Pervasive and Mobile Computing 17 (2015) 207–219

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Social context-aware middleware: A survey

Guanqing Liang *, Jiannong Cao

Department of Computing, The Hong Kong Polytechnic University, Hong Kong

ARTICLE INFO

Article history:
Available online 11 December 2014

Keywords:
Middleware
Social context-aware
Survey

ABSTRACT

Social context refers to a set of characteristics associated with multiple users such as social tie and group behaviours. By leveraging users' social context, social context-aware applications are able to provide seamless services accordingly, which creates a tremendous amount of economic and social value. To obtain social context, social context-aware applications need to collect and process various data over heterogeneous hardware and software platforms, which brings critical challenges for application developers. To address the above-mentioned challenges, social context-aware middleware is proposed to offer social tie inference, group detection services and thus facilitating the application development. In this paper, we provide a software architecture that contains the main services provided by a social context-aware middleware. We then systematically survey and classify the existing works on social context-aware middleware. Finally, we discuss open challenges and point out the emerging directions in designing social context-aware middleware.

1. Introduction

Social context refers to a set of information that characterizes multiple users such as social tie, social group or group dynamics [1]. By exploiting social context information, a variety of social context-aware applications are made possible to provide seamless services effectively. Social context-aware applications have the potential to create a tremendous amount of economic and social value. For example, based on social group information, companies are able to optimize advertisement strategy by identifying the influential individuals [2]. It also enables financial companies to recognize money laundering and assess someone's credit more accurately. Furthermore, social group information can help improve the accuracy of the inference of individual's context such as location [3] and activities [4]. Social context also plays an important role in public security and public health. With the information of location-based groups, security department can carry out crowd anomaly detection and criminal analysis [5]; while health department is able to monitor the spread of infectious disease and thus take actions timely [6].

Recently, new opportunities are opened up to obtain social context due to the unprecedented growth of the adoption of sensor-rich smart phones [7], the large scale deployment of a variety of sensor networks and the popularity of social network services [8]. Those ambient physical and cyber sensors are able to collect users' large amount of digital traces, from which users' social context can be inferred [9]. Therefore, researches in context awareness are shifted from a single-system, single-user's perspective towards large-scale ensembles of networked systems interacting with communities of users [10], and thus igniting the research on social context-awareness.

However, significant challenges remain for application developers when implementing social context-aware applications from scratch, due to the complexity and heterogeneity of the underlying platforms. Specifically, developers need to

* Corresponding author.
E-mail addresses: csgliang@comp.polyu.edu.hk (G. Liang), csjcao@comp.polyu.edu.hk (J. Cao).

http://dx.doi.org/10.1016/j.pmcj.2014.12.003
1574-1192/

deal with the challenging issues, including: social tie/group inference, hardware coordination and management, data accuracy and privacy, system reliability and security, etc. [11]. In order to simplify and facilitate the application development, an effective social context-aware middleware platform is necessary to address the issues posed by the underlying platforms. A middleware refers to the software that hides the complexity and heterogeneity of underlying hardware and network platforms, eases the management of system resources, and increases the predictability of application executions. Particularly, social context-aware middleware refers to the middleware that supports social context-aware applications.

The objective of this paper is to systematically survey the existing literatures on social context-aware middlewares, aiming at reviewing what have been investigated and discussing open challenges remaining to be solved. Note that some survey papers about context-aware middleware [12–14] have been published, however, they focuses on single user's context rather than social context. Some researchers have also surveyed the existing mobile social networking middlewares [15,16], whose focus is merely on social tie inference. The contributions of this paper are as follows:

1. We present a systematic review of existing works on social context-aware middleware.
2. We discuss open challenges remaining in social context-aware middleware and point out some future directions.

The rest of the paper is organized as follows. Section 2 presents the requirements of social context-aware middlewares. Detailed middleware services are discussed in Section 3, and cross layer support is presented in Section 4. Section 5 presents classification of social context-aware middlewares and Section 6 points out open challenges and future directions. Section 7 concludes the paper.

2. Requirements of social context-aware middleware

In this section, we focus on discussing the functional requirements of a social context-aware middleware. In particular, we present the requirements posed by application needs and underlying system support.

Social context is defined as *a set of information derived from direct or indirect interactions among people in both virtual and physical world*. Direct interaction contains face to face conversation, video conferencing, etc.; while indirect interaction includes co-locating for a period of time, joining the same event, etc. The main objective of a social context-aware middleware is to provide support for social context-aware applications, through addressing the application needs and offering the underlying system support.

To fully support social context-aware applications, a social context-aware middleware should offer inference services of different social contexts. According to social network analysis [17], social context can be roughly categorized into three levels: actor level, dyad level and subgroups level. Actor level social context refers to individual's social profile, including background, interest, friends, etc.; dyad level social context characterizes the relationship between a pair of users such as social tie strength; and subgroups level social context describes the information related to a social group such as group preference and group dynamics. Thus, we argue that a social context-aware middleware should offer social context inference services in different levels. More specifically, system services of the middleware should include: individual context management, social tie inference, group detection and group context management.

- *individual context management*: is responsible for single user's context modelling, collection, fusion and storage.
- *social tie inference*: specifies the factors that indicate social tie and recognizes social tie between pairs of users.
- *group detection*: provides the mechanisms to cluster users according to the particular metrics.
- *group context management*: focuses on modelling and recognizing contexts that characterizes a group of people.

In addition to meet the application needs, the underlying platform support is required to hide the complexity of the associated operations. In social context-aware applications, inferring users' social context relies on the communication and computation of underlying heterogeneous devices and platforms. Moreover, due to the sensitive nature of social context, it is desirable to have privacy-preserving protocols to control data access and protection mechanisms to ensure system security. Quality of service (QoS) is also an essential service to set priority for different social contexts. Finally, programming abstraction should be considered to offer high-level abstraction for application developers, including abstraction level, programming paradigm and programming interfaces. The components of the underlying platform support are summarized as follows.

- *networking*: provides services to deal with the low-level communication among devices.
- *security and privacy*: offers the mechanisms for data encryption and access control.
- *quality of service*: ensures the quality of transmission and processing of delay-sensitive data.
- *programming abstraction*: defines the abstraction level, programming paradigm and programming interfaces.

