CoMiHoC : A Middleware Framework for Context Management in MANET Environment

Waskitho Wibisono
Caulfield School of IT
Monash University
Melbourne, VIC, Australia
waskitho.wibisono@infotech.monash.edu.au

Arkady Zaslavsky
Dept of Computer Science and Electrical Engineering
Lulea University of Technology
Lulea, Sweden
arkady.zaslavsky@ltu.se

Sea Ling
Caulfield School of IT
Monash University
Melbourne, VIC, Australia
chris.ling@infotech.monash.edu.au

***Abstract*—Advances in wireless and sensing technology and the proliferation of context-aware systems have highlighted the requirement to reduce the complexity of system development by establishing a middleware framework of context management. The middleware aims to manage different aspects of context and location management as well as situation reasoning and other related tasks. In this paper, a middleware framework of context management and situation reasoning in MANET environment is proposed. It includes components to model and reason about contexts and situations, to construct location models and estimate the relevance of contexts used for situation reasoning. We also incorporate On-Demand Multicast Routing Protocol (ODMRP) as the underlying messaging protocol among the collaborative hosts to deal with MANET communication issues. We discuss implementation and experimentation issues of the framework in personal digital assistance (PDA) devices which form the ad hoc network as proof-of-concept of the framework. Our preliminary implementation and experimentation shows the capability of the framework to work in such limited devices and to perform the designed capabilities of the middleware.**

Keywords: MANET, Context-Aware System, Situation Reasoning, Distributed Applications, ODMRP

I. INTRODUCTION

Advances in wireless communication, sensing technology and mobile devices have led to the emergence of *context-aware systems* capable of adapting to situation changes in the environment without explicit user intervention [1][3]. The proliferation of the systems has stimulated research to reduce the complexity in their development by providing generic support of context management in the form of a *context middleware* framework [8]. However, providing such a service framework in a mobile ad hoc network environment (MANET) is challenging due to the environmental characteristics and limitations.

MANET is a collection of mobile hosts which form a dynamic network without any fixed infrastructure in which only nodes within a certain network range can communicate directly with each other [10][19]. It is also characterized by lack of central authority and is very dynamic in terms of the availability of network resources, communication partners, limited bandwidth and connectivity [10][21].

Generally, context-aware applications that work in the MANET environment have to establish collaborations in a peer-to-peer manner. However, frequent disconnection may cause difficulty in the establishment of the necessary coordination since the latest context information is often difficult to obtain. The availability and the quality of context are also dynamic, while similar contexts can be offered by multiple hosts, raising problems of context ambiguity that need to be dealt with. Context also has temporal and spatial characteristics due to the mobility of both context and its users where a context can be created by one host and be re-distributed by the other hosts in the environment. These problems relate to the *quality* and *uncertainty* of context. We have identified several challenges which need to be addressed in managing context in MANET and establishing a context middleware framework. They are as follows:

- **Temporal relevancy**: Context may be made obsolete by mobility of hosts and can be closely related to physical location and may become less relevant or even not applicable in certain locations.
- **Context uncertainty**: Context has temporal relevancy due to the mobility of users while the necessary context may not be available, insufficient or even contradictory. These problems can lead to related problems of context uncertainty
- **Distributed control**: Hosts in an ad hoc network are categorized as equal and data is managed distributively. Furthermore, frequent disconnection from the origin context sources makes it difficult to maintain the up-to-dateness of the context.
- **Fault-tolerant**: Frequent disconnection that may occur in the MANET environment requires a fault tolerant approach to overcoming the problems that may arise.

Given the characteristics and limitations of MANET, a new approach of context management is required to overcome the limitations of the environment and to provide reliable support for the applications. In this paper we propose a middleware framework that aims to facilitate collaborative context management among mobile hosts in a MANET environment..

1550-445X/10 $26.00 © 2010 IEEE
DOI 10.1109/AINA.2010.153

The proposed middleware includes components for modeling and reasoning about situations based on the Context Mobile Spaces (CoMoS) as the basis of context model and situation reasoning proposed model. A technique to compute the relevancy value of context information to a context-aware application has been proposed and developed. We have also developed a *hybrid location model* [2] and integrated the approach to the model. Furthermore, the framework adopted On-Demand Multicast Routing Protocol (ODMRP) [7][19][21] as the underlying messaging protocol among the collaborative hosts to distribute contexts. We also discuss the implementation and experimentation issues of the framework over the ad hoc network of personal digital assistance (PDA) devices as a proof-of-concept of the framework.

II. BACKGROUND

A. Context Spaces

The Context Spaces (CS) theory [13][14] defines situation in pervasive environment as a collection of accepted regions in a multidimensional space. It also defines *context attribute* as any type of data used in the process of situation reasoning that can be associated with sensor data and denoted as c_i^t [13]. The CS theory applies a geometrical concept that describes context and situations from the *state-space* model.

