Chapter 13
Context-Sensitive Trust Evaluation in Cooperating Smart Spaces

Nicolas Liampotis, Ioanna Roussaki, Nikos Kalatzis, Eliza Papadopoulou, João Miguel Gonçalves, Ioannis Papaioannou and Efstathios Sykas

Abstract Social networking is a dominant computing paradigm of the last decade that enables users to virtually interact and socialise, to collaborate and to share any kind of content. A drawback in current social networking systems is that they integrate poorly with the wealth of hardware and software resources that the users have access to locally or remotely. To overcome this, the Cooperating Smart Spaces (CSSs) notion has been introduced that couples the advantages of social computing with those of pervasive systems. However, as this promising merging is largely based on the collection and exploitation of various user-related information, and enables the discovery and interaction among users, groups of users and resources that are not necessarily trustworthy, a reliable trust management and evaluation system needs to be established. This chapter elaborates on such a system that has been prototyped, tested and evaluated via real user trials, in order to address the needs of CSSs. It considers context information in the trust evaluation process and it enables trust-sensitive community lifecycle and membership management; automated discovery of trusted entities; and trust-based control of personal data disclosure.

N. Liampotis (✉) · I. Roussaki · N. Kalatzis · I. Papaioannou · E. Sykas
National Technical University of Athens, 9 Heroon Polytechneiou Str, Athens 15773, Greece
e-mail: nicolas.liampotis@cn.ntua.gr

I. Roussaki
e-mail: ioanna.roussaki@cn.ntua.gr

N. Kalatzis
e-mail: nikosk@cn.ntua.gr

I. Papaioannou
e-mail: jpapai@cn.ntua.gr

E. Sykas
e-mail: sykas@cn.ntua.gr

E. Papadopoulou
Heriot-Watt University Riccarton, Edinburgh EH14 4AS, Edinburgh, UK
e-mail: E.Papadopoulou@hw.ac.uk

J. M. Gonçalves
Portugal Telecom Inovação S.A., Rua Eng. José Ferreira Pinto Basto,
Aveiro 3810-106, Portugal
e-mail: joao-m-goncalves@ptinovacao.pt

P. Brézillon, A. J. Gonzalez (eds.), *Context in Computing,*
DOI 10.1007/978-1-4939-1887-4_13

13.1 Introduction

The integration of social networking services (Kietzman et al. 2012; Lin and Lu 2011) with current pervasive computing systems (Hansmann et al. 2003; Obaidat et al. 2011) has the potential to support users to identify and interact with other individuals that share common interests, preferences or expectations, and in general, demonstrate similar context. This can eventually lead to enhancing the overall user experience, to optimised assistance of their communication and socialisation, as well as, to proactively facilitate their everyday activities with minimal effort. In order to bridge the gap between current pervasive systems and social networking services, the notion of Cooperating Smart Spaces (CSSs; Doolin et al. 2012) has been introduced. CSSs aim to extend pervasive systems beyond the individual to dynamic communities of users and are the building blocks for enabling pervasive computing in physical or virtual social communities. CSSs enable groups of users that share commonalities to join together in pervasive communities. A Pervasive Community, once constituted, forms a Community Interaction Space (CIS; Doolin et al. 2012). There is a one-to-one mapping between pervasive communities and CISs. Individuals may belong to any number of pervasive communities, and thus CISs, at the same time.

The functionality of CSSs entails three broad phases, namely, Discover, Connect and Organise (Doolin 2013), each of which contributes to the formation of CISs. More specifically, the system enables the Discovery, Connection and Organisation of relevant people, resources and things, crossing the boundary between the physical and the virtual world. The role of trust in all three phases is evident given, on one hand, the plethora and diversity of available resources and, on the other hand, the sensitivity of information that is disclosed, communicated and processed. Thus, the various resources provided should be accessible on top of a trust-enabled layer, designed to prevent abuse of resources and to preserve user privacy. In this context, the role of trust is threefold: (i) to support automatic discovery of trusted people, communities, services and resources in general; (ii) to assess what data to disclose to whom and when; and (iii) to facilitate trust-based community membership management, as well as, sub-community formation based on the trust relationships among members of parent communities.

Most trust evaluation systems assume that trust ratings of resources are available, as in the case where users are given the option to rate items, for instance, through like/dislike or star classification schemes. Relying on a rating system that is averaged across all users aiming at a global trust calculus (reputation) cannot be personalised, and is particularly poor in tasks where there is large variation in the items of interest, as in the case of CSSs where the available resources span from individuals to communities and a multitude of services. Contrary to the artificial behaviour imposed by any rating system, trust evaluation should be based on unobtrusive observations of actual user behaviour. Context-awareness (Roussaki et al. 2012), which is a key feature of CSSs, may significantly contribute to this task as a plethora of context data about users and their environment is made available. For example, the user's current activity, interests, preferences, biological/emotional state, agenda, social interconnections can be used to evaluate the user's interactions with individuals, communities and resources. In order to support this, a context-aware trust management and evaluation system must be established. Such a system should enable user interaction monitoring, which heavily relies on a robust context model incorporating all the necessary concepts for efficiently representing the contextual information of both individuals and pervasive communities.