According to the requirements of social context-aware middleware as discussed previously, we propose a software architecture for social context-aware middleware as shown in Fig. 1. Typically, a social context-aware middleware should consist of three main components: programming abstraction, system services and cross-layer support [18]. Programming abstraction offers high-level abstraction interfaces for application developers, while system services and cross-layer support provide the implementation that enables the abstraction of middleware. System services mainly support the application deployment and execution. Cross-layer support ensures the system security, privacy and quality of service (QoS). Note that

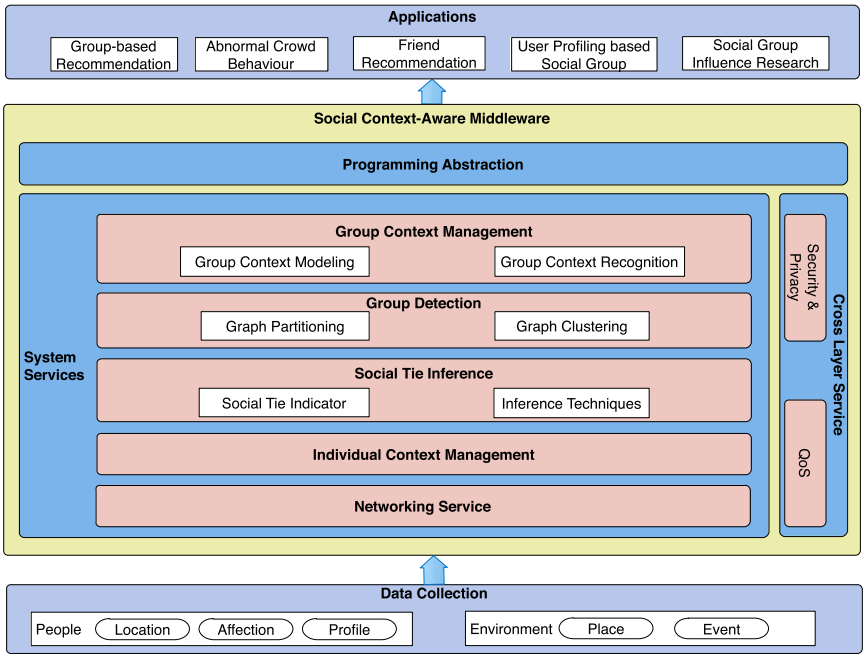

Fig. 1. Software architecture of social context-aware middleware.

not all the system components have to be included in a social context-aware middleware. For instance, group detection and group context management may not be needed, if a social context-aware middleware aims at social tie inference.

Compared with the traditional context-aware middleware [19,20], social context-aware middleware presents some new requirements and challenges. First, in terms of the context aspect, social context-aware middleware aims to infer the context that associates with multiple users, whereas the traditional context-aware middleware only focuses on single user's context. Second, in term of the supporting techniques, scalable techniques for relationship/group modelling and reasoning are required in social context-aware middleware, which may not always be expected in the traditional context-aware middleware.

3. Middleware system services

To conduct social context inference, we need to collect context, represent context and integrate context to recognize higher-level social context. The context collection requires the communication among nodes, which is conducted by networking service and context acquisition component. Context modelling component is proposed to represent context and context reasoning component is responsible for integrating the context to infer higher-level social context. In particular, the main system services support both link-based and group-based context inference. Social tie inference is responsible for modelling and recognizing link-based context (e.g., social relationship strength), while group detection and group context management focus on the group-based context inference. In general, group-based context inference requires the input of link-based context, since the detection of a group is usually based on the connections among users.

3.1. Networking

Networking service enables the communication among devices through running networking protocol over radio channel. There are several radio technologies, including Wi-Fi, ZigBee, Bluetooth, etc. How to select them depends on the application requirements and platform constraints. For example, due to the energy constraints of mobile devices, MobiClique [21] uses Bluetooth for communication. Generally, energy-efficient radio techniques are suitable for the communication among mobile devices. Due to the large-scale deployment of infrastructures, Wi-Fi/Cellular technique supports the communication between mobile devices and server better.

Different networking architectures have been considered in the existing social context-aware middlewares. Most of the middlewares are based on peer-to-peer (P2P) architecture, including: CAMEO [22], MobiClique [21], Prometheus [23], VENETA [24], Whozthat [25] and Yarta [26]. The centralized architecture is adopted in CircleSense [27], MobiSoC [28] and SCIMS [29]. SocialNetwork middleware [30] uses a semi-distributed architecture. Peer-to-peer architecture allows the end users to better control the data privacy. However, it puts a heavy load on individual devices in terms of processing and storage. Compared with peer-to-peer architecture, the centralized architecture is simpler and able to obtain a global picture of social contexts. Furthermore, it relieves the load of individual devices. However, it may bring about serious privacy issues.

On top of the networking architecture, a variety of network protocols have been used for message delivery. In the centralized architecture, networking protocol mainly supports the communication between mobile devices and the server. The communication infrastructure of MobileHealthNet [31] is based on a scalable data distribution layer (SDDL) [32]. SDDL supports two communication protocols: reliable UDP protocol (RUDP) and data distribution service (DDS). The reliable UDP protocol is used for the communication between mobile nodes and the core network, while the data distribution service (DDS) is responsible for the wired communication within the core network. In particular, the reliable UDP protocol provides a publish/subscribe mechanism with an unicast communication between mobile nodes and the gateway, and being able to handle the intermittent connectivity. MobiSoC [28] adopts simple object access protocol (SOAP) to enable the communication between clients and server over HTTP. The reason to choose SOAP is that it offers language independence and its clients are supported by many popular languages. SOAP uses XML information set for its message format. However, when social context is updated quite frequently, SOAP is not efficient due to the verbosity of the protocol.

Within Peer-to-peer (P2P) architecture, different routing and forwarding protocols for opportunistic networking have been designed [33]. CAMEO [22] adopts HiBOp forwarding protocol [34] for multi-hop message delivery. HiBOp is a context-based routing protocol for opportunistic network. At a high level, messages in HiBOp are only forwarded to those nodes with higher probability of reaching the destination. In MobiClique [21], the forwarding protocols rely on two simple rules: (1) messages for a specific user will be sent either by meeting the destination directly or being forwarded via friends of the destination. (2) group messages are flooded within the corresponding group until each member of the group receives the data. VENETA [24] has implemented a Bluetooth based messaging protocol that can deliver messages up to three hops via epidemic routing. Prometheus [23] runs on top of a distributed hash table (DHT)-based overlay called Pastry [35]. In order to forward a message with a key, a Pastry node will route the message to the node whose node identifier is closest to the key numerically.

SAMOA [36] implements a UDP-based protocol which supports point-to-point and multi-point communication. Point-to-point communication enables entities to send messages to a destination with a known IP, while multipoint communication allows SAMOA entities to broadcast a message to other entities in the same place. WhozThat [25] uses Bluetooth for communication among devices and a wide-area wireless Internet connection to support the communication between devices and online social networking site. Yarta [26] exploits iBICOOP middleware [37], which enables communication over different available network interfaces of a device. In particular, communication manager in Yarta [26] can support both synchronous and asynchronous message transmission via multi-radio links. In addition, it supports data transmission over heterogeneous network and is able to handle the temporal disconnection.

In social context-aware applications, devices are mostly mobile and dynamic. Therefore, the networking protocol should be able to handle intermittent connectivity and long-duration disconnection. Among the proposed routing protocols, epidemic routing is not efficient, due to its flooding-based nature. Compared with epidemic routing, context-based routing protocol is a better choice, which leverages the context to select an effective routing path. In order to support the communication between mobile devices and server, reliable UDP protocol (RUDP) is suggested, which achieves the balance between reliability and overhead.

