Department of Computer Science
George Mason University
Technical Report Series

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu.edu/
703-993-1530

Towards Generalized Trust in Smart Spaces

Dalal Ahmed Al-Arayed dalaraye@gmu.edu
João P. Sousa jpsousa@cs.gmu.edu

Technical Report GMU-CS-TR-2009-8

Abstract

Existing work in trust management does not satisfactorily provide a combination of functionality and generality. A model that sufficiently addresses the aspects of trust management and is general enough to be applicable in multiple problem scenarios is needed. This paper presents a base trust model that can be used in multiple problem scenarios (content management, service provision, and routing). In addition, smart spaces introduce new issues that are not addressed sufficiently by the available trust management models. The base trust model is designed to be simple to facilitate its enrichment to accommodate the additional requirements of Smart Spaces in the future.

1 Introduction

Today the area of Trust Management is nowhere near maturity. Current solutions, discussed in Section 3, either provide high-level frameworks with little functionality, or functionality in the form of a point solution targeting a specific problem scenario. Most models deal with ad-hoc mobile P2P networks and handle simple relationships between peers. Existing work does not satisfactorily provide a functional model that sufficiently addresses the aspects of trust management and is general enough to be applicable in multiple problem scenarios. In addition, Smart Spaces introduce new issues that are not addressed sufficiently by the available point solutions.

Current point solutions targeting trust management in Smart Spaces deal with only part of the aspects of trust management. Specifically, there is no model that satisfactorily integrates a decision making model and the essentials of trust and privacy management like trust formation, dissemination, evolution, dispositional trust, detection and isolation of misbehavior, dynamic bootstrapping of new entities, handling multiple classes of participants and storage of trust data. Our ultimate goal is to extend trust management to address smart spaces. This involves dividing the participants into classes and implementing a multi-level decision model. Furthermore, privacy policies and notions of locality would enable personalized fine-grained trust decisions. For example, using privacy policies facilitated by the use of classes would enable stereo-typing, which allows peers to select certain classes or alternatively eliminate certain classes to apply the trust model on. Locality would mean that users can also select to apply the trust model only on a subset of peers that are located in a certain area.

This report presents a trust model designed as a prelude to the design of a functional trust model targeting smart spaces. Hence, the base model is designed to be simple to make it easier to enrich in the future to be able to handle more complex problem scenarios, namely smart space scenarios. This base model is applicable in three problem scenarios: content management, service provision, and routing.

In Section 5, we look at smart space scenarios and analyze them to come up with different options for the enrichment of the base model to be able to satisfy the needs of smart spaces. A trust model for smart spaces must be multi-lateral enabling complex relationships that involve multiple classes of diverse participants. Also, a general model that is applicable across multiple problem domains is necessary for wider acceptance and adoption of trust management. Therefore, starting with a base model that applicable in multiple problem scenarios and then proceeding to enhance it for applicability to smart spaces would make our goal of achieving generality in final model easier.

Smart Spaces are infrastructures that incorporate mobile devices, sensors, and networks that sense ongoing human activities and respond to them. There are numerous examples of research in smart environments including homes [1] [2], labs [3], workspaces [4], hospitals [5] [6] [7]., classrooms[8], museums [9], etc.

A base trust model applicable in multiple problem scenarios is presented in Section 2. The base trust model is applied to three problem scenarios: service provision, content management and routing. Scenario 3 presents current work in trust management. Section 4 discusses the base trust model justifies the need for it. The requirements for extending the base trust model to satisfy smart space scenarios are presented in Section 5 using a sample of smart space scenarios. Finally, the conclusion and future work is presented in Section 6.

2 Synthesis of Base Trust Model

The base trust model is synthesized from existing trust models. Three trust models that cover diverse scenarios are selected in order for the derived model to be applicable across a broad spectrum of problem scenarios. The synthesis is conducted by combining elements from the three models and abstracting functional details that enable applicability across diverse scenarios. The result is a model that is applicable across the scenarios addressed by the original three trust models. The reason we take this approach is that our ultimate goal is to produce a generalized trust model for smart spaces that is rich enough to handle the needs of smart spaces and yet is applicable in multiple problem scenarios. Hence, it makes sense to start with a general model and then enhance it to be applicable in smart spaces.

2.1 Problem Scenarios of Models selected from Literature

This section lists the three trust models chosen from the literature from the service provision problem domain, the content management problem domain and the routing problem domain. They provide the diverse problem scenarios that the base trust model needs to apply on.

2.1.1 Service Provision Domain

Reference [10] provides a trust model for the selection of service providers in an ad-hoc peer-to-peer setting. In this Scenario, demonstrated in **Figure 1**, Peer 4 is requesting Service A. Peer 1, Peer 2 and Peer 4 offer to provide Peer 4 with Service A. Now Peer 4 needs to select one of the service providers based on its own past experience and based on recommendations from other 'trustworthy' peers.

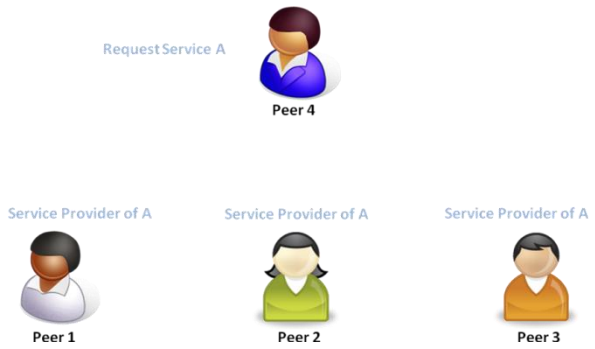

Figure 1: Service Provision Problem Domain

Content Management Problem Domain- "Trullo- local trust bootstrapping for ubiquitous devices," by Daniele Quercia, Stephan Hailes, and Licia Capra [11].

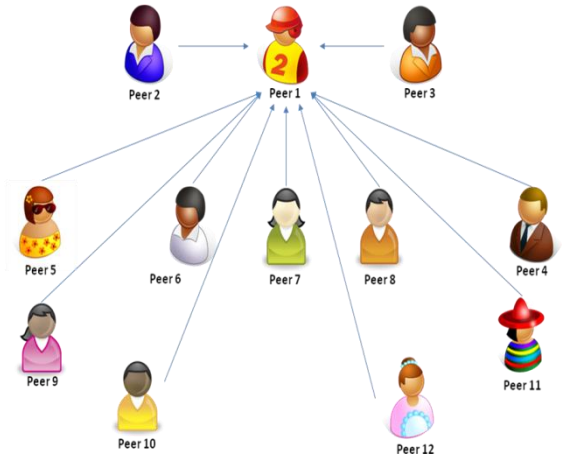

Figure 2: Content Management Problem Domain

Reference [11] provides an example of the content management problem domain. This Scenario describes a situation where a mobile phone user enters an environment where situated advertising is offered like a mall or a fair. The mobile phone user is bombarded with advertisements from numerous providers in numerous contexts (types of product or service provided in the ads) and needs to sort through them and select which ones to accept or view. This could be looked at as different classes of providers (shoe stores, restaurants, entertainments, department stores, family, cinemas, etc. For example, a peer might be interested in buying shoes and only want to view 10 ads but receives 100+ ads. A trust model is used to select 10 ads that are likely to be most interesting and relevant for this particular peer. Reference [11] uses only the peer's direct experience for the trust decision. Recommendations from other peers are not used in the decision making process.

