Re Reasoning about Probability of Correctness using Bayesian Networks

Christiaan Baaij
c.p.r.baaij@student.utwente.nl

ABSTRACT
Context-aware ubiquitous systems use a large number of sensors to collect data from the environment. The system adapts its behavior to the environmental state and conditions. If this data or the source of the data gets corrupted then the context derived from such data would be erroneous and result in over-all system misbehavior. This research shows the use of Bayesian networks in determining the probability of correctness, a Quality of Context parameter. It describes the Bayesian networks for different sensor types and the fusion of these models. It shows how to use those Bayesian networks to infer the probability of correctness, even when some variables have unknown values.

Keywords
Quality of Context, Bayesian Networks

1. INTRODUCTION
In order to facilitate the interaction of humans with computers, ubiquitous systems integrate computation into the environment. This is done so that computers are not viewed as separate objects, but become part of the environment itself. Recent work includes making such ubiquitous devices and systems context-aware, enabling the devices or the systems to react and adapt to changes which take place in their domain of concern [NSL-05].

The context-aware services (CAS) provided by these ubiquitous systems will depend on the availability of context information which must be provided at the right time, in the right quality, and at the right place. The quality of this context information is neither identical to Quality of Service (QoS), nor the quality of the underlying components, i.e., Quality of Device (QoD). Rather, the precision, probability of correctness, trustworthiness, resolution, and up-to-dateness of context information form a new set of quality parameters which is known as Quality of Context (QoC) [BKS03].

For example, a temperature in a room can be sensed with two different thermometers, one measuring 17 degrees Celsius with a precision of 2 degrees and the other providing a value of 18.2 with a precision of 0.5 degrees. It is obvious that the latter piece of context information is more accurate than the former.

[BKS03] addresses why the introduction of QoC as quality notion is necessary. Several reasons are mentioned, such as:

- Reconstructing CAS behavior: Context information is used to automatically adapt services or the content they provide. Therefore, the imperfection of context information has a significant impact on the experiences users make with CASs.

- Selection of appropriate context providers: It is not unlikely that competing context providers deliver the same context information with different QoC. From the point of view of a CAS provider, QoC is then a valuable indicator to select an appropriate context provider.

- Adaptation of context refinement: The quality of low-level context information is an important indicator of whether or not the generation of high-level context information makes sense at all, and, if so, how to determine the quality of the produced context information.

This research will focus on using Bayesian networks for determining the probability of correctness, a quality parameter in Quality of Context. Probability of correctness, in the case of Quality of Context, can be best defined as: The probability that the derived context provided by a CAS corresponds with the actual environmental context. Previous research has investigated the use of the probability of correctness [BKS03][GS01][BG95]. However, this research adds to this field of study by proposing a technique that actually determines/acquires the probability of correctness.

For scoping reasons: This research will only focus on probability of correctness, and not on any of the other parameters in Quality of Context mentioned in [BKS03]. The reasons Bayesian networks are used is that this form of belief representation does not require every influential evidence to be known but will still be able to provide a representation of the probability of correctness. Also it allows for domain knowledge about causality and fault probabilities to be easily applied in the belief representation network.

The remainder of this paper is structured as follows. Section 1.1 contains related work as part of the literature search. Section 2 gives the definition of Probability of Correctness as it is used in this research. Section 3 gives a basic overview of the workings of a context-aware system. Section 4 describes the use of Bayesian networks to determine the Probability of Correctness. Finally, section 5 draws some concluding remarks.

1.1 Related Work
A Bayesian network is a form of a probabilistic graphical model, also known as a Bayesian belief network. A Bayesian network is a directed acyclic graph whose:

- Nodes represent variables. Each node is annotated with quantitative probability information.

- Arcs represent statistical dependence relations among the variables and local probability distributions for each variable given the values of its parents.