3.2. Individual context management

Individual context management generally provides services for the acquisition, storage, model, fusion and reasoning of single user's context. In this paper, we focus on context acquisition, context model and context reasoning for single user's context. Readers who are interested in context storage and context fusion, can refer to the literature [12].

3.2.1. Context acquisition

There are two mechanisms for context acquisition: event-driven and query-based. The event-driven method first defines the event that specifies certain state changes of the data. After detecting the event, middleware sends an event notification to the applications that are interested in that event. Query-based method allows applications to send queries to obtain interested contexts from server or underlying sensing platforms. The query-based method can provide declarative, SQL-like interface for query operations.

The core system of MobiClique [21] is event-driven. It consists of a single event queue and a set of managers which specify and respond to various system events. System events cover new neighbour discovery or incoming data from the local applications. Note that Mobiclique [21] is implemented based on Haggle architecture. The performance of Haggle architecture in terms of context collection and management is evaluated in paper [38]. Because Haggle kernel is implemented as a single event queue, all the communication messages will be stored before being processed. As a result, the processing time for a request in Haggle depends on the condition of the queue.

In addition to obtain local context, context acquisition needs to collect context information from other nodes. In MobiClique [21], once two mobile devices encounter opportunistically, they exchange the local contexts such as interest and friendship over Bluetooth communication channel. The message can be sent to a specific user or a group. In SCIMS [29], a generic query interface is provided to retrieve social data from different online social networks. Yarta [26] middleware also uses remote queries to acquire contextual information from other nodes. The data can be transferred over heterogeneous network interfaces and connecting technologies. In WhozThat [25], a mobile device uses queries to retrieve contextual

information of another device remotely from online social networking sites, so as to infer social tie strength between users. The proposed protocol in WhozThat [25] can support heterogeneous wireless links and different online social networking sites. In particular, Bluetooth is used for message transmission among users and a wide-area wireless Internet connection is used to retrieve context information from online social networking site.

Event-driven method uses asynchronous communication and is suitable for resource constrained platforms such as sensor networks. Query-based method is better at retrieving data from database such as online social network servers.

3.2.2. Context model

(1) *Key–value model*. This model [39] uses key–value pairs to describe contextual information. This is the simplest technique for context representation. Key–value model can be applied to represent simple social profiles such interest, contact list, etc. However, key–value model has a number of critical drawbacks to model social context which is dynamic and uncertain. First, it cannot model a variety of context types and relationships. Second, it is limited in modelling context uncertainty and supports limited reasoning. Furthermore, it is not scalable when the dealing with a large amount of data.

(2) *Object-role model*. It uses relationships to model context. Context modelling language (CML) [40] is one of the object-role model techniques. CML provides the modelling constructs to describe the types of information, dependencies amongst different types of information and their classification (sensed, static, profiled or derived), etc. CAMEO middleware [22] adopts CML proposed by Henricksen et al. [41] to model context. To represent social context components, CAMEO [22] specifies three object types: Person, Neighbour and Community. In particular, several fact types are defined to model the relationships among those social context components. For example, "Person belongs to Community" is defined to describe the membership of the local user of a home community.

CML can model complex relationships among people, uncertain and historical contextual information. However, CML model does not support hierarchical structure modelling and interoperability well. As a consequence, problems might arise when inferring social context from individual contexts which are modelled in different ways.

(3) *Ontology-based model*. Ontology-based model considers the context as knowledge and makes use of ontology to specify contexts formally and explicitly. Among emerging ontology standards, ontology web language (OWL) [42] and resource description framework (RDF) [43] are widely used. Most of social context models [44,45] adopt ontology-based model. SCIMS [29] middleware models social links between people based on OWL 2 DL. SCIMS [29] specifies both upper and domain-specific ontology. Particularly, upper ontology defines four first-class entities: Person, SocialRole, Relationship and CurrentStatus. On top of upper ontology, SCIMS [29] defines some domain-specific ontologies. For example, fine-grained relationship model defines the ontology for family, education-based friend, work and common-interest friend. CurrentStatus ontology could be extended to represent the status of a people in particular domains such as home, office, shopping and travel. SAMOA [36] middleware represents and stores user's context (e.g., place profiles, user profiles, etc.) using OWL ontologies via the Jena semantic web framework. Resource description framework (RDF) is used in Yarta [26] to model social context. The context model in Yarta [26] can be further extended by defining classes and properties that are related to the base classes.

Ontology-based model supports dynamic relations among entities. Therefore, to a large extent, ontology-based model is suitable to represent social context. However, ontology-based model cannot model the uncertainty of context and the extension of ontology is quite complex.

(4) *Hybrid model*. This model aims to integrate different models to satisfy more complex system requirements. One example is hybrid fact-based/ontological model proposed in [46]. Hybrid fact-base/ontological model combines the ontology-based model with fact-based model, through the mapping from CML modelling constructs to OWL-DL classes and relationships. This hybrid model can exploit the advantages of both models, and thus being able to model ambiguous and imperfect context and provide interoperability support.

Social context describes complex relationships among people, which is dynamic, uncertain and private in nature. Therefore, an effective and efficient social context model should support heterogeneity, interoperability, relationship modelling and uncertainty modelling, scalability and privacy. Based on the evaluation of the existing context modelling techniques, a single model cannot satisfy all the requirements. Clearly, in order to fully represent social context, hybrid model should be considered [47].

3.2.3. Context reasoning

(1) *Rules*. This model uses the IF–THEN–ELSE framework for reasoning, which is the simplest reasoning method [48]. To some extent, the concept of rules mimics the human thinking and reasoning. By specifying a list of rules, high-level social context can be inferred from low-level context. For instance, we can define a rule like "IF A and B are friends AND interaction intensity $>$ 0.8 THEN A and B are close friends.". However, rule-based reasoning method has a number of limitations. First, it cannot model the uncertain or procedural information. Besides, it is difficult to manage when the number of rules becomes very large. Thus, rule-based reasoning is suitable to infer pre-defined social context rather than dynamic social context (e.g., social link strength).

(2) *Fuzzy logic*. Fuzzy logic [49] is an approximate reasoning method. It is able to represent the imprecise concept by defining the degree of membership ranging from 0 to 1. We can say user A and user B are close, with a degree of membership of 0.6. Therefore, fuzzy logic can be used to detect subjective social contexts.

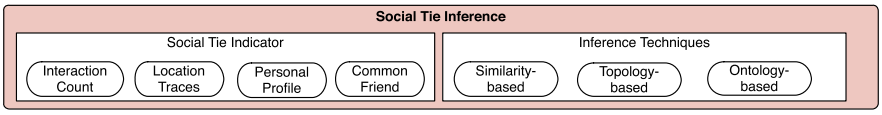

Fig. 2. Social tie inference.