2.1.2 Packet Routing Problem Domain

Reference [12] presents a scenario where nodes route packets to their destination node through other nodes. For example, Peer 1 needs to send a packet to Peer 4 by routing it through other peers. Hence, Peer 1 needs to select among next-hop peers using trust to select the peer to route the packets through.

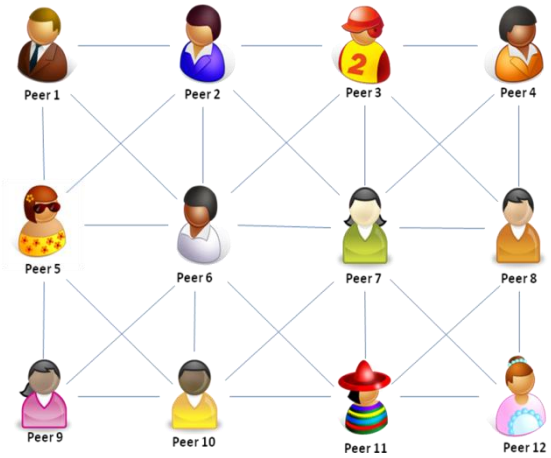

Figure 3: Routing Problem Domain

2.2 Base Trust Model

In this section the base trust model is presented using an example of a service provision problem scenario. Apply the base trust model to content management and routing problem scenarios is discussed in Section 2.3.

First, storage of trust data is explained in Section 2.2.1. Next the three main trust management functions are explained using an example service provision problem scenario: trust formation in Section 2.2.2, trust dissemination in Section 2.2.3, and trust evolution in Section 2.2.4.

2.2.1 Storage of Trust Data

Each peer stores locally two tables of trust data. One table represents direct trust in peers as service providers, and another table represents trust in peers as recommenders of other peers. It is assumed that all nodes have a unique identifier and nodes cannot impersonate other nodes.

2.2.1.1 Service Provider Trust

Each entity will store a table representing its own trust in other entities as service providers in a specific context. TABLE I presents an example of this table. This table stores trust in nodes as service providers. It is updated by direct experience as explained in Section 2.2.4.

Trust is a value between 0 and 1, 0 representing total distrust and 1 representing complete trust. Knowledge represents how well the node is known to the decision maker, and is basically a measure of the number of interactions that occurred directly between them. The timestamp is used to decide how valid the knowledge is, and it is basically a record of when the last time direct experience occurred. The date column is not stored but is only shown here for the reader to know what date the timestamp refers to.

TABLE I : Trust in Service Providers

provider	trust	context	knowledge	timestamp	date
Bob	0.9	Service A	0.9	40057	9/1/2009
Chris	0.8	Service A	0.8	40026	8/1/2009
David	0.7	Service A	0.2	39995	7/1/2009
Emily	0.9	Service A	0.3	39965	6/1/2009
Frank	0.8	Service A	0.6	39934	5/1/2009
Greg	0.7	Service A	0.7	39904	4/1/2009
Helen	0.9	Service A	0.5	39873	3/1/2009
Ian	0.3	Service A	0.4	39845	2/1/2009
Jane	0.5	Service A	0.8	39814	1/1/2009
Kylie	0.4	Service A	0.9	39783	12/1/2008

2.2.1.2 Recommender Trust

Trust in peers as recommenders of other peers is also stored locally by each peer. Both trust in service providers and recommenders can be stored in one table and differentiated by the context field. However, for the base trust model, trust in service providers and trust in recommenders are stored in two separate tables. Again, the date column is not stored but is included here for clarity.

TABLE II : Trust in Recommenders

provider	trust	context	knowledge	timestamp	Date
bob	0.8	Recommender	0.9	40057	9/1/2009
Chris	0.7	Recommender	0.8	40026	8/1/2009
David	0.9	Recommender	0.7	39995	7/1/2009
Emily	0.9	Recommender	0.9	39965	6/1/2009
Frank	0.7	Recommender	0.9	39934	5/1/2009
Greg	0.2	Recommender	0.4	39904	4/1/2009
Helen	0.6	Recommender	0.3	39873	3/1/2009
Ian	0.9	Recommender	0.2	39845	2/1/2009
Jane	0.4	Recommender	0.6	39814	1/1/2009
Kylie	0.8	Recommender	0.3	39783	12/1/2008

2.2.2 Trust Formation

Trust Formation is the process that enables decision-making in the Trust model. Here a trust opinion or rating is formed for each available provider (service provider, content provider, etc) by composing stored trust information and recommendations from other peers. The trust opinion is formed by combining the stored trust value and the received recommendations weighted by the trust in the recommender.

Only recommendations from recommenders with a recommender trust value above a set threshold (x) are used in the trust formation function. For this example x is set to '0.5'.

The trust formation function will be explained with the use of the example provided in Figure 4. Alice needs Service A offered by Bob, Chris and David. Alice needs to select one of the three service providers.

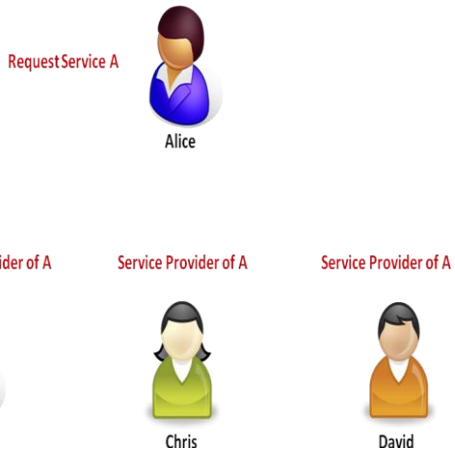

Figure 4: Base trust model example problem scenario: Service Provision

Alice's trust in other entities as providers of Service A, is represented by TABLE I. A trust opinion is formed for each of the three service providers by combining Alice's locally stored trust value for that provider (representing direct experience) and recommendations about that provider obtained from other peers. Initially a trust opinion is formed from direct experience and independently from recommendations. Then the two opinions are used to compose one trust opinion which is used for the decision making.

2.2.2.1 Trust Opinion based on Direct Experience

First a trust opinion is formed from direct experience by discounting the stored trust value by knowledge and time.

TABLE III: Direct Trust Opinion

Provider	Trust (t)	Knowledge (k)	time discounted k	Direct Trust Opinion
Bob	0.9	0.9	0.8998	0.809843
Chris	0.8	0.8	0.7991	0.639241
David	0.7	0.2	0.1983	0.138794

The stored knowledge is first discounted by time by applying Equation 1, so that newer knowledge would have more value than older knowledge. Next, the stored trust value is discounted using the time discounted knowledge by applying Equation 2. This forms the direct experience trust opinion. This is applied for each provider of the requested service and the results are displayed in TABLE III.

Equation 1: Discount Knowledge by Time

$$TimeDiscountedKnowledge = StoredKnowledge - \frac{Today - StoredTime}{Today}$$

Equation 2: Direct Trust Opinion

$$DirectTrustOpinion = StoredTrust \times TimeDiscountedKnowledge$$

2.2.2.2 Trust Opinion based on Recommendations

A trust opinion is composed for each service provider using recommendations about that service provider. Each recommendation is signed by the recommender. TABLE IV lists the recommendations about Bob. The recommended trust value is first discounted by the recommender's knowledge and by time. The results are shown in the last column of TABLE IV. This is done by applying Equation 3 to each recommendation.