Trust assessment should be based not only on the experiences and evaluation of a user's own interactions, but also on those of other trustworthy individuals and communities. Existing collaborative filtering systems allow making automatic predictions with regards to the interests of a user by collecting preferences or taste information from a multitude of users (Su and Khoshgoftaar 2009). Thus, these systems aim to predict the ratings of individuals based on their past ratings, as well as, based on a set of ratings collected from other users. It should be noted that these predictions are user-specific, as opposed to the simpler approach of assigning an average global reputation score for each item of interest that is applicable for the entire set of users considered.

The trust management and evaluation system presented in this chapter, is suitable for Cooperating Smart Spaces, and is able to infer the user-perceived trust of parties (individuals, communities, or services) based on the user's own interactions (direct trust evidence), as well as, on the trust perceptions of other users (opinions or indirect trust evidence). The collection, processing and evaluation of direct and indirect trust evidence data associated with each party is context-sensitive and inherently personalised aiming to support interconnections with trusted people and resources in a privacy-aware manner.

This chapter is structured as follows. After the introductory section, a literature review on context-aware trust management systems is presented. Section 13.3 describes example use cases where trust information can be exploited by CSSs. Section 13.4 elaborates on the context-based modelling approach that has been adopted to infer user interactions which provide the basis for assessing the trustworthiness of the entities that the user directly engages with. Subsequently, the designed trust management architecture is described, along with the respective trust model employed. Section 13.6 provides an overview of the implemented trust evaluation mechanisms with emphasis on the context-related aspects. Finally, the chapter's conclusions are drawn and the respective future plans are presented.

13.2 Related Work

Trust management has been defined as the "activity of collecting, encoding, analyzing and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding trust relationships" (Grandison 2003). Based on the categorisation proposed in (Beth et al. 1994), trust can be either *direct* or *indirect*. A trust relationship formed from direct experience or interactions can be characterised as direct trust, while a trust relationship built from recommendations by trusted third parties is called indirect trust. While several trust management and evaluation systems have been proposed in the literature, the remainder of this section, focuses on those which utilise context. On the one hand, as context incorporates personal and thus, sensitive user information, trust can facilitate the privacy-preserving mechanisms. On the other hand, exploiting context information can greatly improve the efficiency of the trust evaluation facilities.

In the context-aware trust management system presented in (Abdul-Rahman 2005), each node decides which peers are trustworthy based on information collected from the interactions with other nodes. Furthermore, the model supports trust recommendations, meaning that a third node can be considered trustworthy if it is "proposed" by other trusted nodes. The evaluation is based on specific context parameters but faces scalability issues that restrict its application.

The model proposed by (Wang et al. 2008), is based on a Bayesian network integrating context information from reliable entities in order to decide on the trustworthiness of a new entity or an existing one upon changes in its behaviour. The model is updated periodically based on observations and the system evaluates each (new) entity in due time. Similarly, in (Uddin et al. 2008), context information is collected and specific trust attributes are extracted either directly or indirectly, while context similarity criteria are applied aiming to improve indirect trust calculations. Furthermore, in (Sydow 2008), machine learning approaches are employed in social networking platforms in order to predict the trust levels of entities; however, context parameters are faced statically and are, therefore, difficult to be applied in real and rapidly changing environments.

The design of a context-aware trust management system, suitable for an Internet of Things environment consisting of wireless sensors, is presented in (Chen et al. 2011). More specifically, the trustworthiness for a specific packet forwarding service is evaluated, where trust computation is performed on entity level and disseminated among adjacent entities. This allows for the employment of recommendations to entities which do not perform trust computations, while the context parameters that affect the results include the resources available and other qualitative and quantitative parameters, such as mobility, availability, location.

Metrics to compute the trust level of a node, based on various context parameters, such as cooperativeness as a service provider, community interests and evaluation of trust level recommendations, are proposed in (Bao and Chen 2012). The described system defines a weighting factor to evaluate the trust level received from other nodes. This factor increases proportionally with the global trust level of the node and characterises the entity in its interactions with other entities as well.

The authors in (Ben Saied et al. 2013), propose a system that calculates and periodically reevaluates dynamic trust scores of the entities (nodes) that are bound in some way, taking into account different context attributes, such as status, preferences and past behaviour. The calculations lead to a recommendation score that reflects the trustworthiness of the entity, while the other nodes can either accept or reject the score based on their own thresholds.

In (Kim and Park 2013), a trust management approach for reliable data integration and management in mobile cloud computing environments is proposed. Trustworthiness is measured based on the information collected from the phone call interactions among users and the information is flooded among mobile peers upon request. However, as the authors note, their model needs to be integrated with social and personal information, as well as, environmental data for each user, apart from the basic attributes they have considered.