(3) *Probabilistic logic*. Probabilistic logic reasoning [50] associates logical statements with a probability such as "The probability of the existence of social relationship between A and B is 3/4". Dempster–Shafer theory [51] and Hidden Markov Models [52] can be considered as probabilistic logic techniques. Dempster–Shafer theory is a theory of evidence, which combines evidences from different sources to calculate the probability of an event. A Hidden Markov Model represents the hidden state using observable evidence and models the transitions of states with Markov chains. For example, the co-located traces can be considered as the observable evidence, while social relationship is the hidden state. Probabilistic logic reasoning can conduct multi-sensor fusion to combine contexts with different qualities. However, probabilistic logic model needs a significant amount of data to build up the conditional probabilities of an event given a particular evidence and the transition probabilities.

(4) *Ontology based*. Ontology-based reasoning stems from description logic, which is supported by semantic web languages such as RDF and OWL. It can derive new knowledge from the current context automatically based on the given classes and properties. In the existing social context-aware middlewares, Yarta [26], SCIMS [29] and SAMOA [36] use ontology-based reasoning techniques. Specifically, SAMOA [36] exploits description-logic based subsumption reasoning to infer whether two users are similar in terms of the purchased objects. Ontology-based reasoning technique is not capable of handling the incomplete or uncertain information. Moreover, it is computationally intensive. Thus, ontology-based reasoning technique might not be a good choice to infer complex and dynamic social context, since it presents performance issues in terms of computational load, scalability, and processing time with the increasing amount of social data.

(5) *Machine learning*. This model [53] relies on the collected dataset to bridge the gap between raw data and high-level context. Many machine learning techniques have been developed, including: decision tree, Bayesian networks, neural networks, support vector machine and k-mean clustering, etc. CircleSense middleware [27] applies the metric learning technique to recognize social activities based on the co-located user set and the temporal information. Machine learning technique is good at dealing with noisy and incomplete data and able to represent the complex knowledge. However, it heavily relies on the collected dataset and usually requires human efforts to label data.

Every reasoning technique has its own pros and cons, and thus integrating different techniques together could be a promising direction. As suggested in paper [54], different techniques can be used at different levels for context acquisition, aggregation and reasoning. Fuzzy logic could be used to convert the numeric data to be more human-readable. Then Dempster–Shafer theory can be applied to aggregate the data from multiple sensors. Furthermore, high-level context reasoning can be performed using machine learning techniques. At the higher level, ontological reasoning can be employed to extract fine-grained context.

3.3. Social tie inference

Social tie inference infers the social link between a pair of users. Social tie inference consist of two components: social tie indicator and inference techniques as shown in Fig. 2. Social tie indicator contains the information that indicates certain relationship such as common friend [55], common interest [56] or co-location [57]. Based on social tie indicator, inference techniques are used to extract the link among a pair of users. There are three widely used inference techniques: similarity-based, topology-based and ontology-based [58]. Similarity-based method quantifies the relation between a pair of users based on similarity measurement. Topology-based method relies on partially given topology of a social graph to extract social tie, while ontology-based method uses the domain ontology to infer social tie.

Similarity-based method is used in MobiSoC [28] and VENETA [24]. Topology-based method is adopted in Prometheus [23], and ontology-based method is used in SCIMS [29]. In particular, MobiSoC [28] contains a social tie inference component called social state learning. In social state learning component, social tie indicator includes common events, mutual friends, common interests, common places or co-presence. Based on those social tie indicators, social tie strength between every pair of users can be quantified. SAMOA [36] leverages co-location traces to infer social tie. If two users appear in a book store for a period of time, a social link exists between them in terms of interest perspective.

In summary, social tie inference aims to extract relationship between users from both cyber and physical information. However, how to fuse and integrate both cyber and physical information to obtain infer social relationship still remains an open question.

3.4. Group detection

Group detection focuses on clustering users. In the following, the group definition is first discussed, and then two kinds of group detection algorithms are presented in detail.

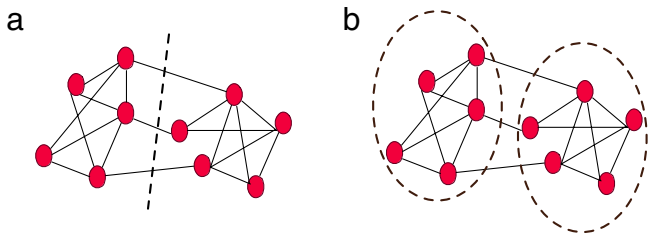

Fig. 3. Group detection algorithms—(a) Graph partitioning; (b) Graph clustering.

3.4.1. Group definition

The starting point of group detection is to define what is a group. There are three main definitions of a group [59]. The first definition of a group is a subgraph with complete mutuality, reachability, high vertex degree and high cohesion. It is also called clique. However, clique might be too strict to reflect the real properties of social group, since complete mutuality may not hold. Therefore, a relaxed version k-clique is proposed, which is a subgraph satisfying the condition that the distance of each pair of its vertices is not larger than k. The second definition of a group is a graph which is different from a random graph. The third definition of a group is a number of vertices similar to each other according to some reference properties.

Most of the existing works on social context-aware middleware adopt the third definition of a group. CAMEO [22] defines a group as a number of users that are similar to each other according to some metrics (e.g., user interest, habits, co-location, etc.). SAMOA [36] defines a group as a number of users who are in physical proximity and share common affinities, attitudes, or social interests. In Whozthat [25], a group refers to a set of co-located people. The first two definitions of a group are usually adopted in online social network analysis, where explicit social links among users are given, while the third definition of a group is considered when there are no explicit links between users.

3.4.2. Partition-based algorithm

To detect groups among a set of users, many group detection algorithms have been designed. As shown in Fig. 3, group detection algorithms can be roughly divided into partitioned-based and clustering-based. The main idea of partition-based algorithm is to cut the edges to extract groups, while clustering-based algorithm includes vertices similar to each other to identify groups.

Partition-based algorithm needs to divide the vertices in a number of groups, so that the number of edges across the groups is minimal. One frequently used method is Kernighan–Lin algorithm [60]. Its basic idea is to optimize a benefit function, which measures the difference between the number of edges inside the groups and the number of edges across them. The complexity of the algorithm is $O(n2 \log n)$, where n denotes the number of vertices.

3.4.3. Clustering-based algorithm

Clustering-based algorithms can be divided into hierarchical clustering, partitional clustering and spectral clustering. Hierarchical clustering can be implemented using agglomerative algorithm and divisive algorithm. Agglomerative algorithm iteratively merges vertices with sufficiently high similarity, while divisive algorithms iteratively removes edges connecting vertices with low similarity. Hierarchical clustering is commonly used in social network analysis, since social network exhibits multi-level structures.

Partitional clustering method can cluster the vertices into k clusters through minimizing a given cost function. The most commonly used technique is k-means clustering [61], whose cost function is the total squared error function. Spectral clustering [62] is another kind of clustering-based algorithm. It first transforms the initial points into another space, and then cluster them using standard techniques. The transformation of points can be done using the eigenvectors of similarity matrices of points. The main insight of spectral clustering is making the cluster of points to be more evident by transforming the initial points.

