Abstract—In open, heterogeneous, context-aware pervasive computing systems, suitable context models and reasoning approaches are necessary to enable collaboration and distributed reasoning among agents. This paper proposes, develops and demonstrates: (1) A novel context model and reasoning approach developed with concepts from the state-space model, describing context and situations as geometrical structures in multidimensional space. (2) A context algebra based on the model, which enables distributed reasoning by merging and partitioning context models that represent different perspectives of computing entities over the object of reasoning. We show how merging and reconciling different points of view over context enhances outcomes in reasoning about context. We develop and evaluate our proposed algebraic operators and reasoning approaches with cases using real sensors and with simulations. We embed agents and mobile agents with these modeling and reasoning capabilities, thus facilitating context-aware and adaptive mobile agents operating in open pervasive environments.

Index Terms—context-awareness, context modeling and reasoning, agents and mobile agents, spatial metaphors

I. INTRODUCTION

Research in context-aware pervasive computing investigates the context in which entities in a pervasive system operate and their awareness to changes in that context. Research in context-awareness progressively evolves from establishing the viability of using context towards dealing with major challenges originating from the underlying characteristics of pervasive systems. From stand-alone applications, which demonstrate the benefits of using context, research has begun to look at modeling and reasoning about context in uncertain, distributed, open and heterogeneous computing environments. The inherent pervasiveness, heterogeneity and information uncertainty in such systems raises challenges for computing methods and feasible architectures that would support diverse clients and applications.

This paper addresses two key challenges in context-aware pervasive computing:
(i) Representing context and situations in a unifying model, including reasoning about context under uncertainty with information represented in the model.
(ii) Enhancing context-awareness in distributed, open and multi-agent pervasive systems by dynamically updating context models and merging perspectives of different entities into a single context representation.

We propose a novel unifying model, termed “Context Spaces”, which describes context and situations using geometrical metaphors and concepts from the state-space model [15]. Based on this model we present an approach to reason about situations under uncertainty. We expand our model by developing a context algebra, comprising operators that enable to merge between the perspectives of different entities in the pervasive environment and enhance the reasoning outcome. Our approach facilitates distributed context reasoning in cooperative context-aware multi-agent systems. We propose, formalize and demonstrate an approach that enables individual agents to reason about occurring situations by considering additional information available from other peer agents at runtime. We develop an ability to transform instantiated context models so that merged perspectives can be obtained and directly reasoned about by agents, individually.

A motivating use-case

Agents capable of independently reasoning about their context without relying on additional third party centralized reasoning services can be advantageous in large scale and open multi-agent systems. This view differs from existing work whereby agents utilize (and possibly becoming dependent on) centralized brokers or reasoning servers (see for example [5, 11]). While the latter approaches make use of the agent paradigm, performing reasoning fundamentally depends on client-server architecture. This can incur restrictions limiting the cases for which such approaches are appropriate, e.g., when the task of reasoning cannot be predetermined, or cannot be shared with the centralized server, or when heterogeneous entities with individual access to information and reasoning rules are involved. Consider the following motivating scenario of agents performing collaboration and distributed context reasoning.

Scenario. Suppose we have a collection of software agents in a pervasive system servicing different users or
organizations. Each agent maintains a context model describing relevant context and situations of interest as defined by its initiator. While these agents might be capable of reasoning about similar or identical context, their underlying description of context and sensing capabilities / information resources are potentially different. In these circumstances what one entity knows about the environment, based on sensory information it uses or has access to, can be significantly different from another entity’s knowledge. This is the kind of client environment ultimately envisioned for pervasive computing systems, motivated by heterogeneity and mobility characteristics of clients. Further, consider the relative point of view of each agent over the object of reasoning. Not only accessible information might be different but also the model or representation of important context, based on available knowledge, can be different. For example, two or more agents might wish to reason about the same object (or situation), but each has a different perspective over the circumstances, e.g., the first agent is situated to the left of the object and the other agent to its right, or, one agent is in the open, sensing light in a room whereas another resides on a PDA in a briefcase, detecting dim lights. The contextual information is contradictory while the object of reasoning is identical.

Finally, uncertain conditions governing sensory originated information, such as unreliable sensor technology or frequent wireless communication disconnections require agents also to cope with imposed lack of information.

Agents evidently hold a partial view of the environment and wish to obtain a more reliable reasoning outcome by also considering other agents’ perspectives. Enabling such reasoning capability by an agent calls for an ability to merge context perspectives of other agents that possess different models about the environment. When information represented in their models is missing, agents also need to update their context model to reflect their current limited view. □

This paper proposes an approach to enhance reasoning in circumstances such as in the motivating case above, by developing distributed context reasoning in cooperative context-aware multi-agent systems. It is organized as follows. In Section 2, we discuss related work in reasoning about context under uncertainty and architecting context-aware applications. In Section 3, we describe the Context spaces model and the reasoning applied over information represented with the model. In Section 4, we develop and formalize algebraic operators for the model, enabling: (1) an approach for merging context models while preserving the characteristics of the original models, and (2) an approach to updating models due to lack of available context information in a way that best reflects the situation being modeled and reasoned about. In Section 5, we discuss application, implementation and evaluation of our proposed approaches. Our evaluation comprises both simulations and cases using real sensors. We conclude the paper in Section 6.

II. RELATED WORK

Much research in pervasive computing aims towards operating successfully in open, heterogeneous and context-aware computing environments [10, 18, 23]. Realizing this vision requires suitable approaches in modeling, reasoning and architecting effective context-aware systems that can handle reasoning in uncertain and rapidly changing environments. In recent years, research efforts have focused on these aspects of context, including context middleware and toolkits [5, 7, 4] for information acquisition, ontologies that provide vocabularies to describe and share context information [6], different approaches to reason about context [9, 14, 20] and a variety of context models [25].

Of particular interest in dealing with complex and open pervasive systems are: (1) context models that represent context in a general way, (2) reasoning algorithms that can handle varying degrees of uncertainty, and (3) architectures that promote agility and autonomy of individual computing entities.

The agent paradigm has been adopted as a suitable way to architect such systems [5, 11, 12]. With agent oriented computing, autonomous entities in the system individually manage specific tasks and cooperate among themselves with standardized protocols to achieve a greater goal.