Bayesian networks have been used in fault detection and diagnosis of dynamic systems [LPKB00]. The work has been focused on domains related to the control and supervision of large industrial processes involving mixtures of continuous and discrete variables. The main technique in this work includes hybrid dynamic Bayesian networks which capture the stochastic nature of the process and accommodate all type of system variables both discrete and continuous. The application of learning Bayesian networks from system data has also been used for fault detection in large dynamic systems. This method explores the learning capability of Bayesian networks from measurements of the relevant signals that are present in the dynamic system by the use of a learning algorithm [MY04].
Bayesian networks have been successfully used in anomaly detection. Naïve Bayesian networks have been employed for detecting anomalies in active networks for providing intrusion detection services [SOS02]. Similarly Bayesian networks have also been used for developing self-aware services which use Bayesian networks to detect any anomaly in their own behavior while functioning on the internet [BDD+01].

The ontology-based context model for the SOCAM Architecture [GWPZ04] allows associations (OWL properties) between properties of entities and quality constraints indicating QoC. Four types of most common quality parameters are defined: accuracy, resolution, certainty and freshness. A metric consists of the triple (unit, type, value).

[ALLZ05] describes the use of Bayesian networks in default scenario’s in context-aware ubiquitous systems for fault detection. It describes default Bayesian networks for sensors, actuators and the combinations of these sensors and actuators to form a representation of a Context-Aware system. Using the constructed Bayesian Network, the source of a fault is inferred. The sensor model described in [ALLZ05] is being used in this research for the physical sensor described in section 4.1. This model allows for the probability of correctness to be inferred based on the sensor reading and the state in which the sensor resides. The way this research differs from the research in [ALLZ05] is that [ALLZ05] uses Bayesian networks to determine if a sensor is working incorrectly, whereas this research uses Bayesian networks to determine the probability that the sensor reading correctly corresponds with the actual measured quantity. It should be noted that even if a sensor is in a faulty (incorrect) state, the sensor reading might still correspond with the actual physical quantity. For example, a faulty temperature sensor might be able to correctly measure extreme temperatures but fail to discriminate between moderate temperatures.

2. PROBABILITY OF CORRECTNESS

Probability of Correctness denotes the probability that a piece of context information is correct. Consider a sensor network of temperature sensors. These sensors might fail and start providing wrong data, e.g., measuring 10 degrees Celsius, while the correct value is 20 degrees Celsius. With the parameter Probability of Correctness a context provider estimates how often context information will be unintentionally wrong because of internal problems [BKS03]. Like mentioned earlier, a faulty sensor doesn’t always have to provide incorrect context information, but the probability that the context information is incorrect might increase.

Note that Probability of Correctness is different from resolution effects where a system, in the case of a language detection system, might not be able to distinguish between various Slavic languages (Russian, Polish) [GS01].

3. CONTEXT AWARE SYSTEMS

In a context-aware ubiquitous system the entire perception mechanism of a system is composed of a number of diverse sensors deployed in the environment to monitor various (physical) quantities. A number of controllers or actuators are used by the system to respond to various changes which take place in the environment. The detection of such changes and the formation of context based on these changes is dependent on the data sensed from the monitored environment [NSL+05].

In any particular scenario the steps taken by the system can be defined as sensing some data from the environment and acting/adapting to this data. The action taken in the light of the sensed data is determined through various factors such as available resources, the derived context and user preferences. Every step taken by the system that changes the environment also involves sensing data which is needed for validating if the action has indeed succeeded [ALLZ05]. Such ‘sensing’ can either be done using an actual sensor, or by user interaction. The complete interaction cycle in a scenario is shown in figure 1.

![Sensing Data](image)

Figure 1: Interaction cycle of a context-aware system.

3.1 Sensor Types

The word ‘sensor’ not only refers to sensing hardware but to every data source which may provide usable context information. Concerning to the way data are captured, sensors can be classified in three groups [IS03][BD04].

