Supporting Context-Centric Relations in Heterogeneous Environments

Jamie Walters, Theo Kanter and Rahim Rahmani

Department of Computer and System Sciences, Stockholm University
Forum 100, Kista, Sweden

Abstract
Massive Immersive Participation is enriched through the use of context information describing the dynamic states and relations among people places and things. This in turn mandates the creation of methods and models for establishing and supporting these relationships. Previous approaches are undermined by their limited interpretation of context centric relations and subsequently do not offer support for multi-criteria relationships. In this paper, we extend on our previous work on establishing multi-criteria context relationships, to adding the support required for maintaining these relationships over heterogeneous and dynamic context information. We introduce a query language that supports an extended publish-subscribe approach and define solutions for dynamically evaluating and adjusting these relationships while minimizing overall costs.

Keywords: *Context-Awareness, Immersive Participation, Context, Context Models, Internet Of Things, Context Proximity, Sensor Information, P2P Context*

1. Introduction

The increase in massive immersive participation scenarios is one part of our trend towards a pervasive computing reality. Such realities range from immersive games such as Google Ingress [1] to theatrical productions such Maryam [2] produced by RATS Theatre [3]. Users are immersed in realities that range from world domination to complete theatrical performances where people places and things are fused together in dynamic participatory environments.

The resulting immersion is enriched through the addition of the underlying context information driving the interactions among the collection of connected things. Supported by an Internet of Things (IoT), this additional information presents itself as the backbone of our pervasive realities, which responds to and accommodates for the establishing of the dynamic relationships that exist between a user, his environment and services. Systems such as SenseWeb [4], IP MultiMedia Subsystem (IMS) [5] , MediaSense [6] and SCOPE [7] were developed in response to this need to provision information supporting immersive realities.

Their limitations with respects to expressivity however limits their suitability in answering the question of *"Who you are, who you are with and what resources are nearby"* as required by Schilit and Adams suggest in [8] and further reiterated and summarized by Dey in [9], who expects that applications and services be provided answers to the question of *[which] entity is considered relevant to the interaction between a user and an application*.

While semantic approaches such that described by Dobslaw et al. in [10], Toninelli et al in [11] and Liu et al. in [12] offer some support towards this problem, Adomavicius et al. in [13] suggested that these types of approaches are limited and should be complemented by metric type approaches thus realizing the ability to answer the question of *"nearness"* as posited by both Schilit et al. in [8] and Dey and [9]. This further characterization would permit us to better identify and establish context relations between related entities. Therefore, establishing the types of relationships shown in Figure 1 is premised on our ability to support the complementing metric-type similarity models which, according to Hong et al. in [14], is critical in realizing applications and services that can discover nearby sensors or points of information.

Supporting these types of relationships is a multifaceted problem involving the identification and selection of candidate entities and managing the subsequent volume of required context information. One approach to this is through the use of centralized presence systems such as described by Petras et al. in [15]. Here an entity watches other entities contained in its address book. While this reduces the volume of context information required to maintain relationships, the resulting relationships are not context centric and limits the *watcher's* ability to discover entities of interest with which to establish common context relationships. With the average address book estimated to be limited in size to *0.005 * population* [15], the alternative of subscribing to all users would not present itself as a feasible solution with the volume of messages per status change would be approximated to *population x population*. This solution would not scale well and simply pruning the message queue as suggested by Petras et al. would offer little guarantee with regards to the quality of the context information.

In defining an *Operational Approach to Context* Zimmermann et al. in [16] describes the notion of proximity as the overarching factor in establishing context relationships. This subsumed the earlier address book approaches and moved towards realizing truly context-centric networks where interactions, discovery and relationships are underpinned by the degrees of relationships between entities over their underlying context information. Zimmermann et al. equated the notion of proximity to spatiality, essentially disregarding the types of higher level relational proximity expected by Hong et al. in [14].

In [17] we defined an approach to establishing context centric relationships between entities on an Internet of Things. Here, relationships are established between entities over the similarity of their underlying context behaviors evaluated over a pre-determined time window. This extends the work of Zimmermann et al. in [16] towards a context-centric model while subsuming it with respects to expressiveness. This satisfies the initial requirement of a context relational model capable of supporting the establishing, adjusting and exploiting of implicit context based relationships in massive immersive environments. With this approach, we are capable of identifying and discovering candidate entities that can be fused to realize new user experiences and deliver more expressive applications and services.

However the problem of support remains as Zimmermann et al. in [16] provided no solution for discovering the candidate entities and establishing relationships in light of the highly dynamic nature of context information. Schmohl partially addressed this in [18] proposing a multi-dimensional hypersphere of interest in which entities entering are deemed to be candidates for the *watcher* and are evaluated and selected according to a proposed proximity measure. Here entities are discovered through the use of multi-dimensional indexing structures such as R-trees, kd-trees and space partitioning grids. These solutions are less optimal for multi-dimensional dynamic context environments as the cost of indexing increases exponentially with a linear increase in the number of dimensions. Queries therefore risk being executed against outdated indexes with no guarantees of information freshness. As a solution to this problem, Schmohl suggested that dimensions could be selectively pruned from the indices. By taking this solution applications depending on less popular dimensions would not stand to realize any benefit from this optimization. Alternatively, accuracy could be sacrificed for speed, which would not offer any guarantee of information accuracy for applications where this is critical.

