Rahul Chini Dwarakanath*, Dominik Stingl and Ralf Steinmetz

Improving Inter-user Communication: A Technical Survey on Context-aware Communication

DOI 10.1515/pik-2015-0003

Abstract: Inter-user communication has seen radical changes over the past two decades, owing to the precipitated progress in the field of information and communication technology. These revolutionary changes have brought about the decoupling of location from the actual situation of the users, leaving the users *always online*. Untimely inter-user communication, especially in the work domain, can have undesirable consequences, leading to stress, irritability, or even a complete burnout. Context-aware communication is a viable solution to solve these inherent problems by making inter-user communication aware of the situation of the users. In this paper, we deduce the key aspects of the context-aware communication paradigm by analyzing the related literature with respect to the main requirements for one such solution. In particular, we present a technical survey on the inter-device communication aspects governing the different components of the context-aware communication architecture.

I Introduction

With the advent of modern high-performance mobile devices (e.g. smartphones), and the radical advancements in the field of information and communication technology (ICT), user availability has improved significantly over the past two decades [CHW04, BMT04]. These revolutionary changes have transformed the lifestyles of modern-day society, reducing the rigidity in time and place during the planning of everyday life, and in effect, leaving the society always online. However, these exact changes also pose significant disadvantages, especially to the working society, due to the apparent merging of one's work and private lives. In effect, this leads to a misbalanced work-life ratio, or a *work-life imbalance*. Interruptions at work or disturbances in one's private life can cause adverse effects such as mood changes, irritability, stress [MGK08, Kro11] and consequently, inefficient productivity and even a complete burnout [SLM09, Ala11].

*** Corresponding author: Rahul Chini Dwarakanath:**
E-Mail: Rahul.Chini.Dwarakanath@KOM.tu-darmstadt.de
Dominik Stingl: E-Mail: dominik.stingl@kom.tu-darmstadt.de
Ralf Steinmetz: E-Mail: ralf.steinmetz@kom.tu-darmstadt.de

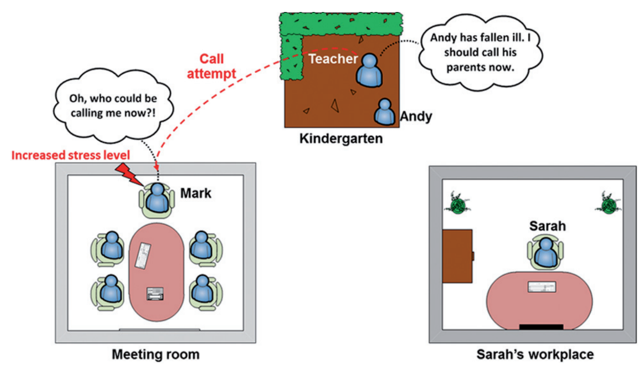

Fig. 1: Illustration of a motivating scenario.

Consider the following scenario described in Fig. 1, which illustrates the fundamental issue in a better manner. Mark is currently sitting in an important meeting with his manager and three other colleagues. Around the same time, Mark's son, Andy, falls ill at the kindergarten. The kindergarten teacher decides to inform Mark about this immediately and attempts to call him. Mark has set his phone to vibration and only *feels* the incoming call from the kindergarten teacher; but out of professional courtesy, he does not answer the call. However, he gets more anxious due to the missed calls. During all this, Mark's wife, Sarah, is sitting alone in her office working on her emails and could very well have taken the call on behalf of Mark.

In general, untimely and/or undesired inter-user communication can lead to adverse consequences, especially in the work domain. The main problem is that inter-user communication nowadays is unaware of the situation that the users experience. From a technical perspective, an improved inter-user communication paradigm, which considers the user context before letting communication take place, is a viable solution to the issue at hand. Over the past few years, there has been a rise in context-aware systems, which basically involve the recognition and interpretation of user context with the help of available sensors in the environment [HGT13, K+11], and adapting application services based on the recognized context [DAS01].

These applications may be designed for either personal (e.g. personal finance, weight check, etc.) or group purposes [AF13]. *Context-aware communication* addresses the *group-based* paradigm, where the context information of the pertinent users in a given communication attempt is assessed to regulate the *communication attempts* towards the users. Here, communication attempts comprise all electronic mediums of communication, including phone calls, SMS, emails, chat services, and push-based notifications. The main goal is to make inter-user communication *situation-aware*.

In the above scenario, by comparing the context information of Mark, Sarah, and the kindergarten teacher, we can determine that the kindergarten teacher's call is important, but Mark is currently *busier* than Sarah. Hence, it would be more appropriate if Sarah receive the call from the kindergarten teacher, by forwarding the initial call to Sarah and notifying the teacher accordingly. In doing so, one such context-aware communication system makes a user less susceptible to unnecessary interruptions, and thereby, provides for a desirable quality of living standard.

In this paper, we present a survey of the existing approaches on context-aware communication, and primarily deal with interconnection of the devices necessary to facilitate its proper functionality. We identify the key user context parameters that are required for one such solution, and consequently, present the principle requirements and challenges that have to be met. Subsequently, we analyze the related literature and classify them based on the established set of system requirements.