The CS defines a real situation in context-aware environment as a situation space. Formally, a situation space j, denoted as $s_j = \{C_1^j, C_2^j, .., C_n^j\}$ where a region C_i^j is a set of element X that satisfies a predicate P, i.e., $C_i^j = \{X \mid P(X)\}$.

B. Data Fusion for Situation Reasoning

We propose *CoMoS* (Context Mobile Spaces) as the basis of context model and situation reasoning. The proposed concept adopts and extends the concept of *situation spaces* in the Context Spaces Theory [13][14] and integrates the model with the Dempster-Shafer (DS) [17] theory of evidence a data fusion technique for reasoning about situation. The *Discount rule* [16][17] is applied to incorporate the uncertainty factor that may exist in the process of situation reasoning.

In the DS theory, a *frame of discernment* is defined as a set of possible hypotheses under consideration denoted as Θ. A basic probability assignment or mass (m) is defined as function $m: 2^{\Theta} \rightarrow [0,1]$ where $m(\varnothing) = 0$ and $\sum_{A \subseteq 2^{\Theta}} m(A) = 1$. The mass value of a given set A, $m(A)$ denotes the strength of relevant evidences that support the proposition of A where $m(\Theta)$ denotes mass value of all possible situations and is used to represent ignorance or uncertainty of situation given the available evidences. The DS theory can be applied to the CS model by computing mass distribution for each acceptable region in the defined situation space, i.e. when the value of a context attribute c_i is within a corresponding region in the defined situation space [13]. The DS rule of combination is then applied to combine relevant contexts as the valid evidences to the defined situation spaces. If we have two relevant evidences from different information sources, the combined belief function for each possible hypothesis A is defined as follows [17][20]

$$Bel_{1,2}(A) = \frac{\sum_{X_i \cap Y_j = A} m_1(X_i) m_2(Y_j)}{1 - K_{1,2}} \qquad (1)$$

The notion of *conflict* between evidences denoted as K, which is obtained from sources 1 and 2, is computed using the following equation:

$$K_{1,2} = \sum_{X_i \cap Y_j = \varnothing} m_1(X_i) m_2(Y_j) \qquad (2)$$

However, the use of the DS rule of combination directly to combine available evidences is based on an optimistic assumption that the available evidences are equally relevant or reliable. In fact, context has temporal and spatial relevancy value due to its characteristics and mobility. We apply the *discount rule* [16][17] to adjust the corresponding mass by incorporating the context's *relevancy* value before using it for reasoning about situation at the corresponding location. We discuss our approach to computation of the context relevance value in Section III.

C. Location Modelling

Common queries regarding information processing in pervasive environment are based on location [2]. Hence location information plays a vital role in context-aware systems. Location information also becomes an important requirement to validate contexts and to reason about situations in particular locations that may affect the system's users.

In this paper, we adopt the concept of a *symbolic location* model [2] to represent location information. It defines a position in an abstract symbol such as a building, room, lift, park, road, parking area, road or parking area as a set of *location containers*. Details of the model are provided in the following subsections.

1) Hybrid-Location Model

Let L be the set of all location containers. We define a location model $l \subseteq L$ in the *hierarchical-based* location model [11] as shown in Figure 1. The figure illustrates the hierarchical concept of the location model in which each of the location containers is decomposed into a different level of location containers that represents the containment relationship among them. For example, building H consists of several floors in which each floor may have several corridors that contain various types of *location object* $l_i \in l$ as the lowest level of the hierarchical location model of the building H.

Connections among the location object are denoted as a connected graph $G = (l, p)$ which represents spatial connections among the location object paths. p can be added if there is a direct connection between the location objects with distance d as its weight.

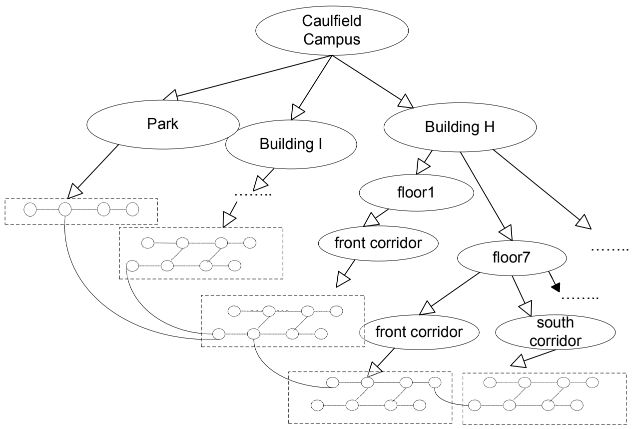

Figure 1. Hybrid Location Model

2) Location Identifier

We use a hierarchical format to represent an address of a location called *location identifier*. Given the floor layout shown in Figure 2, the following examples depict the location format:

- H/7/south/R.00 represents a room (R.00), located in the southern corridor on the 7th floor of building H.
- H/7/front/E.S1 represents an exit point (E.S1), located in the front corridor on the 7th floor of building H
- Park/meeting point1 represents an outdoor location node i.e. a meeting point in a campus park.

