The Right Thing To Do: Automating Support for Assisted Living with Dynamic Decision Networks

Nayyab Zia Naqvi*, Davy Preuveneers*, Wannes Meert† and Yolande Berbers*
*iMinds-DistriNet,†DTAI
Department of Computer Science, KU Leuven, 3001 Leuven, Belgium
Email:{nayyab.naqvi, davy.preuveneers, wannes.meert, yolande.berbers}@cs.kuleuven.be

***Abstract*—In an era where ubiquitous systems will be mainstream, users will take a more passive role and these systems will have to make smart decisions on behalf of their users. Automating these decisions in a continuously evolving dynamic context can be challenging. First of all, the right thing to do usually depends on the circumstances and context at hand. What might be a good decision today, could be a bad one tomorrow. Secondly, the system should be made aware of the impact of its decisions over time so that it can learn from its mistakes as humans do. In this paper, we formulate a technique for decision support systems to mitigate runtime uncertainty in the observed context, and demonstrate our context-driven probabilistic framework for ubiquitous systems that addresses the above mentioned challenges. Our framework incorporates end-to-end Quality of Context (QoC) as a key ingredient to make well-informed decisions. It leverages Dynamic Decision Networks (DDN) to deal with the presence of uncertainty and the partial observability of context information, as well as the temporal effects of the decisions. Our experiments with the framework demonstrate the feasibility of our approach and potential benefits to automatically make the best decision in the presence of a changing environment.**

Keywords: quality of context, dynamic decision networks, autonomic decision support

I. Introduction

With the rapid development of ubiquitous computing technologies, the concept, the principle architecture, and the mechanisms behind decision support systems evolve significantly. Ubiquitous support systems operate in an environment that proactively assists people in their daily lives under changing circumstances. In such a dynamic situation, what might be a good decision in one scenario, could be an awful one for another scenario. As such, the best decision usually depends on the current situation. The major challenge that such ubiquitous systems face is being aware of the impact of their decisions and actions over time so that they can reason about the outcomes of their actions and learn from their mistakes as humans do.

Context [1] is defined as *any information that can be used to characterize the situation of an entity*, where an entity can be an object, place or person relevant to the current scope of the system. Context data drives the decisions and actions in an ubiquitous environment. Making well-informed decisions is a key to reduce the risk of negative outcomes. For instance, in healthcare applications, it is crucial to have high confidence in the quality of the information upon which decisions are based. However, the Quality of Context (QoC) can highly affect the decision outcomes of the system. As context can be inherently ambiguous, we need certain attributes that signify the adequacy or degree of suitability of the context data. This degree of suitability is often regarded as the *Quality of the Context*. QoC can be defined [2] as any information that describes the quality of the information used as context data. It involves any attribute that reflects the desired quality characteristics in order to make well-informed decisions in an ubiquitous system. QoC addresses the erratic nature of context data, especially when gathered from sensors or when requested from unfamiliar resources. This way we can express the confidence in, for example, getting accurate location information of a person from a GPS device or information about his current activity based on audio or acceleration sensors in his smartphone. The relevant QoC attributes – such as precision, accuracy, timeliness, trustworthiness – are influenced by the required type of context and vary from case to case.

Due to the dynamism in ubiquitous environments, we cannot anticipate all eventualities. Hence, there is a strong need to deal with the uncertainty in the context data. Decisions made solely on the available context information can hamper the overall working of the system. Quantifying and ensuring the QoC is a fundamental issue since a subpar QoC can compromise the correctness and desirability of the decisions. To assure the effectiveness and user satisfaction, the QoC has to consider the quality of both the exchanged context data and the context distribution process in an end-to-end manner. Also, the runtime quality assessment can significantly impact the decisions and performance. Furthermore, these systems face a challenge in terms of proactive support to the users. To initialize the desirable services automatically in an unobtrusive way is not straightforward due to the runtime uncertainty in the context [3]. The uncertainty in the context data can lead to annoying or wrong decisions. It can also have a negative impact on the ability to reliably predict future contexts for proactive decision support. For example, a diabetic patient could receive smart advice for insulin dosage from a medicine recommender system based on his current insulin level and the physical activities he has planned for the remainder of the day. However, low-quality context could lead to wrong context predictions and possibly bad decisions for insulin dosage. Our approach aims to learn from the previous decisions in order to improve the quality of the decisions and actions by our system.

In this paper we discuss a scenario in the area of assisted living and demonstrate our context-driven decision support framework that addresses the above challenges. Given a decision that requires certain context types, the impact of desired QoC attributes can be modeled using probability distributions. To the best of our knowledge, none of existing approaches address the QoC requirements for an ubiquitous assisted living

environment capturing the runtime uncertainty in decision making. Therefore, the major contribution of our work is a novel approach that incorporates end-to-end QoC as a key ingredient to make well-informed decisions. Our framework leverages Dynamic Decision Networks (DDN) to deal with the presence of uncertainty and the partial observability of context, as well as the temporal effects of the decisions. Experiments with our framework demonstrate the feasibility of the approach and potential benefits to automatically make the best decision in the presence of changing situations or circumstances. The applicability of DDNs is an emerging research topic, and its use in the area of self-adaptation for autonomous systems was recently investigated [4] for a specific application on remote data mirroring.

This paper is structured as follows. In the next section, we give an overview of the related work. Section III highlights our use case scenario and the subsequent section explores the requirements for the scenario. Section IV provides a brief account of DDNs and the details about our approach to ensure end-to-end QoC management using a probabilistic decision support model to mitigate the runtime uncertainty. Finally, after evaluating our approach applied on our use case scenario in Section V, the paper concludes and offers a discussion of topics of interest for future work in Section VI.