The remainder of the paper is structured as follows: Section II looks briefly into the concepts behind user context and describes the context data that are of relevance in context-aware communication. Section III presents the core components of one such architecture, including its main non-functional requirements and challenges. Section IV presents a survey of the related literature, where we present an objective overview from a communications perspective. Finally, Section V concludes the paper, with a brief discussion on possible improvements to the existing approaches for efficient context-aware communication.

II User context

In order to understand the main system requirements, it is essential to clarify what we understand by the term *user context*. One of the most commonly stated definitions in context-based literature is provided by Dey et al. [DAS01], where they define context as "any information that can be used to characterize the situation of an entity". In general, as proposed by Chen and Kotz [CK00], context is a multi-dimensional concept comprising the physical, temporal, computing, and user-related aspects. The physical context represents all the aspects that deal with the real world and that can be accessed with the help of sensors present in the vicinity of a user. This includes information on user location as well as environmental conditions like noise level, temperature, traffic condition, etc. Temporal context comprises the time dimension, with information about the time of the day/week when an activity is performed by a user. The computing context includes information about the computational capabilities of the devices in the system and the resources available (e.g. battery level). Last but not least, the user-related context comprises higher-level information about the user and his surroundings. This includes the user's higher-level activity – which can be partially derived from the physical context of the user (e.g. in a meeting, at a bar, etc.) –, their mental state and preferences, as well as their social setting.

Danninger et al. deploy their user-input based platform called *MyConnector* to analyze how different context parameters, such as a user's calendar entries, location, activity, co-location with other users, physical and mental engagement in the current activity, and the importance and urgency of the activity, have an effect on user availability [DKS06]. One of the logical findings was that user availability is dependent on user location and the current time of the day. With regard to the co-location of other users, the authors found that the users were less available in larger groups than in smaller ones. Information about a user's mental engagement in an activity (e.g. "being in the flow") has a big impact in determining their availability, as expected. Combining the importance and urgency of an activity, together with the mental and physical engagement of a user in the activity, the authors could predict a user's availability in more than half the cases.

In general, a user's context represents the environment or situation experienced by the user at a particular point of time, with the main purpose of determining their preparedness to receive any communication attempt. In the scope of context-aware communication, a user's complete context must include the user's location, current activity, information about the co-located users, device status, user mood and stress level, and environmental conditions, so that a judicious inference of their availability can be drawn.

III Main system requirements and challenges

In general, a group-based context-aware system recognizes user context and adapts the application services provided to the users based on their context. As mentioned earlier, context-aware communication essentially entails the recognition of user context, and consequently, the regulation of the communication attempts towards the users. In this section, we go deeper into the main requirements for context-aware communication, and thereby, elucidate the core components of the same on the basis of our analysis on related literature. We then identify the main challenges that have to be addressed in any context-aware communication architecture.

Key system components

Upon analyzing the existing related literature on general group-based context-aware approaches, we can identify three main system components for context-aware communication – the context recognition component, the context exchange component, and the decision making component, as illustrated in Fig. 2. The context recognition component can be further divided into the following: (a) Sensor data capture, (b) Context reasoning, (c) Context data processing, and (d) Context data representation. In the following subsections, we will describe each of these components in more detail.

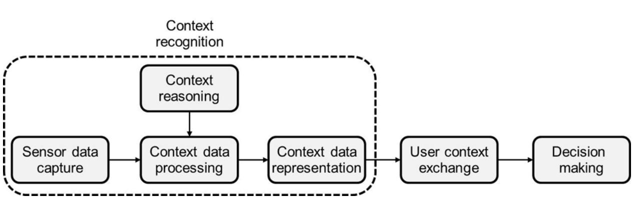

Fig. 2: Component view of a general context-aware communication system.

Sensor data capture: Most commonly, sensors present in user smartphones – comprising hard sensors (e.g. accelerometers, microphones, GPS, etc.) as well as soft sensors (e.g. calendar entries, phone activity, etc.) – and sensors present in the environment – light, humidity, pressure, etc. – as well as wearable sensors – e.g. smart watches and belts – are accessed to obtain necessary data for context recognition. The type of data, sampling frequency, and the length of time the data must be stored depend on the application and the algorithms used for processing these data. The data are obtained either at regular intervals (event-driven paradigm) or are queried for whenever necessary (query-based paradigm) [Z+13].

Context reasoning: Data from the sensors need to be fused together and interpreted in an appropriate manner to obtain the requisite higher-level context information about the given user. Context reasoning algorithms dictate the operations that need to be executed (in a particular order) in order to obtain the final context parameters [HGT13], so that higher-level situational context of a user – for instance, the type of meeting [Lok06], or how stressed a user is – may be interpreted.

Context data processing: The context data processing component facilitates the communication between the various devices in a local environment for the purpose of context recognition. In doing so, it executes the operations of a context reasoning algorithm on the available devices, either individually or through collaboration between the devices [GMB08, M+10, Y+13, HGT13]. This primarily depends on the sources of the data required for these operations, as well as stipulated conditions on e.g. latency, energy, privacy (in the form of user access policies), etc.

Context data representation: It is imperative to consolidate the processed context into an appropriate model such that the complexity of the group-based context-aware system reduces. This uniformity throughout the system allows for feasible sharing and comparison of context information. Simple forms of context modelling include key-value or markup models [Be+10]. Other complex models, based on Object-Role Modelling (ORM), offer a conceptual representation of user context, so as to capture dependencies between context parameters and support queries over ambiguous information [HI06].