In [20, 21, 22], an agent-based middleware for ubiquitous computing environments is proposed. The architectural approach builds on the notion of agents as autonomous entities, which can be applications, services and devices in the ubiquitous environment. ACAI [11] is another agent-oriented infrastructure for context-aware computing. The design employs agents, which handle various issues in the lifecycle of a context-aware computing system. The overall architectural approach appears to be geared towards a centralized architecture with agents using the system’s centralized services. The Context Broker architecture (CoBrA) [5, 7] is another agent oriented approach for building middleware for context-aware computing. While it mainly deals with a specific use case of smart rooms, the architectural approach follows the design of broker agents, which maintain a shared model of context. The broker is responsible for aggregating and maintaining a consistent representation of context. The work in [12, 13] presents agent-oriented architecture to support context-aware services in mobile environments. It describes a multi-agent system providing information and context to mobile users. The middleware deals with storing and retrieving information, reasoning based on templates and Case Based Reasoning [1], communication and aggregation of contextual information.

The above examples demonstrate the use of agents as autonomous entities but not as ones which collaborate to enhance their reasoning about context. Existing approaches mostly rely on centralized reasoning services and shared repositories of context descriptions. This can limit the benefits of using the multi-agent paradigm in promoting agility in complex pervasive systems. Further, these architectures
generally require that context description be kept and managed by central middleware, perhaps contradicting the aim of open decentralized pervasive systems. Such systems need to support heterogeneous agents with their own context description, reasoning approaches and goals. Enabling agent collaboration and distributed reasoning about context requires new modeling and algorithms for agents in a pervasive system.

Finally, as an agility mechanism that can be deployed in future pervasive systems we suggest the mobile agent paradigm. Mobile agents are software components that provide agility and re-configurability to a system by being able to move on their own volition or invited, from one host to another in a fairly autonomous way (i.e., not only is data transferred, but also code and computation state). While not all agents require mobility, this property has been recognized as beneficial in a number of large scale distributed applications, including large networks of embedded systems such as wireless ad-hoc sensor networks [19].

Very little work has been done on incorporating mobile agents into pervasive computing systems. We note [24, 8, 2] as first attempts to use mobile agents in such environments.

III. CONTEXT SPACES MODELING AND REASONING

Context Spaces aims towards a general context model to aid thinking and describing context, and to design operations for manipulating and utilizing context. The concepts use insights from geometrical spaces and the state-space model, hypothesizing that geometrical metaphors such as states within spaces are useful to guide reasoning about context. The model provides a unifying and general way to represent context and enables effective reasoning to be applied over the modeled information. We describe five key concepts of the Context Spaces model. We then describe an approach to reason about context under uncertainty based on the model.

A. Modeling principles

The model distinguishes between the concept of context and that of situation. Context is the information used in a model for representing real world situations. Thus, situations are perceived as a meta-level concept over context. The process of assessing the association and mapping between context and situations, or in other words, determining occurrences of situations based on information in a model, is the task of reasoning.

This relationship between context and situations is represented in a general way by the concepts of state and space. We perceive an application space – the universe of discourse in terms of available contextual information for an application. The application space is determined by the types of information, deemed relevant and obtainable by the system designers. It is a multi-dimensional space made up of a domain of values for each relevant information type, in which context can be sensed. Within it we perceive subspaces (possibly defined in fewer dimensions), which reflect real-life situations. We call these subspaces situation spaces. Situation spaces are defined over regions of values in selected dimensions and represent collections of values that reflect real-life situations. The actual values of sensory originated information are defined by the context state, e.g., the collection of current sensor readings representing a specific context.

We consider the following concepts:

Context Attribute (Definition 1) A context attribute (denoted by $a_i$) is defined as any type of data that is used in the process of reasoning about context. A context attribute is associated with a sensor, logical, computed or physical. The value of the sensor reading at time $t$ is the context-attribute value at time $t$ (denoted by $a_i^t$).

For example, light-level (as a type of information) can be represented by a context attribute, say $a_1$, (and sensed by a light sensor). A specific value, sensed at time $t$ is denoted by $a_1^t$.

Context State (Definition 2) A context state is a collection of context attributes’ values that are used to represent a specific state of the system at time $t$. A context state is $C^t = (a_1^t, a_1^t, ..., a_n^t)$ where $C^t$ denotes a tuple defined over a collection of $n$ attribute-values, where each value $a_i^t$ corresponds to an attribute $a_i$’s value at time $t$.

For example, a context state $C^t$ at time $t$, is made up of specific context attributes values such as say, light-level ($a_1^t$), noise-level ($a_2^t$) and motion-level ($a_3^t$).

Situation space (Definition 3) A situation space represents a real-life situation. It is a tuple of regions of attribute values corresponding to some situation and denoted by $S_j = (A_1^j, A_2^j, ..., A_n^j)$ (consisting of $n$ acceptable regions for these attributes). An acceptable region $A_i^j$ is defined as a set of elements $V$ that satisfies a predicate $P$, i.e. $A_i^j = \{ V | P(V) \}$.

For example, in numerical form the accepted region would describe a domain of permitted real values for an attribute $a_i$ such as the region of values of body temperature between 36.2°C and 36.9°C, representing the range of temperature values of a “healthy person” situation (denoted by, say $A_i^j$). Or, a collection of non-numerical values such as ‘{Fine’, ‘Sunny’, ‘Partly Cloudy’} which represents acceptable values for weather condition context attribute in some situation.

An illustration of Context Spaces’ concepts discussed so far is presented in Figure 1. Dimensions are defined by context attributes, representing relevant and obtainable types of information. Situation spaces (figuratively illustrated in multidimensional space) represent specific regions of values in selected dimensions, which reflect situations of interest. The context state fluctuates with respect to time and could
where occasionally match the definitions of some situation space. The context state trajectory, i.e. a curve in space that the system follows over time in a subset of context attributes, can be computed and related concepts such as the pervasive systems’ stability or instability in a given context (i.e. the ability to identify and possibly affect the steadiness of the system state in a given context) become possible [17].

Fig. 2. Illustration of the model’s elements. A collection of situation spaces and the context state, figuratively in multidimensional space, defined over the system context attributes

To gain more comprehensive modeling and attain effective reasoning under uncertainty we extend this basic representation with the following modeling functions.

Relevance function

In many cases some types of information are more important than others for inferring a situation, e.g., high body temperature may be a strong indication of a general sickness of a person while other attributes may not be so important in inferring that specific situation. For example, high respiratory rate may be caused by a person doing physical exercise and therefore also indicative of other situations. To model this difference in the importance of context attributes for inferring a situation, we define the relevance function, which assigns weights to context attributes. The weights reflect how important each attribute is (relative to other attributes) for describing a situation.

