

Integrating Facts and Beliefs to Model and Reason about Context

Waltenegus Dargie and Thomas Springer

TU Dresden, Institute for Systems Architecture, Computer Networks Group,
Helmholtzstrasse 10, 01062 Dresden, Germany
{waltenegus.dargie, thomas.springer}@tu-dresden.de

Abstract. This paper presents a twofold context modelling approach that integrates beliefs (uncertain knowledge) and facts to reason about various everyday situations. Awareness of everyday situations enables mobile devices to adapt to the social and conceptual settings in which they operate; it also enables resources which share a similar context to cooperate in order to carry out a distributed task on behalf of their user. Our context modelling process involves the identification of the context of interest, the determination of those aspects of a context which can be captured by employing sensors, the determination of contextual states for each aspect, and finally, the determination of logical and probabilistic relationships between the contextual aspects and the context they represent. We demonstrate our approach by modelling physical places. Data from various heterogeneous sensors build our system's belief, while containment relationships build its factual knowledge regarding places. The system utilises its belief and factual knowledge to reason about the whereabouts of a mobile user.

1 Introduction

Human beings are apt to adapt to their surrounding by perceiving what is taking place around them and by relating the perceived change in their surrounding with their expectations and experiences. As he laid out his vision for ubiquitous computing, Weiser asserted that the idea first arose from "contemplating the place of today's computer in actual activities of everyday life. In particular, anthropological studies of work life teach us that people primarily work in a world of shared situations and unexamined technological skills [14]."

For example, when people attend a meeting, their eyes communicate to convey agreements or disagreements to what is said or unsaid; voices are whispered to exchange impromptu opinions; facial expressions reveal to the other participants fatigue, boredom, or disinterest. More importantly, speeches may not be grammatically correct or complete. Previous as well as unfolding incidents enable the participants to capture what cannot be expressed verbally. Speakers shift from one language to another and use words with multiple meanings, and still the other participants can follow.

Flexibility and adaptation is possible because the social and conceptual setting (i.e., the context) encompassing the interaction is effortlessly recognised by all participants. As a result, within the perceived context, many activities unfold, some of which are unpremeditated, yet consistent with the context, while other activities express the freedom associated with the recognition of the context - for example, using incomplete or incorrect statements, or using words with multiple meanings. Still other activities reflect the participants' adjustment of behaviour in compliance with the context of the setting - for example, participants whispering to exchange impromptu ideas.

Capturing a context of interest and representing it in a meaningful way has been and still is the main focus of context-aware computing. The research community has approached this task from different perspectives. For example, Schilit et al. [12], Dey et al. [5], and Pasco [9] offer conceptual frameworks in which the different components required for context acquisition are proposed and explained. The various subtasks the components carry out represent a context at different levels of abstraction. Additional subtasks include discovery services and context storage. While their work identify the essential aspects of context computing, it offers little insight as to how the actual task of context recognition is carried out.

Gellersen et al. [6] offers a layered, conceptual architecture for context recognition which includes a sensor layer, a cue layer and a context layer. The sensor layer is responsible for obtaining raw data from physical sensors; the cue layer is responsible for obtaining meaningful features; the context layer is responsible for obtaining a context of interest by interpreting the relationship between several cues. Schmidt et al. [11] demonstrate the usefulness of the conceptual architecture by capturing various everyday contexts such as the state of a smart cup and the activity of a smart mobile phone.

While the conceptual approach proposed by Gellersen et al. is the basis of our work, we aim at complementing a context recognition task with a context modelling task. Subsequently, we propose a context modelling and reasoning guideline for presenting the real-world to computers wholly, conceptually, and meaningfully. Our approach permits the integration of facts and beliefs regarding entities (people, places, device, etc.) which are useful for computing a context as a representation of a dynamic real-world situation. We will demonstrate that it is possible to compute a context even though only a subpart of its aspects can be captured by employing sensors, and even if it may not be possible to foresee which of these aspects can be captured at a given time.

The rest of this paper is organized as follows: in section 2 we discuss challenges associated with a context computing task; in section 3 we discuss related work; in section 4 we introduce our approach, and illustrate its implementation. Finally, in section 5 we will close the paper with a discussion and concluding remarks.