II. Related Work

The importance of the relevance of the data has been identified since the very early work in context-aware ubiquitous systems. Despite of its importance few works have been carried out to quantify the QoC and ensure the quality of the information on which the decisions and actions of context-aware systems are based. Moreover, many decision support techniques are insufficient to address the runtime uncertainty in the context information as they do not assess the QoC in an end-to-end fashion. We formulate the related work into two parts:

1) QoC attributes and modeling aspects
2) Runtime decision support addressing uncertainty

A. QoC attributes and modeling aspects

QoC attributes and their modeling comes into play to capture the uncertainties in context data. A well designed model is a key assessor to the quality of the context. Krause and Hochstatter [5] discussed the challenges in modeling the QoC and its usage. Several QoC models emerged but most of the QoC models were not properly quantified according to the QoC requirements. Castro et al. [6] investigated the accuracy of a location sensing service was compromised. Ranganathan et al. [7] used predicate and probabilistic logic to assign confidence values from 0 to 1 for every type of context. Dey et al. [1] suggested that mediation with the user can help to reduce the uncertainty. QoC is modeled as meta-data where additional information about the quality of the context source is utilized to capture the uncertainty. Krause and Hochstatter [5], Buccholz et al. [2] and Manzoor [8] realized the imperfection of the context data as QoC classifying the quality in two classes as QoC parameters and QoC sources.

Sheikh et al. [9] identify several quality indicators like precision, freshness, spatial resolution, temporal resolution and probability of correctness. The authors propose that these quality indicators are well-suited in ubiquitous systems for healthcare. However, no quantification mechanism has been proposed by the authors in order to evaluate the role of these parameters in critical decision support. Kim et al. [10] present the quality dimensions such as accuracy, completeness, representation consistency, access security and up-to-dateness for measuring QoC in ubiquitous environments. Moreover, they present an average based aggregation method for evaluating the Quality of Context Information (QoCI) completeness of a set of raw context information. The authors describe an objective evaluation of the level of information quality through the proposed method. The evaluation mechanism provided in this work is only suitable under circumstances where continuous data is being generated by the sensors. Similarly, the completeness measure also revolves around the ratio of available attributes provided by the sensors and the total number of attributes. Thus, in general the proposed mechanism is not valid for dynamic open ended transient networks. Filho et al. [11] present an approach for measuring QoCI that can be applied to raw, inferred and derived information offering three different aggregation approaches: minimum, maximum, and average methods. The work of Manzoor et al. [8] presents methods for evaluating QoCI up-to-dateness, trust-worthiness, completeness, and significance.

B. Runtime decision support addressing uncertainty

Ubiquitous systems have to take into account uncertainty in context data at runtime. In these systems context sources are dynamic in nature. They can disappear and re-appear at any time and context models change to include new context entities and types. The properties of context sources and context types can change randomly and the uncertainty can vary too. Fenton and Neil [12] have used Bayesian networks for predictions of the satisfaction of non-functional aspects of a system. Esfahani et al. [13] employ fuzzy mathematical models to tackle the inherent uncertainty in their GuideArch framework while making decisions on software architectures. Dynamic configuration of service oriented systems was investigated by Filieri et al. [14]. In contrast to our model, they used Markov models to investigate the decision making under uncertainty and quality requirements. Our approach focuses on evaluating the role of QoC for decision support in highly dynamic and open ended ubiquitous systems. We have a different perspective to tackle the challenge of ever changing contexts and making decisions in time based on that context. We have used probability reasoning with Bayesian networks and Decision networks [15] to tackle the real-time decision problems under uncertainty and QoC requirements of the users. Moreover, we focus on runtime aspects of the uncertainty of the context data and their impact on the actions of these systems leveraging DDNs. We model a real-time ubiquitous system that dynamically changes over time. Its context and QoC requirements for each user also evolve over time. Our model aims to learn from the previous decisions in order to improve the quality of the decisions and actions by our system.

III. Use Case Scenario

Recent advances in ICT have changed the notion of active aging from merely enhancing the quality of life as people age,

Function	Brief description
Automate Home	Deals with independent living
Remind Medication	Deals with the forgetfulness for well being
Schedule Exercise	Deals with physical fitness
Active	Deals with physical problems and supports good routines
Reminiscence	Cops with the cognitive ageing
Fall	Deals with the care to fight frailty

TABLE I. FUNCTIONS OF AN AMBIENT ASSISTED LIVING SYSTEM

towards policies that enable elderly to lead socio-economically independent lifestyles. Mobility is seen as an essential decisive factor to maintain an altogether autonomous living. Physical wellness and active aging are highly inter-linked. Being physically active would not only assist the elderly in managing certain illnesses (such as diabetes, Parkinson disease) but also encourage them to actively participate in social interaction, self-development and recreation. Assisted living systems and the relationship with context awareness have been outlined by Juan et al. [16] and a more detailed account of assisted living technologies and functions have been outlined by Sun et al. [17]. Ensuring the safety and security of the user with the help of alarms, monitoring the health and well-being of the user, and the use of interactive and virtual services to help support the user are few of them. Table I enlists the functional requirements of an assisted living system. Here, we will elaborate our use case in assisted living for motivating our research work and briefly discuss the objectives and requirements to explain our choice of solutions and evaluation criteria.