(Definition 4) Given a situation space \( S_j = (A_1^j, A_2^j, \ldots, A_n^j) \), a relevance function \( \varepsilon_R \) associates weights \( w_1^j, w_2^j, \ldots, w_n^j \) with regions of values \( A_1^j, A_2^j, \ldots, A_n^j \) of \( S_j \), respectively, where \( \sum_{k=1}^{n} w_k^j = 1 \) (1 \( \leq k \leq n \)). A weight \( w_k^j \in [0,1] \) represents the importance of an attribute region \( A_k^j \) relative to other regions in the situation space.

Contribution function

In the relevance function, we model the relative importance between the attributes of a situation space, whereas in the contribution function we model the individual contribution of elements within a specific region for inferring a situation. That is, more than merely knowing that the sensed value is within or not within a region, if the value is within, we also consider the particular value itself. The fact that the value is within the region is indicative of the situation, and even more so indicative if the value is of some range (within the region) - how much so is what the contribution function represents. For example, for a context attribute of ‘body temperature’ in the definition of ‘subject is healthy’ the values between 36.5°C and 36.7°C would reflect high contribution and values between 36.3°C and 36.5°C and between 36.7°C and 36.8°C would reflect a lesser degree of supportive contribution.

(Definition 5) Given an acceptable region of values \( A_k^j \) (corresponding to some context attribute \( a_j \)), a function \( \eta_k^j \) assigns a contribution level \( c \in [0,1] \) for each element (i.e. a value of a context-attribute) in the region \( A_k^j \).

The contribution level of an element in a region reflects how well that element is associated with the modeled situation. In each region of values there must be at least one element with a complete contribution (i.e. with a value of one).

B. Reasoning approach

Given the model, a reasoning approach for inferring that a situation modeled with a situation space is occurring, is achieved by accumulating evidential indications. An indication is the fact of a context attribute value (a part of the context state) being contained within the corresponding region of acceptable values in the definition of the situation space. In other words, the occurrence of a situation is represented by a tuple of values (i.e., the context state comprising values obtained via sensors (and perhaps some reasoning)) being within a tuple of accepted regions (i.e. the situation space representing the situation). Therefore, the multi-dimensional position of the context state in reference to a chosen situation space provides an indication for the occurrence of the real-life situation. Once there is accumulation of sufficient supportive indications, the occurrence of a situation can be inferred.

Based on the above approach we compute a measure of confidence in the occurrence of a situation by accumulating evidential supportive indications. Mathematically, given a context state, a reasoning procedure computes a measure of confidence in a situation space by:

\[
\mu(S) = \sum_{i=1}^{k} w_i c_i \text{, where} \mu \text{ is the reasoning function applied over a situation space } S, w_i \text{ is a weight value of importance assigned to a context-attribute } a_i \text{ (in } S) \text{ and}
\]
\( c_i \) denotes the contribution level of the specific value of a context.attribute \( a_i \).

The above formula and terminology to compute confidence using the relevance and contribution functions draws on Multi-Attribute Utility Theory, which provides a convenient way for combining together seemingly different contributions into a single measure, expressing the result in terms of utility [26]. In our case, we see utility (or contribution towards our goal of determining the occurrence of a situation) as the degree of evidential support given to the hypothesis of a situation occurring when a context attribute value is within the corresponding region. The more indicators we have that the context state matches the definition of a situation space, greater utility (or confidence in the situation) is gained.

We further extend the reasoning procedure by proposing an approach to incorporate the impact of sensors inaccuracies and unreliability into the contribution computation (which can be performed at run-time rather than at design-time). We consider a heuristic stating that the greater likelihood of a context attribute being contained within a region, the greater contribution should be evaluated for that context attribute, and vice-versa. So, for example when we sense dimmed lights in the smart room, it might be a strong indication of a presentation taking place. However, if the light sensor is inaccurate (and might actually sense normal light levels) then the contribution of sensing dimmed lights towards inferring a presentation should be reduced. This kind of heuristic enables more accurate distinction between available information for reasoning under uncertain condition. We represent the confidence measure using this heuristic as:

\[
\mu(S) = \sum_{i=1}^{n} w_i \cdot \Pr(\tilde{a}_i \in A_i), \quad \text{where the term } \Pr(\tilde{a}_i \in A_i) \quad \text{represents the confidence of having the correct value being sensed contained within its corresponding region of acceptable values. Here the term } \Pr(\tilde{a}_i \in A_i) \quad \text{replaces } c_i \quad \text{as a proposed method to compute contribution values at run-time.}
\]

In the process of reasoning about a situation the computed confidence is compared with a confidence threshold for the specific situation, facilitating a decision regarding the occurrence of that situation. (We contrast the computed confidence with the individual threshold of the particular situation, thus gain the ability to compare between outcomes computed for different situations.) For example, suppose one computes the confidence in the occurrence of situation \( S_j \) using equation (1) or (2), and let \( T_j \) be the threshold for situation \( S_j \), then when \( \sum_{i=1}^{n} w_i c_i > T_j \) we conclude that situation \( S_j \) is occurring. Situation thresholds can be predetermined by the system developers or computed. For example, by selecting a context state that sufficiently reflects that situation, according to the application designers’ judgment. Applying function (1) or (2) over this information would yield a representative threshold of confidence.

### IV. REASONING BY MERGING PERSPECTIVES

The context model and reasoning approach we have outlined is useful in describing and reasoning about context in a general and effective way. It does not provide, however, in its current form, a way to apply reasoning over a combined model, which is the key ingredient for achieving our objectives. We now develop approaches, which can be used at runtime by agents to merge seemingly dissimilar context models (i.e., situation spaces) into a consistent and richer representation of modeled situations.

The task of merging information, as dictated by the underlying Context Spaces model consists of two tasks. The first concerns acquisition of relevant contextual information, i.e., aggregating context.attribute values to build a richer context-state. This task is, arguably, relatively simple as it relies on appropriate communication mechanisms to share sensor reading values between agents. (This task becomes more challenging once agents are also mobile; we have addressed this issue in [16]). The second task concerns combining individual context models that reflect the beliefs and capabilities of different agents into a single model. Having such a model provides the necessary information for requesting relevant context-state information. The second task is more complex as it requires an approach that preserves the characteristics of the original situation spaces.