Finally, trust management in web-based social networks is discussed in (Sherchan 2013). The authors argue that the large number of online social networks emerging on the Web extend the concept of trust with various elements originating from the personal and digital characteristics of each user. Therefore, trust management models designated for social networking should incorporate context information, independently of its dynamic (e.g. preference for a specific status/location/time of day) or static (e.g. the user's hobbies) nature.

Based on the context-aware trust management and evaluation approaches previously discussed, it is evident that many of them have been designed for very specific application domains; others consider only a restricted number of mostly static context attributes, often neglecting the richness of information that can be obtained from existing social networking services. To address these limitations, we propose a Trust Management and Evaluation system which is suitable for a wide range of applications and, by means of the pervasive and social computing facilities of the CSS platform, provides a trust computational model that is context-aware and fully personalised aiming to support interconnections with trustworthy people, communities and resources, while safeguarding user privacy.

13.3 Exploitation Use Cases of Trust Information

A CSS can interact, communicate, or share its resources with other CSSs, communities of CSSs, or services. The degree of collaboration with a given entity should be determined by the trustworthiness of that particular entity. The context-aware Trust Management and Evaluation system presented in this chapter, provides the necessary infrastructure for assessing dynamically changing trust levels of entities with respect to different domains of collaboration. Example use cases, where the assessed level of trust can be exploited, are discussed hereafter.

Trust-Based Grouping CSSs can form groups, i.e. CISs, for sharing context information, services and other resources. This raises substantial privacy considerations as a member of a CIS does not only share personal information with a service, but, at the same time, provides access to this information to other members of the group. From the point of view of a non-member, its decision whether to join a CIS relies on the trustworthiness of group members, as well as, that of the services that are shared among CIS members. On the other hand, the trustworthiness of a CSS requesting to gain membership is also considered prior to acceptance.

Trust-Based Privacy Policy Negotiation Privacy policy negotiation is required in order to define the privacy practices of service providers with respect to user context data. More specifically, the privacy policy associated with a service is evaluated against the privacy preferences of the user and the two parties negotiate on the personally identifying information that needs to be made available to the service in question. The evaluation process relies on the trustworthiness of both the service provider and the service itself.

Trust-Based Service Recommendations The recommender system of a CSS can provide its owner with suggestions on a particular type of service based on their previous choices or those of other CSSs. The factor of trust is crucial in collecting and evaluating these suggestions. Therefore, the Trust Management and Evaluation system supports the assignment of a confidence level to each service recommendation based on the trustworthiness of the party that provides the specific service.

13.4 Context-Based User Interaction Modelling

This section describes the modelling scheme employed by the presented Trust Management and Evaluation system in order to represent user interactions. Trust assessment is not an one-off process; it is rather a continuous activity, whereby an individual reevaluates their experience regarding the entities with which they interact over the course of time. Similarly to the human process of identifying, analysing and assessing interactions with surrounding entities and the environment, a CSS is able to monitor user behaviour with respect to other individuals, communities and services. This heavily relies on a robust context model incorporating all the necessary concepts for efficiently representing the identified user interactions.

The CSS context model, which is suitable for characterising the situation of both individuals and communities, has been detailed in (Kalatzis et al. 2014). Nevertheless, it is worth outlining the main concepts, namely, the `CtxEntity`, the `IndividualCtxEntity`, the `CommunityCtxEntity`, the `CtxAttribute`, and the `CtxAssociation`. These classes comprise the core of the model, which is further enriched by additional meta-data, such as quality characteristics, that mainly address context management requirements. A `CtxEntity` is used to represent an object of the physical or conceptual world, such as a “person” or “service”. The two specialisations of the `CtxEntity`, i.e. the `IndividualCtxEntity` and the `CommunityCtxEntity`, correspond to the notions of CSS and CIS, respectively. The various properties of a `CtxEntity` are modelled as key-value pairs referred to as `CtxAttributes`. For instance, the “name” and the “location” of a “person” `CtxEntity` can be represented as `CtxAttributes`. Furthermore, `CtxEntities` may be interrelated via `CtxAssociations`, such as “isFriendsWith” or “isMemberOf”. All the aforementioned model classes are associated with a timestamp denoting their most recent time of modification.

After having sketched the main CSS context model concepts, we present an example demonstrating how the represented information can be exploited in order to infer the interactions of a user with other individuals, communities and services. The main character in this example is Alice; the context information maintained in her CSS is illustrated in Fig. 13.1. It should be emphasised that this is a highly abstract view of the