Yoo et. al in [19] and Santa et. al in [20] proposed the use of publish-subscribe approaches as suitable alternatives with Kanter et. al in [21] showing that such approaches are scalable and can realize dissemination times on par with UDP signally used in SIP implementations. Frey and Roman in [22] extended this approach to provide for event driven subscriptions in context networks, however this is based on events rather than the raw underpinning context information.

Supporting the establishing of context-centric relationships over heterogeneous context information therefore requires new approaches to maximizing the identification of candidate entities while minimizing the overall resource costs

In this paper we introduce an approach to discovering related context entities through the use of an extended publish subscribe module coupled with a context query language. Additionally we introduce solutions for reducing the corresponding resource demands on established relationships. In Section 2 we summarize our approach to establishing context relationships. Section 3 details our approach to supporting these relationships while Section 4 discusses early analysis and results. Section 5 completes with a conclusion and discussion.

2. The Context Relational Model

In [17], we introduced a dynamic heterogeneous approach to context relationships where notion of context proximity is one that considers the situation, attributes, relations, accuracy and heterogeneity of both the underlying information and the vast array of requirements for metrics supporting application domains. We defined context proximity as: *the amount of work required to transform the context behavior of one entity into that of over the characteristics of their current underlying context states*

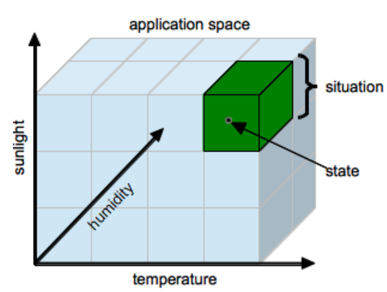

Figure 1 - Context Relational Model

Here, we model context as an n-dimensional domain space; the universe of discourse of a problem domain; the subset of all global information considered relevant and supports the delivery of any application or service relative to this domain.

In an immersive participation environment, such a domain could be the play "*Maryam*". This domain is then partitioned into situations or activities representing an acceptable range of context information defining a real world situation or activity. For the domain *Maryam*, activities could be *Scene 1, Scene 2, Scene 3*, etc. Activity definitions are not mutually exclusive and therefore several activity sub-spaces can overlap.

Each situation in turn contains context states; a combination of unique attribute values within a situation or activity space corresponding to a context observation made on an entity. For the domain *Maryam,* a state could be the context information recorded from body sensors at *Scene 1*.

2.1 Activity Classification

Citing the lack of consideration given to the context activity by existing proximity approaches, an underpinning element of our approach is use of activity information for deriving relational proximity. We identify activities of using the probabilistic approach described by Padovitz in [23].

The activity of an entity P bearing a context state $\forall S^{\mathcal{A}} = \{s_1, s_2, \dots\ s_i\} : s_i \in \mathcal{A}_i$ can be determined by assigning it the activity with the highest confidence calculated as:

$$q_1 \sum_{i=1}^{n} \widehat{w}_1 \,.\, \Pr(a_i^t \in A_i) + q_2 \prod_{k=1}^{m} Pr(a_k^t \in A_k) \qquad 1$$

where $q1 + q2 = 1$. This is discussed in detail in [24]. With this approach, we consider state value membership, information accuracy and the importance of each context attribute to determining an activity. Here, we could observe states in *Maryam* and identify the current activity being experienced by the user.

2.2 Activity Similarity

	stand	walk	sit	lay
stand	1	0.7	0.5	0.25
walk	0.7	1	0.4	0.05
sit	0.5	0.4	1	0.70
lay	0.25	0.05	0.70	1

Figure 2 Activity Similarities

We define a similarity matrix between the activities within an application or domain space.

As shown in Table 1, this is an MxN matrix of real values between 0.0 and 1.0 conveying the similarity between activities in a domain space; the ease with which one activity can be transformed into another. Here an activity is not simply confined to primitive determinations such as walking, running, sitting laying but rather encompasses higher level notions of activities such as *going to work*, *going home*, *shopping*, *watching television*, *washing*, *cooking*.

These higher-level activities are not necessarily discernable from raw context information but can be derived by applying learning methods, human annotation and assumptions. The underlying context information could be very similar or even identical while the perceived higher-level activities are not.

2.3 Relational Context Proximity

Relational proximity is derived between the states of entities as observed over a time window W. For solving this, we used the Earth Movers Distance as described by Rubner et al. in [25] setting the distributions as the sets of observable context states for each window W, the weighted edges being the activity similarity between P and Q and the ground distance d_{ij} being the distance between pairs of states s_i, s_j derivable as:

$$d_{ij}(s_i, s_j) = \frac{\left(\sum_{k=1}^{n} \left[w_a * \left|\mathcal{F}_a^{\mathcal{D}}(a_i, a_j)\right|^r\right]_k\right)^{\frac{1}{r}}}{\left(\sum_{k=1}^{n} \left[w_a * \left|\mathcal{F}_a^{\mathcal{D}}(a_i, a_j)_{max}\right|^r\right]_k\right)^{\frac{1}{r}}} \qquad 2$$

where $a_i \in \mathcal{A}_i^{\mathcal{D}}, a_j \in \mathcal{A}_j^{\mathcal{D}}$

Here, w is the weighting for each attribute. The value of r can be adjusted to reflect the perceived distance between P and Q as shown by Shahid et al. in [26]. The distance is normalized with respects to the maximum distance between states in the encompassing application space. Our measure of proximity therefore logically subsumes and extends existing $Lp - norm$ approaches.