We distinguish between two kinds of operations for merging context models governed by different assumptions. The first assumes a general case where models describing some situation are potentially very different, making use of inherently different kinds of sensory information. In such conditions our goal is to combine these models into a single model and achieve a reasoning result which is comparable to the averaged result of performing reasoning separately with each original model. In the second approach we assume that certain similarities between models exists so that by merging them into a single representation we can achieve reasoning identical to a global model, which is defined over all existing information.

**Example**

Suppose that two situation spaces are defined for inferring the situation of a ‘business meeting’, each held by a different agent with different sensing capabilities. The first agent is a context reasoning application operating in the user’s PDA. It has access to light and motion sensors that are integrated with the PDA and communicates with the user’s scheduling application. It also communicates with a remote positioning engine (e.g., Ekahau engine [27], which tracks locations of devices such as PDA’s) to receive the user location estimates. The second agent is a context manager of the smart room, which utilizes different sensors within the room to compute situations such as the activity of users in the
room. The following table contains the definitions used by these entities to represent a business meeting situation.

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td>SITUATION SPACES DEFINED IN DIFFERENT CONTEXT</td>
</tr>
<tr>
<td>Attribute name</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Agent A</td>
</tr>
<tr>
<td>Location</td>
</tr>
<tr>
<td>Motion level</td>
</tr>
<tr>
<td>Light level</td>
</tr>
<tr>
<td>Scheduler</td>
</tr>
<tr>
<td>Agent B</td>
</tr>
<tr>
<td># of People</td>
</tr>
<tr>
<td># of People standing</td>
</tr>
<tr>
<td>Projector activity</td>
</tr>
</tbody>
</table>

Suppose that the confidence computed for the meeting given the information sensed by and defined for Agent A and Agent B is \( \mu_A(\text{meeting}) = \mu_{21} \) and \( \mu_{12}(\text{meeting}) = \mu_{22} \), respectively. We want to enable different entities, for example the context reasoning process running on the user’s PDA, to benefit from other entities’ context description and available information. This would enhance Agent A’s reasoning capabilities. While sensory originated information can be shared with appropriate communication mechanism the more challenging task is to construct a model (a situation space) that takes into account situation descriptions of other entities (all relevant information is represented there). In the first proposed merge operation our goal is to produce a reasoning outcome over such a model that is comparable to averaging the computed confidence of both original models, performed separately. How much we trust the models and relevant information would determine the way we average these results. In the general case, for simplicity, we will assume equal trust in the two definitions for representing the meeting situation and compute average results. In our second proposed merge operation, we will show how the reasoning outcome can be even further improved, gaining better reasoning outcomes (i.e. more reliable and indicative) than the averaged result.

Rather than performing inference over separate situation spaces by each and every computational entity we create a single situation space and reason about it. An entity can benefit from other entities context representations without the need to request their inference results each time.

A. Combining models for the general case

Let us develop a combination operation, denoted by the symbol \( \oplus \) applied between situation spaces. The operator combines information from different definitions obtained by different entities and produces a single situation space that can be directly reasoned about.

To obtain a more reliable result we require that the support for a situation, which is created by combining several situation spaces, would be the same as the averaged support for the original situation spaces.

Our first merge operation consists of three procedural definitions:
(i) Represent the original situation spaces in a new situation space (i.e. composing the regions of acceptable values).
(ii) Transform the relevance values (i.e. the weights) of the original situation spaces into relevance values in the new situation space.
(iii) Transform the contribution values from the original situation spaces into contribution values in the new situation space.

Below we define these three procedures.

(i) Composition of Situation Spaces (Definition 6).

Given the regions of acceptable values \( A_k^1 = \{V | P(V) \subseteq T \} \) and \( A_k^1 = \{W | Q(W) \subseteq T \} \), where \( T \) is the domain of all acceptable values (i.e., \( A_k^1 \) and \( A_k^1 \) are subsets of the same set, effectively of the same type), then the composition of \( A_k^1 \) and \( A_k^1 \) is a set, denoted by \( A_k^1 \oplus A_k^1 \), and every element \( e \) of this set satisfies:

\[ e \in (A_k^1 \oplus A_k^1) \iff P(e) \lor Q(e) = 1. \]

The composition of two situation spaces \( S_i = (A_i^1, A_i^2, ..., A_i^n) \) and \( S_j = (A_j^1, A_j^2, ..., A_j^m) \) is denoted by \( S_i \oplus S_j \) and is defined as follows:

\[ S_{ij} = S_i \oplus S_j = (A_{ij}^1, A_{ij}^2, ..., A_{ij}^k, A_{ij}^{k+1}, ..., A_{ij}^{n}, A_{ij}^{m+1}, ..., A_{ij}^{m}) \]

where we assume that the first \( k \leq \min(n, m) \) attributes are common in both \( S_i \) and \( S_j \) (without loss of generality) though their accepted regions might not be, and are computed by:

\[ A_{ij}^k = A_i^k \oplus A_j^k \]

The merged situation space guarantees that no information associated with the original situation spaces definitions is lost and is represented in the new situation space.

(ii) Relative weight transformation (Definition 7).

Let \( w_i^j \) and \( w_k^j \) denote weights assigned to the regions \( A_i^j \) of \( S_j \) and \( A_k^j \) of \( S_k \) corresponding to the same context attribute, respectively. Let \( q_i \) denote the sum of weights for a region of values of the same context attribute in each original situation space:

\[ q_i = \begin{cases} w_i^j + w_k^j & \text{if both } w_i^j \text{ and } w_k^j \text{ exist} \\ w_i^j & \text{if only } w_i^j \text{ exists} \\ w_k^j & \text{if only } w_k^j \text{ exists} \end{cases} \]

E.g. if the pairs \((w_i^j, A_i^j)\) and \((w_k^j, A_k^j)\) denote a region of values and associated weights for context attribute \( a_i \) in situation spaces \( S_j \) and \( S_k \) respectively, then \( q_i = w_i^j + w_k^j \) or, if such context attribute only exist in \( S_j \)
then \( q_i = w_i^j \). A transformed weight in the new composed situation space for context attribute \( a_i \) is the relative weight between \( q_i \) and the sum of the total weights. This is computed as follows:

\[
(1) \hat{w}_i = q_i / \left( \sum_{j=1}^{n} q_j \right), \quad \text{where } \hat{w}_i \text{ denotes the new weight and } n \text{ is the total number of unique context attributes.}
\]

(The computation applies to any composition of any combination of situation spaces, e.g. if we transform three spaces: \( S_1, S_2 \) and \( S_3 \) then we consider all elements of those three spaces.)