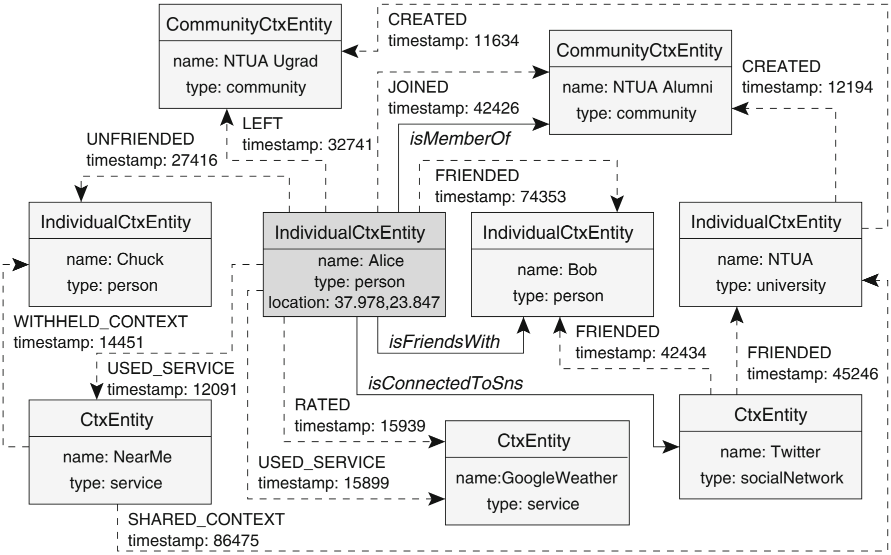

Fig. 13.1 A simplified example of context-based interaction modelling in Cooperating Smart Spaces

actual data. More specifically, the identifiers which are required to uniquely address context data items have been omitted from the depicted `CtxEntities`, while only a minimal set of the accompanying `CtxAttributes` and `CtxAssociations` is included in the diagram. Apart from Alice (highlighted in grey), the illustrated example involves three additional CSSs which, as already mentioned, are represented as `IndividualCtxEntities`. The "type" `CtxAttribute` indicates the type of the CSS owner, thus, Alice, Bob and Chuck are of type "person", while the National Technical University (NTUA) is denoted as "university". The "location" `CtxAttribute` holds the geographic coordinates of Alice's last known location.

Friend relationships among CSSs are modelled via "isFriendsWith" `CtxAssociations`, the modifications of which, are interpreted as `(UN)FRIENDED` interaction events escorted by the respective timestamp. In our example, once the CSS of Alice befriended that of Bob, an "isFriendsWith" `CtxAssociation` was established between them, hence the recorded `FRIENDED` interaction; conversely, when Alice removed Chuck from her list of friends, their "isFriendsWith" `CtxAssociation` was eliminated, leading to an `UNFRIENDED` event. It should be noted that the system is also able to detect friendships, which have been established through external Social Networking Services (SNS). Thus, based on the information from Alice's Twitter account that has been connected to her CSS profile (see the "isConnectedToSns" `CtxAssociation`), two additional `FRIENDED` interactions have been automatically inferred by the system: one with Bob and another with the CSS of NTUA.

In addition to the interactions related to CSSs, Fig. 13.1 depicts two CISs represented as `CommunityCtxEntities`. The "isOwnerOf" `CtxAssociation` and, thus, the respective CREATED interaction indicates the creator of a CIS, which, for both communities in our example, is the CSS of NTUA. Moreover, the CSSs that are members of a CIS can be interrelated through "isMemberOf" CtxAssociations, from which, either JOINED or LEFT interactions can be derived.

With regards to services, these are modelled as "service" CtxEntities. Every time Alice is consuming a service, a USED_SERVICE interaction is recorded. In the case of context-aware services, additional interactions can be inferred which can be better understood through our simplified example. More specifically, Alice is using NearMe, which is a location-based service capable of showing other CSSs in the vicinity. Contrary to Chuck, the CSS of NTUA has been granted access to Alice's location, hence, the SHARED_CONTEXT and WITHHELD_CONTEXT interactions.

Thus far, we have confined our modelling analysis to interactions which are derived from unobtrusive observations of actual user behaviour recorded in the CSS context repository. As described in Sect. 13.6.1, these interactions serve as the basis for evaluating the trustworthiness of the entities which the user engages with. However, when acting on behalf of the user, it is essential that they, too, can be involved in the process. In this respect, the CSS employs feedback mechanisms for allowing the user to rate the entities they interact with. The resulting RATED interactions are not extracted from context information, nevertheless, they have been included in this section to provide a more complete picture of the interaction modelling scheme.

13.5 Context-Aware Trust Management and Evaluation Architecture

This section presents the components of the Trust Management and Evaluation architecture. The functional view of this architecture is illustrated in Fig. 13.2, where the interconnections with Context Management (CM) components have also been depicted. It should be noted that this architecture diagram only includes the CM components which pertain to the functionality of the Trust Management and Evaluation system. A full description of the CM architecture is provided in (Roussaki et al. 2012).

The *Direct Trust Engine* is responsible for evaluating the trust evidence that result from direct interactions among the CSS owner (trustor) and the trusted entities (trustees), in order to estimate the trust level of the latter. A number of factors influence the (re)evaluation of the trustor's direct trust in a certain entity, such as the history of their interactions which includes the number of previous interactions, as well as, the frequency of interactions and their duration. The *Direct Trust Engine* retrieves such information from the *Trust Evidence Repository*, processes it, estimates the direct trust and stores the estimated value in the *Trust Repository*.