Classes of the location model and API methods have been developed based on the hybrid location model as described in the previous sub-section. These include classes that represent location objects (e.g. building, exit, floor, room, corridor, room, open space) in the model and methods to build the model and construct connections among them. In order to establish a complete location system, we need to combine the location model with a positioning system.

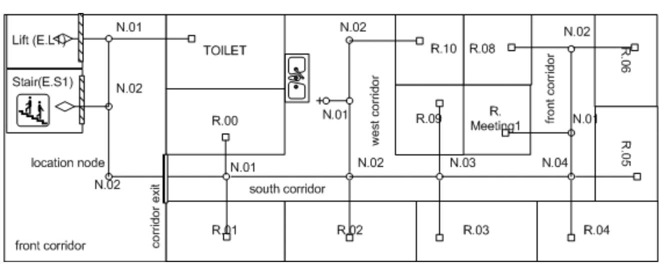

Figure 2. Layout of a 7th floor in the building H

A GPS system can be used to identify the coordinate positions of the object in an outdoor environment. For indoor locations, sensor arrays consisting of RFIDs, Bluetooth or infra-red beacons can be used as the positioning system to determine the location of a mobile object.

III. Aproximating Relevancy of Context

We propose a notion of *context relevancy* that is defined as the degree to which particular context information is applicable to the current situation of a context-aware application. The value is estimated based on context quality attributes, the application's preference and the current situation of the corresponding context-aware application as discussed in the following subsections.

A. Context Element

We define *context element* as any type of data that is used in the process of reasoning of the relevant situations. It is denoted as a triple $ce = (c_i, Q, l_k)$ where c_i is a *context attribute*, l_k is the original location of context creation and $Q = \{q_1, q_2, .., q_n\}$ represents a set of context quality attributes. The *quality of context* is described as any inherent information that can be used to determine the worth of information to the applications [9]. The characteristics of the sensing device, situation of measurement and transformation process are among the factors that influence the quality of the created context information [9].

B. Temporal Validity of Context

Temporal validity of a context $\tilde{v}^t_{ce}$ denotes the degree to which the context element ce is still relevant for use at a certain time *t*. The value is calculated by comparing the context age a_c ac with its maximum context age $\hat{a}_c$ of the corresponding context element ce. There are several ways to determine the maximum age of context information. For a low-level context such as sensor data, a common mechanism to determine the maximum age of the context is by using the refresh rate of the sensor such as room temperature which is collected every hour.

For a high-level context such as an inferred situation in a certain area, the maximum age of context information often cannot be predicted exactly, since the length of the situation depends on changes of other context information based on the situation in the real world. For example, the safety status of a floor is related to fire or the possibility of gas leakage. The maximum age of this context information can be assigned as *'undecided'*. This means that this particular context is temporarily valid until updated by the associated system. A similar condition applies to low-level context information if the source cannot specify its refresh rate. We calculate the temporal validity of context information by using the following equation:

$$\tilde{v}^t_{ce} = \begin{cases} 1 - \dfrac{a_c}{\hat{a}_c} & , \textit{ if } a_c < \hat{a}_c \\ 1 & , \text{ if } \hat{a}_c = \text{u}\textit{ndecided} \\ 0 & , \textit{otherwise} \end{cases} \qquad (3)$$

C. Spatial Coverage of Context

Spatial coverage of a context $\tilde{s}_{ce}$ is defined as the geographical scope in which the context information ce will have its impact [18]. For example, the safety status of a room on a floor may impact only on the nearby corridors, while the situation in a corridor may impact on the entire floor. For a particular location model l, the context spatial coverage is defined as a set of tuples

$$\tilde{s}_{ce} = \{(l_1,(\varphi(ce,l_1)), l_2,(\varphi(ce,l_2)),..., l_1,(\varphi(ce,l_n))\}$$

where $l_i \in l$ and $\varphi(ce,l_i) \in [0,1]$ represents the degree that the context information has influence to be used in the corresponding location. For example, the safety status of a room on a floor may impact only on the nearby corridors, while the situation in a corridor may impact on the entire floor. However, determining the context *spatial coverage relevancy* for a particular location i.e. $\varphi(ce,l_i)$ is a domain specific that requires expert knowledge of the associated domain and is currently beyond the scope of this paper.

D. Probability of Correctness

The probability of correctness (PoC) of context element ce, denoted as $\tilde{p}_{ce}$ is defined as the correctness probability of the provided context information. For a *low-level* context (e.g. sensor data), this value is commonly related to the accuracy of the corresponding sensor. It refers to the ratio of correctly detected events to the total number of events. To obtain the value, physical observations need to be conducted to obtain this value by comparing sensor observations with reference knowledge. For example, if a sensor is known or statistically computed [6] to be accurate with 5% error, we can then state that the accuracy of the sensor reading for the corresponding context it may produce is 0.95. Accordingly, probability of correctness towards contexts that the sensor may produce should be discounted respectively.