TABLE IV: Recommendations about BOB

Recommender	trustee	trust	Context	k	timestamp	discounted recommend-ation
Chris	Bob	0.9	Service A	0.9	39904	0.8060
Emily	Bob	0.8	Service A	0.3	39873	0.2352
Frank	Bob	0.2	Service A	0.5	39845	0.0945
Jane	Bob	0.7	Service A	0.8	39814	0.5538

Equation 3: Discounted Recommendation

$$DiscountedRecommendation = RecommendedTrust \times \left(RecommendedKnowledge - \frac{Today - RecommendedTime}{Today}\right)$$

Next, the trustworthiness of the recommender is looked at from TABLE II. Trust in each recommender is discounted using knowledge and time using Equation 4. Next, recommendations from the recommenders with Discounted Recommender Trust values below a threshold y=0.5 are discarded. The other recommenders are ranked based on their Discounted Recommender Trust. The results are displayed in TABLE V. Next the Recommendation Trust Opinion for Bob is composed from the ranked recommendations by weighing each recommendation by rank using Equation 5. The trust opinion from Recommendations for Bob is calculated as *0.28*. The above steps are repeated to calculate the trust opinion from recommendations for the other service providers and the result is shown in TABLE VI.

Equation 4: Discounted Recommender Trust

$$DiscountedRecommenderTrust = RecommenderTrust \times \left(RecommendedKnowledge - \frac{Today - RecommenderTime}{Today}\right)$$

Equation 5: Recommended Trust Opinion

$$RecommendationTrustOpinion = \sum_{recommenders}\left(\frac{\sum ranks - rank}{\sum ranks} \times DiscountedRecommentation\right)$$

TABLE V: Rank the Recommenders

Recommender	Discounted Recommender trust	Rank
Chris	0.559336062	3
Emily	0.807776058	1
Frank	0.627728634	2
Jane	0.237503994	

TABLE VI: Trust Opinion from Recommendations

Service Provider	Recommended Trust Opinion
Bob	0.28
Chris	0.45
David	0.54

2.2.2.3 Trust Opinion based on Direct Experience and Recommendations

Finally, the Trust Opinion is composed from the Trust Opinion from Direct Experience presented in TABLE III and the Trust Opinion from Recommendations presented in TABLE VI using Equation 6. '*a*' is a user defined constant that reflects the decision-maker's disposition to trust its own experience versus recommendations from others. In this example '*a'* is set to *0.8.* The trust opinions for all the service providers are shown in TABLE VII. In this example scenario, Alice needs to select one service provider of Service A. Bob is selected because he scored the highest trust opinion.

Equation 6: Trust Opinion for Service Provider

$$TrustOpinion = (a \times DirectTrustOpinion) + ((1 - a) \times RecommendedTrustOpinion)$$

TABLE VII: Service Providers Trust Opinion

Service Provider	Trust Opinion
Bob	.704
Chris	.602
David	.22

2.2.3 Trust Dissemination

In this base trust model, trust information is shared upon request. We assume that when asked for recommendations, peers respond with a signed tuple representing their own trust in the peer in question in a specific context. An example of recommendations is provided in TABLE IV.

2.2.4 Trust Evolution

In this base trust model, whenever an interaction takes place the trust in the service provider is updated. In addition, after the interaction takes place the resulting trust is compared to the trust recommended by other peers and trust in recommenders is updated.

2.2.4.1 Updating trust in the service provider

Using the previous example scenario, let us assume that interaction took place between Alice and Bob. Bob is assigned a trust value for that specific interaction between [0, 1]. The way the evaluation of the interaction takes place is out of the scope of this model. Next, the interaction trust value is compared to the stored direct trust value for that service provider (Bob) in TABLE I. If the interaction trust value is more than the stored direct trust value, the stored direct trust value is incremented with a constant 'c' that is user defined and can be tuned to reflect the user's disposition to building trust. Similarly, if the interaction trust value is less than the stored direct trust value, the stored direct trust value is decremented with a constant 'd' that is user defined and can be tuned to reflect the user's disposition to decreasing trust.

The constants used for the sample example are c = 0.01 and d = 0.01. Let us assume that the result of the current interaction between Alice and Bob is 0.95. The stored trust value for bob is retrieved from TABLE I, and compared to the current interaction trust value. If the current interaction trust value is greater or equal to the stored trust value, the stored trust value is incremented with a constant c using Equation 7.

Equation 7: Increment Stored Trust

$$\begin{aligned}&\textit{if } ((\textit{StoredTrust} + c) < 1)\\&\quad \text{NewStoredTrust} = \textit{StoredTrust} + c\\&\text{else}\\&\quad \text{NewStoredTrust} = 1\end{aligned}$$

However, if the current interaction trust value is less than the stored trust value, the stored trust value is decremented with a constant c using Equation 8.

Equation 8: Decrement Stored Trust

$$\begin{aligned}&\textit{if } ((\textit{StoredTrust} - d) > 0)\\&\quad \text{NewStoredTrust} = \textit{StoredTrust} - d\\&\text{else}\\&\quad \text{NewStoredTrust} = 0\end{aligned}$$

On the other hand, the stored knowledge for Bob in TABLE I is incremented by a constant e, regardless of the outcome of the interaction using

Equation 9. The stored timestamp is replaced with a timestamp representing the time the current interaction took place. The results of the update is shown in TABLE VIII.

Equation 9: Increment Knowledge

$$\begin{aligned}&\textit{if } ((\textit{StoredKnowledge} + e) < 1)\\&\quad \text{NewStoredKnowledge} = \textit{StoredKnowledge} + e\\&\text{else}\\&\quad \text{NewStoredKnowlege} = 1\end{aligned}$$

TABLE VIII: Updated Stored Trust in Service Providers

provider	trust	context	k	Time	Date
Bob	0.91	Service A	0.901	40064	9/8/2009
Chris	0.8	Service A	0.8	40026	8/1/2009
David	0.7	Service A	0.2	39995	7/1/2009
Emily	0.9	Service A	0.3	39965	6/1/2009
Frank	0.8	Service A	0.6	39934	5/1/2009
Greg	0.7	Service A	0.7	39904	4/1/2009
Helen	0.9	Service A	0.5	39873	3/1/2009
Ian	0.3	Service A	0.4	39845	2/1/2009
Jane	0.5	Service A	0.8	39814	1/1/2009
Kylie	0.4	Service A	0.9	39783	12/1/2008

2.2.4.2 Updating trust in the recommenders

Similar to the previous section, stored trust values (TABLE II) for each of the recommenders who provided recommendations about Bob is updated. This includes Jane whose recommendation was excluded by the trust formation function. For each recommender, the recommended trust value is compared to current interaction trust value. If the difference is below a threshold f, then the stored recommender trust is incremented by a constant g. If the difference is above f, then the recommender trust is decremented by a constant h using Equation 10. The stored knowledge is also updated. However, knowledge is incremented by a constant e regardless of how useful the recommendation was using

Equation 9. e is set as 0.025 this time. The timestamp in updated to the time the recommendations were used in the decision making, i.e. the interaction time. The results are displayed in TABLE IX.