The EMD algorithm is then applied to derive the largest possible transformation between P and Q that minimizes the overall context transformation cost, where:

$$WORK(P \rightarrow Q, \boldsymbol{F}) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij} \qquad 3$$

Subjected to the following constraints:

$$\begin{aligned}
&1.\quad f_{ij} \geq 0 \quad 1 \leq i \leq m,\ 1 \leq j \leq n \\
&2.\quad \textstyle\sum_{i=1}^{m} f_{ij} \leq P \quad 1 \leq i \leq m \\
&3.\quad \textstyle\sum_{j=1}^{n} f_{ij} \leq Q \quad 1 \leq j \leq n \\
&4.\quad \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = min\left(\sum_{i=1}^{m} P, \sum_{j=1}^{n} Q\right)
\end{aligned} \qquad 4$$

The first constraint permits the transformation and hence the proximity from $P \rightarrow Q$ and not the opposite. The second and third constraints limit the transformation $P \rightarrow Q$ to the maximum number of context observations made for P or Q. The final constraint forces the maximum transformation possible between both entities. The context proximity, $\delta_{(P,Q)}$, is the earthmover's distance normalized by the total flow.

$$\delta_{(P,Q)} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}\right) * \left(\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}\right)^{-1} \qquad 5$$

We using the maximum possible flow between P and Q. It is important to note, that $\delta_{(P,Q)}$ is indifferent to the size of both sets of observations and permits partial similarity where the behavior of P is subsumed by the behaviour of Q. Therefore $\delta_{(P,Q)} \mid w = \delta_{(P,Q)} \mid \frac{1}{2}\, w$. This is a distinct advantage of our approach and excess observations are inherently discarded.

However, where partial matching is desirable and the completeness of containment is important for relations such that $P \cap Q = P \cup Q$, we extend the proximity measure to be normalized relative to the maximum potential transformation of either P or Q, such that

$$\delta_{(P,Q)} = \left[\left(\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}\right) * \left(\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\right)^{-1}\right] \quad 6$$

$$\left|\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = max\left(\sum_{i=1}^{m} P, \sum_{j=1}^{n} Q\right)\right.$$

The Confidence Constraint

In order to consider scenarios over unreliable context information, we can adjust the distance d_{ij} to reflect the potential errors in the underlying context information such that:

$$d_{ij} = d_{ij} * \left[c_{ij} + \left((1 - c_{ij}) * (1 - k)\right)\right]$$

$$where\ c_i = \sum_{i=1}^{n} \widehat{w}_1 \,.\, \Pr(a_i^t \in A_i) \quad 7$$

This confidence measure is described by Padovitz et al. and considers the accuracy of the sensors using several factors described in [24]. However for scenarios where the confidence is a trade off, we, we add the confidence factor k, which allows us to adjust this trade-off.

The Temporal Constraint

For calculating proximity considering the time constraint, we can adjust the size of the observation window W. For clarity:

$$\lim_{W \to 1} EMD\,(P,Q) = Lp_{norm}(P,Q) \quad 8$$

By adjusting W we permit wider variations in the time differences between state observations reducing the time constraints. Increasing W increases the constraint on the nearness of observations with respects to their temporal attribute.

The Continuity Constraints

Furthermore, we derive the measure of proximity stability between two entities as a means of filtering entities based on the stability of potential relationships. The first constraint finds the standard deviation of $\delta_{(P,Q)}$ as the window W progresses. We call this the co-relational constraint defined as:

$$R_{(P,Q)} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\delta_{(P,Q)_i} - \mu\right)^2}$$

$$and\ \ \mu = \frac{1}{N}\sum_{i}^{N} \delta_{(P,Q)_i} \quad 9$$

Where the greater the deviation the more unstable the relationship between is. Secondly, we derive the convergence factor between two entities; the rate at which their context proximity is converging defined as:

$$Cf_{(P,Q)} = \frac{\Delta\delta_{(P,Q)}}{\Delta\, W} \quad 10$$

With this factor we can consider entities that are diverging or moving apart or entities that are getting closer or merging over time. Having established context relationships of the types described we are subsequently required to maintain these relationships at a minimal cost overhead.

3. Supporting the Context Relational Model

In Section 2.3, we described an approach to establishing context relationships over a proximity function defining the relationship between the behaviors of two given entities. Supporting this type of relationship requires an approach to finding, establishing and maintaining the relationships between the entities satisfying the proximity function. Figure 3 illustrates this process of supporting relationships over relational proximity.

Firstly, we created a query language for defining the proximity; the bounds of the hypercube of interest. The query is then executed across a distributed heterogeneous data store with the candidate entities selected for establishing a relationship. The resulting relationships are maintained by subscribing to the entities of interest and continually evaluating the relationships with each derived context state ranking each entity by its current proximity. New entities are continually added while non-relevant entities are consequently pruned.