Most of social context-aware middlewares use agglomerative algorithms for group detection, which iteratively add users into a social group if they are similar. For example, a group discovery algorithm is presented in paper [63] to detect groups based on user meeting frequency and duration. First, the algorithm extracts the potential user clusters according to the pair-wise co-location records. In this phase, all the users that co-locate for a period of time will be considered as a potential cluster. Then some clusters will be eliminated according to the total meeting time and meeting frequency. Finally, small group will be merged into large group if users in small group are the subset of large group and the meeting time of small group is less than the one of large group.

Compared with hierarchical clustering, partitioning-based algorithm might not be a good choice for group detection, because it requires the number of groups or even their sizes to be given. However, those required information normally is unknown and expected to be discovered.

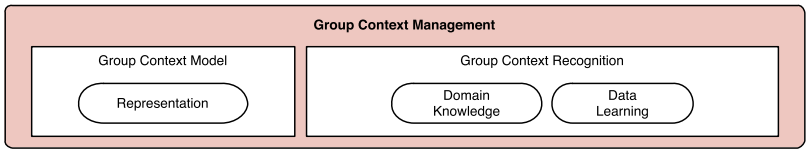

Fig. 4. Group context management.

3.5. Group context management

After group detection, different groups can be identified. Then group context management models and recognizes the context that associates with a group. As depicted in Fig. 4, group context management contains two functions: group context modelling and group context recognition. Group context modelling provides the service to represent group context using a set of features, while group context recognition infers group context based on the features.

3.5.1. Group context modelling

How to model group context depends on the understanding of group context. Actually different sets of features can be used to model the same group context. Take social activity as an example, it could be modelled using location and time. However, spatial or temporal features [64,65] might not be adequate to represent social activities. Therefore, CircleSense middleware [27] proposes a social activity model based on social circle and time. Social circle refers to a set of people that appear together frequently in a social activity.

Group context modelling is critical to the performance of group context recognition. If group context can be modelled in a robust and effective way, then group context recognition can be done using simple techniques. Otherwise, if group context is not well modelled, the performance of group context recognition will be compromised.

3.5.2. Group context recognition

Based on group context modelling, group context recognition obtains high level group context using inference techniques. Generally, group context recognition can be performed by two techniques: data-based technique and knowledge-based technique. Data-based technique recognizes context via learning of dataset. Data-based technique can be further divided into generative learning [66], discriminative learning [67] and heuristic approach [68]. Knowledge-based technique takes advantages of domain knowledge to infer user's context. Knowledge-based approach consists of three categories, including mining-based [69], logic-based [70] and ontology-based [71]. Interesting readers can refer to paper [72] for further details.

In CAMEO [22], group dynamics is detected by analysing the evolution of social interactions over time, which characterizes the presence of local users in different social communities. Whozthat [25] models music preference of a group as the joint songs among members' playlists. In this case, the group context is inferred based on the intersection among individual contexts. CircleSense [27] is a middleware which focuses on modelling and recognizing social activity. CircleSense [27] recognizes social activity using data-based technique. Social activity is first modelled using social circle and time. Then data-based technique is used to conduct social circle classifier training and time classifier training. Finally, both social circle classifier and time classifier are combined to recognize the social activity.

Both data-based and knowledge-based techniques have pros and cons. Data-based technique is good at handling uncertainty and temporal information. However, it requires large amount of datasets for learning, and does not support scalability and reusability. Rather than counting on data, knowledge-based technique is based on rules or patterns extracted from domain experience. As a result, it can be easy to start without dataset, and supports reusability and scalability. But it cannot handle uncertainty and temporal information.

4. Cross layer support

In the following, we will present the existing techniques used for cross layer support in social context-aware middlewares. Cross layer support is essential in a social context-aware middleware to ensure the system security, privacy and QoS. In particular, access control, encryption and cryptographic techniques have been proposed to guarantee security and privacy.

Access control is a privacy protection mechanism which defines the restriction of access to information. In social networking applications, a variety of access control mechanisms have already been proposed, which could be roughly categorized into trust-based, semantic rule-based, role and relationship-based. Trust-based access control mechanism controls someone's access to resources based on trust. Trust metric can be quantified based on relationship type, degree of separation, and trust level between users in the network [73]. Semantic rule-based access control mechanism defines the access policies using semantic web rule language and represent the social knowledge base according to ontology framework [74]. Role and relationship-based access control mechanism leverages the role and relationship to make decisions

Table 1
Classification of social context-aware middleware.

Project	Social context definition	Social context model	Reasoning techniques
CAMEO [22]	The information that is derived from both virtual and physical social interactions among users.	Object-role model	Knowledge-based
CircleSense [27]	The information that characterizes the physical interactions among a group of users.	N/A	Supervised machine learning
MobiClique [21]	The information that characterizes that relationships and interactions among the co-located users.	N/A	N/A
MobileHealthNet [31]	The information that characterizes the virtual interactions among a group of users.	N/A	N/A
MobiSoC [28]	Information that specifies the interactions among people and the interactions between people and places.	N/A	Similarity-based
Prometheus [23]	The information that characterizes the actual social interactions between users.	N/A	Topology-based
SAMOA [36]	The information which characterizes the interactions among a group of people who are in physical proximity.	Ontology-based	Ontology-based
SCIMS [29]	A set of information that is derived from virtual and physical interactions among users.	Ontology-based	Ontology-based
SocialNetwork [30]	The information that characterizes the interactions among users in both virtual and physical worlds.	N/A	Similarity-based
VENETA [24]	The information that characterizes the relationships among users.	N/A	Similarity-based
WhozThat [25]	The information that characterizes the relationships and interactions among the co-located users.	N/A	Similarity-based
Yarta [26]	The information which characterizes the relationships between users who are in physical proximity.	Ontology-based	Ontology-based

for data access and distribution [75]. Access control is enabled in MobiSoC [28], SCIMS [29], Yarta [26] and VENETA [24]. In MobiSoC [28], privacy preferences are expressed in privacy statement which has a primary entity and secondary entity. Primary entity issues the statement and applies it on secondary entity. Yarta [26] and MobileHealthNet [31] use authentication mechanisms for access control, to prevent the unauthorized access.

Each access control method has its own limitations. The main drawback of trust-based method is that it is difficult to specify a fixed trust threshold that works in all different scenarios. Semantic-rule based method requires domain experts to manage the social knowledge base, and it is not scalable to conduct reasoning over the whole knowledge base. Role and relationship-based method requires explicit roles or relationships of users to be given, which may not hold in most cases. In the future, social contexts with higher granularities are expected to become increasingly available, suggesting that role and relationship-based method could be a promising way for access control.

Encryption and cryptography offer secure communication to prevent the leakage of sensitive information. VENETA [24], MobileHealthNet [31] and MobiSoC [28] use certain encryption methods. Particularly, MobiSoC [28] conducts an experiment to evaluate the performance of different encryption methods. The experiment shows that RSA performs the best in providing integrity and authentication for location updates, while AES with or without RSA encryption of its cipher key achieves the best performance to provide confidentiality for larger messages. Some advanced cryptographic techniques have been developed to ensure the privacy of applications that involve multiple participants. In particular, privacy-preserving interest sharing and private scheduling can be achieved, with the assistance of a semi-trust server [76].