Equation 10: Updating Trust in Recommenders

$$\begin{aligned}&\textit{if } (\left|\text{InteractionTrust} - \text{RecommendedTrust}\right| < f)\\&\{\quad \text{if}(\textit{StoredTrust} + g < 1)\\&\qquad \{\quad \text{NewStoredTrust} = \textit{StoredTrust} + g\\&\qquad \}\\&\qquad \text{else}\\&\qquad \{\quad \text{NewStoredTrust} = 1\\&\qquad \}\\&\}\\&\text{else}\\&\{\quad \text{if}(\text{StoredTrust} - h > 0)\\&\qquad \{\quad \text{NewStoredTrust} = \text{StoredTrust} - h\\&\qquad \}\\&\qquad \text{else}\\&\qquad \{\quad \text{NewStoredTrust} = 0\\&\qquad \}\\&\}\end{aligned}$$

TABLE IX: Updated Trust in Recommenders

Provider	trust	Context	K	time	Date
Bob	0.8	Recommender	0.9	40057	9/1/2009
Chris	0.8	Recommender	0.825	40064	9/8/2009
David	0.9	Recommender	0.7	39995	7/1/2009
Emily	1	Recommender	0.925	40064	9/8/2009
Frank	0.6	Recommender	0.925	40064	9/8/2009
Greg	0.2	Recommender	0.4	39904	4/1/2009
Helen	0.6	Recommender	0.3	39873	3/1/2009
Ian	0.9	Recommender	0.2	39845	2/1/2009
Jane	0.3	Recommender	0.625	40064	9/8/2009
Kylie	0.8	Recommender	0.3	39783	12/1/2008

2.3 Apply Base Trust Model to other problem scenarios

In Section 2.2, the base trust model was explained by applying it to the service provision problem domain. Now, its applicability in the content management and routing problem domains is demonstrated.

2.3.1 Content Management Problem Domain

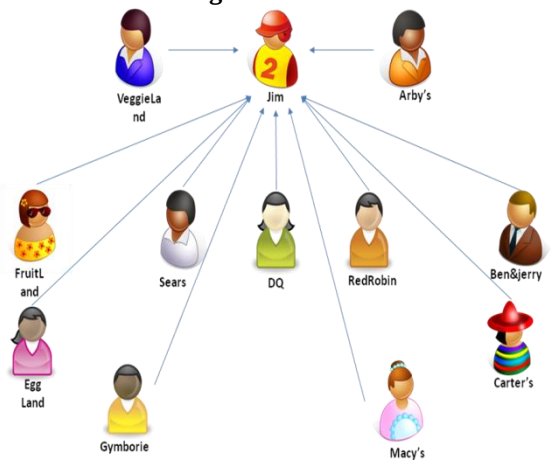

Figure 5: Base trust model example problem scenario: Content Management

Figure 5 shows a sample content management scenario. Jim enters a shopping mall and is bombarded with advertisements from different content providers. Jim only wants to view five advertisements. Jim stores a table of direct trust values locally as shown in TABLE X. These values were created through Jim's own direct experiences. Jim also stores locally a table of trust in peers as recommenders, shown in TABLE II.

TABLE X: Direct Trust in Content Providers

Provider	trust	context	Knowledge	time	Date
Arby's	0.9	Ad Provider	0.9	40057	9/1/2009
Ben&jerry	0.8	Ad Provider	0.8	40026	8/1/2009
Carter's	0.7	Ad Provider	0.2	39995	7/1/2009
DQ	0.9	Ad Provider	0.3	39965	6/1/2009
EggLand	0.8	Ad Provider	0.6	39934	5/1/2009
FruitLand	0.7	Ad Provider	0.7	39904	4/1/2009
Gymborie	0.9	Ad Provider	0.5	39873	3/1/2009
Macy's	0.3	Ad Provider	0.4	39845	2/1/2009
RedRobin	0.5	Ad Provider	0.8	39814	1/1/2009
Sears	0.4	Ad Provider	0.9	39783	12/1/2008
VeggieLand	0.9	Ad Provider	0.6	39934	5/1/2009

In the Trust Formation Function, first Jim asks his peers for recommendations about the content providers. Four peers respond: Chris, Emily, Frank and Jane. As in Section 2.2.2.2, the stored value for recommender trust for each peer is discounted by knowledge (how well does Jim know the recommender) and time. Recommendations from peers with a recommender trust value below a threshold y=0.5 are discarded. For each content provider, a recommendation trust opinion is calculated as explained in Section 2.2.2.2. A direct trust opinion is also computed for each content provider as explained in Section 2.2.2.1. Next, both the direct trust opinion and recommendation trust opinion are combined to produce a final trust opinion that is used in the decision making process as discussed in Section 2.2.2.3. Finally, the content providers are ranked based on their final trust opinion and the top 5 ads are displayed to Jim. The results of the calculations are displayed in TABLE XI. After Jim views the ads, he rates their usefulness. Next, the stored trust values for content providers are updated as explained in Section 2.2.4.1. Similarly, trust in recommenders is updated as explained in Section 2.2.4.2. The recommendations used in the calculations, and other details of the results displayed in TABLE XI are provided in Appendix 1.

TABLE XI: Content Provider Trust Results

Content Provider	Direct Trust Opinion	Recommendation Trust Opinion	Final Trust Opinion	Rank
Arby's	0.809843	0.704967	0.704967	1
Ben&Jerry	0.639241	0.603623	0.603623	2
Carter's	0.138794	0.198919	0.198919	11
DQ	0.267776	0.292528	0.292528	9
EggLand	0.477404	0.4826	0.4826	4
FruitLand	0.487204	0.459268	0.459268	6
Gymborie	0.445709	0.448546	0.448546	7
Macy's	0.11836	0.209433	0.209433	10
RedRobin	0.39688	0.476844	0.476844	5
Sears	0.357194	0.344391	0.344391	8
VeggieLand	0.53708	0.534928	0.534928	3

2.3.2 Routing Problem Domain

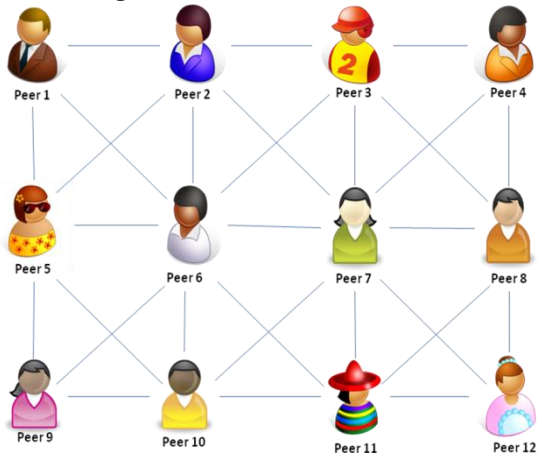

Figure 6: Base trust model example problem scenario: Routing

Figure 6 provides a sample routing problem scenario where peers route packets to their destination through other peers. For example Peer 7 wants to route a packet to Peer 1, and wants to select which of its first hop neighbors to forward the packet through. The base trust model can be tuned to rely only on direct trust experience by setting a= 1 in Equation 6. This would result in a trust decision based solely on direct experience. The calculations in this example are similar to the examples in the previous two domains. Peer 7 maintains trust values for peers in the "routing" context. These values are updated with direct experience. Peer 7 can also maintain trust values for peers in the "recommender" context. Here Peer 7 has to select between Peers 2, 3, 4, 6, 8, 10, 11 and 12. This is performed in the same way as in the service provision example explained in Section 2.2.