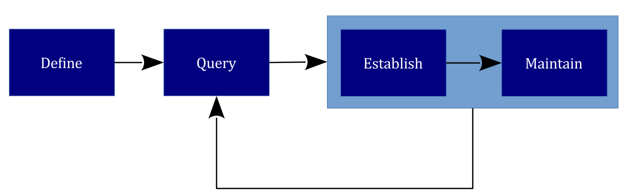

Figure 3 Supporting Relational Proximity

Finding related entities requires that the underpinning context information is readily stored and accessible and is organized in such that it permits discovery in a scalable manner with respects to real time qualities. While centralized approaches may be supported by databases, they are inhibited by their inability to scale well in response to service demands and availability. In response to this, we proposed a distributed approach to organizing context information using a self organizing peer-peer protocol [27].

3.1 Distributed Organization

At the base of our approach is the existence of a distributed data store capable of locating context information in response to the interest of applications and services. A fundamental point of departure from similar approaches is that this distributed data layer is capable of responding to queries on ranges of data and capable of answering queries with a range. Our solution is detailed in [27]and consists of an organized peer-to-peer data store with a lookup complexity of $0.5 \log n$, and a protocol for persisting, locating and subscribing to entities of interest. This permits us to find entities based on an area of interest by defining range of context values encompassing this area of interest. However defining areas of interest requires more expressive means of expressing context based queries than the primitive constructs of the associated protocol defined in [28].

3.2 Relational Context Query

In response to this we introduce a declarative query language, the Context Proximity Query Language (CPQL), for defining an area of interest relative to an entity. This is a natural extension of the interest based approach we introduced in [21], however with the interest area defined as a complex distance function over any underlying context information.
Similar to type and structure to SQL, the query language sits at the core of our query functionality and has two main constructs: *GET* or *SUB*. A *GET* is similar to an *SQL-SELECT* and retrieves all the states that currently match the defined proximity function. This can be used for a single evaluation for finding candidate entities. A *SUB* is also the equivalent to the *SQL-SELECT* with the addition that its function is to supporting existing established relationships between the entities bearing the satisfying states through an extension of publish-subscribe approach as described by Kanter et.al in [6]. Each application space defines function modules that satisfy $\mathcal{F}_a^{\mathcal{D}}(a_i, a_j)$ in Equation 2. Function modules are persisted on the distributed overlay architecture described in [6] and are identifiable, and retrievable by all nodes within the distributed data space.

A CPQL query is the defined of the type:

GET|SUB PRESENTITY
WHERE DISTANCE $DIST\_NAME$ $<$ 0.25
[ORDER ASC|DESC]
DEFINING DIST_NAME
AS sqrt(pow($\mathcal{F}_{lat}(P_{lat}, Q_{lat})$, 2) + pow($\mathcal{F}_{lon}(P_{lon}, Q_{lon})$, 2)); 11

For modularity and re-use, proximity functions can be defined prior to usage be saved as:

DEFINING DIST_NAME
AS sqrt(pow($\mathcal{F}_{lat}(P_{lat}, Q_{lat})$, 2) + pow($\mathcal{F}_{lon}(P_{lon}, Q_{lon})$, 2)); 12

Successive queries reference the defined function as:

GET|SUB PRESENTITY
WHERE DISTANCE $DIST\_NAME$ $<$ 0.25
[ORDER ASC|DESC] 13

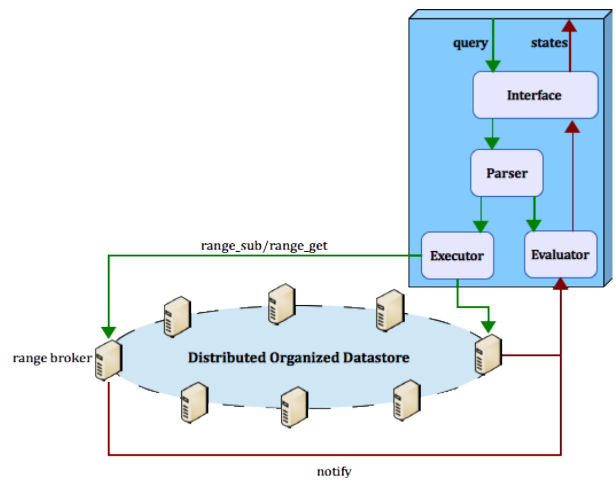

Figure 4 Relational Proximity Query-Subscribe Model

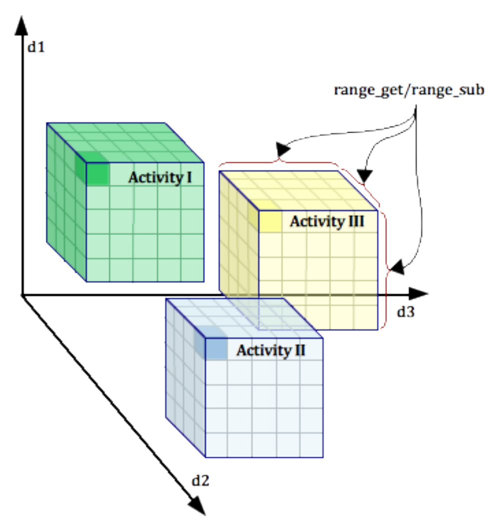

Figure 5 Range Query Subscribe

This permits us to share query definitions in a global data store that can be found and re-used by applications and services. Queries are accepted by an interface layer, which also provides the support for defining proximity definition functions. Each query is parsed and analyzed for correctness and decomposed into its corresponding parse tree. The resulting tree is passed to an executor for execution across the distributed data store. The proximity function as described in Equation 9 uses $(\boldsymbol{a_{min}}, \boldsymbol{a_{max}})$ as the upper and lower boundaries for each dimension of the application space and are used as the limits for the *range_sub* and *range_get* as shown in Figure 5. A range get for any proximity function is then expressed as: range_get$(\boldsymbol{a_{min}}, \boldsymbol{a_{max}})$. This limits the context states and entities that are queried to only those being relevant to and entity for the execution of a specific application or service.