2 Aspects of Context Computing

Capturing a context as a representation of a conceptual or a social setting is not a straightforward achievement. Firstly, because conceptual and social settings are difficult to directly capture by employing sensor alone, a context as an abstraction of these settings is not explicitly available. Data have to be gathered from heterogeneous sources in a seamless fashion, desirable features should be extracted from the data, and a reasoning operation has to be performed. A context data source may abstract any type of sensor, framework, database, user input, or application. As far as a context is concerned, these data can be classified as factual or approximated data [4].

While factual data remain unchanged regardless of repeated observations of a given phenomenon, approximated data are modelled as *beliefs*, since different or repeated observations of the same phenomenon under similar circumstances may result in disparity. An example for the former is the profile of a user or a device in a database. Another example is the status of a device or an application: on, off or idle; an example for the latter is the thermal or acoustic property of a room. Secondly, a concomitant effect associated with approximated data is uncertainty – most physical sensors have different technical specifications and may be affected by environmental conditions differently. Thirdly, in a pervasive computing environment, the availability of sensing devices is dynamic; sensors may come and go over time, i.e., one may not be able to foresee what sensors can be employed for a given sensing task. In the literature, this problem is addressed by separating the concern of context acquisition from the context consumption, since applications are interested in the context they employ rather than in its acquisition process [5]. This, however, does not warrant the constant availability of a mechanism for capturing a certain contextual aspect at all time. Therefore, a context computing process should involve learning about new contextual aspects which are not foreseen previously but could provide indirect evidence about a situation of interest.

Finally, data fusion and recognition operations for manipulating sensed data entail assumptions and incomplete knowledge (world models), producing additional uncertainty. To minimise the effect of this, a context computing process should entail dynamic belief revision and update of models.

3 Related Work

Early research in context-aware computing focused on implementing specific applications. Based on the experiences learned, more generic solutions were developed introducing different levels of abstraction for gathering, interpreting, deriving and aggregating a context (refer, for example, Schmidt et al. [11] and Salber et al. [10]). The approach described in Schmidt et al. outlines a four layered approach to capture and process a context in order to drive higher-level situational information. Hence, sensors data are first transformed into predefined cues, and cues are processed by logical rules to determine the current situation of an application.

More recent approaches focus on the creation of comprehensive and generic models of a context to facilitate interoperability and context reuse. Henricksen et al. [7] employ ORM (Object Role Modelling) to model a context as an association of fact types and roles. Fact types represent physical objects while roles represent dependencies between them. The model enables the representation of a context as a fact type the property of which can be static, sensed, profiled, derived, or alternative; an alternative property signifies the potential presence of several (possibly contradictory) reports about a particular attribute. An interesting aspect of the modelling concept is the recognition of a context as a dynamic construct.

The model of Crowley et al. [3] consists of three basic elements of a dynamic real world situation: entities, roles, and relations. An entity is an association of correlated observable variables corresponding to a physical object; a relation is a predicate function describing the properties of entities; and a role is a potential set of actions within a task; where a task is defined to be the association of a current state and a goal state. Coutaz et al. [2] propose a conceptual architecture which manipulates the model proposed by Crowley et al. in order to reason about the context of a mobile user. The architecture 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. The exploitation layer serves as an adapter between application semantics and the infrastructure, enabling applications to put declarative requests for context services. The architecture does not prescribe to any particular algorithm or schemes; thus, it is difficult to scrutinize the modelling concept as well as the architecture.

Chen et al. [1] propose a common context vocabulary based on a concept hierarchy. The proposed COBRA-ONT ontology contains general concepts which can be reused in the domain of pervasive computing. It models, among others, physical locations, devices, temporal concepts and privacy policies. The reasoning over this information is based on the OWL ontology requiring no additional rule framework.

Wang et al. [13] propose a hierarchical ontology consisting of upper and domain-specific ontology. The upper ontology models basic contextual entities while the domain-specific ontology contains domain or application related concepts which are modelled on the basis of the generic concepts of the upper ontology. They employ OWL DL for ontology representation and two additional reasoning schemes for consistency check: DL-reasoners and logic-based rules.

The approaches above consider a context as a construct which can be captured in its entirety. Subsequently, their usefulness is limited to model a dynamic real-world situation with factual data only.

Korpipää et al. [8] propose a context recognition framework, describing a context as an uncertain and dynamic construct. Among its most important tasks, the framework manages uncertainty of sensed data through the use of proba-

bility based inference and fuzzy membership. They employ ontology to model context which will be used by a naïve Bayesian classifier to reason about various real-world situations. The framework supports the computation of the model parameters from training data. Though the Bayesian classifier employed a variety of contextual aspects, each was extracted from a single audio input. An additional limitation of the framework is its employment of a naïve Bayesian classifier, which assumes the absence of causal dependencies between the input context atoms. In most practical cases, however, this assumption may not hold true. Similarly, Mäntyjärvi et al. [16] apply k-means clustering and minimum-variance segmentation algorithms to capture the activity of a mobile user. The sensors they employ include motion, temperature, skin conductance, and temperature sensors.