It should be highlighted that in many cases there is no direct trust relation between the trustor and the trustee, or the history of direct interactions is limited. However, with the use of trust values from other CSSs, the trustor is able to infer trust. This

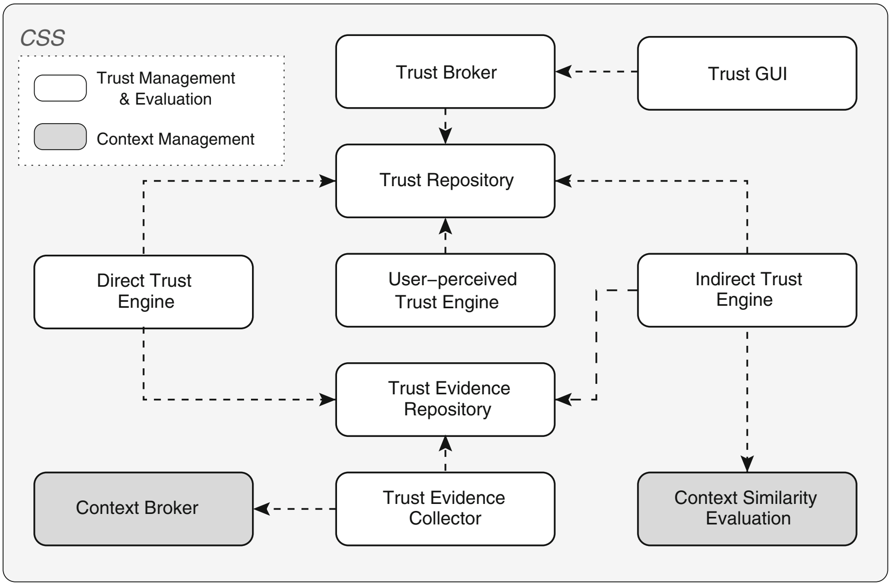

Fig. 13.2 Context-aware Trust Management and Evaluation architecture of Cooperating Smart Spaces

is the responsibility of the *Indirect Trust Engine* which retrieves the indirect trust evidence (trust opinions) from the *Trust Evidence Repository*, processes it, estimates the indirect trust and stores the estimated value in the *Trust Repository*. As detailed in Sect. 13.6.2, each trust opinion is weighted based on the context similarity between the trustor and the entity providing that particular opinion. Thus, the *Indirect Trust Engine* relies on the *Context Similarity Evaluation* component which is capable of quantifying this similarity.

The *User-perceived Trust Engine* is then responsible for fusing the direct and indirect trust values of an entity in order to assess the aggregate value as perceived by the CSS owner. The direct trust value generally outweighs the indirect one in this fusion process. However, the weight of each factor also depends on the confidence level with which it has been estimated. For instance, when the direct trust evidence is not sufficient, the opinions from other CSSs have a greater effect in assessing the aggregate trust value.

As already described, the evaluation of direct and indirect trust in an entity is based on trust evidence. This information can be of various forms and originate from diverse sources, including trust opinions from other CSSs (indirect trust evidence), as well as, locally collected data from direct interactions with services, CSSs and CISs (direct trust evidence). The *Trust Evidence Collector* is responsible for obtaining such information and storing it in the *Trust Evidence Repository*. As far as direct trust evidence are concerned, the respective interactions are modelled as context information (refer to Sect. 13.4) which the *Trust Evidence Collector* is able to access through the *Context Broker*. Regarding the collection of trust opinions, this requires remote communication between the trustor and other CSSs willing to share their trust values. The communication endpoint of each remote CSS Trust Management and Evaluation system is the *Trust Broker*, the functionality of which, is presented hereafter.

The *Trust Broker* acts as a gateway to the trust calculations maintained in the *Trust Repository*. In this respect, it provides both a local and a remote interface through which trust information consumers can specify the `TrustedEntityId` (refer to Sect. 13.5.1) of a particular entity in order to retrieve its direct, indirect or user-perceived trust value. It should be emphasised that trust queries originating from third party services or remote CSSs are subject to access control. Finally, it is worth noting that trust queries can be performed either synchronously or asynchronously. In the latter case, the consumer is notified upon trust value update events.

A subset of the Trust Broker functionality is exposed through a Graphical User Interface (GUI). More specifically, the *Trust GUI* allows the CSS owner to access the trust values evaluated by the system on their behalf. The displayed results can be filtered based on various criteria, while the user can specify a sort order based on the assessed value or time of evaluation. Assigning trust ratings to the users, communities or services which have been evaluated by the system is also supported.

13.5.1 Trust Model

This subsection elaborates on the Trust Model which has been designed in order to allow for the efficient management and exploitation of trust information. As illustrated in Fig. 13.3, the `TrustedEntity` is the core concept upon which the Trust Model is built. This class is used to represent an entity trusted by the trustor, i.e. the owner of a CSS. Each `TrustedEntity` can be referenced by its `TrustedEntityId`, which uniquely identifies the trusted entity. Trusted Entity Identifiers (TEIDs) are formatted as Uniform Name Numbers (URNs).