Quality of Service (QoS) can provide guaranteed quality level of data service for consumers such as reliable communication, timely message delivery, and data persistency. In MobileHealthNet [31], the messages related to remote monitoring of patients are set high-priority.

5. Classification of social context-aware middleware

In the following, we classify the existing works of social context-aware middleware in terms of six dimensions: social context definition, social context model, social context reasoning techniques, system architecture, system services and cross-layer support. Tables 1 and 2 depict the classification result.

Interestingly, different projects have their own definitions of social context. Some projects define social context as the information that is solely derived from physical/virtual social interaction. However, in some projects, social context refers to the information that characterizes both physical and virtual interaction. In general, the system services are more complex in projects which adopt the second definition of social context, since they need to consolidate information from both cyber and physical world.

The classification result reveals the main focus of current research in social context-aware middleware. First, most of the existing works focus on social tie inference, whereas very few efforts have been contributed to group detection and context management. Specifically, only CAMEO [22], CircleSense [27], SAMOA [36] and WhozThat [25] provide group detection and group context management service.

Table 2
Classification of social context-aware middleware.

Project	Architecture	System services	Cross layer support
CAMEO [22]	Distributed	Group detection; group context management	None
CircleSense [27]	Centralized	Group detection; group context management	None
MobiClique [21]	Distributed	Social tie inference	Privacy
MobileHealthNet [31]	Centralized	Social tie inference	Security; privacy; QoS
MobiSoC [28]	Centralized	Social tie inference	Privacy
Prometheus [23]	Distributed	Social tie inference	Privacy
SAMOA [36]	Distributed	Social tie inference; group detection; group context management	None
SCIMS [29]	Centralized	Social tie inference	Privacy
SocialNetwork [30]	Semi-distributed	Social tie inference	None
VENETA [24]	Distributed	Social tie inference	Privacy
WhozThat [25]	Distributed	Social tie inference; group detection; group context management	None
Yarta [26]	Distributed	Social tie inference	Privacy

Second, very few research efforts have been put on designing appropriate mechanisms for cross layer support. Security and privacy will inevitably become urgent and essential, with the popularity of social context-aware applications. In social context-aware applications, both cyber and physical data about users in different aspects [11] are collected, including location, personal profile, social relationship, etc. These data are inherently sensitive, so that effective and efficient security and privacy mechanisms are desirable at different levels for the data collection, analytics and dissemination.

6. Open challenges and directions

A fair amount of research efforts have been devoted to social context-aware middleware, however, several critical challenges remain to be resolved. Specifically, we highlight four challenging issues: data heterogeneity, group context inference, efficient context recognition and system privacy.

6.1. Data heterogeneity

Due to dynamic user characteristics, different infrastructures and attacks of malicious users, the quality of data varies as to different users and areas. For instance, when the embedded accelerometer sensor of mobile phone is used to collect data of body movement, the data quality actually depends on the placed position of the sensor. Better data quality can be achieved when the sensor is placed in user's pocket rather than on the desk. The data quality also depends on the supporting infrastructures. For example, user's location data can be precisely obtained with dense Wi-Fi infrastructures, while in some indoor areas only with cellular network, user's location data becomes coarse-grained. What is worse, some wrong information might be posed by malicious users, leading to data inconsistency. Therefore, further investigations are needed to measure the data quality and incorporate the data with different qualities.

From various sources, different types of data are collected including: text, picture and voice, etc. However, the problem arises when integrating data with different types for social context recognition. One research direction to integrate different types of data is transforming the data into the same framework, where they are measured with the same metric or represented in a shared ontology. For example, the data collected from email and the data collected from physical interaction can be integrated, as long as they describe a common entity.

6.2. From individual context to group context

Inferring group context from the collected individuals' context is one of the distinctive functions in social context-aware middleware. Group context includes a set of features that characterize a social group. Typically, group context can be extracted by consolidating the individual contexts together [77]. Social group preference, for example, can be obtained by summarizing the individuals' preferences. However, how to bridge the semantic gap between low-level individual context and high-level group context still remains an open challenge. One important research issue is how to resolve the conflicts among users to infer a group context such as group preference. Furthermore, a lot of group contexts (e.g., group emotional state, group interaction [77]) have not been studied and thus require further investigations.

6.3. Efficient context recognition

Most of the social context recognition tasks are computation-intensive. For instance, inferring group dynamics needs to collect and process massive amount of users' location traces, which takes a lot of time. However, real-time social context inference is required in some critical applications such as disease control and public security. Possible remedies for efficient social context recognition include: optimized sampling methods, low-complexity mining algorithms and the parallelism of mining algorithms [78].

6.4. Privacy

Social context is inherently sensitive and thus an effective privacy preservation mechanism is indispensable for a social context-aware middleware. To preserve privacy, many methods have been developed, including data anonymization, data decomposition, data cryptography and access control. Those methods, however, are hard-wired and not suitable in social computing paradigm when users tend to share information spontaneously. Thus, the problem is how to achieve a balance between capability to share and privacy preservation.

One intriguing research direction is context-aware privacy preservation. For example, some personal profiles can be disclosed in a party, while in public place, the access to personal profile will be automatically restricted. Recently, an interaction intensity-aware method [79] is proposed to automatically tune the privacy settings according to the interaction intensity which serves as a relationship intimacy indicator. The more interaction between you and a friend, the more information he/she is allowed to access. Compared with the traditional default privacy settings, context-aware privacy preservation method incurs the minimum additional management overhead while ensuring user privacy. Context-aware privacy preservation presents some interesting insights to make a tradeoff between capability to share and privacy preservation, and deserves further explorations.

7. Conclusion

In this paper, we systematically survey the existing literatures in social context-aware middleware. In particular, we first propose a software architecture that contains the main services provided by a social context-aware middleware. Based on the architecture, we summarize and discuss the existing literatures. Furthermore, we classify the existing social context-aware middlewares according to six perspectives: social context definition, social context model, reasoning techniques, system architecture, system services and cross layer support. Finally, we present the challenges posed by the implementation of a social context-aware middleware and point out the future direction.

Acknowledgements

We thank the reviewers for their insightful comments. The work was supported by the NSFC/RGC Joint Research Scheme No. N_PolyU519/12, Germany/HK Joint Research Scheme No. G-PolyU508/13 and Chinese National 973 project grant No. 2015CB352202.