3.3 Relational Publish Subscribe

Establishing relationships requires a new approach towards publish-subscribe solutions in context aware systems. Previous publish subscribe solutions such as that detailed by Kanter et. al in [6] supported primitives for getting or subscribing to the context information of an entity. Unlike Petras et al. in [15] such a solution was distributed reducing issues of scalability; however like Petras et al. a watcher cannot subscribe to greater than the size of its address book. This is estimated in [15] to be around 0.005*N,* the number of global presentities. However, in context centric approaches, a watcher's address book does not determine the number of presentity it watches. This should be determinable by the number of entities with which it can potentially establish a relationship. The absolute maximum number of subscriptions for each watcher would therefore be *N* which is not scalable.

To address this shortcoming and provide scalable support for context centric networks, we extend the publish-subscribe approaches to enable subscriptions to relationships and areas of interest as defined by an underlying proximity function. A *watcher* issues a subscription for all (a_{min}, a_{max}) of the underlying application space. This is issued to *range_brokers*, distributed nodes responsible for brokering ranges of values between *watchers* and their *presentities*. The *range_sub* or *range_get* is routed by the underlying support to the node or nodes in the data space shown in Figure 4 that are responsible for the range of values in (a_{min}, a_{max})**.** Each *presentity* each presentity subsequently publishes its current context value to the corresponding *range_broker*. The *range_broker* in turn sends the current set of states to the watcher. The *watcher* evaluates the list of states over its proximity function and establishes a context relationship with selected *presentities*. In order to

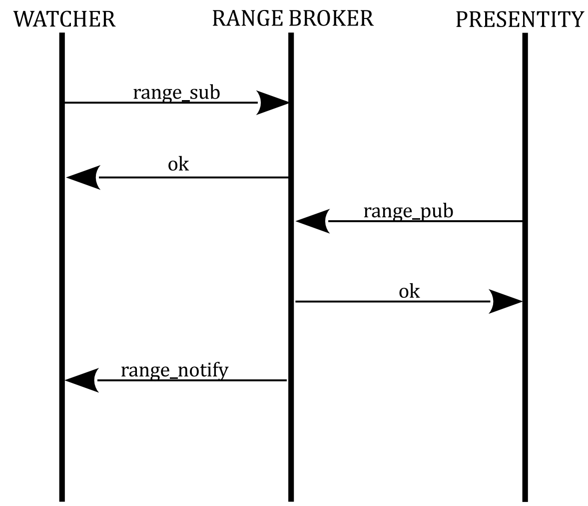

Figure 6 Range Publish-Subscribe Messaging

maintain the relationship, the *watcher* then subscribes to the *presentities* and continually evaluates the relationship with each new context states received. The *watcher* also maintains a subscription with the range*_broker*, which continually send lists of context states matching the range subscription to the watcher. Context entities publish new context states as the supporting context information changes; this can range from very frequently in highly dynamic situations to seldom in lesser dynamic situations. Petras et al. in [15] stated that the number of messages originating with each state change of a presentity is equal to the number of watchers for that presentity. In prescence systems without context information this can be taken as the size of the presentity's address book. However, in context centric approaches, the number of watchers for each presentity would have no relation to the size of such an address book.

Therefore, given that the known context universe contains *D* dimensions with *N* uniformly distributed entities. Each application *A* is enclosed by an application space with *M* dimensions such that $M \in \binom{D}{k}$ where $0 < k < D$. With existing publish-subscribe approaches the cost of subscribing to all entities is of the order $(N-1)$ for each entity and a global subscription cost of $(N^2 - N)$ for all known entities.

We however reduce the number of subscriptions, *n*, per entity by defining the number of potential candidate entities for each *watcher*. Given the universe of discourse of the application *A,* where each dimension *d* occurs with a probability distribution θ, the number of subscriptions, we limit the number of potential candidates to:

$$N \prod_{d \in M} P(d|\theta_d) \quad 14$$

and the overall number of subscriptions for all entities contained in each application is:

$$\sum_{n \epsilon A} \left(N \prod_{d \in M} P(d|\theta_d) \right) \quad 15$$

That is to say we do not allow subscriptions to entities without a probability of being a candidate entity i.e, entities without the required context dimensions. Additionally, given that an application space is further limited by the dimensions having values between (a_{min}, a_{max}), we further limit this to:

$$N \prod_{d \in M} P(d|\theta_d) * P(\, a_{min} < d_{val} < a_{max}|\theta_{dval}) \quad 16$$

for each entity within the application space and:

$$\sum_{n \epsilon A} \left(N \prod_{d \in M} P(d|\theta_d) * P(\, a_{min} < d_{val} \right. \quad 17$$

for all entities contained within each application. This is achieved through the use of the relational publish-subscribe approach. From this we derive the following proposition:

Proposition 1:
The number of presentities required for an application or service is limited by the universe of discourse of the application or service itself.