User context exchange: The user context exchange component accounts for the transfer of context information between the users involved in a particular conversation. The primary premise behind an exchange is to enable a comparison of the context information of the involved users and potential third parties before arriving at a decision as to whether the concerned communication attempt should be allowed or not.

Decision making: The decision making component is responsible for determining the course of actions concerning any incoming communication attempt, based on the available context information of the relevant communication partners. This component can be realized in the form of an automated rule-based algorithm that adapts to the available context information [S+03, Sch09], which we call *'system-controlled'* approaches. At the same time, the system can allow the users to make the decision by providing them with the requisite context information beforehand [R+05, LH11]. We term these approaches as *'initiator-entrusted'*. For example, if user *A* is provided with the context information of user *B*, user *A* can make an educated judgement of whether user *B* would be in a position to take part in a conversation right now, or not.

Main challenges

Given the above required functionalities, we can now identify the main challenges, which, in turn, specify the key non-functional requirements. In this paper, we focus on the context data processing and context exchange components, since our main focus is the inter-device communication in the grand scheme of context-aware communication.

Robustness against user dynamics: User context is considerably dynamic and subject to constant change with change in user activity and surroundings. User mobility poses a considerable challenge towards the consistency and availability of context information [G+02, Z+13]. User context varies depending on user location, the co-location of other users, as well as the time of the day or week. Varying battery and memory limitations introduce further significant constraints during context processing, especially for resource-constrained devices like user smartphones. One such context-aware system has to be *robust* against these changes so as to support proper functionality.

Privacy-preservation: Privacy-by-design is a very important prerequisite for any context-aware system [KC06, AF13]. Given that user context consists of inherently sensitive information, it is pivotal to *preserve* the same during their processing as well as exchange between users. For example, a user may not want his mental disposition to be computed elsewhere or be divulged to unauthorized users. Also, a user's spouse may receive detailed context information, whereas certain acquaintances may only receive a restricted or distorted (e.g. coarse location information, or just "busy") set. Furthermore, the amount of information divulged may also depend on the time of the day and the location of the user.

Scalability: The system must *scale* well with an increase in the number of user devices as well as the amount of sensor data [H+05, AF13]. On the one hand, the system has to support the context recognition of multiple users, by processing the appropriate sensor data locally as far as possible. On the other hand, the system must support multiple user devices and their requests for the apposite context information at the right time and place.

IV Survey on context-aware systems

In this section, we present a survey and discussion on existing context-aware systems and specifically, the inter-device communication aspects in these approaches. First, we will provide a brief overview on the context-aware communication approaches that have been developed so far, and discuss their main contributions. Next, we present a survey on the context recognition and the context exchange components from a communications perspective. In each of these categories, we analyze the different approaches based on the system requirements presented in the previous section.

General context-aware communication approaches

Context-aware communication in itself is not a completely novel topic. The proliferation of sensors and sensor-based devices in the environment proved to be promising grounds for context-aware systems [DAS01, CK00]. In this subsection, we analyze the existing context-aware communication approaches in literature. We examine these approaches based on the following questions: Which context parameters are sensed and how? What kind of exchange mechanism is employed? And what kind of decision making mechanism has been used?

General overview of existing approaches

One of the first context-aware communication approaches was proposed by Elin Pedersen, called *Calls.calm* [Ped01]. Calls.calm implements a web-based application, where the users can maintain their current context information, as well as contact information and relationship settings. User context information – location (work, home, nearby, away), and social setting (alone, in meeting) – is obtained through manual inputs by the users. Pedersen asserts that user availability should be left to the interpretation of the reader.

Fogarty et al. present the client *MyVine* [FLC04], which accesses a user's laptop and its embedded sensors to process user context and hence, the user availability. The authors mainly consider a typical meeting scenario and analyze speech (using the embedded microphone), location, computer activity, and calendar entries. By recognizing a user's context, *MyVine* helps in managing phone

Table 1: Overview of existing context-aware communication approaches.

	Sensing equipment	Exchange mechanism	Decision-making mechanism
Calls.calm [Ped01]	User input	Proactive	Initiator-entrusted
MyVine [FLC03]	Hard sensors, soft sensors	Proactive	System-controlled
SenSay [S+03]	Hard sensors	Reactive	System-controlled
ContextPhone [R+05]	Hard sensors, soft sensors	Proactive	Initiator-entrusted
ContextFramework.KOM + The Virtual Assistant [Sch09]	Hard sensors	-	System-controlled
Undistracted driving [LH11]	Hard sensors, user input	Reactive, Proactive	System-controlled, Initiator-entrusted

calls, instant messages, and emails. Siewiorek et al. present *SenSay* [S+03], a context-aware mobile phone, which employs multiple sensors like accelerometers, light sensors, and microphones at different parts of the body to recognize a user's context. By analyzing the user posture and ambient conditions, along with additional information (e.g. calendar entries), they attempt to classify user availability into one of four states – active, idle, uninterruptible, and normal – and thereby, take decisions regarding incoming messages on behalf of the user.