A. Objectives

An important parameter that characterizes the quality of independent life is the safety of the users in their own homes. Ageing can affect all domains of life leading to physical infirmity and loss of mental or cognitive abilities necessitating safety monitoring applications. These applications use several sensors (e.g. body worn accelerometer and heart rate sensors). Activity recognition algorithms and fall detection algorithms are used to infer the mobility of the user. A context processing and management infrastructure ensures the provisioning of the context data. Poor-quality data may result in the system not being able to accurately detect the users' location which may lead to an inadequate provision of care and assistance.

Assisted living systems operate in real-time by processing events as they occur in the environment, and provide immediate support based on these detected events. Activity recognition and reasoning algorithms enrich these systems with contextual knowledge. Lack of QoC control may result in such a system providing assistance and support based upon incorrect data, information and knowledge inputs, and this may have an unfavorable effect on the users of the system (e.g. the elderly and their caregivers, health professionals, as well as relatives and friends). This adds to the complexity and suggests a need for quality control for the context data.

Based on the above mentioned shortcomings, we contrive the methodology for the well-informed decision making in assisted living as follows:

1) A traditional QoC *requirements gathering* to identify and model the required quality attributes at design time. It is domain specific and involves the type of context being utilized.
2) Runtime support to detect a change of quality in a particular context type and to measure its impact on other context types before making any decision.
3) Enforcement of QoC policies or ways to ensure the QoC requirements while making a decision.

As a case study we have modeled a smart notification assistant for monitoring elderly people. The objective of this use case is to realize an effective, automated recognition and alarm system for emergency situations in a smart home environment, and more particularly for selecting an appropriate caregiver. The choice of an appropriate caregiver depends upon the real-time context of the caregiver. The objective of the system is to make a smart choice for the caregiver based on certain QoC requirements. In our use case scenario we have modeled *Timeliness* and *Reliability* as the two main factors that will influence the choice of caregiver in an alarming situation. As such, the contextual uncertainty comes from the caregiver's end. His/her *locality* and *availability* are the major causes of runtime uncertainty in our use case.

B. Requirements

The major functional requirements of our smart notification assistant use case are:

- R1: The system can detect the physical state of the user immediately before and after the impact to assess whether assistance is needed in order to eliminate/minimize false alarms.
- R2: The system should detect the impact in real-time to facilitate immediate notifications to appropriate caregivers, and should detect the context of the caregiver, i.e. his *locality* and *availability*. It should take into account the runtime evidence to mitigate contextual uncertainty and take the most rational decision.
- R3: The system should select the caregiver with respect to the QoC requirements from the patient's perspective (such as T_{high} for *High Timeliness* and R_{max} for *Maximum Reliability*).

Suppose the caregiver's locality is unknown, and that his availability is known. Obviously, the confidence on *Timeliness* will be affected. We consider the *Reliability* of the system to be good if the system was able to notify an appropriate caregiver. As such, the context information of the caregiver influences the ability to address these QoC requirements. However, for the next decision, the system should take into consideration the real impact of its choices on *Reliability* and change it to maintain the *Maximum Reliability* (R_{max}).

- R4: The system should be able to update the future decisions if the QoC requirements were not met. The system should also be able to track any decisions taken in the past as well as their impact.
- R5: As some decisions might come with a cost (e.g. calling an emergency response team in case of a false alarm), the system should balance the costs with the benefits for any decision it makes with respect to the QoC requirements.

Based on the above mentioned requirements, our system should be able to take the following actions considering the options available at the moment:

- Action (A1): Address and verify all the QoC requirements before making any decision or taking any corresponding action.
- Action (A2): Take into account the current situation of the caregiver as well as that of the patient, choose the most appropriate caregiver and notify him/her.
- Action (A3): Learn from the previous decisions w.r.t. the uncertain context of the caregiver (i.e. *locality* and *availability*) and improve future choices accordingly.

IV. Quality-aware smart decisions under uncertainty

This section details the framework to process the QoC at each level of the context processing chain for our ubiquitous notification assistant. Additionally, it presents a probabilistic decision support model to tackle the runtime uncertainty. A brief account of our DDN model is also presented in this section.

A. *QoC-awareness in smart notification assistant*

The actions of the smart notification assistant are influenced by the uncertainty involved while processing multiple context types at runtime, in this case the *locality* and *availability* of the caregiver. As context data is extremely volatile, it requires certain QoC attributes (e.g. precision, accuracy, timeliness, trustworthiness) to achieve the context confidence defined as system's belief in the truth of the context. These QoC attributes may change over time in response to changes in the context. These attributes might be relevant in one context, but completely irrelevant in some other or might be weighed against each other, resulting in a trade-off between the attributes and their partial fulfillment. Further, quality attributes are difficult to quantify due to their fuzzy nature and the runtime uncertainty in the context data. Being QoC-aware demands a great deal of runtime support to ensure the end-to-end QoC, from the low-level context of the sensor to the high-level aggregated context that the application requires.