In general, the above approaches employ either factual or probabilistic inputs to model and reason about context. We build on the experiences learned previously, but combine both factual and probabilistic aspects to tackle the problem of uncertainty at various stages.

4 A Guideline for Modelling Everyday Situations

In section 2, we discussed the causes of uncertainty in a context computing process. In this section, we will introduce a context modelling process which integrates facts and beliefs with the goal of reducing uncertainty. While the model enables probabilistic reasoning schemes to deal with inaccurately captured dynamic aspects, the additional factual knowledge in the model makes possible containment tests to resolve between equally probable contextual states. To motivate our approach, we give a brief scenario.

4.1 Scenario

Active monitoring is one example of an application domain where context-awareness plays a role. In this scenario, the task is to detect the presence of interesting entities and to determine the relationship between these entities. If a given pattern of relationship seems to be likely to be broken or violated, the application should take a predictive measure. A typical example of active monitoring can be watching children and their behaviour towards each other in a kindergarten. The specific task of the application may be to determine which child is playing with whom, but it can also be to determine whether there are children (regardless of their identity) in certain places and whether the social atmosphere is healthy. If, for instance, the application picks up an aggressive tone while children are playing, it should sound an alarm to avoid further escalation of events. An additional example is suppose we want the application to monitor the whereabouts of a particular child. To recognise and avoid dangerous situations, the application should determine whether the child is inside a room, in a corridor, or outdoors. If the child happens to be in a corridor or outdoors by itself, i.e., if no supervisor is with it, the system should alert the responsible

supervisor; if the responsible supervisor does not respond, or if she is currently busy with another child, the system should alert the head supervisor.

In the second scenario, the relative whereabouts of a child triggers an action, but it may also trigger a set of actions, depending on the activity of its supervisor. The whereabouts of the child in itself cannot be taken as a context of interest. The same is true to the activity of the supervisor. This very well reflects the relative and relational nature of a context. Subsequently, the contexts of interest are: the relative whereabouts of a child in reference to its supervisor and the activity of the supervisor in reference to the whereabouts of the child.

The different aspects of a context of interest (the context model) depend on the available sensing mechanisms. For example, there is a plethora of location sensing mechanisms which are based on either infrared or RF or ultrasound technologies or a combination of some of these technologies. Use of any of these technologies makes the modeling task less of an issue compared to the actual sensing task. If, on the other hand, there is no direct mechanism to locate a person, the context of interest should be derived from other aspects which describe a place - these could be, for example, light intensity, temperature, humidity, sound pressure, ambient noise, etc., which provide indirect evidence about a place.

At an abstract level, the various contextual features which make up the model are the different forms of places (room, corridor and outdoors), the relationship of the supervisor not only with the child of interest, but also with other children, i.e., her activity. In the next subsections, we will present our context modelling approach step by step.

4.2 Determination of the context of interest

The main purpose of a context model is to describe a context of interest as well as its various states as exhaustively and as completely as possible. Since an interest in a context comes from applications which respond to it, the first step in the modelling process is to identify the set of contextual states which are relevant for a particular application. In the following subsection, we take the whereabouts of the child as a context of interest.

4.3 Identification of aspects of a context

Once a context of interest is known, the next step is to identify all of its aspects which can directly be captured by employing sensors. This decision is made based on the requirements of the application developer or the user. For the whereabouts of the child, the application developer may decide to employ already available sensors which capture humidity, temperature, light intensity, etc.

At this stage, it is not necessary to determine how exactly these aspects describe the context of interest. It is sufficient to know that some kind of relationship exists. In the next subsection, we shall demonstrate how a system can be trained to determine (at least in part) the nature of the relationship. The existence of a relationship between a context of interest and its aspects is modelled as an extensible factual relationship.