Similarly, Raento et al. present a prototyping platform *ContextPhone* [R+05], which employs location sensors and information on user interaction (e.g. computer activity), communication behavior, and the physical environment (e.g. number of Bluetooth devices) to determine a user's context. However, the details of the user recognition process are unavailable and have not been evaluated. *ContextFramework.KOM* [Sch09] is a more recent effort and concentrates mainly on the integration of multiple heterogeneous sensors using an extensive semantic description to obtain the requisite user contexts. Lindqvist and Hong present a system which facilitates undistracted driving [LH11] by using mobile phones to determine a user's basic activity and location. Thereby, they propose to avoid undesired interruptions while driving by predicting instances where a caller may be allowed to communicate with the intended callee.

Context exchange and decision making

Going by the related literature, the context data exchange and decision making processes go hand in hand. As discussed briefly in Sec III, we can distinguish the decision making process into two main categories – *initiator-entrusted* or *system-controlled*. In the former approach, the main goal is to provide communication initiators with the context of the recipients, and thus, entrust them with the decision on the time and means for communication. *Calls.calm*, *ContextPhone*, and *Undistracted Driving* employ this variant (here, *caller-entrusted*). A *proactive* exchange of context information between the users is necessary so that the information is made available to the initiators in advance. All of these systems utilize a server-based approach for storage and exchange of context information (e.g. *ContextPhone* uses *Jabber*).

On the other hand, the *system-controlled* approach has been employed by *MyVine*, *SenSay*, and *Virtual Assistant* [Sch09]. In these particular approaches, there is no exchange of context information between users before an event. Therefore, the decision – as to whether a particular communication attempt has to be put through or not – is solely dependent on the recipient's (e.g. callee's) context information. The context of the communication initiator is not taken into consideration. *MyVine* and *SenSay* do, however, *reactively* convey the callee's availability as well as the decision taken to the caller via SMS.

Table 1 summarizes the above findings and discussion on existing context-aware communication approaches. Most of these approaches entail a proof-of-concept and are intended for a restricted set of scenarios. We can ascertain that the general requirements of scalability and privacy-preservation are not adequately dealt with. In general, these approaches attempt to estimate user availability by using a combination of hard and soft sensors. User inputs are used to obtain higher-level information, such as co-location of other users or their mental and physical engagement in an activity. The decision making process and the exchange mechanism go hand in hand, where either a proactive initiator-entrusted or a reactive system-controlled mechanism has been employed.

Context recognition approaches

For context-aware communication, a user's context is a collection of their location, activity (as well as co-location of other users), mental disposition, available communication mediums, and device status, such that user availability can be assessed suitably. In this section, we delve into existing approaches on context recognition in literature and analyze them with respect to these context parameters. We particularly analyze them from a communications perspective and not from a reasoning perspective. In doing so, we pose the following questions: Which context parameters are sensed and how? Which devices are used for processing the context parameters? And, how are the sensing and processing devices interconnected? From a sensing point of view, we divide the related literature into two simple categories – those which use a single sensing device (mainly, smartphones), and those which use multiple sensing devices.

Approaches using a single sensing device

There is a plethora of related literature where single sensing devices have been used for user context recognition. The focus of these approaches ranges from single context parameters, like user location [S+06, Ba+10, KWM11] or basic activity (sitting, standing, walking, etc.) [GFH09, Yan09], or even a combination of these parameters [M+07, S+09, W+09]. In general, such approaches use a single device to sense and process the data. The authors primarily focus on accuracy of the estimates, and largely neglect energy concerns [G+08, C+09]. Above all, given the restricted set of sensor data, these approaches fail to provide a comprehensive estimate of user availability. In this article, we restrict our focus to a few approaches that address many of the requisite context parameters. We refer the reader to a survey by Hosseini et al. [HGT13] for an analysis of other existing phone-based context recognition approaches.

Miluzzo et al. propose *CenceMe* [M+07], a thin software client that uses embedded sensors like accelerometer, microphone, camera, and GPS present on mobile phones or laptops. Apart from location and basic activity, the mental disposition of the users as well as common user habits are also considered, primarily provided by the user using certain electronic avatars. The devices poll their sensors and push the collected data to a backend server (using a secure channel) for complex processing operations and storage. The authors of *SurroundSense* [ACC09] and *SoundSense* [L+09] use smartphone cameras and microphones, respectively, to recognize the user surroundings. In both these approaches, the processing takes place on the respective user smartphones. Begole et al. propose *LilSys* [BMT04], a single multi-sensor (processing) device, which consists of motion detectors, sound sensors, and other toggle switches, in order to detect the (un-)availability of the users.

Approaches using multiple sensing devices but a single processing device

Keally et al. [K+11] present *PBN* (Practical Body Networking) that interconnects on-body wearable sensors with the embedded sensors in a smartphone to recognize user activity. An Android-based smartphone is used to process the collected data. The main objective of their work is to implement a prototypical version by connecting the smartphones with the body sensors via USB, and to obtain accurate context information with minimal reliance on ground truth. Kern et al. [KSS07] also propose a similar mechanism to obtain user context by placing accelerometers and microphones at different parts of the human body. However, it is not clear where the processing of the collected data takes place.

Xu et al. [X+14] discuss the mobile crowdsensing (MCS) application called *Crowd++* to accurately estimate the number of people speaking in a particular setting. The chief premise of their work is to use the microphones present in mobile phones and leverage this set of information from different phones in a local environment to recognize the speakers, and hence, determine the social setting. It is assumed that the processing takes place at a backend server. In a similar effort, Weppner and Lukowicz [WL13] present a Bluetooth-based approach for collaborative crowd density estimation. Although this approach also involves an interconnection of multiple mobile phones, the main processing operations are executed offline on a backend server.