3 Related Work in Trust Management

This section briefly describes the genealogy of trust models and the evolution of trust management ideas. Trust management started as an attempt to enable communication in dynamic environments where servers can no longer assume prior knowledge of clients, and where mobile clients may also want to have a mechanism to select service providers who may be more beneficial. PolicyMaker [13] and KeyNote [14] did that by enabling each entity to select the parties to trust and to define trust relationships in local policies. Hence, this enabled communication without the need for centralized root certification authorities. This also enabled to some extent anonymous communication with unknown entities. However, these unknown entities need to present certificates issued by an entity trusted by the service provider to issue certificates for the requested service. The identity of trustee is known to the certificate issuer but unknown to the trustor, as certificates bind keys to actions and not to identities.

TABLE XII: Evolution of trust management models

	Idea
[16] – Beth et al. 1994	– Introduced many trust management ideas (unpolished)
[13] Policy Maker 1996 and [14] Keynote 1998	– Term "Trust Management" coined. – **Certificate-based trust model** (analysis of policies & credentials) – Look a lot like access control, but binding is between keys & authorized action
[17] Abdul-Rahman & Hailes 2000	– Introduction to **reputation-based trust** models & agents autonomy – Each Agent maintains a database of recorded experience – Recommendations exchanged
[18] Aberer & Despovic 2001	– **Distributed Storage** of Trust information (complaints) – No mechanism for dynamic bootstrapping of devices
[12] CONFIDANT 2002	– Incorporation of **detection & isolation of misbehavior** – Local storage of trust values – Sharing of trust data limited to a static list of peers
[19] SECURE 2003	– Incorporates trust model & risk model – Distinguishes between unknown and distrusted entities – Enables delegation of trust evaluation
[10] hTRUST 2004	– Incorporates interpersonal and **dispositional trust** – **Handles formation, dissemination & evolution of trust** data – Detection & isolation of malicious recommenders
[20] McNamara et al. 2006	– Mobility introduced as a factor in decision making
[21] STRUDEL 2006	– Combat Tragedy of the commons (node selfishness)
[15] MATE 2007	– **Integrated management of trust (interpersonal and dispositional) and risk** – Risks limited to timeliness of service delivery

Since then, trust management models have evolved to provide for autonomous decision making and rely on probabilistic trust values that continuously evolve due to experience or information sharing between peers. hTrust is one of the more recent models that explicitly deals with trust management functions including trust formation, dissemination and evolution [10]. It also incorporates a mechanism for bootstrapping new entities. However, risk is excluded from the equation. MATE attempts to incorporate risk analysis but their solution is only restricted to 'timeliness of delivery', which excludes all other risks, such as malicious code [15].

Trust models have built on each other. TABLE XII lays out the main ideas and contributions of each trust

model. The first four papers are the ancestors of the remaining trust management approaches.

4 Discussion

Today the area of Trust Management is far from maturity. Current solutions, discussed in Section 3, either provide high-level frameworks with little functionality, or functionality in the form of a point solution targeting a specific problem scenario. Most models deal with ad-hoc mobile P2P networks and handle simple relationships between peers. Existing work does not satisfactorily provide a functional model that sufficiently addresses the aspects of trust management and is general enough to be applicable in multiple problem scenarios. Specifically, all the surveyed models provided point solutions for specific problem scenarios, and even then restricted them with assumptions and limitations. Hence, there lacks a general trust management model that is applicable across multiple problem scenarios.

In addition, new environments, like ubiquitous computing environments, introduce new issues that are not addressed sufficiently by the available point solutions. For example, current trust models deal with a single class of participants and lack the ability to deal with multiple classes of users that might be useful when applying trust management to more complex environments.

Another consideration is privacy management. Out of the surveyed models only [13], [14] and [12] directly try to incorporate privacy policies in their trust models. In [13] and [14], privacy policies are restricted to assigning privileges to public keys, so essentially they are restricted to helping enforce access control in public key environments. In [12] the use of privacy policies is outlined as part of the trust formation function in a high-level framework. However, the semantics of this incorporation are left out. Hence, none of the surveyed models satisfactorily addresses privacy management.

The base model presented in this paper provides a straight forward, easy to understand trust model that deals explicitly with storage of trust information, trust formation, trust dissemination and trust evolution. The storage of trust data is local for each peer, and each peer independently reasons about trust in the trust formation function. Trust information is shared upon request and stored trust data is updated at the conclusion of each interaction. The design of the base model was intended to be simple and straight forward to ease its enhancement to satisfy the needs of smart spaces in future work. In this paper we provide examples of applying the base trust model to scenarios from diverse problem domains namely service provision, content management and routing.

5 Requirements for extending the base trust model to satisfy smart space scenarios

The trust model produced in Section 2.2 will be used as a base model for the production of a trust model for smart spaces, by enhancing it to satisfy the select smart space scenarios. The scenarios are arranged by their simplicity, the simplest coming first. The model will be modified to satisfy the issues of one scenario at a time. Before moving on to the following scenario, the model has to cover all the issues of the current scenario plus the previously applied smart space scenarios. This process is done iteratively until all the smart space scenarios are covered. The result of this phase would be a functional trust model, rich enough to handle diverse smart space scenarios.

5.1 Tailor to Smart Space Scenario1

The base trust model produced in Section 2.2 is enhanced to satisfy the requirements of : Smart Space Scenario 1.

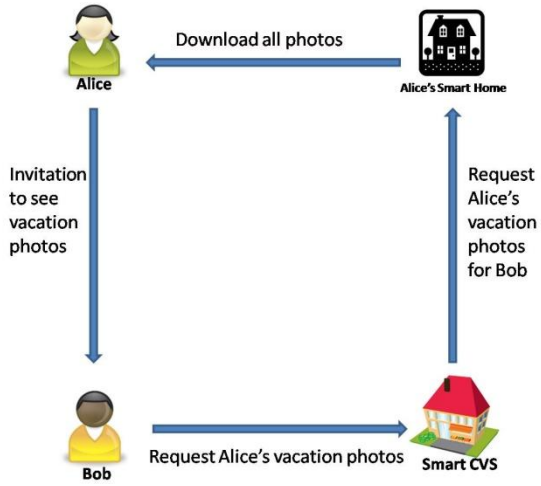

Figure 7 : Smart Space Scenario 1

In this scenario, Alice downloads her vacation photos to her album manager in her smart home. She sends invitations to her friends, including Bob to view her vacation album. Bob goes to Smart CVS and tries to access Alice's vacation album to print her photos on Smart CVS's Kodak photo machine. This scenario deals with transitivity of trust. If Alice trusts Bob to view her photos, and Bob trusts Smart CVS, does that mean that Alice trusts Smart CVS too? In addition, Smart CVS and Alice's Smart Home are unknown to each other, which introduces the issue of building initial trust (trust formation) between smart spaces.

In this scenario we deal with whether to grant access to a particular entity to a specific resource, "vacation photos". When Alice sends an invitation to Bob to view "vacation photos", Trust= 1 and Knowledge= 1, is entered in Alice's direct trust table to reflect that Alice elected to grant Bob access to her "vacation photos". In this Scenario, Bob tries to access the resource through Smart CVS. There are a few options on how Alice can decide whether or not to grant access based on trust in this situation.