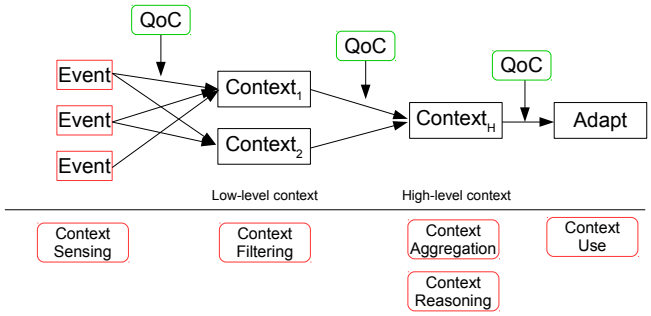

Fig. 1. End-to-end quality management in context processing

We see a strong need of quality-aware processing at each level of context processing as shown in Fig. 1. It depicts the significance of QoC requirements at each level of the context processing chain, from context fetching (low-level) to the context usage (high-level). The main data quality issues for our use case at the lower level can be the place and position of the sensor and the fact that the sensor may be accidentally altered resulting in wrong events. In case of a GPS sensor to infer the location of the user, the signals may be lost indoors. False events generated from these sensor's can pose a non-negligible impact on a user's (patient) health. Moreover, inferring context knowledge from this data will hamper the quality of the overall context data from lower level to the higher level of context processing and knowledge generation. A lack of quality at this level may harm the reasoning and actuation properties and reduce the ability of the ubiquitous notification assistant to determine the actual situation of the patient. It can have an unexpected effect on the real-time operations of the system as the derived context knowledge may be incorrect, consequently affecting the smart decisions of the system and preventing it from offering tailored interactions based on the user's (patient) current QoC requirements. Global attributes like *Timeliness* must be satisfied over the entire life time of the system. Different local attributes within a context might guarantee a global attribute. Moreover, a local attribute can guarantee that a global attribute is achieved and is not invalidated by the changes.

The QoC processing at the lower levels of the context processing chain – i.e. at the sensor data level – has been well researched [18]. To ensure QoC-awareness at each level of the context processing and provisioning chain, we propose a QoC processing framework to incorporate QoC-awareness in decisions of our smart notification assistant, shown in Fig. 2. The QoC processing engine takes the decisions at each level of the context processing taking into account the QoC requirements. At context gathering, it considers the QoC requirements as well as the event characteristics along with the event's own context. At the next level of context processing, it takes into account the QoC requirements and the low-level context characteristics. Finally, at the high-level context processing, it uses the QoC requirements and the context requirements in order to take a smart decision. In the subsequent section, we will discuss the QoC-aware decision making at context reasoning level for our notification assistant (highlighted with a dotted line in Fig.

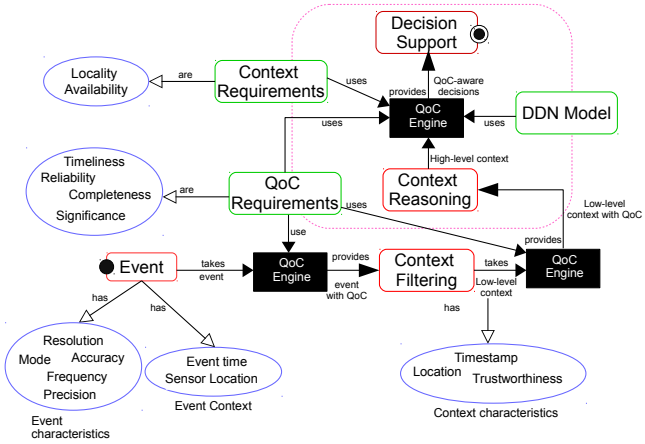

Fig. 2. End-to-end QoC model in context-aware decision support; dotted squared area highlights the QoC-aware decision support with high level context

2). We propose a model driven approach for smart decisions of our notification assistant under an uncertain context and its quality attributes. The proposed QoC framework not only makes the smart choices but it will also be well-informed about its previous decisions and their ability to fulfill the QoC requirements of the user.

B. DDN-based decision model conforming QoC requirements

Model-driven engineering and runtime models play a crucial role in tackling uncertainty in the data. A key issue in this approach is keeping the runtime models synchronized with the changing system. Uncertain attributes can be described using probability distributions derived by analyzing historical attribute values. These methods can take advantage of probability theory and statistics that helped solve stochastic problems in the past. Probabilistic reasoning systems use network models to reason with uncertainty. Probabilistic reasoning allows the system to reach rational decisions even when complete information is not available.

Knowledge about runtime uncertainty can be captured by a data structure for probabilistic inference called a Bayesian network (BN) and can be extended with temporal decisions, their utility and chance nodes to become a DDN. Bencomo and Belggoun [4] have advocated to use DDNs to deal with the runtime uncertainty in self-adaptive systems. DDNs can be used to model the decision support system that passively monitors and predicts the environment over time to take correct actions while considering any preferences. We present a mathematical model supported by DDNs as a solution to address the uncertainty in the context data and its quality while taking into account QoC requirements. A BN is a Directed Acyclic Graph (DAG) represented by a triplet (N, E, P), where N is the set of chance nodes, E is the set of arcs to represent causal influence of the chance nodes and P is the conditional probability distribution for each chance node.

A Decision Network is a BN that also includes a set of decision nodes and utility nodes. The utility nodes express the preferences among possible states of the world in terms of a subset of the chance nodes and the decision nodes. A probability-weighted expected utility is calculated for each decision given the evidence. To represent variables that change over time, it is possible to use a time-sliced network such that each time-slice corresponds to a time point. A DDN is used for the states, preferences and the decisions that change over time. Fig. 3 shows the structure of a general DDN. To process the QoC at each layer of context processing, its requirements can be modeled using a DDN that evolves over time where each time slice contains an action taken by the system. Utility functions can be used to assign priorities to different QoC requirements. The random variables associated with the chance nodes in a DDN can represent the QoC requirements for all the possible actions.