Approaches using multiple sensing and processing devices

Complementing the collaborative theme adopted above, a few approaches address a distributed processing of context information. Inherently, these approaches use data coming from multiple sensors. They are essentially not specific to any particular context parameters, but primarily focus on processing the sensor data by adhering to the stipulated conditions (e.g. latency, energy, etc.).

Table 2: Classification of existing context recognition approaches.

Using a single sensing device	**Using a single processing device**	**LilSys** [BMT04], **CenceMe** [M+07], **SurroundSense** [ACC09], **SoundSense** [L+09]
Using multiple sensing devices	**Using a single processing device**	[KSS07] **PBN** [K+11], [WL13], **Crowd++** [X+14]
	Using multiple processing devices	**Darwin Phones** [M+10], **SociableSense** [R+11], **ErDos** [VC11], **CQP** [Y+13]

Miluzzo et al. present *Darwin* [M+10], which enables mobile phone sensing of human behavior through collaboration with other phones. Darwin phones pool for models on co-located mobile devices, individually process the context information, and then, perform a consensus among each other for improved reliability. The pooling of models as well as the processing of context information can also occur on backend servers. The authors introduce the concept of trusted servers/devices and allow for authentication mechanisms to improve security. Rachuri et al. present *SociableSense* [R+11], which attempts to capture the interactions and social relations among users in workplaces, and thereby, develop a level of *sociability* for each employee. The main focus of their work is on efficient sensor data extraction, and an efficient processing distribution scheme that dynamically executes computational tasks between the individual smartphones and a backend server. Thereby, they account for conditions on energy, latency, and data traffic. Vallina-Rodriguez and Crowcroft introduce an energy-aware operating system, ErdOS [VC11], which improves the battery life of user mobile devices through proactive resource management and opportunistic access to resources on co-located devices. Their main focus lies in energy conservation and secure transfer of requisite data between users in ad-hoc scenarios.

In an attempt to improve on the aforementioned approaches, Yang et al. devise a CQP framework (**C**ollaborative **Q**uery **P**rocessing) [Y+13] to enable shareable execution of queries between user mobile devices and to avoid processing overhead due to repetition and data transmission. Thereby, they propose to primarily reduce the average energy consumption in the system. The authors divide sensor sources into three types: mobile and personal, mobile and non-personal, and cloud-based and remote, and consider these distinctions during query processing. In doing so, the authors propose different variants of the CQP framework, shifting between local and distributed processing. Among the co-located mobile devices, the authors consider a group leader that initiates the query exchange and takes over most of the processing operations on behalf of the group. Furthermore, a central server provides additional data collected from other devices.

Table 2 presents a parallel classification of the context recognition approaches discussed above. We can distinguish the different approaches based on the number of devices used for sensing and processing the required context information. Typically, modern sensor-rich high-performance smartphones are used quite extensively for both sensing and processing in most approaches. Certain other approaches consider additional sensors (e.g. wearable sensors in *PBN*) to estimate the context information more comprehensively. These approaches particularly focus on the accuracy of the obtained context information, and generally tend to neglect other concerns such as energy consumption or processing latency. Other approaches aim to utilize the processing power of other devices (either neighboring devices or a central server) and involve a collaborative means to obtain the required context information, where they primarily address the energy restrictions of the resource-constrained processing devices.

Approaches for user information exchange

Context-aware communication adapts the incoming communication attempts towards a user in accordance with the recognized user context. In this section, we exclusively look at approaches where an exchange or transfer of context data is a necessity. In general, an exchange of user (context) information is prevalent in many existing group-based context-aware approaches, as recounted by Bellavista et al. [B+12]. Furthermore, we also consider the field of online social networks (OSNs), which presents many approaches that address an exchange of user information over the Internet. We identify scalability and privacy as two of the primary requirements of such systems, as mentioned at the beginning of this article. The two key questions that need to be answered by such information exchange systems are: Where are the (context) data stored? And, how are the data transferred between system entities?

Generic context-aware systems

There are many generic context-aware systems that have been developed to cater to basic services, entailing a distribution of user context and subsequently, their adaptation based on the user context. *CARMEN* [B⁺03], *CoBrA* [CFJ05], and *PACE* [H⁺05] are three prominent examples of such systems. In their paper on *CARMEN*, Bellavista et al. propose a context-aware resource management system that supports automatic reconfiguration of wireless services based on contextual changes. Basically, the authors present a middleware that uses so-called *shadow proxies* or Mobile Agents (MAs), which migrate with the user over the fixed network and in turn, handle user mobility, data asynchronicity, and location-awareness. Chen et al. propose a context broker architecture *CoBrA*, where a set of brokers maintains a shared model of user context and enforces privacy policies during the sharing of contextual information. The chief contribution of their paper is to use semantic descriptive languages like OWL (Web Ontology Language) to model user context and define policy languages for the users. They demonstrate their approach in the form of a prototype for context exchange among the users in a room (e.g. in a meeting). Henricksen et al. propose a more elaborate context-aware middleware system called *PACE*, which addresses the heterogeneity in distributed systems. They particularly focus on the issues such as mobility, fault-tolerance, and privacy. Context data are stored on the user devices themselves as well as on so-called context repositories, which apply user-specified access policies while sharing user context with other users. The context data transfer itself takes place using so-called proxy transmitters that are responsible for a single network. Scalability and general system performance are issues that have been left for future work by the authors.