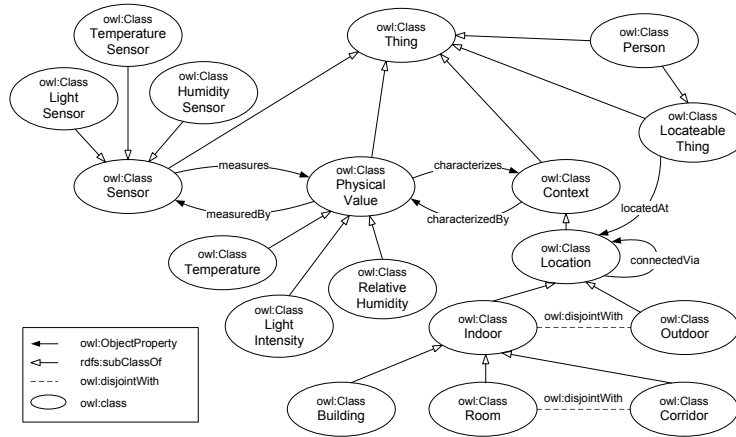


Fig. 1. A factual context model using the OWL terminology

Figure 1 shows the major concepts and properties defining facts about persons, places, sensors, and physical values which can be measured by sensors. We use the Web Ontology Language (OWL) terminology in the model, depicting relationships between the concepts by labeled arrows. For example, a sensor measuring a certain physical property is modelled by the owl:ObjectProperty "measures" with its domain of owl:class "Sensor" and its range owl:class "PhysicalValue".

Relevant places are modelled as a hierarchy of location concepts. We distinguish between indoor and outdoor and model a building as a composition of rooms and corridors. The concepts indoor and outdoor as well as room and corridor are modelled as disjoint concepts using the owl:disjointWith constructor, since they mutually exclude each other. Moreover, direct connections (such as a door, a passage, a stair case, an elevator, etc.) between places are described by the role "connectedVia" and by additional roles derived from the role "connectedVia"; for example, "connectedViaDoor". These factual relationships will be useful to validate a location context computed by a probabilistic reasoning scheme (to be discussed in section 4.5).

The relations between sensors and the corresponding context are modelled by the owl:ObjectProperty "characterizedBy" and "characterizes". The concept "Location" is described as a subclass of the "Context" concept. At the same time, location is characterized by the physical values temperature, light intensity, and relative humidity. To dynamically detect and bind to available sensors, an additional individual concept called "Sensor" is defined.

4.4 Determination of factual and probabilistic states

An aspect of a context can be modelled as either a discrete aspect or a continuous random variable. A discrete aspect has enumerable values, and it can

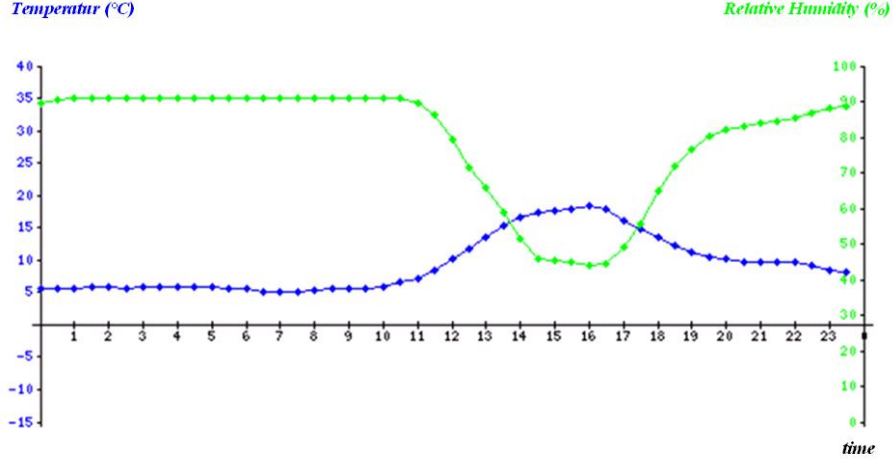


Fig. 2. An average outdoor temperature (blue) and relative humidity (green) measurement

be modelled as a factual concept, for example, the status of a device (on, off, or idle). Those aspects which are described by continuous numerical values are not straightforward to model. For example, the temperature of a place may be between -5°C and 37°C , depending on its spatial and temporal properties as well as other factors such as whether a heater is turned on. Moreover, such properties change frequently and thus two-valued assertions about them are not possible. Numerical properties can be modelled as beliefs by transforming the continuous values into discrete fuzzy values to imitate human reasoning. This process can be made semi-automatic. Consider a series of quantities, $x^{(1)}, x^{(2)}, \dots, x^{(n)}$, representing some sensor measurements, each $x^{(i)}$ being independently subject to a random variation. It is possible to define a probabilistic model for the random process in which a set of unknown model parameters, Θ , determine the probability distributions of $x^{(i)}$. Such probabilities, or probability densities, will be written in the form of $P(x^{(i)}|\Theta)$. Learning about Θ is possible if the system has observed the values of some of the $x^{(i)}$. The impact of these observations can be captured by the likelihood function:

$$L(\Theta) = L(\Theta|x^{(1)}, x^{(2)}, \dots, x^{(n)}) \quad (1)$$

Equation (1) yields the probability of the observed data as a function of the unknown parameters, which in turn is proportional to:

$$P(x^{(1)}, \dots, x^{(n)}|\Theta) = \prod_{i=1}^n P(x^{(i)}|\Theta) \quad (2)$$

Figure 2 shows temperature and relative humidity readings of an outside place for October 2006 in Dresden, Germany. To make the sensor readings more meaningful to human consumers, we transformed them into meaningful fuzzy sets. To compute the model’s parameters, i.e., the model’s statistical parameters, we identify various fuzzy regions for each reading, and determine its temporal characteristics. The regions were classified into those which exhibit constant, decreasing, and increasing characteristics. The time context, *ceteris paribus*, forces the measurements to decrease or increase or to remain constant, and hence was classified as: *morning*, *afternoon*, and *noon*. These regions were in turn used to determine the fuzzy members of the temperature and humidity measurements. For temperature, these regions were labeled as: *very cold*, *cold*, *lukewarm*, *warm*, and *hot*; for relative humidity, they were labeled as: *dry*, *moderate*, and *moist*. The same process was applied for other aspects (light intensity: *dark*, *visible*, *bright*, and *very bright*; and sound pressure: *quite*, *normal*, *loud*, *noisy*).

Once the fuzzy regions of a given aspect for a given place were identified, the next task is to compute the model’s parameters, $P(\textit{aspect} = x | \textit{place} = y, \textit{time} = z)$; for example, for the above readings, the probability, $P(\textit{temperature} = \textit{lukewarm} | \textit{place} = \textit{outdoor}, \textit{time} = \textit{noon})$, is equal to 0.45. The overall probability distribution of the temperature of an outdoor place at noon is given as: $\{\{\textit{very cold}, 0.15\}, \{\textit{cold}, 0.35\}, \{\textit{lukewarm}, 0.45\}, \{\textit{warm}, 0.05\}, \{\textit{hot}, 0.0\}\}$. After the model’s parameters were computed, the content of the ontology of figure 1 was updated to reflect the newly acquired knowledge.

4.5 Determination of logical and probabilistic relationships

Establishing relationship – logical or probabilistic – between the context of interest and the various aspects by which it is represented is the necessary step to determine a reasoning scheme for a context computing task. Since the context model integrates both beliefs and facts, the reasoning scheme should be able to manage beliefs and facts. In general, probabilistic schemes are suitable for low-level context recognition, whereas logic- or rule-based reasoning schemes can be employed for higher-level context disambiguation. This will be illustrated shortly.

A reasoning scheme should also be able to deal with missing data as it may not be feasible to foresee which of the aspects of a context will be captured at a given time. This requires the reasoning schemes to deal with all possible combinations of available sensors. For example, in the previous section we identified *time*, *temperature*, *relative humidity*, *light intensity* and *sound pressure* – five aspects – as relevant aspects of the context of interest. Provided a time context will be available all the time – as the other aspects are influenced by it –, there are altogether 15 different combinations of sensors if we assume a random variation of sensors. In general, the random availability of sensors is described by equation (3) and (4), where p refers to the number of all possible combinations; n refers to the total number of aspects which can describe a place as accurately and unambiguously as possible; and r refers to the aspects which can be captured by the available sensors.

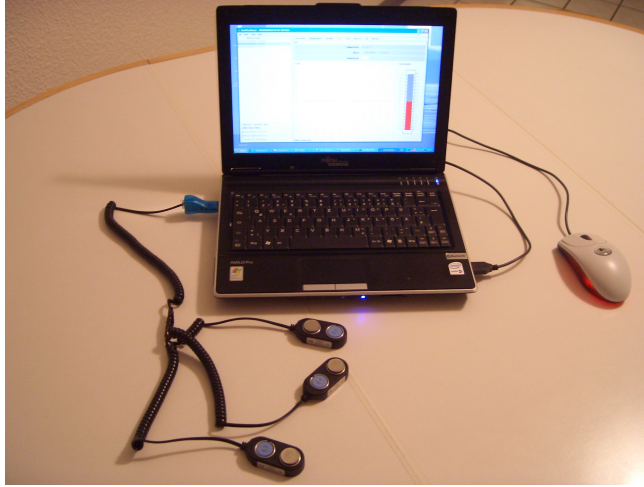


Fig. 3. The iButton® Sensor network implementing the 1-wire protocol

$$p = \sum_{r=1}^n \frac{n!}{(n-r)!r!} \quad (3)$$

For our scenario, this will be:

$$\sum_{r=1}^4 \frac{4!}{(4-r)!r!} = 15 \quad (4)$$

5 Validation

In the previous section we illustrated by examples the four steps of a context modelling approach. To demonstrate the expediency of this approach, we implemented a system which reasons about the whereabouts of a mobile user.

We set up a 1-wire network (Figure 3) with various iButton® sensor nodes on a laptop. The sensors we employed include: a DS1971-F3 and Java powered DS1957B data logger for storing secured profile information (the user’s name and password to monitor which user has logged on to a device); temperature and humidity loggers with different sensing parameters (DS1921G, DS1921Z-F5, and DS2422, DS1923), and a light intensity data logger (PCE- 172). The sensors have different accuracy, sensing range, and resolution. We randomly vary the sensor nodes to simulate dynamic availability of sensors. A context provider will receive an event notification whenever a node arrives at or departs from a 1-wire network. When a node arrives, the provider binds to it and queries it periodically. The query interval and duration is defined by the user. The context provider translates each context type into a fuzzy set to enable human-like reasoning.

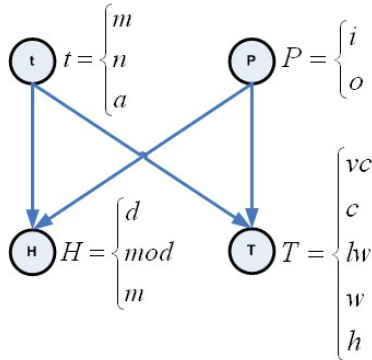


Fig. 4. A Bayesian Network with two parent nodes and two child nodes (relative humidity and temperature)

The output of the context provider are supplied to a Bayesian probabilistic reasoning scheme, which computes a posterior probability distribution for all potential places – the whereabouts of a person – and determines the one which is most likely. However, there are situations in which some places appear equally probable – for example, a corridor and a room – in which case, the system applies containment test to avoid inconsistent reasoning and random decision.

Figure 4, Figure 5, and Figure 6 display three different configurations of a Bayesian network. In Figure 4, three aspects of a place are captured, namely time, temperature, and relative humidity which were all measured by the sensors. The model’s belief for the probabilistic relationship is that *place* (P) and *time* (t) are parent nodes and *temperature* (T) and *relative humidity* (H) are child nodes². The various states each node can assume are as described in section 4.4.

When the data is obtained from the available sensors, they are mapped to corresponding linguistic variables each of which represents a particular aspect of a place, which is described by a fuzzy set. For example, 17°C is mapped to the linguistic variable *lukewarm* of a temperature fuzzy set; 45% is mapped to the linguistic variable *dry* of a relative humidity fuzzy set; 1000 Lux is mapped to the linguistic variable *bright* of a light intensity fuzzy set, etc.

Once the linguistic variables of each node are determined, the Bayesian Network computes posterior probability for each potential place (corridor, room, or outdoors) using equation (5), which computes the posterior probability of a place given measurements of temperature, humidity, and time. The place with the highest posterior probability becomes the most likely place to which the sensed data refer.

² Except the time context, which influences other primitive contexts and hence should be modelled as a parent context, all primitive contexts are modelled as child nodes. On the other hand, the higher-level context which should be inferred is modelled as a parent node.