For high-level contexts such as situations in a certain location, data transformation and aggregation, reasoning processes as well as the accuracy of the sensor used create uncertainty of the inferred context. The confidence value of the inferred context denotes the probability of correctness of the corresponding high-level context information.

E. Estimating Context Relevancy

In general, the relevance of context to the user's situations will decrease as its quality decreases. Given the three previous examples of context quality attributes, a set of weights (δ, γ, λ) is assigned. The weights represent the relative importance of an attribute to the other attributes to infer the situations under consideration.

Suppose we define a set of possible situations $\Theta^{l_k} = \{s_1, s_2,.., s_n\}$ for the location $l_k \in l$, the relevance of a context element ce for the considered situations is computed using the following equation:

$$r_{ce}(\Theta^{l_k}) = \frac{\delta.\tilde{v}_{ce} + \gamma.\varphi(ce,l_k) + \lambda.\tilde{p}_{ce}}{\delta + \gamma + \lambda} \tag{4}$$

where $\delta, \gamma, \lambda >= 0$, and t represent time of context usage.

F. Discount Rule

Suppose we have n numbers of possible situations that may occur at a location l_k denoted as $\Theta^{l_k} = \{s_1, s_2,.., s_n\}$ and the relevancy of the context information for inferring the considered situations is denoted as $r_{ce}(\Theta^{l_k})$, then by adopting the discount rule [16][17], the corresponding discount mass m_{ce} for each situation $s_i \in \Theta^{l_k}$ given a context information ce is computed as follows

$$m_{ce}^{disc}(s_j) = r_{ce}(\Theta^{l_k}) m_{ce}(s_j) \tag{5}$$

$$m_{ce}^{disc}(\Theta) = 1 - r_{ce}(\Theta^{l_k}) + r_{ce}(\Theta^{l_k}) m_{ce}(\Theta) \tag{6}$$

. The discounted masses become the new evidences which then are combined to infer belief of situation occurrence using the DS-rule of Combination (Eq.1, Eq.2). The lower the relevance value of evidence, the greater the mass for $m(\Theta)$ that corresponds to the degree of ignorance or uncertainty of the inferred situation given the evidence. A heuristic approach to approximation of the relevancy of context is discussed in the following section.

IV. CONTEXT MOBILE SPACES (COMOS)

We propose *CoMoS* (Context Mobile Spaces) as the basis of context model and situation reasoning which adopts and extends the concept of *situation spaces* in the Context Spaces Theory [13][14] and integrates the model with the Dempster-Shafer (DS) [17] theory of evidence as described in the following concepts:

Relevant Situations (Definition 1): The relevant situations are possible situations (which are sensed/considered) for a context-aware application. We denote the relevant situations that may occur at a location $l_k \in l$ as $\Theta^{l_k} = \{s_1, s_2,.., s_n\}$ which are enumerated as the *frame or discernment* [17] in the DS model.

Region space (Definition 2): A *region space* is defined as a set of basic probability assignments (BPA) for each possible situation hypothesis of an acceptable region C_j^i. The acceptable region C_j^i is defined as a set of context attributes c_i of a context source i that satisfies a predicate P_j, i.e. $C_j^i = \{c_i \mid P_j(c_i)\}$. The *region space* is denoted as $R_j^i = \{m_i(A_1), m_i(A_2),.., m_i(A_k)\}$. The mass value of a given set A, $m(A)$, denotes the strength of all relevant evidences that support the proposition of A.

Evidence Space (Definition 3): The *evidence space* denotes mass assignments for a context attribute c_i for all of its assigned predicates $P = \{P_1, P_2, .., P_k\}$. The evidence space is denoted as $E^i = \{R_1^i, R_2^i, .., R_j^i\}$ that represents the complete mass assignments for all of the defined *region spaces* R_j^i for the context attribute c_i.

Situation Space (Definition 4): The situation space represents the application's knowledge about environment situations under its consideration for a particular location l_k. It consists of n number of *evidence spaces* and is denoted as $S^{l_k} = \{E^1, E^2, .., E^n\}$ that represents knowledge about the situations being reasoned from n number of context attributes c_i.

Imagine a scenario where a context-aware application needs to adapt to j number of possible situations that may occur at a location $l_k \in l$. After preliminary observation, a safety expert determines n number of possible situations that may occur at the location ($\Theta^{l_k} = \{s_1, s_2, .., s_n\}$) such as *safe (S)*, *risky (R)* and *dangerous (D)* where each of these situations corresponds to different warnings or possible situation escalations.

Suppose that four different *context attributes* (i.e. *Temperature, Smoke Present, Gas Leak*, and *Physical Damage*) are used as evidences to define the situation spaces. For each evidence, a set of predicates that form *region spaces* are composed along with the corresponding basic probability assignments and form *evidence spaces* for the corresponding context attribute.