- **Physical sensors**: The most frequently used type of sensors are physical sensors. Many hardware sensors are available nowadays capable of capturing almost any physical data.
- **Virtual sensors**: Virtual sensors source context data from software. E.g. it is possible to determine an employees location not only by using tracking systems (physical sensors) but also by a virtual sensor, e.g. by browsing an electronic calendar, a travel booking system, emails etc. for location information. Other context attributes that can be sensed by virtual sensors include e.g. the users activity by checking for mouse movement and keyboard input.
- **Logical sensors**: These sensors make use of a couple of information sources, they combine physical and virtual sensors with additional information from databases etc. to solve a higher task. E.g. by analyzing logins at desktop PC’s and a database mapping fixed devices to location information a logical sensor can be constructed to detect an employees current position.

Models for the sensor types are defined in the next section. Developers can categorize sensors under the above classifications so that they can apply the correct model for each sensor. These models are used to determine the probability of correctness.

4. USING BAYESIAN NETWORKS

The following section describes the scheme to determine probability of correctness using Bayesian Networks. The section starts with models of the different sensor types described earlier. At the end of the section a complete belief representation for a combined sensor system is given and its workings explained in an example.

4.1 Modeling Sensors

Modeling a physical sensor requires three types of information.

- The status of the sensor.
- The quantity or event monitored.
- The actual reading of the sensor.
The state of the sensor represents the correctness and can be determined by taking into account certain factors such as the age of the sensor equipment, its reliability as provided by the vendor etcetera. The quantity or event being monitored can for example be temperature for a heat sensor. The main reason for taking the physical quantity into account in the belief network is purely causal, because it is the physical quantity which causes the sensor to change its value. The belief network also includes the behavior of the sensor as represented by its actual reading. The state of the sensor should also be a part of the network, as an incorrectly working sensor most likely responds differently to the physical quantity compared to a correctly working sensor [ALLZ05]. These three variables are sufficient to correctly model a sensor. The state of the sensor and the monitored quantity are assumed to be mutually independent and are hence modeled as such. Figure 2 shows a Bayesian network depicting a sensor.

\[ P(SS, QE, SR) = P(SR|SS, QE) \times P(SS) \times P(QE) \]  

(1)

The initial specification of the model includes three potentials namely:
- \( P(SS) \): The prior beliefs of the sensor state
- \( P(QE) \): Prior beliefs about the monitored quantity
- \( P(SR|SS, QE) \): The conditional beliefs about sensor behavior given the sensor state and the monitored quantity.

Evidence for the model comes in the form of sensor data and is absorbed at the connecting variable, the node that represents the sensor reading (SR). This evidence renders the other two nodes dependent [Jen01], such that based on their prior belief measures their posterior beliefs in light of recent evidence can be computed easily [Jen01]. This can be done using evidence propagation algorithms for Bayesian belief networks, such as Pearl’s Belief Propagation Algorithm [Pea88]. This means that at any instance a sensor reading can be used for determining the actual physical quantity. The actual sensor reading is entered as an evidence ‘e’ into the network, using any algorithm for evidence propagation the posterior beliefs about the sensor state and the physical quantity being measured can be calculated separately.

\[ P(SS, QE, SR, e) = P(SS, QE, SR) \cdot e \]  

(2)

Equation (2) represents the absorption of the evidence into the network and equation (3) shows the belief about the physical quantity as a consequence of the evidence. The belief measure about the sensor state can also be calculated using equation (3) by interchanging \( QE \) with \( SS \).

\[ P(QE|e) = \frac{\sum_{SS, e} P(SS, QE, SR, e)}{P(e)} \]  

(3)

If the systems suspects the sensor is in a certain state, this information can then also be entered into the network. The actual sensor reading is then entered as an evidence \( e_1 \) and the suspected state of the sensor is entered as an evidence \( e_2 \). Equation (4) represents the absorption of this evidence into the network. Using equation (5) the belief about the physical quantity can be calculated as a consequence of the evidence \( e_1 \) and \( e_2 \).

\[ P(SS, QE, SR, e_1, e_2) = P(SS, QE, SR) \cdot e_2 \cdot e_1 \]  