Our model express QoC requirements (QoC_i) and the context of the caregiver by chance nodes. These chance nodes make a Bayesian network with the conditional probabilities corresponding to the effects of different actions D_j over QoC_i expressed as $P(QoC_i| D_j)$. Evidence nodes, defined as "Obs" in Fig. 3, express the uncertainty factors connected to the chance nodes (QoC_i) to take a favorable decision. For each QoC_i, the utility nodes express the utility function that takes

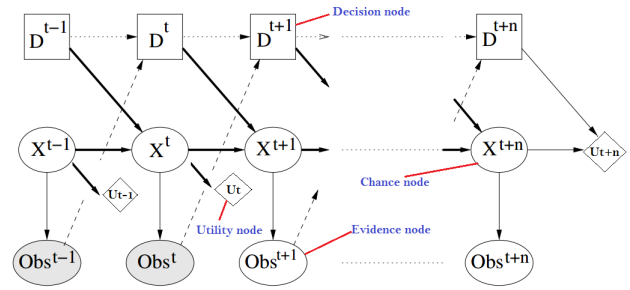

Fig. 3. Structure of a general Dynamic Decision Network [15]

the conditional probabilities of QoC requirements and their priorities into account. We can compute the expected utility for each decision taking into account the $P(QoC_i|D_j)$ and a weight for the decision. The DDN is evaluated using eq. (1) for every decision D_j to compute the probability-weighted average utility for that decision, also known as the expected utility [15].

$$EU(D_j \mid E) = P(QoC_i \mid E, D_j)U(QoC_i \mid D_j) \qquad (1)$$

The set of preferences for (QoC_i) is represented by $U(QoC_i \mid D_j)$, and the conditional probability for each QoC using Bayesian inference given the available evidence E is represented by $P(QoC_i|E,D_j)$. The action with the highest expected utility is chosen. Fig. 4 depicts our DDN model in two time slices. T_{high} is a dynamic node affected by the runtime context types *locality* and *availability*. R_{max} corresponds to the evidence influencing T_{high}. The transition model for dynamic nodes in each time slice needs to be represented by the conditional probabilities placed in a conditional probability table (CPT). Domain experts are required to fill in the initial values and DDN will update the subsequent values by learning from previous behaviour. The utility node takes all the nodes as parents that affect the outcome. Each type of node in our DDN model is described as follows:

a) Chance nodes *Locality* and *Availability* of the user represent the ever changing context (see R2 & R3 from Sect. III-B) of the caregiver.
b) The decision node *Choose Caregiver* (D) represents the action (see A2 from Sect. III-B) of choosing a caregiver to be called in time slice t. The possible decisions in our use case are; choose the caregiver whose current locality or availability is known, to notify him/her, or to not choose this caregiver.
c) The node T_{high} represents the QoC requirement (see R3 from Sect. III-B). It is a dynamic chance node, which means its probability distribution can be affected by the temporal dimension due to the context of the user (see R4 from Sect. III-B). The transition model $P(T_{high_{t+1}}|T_{high_t},D_t)$ is represented by a conditional probability table (CPT) for which the initial values are filled in by domain experts. The DDN will update the CPT in the subsequent time slices.
d) The node R_{max} represents the QoC requirement: Max reliability (see R3, R4 & R5 from Sect. III-B). It is a static chance node and behaves as an observable affecting another QoC attribute: *Timeliness*. Domain experts should fill in the initial values of its CPT. The

$$EU(D_{t+1} \mid R_{max_{1:t}}) = \sum_{T_{high_{t+1}}} u(T_{high_{t+1}} \mid D_{1:t}).P(T_{high_{t+1}} \mid R_{max_{1:t}}, C_{t+1}, D_{1:t}) \quad (2)$$

$$P(T_{high_{t+1}} \mid R_{max_{1:t}}, C_{t+1}, D_{1:t}) = \sum_{T_{high_t}} P(T_{high_{t+1}} \mid T_{high_t}, C_{t+1}, D_t).P(T_{high_t} \mid R_{max_{1:t}}, C_t, D_{1:t-1}) \quad (3)$$

$$P(T_{high_t} \mid R_{max_{1:t}}, C_t, D_{1:t-1}) = \alpha P(R_{max_t} \mid T_{high_t}).tm \quad (4)$$

$$tm = \sum_{Thigh_{t-1}} P(T_{high_t} \mid T_{high_{t-1}}, C_t, D_{t-1}).P(T_{high_{t-1}} \mid R_{max_{1:t-1}}, C_{t-1}, D_{1:t-2}) \quad (5)$$

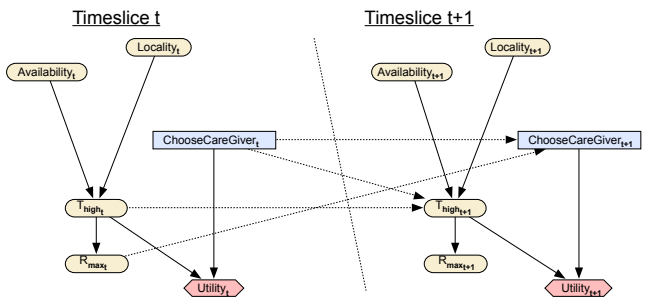

Fig. 4. DDN model for QoC-aware notification assistant

CPT does not change over time, and its conditional probability is represented by P(R_{max_t}| T_{high_t})

e) The utility node *Utility$_t$* represents the utility function to be used to compute the utility of the action. A utility node has preferences (see A3 & A4 from Sect. III-B) modeled as chance nodes, and decision nodes influencing these preferences as its parents fulfilling R5 from Sect. III-B.