The weight transformation procedure computes the new relative weight of a context attribute in the composed situation space. This is achieved by examining the overall weight or importance of that attribute in both original spaces, compared with the overall weight of all existing context attributes.

(iii) Contribution transformation (Definition 8). Let \( A_{i}^{S_1}, A_{i}^{S_2}, \ldots, A_{i}^{S_n} \) denote corresponding regions of values of the same context attribute \( a_i \) in \( n \) original situation spaces. Let \( w_{i}^{S_1}, w_{i}^{S_2}, \ldots, w_{i}^{S_n} \) denote the original weights associated with each of these regions, respectively. Let \( e \) denote a common element within some of the regions \( A_{i}^{S_1}, A_{i}^{S_2}, \ldots, A_{i}^{S_n} \). Let \( c_{i}^{S_1}, c_{i}^{S_2}, \ldots, c_{i}^{S_n} \) denote the original contribution of an element \( e \) in these regions, respectively.

For each element \( e \) compute a contribution level, as follows.

\[
\bar{c} = (w_{i}^{S_1} c_{i}^{S_1} + w_{i}^{S_2} c_{i}^{S_2} + \ldots + w_{i}^{S_n} c_{i}^{S_n}) / (w_{i}^{S_1} + w_{i}^{S_2} + \ldots + w_{i}^{S_n})
\]

where \( \bar{c} \) denote the new contribution level.

The contribution transformation computes the new contribution of an element in the region of values. It is achieved by analyzing the relative contribution of the element in the original definitions. This yields a new contribution level for an element, such that if it is associated with high accumulated weights in original regions, it yields high contribution and vice versa.

We now prove that reasoning over the merged situation space is identical to averaging results of separately reasoning on original situation spaces – the significance of this is a compositional measure expressible in terms of constituent measures.

**Proposition:** Given \( m \) situation spaces \( S_1, S_2, \ldots, S_m \), we have:

\[
\mu(S_1 \pm S_2 \pm \ldots \pm S_m) = \frac{1}{m} \mu(S_1) + \frac{1}{m} \mu(S_2) + \ldots + \frac{1}{m} \mu(S_m).
\]

**Proof:** Let \( w_i^{S} \) and \( c_i^{S} \) denote the weight and contribution level of a region of acceptable value \( A_i^{S} \) in a situation space \( S \). Let \( Y = S_1 \pm S_2 \pm \ldots \pm S_m \) denote a composed situation space with \( n \) context attributes (i.e. we have \( w_i^{Y}, c_i^{Y} \) for \( i = 1 \ldots n \)). The computation of a weight for a region \( A_i^{Y} \) in \( Y \) by definition 7 can be written as:

\[
w_i^{Y} = (w_i^{S_1} + w_i^{S_2} + \ldots + w_i^{S_n}) / \sum_{i=1}^{n} (w_i^{S_1} + w_i^{S_2} + \ldots + w_i^{S_n})
\]

By definition 4, the sum of weights in a situation space is 1, i.e. \( \sum_{i=1}^{n} w_i^{S} = 1 \) for every situation space \( S \) (weights associated with context attributes not in the definition of a situation space are zero). hence:

\[
w_i^{Y} = (w_i^{S_1} + w_i^{S_2} + \ldots + w_i^{S_m}) / m
\]

The computation of a contribution level for an element in a region \( A_i^{Y} \) in \( Y \) by definition 8 can be written as:

\[
c_i^{Y} = (w_i^{S_1} c_i^{S_1} + w_i^{S_2} c_i^{S_2} + \ldots + w_i^{S_m} c_i^{S_m}) / (w_i^{S_1} + w_i^{S_2} + \ldots + w_i^{S_m})
\]

We compute the confidence for situation \( Y \) by evaluating each region’s weighted contribution. Hence we can write the matching function as:

\[
\mu(Y) = \sum_{i=1}^{n} w_i^{Y} c_i^{Y} = \sum_{i=1}^{n} \left( \frac{1}{m} \left( w_i^{S_1} + w_i^{S_2} + \ldots + w_i^{S_n} \right) \right) \left( w_i^{S_1} c_i^{S_1} + w_i^{S_2} c_i^{S_2} + \ldots + w_i^{S_m} c_i^{S_m} \right) = \frac{1}{m} \left( \sum_{i=1}^{n} w_i^{S_1} c_i^{S_1} + \sum_{i=1}^{n} w_i^{S_2} c_i^{S_2} + \ldots + \sum_{i=1}^{n} w_i^{S_m} c_i^{S_m} \right)
\]

(During matching, only relevant context attributes are considered)

**B. Merging different situations**

We have presented a procedure for combining information from different sources that model the same situation. In much of the same way, in some cases, we can use the merge operation to combine definitions of inherently different situations. During reasoning (i.e. the function \( \mu \) ) each context attribute is evaluated separately, independently of other context attributes. Thus, as long as the information in different situation spaces can be separately evaluated for inferring another situation space, these spaces can be combined together. So, a situation, which is made up of a composition of other situations but which does not require all of these situations to occur at the same time, can be composed by combining the original situation spaces. If situation spaces \( A, B \) and \( C \) contain information indicating the possibility of \( D \),

\(1\) We distinguish between combining situations from intersecting them, in which case all original situations should occur, rather than just contributing support.
they can be combined into a single space to represent D. For example, we can compose a situation of ‘Person Is Sick’ by composing together indicative situation spaces of, say, ‘Low Physical Activity’, ‘Low Productivity at Work’ and ‘Conversations with Physicians’. Reasoning would then assess the combined information to evaluate the support for the new situation space.

We can also evaluate some situations more than others in the composition of the new situation, simply by repeating them in the combination. For example, A ± A ± B would result in a combined situation, which inference treats the original situation A twice as important as B (yielding results identical to \( \frac{2}{3} \mu(A) + \frac{1}{3} \mu(B) \)).

Example

Let us perform a merge of two situation spaces, reason about the resulted combined situation and compute a confidence measure supporting the new situation. We will then compare this result with averaging the degree of support that is achieved by reasoning about each of the original situations separately. Table II depicts the regions and their associated weights and contribution levels for two situations A and B. For brevity and clarity we assume continuous regions of acceptable values.