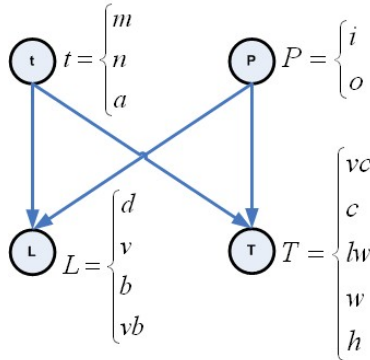


Fig. 5. A Bayesian Network with two parent nodes and two child nodes (light intensity and temperature)

For the case of Figure 4, the following input is an example: {time = September 15, 2005, 1:35 PM}, {temperature = 17°C}, {relative humidity = 45%}. As can be seen in equation (6), the sensed data are mapped to *lukewarm* (for temperature), *moderate* (for relative humidity) and *noon* (for time).

$$P(P|T, H, t) = \frac{P(T|P, t)P(H|P, t).P(P)}{\sum_P P(H|P, t)P(T|P, t).(P)} \quad (5)$$

$$P(P|T = lw, H = mod, t = n) = \frac{P(T = lw|P = c, t = n)P(H = mod|P = c, t = n)P(P = c)}{P(T = lw|P = c, t = n)P(H = mod|P = c, t = n)P(P = c) + P(T = lw|P = r, t = n)P(H = mod|P = r, t = n)P(P = r) + P(T = lw|P = o, t = n)P(H = mod|P = o, t = n)P(P = o)} \quad (6)$$

Applying Bernoulli's Principle of Insufficient Reason – a person can be anywhere with equal probabilities – to determine the probability distribution of a place, the posterior probabilities yield: {*corridor*, 0.457}, {*room*, 0.44}, and {*outdoors*, 0.1}. The difference in posterior probabilities between a CORRIDOR and a ROOM is not sufficiently large to discriminate between the two places. However, from the model description, we know that the two places are disjoint places, but both are subsumed by a building, which is an indoor place. Hence, the difference in posterior probabilities between both indoor places and outdoors is large enough to discriminate between INDOORS and OUTDOORS³. Here knowledge of containment relations between places has contributed to minimize uncertainty.

Figure 5 shows a different configuration for a similar topology, but for different availability of sensors - temperature and light intensity sensors. The posterior

³ To minimize error in decision making, we applied a heuristic-based decision threshold by setting the difference in posterior probability between two places to be more than 15%.

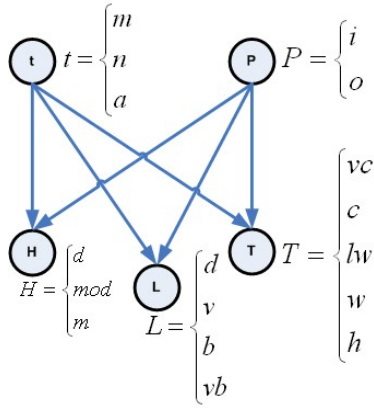


Fig. 6. A Bayesian Network with two parent nodes and three child nodes (relative humidity, temperature, and light intensity)

probabilities for this configuration are computed using equation (7), i.e., the conditional probability that a place is P given temperature, light intensity, and time measurements. This time the Bayesian Network was provided with the following sensor measurement: {time = September 15, 2005, 1:35 PM}; {temperature = 17°C}; and {light intensity = 1000 Lux}.

$$P(P|T, L, t) = \frac{P(T|P, t)P(L|P, t).P(P)}{\sum_P P(T|P, t)P(L|P, t)(P)} \quad (7)$$

The light intensity of 1000 Lux was mapped to the linguistic variable: visible. This time the posterior probabilities yield: {*corridor*, 0.39}, {*room*, 0.5}, and {*outdoors*, 0.1}. Still the difference between the posterior probabilities of a room and a corridor is not significant enough to discriminate between the two places; but the difference between the posterior probabilities of the two indoor places and outdoors is significant enough to discriminate between indoors and outdoors.

Figure 6 displays a Bayesian Network with three child nodes: temperature, light intensity, and relative humidity. The posterior probabilities for this configuration are computed using equation (8), according to which the probability of a place having a state P is computed given humidity, light intensity, temperature and time measurements. The sensor measurements provided to the network were: {Time = September 15, 2005, 1:35 PM}; {temperature = 17°C}; {light intensity = 1000 Lux}; and {relative humidity = 45%}.

$$P(P|T, H, L, t) = \frac{P(T|P, t)P(H|P, t)P(L|P, t)P(P)}{\sum_P P(H|P, t)P(T|P, t)P(L|P, t)(P)} \quad (8)$$

Computing the posterior probabilities for the three different places yields: {*corridor*, 0.38}, {*room*, 0.55}, and {*outdoors*, 0.06}. Now the discrimination gap between a room and a corridor has increased. Therefore, with this configu-

ration, the Bayesian Network could discriminate between a room and a corridor, and outdoors with minimised uncertainties.

