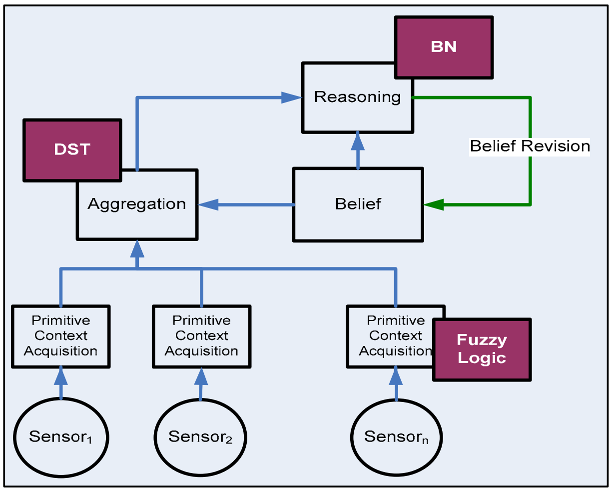

Figure 1: A Conceptual architecture for computing a context.

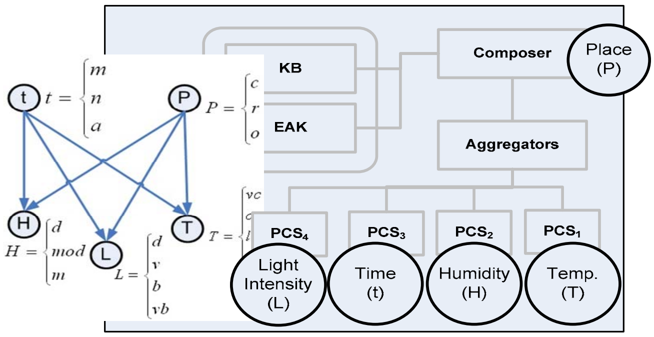

Figure 2: An implementation of the Conceptual architecture.

Since probabilistic schemes deal with beliefs for aggregation, for setting up the reasoning scheme, and for deciding dependency between the various aspects of a context, our conceptual architecture defines the belief component. The blue arrows that extend from the belief component to the aggregator and the reasoning component explain the belief of the system regarding the tasks carried out by the components. The green arrow that extends from the reasoning component to the belief component revises the belief of the system. The model of a computing environment should be updated to reflect the perceived change in the environment. In other worlds, the facts and beliefs a context computing system employs to manipulate primitive contexts should reflect the reality. This is possible only if the system accommodates a belief revision mechanism. Another reason for belief revision (model updating) is the detection of contradictory information within the model [16]. In situations involving imprecise knowledge of entities and relationships between them, it is possible to arrive at a conclusion which might turn out to be incorrect as soon as more reliable evidence becomes available.

Figure 2 shows our implementation of the conceptual architecture. We developed a system to reason about the whereabouts of a mobile person. We identified four main aspects (primitive contexts) to model various places: light intensity, time, relative humidity, and temperature. Each primitive context has conceptual states. For light intensity, these are: dark (d), visible (v), bright (b), and very bright (vb); for time these are: morning (m), noon (n), and afternoon (a); for relative humidity, these are: dry (d), modest (mod), and moist (m); and for temperature, these are: cold (c), lukewarm (lw), warm (w), and hot (h). These conceptual states were defined as members of fuzzy sets. We employed several Dallas Semiconductor iButton® sensors with different accuracy, sensing range, and resolution to deliver the primitive contexts. The aggregators we employed have two tasks: (1) to combine data from similar sensors and improve sensor reading; and (2) to represent a particular place and to gather all relevant primitive contexts each of which describes a particular aspect of the place. Finally, we employed Bayesian Networks to actually determine the whereabouts of the person by computing posterior probability distributions for all potential places which can be represented by the primitive contexts. The belief component of figure 2 is divided into a knowledge base (KB) subcomponent and an empirical ambient knowledge (EAK) subcomponent. The KB is responsible to manage factual relationships – such as containment relationships – between various places, while the EAK manages conditional (dependency) relationships. A PCS is an abbreviation for a primitive context server; it delivers a primitive context to a higher level component such as an aggregator. The actual reasoning task in the implementation is carried out by the Composer, which implements self-organising Bayesian Networks.