References

[1] D. Schuster, A. Rosi, M. Mamei, T. Springer, M. Endler, F. Zambonelli, Pervasive social context-taxonomy and survey, ACM Trans. Intell. Syst. Technol. (TIST) (2012).
[2] P. Adams, Grouped: How Small Groups of Friends are the Key to Influence on the Social Web, Pearson Education, 2011.
[3] M. De Domenico, A. Lima, M. Musolesi, Interdependence and predictability of human mobility and social interactions, 2012. ArXiv Preprint arXiv:12102376.
[4] N.D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A.T. Campbell, F. Zhao, Enabling large-scale human activity inference on smartphones using community similarity networks (CSN), in: Proceedings of the 13th International Conference on Ubiquitous Computing, ACM, 2011, pp. 355–364.
[5] Y. Zhiwen, Z. Xingshe, Socially aware computing, Chinese J. Comput. 6 (9) (2010) 51–54.
[6] S. Eubank, H. Guclu, V.A. Kumar, M.V. Marathe, A. Srinivasan, Z. Toroczkai, N. Wang, Modelling disease outbreaks in realistic urban social networks, Nature 429 (6988) (2004) 180–184.
[7] F. Richter, 1.2 billion smartphones. 2012. URL: http://www.statista.com/topics/840/smartphones/chart/653/prevalence-of-selected-features-in-the-global-installed-base-of-mobile-phones/.
[8] eBizMBA Top 15 most popular social networking sites. 2013. URL: http://www.ebizmba.com/articles/social-networking-websites.
[9] D. Zhang, B. Guo, Z. Yu, The emergence of social and community intelligence, Computer 44 (7) (2011) 21–28.
[10] P. Lukowicz, S. Pentland, A. Ferscha, From context awareness to socially aware computing, IEEE Pervasive Comput. 11 (1) (2012) 32–41.
[11] M. Conti, S.K. Das, C. Bisdikian, M. Kumar, L.M. Ni, A. Passarella, G. Roussos, G. Tröster, G. Tsudik, F. Zambonelli, Looking ahead in pervasive computing: challenges and opportunities in the era of cyber-physical convergence, Pervasive Mob. Comput. 8 (1) (2012) 2–21.
[12] V. Raychoudhury, J. Cao, M. Kumar, D. Zhang, Middleware for pervasive computing: a survey, Pervasive Mob. Comput. (2012).
[13] K.E. Kjær, A survey of context-aware middleware, in: Proceedings of the 25th Conference on IASTED International Multi-Conference: Software Engineering, ACTA Press, 2007, pp. 148–155.
[14] D. Romero, Context-aware middleware: an overview, Paradigma 2 (3) (2008) 1–11.
[15] A. Karam, N. Mohamed, Middleware for mobile social networks: a survey, in: 2012 45th Hawaii International Conference on System Science, HICSS, 2012, pp. 1482–1490.
[16] P. Bellavista, R. Montanari, S.K. Das, Mobile social networking middleware: a survey, Pervasive Mob. Comput. 9 (4) (2013) 437–453.
[17] C. Prell, Social Network Analysis: History, Theory and Methodology, Sage Publications Limited, 2011.
[18] M.M. Wang, J.N. Cao, J. Li, S.K. Dasi, Middleware for wireless sensor networks: a survey, J. Comput. Sci. Tech. 23 (3) (2008) 305–326.
[19] A. Forkan, I. Khalil, Z. Tari, CoCaMAAL: a cloud-oriented context-aware middleware in ambient assisted living, Future Gener. Comput. Syst. 35 (2014) 114–127.
[20] H. Pung, T. Gu, W. Xue, P. Palmes, J. Zhu, W.L. Ng, C.W. Tang, N.H. Chung, Context-aware middleware for pervasive elderly homecare, IEEE J. Sel. Areas Commun. 27 (4) (2009) 510–524.
[21] A.K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, C. Diot, MobiClique: middleware for mobile social networking, in: Proceedings of the 2nd ACM Workshop on Online Social Networks, ACM, 2009, pp. 49–54.
[22] V. Arnaboldi, M. Conti, F. Delmastro, CAMEO: a novel context-aware middleware for opportunistic mobile social networks, Pervasive Mob. Comput. 11 (2014) 148–167.
[23] N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, A. Iamnitchi, Prometheus: user-controlled P2P social data management for socially-aware applications, in: Middleware 2010, Springer, 2010, pp. 212–231.