Location-aware services

Certain other group-based context-aware systems focus primarily on location information of the users and adapt their services accordingly. Barkhuus et al. investigate the sharing of location information between social groups, using a phone-based location sharing application called *Connecto* [B⁺08]. The intended purpose of their application is the sharing of context and location information amongst small groups of friends, where the context mainly comprises the current ringing profile and the duration of time a person has been at a particular location. These user data are uploaded to a central server at regular intervals of time, and correspondingly, the exchange of these data takes place via the central server. The users are also allowed to manually enter their status information (e.g. including a higher-level description of their location). Eugster et al. introduce an integrated middleware called *Pervaho* [EGH08] for mobile context-aware applications. In this paper, the authors focus on proximity-based communication services (e.g. location-based publish/subscribe services), where a client module interacts with the end users, and a centralized and fixed infrastructure undertakes the matching and communication tasks. Furthermore, the authors assert that these tasks can also be performed in an ad-hoc manner between the end users.

Decentralized online social networks

The field of online social networks (OSNs) presents many approaches that focus on the preservation of user privacy by considering a decentralized storage and exchange mechanism. Although these approaches generally suffer from a loss in performance, especially in latency, and are particularly intended for asynchronous transfer of data, they provide a cogent set of mechanisms to satisfy the requirements of scalability and privacy-preservation. Cutillo et al. present Safebook [CMS09], where they exploit real-life trust between people in order to implement data storage and transfer services that preserve user privacy, data integrity and availability. The authors implement a decentralized peer-to-peer (P2P) based system using the concept of distributed hash tables (DHTs) and *Matryoshkas* (concentric circles) in order to facilitate a deterministic addressing and routing mechanism, as well as privacy- and integrity-preserving storage and retrieval of user data. In a similar manner, Shakimov et al. address privacy concerns in centralized services and propose a decentralized framework for OSNs using the concept of a Virtual Individual Server (VIS) [S⁺11]. A VIS is a virtual machine running on a paid cloud computing utility, which provides flexibility in location control by organizing into per-group overlay networks. The authors mainly focus on preserving the privacy of user location information. An-

Table 3: Classification of existing user information exchange approaches.

Centralized storage	**Via central server**		**Connecto** [B⁺08]
Decentralized storage	**Proxy devices (Brokers)**		**CARMEN** [B⁺03], **CoBrA** [CFJ05], **PACE** [H⁺05]
	Individual servers		**Vis-à-Vis** [S⁺11], **Vegas** [DMD12]
	User devices	**Ad-hoc**	**Pervaho** [EGH08]
		Overlay-based	**Safebook** [CMS09]

other approach is offered by Dürr et al. through their system called *Vegas* [DMD12], where they develop a highly restrictive P2P OSN that provides maximum degree privacy, without concerning themselves with the effects on system performance. Users interact with the OSN through one or more client devices, which communicate using *exchangers* that are similar to mailboxes. The authors employ the concept of *Locagrams* to enable a secure transfer of messages between the exchangers. Data storage is performed at so-called *datastores* (in the same vein of a VIS), where users can store their data and set access policies for world-readability.

Group-based context-aware systems and decentralized online social networks have been considered due to their inherent nature of exchanging information between users. We have restricted our focus to only a few pertinent approaches such that we can identify the basic characteristics of (user) information exchange mechanisms. We distinguish between these approaches based on the nature of data storage (centralized or decentralized) as well as data exchange between the users. In general, the preservation of user privacy has a high priority in all these approaches. Table 3 summarizes the above findings and discussion by providing a classification of the existing information exchange approaches.

V Summary and discussion

With the advent of modern high-performance mobile devices (e.g. smartphones), there has been a growing sense of location decoupling from the situation of a user. As a result, the main problem here is that inter-user communication no longer accounts for the factual availability of the users involved in a conversation, be that a phone call, SMS, e-mail, or any of the plethora of messaging applications available nowadays. A viable solution to solve this problem is the context-aware communication paradigm, which regulates the conversations between users based on the context of the users involved. We identified that any context-aware (communication) system primarily consists of a context recognition component, a context exchange component, and a decision making component. The context recognition component can be further described as the following steps: sensor data capture, context reasoning, context data processing, and context data representation. We briefly delved into the key parameters of a user's context, which are necessary for the proper functionality of one such system. We identified scalability, privacy preservation, and robustness as the key non-functional requirements, which, in turn, pose the main challenges towards one such context-aware system. Subsequently, we presented an unbiased critique of the relevant related literature with respect to the above-mentioned components and requirements in context-aware systems, with special focus on the communication aspects.

In general, from the user context recognition point of view, the existing approaches either rely significantly on manual inputs from the user (e.g. *MyConnector*, where the users input their emotional status and/or information about the co-location of other users), or provide an insufficient amount of information about the user's situation, especially the social setting of the user. It would be advantageous to include the co-located users in a given local setting in order to interpret the significance of the gathering and thereby, the availability of the user. The other additional indicators of user availability like the stress level and emotional state (e.g. nervous, calm, worried, etc.) are yet to be estimated in an empirical manner using the available sensors. Of course, the more detailed a user context is, the more complex the processing of these context data becomes. There have to be measures taken to process the requisite context data in an optimal manner by considering the computational limitations of resource-constrained devices (e.g. user smartphones).