6 Discussion and Conclusion

We motivated the integration of factual knowledge and beliefs about entities to model and reason about a context of interest. The motivation was followed by a twofold model comprising a probabilistic (beliefs) and a deterministic model (facts) for a comprehensive description of those aspects of a context we want to reason about. This model was produced in four steps: (1) identification of a context of interest; (2) identification of those aspects of the context which can be captured by employing sensors; (3) determination of meaningful contextual states for each aspect; and (4) determination of logical and probabilistic relationships between a context of interest and the various aspects it abstracts. We demonstrated our approach by modelling various physical places.

Our guideline can be employed for modelling several complex contexts. A health care application developer, for example, can employ our approach to determine the stress level (the higher-level context) of a person. Depending on available primitive context sources (e.g. oxygen sensors, blood pressure sensor, respiratory sensors, etc.) it is possible to determine with various degrees of uncertainty, whether a person is relaxed, mildly stressed, stressed, or significantly stressed. Other activities such as driving or maintaining a machine can also be modelled and reasoned about using our guideline.

Some final remarks regarding the modelling process: Determination of the contextual states of a primitive context may not be a straightforward process. Even though we encouraged the use of fuzzy sets, defining linguistic variables as well as their corresponding membership in a fuzzy set requires an adequate knowledge of the application domain as well as the characteristic and the range of measurements taken from the available sensors. For many real-world examples, membership functions are complex. This can be an indication to the complexity of context modelling and reasoning in general. The other challenge is the way the states of the primitive contexts are related to the higher-level context of interest. In our demonstration, we used Bayesian Networks to model conditional independence, and the conditional probabilities were derived from the membership functions of the fuzzy sets. However, direct transition of membership functions into conditional probability does not apply all the time.

A typical challenge we faced during our experiment was the response time of the sensors employed. Most of our sensors were Dallas semiconductor sensors, which were enclosed in a 16 mm thick stainless steel can. There was a delay of about 12 seconds before the sensors perceived an actual change in the environment. Though this might be acceptable for many human activities, for time critical applications this is not acceptable. Additional challenges included dealing with light intensity sensors: we had to make certain that light was received within the sensors specified incident angle; otherwise the amount received would fall significantly. A similar problem associated with light intensity was the

sensors sensitivity to surrounding objects which absorb or reflect light. In the presence of such objects, we frequently observed counterintuitive results.

References

1. H. Chen, T. Finin, A. Joshi, F. Perich, D. Chakraborty, and L. Kagal. Intelligent agents meet the semantic web in smart spaces. *IEEE Internet Computing*, 8.
2. J. Coutaz, J. Crowley, S. Dobson, and D. Garlan. Context is key. *Communications of the ACM*, 2005) pages = 49-53.
3. J. Crowley, J. Coutaz, and F. Bérard. Perceptual user interfaces: things that see. *Communications of the ACM*, 43(3).
4. W. Dargie. Dynamic generation of context rules. In *Lecture Notes in Computer Science*, pages 102–115, 2006.
5. A. Dey and G. Abowd. Cybreminder: A context-aware system for supporting reminders.
6. H. Gellersen, A. Schmidt, and M. Beigl. Multi-sensor context-awareness in mobile devices and smart artifacts. *Mob. Netw. Appl.*, 7(5), 2002), pages = 341-351.
7. K. Henriksen and J. Indulska. A software engineering framework for context-aware pervasive computing. In *Proceedings of the Second IEEE international Conference on Pervasive Computing and Communications (Percom'04)*, 2004).
8. P. Korpipää, J. Mäntyjärvi, J. Kela, H. Kernen, and E.-J. Malm. Managing context information in mobile devices. *IEEE Pervasive Computing*, 2003.
9. J. Pascoe, N. Ryan, and D. Morse. Using while moving: Hci issues in fieldwork environments. *ACM Trans. Comput.-Hum. Interact.*, 7(3):417–437, 2000.
10. D. Salber, A. Dey, and G. Abowd. The context toolkit: Aiding the development of context-enabled applications. *CHI*, pages 434–441, 1999.
11. A. Schmidt, M. Beigl, and H. Gellersen. There is more to context than location. *Computers and Graphics*, 23(6), 1999).
12. B. Shilit and M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, pages 22–32, 1994.
13. X. Wang, J. S. Dong, C. Chin, S. Hettiarachchi, and D. Zhang. Semantic space: An infrastructure for smart spaces. *IEEE Pervasive Computing*, 3(3):32–39, 2004.
14. M. Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun.*, 3(3):3–11, 1999.