(4)

\[ P(QE|e_1, e_2) = \sum_{SS, SR, e_1, e_2} P(SS, QE, SR, e_1, e_2) \cdot \frac{P(e_1, e_2)}{P(e)} \]  

(5)

### 4.1.1 Virtual Sensors
Virtual sensors can be modeled much like the physical sensors described earlier. The physical quantity or event is replaced by a virtual quantity or event. Just like the physical sensor, the virtual sensor also has a state that represents the correctness. The belief network also includes the behavior of the sensor as represented by its actual reading. Because virtual sensors are modeled just like physical sensor, the Bayesian network in figure 2 also depicts a virtual sensor.

The equations (1) - (5), used for physical sensors can also be applied to virtual sensors as they have the same Bayesian network.

### 4.1.2 Logical Sensors
Much like the physical and virtual sensor, four types of information are needed:
- The status of the sensor.
- The actual reading of the sensor.
- The quantities or events used.
- The quantity or event generated

Again, the state of the sensor represents the correctness. The way logical sensors differ from the physical and virtual sensors is that up to \( n \) quantities or events are used for the reading. The sensor uses those quantities combined with additional resources to determine the logical quantity. Figure 3 shows a Bayesian network depicting a logical sensor.

\[ P(S, Q, R, e) = P(SS, QE, SR) \cdot e \]  

(2)

Equation (2) represents the absorption of the evidence into the network and equation (3) shows the belief about the physical quantity as a consequence of the evidence. The belief measure about the sensor state can also be calculated using equation (3) by interchanging \( QE \) with \( SS \).

\[ P(Q|e) = \frac{\sum_{SS, R, e} P(SS, Q, SR, e)}{P(e)} \]  

(3)

According to the model shown in figure 3, the logical quantity or event is conditionally independent from the sensor state and the \( n \) quantities or events, given the additional resources in \( SR \). What this means, is that the logical quantity is certainly influenced by the state and the \( n \) quantities, but not directly influenced. The
state and the quantities only have influence on the logical quantity through the workings and configuration of the sensor. Any evidence on the sensor reading makes the sensor state and the used quantities or events dependent on each other. According to the chain rule for Bayesian networks the joint probability distribution of the model is given by the equation:

\[
P(SS, QE_s, SR, LQE) = \frac{P(SS) \times P(QE_s | SS) \times \ldots \times P(QE_n | SS) \times P(SR) \times P(LQE | SR)}{P(SR) \times P(LQE | SR)}
\]

The initial specification of the model includes the following, potentials namely:

- \( P(SS) \): The prior beliefs of the sensor state
- \( P(QE_1) \ldots P(QE_n) \): Prior beliefs about the \( n \) quantities or events
- \( P(SR | QE_1 \ldots QE_n, SS) \): The conditional beliefs about sensor behavior given the sensor state and the used quantities or events.
- \( P(LQE | SR) \): The conditional beliefs about the logical quantity or event given the sensor settings and information at the additional resources.

Evidence, just like for the physical sensor, comes in the form of sensor data and is absorbed at the connecting variable, the node that represents the sensor reading (SR). This evidence renders the other nodes dependent, such that based on their prior belief measures, their posterior beliefs in light of recent evidence can be computed easily. This can be done using evidence propagation algorithms for Bayesian belief networks, such as Pearl’s Belief Propagation Algorithm [Pea88]. Similarly to the equations for physical sensors, equation (7) represents the absorption of the evidence into the network and equation (8) shows the belief about one of the used quantities or events as a consequence of the evidence.

\[
P(SS, QE_1 \ldots QE_n, SR, LQE, e) = P(SS, QE_1 \ldots QE_n, SR) \cdot e \quad (7)
\]

\[
A = \{ SR, SS, QE_1 \ldots QE_{n-1}, QE_n, LQE, e \}
\]

\[
P(QE_i | e) = \frac{\sum_{A} P(SS, QE_1 \ldots QE_n, SR, LQE, e)}{P(e)} \quad (8)
\]