We also opine that an appropriate mechanism for user context exchange is necessary to support an efficient functionality in a context-aware communication system. On the one hand, we have approaches, as discussed in Section IV, where a proactive exchange of user context information is employed, where users are provided with the context information of their counterparts beforehand, so that they may assess and interpret the availability of their counterparts by themselves before attempting to communicate with them. We claim that, given the right parameters in a user's context data, we should be able to automatically recognize the availability of the user. By using this information, it would be possible to automate the decision making process by allowing the system to take decisions on behalf of the user, thus reducing the user's susceptibility to interruptions. The exchange/transfer of context information may take place proactively at regular intervals of time or reactively only when a communication attempt is initiated. Although this is matter of trade-off between increased network traffic and consistency, and latency in context availability, the system should allow for both modes of operation. Consequently, it is necessary to have a quick rule-based algorithm for decision making that adapts itself to the available context data of the users involved in the communication attempt.

Acknowledgment: This work has been [co-]funded by the Social Link Project within the Loewe Program of Excellence in Research, Hessen, Germany.

References

Ala11 Gene M. Alarcon "A meta-analysis of burnout with job demands, resources, and attitudes." Journal of Vocational Behavior 79.2 (2011): 549–562.

AF13 Pedro Alves, and Paulo Ferreira. "Radiator: context propagation based on delayed aggregation." In ACM CSCW, 2013.

ACC09 Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. "SurroundSense: mobile phone localization via ambience fingerprinting." In ACM MobiCom, 2009.

Ba+10 Sean J. Barbeau et al. "Travel assistance device: utilising global positioning system-enabled mobile phones to aid transit riders with special needs." Intelligent Transport Systems, IET 4.1 (2010): 12–23.

B+08 Louise Barkhuus et al. "From awareness to repartee: sharing location within social groups." In ACM CHI, 2008.

BMT04 James Bo Begole, Nicholas E. Matsakis, and John C. Tang. "Lilsys: sensing unavailability." In ACM CSCW, 2004.

B+03 Paolo Bellavista et al. "Context-aware middleware for resource management in the wireless internet." Software Engineering, IEEE Transactions on 29.12 (2003): 1086–1099.

B+12 Paolo Bellavista et al. "A survey of context data distribution for mobile ubiquitous systems." ACM Computing Surveys 44.4 (2012): 24.

Be+10 Claudio Bettini et al. "A survey of context modelling and reasoning techniques." Pervasive and Mobile Computing 6.2 (2010): 161–180.

CK00 Guanling Chen and David Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

CFJ05 Harry Chen Tim Finin, and Amupam Joshi. Semantic web in the context broker architecture. Maryland Univ Baltimore Dept Of Computer Science And Electrical Engineering, 2005.

C+09 Igor Crk et al. "Understanding energy consumption of sensor enabled applications on mobile phones." In IEEE EMBC, 2009.

CMS09 Leucio Antonio Cutillo, Refik Molva, and Thorsten Strufe. "Safebook: A privacy-preserving online social network leveraging on real-life trust." Communications Magazine, IEEE 47.12 (2009): 94–101.

CHW04 Mary Czerwinski, Eric Horvitz, and Susan Wilhite. "A diary study of task switching and interruptions." In ACM CHI, 2004.

DKS06 Maria Danninger, Tobias Kluge, and Rainer Stiefelhagen. "MyConnector: analysis of context cues to predict human availability for communication." In ACM ICMI, 2006.

DAS01 Anind K. Dey, Gregory D. Abowd, and Daniel Salber. "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications." Human-computer interaction 16.2 (2001): 97–166.

DMD12 Michael Durr, Marco Maier, and Florian Dorfmeister. "Vegas: A secure and privacy-preserving peer to peer online social network." In IEEE PASSAT and IEEE SocialCom, 2012.

EGH08 Patrick Eugster, Benoît Garbinato, and Adrian Holzer. "Design and implementation of the pervaho middleware for mobile context-aware applications." In IEEE MCETECH, 2008.

FLC04 James Fogarty, Jennifer Lai, and Jim Christensen. "Presence versus availability: the design and evaluation of a context-aware communication client." International Journal of Human-Computer Studies 61.3 (2004): 299–317.

G+08 Shravan Gaonkar et al. "Micro-blog: sharing and querying content through mobile phones and social participation." In ACM MobiSys, 2008.

G+02 David Garlan et al. "Project aura: Toward distraction-free pervasive computing." Pervasive Computing, IEEE 1.2 (2002): 22–31.

GMB08 Jon Gu, Ramakrishnan Mukundan, and Mark Billinghurst. "Developing mobile phone AR applications using J2ME." In IEEE IVCNZ , 2008.

GFH09 Norbert Győrbíró, Ákos Fábián, and Gergely Hományi. "An activity recognition system for mobile phones." Mobile Networks and Applications 14.1 (2009): 82–91.