V. Evaluating the DDN to make decisions maintaining QoC requirements

After setting up the structure of the DDN model with respect to the system's requirements and objectives, we have carried out our experiments for decision making using the Netica development environment (http://www.norsys.com) [19]. To formulate the inference computation, let us denote the set of decisions and the evidences over the time t as $D_{1:t}$ and $E_{1:t}$, respectively. The context of the caregiver is denoted as C_t. The inference computation that must be solved in order to make a decision with the DDN is given in eq. (2), where $EU(D_{t+1}|R_{max_{1:t}})$ is the expected utility of the decision D_{t+1} given the evidence $R_{max_{1:t}}$.

The Markov property [15] can be used to compute the probability of $P(T_{high_{t+1}}|R_{max_{1:t}}, C_{t+1}, D_{1:t})$ as given in eq. (3). Next, the value of $P(T_{high_t}|R_{max_{1:t}}, C_t, D_{1:t-1})$ in eq. (3) can be computed as in eq. (4). In eq. (4) $P(R_{max_t}|T_{high_t})$ is the observation model, α is the constant to ensure the probabilities sum up to one and *tm* is the transition model. The optimal decision suggested by the DDN at time slice *t+1* is the action that maximizes the expected utility [15] and is expressed as follows:

$$arg\ max_j[EU_{d_j}(D_{t+1}|R_{max_{1:t}})] \quad (6)$$

We have performed experiments to see how our DDN model makes decisions under a continuously evolving context of the caregiver.

A. Decision making under an evolving context

In our first experiment, we have examined the role of DDNs to trigger the correct actions needed by the notification assistant. We observed the capability of our model to learn from previous decisions. In order to evaluate the DDN model shown in Fig.4, we have considered the following initial conditional probabilities for T_{high} defined by the domain experts [20].

P(Timeliness = high | Locality, Availability) = 0.60
P(Timeliness = high | Availability) = 0.20
P(Timeliness = high | Locality) = 0.08

Conditional probabilities for the temporal aspects of T_{high} for being high are given as follows:

P($T_{high_{t+1}}$ = high | T_{high_t} = low) = 0.20
P($T_{high_{t+1}}$ = high | T_{high_t} = medium) = 0.30
P($T_{high_{t+1}}$ = high | T_{high_t} = high) = 0.80

The utility values to be used for the calculation of the expected utilities are shown in Table II. Each row represents the utility value associated with a decision (caregiver chosen) and its effects on the chance node (i.e., QoC requirements). Given a row and its decision, a value *low* for a chance node states that the decision has a negative effect on the QoC requirement represented by the chance node. A value *medium* states that the decision has a mild effect on the QoC requirement, and a value *high* states that the decision has a mild effect on the QoC requirement. For example, the first row of Table II states that the decision *ChooseCareGiver* has a negative effect on *Timeliness* (T_{high}), hence the utility is 5. The range for the utility values in Table II is [0 ... 100]. Fig. 5 shows the results of the computation of the expected utility (EU) for each decision chosen during 5 time slices t=0 to t=4. In this experiment the *availability* and the *locality* of the caregiver are known. The DDN suggests *ChooseCareGiver* in

Decision	Timeliness	Utility 1	Utility 2
ChooseCaregiver	low	5	90
ChooseCareGiver	medium	60	50
ChooseCareGiver	high	90	90
DoNotChooseCG	low	80	80
DoNotChooseCG	medium	15	15
DoNotChooseCG	high	80	15

TABLE II. Utility table for the QoC requirement Timeliness

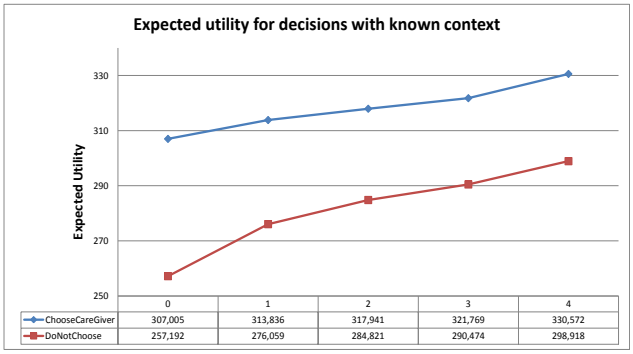

Fig. 5. Expected utility of choose caregiver is higher when context is known and the learning capability is increasing in subsequent time slices

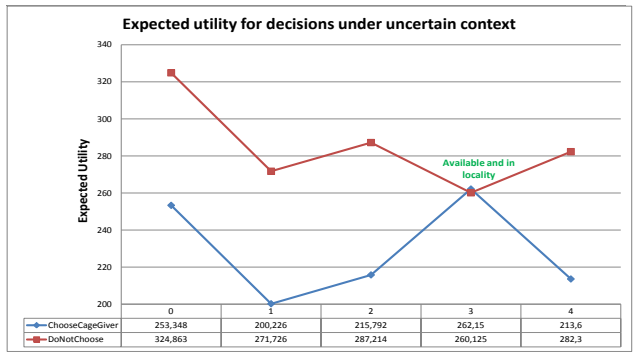

Fig. 6. Expected utility of decisions under uncertain context where the system can adapt to the change in context

each time slice as its expected utility is higher and the learning capability increases in each subsequent time slice. We did not observe the *Reliability* of the system so the QoC requirement *Timeliness* is set on the basis of the conditional probabilities.