### 4.1.3 Combining the sensor models

Now that the independent models for representing physical/virtual sensors are defined, they need to be linked together to represent an actual sensor system that combines the various sensor types. The combinations of using various physical/virtual sensors in combination with a logical sensor is shown in figure 4. In this figure it can be seen that the \( QE \) node of the original physical/virtual sensor model has been replaced by a \( QE \) node of the logical sensor model. This is because \( QE \) and \( QE_e \) are the same quantity or event, where in a physical/virtual sensor it is monitored and in a logical sensor it is used in combination with other quantities and events. The original \( SR \) in the logical sensor modeled shown in figure 3 is named \( LSR \) in figure 4 to avoid confusion with the \( SR_i \)’s of the linked models of the physical/virtual sensors.

Now that this combination is defined, all that is left is to link a logical sensor to another logical sensor. This combination is seen...
in figure 5. In this figure it can be seen that the LQE nodes of the original logical sensor model of one sensor has been replaced by the QE, node of the logical sensor model of the other sensor. This is because LQE is a quantity the other logical sensor will use to determine its sensor reading $S_R$.

To completely model a combination of physical, virtual and logical sensors linked together the models depicted in figure 4 and figure 5 simply need to be fused. In such a completed model actual sensor values would act as the evidence for the completed model. This evidence would then be propagated through the network using any of the algorithms for evidence propagation in Bayesian Networks [Jen01]. In light of this evidence the probabilities about the monitored quantities can be calculated and the probability of correctness of the supplied contexts can then be determined. However, to correctly calculate the probability of correctness of the contexts supplied by the logical sensors, evidence shouldn’t be entered at the $S_R$ nodes of the logical sensor model. Doing so would directly determine the probabilities at the $LQE$ node, rendering it independent from the probabilities of the state and the used quantities of the logical sensor, which is incorrect.

4.2 Examples

The following 2 examples will describe how to use the different models and how to interpret their results. The first example will use the model for the physical sensor and the second example will use a fusion of the models in figures 4 and 5. The examples are simulated using the MSBNx [SI] Bayesian network authoring and evaluation tool.

4.2.1 Example 1: Temperature Sensor

As an example for a physical sensor a temperature sensor is chosen. Figure 6 shows a Bayesian network for this example developed according to the presented technique. Table 1 gives the description of the variables in the network. The state variable $State_{TS}$ has been given the probability distribution of $<0.9, 1.0>$, where 0.9 corresponds with the ‘Correct’ state and 0.1 corresponds with the ‘Incorrect’ state. Similarly, the prior distribution for the external temperature is given as $<0.15, 0.65, 0.15, 0.05>$, corresponding with the variable state description in table 1. These prior distributions for the external temperature (during a certain season) could for example be offered by a national weather service. The probability for the temperature reading ($Temp_{Read}$) is given in table 2. Now, if evidence is entered into the network in the form of sensor readings, making no assumptions about if the sensor is malfunctioning or not, the probability of correctness can be derived. The following evidence is entered into the network:

- $e_1$: $Temp_{Read} <0.0, 0.0, 1.0, 0.0>$

The hypothesis variables, namely the sensor state and the external temperature are given as follows:

- $State_{TS} =$ $<0.8654, 0.1246>$
- $Ext_{Temp} =$ $<0.0000, 0.0625, 0.9327, 0.0048>$

The above hypothesis variables show that the sensor is working correctly with a belief of 86.54%. The probabilities that this research is most interested in are the ones which corresponded to the belief of the external temperature. These show that the external temperature is believed to be in the range 21-30, corresponding to the sensed range, with a belief of 95.27%. According to the presented technique, the probability of correctness of this context information is thus 93.27%.