An example of this concept is shown in Table 1 which illustrates the BPA assignment for each *relevant situation*, i.e. *Safe (S)*, *Risky (R)* and *Dangerous (D)*. For each evidence, basic probability assignments of each region space are assigned for each relevant situation, i.e. m(S), m(R) and m(D). All mass assignments for all of the region spaces then form the *evidence spaces* for the corresponding context attribute. For example, for a region space (*SmokeLevel=LIGHT)*, masses for the situation's occurrence given the region of acceptable value are assigned as follows: Safe (0.1), Risky (0.35), Dangerous (0.45) or Θ/Unknown (0.1) respectively, where m(Θ) is used to represent ignorance or uncertainty of situation given the corresponding evidence.

Assigning these mass values is a crucial task since it may need to involve expert opinion since they represent knowledge about the situations under consideration as well as the uncertainty incorporated in the captured evidences. In this paper we obtain general senses of the BPA using expert knowledge rather than an exact resolution which may be harder to achieve and is beyond the scope of this paper. The complete mass assignments for all situation spaces in the scenario are shown in the following table.

TABLE I. SITUATION SPACES AND MASSES ASSIGNMENT

	Evidence Space	Region Space	m(S)	m(R)	m(D)	m(Θ)
SITUATION SPACES	Temperature	Cold	0.3	0.4	0.2	0.1
		Normal	0.45	0.3	0.15	0.1
		Hot	0.05	0.2	0.7	0.05
	Smoke Level	No	0.5	0.2	0.2	0.1
		Light	0.1	0.35	0.45	0.1
		Heavy	0.05	0.25	0.6	0.1
	Gas Leak	No	0.4	0.3	0.2	0.1
		Yes	0.05	0.2	0.7	0.05
	Physical Damage	No	0.4	0.25	0.2	0.15
		Light	0.2	0.4	0.3	0.1
		Heavy	0.2	0.3	0.4	0.1

V. COMIHOC MIDDLEWARE FRAMEWORK

In this section we discuss the proposed middleware framework and describe its constituent components. The preliminary middleware framework consists of several components that run on all hosts willing to cooperate in a MANET environment as a collaborative host.

A. *CoMiHoC Architecture*

The CoMiHoC (COntext MIddleware for adHoC network) framework is developed to support programmers in addressing a broad range of issues in the development of context-aware applications including location models, situation reasoning and modeling and context validation as well as integration with the underlying communication protocol.

The framework is grouped into three different groups of components as depicted in Figure 3. Details of the component groups are as follows:

1) Context Provisioner: This component group is responsible for the provision of relevant contexts to the submitted situation spaces. The *context validator* component performs validation of any contexts from the environment to approximate their relevancy to the current situation of application. This component is also responsible for the management of the local *context buffer* by which the less relevant contexts can be removed and replaced by more relevant ones if they are available at the current location.

2) Request manager deals with incoming context query packages from other hosts and from its local situation manager. The request manager will reply to the context query with the required contexts if they are available in the buffer, otherwise it will rebroadcast the query to the neighboring hosts.

3) Situation Reasoner. This component group consists of *situation manager* and *preference manager* that manage a set of *situation spaces* along with their *context preference* submitted by the context application's agent. The context preference is a set of weights for measuring the relevancy value of available contexts of the corresponding situation space. The event manager is responsible for monitoring changing situations that may occur and notifying the corresponding context-aware agent.

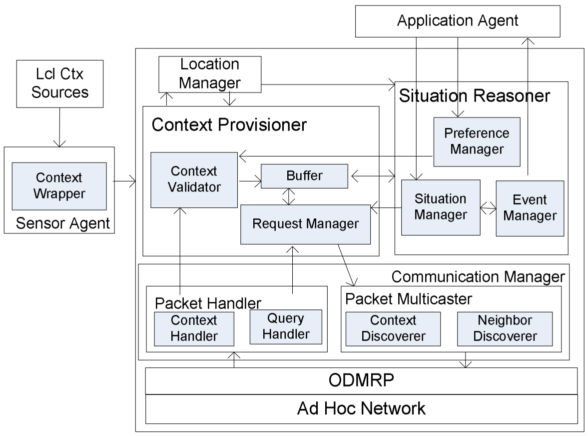

Figure 3. CoMiHoC Architecture

4) Location Manager. This component is responsible for the management of location data represented as an XML file. It provides support for location queries for both the context provisioner and situation reasoner components and updates the location information.

5) Communication Manager. This component group handles collaboration and discovery tasks and consists of a *packet handler* and *multicaster*. The packet handler component continuously monitors incoming message packages from the other collaborative hosts provided by the ODMRP component. A message can contain a context or context query coming from other hosts and is then forwarded by the context provisioner component. The packet multicaster component enables a context or context query to be broadcast or re-broadcast to other hosts in the vicinity as part of the discovery processes. The component is also responsible for monitoring the available hosts or resources in the host's vicinity.

6) ODMRP. The communication manager component is built on the top of On-Demand Multicast Routing Protocol (ODMRP) [7][19][21] as the underlying messaging system to update the distributed contexts among the collaborative hosts. We have adopted and extended an ODMRP-based messaging framework as presented in [19] as the communication layer to support coordination among the mobile hosts.