[24] M. Von Arb, M. Bader, M. Kuhn, R. Wattenhofer, VENETA: serverless friend-of-friend detection in mobile social networking, in: IEEE International Conference on Wireless and Mobile Computing Networking and Communications, 2008, WIMOB'08, IEEE, 2008, pp. 184–189.
[25] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, et al., WhozThat? Evolving an ecosystem for context-aware mobile social networks, IEEE Netw. 22 (4) (2008) 50–55.
[26] A. Toninelli, A. Pathak, V. Issarny, Yarta: a middleware for managing mobile social ecosystems, in: Advances in Grid and Pervasive Computing, Springer, 2011, pp. 209–220.
[27] G. Liang, J. Cao, W. Zhu, CircleSense: a pervasive computing system for recognizing social activities, in: PerCom, 2013, pp. 201–206.
[28] A. Gupta, A. Kalra, D. Boston, C. Borcea, MobiSoC: a middleware for mobile social computing applications, Mob. Netw. Appl. 14 (1) (2009) 35–52.
[29] M.A. Kabir, J. Han, J. Yu, A. Colman, SCIMS: a social context information management system for socially-aware applications, in: Advanced Information Systems Engineering, Springer, 2012, pp. 301–317.
[30] S.B. Mokhtar, L. McNamara, L. Capra, A middleware service for pervasive social networking, in: Proceedings of the International Workshop on Middleware for Pervasive Mobile and Embedded Computing, ACM, 2009, p. 2.
[31] A. Teles, D. Pinheiro, J. Gonçalves, R. Batista, F. Silva, V. Pinheiro, E. Haeusler, M. Endler, MobileHealthNet: a middleware for mobile social networks in m-health, in: Proceedings of the 3rd International Conference on Wireless Mobile Communication and Healthcare, Vol. 12, 2012.
[32] L. David, R. Vasconcelos, L. Alves, R. Andre, G. Baptista, M. Endler, A communication middleware for scalable real-time mobile collaboration, in: 2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE, 2012, pp. 54–59.
[33] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: data forwarding in disconnected mobile ad hoc networks, IEEE Commun. Mag. 44 (11) (2006) 134–141.
[34] C. Boldrini, M. Conti, J. Jacopini, A. Passarella, HiBOp: a history based routing protocol for opportunistic networks, in: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007, 2007, pp. 1–12.
[35] A. Rowstron, P. Druschel, Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems, in: Middleware 2001, Springer, 2001, pp. 329–350.
[36] D. Bottazzi, R. Montanari, A. Toninelli, Context-aware middleware for anytime, anywhere social networks, IEEE Intell. Syst. 22 (5) (2007) 23–32.
[37] A. Bennaceur, P. Singh, P.G. Raverdy, V. Issarny, The iBICOOP middleware: enablers and services for emerging pervasive computing environments, in: IEEE International Conference on Pervasive Computing and Communications, 2009. PerCom 2009, 2009, pp. 1–6.
[38] C. Boldrini, M. Conti, F. Delmastro, A. Passarella, Context-and social-aware middleware for opportunistic networks, J. Netw. Comput. Appl. 33 (5) (2010) 525–541.
[39] B. Schilit, N. Adams, R. Want, Context-aware computing applications, in: First Workshop on Mobile Computing Systems and Applications, 1994, WMCSA 1994, IEEE, 1994, pp. 85–90.
[40] T. Halpin, T. Morgan, Information Modeling and Relational Databases, Morgan Kaufmann, 2010.
[41] K. Henricksen, J. Indulska, Developing context-aware pervasive computing applications: models and approach, Pervasive Mob. Comput. 2 (1) (2006) 37–64.
[42] D. Riboni, C. Bettini, OWL 2 modeling and reasoning with complex human activities, Pervasive Mob. Comput. 7 (3) (2011) 379–395.
[43] G. Klyne, J.J. Carroll, Resource Description Framework (RDF): Concepts and Abstract Syntax, 2006.
[44] T. Paul-Stueve, S. Wachsmuth, Towards a social context model and architecture for large-scale pervasive environments, in: 2012 IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops, 2012, pp. 619–624.
[45] G. Biamino, Modeling social contexts for pervasive computing environments, in: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops, 2011, pp. 415–420.
[46] K. Henricksen, S. Livingstone, J. Indulska, Towards a hybrid approach to context modelling, reasoning and interoperation, in: Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning and Management, in conjunction with UbiComp, 2004.
[47] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A survey of context modelling and reasoning techniques, Pervasive Mob. Comput. 6 (2) (2010) 161–180.
[48] C. Choi, I. Park, S. Hyun, D. Lee, D. Sim, MiRE: a minimal rule engine for context-aware mobile devices, in: Third International Conference on Digital Information Management, 2008. ICDIM 2008, 2008, pp. 172–177.
[49] C. Negoita, L. Zadeh, H. Zimmermann, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 (1978) 3–28.
[50] R. Fagin, J.Y. Halpern, N. Megiddo, A logic for reasoning about probabilities, Inform. and Comput. 87 (1–2) (1990) 78–128.
[51] G. Shafer, et al., A Mathematical Theory of Evidence. Vol. 1, Princeton University Press, Princeton, 1976.
[52] L. Liao, D.J. Patterson, D. Fox, H. Kautz, Learning and inferring transportation routines, Artificial Intelligence 171 (5) (2007) 311–331.
[53] C.M. Bishop, et al., Pattern Recognition and Machine Learning. Vol. 1, Springer, New York, 2006.
[54] W. Dargie, The role of probabilistic schemes in multisensor context-awareness, in: Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007, PerCom Workshops'07, IEEE, 2007, pp. 27–32.
[55] X. Yu, A. Pan, L.A. Tang, Z. Li, J. Han, Geo-friends recommendation in GPS-based cyber-physical social network, in: 2011 International Conference on Advances in Social Networks Analysis and Mining, ASONAM, IEEE, 2011, pp. 361–368.
[56] J. Zhuang, T. Mei, S.C. Hoi, X.S. Hua, S. Li, Modeling social strength in social media community via kernel-based learning, in: Proceedings of the 19th ACM International Conference on Multimedia, ACM, 2011, pp. 113–122.
[57] C. Ma, J. Cao, L. Yang, J. Ma, Y. He, Effective social relationship measurement based on user trajectory analysis, Journal of Ambient Intelligence and Humanized Computing (2012) 1–12.
[58] L. Lü, T. Zhou, Link prediction in complex networks: a survey, Physica A 390 (6) (2011) 1150–1170.
[59] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3) (2010) 75–174.
[60] B.W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, Bell Syst. Tech. J. 49 (2) (1970) 291–307.
[61] J. MacQueen, et al. Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, California, USA, Vol. 1, 1967, pp. 281–297.
[62] D.A. Spielmat, S.H. Teng, Spectral partitioning works: planar graphs and finite element meshes, in: 37th Annual Symposium on Foundations of Computer Science, 1996. Proceedings, IEEE, 1996, pp. 96–105.
[63] S. Mardenfeld, D. Boston, S.J. Pan, Q. Jones, A. Iamntichi, C. Borcea, GDC: group discovery using co-location traces, in: 2010 IEEE Second International Conference on Social Computing, SocialCom, IEEE, 2010, pp. 641–648.
[64] N. Eagle, A. Pentland, Reality mining: sensing complex social systems, Pers. Ubiquitous Comput. 10 (4) (2006) 255–268.
[65] V.W. Zheng, Q. Yang, User-dependent aspect model for collaborative activity recognition, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Vol. 3, AAAI Press, 2011, pp. 2085–2090.
[66] M. Brand, N. Oliver, A. Pentland, Coupled hidden Markov models for complex action recognition, in: 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings, IEEE, 1997, pp. 994–999.
[67] L. Bao, S.S. Intille, Activity recognition from user-annotated acceleration data, in: Pervasive Computing, Springer, 2004, pp. 1–17.
[68] V. Guralnik, K.Z. Haigh, Learning models of human behaviour with sequential patterns, in: Proceedings of the AAAI-02 Workshop "Automation as Caregiver", 2002.
[69] M. Perkowitz, M. Philipose, K. Fishkin, D.J. Patterson, Mining models of human activities from the Web, in: Proceedings of the 13th International Conference on World Wide Web, ACM, 2004, pp. 573–582.
[70] S. Carberry, Techniques for plan recognition, User Model. User-Adapt. Interact. 11 (1–2) (2001) 31–48.
[71] D. Chen, J. Yang, H.D. Wactlar, Towards automatic analysis of social interaction patterns in a nursing home environment from video, in: Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval, ACM, 2004, pp. 283–290.
[72] L. Chen, J. Hoey, C. Nugent, D. Cook, Z. Yu, Sensor-based activity recognition, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 42 (6) (2012) 790–808.

[73] B. Carminati, E. Ferrari, A. Perego, Rule-based access control for social networks, in: On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, Springer, 2006, pp. 1734–1744.
[74] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, B. Thuraisingham, A semantic Web based framework for social network access control, in: Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, ACM, 2009, pp. 177–186.
[75] I. Kayes, A. Iamnitchi, Aegis: a semantic implementation of privacy as contextual integrity in social ecosystems, in: 2013 Eleventh Annual International Conference on Privacy, Security and Trust, PST, IEEE, 2013, pp. 88–97.
[76] E. De Cristofaro, A. Durussel, I. Aad, Reclaiming privacy for smartphone applications, in: 2011 IEEE International Conference on Pervasive Computing and Communications, PerCom, IEEE, 2011, pp. 84–92.
[77] I. Roussaki, N. Kalatzis, N. Liampotis, P. Kosmides, M. Anagnostou, K. Doolin, E. Jennings, Y. Bouloudis, S. Xynogalas, Context-awareness in wireless and mobile computing revisited to embrace social networking, IEEE Commun. Mag. 50 (6) (2012) 74–81.
[78] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii, Scalable k-means++, Proceedings of the VLDB Endowment 5 (7) (2012) 622–633.
[79] L. Banks, S.F. Wu, All friends are not created equal: an interaction intensity based approach to privacy in online social networks, in: International Conference on Computational Science and Engineering, 2009, Vol. 4, CSE'09, IEEE, 2009, pp. 970–974.