After selecting the candidate entities and establishing a relationship, we are now required to maximize the quality of the context relationship while minimizing the message overheads required for support. The maximum quality achievable by a context relationship is calculating with every change in the underlying context states. Current approaches to deriving context proximity between two entities calculates a notion of proximity either in response to continual changes in context information or on demand by the utilizing application or service. Therefore, the current state of the context relationship does not influence the flow of context information between two entities with an established relationship. From this we derive:

Proposition 2:
The cost of maintaining the context relationship between two given entities is determined by the state of the context relationship itself.

Given two entities P and Q, The cost of maintaining the relationship between any two entities $P \rightarrow Q$, is related to the current state. When the two entities are in close proximity the cost of maintaining their relationship is higher and decreases as the proximity between both entities approaches 1. Therefore with respects to proximity the $interval$ between refreshes of the underlying information is:

$$\begin{aligned} &\lim_{\delta \to 0} interval\,(P,Q) = \; interval_{\,min}(P,Q) \\ &\lim_{\delta \to 1} interval\,(P,Q) = \; interval_{\,max}(P,Q) \end{aligned} \quad 18$$

We extend the publish-subscribe module to additionally accept a parameter for the *interval* delays between publishing messages to entities with an existing relationship. Entities, according to the model described in Section 2 generate states in response to changes in the underlying context information. Each change in the value of a context attribute generates a new context state. Each relationship has three values: $i_{\,min}$, the native interval between states for the entity Q, $i_{\,min}$, the maximum interval permitted by the application before the states must be refreshed and $i_{\,cur}$ the current interval between context states calculated as:

$$i_{\,curr} = \begin{cases} i_{\,min} + \; \delta C f_{(P,Q)} * (i_{\,max} - i_{\,min}), \;\; , & f \neq 0 \\ i_{\,min} + \; \delta * (i_{\,max} - i_{\,min}), & f = 0 \end{cases} \quad 19$$

The interval value is therefore calculated for each publish instance based on the last known proximity and the rate at which the proximity moves towards 0. This is an adaptive algorithm where the rate is kept relative to the known relationship quality. The intuition being that the closer two entities are the more resources that can be expended on maintaining their relationship, while distant entities require less updates and can make resources available to more critical relationships. The rate is adjusted based on the current rate of state generation, therefore we will not request updates faster than they are produced and not slower than the minimum required for the application's quality of experience. The closer the value of $i_{\,\mathrm{min}}$ is relative to $i_{\,max}$, the smaller the penalty. The intuition here being that, as $i_{\,min} \rightarrow i_{\,max}$, the back off potential gets smaller.

With the frequency of refreshes derived, we are now required to adjust the volume of context information used for maintaining the context relationships. Existing publish-subscribe modules transmit each updated element of context information to end points with no processing. Transmitting context information, which, in some cases offers no significant knowledge or variation in the perceived proximity between the entities. The context proximity model described in Section 3 creates a new state with every change in context information. The result being that context information such as GPS sensors, which might change continually would provide a continuous stream of information even in cases where the difference between states is marginal and does not serve to noticeably influence the resulting proximity values.

Firstly, we estimate the cohesion of points in I by randomly selecting a sample of the relationships between all the observed states. We take a sample set to avoid computing all the relationships within I, which is of order of complexity $O\ (n^2)$. The size of the sample set is determined by taking the normal approximation to the hypergeometric distribution for

$$n = \frac{N\, z^2 pq}{(E^2(N-1) + z^2\, pq)}$$

$$\begin{aligned} &n = required\ sample\ size \\ &N = I^2 \\ &p, q = set\ to\ 0.5 \\ &z = confidence\ level,\ set\ to\ 1.96 \\ &E = accuracy,\ set\ to\ 0.03 \\ &0 < n < I \end{aligned} \qquad 20$$

This, as the hypergrometric approximation is more suitable for the relatively small numbers of states observed. We then sample n distances between states using the distance function described in Section 3.4, setting all the weights to 0.

With n selected distances, we then compute the cohesion as the standard deviation of the distances between states in I

$$cohesion = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}((s_1 - s_2)_i - \mu)^2} \qquad 21$$
$$where\ \ \mu = \frac{1}{N}\sum_{i}^{N}(s_1 - s_2)_i\ \ and\ N = I^2$$

The final set of states transmitted for each observation window is therefore:

$$cohesion * I \qquad 22$$

Here, the intuition is that a relatively stable entity expressing little change in its state over time would share less context information while an active entity would share almost all the context information generated. We therefore share as much information as is required for establishing and maintain the context relationships over time, deriving the following proposition:

Proposition 3:
The number of context states shared by an entity with each publish instance is a factor of the cohesion of the total set of observed context states over the observation interval

4. Evaluation

For simulation purposes, we assumed a global population N, of between 1000 and 100000 presentities each assigned a random context profile from a context universe D with 20 dimensions. The application A was assigned three random dimensions (d_a, d_b, d_c) such that $d\ \in\ D$.

The number of presentities with a dimension d was taken as a random value sampled from a binomial distribution $B\ (p/10, N)$ where the value of p is randomly sampled from the Gaussian distribution $\mathcal{N}\ (5.0, 1.6)$. The simulation was run 20 times for each network size and the results shown below in Figure 7. We show the min, avg and max for each network size.