We conducted another experiment to see the behavior of our model in the presence of uncertainty and partial observability of context i.e. the *availability* and the *locality* are unknown in most of the time slices. Fig. 6 shows the results of the computation of the expected utility (EU) for each decision chosen during time slices t=0 to t=5. In this experiment our DDN suggests *DoNotChoose* this caregiver when context is unknown to maintain the QoC. But as soon as the context of the caregiver changes at t=3, our DDN model computes *EU(ChooseCareGiver)* > *EU(DoNotChoose)*. Fig.7 shows the results of another experiment where we observed the role of the observational model and the behavior of our DDN to trigger correct decisions in each time slice. We evaluated how evidences about the R_{max} being min, average or max can trigger the need of a runtime change for the next decision. Given the initial conditional probabilities and the utilities provided by experts at time slice 0, the DDN suggests as expected that the best decision is *ChooseCareGiver* when its context is known. In terms of a DDN, this means that with no evidence about the reliability being min, the apparent suitable decision is to use *ChooseCareGiver* as the expected utility *EU(ChooseCareGiver)* > *EU(DoNotChoose)*.

We observed that R_{max}=*average*. This choice is based on the observation of the runtime change in reliability. In the next time slice t=1, our DDN model computed *EU(ChooseCareGiver)* slightly higher than *EU(DoNotChoose)*. At t=3 we again observed R_{max}=*min*, and our DDN model changed the decision and computed *EU(DoNotChoose)* > *EU(ChooseCareGiver)* in the next time slice t=4 even though the context of the caregiver was known. Certainly, the decision *DoNotChoose* is considered by the DDN as the best decision at this time. It is the most suitable as the *ChooseCareGiver* action does not necessarily fulfill the QoC requirements for *Timeliness* (T_{high}) and *Reliability* (R_{max}). Following the scenario, later on, at time slice 4, the monitoring infrastructure finds the *Reliability* is max again, and the DDN correctly suggests to take the choice *ChooseCareGiver* on the basis of the available context.

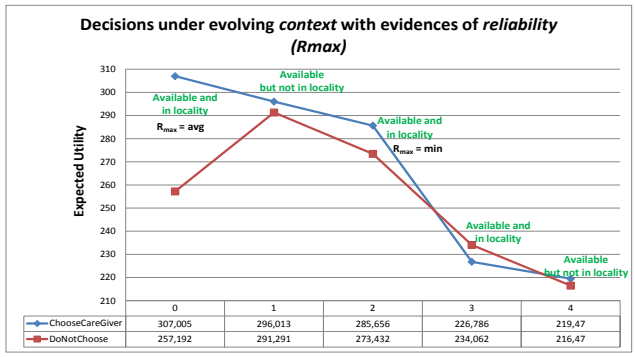

Fig. 7. The role of the observational model and decision under evolving context with evidences of reliability

B. Impact of the utility on the decision making

Finally, we did a sensitivity analysis to examine the effects of the weights described in the utility table II while deriving the best action. We have evaluated the DDN's sensitivity to these weights on our notification assistant scenario using the same scenario presented earlier to compute the best decision during time slice 1 until time slice 5. However, different from the previous experiment where we kept the values of the utility weights constant (using just the column Utility 1), in this experiment we have assumed that the weights assigned to QoC requirements can be changed on-the-fly at runtime (using values from both columns Utility 1 and 2 in Table II). The different set of weights were used at time slice 3 using the values dictated by the column Utility 2. The DDN started running with the same initial configuration. The results of this experiment are shown in Fig. 8.

The DDN adapts accordingly when new information becomes available at time t=2, and again as expected, the DDN selects *ChooseCareGiver* (as this action has greater value of EU than the *DoNotChoose*). However, the value min of the reliability is monitored at time slice 4, it can be observed that *EU(ChooseCareGiver)*> *EU(DoNotChoose)*. This effect is due to the newly higher weight associated with the action *DoNotChoose* at time slice 4. As observed, the values of the weights to calculate the expected utilities of decisions can have an important impact on the evaluation of alternative decisions. In this experiment, we have discussed the sensitivity analysis as an important step to check how sentient the decision-maker is to changes on the values of utilities by systemat-

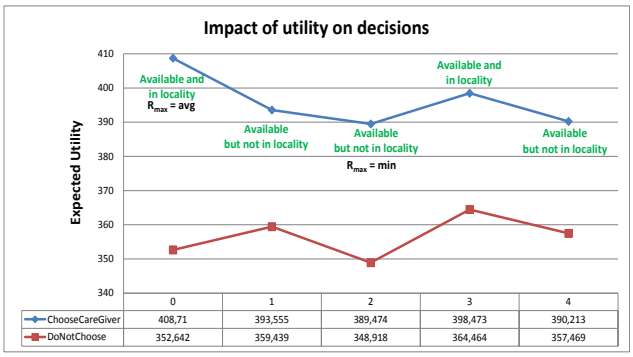

Fig. 8. Impact of utility on decision

ically varying those values when running the DDN. One of the main criticisms of DDN-based approaches is that of the effort needed during the assessment of the numerical weights required. However, often those values can be successfully specified by expert knowledge elicitation.