In addition, every `TrustedEntity` is associated with three trust value representations, namely the `DirectTrust`, `IndirectTrust` and `UserPerceivedTrust`. The `DirectTrust` class is used to represent the direct trust value in a `TrustedEntity`. This value is evaluated based on the experiences from direct interactions between the trustor and the `TrustedEntity`. On the other hand, the `IndirectTrust` class is used to model the indirect trust in a `TrustedEntity`. This value is evaluated based on the recommendations or trust opinions originating from other `TrustedEntities`. Finally, the `UserPerceivedTrust` class represents the trust in an entity as perceived by the trustor. In this respect, its value is evaluated based on an accumulation of the `DirectTrust` and `IndirectTrust` in that entity.

Considering the main types of entities with which a CSS may interact, we have extended the `TrustedEntity` class in order to model their interrelations. More specifically, the `TrustedCis` class is used to represent a CIS and is assigned a set of

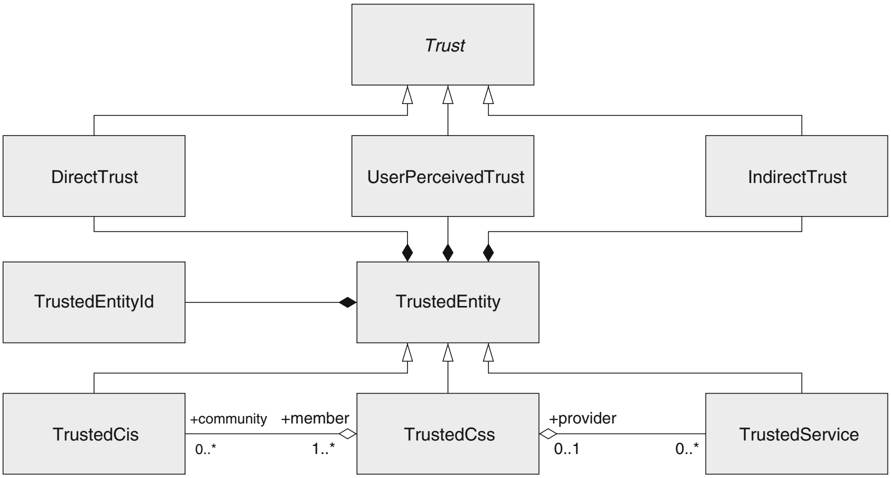

Fig. 13.3 Trust model of Cooperating Smart Spaces

`TrustedCss` objects which correspond to its members. The `TrustedCss` class is, in turn, associated with a set of `TrustedService` objects which correspond to the services provided by the modelled CSS.

13.6 Context-Sensitive Trust Evaluation

The trust evaluation mechanisms that have been designed to support CSSs are described in this section with emphasis on the context-related aspects. More specifically, Sect. 13.6.1 presents the evaluation process regarding the direct interactions extracted from context, while Sect. 13.6.2, elaborates on the role of context in the assessment of trust recommendations from which indirect trust can be inferred.

13.6.1 Direct Trust Evaluation

User interactions with another individual, community, service or resource in general, serve as the basis for evaluating the direct trust in that particular resource. These interactions, as already described in Sect. 13.4, can be inferred through context information which is enriched by the pervasive and social computing facilities of the CSS platform.

The main interactions, along with the related context types, have been included in Table 13.1. Depending on their impact on the evaluated trust, interactions can be classified as either *positive* or *negative*. More specifically, positive interactions (denoted with +) tend to increase the trustworthiness of an entity, while negative

Table 13.1 Context-based interactions and their impact on evaluating direct trust in Cooperating Smart Spaces

Interaction type	Related context type(s)	Impact
`FRIENDED/UNFRIENDED`	`isFriendsWith`[a], `isConnectedToSns`[a]	+/−
`JOINED/LEFT`	`isMemberOf`[a]	+/−
`USED_SERVICE`	`usesService`[a], `lastAction`[b]	+
`SHARED/WITHHELD_CONTEXT`	The type of context that was shared or withheld	+/−
`CREATED`	`isOwnerOf`[a]	+
`RATED`	n/a	+/−

Context Model Types: [a] CtxAssociation, [b] CtxAttribute (cf. Sect. 13.4)

ones (−) have the opposite effect; in fact, the cost incurred by a negative interaction is generally considered higher compared to the gain from a positive one.

Each of the aforementioned interactions is assigned a numeric score and is associated with a timestmap expressing the time it was recorded. Thus, if $\mathbf{x_{u,r}}$ denotes the vector of interactions $x_{u,r}$ recorded between user u and resource r over time, then the direct trust value $d_{u,r}$ assigned to resource r by user u can be calculated as follows:

$$d_{u,r} = \kappa \ f(\mathbf{x_{u,r}}) \qquad (13.1)$$

where function f aggregates the scores which correspond to each interaction and κ is a normalising factor, such that $d_{u,r}$ lies in the range [0, 1]. The highest value in this range denotes full trust, while the lowest one indicates complete distrust.