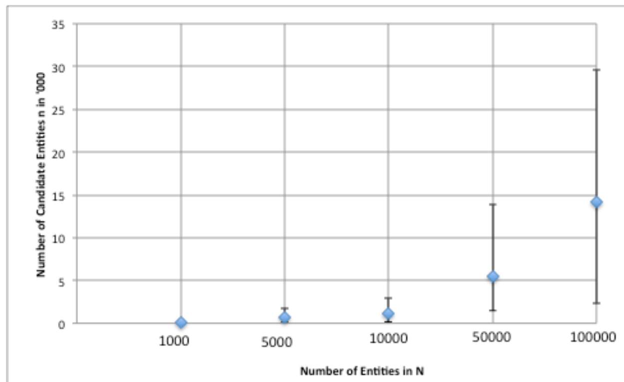

Figure 7 Candidate Entities

As can be seen from Figure 8 the number of potential subscriptions is reduced where subscriptions are made relative to the application's universe of discourse. The number of candidate entities vary widely for each simulation size demonstrating that a priori information such as address books cannot be used to determine the size number of subscriptions required by an application and is best supported by real time approaches such as publish-subscribe. As can be seen from Figure R.x the number of

N	Expected	Pub/Sub	Addressbook
1000	35	36	5
5000	177	186	25
10000	1485	1535	50
50000	2793	2700	250
100000	7611	7691	500

Table 1 Candidate Entities Comparison

Figure 8 shows the comparison between the number of subscriptions that would be issued for the publish subscribe and the addressbook approaches compared with the expected number of subscriptions based on the distribution of the underlying context dimensions. As shown, the publish-subscribe approach of locating entities based on their probability of being candidate maximizes the number of candidate entities while remains significantly less than a full search or subscription to every entity and avoids any expensive indexing approaches.

As shown in Figure 8, where entities are using applications and services with high duplication with respect to the attributes used for each application space, the number of subscriptions relative to applications increases at a lower rate. However, where there is little duplication among entities, the number of subscriptions quickly approaches N(100000).

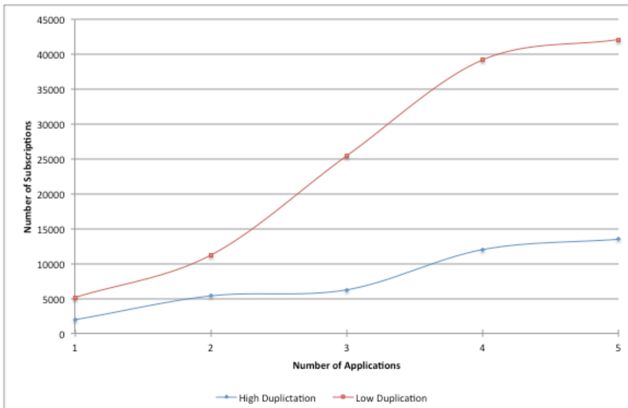

Figure 8 Subscription Increase With Applications

By applying the adaptive ate algorithm we further reduced the number of messages between entities as their proximity approached 1.0 and increased as the proximity approached 0.0. For this simulation we assumed that the distance between an entity is a normal distribution of proximity values between 0.0 and 1.0. Each entity updates its context information using a rate sampled from the Poisson distribution:

$$e^{-\lambda}\frac{\lambda^x}{x!}$$

Where λ is sampled from a set of uniformly distributed values between 1 and 30. This was simulated and the results shown in Table

Rate(msgs/min)	Adaptive Rate(msgs/min)
100184	29438
99799	29426
100159	29314
100052	29175

Table 2 - Adaptive Publish

The adaptive rate, reduced the frequency of updates for each period. This further reduces the overall messaging overhead needed to maintain the types of context relationships described in Section 2.

5. Conclusion

In this paper, we presented an approach to supporting context centric relationships between entities on an Internet of Things. This satisfies the requirement of solutions capable of supporting the types of dynamic relationships that exist over heterogeneous context information. With this approach, we are capable of identifying candidate entities that can be fused to realize new user experiences and deliver more expressive applications and services.

Firstly we proposed a context proximity query language for identifying candidate entities over a distributed data store supporting range queries. The queries are executed across a distributed heterogeneous data store with the candidate entities selected for establishing a relationship.
In order to address the shortcomings of previous approaches to context proximity and provide scalable support for context centric networks, we extended earlier publish-subscribe approaches to enable subscriptions to relationships and areas of interest as defined by the underlying proximity function. The resulting relationships are continually evaluated relationships with each derived context state, ranking each entity by its current proximity. As a result, new entities are continually added while non-relevant entities are consequently pruned. We further introduced an adaptive algorithm for adjusting the publish rate between two entities over the current state of their context relationship and adjust the volume of context information relative to cohesion of the observed context states.

We performed simulations to show that our approach allows us to maximize the number of candidate entities while reducing overall resource costs. We further showed that a priori solutions such as addressbook-based subscriptions do not provide sufficient support by subscribing to significantly less than the relevant number of entities. By avoiding high dimensional indexing and extending proven publish-subscribe approaches we benefit from its proven real time properties. Our solution scales well where applications are using common sets of context information, however where this is not the case, the number of subscriptions converges to N.

Future work includes the in network aggregation and evaluation of proximity as well as the derivation of activity similarity through crowd sourcing.