HI06 Karen Henricksen and Jadwiga Indulska. "Developing context-aware pervasive computing applications: Models and approach." Pervasive and mobile computing 2.1 (2006): 37–64.

H+05 Karen Henricksen et al. "Middleware for distributed context-aware systems." On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE. Springer Berlin Heidelberg, 2005. 846–863.

HGT13 Seyed Amir Hoseini-Tabatabaei, Alexander Gluhak, and Rahim Tafazolli. "A survey on smartphone-based systems for

opportunistic user context recognition." ACM Computing Surveys 45.3 (2013): 27.
K+11 Matthew Keally et al. "Pbn: towards practical activity recognition using smartphone-based body sensor networks." In ACM SenSys, 2011.
KSS07 Nicky Kern, Bernt Schiele, and Albrecht Schmidt. "Recognizing context for annotating a live life recording." Personal and Ubiquitous Computing 11.4 (2007): 251–263.
KC06 Ashraf Khalil, and Kay Connelly. "Context-aware telephony: privacy preferences and sharing patterns." In ACM CSCW, 2006.
Kro11 Matthias Kropff "Sensorbasiertes Monitoring zur kontextsensitiven Unterstützung von Wissensarbeit", Ph.D. Thesis, TU Darmstadt, Darmstadt, Germany, 2011.
KWM11 Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. "Activity recognition using cell phone accelerometers." ACM SigKDD Explorations Newsletter 12.2 (2011): 74–82.
LH11 Janne Lindqvist and Jason Hong. "Undistracted driving: a mobile phone that doesn't distract." In ACM HotMobile, 2011.
Lok06 Seng Wai Loke "On representing situations for context-aware pervasive computing: six ways to tell if you are in a meeting." In IEEE PerCom Workshops, 2006.
L+09 Lu Hong et al. "SoundSense: scalable sound sensing for people-centric applications on mobile phones." In ACM MobiSys, 2009.
MGK08 Gloria Mark, Daniela Gudith, and Ulrich Klocke. "The cost of interrupted work: more speed and stress." In ACM CHI, 2008.
M+07 Emiliano Miluzzo et al. "CenceMe–injecting sensing presence into social networking applications." Smart Sensing and Context. Springer Berlin Heidelberg, 2007. 1–28.
M+10 Emiliano Miluzzo et al. "Darwin phones: the evolution of sensing and inference on mobile phones." In ACM MobiSys, 2010.
Ped01 Elin Rønby Pedersen "Calls. calm: enabling caller and callee to collaborate." In ACM CHI, 2001.
R+11 Kiran K. Rachuri et al. "Sociablesense: exploring the trade-offs of adaptive sampling and computation offloading for social sensing." In ACM MobiCom, 2011.
R+05 Mika Raento et al. "ContextPhone: A prototyping platform for context-aware mobile applications." Pervasive Computing, IEEE 4.2 (2005): 51–59.
S+09 André C. Santos et al. "Context inference for mobile applications in the UPCASE project." MobileWireless Middleware, Operating Systems, and Applications. Springer Berlin Heidelberg, 2009. 352–365.
SLM09 Wilmar B. Schaufeli, Michael P. Leiter, and Christina Maslach. "Burnout: 35 years of research and practice." Career development international 14.3 (2009): 204–220.
Sch09 Johannes Schmitt "Anpassungsfähige Kontextbestimmung zur Unterstützung von Kommunikationsdiensten", PhD Thesis, TU Darmstadt, Darmstadt, Germany, 2009.
S+11 Amre Shakimov et al. "Vis-a-vis: Privacy-preserving online social networking via virtual individual servers." In IEEE COMSNETS, 2011.
S+03 Daniel Siewiorek et al. "Sensay: A context-aware mobile phone." In IEEE ISWC, 2003.
S+06 Timothy Sohn et al. "Mobility detection using everyday gsm traces." UbiComp 2006: Ubiquitous Computing. Springer Berlin Heidelberg, 2006. 212–224.
VC11 Narseo Vallina-Rodriguez, and Jon Crowcroft. "ErdOS: achieving energy savings in mobile OS." In ACM MobiArch, 2011.
W+09 Yi Wang et al. "A framework of energy efficient mobile sensing for automatic user state recognition." In ACM MobiSys, 2009.
WL13 Jens Weppner and Paul Lukowicz. "Bluetooth based collaborative crowd density estimation with mobile phones." In IEEE PerCom, 2013.
X+14 Chenren Xu et al. "Crowdsensing the speaker count in the wild: implications and applications." Communications Magazine, IEEE 52.10 (2014): 92–99.
Y+13 Jin Yang et al. "Energy-efficient collaborative query processing framework for mobile sensing services." In IEEE MDM, 2013.
Yan09 Jun Yang "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones." In ACM IMCE, 2009.
Z+13 Daqiang Zhang et al. "Survey on context-awareness in ubiquitous media." Multimedia tools and applications 67.1 (2013): 179–211.

Rahul Chini Dwarakanath: Technische Universität Darmstadt – FG Multimedia Kommunikation (KOM), Darmstadt, Hessen, Germany

Dominik Stingl: Technische Universität Darmstadt – FG Multimedia Kommunikation (KOM), Darmstadt, Hessen, Germany

Ralf Steinmetz: Technische Universität Darmstadt – FG Multimedia Kommunikation (KOM), Darmstadt, Hessen, Germany