<table>
<thead>
<tr>
<th>Situation Definitions</th>
<th>acceptable values</th>
<th>weight</th>
<th>contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region of a</td>
<td>4 - 8</td>
<td>0.6</td>
<td>1</td>
</tr>
<tr>
<td>Region of b</td>
<td>3 - 7</td>
<td>0.2</td>
<td>1</td>
</tr>
<tr>
<td>Region of c</td>
<td>5 - 8</td>
<td>0.2</td>
<td>1</td>
</tr>
<tr>
<td>Region of a</td>
<td>2 - 6</td>
<td>0.4</td>
<td>1</td>
</tr>
<tr>
<td>Region of b</td>
<td>5 - 9</td>
<td>0.4</td>
<td>1</td>
</tr>
<tr>
<td>Region of d</td>
<td>2 - 5</td>
<td>0.2</td>
<td>1</td>
</tr>
</tbody>
</table>

The situations consists of three different regions of values, two of which are of the same context attribute in the two situation definitions, but containing different set of values. The result of applying combination, yielding a single situation space is depicted in Table III. It results in regions specified for four context attributes some with varying degrees of contribution levels.

<table>
<thead>
<tr>
<th>Situation A ± B</th>
<th>region</th>
<th>weight</th>
<th>sub region</th>
<th>contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>a</td>
<td>0.5</td>
<td>2 - 4</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>0.3</td>
<td>3 - 5</td>
<td>0.333</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>0.1</td>
<td>5 - 8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>d</td>
<td>0.1</td>
<td>2 - 5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>0.5</td>
<td>4 - 6</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>0.3</td>
<td>6 - 8</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>0.3</td>
<td>7 - 9</td>
<td>0.666</td>
</tr>
</tbody>
</table>

Consider the following two context states at two different times:

\[
C^1 = (a = 7, b = 6, c = 6, d = 4) \text{ and } C^2 = (a = 5, b = 6, c = 6, d = 7).
\]

Reasoning over the combined situation space A ± B at time \( t_1 \) we obtain: \( \mu_{t_1}(A ± B) = 0.5 \times 0.6 + 0.3 \times 0.1 + 0.1 \times 0.1 = 0.8 \) and at time \( t_2 : \mu_{t_2}(A ± B) = 0.5 + 0.3 + 0.1 + 0 = 0.9 \).

Let us compare these results with separately reasoning about the original situation spaces at time \( t_1 \) and \( t_2 \), which results in the following computed confidences:

\[
\mu_{t_1}(A) = 0.6 + 0.2 + 0.2 = 1 \text{ and } \mu_{t_1}(B) = 0.4 \times 0 + 0.4 + 0.2 = 0.6.
\]

\[
\mu_{t_2}(A) = 0.6 + 0.2 + 0.2 = 1 \text{ and } \mu_{t_2}(B) = 0.4 + 0.4 + 0.2 \times 0 = 0.8.
\]

Averaging results obtained by separately performing reasoning by different agents, yields identical support as in the case of reasoning with the combined situation.

This implies that different situations can be merged together to represent another situation, enabling intelligent compositions of new situations. It also shows that an agent can independently perform reasoning with a richer model, merging other agents’ perspectives, without requesting those other agents to perform reasoning and then analyze their results.

C. Merging opposing perspectives

Our initial combination so far assumes that models describing the same object are defined consistently over the same type of context attribute. This may not be the case, however, when agents operate under different circumstances, holding different perspectives over the object of reasoning. To reconcile between different perspectives of agents we analyze the combined context attribute of the same type. If the regions of acceptable values are defined differently they are treated as different types of contextual information. Thus, the combined situation definition distinguishes between shared description of context attributes in the situation and individual descriptions originating from different agents’ perspectives. For example, if two agents require the projector to be turned on to indicate a presentation activity; this would be represented with a single context attribute in the combined space. If, on the other hand, one agents requires dimmed lights to indicate the presentation and another requires total darkness (resulting from less capable sensing), then we treat these as two different requirements, each associated with its own sensory originated information.

Example

Consider the following two definitions of a presentation activity and the merged model, presented in Table IV. Context attributes with different acceptable regions of values appear separately in the merged definition. For example, Light Level context attribute is defined twice in the merged situation space. Different original definitions of regions of acceptable values indicate different perspectives over the same context attribute, e.g., using different kinds of sensors to determine light, or using sensors positioned in different locations (e.g., under and above the table).
Table IV

<table>
<thead>
<tr>
<th>Presentation</th>
<th>context attributes</th>
<th>acceptable values</th>
</tr>
</thead>
<tbody>
<tr>
<td>(definition 1)</td>
<td>Light Level</td>
<td>Dimmed, Off</td>
</tr>
<tr>
<td></td>
<td>Noise Level</td>
<td>Low, Medium</td>
</tr>
<tr>
<td></td>
<td># of Users</td>
<td>&gt; 4</td>
</tr>
<tr>
<td>Presentation</td>
<td>Light Level</td>
<td>Off</td>
</tr>
<tr>
<td>(definition 2)</td>
<td>Speaker Status</td>
<td>On</td>
</tr>
<tr>
<td></td>
<td># of Users</td>
<td>&gt; 4</td>
</tr>
<tr>
<td>Presentation</td>
<td>Light Level 1</td>
<td>Dimmed, Off</td>
</tr>
<tr>
<td>(merged)</td>
<td>Light Level 2</td>
<td>Off</td>
</tr>
<tr>
<td></td>
<td>Noise Level</td>
<td>Low, Medium</td>
</tr>
<tr>
<td></td>
<td>Speaker Status</td>
<td>On</td>
</tr>
<tr>
<td></td>
<td># of Users</td>
<td>&gt; 4</td>
</tr>
</tbody>
</table>

D. Merging and partitioning with a global model

We have, so far, used an underlying assumption that each agent has his own unique representation of context and can only assume that other agents have similarly representative descriptions of context. We assumed that in the general case, such models can be significantly different without common attributes shared between them. We now change this assumption and assume that agents hold partial views of a single global context model and that there exist common attributes between these sub-models that can be used to reveal the original global model. With such assumption it is possible to merge models which reflect the relative importance of sensors in the global model. Merging models using this assumption requires that some context attributes and their corresponding regions be shared between models, such that a model built with information from all sensors can be reverse-engineered.

While merging a global situation space from its partial representations may not always be feasible, its inverse operation, i.e. partitioning a model into partial sub-models is always feasible and can become quite useful in pervasive scenarios. With such capability, based on a complete representation of context, a partial model consisting of all attainable relevant information can be computed at runtime. An agent reasoning about context adjusts its model based on what sensory information is available, so that the updated model only describes information of available sensory data (with recomputed weights reflecting relative importance between available attributes), and does not consider unavailable information during reasoning. This would provide the most suitable description about context given a limited subset of sensed/discovered information. Many factors contribute to the limitation in accessible information, such as information freshness, faulty sensors, discarded contradicting sensory data or communication faults.