As can be seen in figure 2, four primitive context servers are available to periodically query the sensors they abstract, and report their reading to a set of aggregators. The aggregators, depending on their specific tasks (i.e., fusion or simple collecting data from various heterogeneous primitive contexts), subscribe to the primitive context servers and push their output to the composer. When the composer receives the report of the aggregators, it determines the configuration of a Bayesian network by querying the KB and the EAK; afterwards, it computes posterior probability distribution to determine to what most likely place the sensor measurement refers to. We could be able to determine whether a person was on a corridor, inside a room, inside a building (i.e., discrimination between a room and a corridor was not possible), or an outdoor place.

V. Conclusion

In this paper, we investigated the role of various probabilistic schemes to reason about various everyday human situations on the basis of data from multiple sensors. Instead of comparing and contrasting the performance of each scheme with respect to other schemes, we attempted to identify its suitability in computing a context of interest. Moreover, we proposed a conceptual architecture and identify a probabilistic scheme for each of its components. We identified fuzzy logic to be useful for defining the conceptual states of a primitive context to enable human-like reasoning; DST for combining the independent observations of multiple sensors each of which observes one and the same phenomenon; and HMM and BN for actually computing a higher-level context. To demonstrate our approach, we reasoned about the whereabouts of a mobile person on the bases of data from temperature, light intensity, and humidity sensors as well as on the basis of time context. We could be able to discriminate whether the person was inside a room, on a corridor, inside a building or outdoors.

References

[1] Yancey P. and Brand P. *In His Image*. Sondervan Publisher, 1987.

[2] N.B. Schilit, N.I. Adams, and R. Want, "Context-Aware Computing Applications," In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, December 1994. Pages 85-90. IEEE Computer Society.

[3] A. Dey, "Providing architectural support for building context-aware applications", Doctoral Thesis, 2000.

[4] H.-W. Gellersen, A. Schmidt, and M. Beigl, "Multi-sensor context-awareness in mobile devices and smart artifacts," *Mob. Netw. Appl.* 7, 5 (Oct. 2002), 341-351.

[5] J. Coutaz,, J.L. Crowley, S. Dobson, and D. Garlan, "Context is key," *Commun. ACM* 48, 3 (Mar. 2005), 49-53

[6] H. Chen, F. Perich, D. Chakraborty, T. Finin, and A. Joshi, "Intelligent Agents Meet Semantic Web in a Smart Meeting Room," *In the Proceedings of the International Conference on Autonomous Agents*. IEEE Computer Society, Washington, DC, 854-861.

[7] T. Gu, H.K. Pung, and D.Q. Zhang, "Toward an OSGi-Based Infrastructure for Context-Aware Applications," *IEEE Pervasive Computing* 3, 4 (Oct. 2004), 66-74.

[8] X. Wang, J. S. Dong, C. Chin, S. Hettiarachchi, and D. Zhang, "Semantic Space: An Infrastructure for Smart Spaces," *IEEE Pervasive Computing* 3, 3 (Jul. 2004), 32-39.

[9] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational auditory scene recognition," in Proc. of International conference on acoustic speech and signal processing, 2002.

[10] P. Korpipää, J. Mäntyjärvi, J. Kela, H. Kernen, and E-J. Malm, "Managing Context Information in Mobile Devices," Journal of Pervasive Computing, IEEE, 2003.

[11] J. Mäntyjärvi, J. Himberg, and P. Huuskonen, "Collaborative Context Recognition for Handheld Devices," in *Proceedings of the First IEEE international Conference on Pervasive Computing and Communications* (March 23 - 26, 2003). PERCOM. IEEE Computer Society, Washington, DC, 161.

[12] H. Wu, "Sensor Data Fusion for Context-Aware Computing Using Dempster-Schafer Theory," Doctoral Thesis, 2004.

[13] L.A. Zadeh, "*Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A. Zadeh*" World Scientific Publishing Co., Inc.

[14] D. Heckerman, "Bayesian Networks for Data Mining," *Data Min. Knowledge Discovery* 1, 1 (Jan. 1997), 79-119.

[15] G.A. Shafer, "*Mathematical Theory of Evidence*," Princeton University Press, 1979.

[16] J.P. Martines and S.C. Shapiro, "A model for belief revision," Journal of Artificial Intelligence. Elsevier Science Publishers, 1988.