5.1.1 Trust Smart Space if Resource Requestor is Trusted.

Alice can simply trust Smart CVS, because Bob trusts it. In this case, the base trust model can be used with minimal changes. Alice looks at direct trust data and recommendations about Bob to come up with a trust opinion. If Bob's trust opinion > threshold, then CVS is trusted. Recommendations can be ignored and hence Bob is granted access regardless of through which smart space access is granted. Of course this option would not demonstrate a multi-level trust decision and is not helpful in our purpose of producing a trust model for smart spaces that provides for multi-level trust decisions.

5.1.2 Independently calculate trust opinion for the Smart Space.

5.1.2.1 Trust formation

Another option is to independently calculate trust for Bob and Smart CVS. Combine both trust opinions and if > threshold, access is granted. The two trust opinions are combined by assigning weights to the trust in the requestor (Bob) and the trust in the smart space (Smart CVS). Another alternative would be to first look at the trust opinion in the requestor (Bob). If it is above a threshold, then look at the trust opinion in the smart space (Smart CVS). If it is above a threshold, then access is granted. The threshold can be different for trust in requestors and trust in smart spaces.

Also, it is important to point out that the trust opinion calculated to the smart space might be in a different context than that calculated for the requestor. For example, while the trust opinion for Bob is calculated for the context "vacation photos", the trust opinion for Smart CVS is calculated in another context like photo management, or data privacy, etc.

This brings up another issue: Which context to look at in the trust opinion calculation for the smart space. This demonstrates a need for the classification of the contexts themselves. For example, "vacation photos" would be members of a class called "Photos". Consequently smart spaces that try to access members of the Photos class are assigned the "Photo Management" context. This has to be incorporated in the Smart Space Trust Model.

This scenario demonstrates that there are two levels to the trust decision. Alice needs to trust Bob via Smart CVS, hence taking the property of transitivity a step further than the traditional trust management models.

5.1.2.2 Trust evolution

After the interaction takes place, Alice updates trust in both Bob and Smart CVS. In addition, Smart CVS needs to trust both Bob and Alice. Bob is identified to Smart CVS by his CVS card, so Smart CVS maintains a trust value for Bob. In this stage, trust in Alice is assumed based on trust in Bob. After the interaction takes place, trust in Bob is updated. No data regarding Alice is saved.

5.2 Tailor to Smart Space Scenario 2

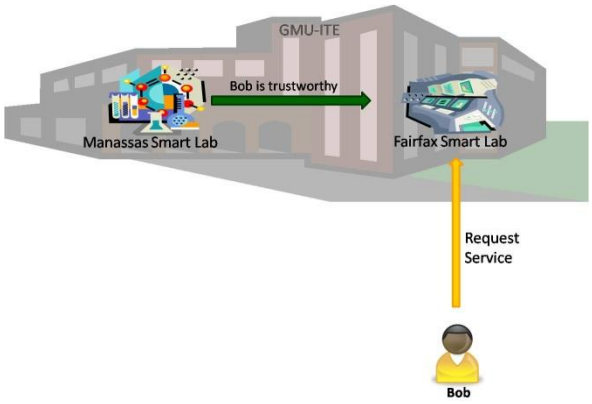

Figure 8: Smart Space Scenario 2

In Scenario 2, Manassas Smart Lab and Fairfax Smart Lab belong to GMU-ITE, which entails that they trust each other to some extent as specified by their owner GMU-ITE. This type of trust relationship falls under Institutional Trust. Bob is a Manassas student trusted (to some extent) by the Manassas Smart Lab. Bob is unknown to the Fairfax Smart Lab. Hence, Fairfax Smart Lab needs to form a trust opinion of user Bob based on Manassas Smart Lab's recommendation. This raises the issue of building initial trust (trust formation) between smart spaces and users from shared trust information (recommendations). Another important issue is how trust information is shared (trust dissemination).

In an institution, affiliated smart spaces might need to share trust information to try and detect misbehaving users quickly.

5.2.1 Share trust data every time a trust decision is made

One option is to request recommendations from affiliated smart spaces every time in the trust formation function to decide on whether or not to grant access. Although, this is a more secure option, it is also a more expensive one.

5.2.2 Share trust data periodically

Another option is that affiliates periodically share recommendations and these recommendations are used to update the stored data. Each smart space updates local stored trust data with every interaction. However, recommendations might need to be requested for unknown users.

5.3 Tailor to Smart Space Scenario 3

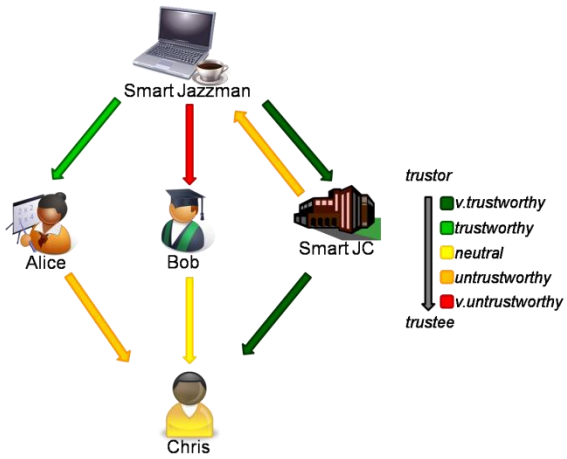

Figure 9: Smart Space Scenario 3

Suppose Chris lives near George Mason University and frequently visits Jazzman Coffee. Chris would like to get internet connectivity and check his email while he is there. Today, Jazzman Coffee does not provide any smart services. However, since Jazzman Coffee is on-campus, the GMU wireless network is available. This network is accessible by authenticating users who have user accounts (students, faculty, staff). Chris is not affiliated with GMU, therefore he cannot access the network.

Ideally, Jazzman Coffee is a Smart Space that provides services to its customers. In this scenario, Chris is unknown to Jazzman Coffee Smart Space. However, there exists a chain of trust between the two entities. Jazzman Coffee Smart Space receives multiple trust opinions (recommendations) about Chris that might be very different than each other. These recommendations are used to form a trust opinion of User Chris.

In this scenario, the trust model is used to decide whether or not grant access to service to Chris. The decision maker is a smart space and we have two classes of peers: users and smart spaces. The use of classes enables for a more fine grained trust model. For example, stereotyping through the use of classes can enable for recommendations from members of a certain class to be given more weight than recommendations from members of other classes. For example, in this scenario we can define two classes: users, smart spaces. Smart Jazzman can elect to trust other smart spaces more than users. A trust opinion can be formed independently for recommendations from smart spaces, recommendations from users. The trust formation function would then combine both trust opinions giving more weight to the preferred class.