(Definition 10) Let there be a situation space $S = \{A_1, \ldots, A_k, \ldots, A_m\}$ and partial models representing the same situation: $S' = \{A_1, \ldots, A_k\}$ and $S'' = \{A_1, \ldots, A_m\}$ with at least one shared context attribute defined in both partial models (as a region of acceptable values). Let $W' = \{w'_1, \ldots, w'_k\}$ and $W'' = \{w''_1, \ldots, w''_m\}$ represent the sets of weights for corresponding context attributes in $S'$ and $S''$ respectively.

**Merge** - a merged situation space $S$ would consist of the union of all regions of acceptable values between $S'$ and $S''$. A weight $w_i$ of a shared context attribute $A_i$ can be computed by:

$$w_i = \frac{w'_i w''_i}{w'_i + w''_i + \sum_{w_j \in W'} w_j}$$

remaining weights can be computed by:

$$w_m = \frac{w'_m w''_m}{w'_m + w''_m}$$

**Partition** - weights for a partial model of $S$, consisting of a subset of regions can be computed by the relative importance of weights in $S$ applied between weights for regions in the partial model. A weight $w'_i$ is then computed by:

$$w'_i = \frac{w_i}{\sum_{w_j \in W'} w_j}$$

**Proofs**: Merge – Given definition 4, sum of weights in merged situation space $S$, represented with weights from partial models $S'$ and $S''$ can be written as:

$$w'_i w'_i + \sum_{w_j \in W'} w_j + \sum_{w_j \in W'' \setminus W'} w_j w''_i + \sum_{w_j \in W' \setminus W''} w''_i w''_j w''_j = 1.$$ 

Given that in the general case some context attributes are shared in $S'$ and $S''$, it can be rewritten as:

$$w'_i (\sum_{w_k \in W'} w'_k + \sum_{w_j \in W'' \setminus W'} w''_j) + \sum_{w_j \in W'} w_j / w'_i = 1.$$  

Hence:

$$w_i = ((\sum_{w_k \in W'} w'_k) / w'_i + (\sum_{w_j \in W'' \setminus W'} w''_j) / w''_i)^{-1}.$$  

By definition 4, $\sum_{w_k \in W'} w_k = 1$, thus:

$$w_i = (\frac{1}{w'_i} + (\sum_{w_j \in W'' \setminus W'} w''_j) / w''_i)^{-1} = w'_i w''_i + \sum_{w_j \in W'' \setminus W'} w''_j.$$  

Partition - Given definition 4, sum of weights in partial situation space $S'$, represented with weights from original situation space can be written as:

$$\sum_{w_j \in W'} w_j = 1.$$ 

Thus: $w'_i \cdot \sum_{w_j \in W'} w_j = 1$. Rewrite as: $w'_i = w'_i / \sum_{w_j \in W'} w_j \square$.

Using the above operations, agents can produce more complete descriptions about situations they reason about, described with the complete set of available sensory information, instead of their own limited representation. Similarly, when information is missing or not trusted, a more...
indicative model can be constructed at run-time, using the partitioning operation.

V. APPLICATION AND EVALUATION

From a theoretical perspective we turn to application, implementation and evaluation. The proposed ideas and operations enable adaptive, context-aware behavior of computing entities in a pervasive system. Using our approach, such entities not only perform reasoning about context in a 'static' manner (i.e., based only on provided information) but can enhance the reasoning process by proactively seeking and considering information from other entities, including using different perspectives over the same object of reasoning.

The discussed operations have been implemented as part of a Java based context-algebra library, providing various operators that are applied between programmable objects of Context Spaces concepts. Operators in context-algebra library deal with several aspects of reasoning about context, including reasoning about logically conditioned situations under uncertainty, effectively managing and analyzing context information, and enabling proactive and adaptive pervasive systems’ behavior.

The algebra library of functionality is used both by different independent software agents as well as a centralized reasoning component termed CORE (Context Oriented Reasoning Engine), which is a part of a larger framework for context-aware computing. We follow an approach combining a robust reasoning engine with an ability to inject collaborative context-aware mobile agents to the pervasive system. Architectural relationships between CORE and other framework components are depicted in Figure 2. A reasoning component (kernel) makes use of operators in the algebra package during the process of providing reasoning services to clients.

We evaluate our ideas with a scenario of agents’ reasoning about office-related activities in a smart room. We perform experimentations at three levels. (i) First, we prove the concept of reasoning about context under uncertainty using our proposed approach of accumulating supportive indications. (ii) Then, we evaluate our merge and partitioning operators as means to enhance reasoning outcomes in a cooperative multi-agent system. (iii) Finally, we incorporate the concept of mobility into the reasoning process using software mobile agents, demonstrating adaptive pervasive systems’ behavior based on reasoning with context algebra operators.

A. Reasoning about situations under uncertainty

In the following experiment we were interested in identifying and distinguishing between three types of user activities, taking place in a smart meeting room, namely, (1) a user giving a presentation, (2) a user attending another’s presentation and (3) a user attending a meeting. We combined together different real sensors and data sources as the information used in defining situation spaces and computing the context state.

To reason about these activities we have selected four basic context attributes, whose sensors were physically positioned in different locations. These context attributes corresponded to the user location, the meeting room light level, the user’s notebook keyboard and mouse activity and presence of active presentation processes in the user’s notebook. We define the corresponding region of values of the context attributes for each situation space with different appropriate values. For user location we used Ekahau Positioning Engine [27] that tracks the user’s personal devices such as his/her PDA and notebook. The positioning service computes spatial positions by analyzing wireless signal strengths and comparing them to previous calibration. It provides an associated confidence of its inferred location with a measure between 0 and 1. We use Berkeley Motes [27] for sensing and communicating light levels in the meeting room. For retrieving information about the user’s presentation activity we have implemented a service that hooks to the notebook operating system and provides information on latest keyboard and mouse activity. We also provide a service that identifies active presentation processes in the user notebook. In the meeting room we use a portable light-weight presentation projector that is connected to the presenting user’s notebook. It is a common practice and the assumption in this experiment that each presenter uses his/her
personal notebook.