13.6.2 Indirect Trust Evaluation

The indirect trust value $e_{u,r}$ which is inferred on behalf of user u for resource r, can be calculated as an aggregation of *similar* users' direct trust values $d_{u',r}$ for that particular resource. Thus, indirect trust can be generally expressed as $e_{u,r} = f_{u' \in U}(d_{u',r})$, where f is the aggregation function and U denotes the set of users u' that are most similar to user u and have assigned a direct trust value to resource r. Key in the above calculation is, therefore, determining the similarity between the reference user u and all available users in order to extract the top N relevant users. To acieve this, existing collaborative filtering systems employ various mechanisms, such as Pearson correlation or vector cosine-based similarity (Linden et al. 2003), whereby the user's direct trust values are compared against those of other users. Our system has adopted the cosine-based similarity, the evaluation of which is more efficient in the case of sparse vectors where only non-zero dimensions need to be considered. We will henceforth refer to this notion of user similarity, as *trust-based user similarity*, as it pertains to the users' trust preferences.

However, user similarity evaluation can be significantly improved by exploiting context information, which leads to the notion of *context-based user similarity*. There are many context-related criteria that can be used to evaluate the commonalities of users, such as the following: sharing the same geographic location; having the same or similar preferences; sharing a common belief, idea, or goal; sharing common interests, experiences, background or knowledge; sharing ties of friendship, kinship, membership in physical/virtual communities, or other forms of social relationships, etc. The component responsible for quantifying the level of similarity with regards to the aforementioned context attributes is Context Similarity Evaluation, presented in Sect. 13.5. A detailed description of the context similarity evaluation process is provided in (McGovern 2013).

In addition to trust- and context-based similarity discussed previously, we also consider the direct trust relationship $d_{u,u'}$ between two users as a factor for inferring indirect trust. More specifically, let $\mathbf{d_u}$ denote the vector of all direct trust values assigned by user u, then we express the trust-based similarity with user u' as $simt(\mathbf{d_u}, \mathbf{d_{u'}})$. Likewise, if $\mathbf{c_u}$ is the vector of the context attributes of user u, let $simc(\mathbf{c_u}, \mathbf{c_{u'}})$ represent the context-based similarity. The formula for inferring the indirect trust on resource r can, thus, be rewritten as:

$$e_{u,r} = f_{u' \in U}(d_{u,u'},\ simt(\mathbf{d_u},\ \mathbf{d_{u'}}),\ simc(\mathbf{c_u},\ \mathbf{c_{u'}}),\ d_{u',r}) \tag{13.2}$$

13.7 Conclusions

This chapter presented the Trust Management and Evaluation system employed by Cooperating Smart Spaces, which combine social network computing with pervasive features aiming to facilitate the discovery, connection and organisation of relevant people, communities and things across the physical and digital worlds. The context information that needs to be collected and processed in such a framework, is by nature highly sensitive, as it comprises profile data, preferences, interests, social interconnections, as well as, past, current, and even future activities. Thus, one compelling reason for evaluating the trustworthiness of the involved parties, is to assess what data to disclose to whom and when, and thereby to support user privacy.

In addition, the trust evaluation process itself can benefit from the plethora of context data about users and their environment. The literature review provided in this chapter, substantiates the need to consider context throughout the entire lifecycle of trust information, ranging from the collection of trust evidence to the processing and dissemination of the derived trust levels. The presented architecture is able to incorporate the wealth of context information made available through the pervasive and social computing facilities of the CSS platform. More specifically, it exploits this information in order to extract user interactions which provide the basis for assessing the trustworthiness of the entities that the user directly engages with. Furthermore, when inferring indirect trust based on recommendations made by other users, context information is utilised to identify commonalities with the sources of these recommendations. Thus, the overall trust assessment considers not only the experience and evaluation of a user's own interactions, but also those of other trustworthy and, at the same time, similar individuals, leading to a trust computation model that is inherently personalised and context-aware.

It should be noted that a prototype implementation of the presented system has been evaluated through a series of trials by users from three different domains: a Student community, an Enterprise community, and a Disaster Management community. User trials were conducted in realistic environments and, despite the prototype nature of the implementation and the restricted number of participants, the analysis of the recorded data along with the feedback captured, indicate that the Trust Management and Evaluation system is able to perform well in a variety of CSS usage situations.

The authors plan to further evaluate and test the prototyped system aiming to identify any scalability problems that may arise. The goal is to address such issues, while enabling rapid and reliable trust assessment, even in cases where the number of users considered in the indirect trust evaluation is very large and the number, nature, experience and rating of their respective interactions greatly vary across time. Finally, it is planned to extend this system to support community-assisted trust learning and prediction to be applied for entities that have minimal interactions.