5.4 Tailor to Smart Space Scenario 4

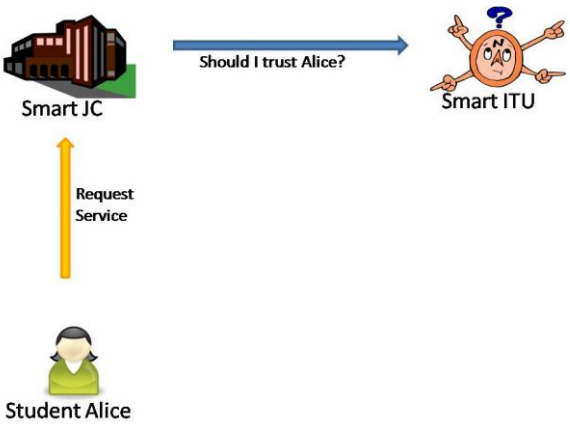

Figure 10: Smart Space Scenario 4

Scenario 4 presents an example of trust delegation where the trust decision is completely transferred to another entity, hence demonstrating an option of centralized storage of trust data. For example, all trust in users is handled by Smart ITU that stores and manages trust values for all users. Whenever a user requests service from a smart space, like Smart JC, the trust value for that user is requested from Smart ITU. The retrieved trust value can then be discounted by Smart JC based on its stored trust, knowledge and time for Smart ITU. Here trust in Smart ITU reflects the variation in opinion that may exist between the ITU and JC and the application of the trust formation function would result in an increase or decrease of the value for Alice recommended by the ITU. Hence, a new mechanism would need to be added to the base model to enable the incorporation of a measure that denotes variation in opinion. In this example, results of the interaction are sent by JC to ITU that updates its global trust value for Alice.

5.5 Tailor to Smart Space Scenario 5

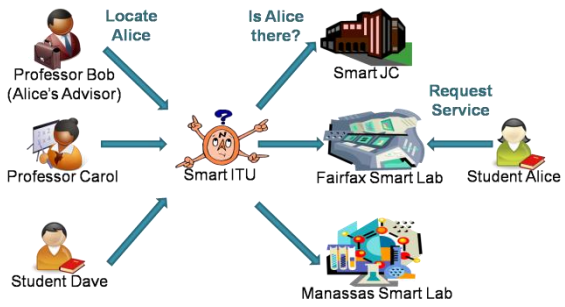

Figure 11 : Smart Space Scenario 5

Professor Bob, Professor Carol and Student Dave would like to find Alice. Today, if an individual wishes to locate another individual on campus, she or he has two options: Manually look for the person, or call the person and ask about his or her location.

This Scenario demonstrates an ideal example of location services. Alice is logged into a work station in the Fairfax Smart Lab. In this Scenario let us assume that location services are provided by the Smart ITU. Smart ITU maintains a database of privacy policies of users and spaces. Whenever an entity's location is requested, Smart ITU checks the entity's privacy policy to determine whether or not to execute the request.

In this example three actors request Alice's location. Let us assume that Alice's policy permits her advisor to retrieve her exact location, permits other professors to find out if she is on-campus or not, and prevents all others from accessing her location. Hence, Professor Bob receives "Alice in Fairfax Smart Lab". Professor Carol receives "Alice is on-campus." Student Dave receives "Request denied." This scenario employs user policies to add granularity to trust decisions, hence enhancing privacy protection. It also introduced "stereo-typing", a novel concept which reflects human trust mechanisms (or bias) where certain "classes" of people are trusted more than others.

An enhancement of this scenario is to introduce privacy policies of smart spaces to the equation. Suppose for example that classified research is conducted in the Fairfax Smart Lab. Fairfax Smart Lab's privacy policy states that only lab personnel are allowed to know who is in the lab at any particular time. Hence, when Smart ITU receives a request for Alice's location, both Alice's policy and the Fairfax Smart Lab's policy, need to be examined to decide whether or not to grant the request. A consideration here is which policy supersedes when there is a conflict. More research is needed to determine a solution for this problem. One possible solution is the implementation of lattices in a way similar to that employed by lattice-based access control.

Assuming Fairfax Smart Lab's policy supersedes Alice's, Professor Bob would receive "Alice on-campus" instead of the previous response. This, however, introduces another problem. Since Professor Bob can access Alice's location at all times except when she is in the Fairfax Smart Lab, he can easily infer her location. Inference channels are out of the scope of this thesis.

After apply Scenarios 1 to 4, our model, like other trust models described in Section 3, provides a 'one-size-fits-all' solution, with little or no personalization that allows individual users to direct the use of the trust model. The application of privacy policies would allow individual users and/or smart spaces to direct the application of the trust model hence enable fine-grained personalization of its use. Therefore, using this scenario as an example, the goal would be to enhance the base trust model using stereo-types or classes and policy constraints to enable individual users to tailor its usage to their preferences. Basically, the decision maker's privacy policy is used to select a subset of the trust data on which the trust model is applied. Hence, the decision maker is allowed to explicitly include or eliminate particular peers from the trust calculation, thereby personalizing the trust experience.

6 Conclusion and Future Work

The field of trust management has made several important contributions to improving the management of protection and quality of service in distributed systems. It has refined the notion of access control by relating protection to a prediction of the *actions* of an entity, in addition to its identity. It has enabled systems to gracefully handle requests from strangers, both by securely exchanging recommendations or credentials among trusted peers, and by building trust over time, while managing risk. It has enlarged the scope of protection, enabling all interacting peers to manage their protection, not just professionally administered servers. And it has successfully applied trust models and mechanisms both for making decisions concerning protection and concerning quality of service assurances.

Nevertheless, some issues remain to be addressed. Different trust models have arisen in different problem domains, and a general trust model that is applicable across domains is missing. Furthermore, even within a domain, trust models have often been developed to address a specific problem or application, thus lacking the generality to be applicable to richer scenarios.

In this paper, we presented a base trust model that is applicable in multiple problem scenarios. We have also provided an example of applying it in three problem

scenarios, namely service provision, content management and packet routing.

Pervasive computing is an especially rich and challenging domain, where in addition to more traditional notions of protection, the protection of privacy plays a very important role. In Section 5 several scenarios were presented to demonstrate that smart spaces introduce a new set of requirements that the base trust model cannot currently satisfy. The next step would be to enhance the base trust model to be applicable in these richer scenarios. This would entail the classification of entities into classes and the employment of privacy policies to enable more fine-grained trust decisions and personalized use of the trust model.

7 References

[1] S. Meyer and A. Rakotonirainy, "A survey of research on context-aware homes," *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21*, Australian Computer Society, Inc. Darlinghurst, Australia, Australia, 2003, pp. 159-168.

[2] "the pad, dubai, ipad, busines bay, dubai investments - find me a property in dubai."

[3] L. Arnstein, C.Y. Hung, R. Franza, Q.H. Zhou, G. Borriello, S. Consolvo, and J. Su, "Labscape: A Smart Environment for the Cell Biology Laboratory," *IEEE PERVASIVE COMPUTING*, 2002, pp. 13-21.

[4] B. Johanson, A. Fox, and T. Winograd, "The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms," *IEEE PERVASIVE COMPUTING*, 2002, pp. 67-74.

[5] J.E. Bardram, "Applications of context-aware computing in hospital work: examples and design principles," *Proceedings of the 2004 ACM symposium on Applied computing*, ACM New York, NY, USA, 2004, pp. 1574-1579.

[6] "'Smart' hospital to improve care," *BBC*, May. 2003.

[7] T. RiisgaardHansen, J.E. Bardram, and M. Soegaard, "Moving Out of the Lab: Deploying Pervasive Technologies in a Hospital," *IEEE PERVASIVE COMPUTING*, 2006, pp. 24-31.