B. CoMiHoC API

1) Creating situation spaces and basic probability assigment . We create a class *cSituation* as a representation of the situation spaces their corresponding basic probability assignments (BPA). Furthermore, the middleware uses a Java Dempster-Shafer library [15] as a support library for combining evidences based on the DS rule of combination (Eq.3, Eq.4) .

Suppose we have four different relevant contexts (i.e., *Temperature*, *Smoke Present*, *Gas Leak*, and *Physical Damage*) used as regions of the situation spaces. For each acceptable region, a set of *masses* (m) are assigned as the *basic probability assignments* (BPA) that represent *beliefs* of possible situation occurrences under the consideration *i.e.* Safe (S), Risky (R), Dangerous (D). The following code snippet illustrates steps to construct a situation space for a corridor location (H/7/south/). Initially, we can assign an equal mass for each of the possible situations (0.3) followed by an unknown situation with the remaining mass (0.1) as illustrated below:

Suppose we have four different relevant contexts (i.e., Temperature, Smoke Present, Gas Leak, and Physical Damage) used as regions of the situation spaces. For each acceptable region, a set of masses (m) are assigned as the basic probability assignments (BPA) that represent beliefs of possible situation occurrences under the consideration i.e. Safe (S), Risky (R), Dangerous (D). The following code snippet illustrates steps to construct a situation space for a corridor location (H/7/south/). Initially, we can assign an equal mass for each of the possible situations (0.3) followed by an unknown situation with the remaining mass (0.1) as illustrated below:

```
// example of creation situation instances with an equal initial belief
// of for each of relevant situations
situationSpaces= new cSituation("southCorridor","H/7/south");
situationSpaces.addInferredSituation('S',"Safe",0.3);
situationSpaces.addInferredSituation('R',"Risky",0.3);
situationSpaces.addInferredSituation('D',"Dangerous",0.3);
situationSpaces.addInferredSituation(situationSpaces.getClutter(),"Unkno
wn",0.1); // clutter = Θ, i.e.{'S','R','D'}
```

The following snippet code illustrates an example of creating an evidence space (*Smoke Level*) followed by creating a region of acceptable values (*region space*) along with their predicates (*value=NO*) which correspond to the situations under consideration. Masses of the *BPA* are then allocated for each possible situation.

```
// example of adding an evidence space into the situation space
 evidenceSpace = new EvidenceSpace("Smoke_Level");
 elementRegionSpace= new regionSpace("Smoke_No");
 p1 = new Predicate("=", "NO"); //
 Predicate[] predicates4 = {p1};
elementRegionSpace.addSubRegion(predicates4);
elementRegionSpace.addSubRegionBPA('S', 0.5);
elementRegionSpace.addSubRegionBPA('R', 0.2);
elementRegionSpace.addSubRegionBPA('D', 0.2);
elementRegionSpace.addSubRegionBPA(U1, 0.1);
evidenceSpace.addElementRegion(elementRegionSpace);
situationSpaces.addEvidence(evidenceSpace);
```

2) Situation Reasoning. To enable awareness of situation changes in environment, the middleware needs to have the capability of situation reasoning support for the context-aware application. The following algorithm depicts steps to obtain the beliefs of each situation occurrence under consideration by using the available contexts in the buffer.

The first part of the algorithm is to find an acceptable *region space* (e.g. *Smoke Level = LIGHT*) for the *evidence space* of the *situation space* given the available contexts in the buffer. The second part of the algorithm are steps to extract basic probability assignment for the valid *region space* followed by computing the discounted mass of each probable situation of the sub-region using Eq.3, Eq.4. The discounted masses are then used as the current valid evidences. The step is then continued by computing belief for each possible situation using the DS rule of combination (Eq. 1, Eq.2)

```
// find valid situation spaces and their BPAs given current contexts
 FOR  each evidence space  in situation spaces {
     FIND relevant context to the evidence space  in context buffer
     IF a relevant context to the evidence space is found {
     FOR  each each region space of the evidence space  {
        check validity of the context towards the region's  predicates
       IF a valid region space  is found (active subregion is found)  {
             extract BPAs from the valid region space
             compute discounted mass for the extracted BPA
             add the discounted mass as the valid evidences
       }
    }
}
//combine valid Evidences using DS Rule
FOR each valid evidences
   Situations_Belief = combine each valid evidences using DS Rule
// get Belief for each possible situation
For each possible situation j in Situations_Belief
      Belief(situation j) = Situation_Belief.getBelief(situation j)
```

3) *Context Distribution*

We have adopted and extended an ODMRP messaging framework [19] as the underlying messaging protocol to update the distributed contexts among the collaborative hosts. In this scheme a context provider's host advertises its context request by building a multicast group. This is done by constructing a group of forwarding hosts between the provider's host and multicast receivers as the context users. The forwarding hosts re-broadcast the packet they have received until it reaches the corresponding users and establish mesh-based routes [10]. The steps to distribute contexts among the collaborative hosts are adopted from [19] as described below:

- A context provider's host periodically advertises its context by broadcasting a context advertisement as a join query message. The message includes a TTL (time-to-live) field that represents the maximum number of hops the context advertisement is expected to travel. Each message also contains context element as an object
- When a host receives a non-duplicate join query message, it stores the upstream sender into its routing table and rebroadcasts the message.
- When a host receives the context advertisement and the embedded context is relevant to the host's situation spaces, it replies to the packet by broadcasting a join reply message to its neighbors. The process in the host is then continued by checking the relevancy of the new context according to the predefined preferences. A similar context in the host's buffer will be replaced if its relevancy value is lower than the new one, otherwise the new context will be discarded.
- When a host receives a join reply message, it checks if the host id is one of the entries of the message. If it is on the path of the source it becomes a member of the forwarding group and broadcasts the message. Otherwise, it will do nothing. The join reply message is then propagated by the forwarding group members until it reaches the origin of the context provider source.

Frequent disconnections due to the mobility of the hosts and changing network topology are common phenomena in opportunistic networks. Hence, context providers must refresh the route and group membership by periodically sending the join query message to refresh the route and group membership [10].

VI. IMPLEMENTATION AND EXPERIMENTATION

A Java-based prototype of the proposed context middleware has been developed and run on a PDA device which uses Mysaifu [12] as a Java Virtual Machine in a Windows Mobile operating system device. In this section we also discuss the implementation and experimentation issues of the framework over the ad hoc network of personal digital assistance (PDA) devices as a proof-of-concept of the framework. Synthetic data were generated and broadcast using laptops which acted as context providers. A route discovery based on the location model has also been developed. The location data were stored in an XML file copied into the PDA devices which acted as users' hosts. In the experimentation, the framework has also been run in laptops acting as context providers' hosts.

A. *Experimentation*

In the experimentation, the MANET group was formed by using two laptops which acted as context providers and two PDAs (HP iPAQ 5500/2200) which acted as the users' devices. The provider hosts advertised their contexts by sending join query messages containing context advertisements to the hosts in the MANET group. In the simulation scenario, synthetic context data were generated and distributed by multiple context providers. In this scenario, similar contexts could be sent by different hosts, which may have raised problems of context ambiguity and redundancy. Accordingly, the relevancy values of the incoming contexts were used to select the most appropriate context for the application's purposes. In addition, current contexts in the buffer longer relevant to the corresponding situations of the application were discarded if their relevancy value was low according to a certain threshold.

Suppose we have a scenario, in which a team member (user2) has to check a room (R.Meeting1) on the 7th floor of building H (see Figure 3). The current location of the user is at the exit door (E.S1) in the front corridor of the floor. The user uses his PDA and requests route information to the destination as shown in Figure 3. The route paths consist of ordered locations to the destination. We use location information of the paths and the defined situation spaces to determine relevant contexts and possible situation occurrences at the related locations. In the experiment, the user first needs to assign a relevance weight for each quality attribute. For instance, the user may specify *spatial coverage* as the most important weight (8) followed by *temporal validity freshness* (6) and *probability of correctness* (4) as shown in Figure 4. By using these values, the discount masses of the basic probability assignments are then computed. Suppose that a host captures relevant contexts to the defined situation space, i.e. *Smoke Level=LIGHT*, *Gas Leak=NO*, *Physical Damage=LIGHT* and *Temperature=NORMAL.* The relevancy values of the contexts are then computed once they have been captured by the users' devices and the discounted masses are calculated. Finally, probabilities of situations occurrences are computed and the corresponding alert or changes of application behavior can be carried out accordingly.

The experimentation of situation reasoning was run in a PDA device (HP iPAQ 5500) as illustrated in Figure 4. It shows that the most probable situation at H/7/south is Risky (0.45454) followed by Dangerous (0.292185), and Safe (0.22909) respectively. The probability of Unknown situation (0.024169) denotes the ignorance of situations given the available evidences. Additional evidences such as fire status, noise level, building structure instability can be added to the defined situation spaces to obtain a more advanced and robust situation model.

VII. CONCLUSION

We propose a Java-based middleware framework of context management in a MANET environment. The middleware aims to manage different aspects of context and situation reasoning and other related tasks. We propose *CoMoS* (Context Mobile Spaces) as the basis of context model and situation reasoning and implement the proposed model in the middleware framework. The middleware framework has also integrated a hybrid location model and context quality attributes for approximating the relevancy of context used for situation reasoning. To deal with MANET issues, we adopt *On-Demand Multicast Routing Protocol* (ODMRP) as the underlying messaging protocol among the collaborative hosts. The Java-based framework prototype has been tested on PDA devices which run *Mysaifu* as the Java virtual machine for the PDAs. Our preliminary implementation and experimentation demonstrate the capability of the framework to work in such a limited device and to perform the designed capabilities of the middleware.