An agent working on behalf of the user and running on the user notebook is equipped with communication protocols for exchanging information with remote sensing services as well as with local processes that provide information on notebook activity. The agent computes the confidence (Section III-B) in each of the candidate situation spaces, which represent the real-life activities. The various sources of information are depicted in Figure 3.

Before applying reasoning, information is pre-processed either by the agent or remotely, depending on the type of data. For example, light levels are sampled by the motes sensors a number of times and then averaged. The Motes Interface Service, which handles this information, matches the averaged result against predefined light levels and communicates back predefined values. In contrast, interpretation of keyboard and mouse activity is preformed directly by the agent. Here, the amount of time lapsed since the last captured activity influences the confidence that the user is currently using his notebook, and this affects the confidence of containment inside a region of values.

During experimentation we have switched between the activities and observed the results of the confidence measures for the different situations. Figure 4 provides confidence measures obtained for the three activities that were computed for a particular user in two experimental runs. In the experimental run depicted on the left we have started with the user presenting first for 15 minutes, then attending a colleague’s presentation for 15 minutes and finally participating in a discussion or general meeting on the topics presented (again for 15 minutes). On the run depicted on the right, a user has given a presentation and then participated in a general meeting.

Interpretation of the results reveals matching confidence levels with the actual activity taking place. Taking the left experimental run as an example, at the time the user is presenting, the confidence for this particular activity averages around 0.9 whereas confidence in other situations is significantly lower. A change in the situation towards the user only attending another’s presentation results in a drop of the confidence in the ‘User Presenting’ situation to below 0.4 and a rise of confidence in the ‘User attending a presentation’ situation to levels around 0.9. Similarly, when a discussion (equivalent to a meeting) over the presentations involving the user is starting immediately after the second presentation, the confidence in ‘User in a meeting’ situation rises to 0.9 and the former situation confidence levels drop significantly.

B. Merging and partitioning situation spaces

From proving the concept of the reasoning approach we turn to evaluate the impact of applying context algebra operators over situation spaces. We consider the case of agents reasoning about the context of a presentation in the smart room. In the following set of experiments, we simulate a variety of sensors including multiple numbers of light-, noise- and motion- sensors, projector status, speakers and microphone, computed number of users in the room and whether a presentation is scheduled. Data is randomly generated, roughly corresponding to the presentation activity, with associated reading errors. Estimated confidences of sensor accuracies are computed and readings suggesting activities other than a presentation are always generated (e.g., a presentation is occurring but is not scheduled, or performed only verbally without lights turned off).

Different agents have access to different sensors in the room and hold different models about the presentation activity. Some models share common context attributes while others provide descriptions based on a unique set of sensory information. Agents actively merge and partition models to
reach a more reliable reasoning outcome.

By sharing sensor readings and merging different descriptions of the presentation situation, agents enhance their individual reasoning. Figure 5 and 6 present outcomes of reasoning about the presentation activity by agents. Figure 5 depicts results of reasoning with individual models and Figure 6 depicts results of reasoning over the merged model, achieved by collaboration between agents. For comparison, we also average the result of reasoning by individual agents, and compute reasoning results performed over a global model, considering the complete set of sensors. Figure 5 illustrates the significantly different levels of support computed for the same situation, obtained by three different agents, each using a different sub-model with different subsets of information. The averaged result reflects the optimal reasoning outcome by sharing models, given an assumption that all models equally reflect the presentation and cannot be merged into a single global model.

Successful reasoning over the merged model is illustrated in Figure 6, where results obtained over the merged model are identical to averaging the results from each agent individually reasoning about the situation. Reasoning about a global model, yields different but significantly similar results to the merged model. This reflects the fact that all sub-models are more or less equally reflective of the presentation situation.

Benefits of partitioning a model into partial representations of the situation are demonstrated in the next experimentation. By using the partition operator, agents update the model describing a situation according to available sensory data. During experimentation we have simulated faults in communication and sensors, yielding only a partial set of sensory information available for reasoning.

In such conditions agents update their existing model according to the available information. Figure 7 and 8 compare reasoning with the original model with missing information and reasoning with the updated model (using partition operator). For comparison we also compute the optimal result, achieved by reasoning with the original model but with all information available (i.e. no faults).

Results clearly show that outcomes of the partitioned model are significantly closer to the optimal results. (We note that while this approach compensates the lack of available information in modeling, the confidence in the reasoning outcome is reduced since less information is used for inference.)
migrate between these servers and retrieve relevant information. Each server is logically positioned in a different part of the smart room and is linked to a specific set of sensors. Type and position of sensors affect the way situations are modeled, e.g., different kinds of noise sensors exhibit different sensitivity to noise, and readings obtained from light sensors in the back of the room are different than those obtained in the front. Therefore, different perspectives are produced over the same situation depending on the kind of information presented.

A context-aware mobile agent is positioned in a specific agent server and reasons about the situation using available information. The agent deals with information in similar ways to the previous experiment, updating its model whenever sensory information is unavailable. However, when reasoning results about the presentation activity are below specified threshold the agent actively migrates to another server and takes advantage of another perspective (other contextual information available there) over the presentation situation. It then merges these viewpoints to reach a more reliable inference regarding the occurrence of the presentation. Figure 9 presents snapshots from the logs of the agent server stations, capturing the reasoning agent migration, partitioning and merging activities.

![Fig. 9. Agent Servers logs: reasoning agent proactively migrates to gain additional viewpoints](image)

VI. CONCLUSION

In an environment typified by complexity, openness, heterogeneity and context uncertainty, approaches to model and reason about context in a distributed manner are essential. Using our approaches we push reasoning to the agents, providing tools and motivation to perform collaboration and distributed reasoning. We have proposed an approach to merge different perspectives of entities over situations (or objects), compose new situations from more basic situations, and partition models into partial representations to deal with lack of pertinent information.

The proposed operations are based on a novel approach to deal with context, describing context in multi-dimensional space and representing situations as geometrical structures in that space. These modeling and reasoning methods can be applied in a general way to different context scenarios and are further enhanced with context algebra that provide operations over this representation. We have demonstrated and evaluated the proposed operations in different cases and applied our implementation with mobile agents. Thus, we have created context-aware mobile agents, which actively collaborate, migrate, model and reason, partition and merge context descriptions for attaining optimal reasoning and context-awareness.

VII. ACKNOWLEDGEMENTS

We thank HP (Hewlett-Packard) for sponsorship of this work, and HP Labs for productive collaboration

VIII. REFERENCES