[8] G.D. Abowd, "Classroom 2000: An experiment with the instrumentation of a living educational environment," *IBM Systems Journal*, vol. 38, 1999, pp. 508-530.

[9] M. Fleck, M. Frid, T. Kindberg, E. O'Brien-Strain, R. Rajani, and M. Spasojevic, "From Informing to Remembering: Ubiquitous Systems in Interactive Museums," *IEEE PERVASIVE COMPUTING*, 2002, pp. 13-21.

[10] L. Capra, "Engineering human trust in mobile system collaborations," *Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering*, 2004, pp. 107-116.

[11] D. Quercia, S. Hailes, and L. Capra, "TRULLO-local trust bootstrapping for ubiquitous devices," *Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on*, 2007, pp. 1-9.

[12] S. Buchegger and J.Y. Le Boudec, "Performance analysis of the CONFIDANT protocol: Cooperation of nodes—fairness in dynamic ad-hoc networks," *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, 2002, pp. 226–236.

[13] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, 1996, pp. 164–173.

[14] M. Blaze, J. Feigenbaum, and A.D. Keromytis, "KeyNote: Trust Management for Public-Key Infrastructures," *Lecture Notes in Computer Science*, vol. 1550, 1999, pp. 33-60.

[15] D. Quercia and S. Hailes, "MATE: Mobility and Adaptation with Trust and Expected-utility," *International Journal of Internet Technology and Secured Transactions*, 2006.

[16] T. Beth, M. Borcherding, and B. Klein, "Valuation of Trust in Open Networks," *Proceedings of the Third European Symposium on Research in Computer Security*, Springer-Verlag London, UK, 1994, pp. 3-18.

[17] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities," *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, p. 9.

[18] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," *Proceedings of the tenth international conference on Information and knowledge management*, 2001, pp. 310-317.

[19] V. Cahill, E. Gray, J.M. Seigneur, C.D. Jensen, Y. Chen, B. Shand, N. Dimmock, A. Twigg, J. Bacon, and C. English, "Using Trust for Secure Collaboration in Uncertain Environments," 2003.

[20] L. McNamara, C. Mascolo, and L. Capra, "Trust and Mobility Aware Service Provision for Pervasive Computing," *Proc. of the*, vol. 1, 2006, pp. 603–610.

[21] D. Quercia, M. Lad, S. Hailes, L. Capra, and S. Bhatti, "STRUDEL: supporting trust in the dynamic establishment of peering coalitions," *Proceedings of the 2006 ACM symposium on Applied computing*, 2006, pp. 1870-1874.

8 Appendix 1

This section provided the details of the trust formation function of the application of the base trust model to the content management problem scenario. Jim's stored trust information is provided in TABLE XIII. It includes trust data in two contexts: Ad provision and providing recommendations.

First the direct trust opinion is calculated using September, 8, 2009 as the current date. First, knowledge is discounted by time using Equation 1. Next, the direct trust opinion is calculated for each Ad provider using Equation 2. The results are displayed in TABLE XIV.

In this sample scenario, recommendations are received from Chris, Emily, Frank and Jane. The stored knowledge value for recommender trust is discounted by time for each recommender to come up with a discounted trust opinion in each recommender using Equation 4. The results for this step are displayed in TABLE XV. Recommendations from recommenders with discounted trust below a threshold y=0.5, are discarded. Hence, Jane is excluded from the ranking process and her recommendations are ignored by the trust formation function.

TABLE XIII: Jim's stored trust data

Provider	trust	Context	k	time	Date
Arby's	0.9	Ad Provider	0.9	40057	9/1/2009
Ben&Jerry	0.8	Ad Provider	0.8	40026	8/1/2009
Carter's	0.7	Ad Provider	0.2	39995	7/1/2009
DQ	0.9	Ad Provider	0.3	39965	6/1/2009
EggLand	0.8	Ad Provider	0.6	39934	5/1/2009
FruitLand	0.7	Ad Provider	0.7	39904	4/1/2009
Gymborie	0.9	Ad Provider	0.5	39873	3/1/2009
Macy's	0.3	Ad Provider	0.4	39845	2/1/2009
RedRobin	0.5	Ad Provider	0.8	39814	1/1/2009
Sears	0.4	Ad Provider	0.9	39783	12/1/2008
VeggieLand	0.9	Ad Provider	0.6	39934	5/1/2009
Bob	0.8	Recommender	0.9	40057	9/1/2009
Chris	0.7	Recommender	0.8	40026	8/1/2009
David	0.9	Recommender	0.7	39995	7/1/2009
Emily	0.9	Recommender	0.9	39965	6/1/2009
Frank	0.7	Recommender	0.9	39934	5/1/2009
Greg	0.2	Recommender	0.4	39904	4/1/2009
Helen	0.6	Recommender	0.3	39873	3/1/2009
Ian	0.9	Recommender	0.2	39845	2/1/2009
Jane	0.4	Recommender	0.6	39814	1/1/2009
Kylie	0.8	Recommender	0.3	39783	12/1/2008

Next, a recommendation trust opinion is calculated for each Ad Provider using Equation 3 and Equation 5. The calculations of recommendations trust for Arby's are displayed in TABLE XVI. As mentioned earlier, Jane's recommendation is excluded in this calculation. Similarly, the recommendation trust opinion is calculated for all content providers and combined with the direct trust opinion to come up with a final trust opinion using Equation 6. The result is shown in TABLE XI.

TABLE XIV: Direct Trust Opinion Results

Provider	trust	context	k	Time	date	time discounted k	Direct Trust Opinion
Arby's	0.9	Ad Provider	0.9	40057	9/1/2009	0.8998	0.8098
Ben&Jerry	0.8	Ad Provider	0.8	40026	8/1/2009	0.7991	0.6392
Carter's	0.7	Ad Provider	0.2	39995	7/1/2009	0.1983	0.1388
DQ	0.9	Ad Provider	0.3	39965	6/1/2009	0.2975	0.2678
EggLand	0.8	Ad Provider	0.6	39934	5/1/2009	0.5968	0.4774
FruitLand	0.7	Ad Provider	0.7	39904	4/1/2009	0.6960	0.4872
Gymborie	0.9	Ad Provider	0.5	39873	3/1/2009	0.4952	0.4457
Macy's	0.3	Ad Provider	0.4	39845	2/1/2009	0.3945	0.1183
RedRobin	0.5	Ad Provider	0.8	39814	1/1/2009	0.7938	0.3969
Sears	0.4	Ad Provider	0.9	39783	12/1/2008	0.8930	0.3572
VeggieLand	0.9	Ad Provider	0.6	39934	5/1/2009	0.5968	0.5371

TABLE XV: Rank Recommenders

Recommender	Discounted trust	Rank
Chris	0.559336	3
Emily	0.807776	1
Frank	0.627729	2
Jane	0.237504	

TABLE XVI: Recommendations for Arby's

Recommender	trustee	trust	context	k	time	Discounted Recommendation
Chris	Arby's	0.9	Ad Provider	0.9	39904	0.8064
Emily	Arby's	0.8	Ad Provider	0.3	39873	0.2362
Frank	Arby's	0.2	Ad Provider	0.5	39845	0.0989
Jane	Arby's	0.7	Ad Provider	0.8	39814	0.5557
combine recommendations weighted by rank of recommender						0.2854629