VI. Conclusion

In this paper we have presented a novel approach for decision support in assisted living by leveraging Dynamic Decision Networks (DDN) to automate decisions in an uncertain context. DDNs build upon Dynamic Bayesian Networks. However, the latter is only able to learn conditional probabilities based on a dataset, whereas DDNs can quantify the impact of the evidence and the effect of the decisions. Furthermore, by exploiting the utility of these decisions our framework can learn how to automatically improve its decisions in the next iteration or time slice. Our first contribution was a feasibility analysis of incorporating DDNs for decision support in an assisted living scenario, and our experiments have clearly demonstrated the ability of adapting its decision in the presence of evolving situations and an uncertain context of the caregiver.

Our second contribution relates to the quality of the context upon which these decisions are based. We extend existing QoC frameworks and modeling paradigms by processing the quality attributes in an end-to-end fashion, rather than considering QoC only at the source of the information (e.g. the sensors). By incorporating QoC in our DDNs, we are able to assess the quality of our context-driven decisions, ascertain their quality and update future decisions and corresponding actions according to the outcome and impact. Suitable decisions that fulfill the QoC requirements are selected from a range of alternative decisions and their expected utilities. Our experiments have shown that our approach is applicable to use cases in the area of assisted living, and the results achieved so far are promising. However, as the sensitivity analysis has shown, one must pay attention to the fact that the quantitative outcome of the experiments themselves are subject to the initially chosen prior and conditional probabilities. These either have to be collected from a data set or be provided by domain experts.

Further work is required towards more systematic techniques for studying the value of the probabilities that change over time and their impact on alternative decisions. Finally, developing tools to specify the QoC requirements and design a DDN would be certainly very helpful as the current tool's support for modeling and using DDNs is fairly limited.

Acknowledgment

This research is partially funded by the Inter university Attraction Poles Programme Belgian State, Belgian Science Policy, and by the Research Fund KU Leuven.

References

[1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Handheld and ubiquitous computing*. Springer, 1999, pp. 304–307.

[2] T. Buchholz, A. Küpper, and M. Schiffers, "Quality of context: What it is and why we need it," in *Proceedings of the workshop of the HP OpenView University Association*, vol. 2003. Geneva, Switzerland, 2003.

[3] N. Chen and A. Chen, "Integrating context-aware computing in decision support system," in *Proceedings of the International MultiConference of Engineers and computer Scientists*, vol. 1, 2010.

[4] N. Bencomo and A. Belaggoun, "Supporting decision-making for self-adaptive systems: from goal models to dynamic decision networks," in *Requirements Engineering: Foundation for Software Quality*. Springer, 2013, pp. 221–236.

[5] M. Krause and I. Hochstatter, "Challenges in modelling and using quality of context (qoc)," in *Mobility Aware Technologies and Applications*. Springer, 2005, pp. 324–333.

[6] P. Castro, P. Chiu, T. Kremenek, and R. Muntz, "A probabilistic room location service for wireless networked environments," in *Ubicomp 2001: Ubiquitous Computing*. Springer, 2001, pp. 18–34.

[7] A. Ranganathan, J. Al-Muhtadi, and R. H. Campbell, "Reasoning about uncertain contexts in pervasive computing environments," *Pervasive Computing, IEEE*, vol. 3, no. 2, pp. 62–70, 2004.

[8] A. Manzoor, H.-L. Truong, and S. Dustdar, "On the evaluation of quality of context," in *Smart Sensing and Context*. Springer, 2008, pp. 140–153.

[9] K. Sheikh, M. Wegdam, and M. v. Sinderen, "Quality-of-context and its use for protecting privacy in context aware systems," *Journal of Software*, vol. 3, no. 3, pp. 83–93, 2008.

[10] Y. Kim and K. Lee, "A quality measurement method of context information in ubiquitous environments," in *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, vol. 2. IEEE, 2006, pp. 576–581.

[11] J. Filho, A. Miron, I. Satoh, J. Gensel, and H. Martin, "Modeling and measuring quality of context information in pervasive environments," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, 2010, pp. 690–697.

[12] N. Fenton and M. Neil, "Making decisions: using bayesian nets and mcda," *Knowledge-Based Systems*, vol. 14, no. 7, pp. 307–325, 2001.

[13] N. Esfahani, K. Razavi, and S. Malek, "Dealing with uncertainty in early software architecture," in *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*. ACM, 2012, p. 21.

[14] A. Filieri, C. Ghezzi, and G. Tamburrelli, "A formal approach to adaptive software: continuous assurance of non-functional requirements," *Formal Aspects of Computing*, vol. 24, no. 2, pp. 163–186, 2012.

[15] S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards, *Artificial intelligence: a modern approach*. Prentice hall Englewood Cliffs, 1995, vol. 74.

[16] J. C. Augusto, H. Nakashima, and H. Aghajan, "Ambient intelligence and smart environments: A state of the art," in *Handbook of Ambient Intelligence and Smart Environments*. Springer, 2010, pp. 3–31.

[17] H. Sun, V. De Florio, N. Gui, and C. Blondia, "Promises and challenges of ambient assisted living systems," in *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*. IEEE, 2009, pp. 1201–1207.

[18] P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini, "A survey of context data distribution for mobile ubiquitous systems," *ACM Computing Surveys (CSUR)*, vol. 44, no. 4, p. 24, 2012.

[19] N. S. Corporation, *Netica Users Guide*, 1997.

[20] M. Amoretti, S. Copelli, F. Wientapper, F. Furfari, S. Lenzi, and S. Chessa, "Sensor data fusion for activity monitoring in the persona ambient assisted living project," *Journal of Ambient Intelligence and Humanized Computing*, vol. 4, no. 1, pp. 67–84, 2013.