If, by some means, the system expects that the sensor is probably working incorrectly, the probability of correctness can be determined in the following way. The evidence that is entered into the network is:

- $e_1$: $State_{TS} <0.0, 1.0>$
- $e_2$: $Temp_{Read} <0.0, 0.0, 1.0, 0.0>$

The hypothesis variable is given as follows:

- $Ext_{Temp} =$ $<0.0000, 0.4643, 0.5000, 0.0357>$

The above hypothesis variable shows that the external temperature is in the range of 21-30, corresponding with the sensor reading, with a belief of 50.00%. According to the presented technique, the probability of correctness of this context information is thus 50.00%.

4.2.2 Example 2: Driving sensor

As an example of the fusion of the logical and physical sensor models, consider the following scenario:

A seatbelt sensor detects if the driver seatbelt is clicked into the lock, another sensor measures the pressure on the driver seat. These sensor values, either combined or separate, are used to determine if the driver seat is occupied. Another sensor measures if the car is accelerating in any direction, this, combined with the knowledge if the driver seat is occupied is used to determine if someone is driving the car. An application using such a context could for example be a fail-safe mechanism that activates the emergency brakes if the car is moving (accelerating) when the drivers seat is not occupied (e.g. when a car is parked on a slope).

Figure 7 shows a Bayesian network for this example in which, according to the presented technique, the seatbelt lock, driver seat pressure, and accelerometer are modeled as physical sensors. These are then fused to the occupancy and driving sensors which are modeled as logical sensors. Table 3 gives the description of the variables in the network. The set of probability distributions

<table>
<thead>
<tr>
<th>Table 1: Variable Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Name</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>$State_{TS}$</td>
</tr>
<tr>
<td>$Ext_{Temp}$</td>
</tr>
<tr>
<td>$Temp_{Read}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Probability Distribution $Temp_{Read}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$State_{TS}$</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>Correct</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Incorrect</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

| Figure 6: The Bayesian network for example 1 |
needed for this example are too large to be completely presented in this paper. Instead, only the prior distributions of the state variables and the physical quantities are provided.

The state variables have been given the probability distributions of \(<0.9, 0.1>\), where 0.9 corresponds with the ‘Correct’ state and 0.1 corresponds with the Incorrect state. Similarly, the prior distribution for the seatbelt being locked in is given as \(<0.75, 0.25>\) and for the driver seat pressure and acceleration it is given as \(<0.8, 0.2>\); all three corresponding with the variable state descriptions given in table 3.

Table 3: Variable Description Example 2

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State_Dr*</td>
<td>Correct</td>
<td>State of the pain driver seat is occupied</td>
</tr>
<tr>
<td>DR_Read</td>
<td>Yes, No</td>
<td>Sensor reading of the driving sensor</td>
</tr>
<tr>
<td>Driving</td>
<td>Yes, No</td>
<td>Prior probability that the car is being driven</td>
</tr>
<tr>
<td>Occupied</td>
<td>Yes, No</td>
<td>Prior probability that the driver seat is occupied</td>
</tr>
<tr>
<td>Acceleration</td>
<td>Yes, No</td>
<td>Prior probability of acceleration of the car</td>
</tr>
<tr>
<td>AC_Read</td>
<td>Yes, No</td>
<td>Sensor reading of the accelerometer</td>
</tr>
<tr>
<td>OC_Read</td>
<td>Yes, No</td>
<td>Sensor reading of the driving seat occupation sensor</td>
</tr>
<tr>
<td>Seatbelt</td>
<td>Yes, No</td>
<td>Prior probability that the seatbelt is locked in</td>
</tr>
<tr>
<td>Pressure</td>
<td>Yes, No</td>
<td>Prior probability that pressure is applied</td>
</tr>
<tr>
<td>SB_Read</td>
<td>Yes, No</td>
<td>Sensor reading of seatbelt lock sensor</td>
</tr>
<tr>
<td>PR_Read</td>
<td>Yes, No</td>
<td>Sensor reading of driver seat pressure sensor</td>
</tr>
</tbody>
</table>

*State_OC, State_AC, State_SB and State_PR are modeled in a similar manner

Probability distribution for the sensor readings of the driver seat pressure sensor are given in table 4. Similar probability distributions have also been specified for the seat belt lock sensor and the accelerometer. The prior beliefs of the ‘Driving’ and ‘Occupied’ nodes are configured to represent the assumption that the sensor readings DR_Read and OC_Read are correct 90% of the time. OC_Read and DR_Read are configured to represent the mechanisms described in the aforementioned scenario.