Further development will include the improvement of the context relevancy model by incorporating more advanced aspects of both context and MANET characteristics. Experimentation based on a MANET simulation tool is also necessary to evaluate the performance of the proposed framework in a large number of mobile hosts and over a wider area. This will form the next phase of our work.

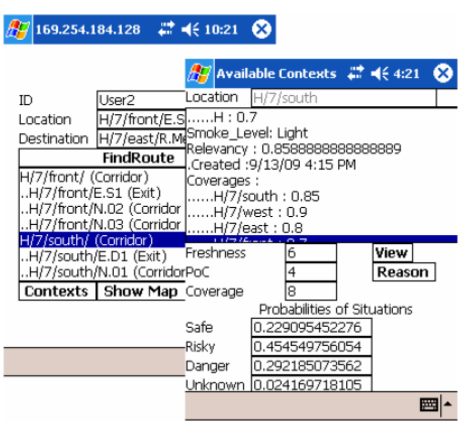

Figure 4. Path discovery and situation reasoning (running on HP iPAQ 5500)

REFERENCES

[1] M. Baldauf, S. Dustdar, and F. Rosenberg, "A Survey on Context-Aware Systems," Int. Journal Ad Hoc and Ubiquitous Computing, vol. 2, pp. 263-277, 2007.

[2] C. Becker and F. Durr, "On location models for ubiquitous computing," Personal and Ubiquitous Computing 9(1), pp. 20-31, 2005.

[3] A. K. Dey, G. D. Abowd, and D. Salber, "A Context-based Infrastructure for Smart Environments," in Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99), Dublin, Ireland, pp. 114-128, 1999.

[4] L. Feng, P. M. G. Apers, and W. Jonkers, "Towards Context-Aware Data Management for Ambient Intelligence," in Proceedings of the 15th International Conference on Database and Expert Systems Applications, LCNS 3180, pp. 422 – 431, 2004.

[5] S. Hadim, J. Al-Jaroodi, and N. Mohamed, "Trends in Middleware for Mobile Ad Hoc Networks," Journal of Communications, vol. 1, pp. 11-21, 2006.

[6] Y. Kim and K. Lee, "A Quality Measurement Method of Context Information in Ubiquitous Environments," in Proceeding of International Conference on Hybrid Information Technology, pp. 576-581, 2006.

[7] S.J. Lee, M. Gerla, and C.C. Chiang, "On-Demand Multicast Routing Protocol," in Proceeding of IEEE WCNC'99, pp. 1298–1302, 1999..

[8] H.Lei, D.M Sow, J.S Davis II, ,G. Banavar, and M.R Ebling, "The Design and Application of a Context Service," Mobile Computing and Communication Review, 6(4), 45-55(2002)

[9] M. Krausse, and I. Hochstatter, "Challenges in Modelling and Using Quality of Context (QOC)," in Proceeding of The 2nd International Workshop on Mobility Aware Technologies and Applications (MATA 2005), LCNS 3744, pp. 324-333, 2005.

[10] M. Mauve, J. Widmer, and H. Hartenstein, "A Survey on Position-Based Routing in Mobile Ad-Hoc Networks," IEEE Network, vol. 15, pp. 30-39, 2001.

[11] C. Jiang and P. Steenkiste, "A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing," in Proceeding of Ubicomp 2002, LNCS 2498, pp. 246-263. 2002

[12] Mysaifu, "Mysaifu JVM - A free Java Virtual Machine for Windows Mobile" http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html

[13] A. Padovitz, A. Zaslavsky, and S. W. Loke, "A Unifying Model for Representing and Reasoning About Context under Uncertainty," in Proceeding of 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems,IPMU 2006, pp. 1983 - 1989 2006.

[14] A. Padovitz, S.W. Loke., A. Zaslavsky, "Multiple-Agent Perspectives in Reasoning About Situations for Context-Aware Pervasive Computing Systems," IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, vol. 38(4), 2008.

[15] T. Reineking. Java Dempster-Shafer library, http://jds.sourceforge.net/

[16] G. L. Rogova and V. Nimier, "Reliability in Information Fusion : Literature Survey," in Proceeding of the 7th Intl. Conference on Information Fusion, pp. 1158-1165, 2004.

[17] G. Shafer, "A Mathematical Theory of Evidence," Princeton University Press, 1976.

[18] 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, pp. 83-93, March 2008.

[19] E. Vollset, D. Ingham, and P. Ezhilchelvan, " JMS on Mobile Ad Hoc Networks," in Proceeding of Personal Wireless Communication (PWC 03), pp. 40-52, 2003.

[20] H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang, "Sensor Fusion Using Dempster-Shafer Theory," in Proceeding of IEEE Instrumentation and Measurement Technology Conference, pp. 7-12, 2002.

[21] E. Yoneki and J. Bacon, "Dynamic Group Communication in Mobile Peer-to-Peer Environments," in Proceeding of ACM Symposium on Applied Computing (SAC 2005), pp. 986-992, 2005.