References

[1] "Ingress." [Online]. Available: http://www.ingress.com/. [Accessed: 26-Feb-2013].

[2] R. Forsberg and W. Sauter, "Digital theater in favor of audiences?," Stockholm, Sweden.

[3] "Kista Teater." [Online]. Available: http://www.kistateater.com/. [Accessed: 26-Feb-2013].

[4] A. Kansal, S. Nath, J. Liu, and F. Zhao, "Senseweb: An infrastructure for shared sensing," *IEEE MultiMedia*, vol. 14, no. 4, pp. 8–13, Oct. 2007.

[5] M. Strohbach, J. Vercher, and M. Bauer, "A case for IMS," *Vehicular Technology Magazine, IEEE*, vol. 4, no. 1, pp. 57–64, Mar. 2009.

[6] T. Kanter, P. Österberg, J. Walters, V. Kardeby, S. Forsström, and S. Pettersson, "The Mediasense Framework," in *2009 Fourth International Conference on Digital Telecommunications*, 2009, pp. 144–147.

[7] R. A. Baloch and N. Crespi, "Addressing context dependency using profile context in overlay networks," in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, 2010, pp. 1–5.

[8] B. Schilit and N. Adams, "Context-aware computing applications," *Systems and Applications, 1994.*, pp. 85–90, 1994.

[9] A. K. Dey, "Understanding and Using Context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, Feb. 2001.

[10] F. Dobslaw, A. Larsson, T. Kanter, and J. Walters, "An Object-Oriented Model in Support of Context-Aware Mobile Applications," *Mobile Wireless Middleware, Operating Systems, and Applications*, pp. 205–220, 2010.

[11] A. Toninelli, S. Pantsar-Syväniemi, P. Bellavista, and E. Ovaska, *Supporting context awareness in smart environments*. New York, New York, USA: ACM Press, 2009, p. 1.

[12] L. Liu, F. Lecue, N. Mehandjiev, and L. Xu, "Using Context Similarity for Service Recommendation," *2010 IEEE Fourth International Conference on Semantic Computing*, pp. 277–284, Sep. 2010.

[13] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems (TOIS)*, vol. 23, no. 1, pp. 103–145, 2005.

[14] J. I. Hong and J. Landay, "An infrastructure approach to context-aware computing," *Human-Computer Interaction*, vol. 16, no. 2, pp. 287–303, Dec. 2001.

[15] D. Petras, I. Baronak, and E. Chromy, "Presence Service in IMS.," *TheScientificWorldJournal*, vol. 2013, p. 606790, Jan. 2013.

[16] A. Zimmermann, A. Lorenz, and R. Oppermann, "An operational definition of context," in *Proceedings of the 6th international and interdisciplinary conference on Modeling and using context*, 2007, pp. 558–571.

[17] J. Walters, T. Kanter, and R. Rahmani, "Establishing Multi-Criteria Context Relations Supporting Ubiquitous Immersive Participation," *International Journal*, vol. 4, no. 2, pp. 59–78, 2013.

[18] R. Schmohl, "The Contextual Map," *deposit.ddb.de*, 2010.

[19] S. Yoo, J. H. Son, and M. H. Kim, "A scalable publish/subscribe system for large mobile ad hoc networks," *Journal of Systems and Software*, vol. 82, no. 7, pp. 1152–1162, Jul. 2009.

[20] J. Santa and A. F. Gomez-Skarmeta, "Sharing Context-Aware Road and Safety Information," *IEEE Pervasive Computing*, vol. 8, no. 3, pp. 58–65, Jul. 2009.

[21] S. Forsström, V. Kardeby, J. Walters, and T. Kanter, "Location-Based Ubiquitous Context Exchange in Mobile Environments," *Mobile Networks and Management*, pp. 177–187, 2011.

[22] D. Frey and G. C. Roman, "Context-aware publish subscribe in mobile ad hoc networks," in *Coordination Models and Languages*, 2007, vol. 4467, pp. 37–55.

[23] A. Padovitz, S. W. Loke, and A. Zaslavsky, "Towards a theory of context spaces," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, 2004, no. March, pp. 38–42.

[24] A. Padovitz, "Context Management and Reasoning about Situations in Pervasive Computing," Monash University, 2006.

[25] Y. Rubner, C. Tomasi, and L. J. Guibas, "A Metric for Distributions with Applications to Image Databases," p. 59, Jan. 1998.

[26] R. Shahid, S. Bertazzon, M. L. Knudtson, and W. a Ghali, "Comparison of distance measures in spatial analytical modeling for health service planning.," *BMC health services research*, vol. 9, p. 200, Jan. 2009.

[27] J. Walters, T. Kanter, and E. Savioli, "A Distributed Framework for Organizing an Internet of Things," in *The 3rd International ICST Conference on Mobile Lightweight Wireless Systems*, 2011, pp. 1–17.

[28] O. Akan, P. Bellavista, J. Cao, F. Dressler, D. Ferrari, M. Gerla, H. Kobayashi, S. Palazzo, S. Sahni, X. (Sherman) Shen, M. Stan, J. Xiaohua, A. Zomaya, G. Coulson, K. Pentikousis, O. Blume, R. Agüero Calvo, S. Papavassiliou, S. Forsström, V. Kardeby, J. Walters, R. Norling, and T. Kanter, *Mobile Networks and Management*, vol. 32. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 57–66–66.