Table 4: Probability Distribution PR_Read

<table>
<thead>
<tr>
<th>State_PR</th>
<th>Pressure</th>
<th>PR_Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>Yes</td>
<td>0.7783</td>
</tr>
<tr>
<td>No</td>
<td>0.2217</td>
<td></td>
</tr>
<tr>
<td>Incorrect</td>
<td>Yes</td>
<td>0.8002</td>
</tr>
<tr>
<td>No</td>
<td>0.1998</td>
<td></td>
</tr>
</tbody>
</table>

Now, if evidence is entered into the network in the form of sensor readings, making no assumptions about if the sensors are malfunctioning or not, the probability of correctness can be derived. The following evidence is entered into the network:

- \( e_1: PR\_Read <1.0, 0.0> \)
- \( e_2: SB\_Read <1.0, 0.0> \)
- \( e_3: AC\_Read <1.0, 0.0> \)

The hypothesis variables used to determine the probability of correctness for the different sensors are given as follows:

- Pressure = \(<0.9910, 0.0090>\)
- Seatbelt = \(<0.9828, 0.0172>\)
- Occupied = \(<0.8679, 0.1321>\)
- Driving = \(<0.8002, 0.1998>\)

The probabilities that this research is most interested in are the ones corresponding to the belief of the logical quantity ‘Driving’. These show that the car is being driven with belief of 80.02%. Now, if the ‘Driving’ sensor reported that the car was being driven, the probability of correctness of this context information would be 80.02% according to the presented technique. Now, if for example the seatbelt lock sensor got disconnected, its input would not be available to the system. Again, making no assumptions about if the sensors are malfunctioning or not, the probability of correctness can be derived. The following evidence is entered into the network:

- \( e_1: PR\_Read <1.0, 0.0> \)
- \( e_2: AC\_Read <1.0, 0.0> \)

The hypothesis variables used to determine the probability of correctness for the different sensors are given as follows:

- Pressure = \(<0.9910, 0.0090>\)
- Seatbelt = \(<0.7500, 0.2500>\)
- Occupied = \(<0.8326, 0.1674>\)
- Driving = \(<0.7783, 0.2217>\)
As can be seen, the hypothesis variable for the seatbelt being locked in has the same distribution as the prior probability distribution defined earlier. This is correct, because without the input of the seatbelt lock sensor no other assumptions could be made in the light of this evidence. However, the probabilities that this research is most interested in, the ones which correspond to the belief of the logical quantity ‘Driving’, can still be inferred. Now, if the ‘Driving’ sensor reported that the car was being driven, the probability of correctness of this context information would be 77.83% according to the presented technique.

5. CONCLUSIONS
This research has presented a technique to determine the probability of correctness in a context-aware system using Bayesian networks. It allows for the use of domain knowledge gathered at the different development stages and can handle unknown values for variables used in context gathering and reasoning. Future developers using this technique are free to choose the granularity for the probability distributions in the nodes, in exchange for the resolution of the probability of correctness.

However, even though this technique allows for domain knowledge to be applied in the form of the probability tables for the nodes in the Bayesian network, this is also its current weakness. Those probability tables might not always be available. Future work lies in a technique to adapt and tune the Bayesian networks over time to compensate for this weakness, for example by using Bayesian learning techniques or by some form of user interaction.

ACKNOWLEDGMENTS
The author thanks Ing Widya and Bert-Jan van Beijnum for their valuable comments and thanks Robyn Herring for helping out with correcting sentence construction.

REFERENCES