Acknowledgements The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 257493 of the *SOCIETIES* (Self Orchestrating CommunIty ambiEnT IntelligEnce Spaces) Collaborative Project

References

Abdul-Rahman, A.: A framework for decentralised trust reasoning. PhD thesis, University College London (2005)

Bao, F., Chen, I.R.: Dynamic trust management for Internet of Things applications. In: Proceedings of the 2012 International Workshop on Self-aware Internet of Things, ACM, New York, NY, USA, Self-IoT'12, pp. 1–6 (2012). doi:10.1145/2378023.2378025

Ben Saied, Y., Olivereau, A., Azzabi, R.: COACH: A context aware and multi-service trust model for cooperation management in heterogeneous wireless networks. In: Proceedings of The 9th International Wireless Communications and Mobile Computing Conference, IWCMC'13, pp. 911–918 (2013). doi: 10.1109/IWCMC.2013.6583679

Beth, T., Borcherding, M., Klein, B.: Valuation of trust in open networks. In: Proceedings of the 3rd European Symposium on Research in Computer Security. Springer, London, UK, ESORICS'94, pp. 3–18 (1994)

Chen, D., Chang, G., Sun, D., Li, J., Jia, J., Wang, X.: TRM-IoT: A trust management model based on fuzzy reputation for internet of things. Comput. Sci. Inf. Syst. **8**(4), 1207–1228 (2011). doi:10.2298/CSIS110303056C

Doolin, K.: Societies Magazine—Issue 1 Feb 2013. http://www.ict-societies.eu/magazine/ (2013). Accessed 14 May 2014

Doolin, K., Roussaki, I., Roddy, M., Kalatzis, N., Papadopoulou, E., Taylor, N., Liampotis, N., McKitterick, D., Jennings, E., Kosmides, P.: SOCIETIES: Where Pervasive Meets Social. Lecture Notes in Computer Science, vol. 7281. Springer, Heidelberg, pp. 30–41 (2012)

Grandison, T.: Trust management for internet applications. PhD thesis, Imperial College London (2003)

Hansmann, U., Merk, L., Nicklous, M.S., Stober, T.: Pervasive Computing: The Mobile World, 2nd edn. Springer, Berlin (2003)

Kalatzis, N., Liampotis, N., Roussaki, I., Kosmides, P., Papaioannou, I., Xynogalas, S., Zhang, D., Anagnostou, M.: Cross-community context management in cooperating smart spaces. Pers. Ubiquit. Comput. **18**(2), 427–443 (2014). doi:10.1007/s00779-013-0654-2

Kietzman, J.H., Silvestre, B.S., McCarthy, I.P., Pitt, L.F.: Unpacking the social media phenomenon: Towards a research agenda. J. Public Aff. **12**(2), 109–119 (2012)

Kim, M., Park, S.: Trust management on user behavioral patterns for a mobile cloud computing. Cluster Comput. **16**(4), 725–731 (2013). doi:10.1007/s10586-013-0248-9

Lin, K.Y., Lu, H.P.: Why people use social networking sites: An empirical study integrating network externalities and motivation theory. Comput. Hum. Behav. **27**(3), 1152–1161 (2011). doi:10.1016/j.chb.2010.12.009

Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003). doi:10.1109/MIC.2003.1167344

McGovern, J.: Context similarity evaluation: Inferring how users can collectively collaborate together in a pervasive environment. In: Proceedings of The 2013 International Conference on Cloud and Green Computing, IEEE Computer Society, pp. 553–557 (2013). doi:10.1109/CGC.2013.93

Obaidat, M.S., Denko, M., Woungang, I. (eds.): Pervasive Comput and Networking, 3rd edn. Wiley, Chichester (2011)

Roussaki, I., Kalatzis, N., Liampotis, N., Kosmides, P., Anagnostou, M., Doolin, K., Jennings, E., Bouloudis, Y., Xynogalas, S.: Context-awareness in wireless and mobile computing revisited to embrace social networking. IEEE Commun. Mag. **50**(6), 74–81 (2012). doi:10.1109/MCOM.2012.6211489

Sherchan, W., Nepal, S., Paris, C.: A survey of trust in social networks. ACM Comput. Surv. **45**(4), 47:1–47:33 (2013). doi:10.1145/2501654.2501661

Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**, 4:2–4:2 (2009). doi:10.1155/2009/421425

Sydow, M.: Towards context-enriched trust prediction: A proposal. In: Proceedings of the 2008 International Workshop on Combining Context with Trust, Security and Privacy, Trondheim, Norway (2008)

Uddin, M.G., Zulkernine, M., Ahamed, S.I.: CAT: A context-aware trust model for open and dynamic systems. In: Proceedings of The 2008 ACM Symposium on Applied Computing, ACM, New York, NY, USA, SAC'08, pp. 2024–2029 (2008). doi:10.1145/1363686.1364176

Wang, Y., Li, M., Dillon, E., Cui, L.G., Hu, J.J., Liao, L.J.: A context-aware computational trust model for multi-agent systems. In: Proceedings of The IEEE International Conference on Networking, Sensing and Control, ICNSC'08, pp. 1119–1124 (2008). doi: 10.1109/ICNSC.2008.4525384